Revision 4 Aug 2025

THE PRAGMATIC SYSTEMS ENGINEER

Hard Earned Lessons on Systems Engineering from the Ages, For the Ages

Abstract

The Pragmatic Approach to Systems Engineering and Program Management - The Origins of Systems Engineering, How to Create a Successful Engineering Organization, Systems Engineering Processes and Procedures, Related Engineering Activities and A View of What's Coming. Many Case Studies

Brian L. Shaub

Brian.shaub@bluepiedmont.com

This Book Belongs To _____

List of Changes

Off Line or printed documents are uncontrolled and may not include Changes

(This is an example of a change table typical of any SE or PM Document)

#	Change	File Name	Date MM/DD/YY	By
0.1	First DRAFT		04/21/20	BLS
0.2	Formatted and Reorganized		05/10/20	BLS
0.3	Fix Typos, Graphics, Final Draft	SE Book 05_12_0907	05/12/20	BLS
0.4	Added DCIM, ISO 21000	SE BOOK 05_14_1025	05/14/20	BLS
0.5	More definition for the 5th Era			BLS
0.6	Finish GVHS Case Study	SE BOOK 08_5_2020	08/07/20	BLS
0.7	Pub Review Board Approval – email from RUTHTM Wed Jan 13th, 2021 5:01:14PM	SE Book 08_05_2020	01/13/21	BLS
0.8	Typos Fixed	SE Book 08_02_2021	08/02/21	BLS
1.0	Copyright Approved - US Copyright Office Aug 26th 2021	SE Book 10_11_2021	10/11/21	BLS
1.1	Fix Typos, clarify content	SE Book Rev 4 Aug 2025	8/4/25	BLS

Executive Summary

Thank you for your interest in *The Pragmatic Systems Engineer*. Whether you intend to read the entire book, that which is of interest, or to simply browse, your time is appreciated.

I used my experiences, my team's experiences and other observations as source information to write this book. It also includes the typical academics of methods, frameworks, models used by Systems Engineers, Program Managers and those in Business Development. On the next page I have listed the topics that are likely different from the textbook systems or program management books. I have many references but no original source empirical information. Your role, as a leader and manager are to coach your team (often leading your own managers) to a better way. That's why it starts with Section 0 that includes how to organize, lead, manage and communicate. Section 1 discusses What, Who and Why of Systems Engineering. Section 2 is the Systems Lifecycle – not one type fits all so carefully devise your lifecycle for your system. Section 3 gets into more detail – Processes and Approval Gates – that everyone has the same game plan. Section 4 is about the creation and use of standards. It is how Systems Engineers collaborate to solve and express problems, and deliver systems. Section 5 includes a brief summary of supporting skills and 'tricks of the trade'. Section 6 is the future of Systems Engineering to which I took some liberties. Common threads run through this book such as the **Reductionist** and the **Holistic** Thinker; **People, Process, Tools; Cost, Schedule, Performance and Risk;** and how all this works with other disciplines. The last part of this book contains several case studies that I use to bring out the topics of this book.

I served in the U.S. Air Force for twenty-two years and retired as a Communication-Computer Chief Master Sergeant, then the last 20 years as an industry contractor. I am still in awe of how far we've progressed from teletype tape, mainframes, client server and small computers, the internet, cellular networks, autonomy, machine learning and now to The Cloud. In my efforts to learn more, I earned two Master's Degrees from the George Washington University – in Telecommunication (2001) and Systems Engineering (2011). I have successfully managed several programs, worked extensively as a Systems Engineer, and in Business Development. I am a Certified Systems Engineering Profession (CSEP) (2015) from INCOSE$^{(T)}$, and a Project Manager Professional (PMP) (2001) from PMI$^{(T)}$. All of these are not simply 'checking the box' – I earnestly aspired to learn more of the skill by study, hard knocks, coaching, training when available and finally to successfully run programs and build systems.

Now I want to let others know that we can be successful in delivering systems. I want to bundle my many years of experience, participating and leading the growth of Systems Engineering to applying it on today's, more complex interwoven systems.

Thank you to my co-workers who inspired me to take on this project. Thank you to my family (Charlene and Phyllis) for their help and encouragement. I would also like to thank the George Washington University and INCOSE for the opportunity to learn more. I am grateful to United States Air Force for allowing me to work in a culture of professionals who ensure focus by asking – 'what are you doing and why are you doing it?'.

What is Different About This Book from Other Systems and Program/Project Management Books

- Starts over 5K Years Ago
- Marks the 'Era of Enlightenment' and 'The Fifth Era' 'The Forth Industrial Revolution'
- The Thread of Reductionist and Holistic effecting us today
- Importance of 'soft skills'
- Special Emphasis on 'Pain Points'
- Use of Modeling and Simulation for Schedules and Budgets
- Blending of SE Models, Methods, Lifecycles
- People, Process, Tool, and Inputs
- How to Use Systems Engineering in Business
- Three Generations of Systems Engineering

Contents

List of Changes i
Executive Summary ii
What is Different About This Book from Other Systems and Program/Project Management Books iii
0.0. Milestone Zero – Before the Beginning 1
0.1 We First Learned to Count 1
0.2. The Organization 3
0.3. Leaders, Managers and Administrators 3
0.4 Effective Communication 4
0.5. Its Personal 5
0.6. Style Guide 5
0.6.1. Colors 5
0.6.2. Fonts 5
0.6.3. Terms 6
0.6.4. ICONS 7
0.7. Learning Model 7
1.0 Systems Engineering Fundamentals 9
1.1. The Origins of System Engineering 9
1.2. What is a System 9
1.3. What is Systems Engineering 10
1.4. Why Perform System Engineering? 12
1.5. Who Are System Engineers? 14
1.6. A Stakeholders Point of View 15
1.7. And Your Job is 15
1.8. Useful Paradigms as Mental Models 16
1.8.1. Cost, Schedule, Performance and Risk 17
1.8.2. People, Process, Tools 17
1.9. The Readiness of Technologies (Science and Engineering) 18
1.10. Summary - Systems Engineering Fundamentals 19
2.0 The Systems Lifecycle 20
2.1. The Systems Vee – A Framework 21
2.2. Transform the Systems Vee to a Linear or Circular Form – Now a Method 23
2.2.1. Circular Lifecycle - Framework 23

2.2.2. Oval Lifecycle - Framework 24
2.2.3. Incremental (Spiral) Method 25
2.2.4. Agile Development Method 25
2.2.5. Seven Software Phase Model 27
2.2.6. Waterfall Model 27
2.2.7. Linear 28
2.2.8. Business Technology Model 28
2.2.9. Parallel 29
2.2.10 National Aeronautics Space Administration (NASA) Space Flight Lifecycle Model 30
2.2.11. Department of Defense Lifecycle Framework 31
2.3. Summary - Lifecycle Models, Methods and Frameworks 32
3.0. Processes 33
3.1. Why Process 33
3.1.1. Decision Gate Review Process 35
3.1.2. A Process - Use a Common Game Plan 38
3.1.3. Process Template 39
3.1.4. Manage Project 40
3.1.5. Manage Schedule 42
3.1.6. Risk, Issue and Opportunity Management 44
3.1.7. Requirements Management 48
3.1.8. Configuration Management/ Document Management (CM/DM) 54
3.1.9. Integration 57
3.1.10. Create Architecture (Architecting) 59
3.1.11. Conduct Trade Study – It's Good to Have Options 62
3.1.12. Manage Defects and Improvements 64
3.1.13. Verification and Validation (VER&VAL) (V&V) 66
3.1.14. Conduct Project Management Review (PMR) 69
3.1.15. Conduct Technical Exchange Meeting (TEM) 71
3.2. Other Processes 72
3.3. Best Practices 74
3.4. Summary - Putting It All Together 74
4.0 Say What You Mean, and Mean What You Say 76
4.1. Languages for Systems Engineers 76

4.1.1. Department of Defense Architectural Framework (DoDAF) 76
4.1.2. Integrated Definition Methods (IDEFx) 78
4.1.3. Uniform Markup Language (UML)™ 80
4.1.4. Systems Modeling Language (SysML™) (SysML, n.d.) 80
4.1.5. U.S. Military Standards 81
4.1.6. Architectural Description Language - ISO/IEC 42010 82
4.1.7. C4 Model 82
4.1.8. Summary - Systems Engineering Languages 83
4.2. Guiding Bodies and Standards Bodies 83
4.2.1. Defense Acquisition University (DAU) 83
4.2.2. Project Management Institute (PMI [(T)]) 83
4.2.3. International Council on Systems Engineering (INCOSE) [(T)] 84
4.2.3. International Standards Organizations (ISO) - 9000 Quality Management System 84
4.2.4. Capability Maturity Model Infrastructure (CMMI) 85
4.2.5. Institute of Electrical and Electronics Engineers (IEEE) 85
4.2.6. American National Standards Institute (ANSI) 85
4.2.7. International Telecommunications Union (ITU) 85
4.2.8. Information Technology Service Management (ITSM) 85
4.2.9. Security Management - Cybersecurity 86
4.3. Summary of Section 4 86
5.0 Supporting Skills – The Tricks of the Trade 87
5.1. More Skills 87
5.1.1. Systems Analysis 87
5.1.2. Operations Research 87
5.1.3. Usability Engineering/Human Systems Integration 87
5.1.4. Human Factors Engineering 87
5.1.5. Environmental, Occupational Health, and Safety (EOHS) 87
5.1.6. Thermal Engineering 87
5.1.7. Information Systems Security Managers (ISSM) 88
5.1.8. Critical Support 88
5.2. Quality Management 88
5.2.1. Everyone Wants Quality 88
5.2.2. Quality is Defined by the Customer 88

5.2.3. House of Quality – The Voice of the Customer 88
5.2.4. Just Give Me Quality! 90
5.2.5. Quality Manager 90
5.3. Model Based Systems Engineering 90
5.4. Techniques 91
5.4.1. Continual Improvement 91
5.4.2. Decision Tree Analysis – probability of a chain of events occurring 92
5.4.3. Linear Algebra 92
5.4.4. Root Cause and Corrective Action (RCCA) 93
5.4.5. Poised to Deliver 94
5.5. How People Think and Learn 95
5.5.1. Learning Model 95
5.5.2. Reasoning 95
5.5.3. Predictive Analysis 96
5.5.4. Artificial Intelligence 97
5.5.5 Managed Services 97
5.5.6. Black Swans 97
5.6 Summary of Section 5 – Supporting Skills 98
6.0 What's Next in Systems Engineering 99
6.1. The Triad of the Systems Business 99
6.2. Systems Engineering in Growing a Business 100
6.3. Three Generations of Systems Engineering 102
6.3.1. First Generation of Systems Engineering -- 1920s to 1970s 102
6.3.2. Second Generation of Systems Engineering – 1970s to 2000s 102
6.3.3. Third Generation of Systems Engineering – we are in it 103
6.4. The Digital Information Eco-System 104
6.5. Summary of The Pragmatic Systems Engineer 106
Appendix A Systems Engineering Checklist 108
Appendix B. Case Studies – Presentations and Solutions 109
B-1 Case Study – Driving in Italy 109
B-2 Case Study – Volts and Bolts 110
B-3 Case Study – Corp of Discovery – Transport 111
B-4 Case Study – Green Mountain 113

B-4.1 PART 1 – From the GVHS Perspective 113
B-4.1 Discussion – Project Start Up – How do you start a project? 114
B-4.2 Discussion – Initial Planning – What activities are performed for initial planning? 114
B-4.3 Discussion – CONOPS – What are the challenges and issues as outlined in the brief CONOPS? 115
B-4.4 Discussion – The Problems 116
B-4.5 Discussion – Your Team –What are the 'people' challenges and tasks? 116
B-4.6 Discussion – Make or Buy – What and How do you do this? 116
B-4.7 Discussion – Initial Activities 117
B-4.8 Discussion – what do you do? Which offers Best Value? Which would you choose? 118
PART 2 FROM THE VENDOR PERSPECTIVE 118
B-4.9 Discussion – Listen and Learn – What and How? 118
B-4.10 Discussion – The Paradigms (20 cents) – KPPs, Processes, the PM and SE Paradigms, the Business Venn Chart – sketch these out 119
B-5 Case Study – Hubble Space Telescope (HST) (Wikipedia, 2020) 119
B-6 Case Study – The FCC's Decision Puts GPS at Risk By Mark Esper 123
B-7 Case Study – The COVID Recovery Comes Down to Engineering by Guru Madhavan 125
B-8 Case Study – Five Engineering Disasters That Could've Been Prevented with Systems Thinking, From Worcester Polytechnic Institute 127
B-8.1 The Microsoft Zune – Discussion 127
B-8.2 The Water of Ayole' – Discussion 127
B-8.3 Fenchurch Street – Discussion 128
B-8.4 The Russian K-141 Kursk Submarine Disaster – Discussion 128
B-8.5 "Galloping Gertie" A.K.A., The Tacoma Narrows Bridge – Discussion 128
B-9 Case Studies – Rapid Fire 129
B-9.1 Data Ingest 129
B-9.2 Medical Systems 129
B-9.3 Voting System in the Cloud 130
Appendix C Solutions and Answers to Case Studies 131
C-1 Case Study - Driving in Italy - Solutions and Answers 131
C-2 Case Study – Volts and Bolts – Solutions and Answers 132
C-3 Case Study – Corp of Discovery – Transport – Solutions and Answers 132
C-4 Case Study – Green Mountain – Solutions and Answers 135
C-4.1 PART 1 – From the GVHS Perspective 135

C-4.1 Discussion – Project Start Up 135
C-4.2 Discussion – Initial Planning 135
C-4.3 Discussion – CONOPS 136
C-4.4 Discussion – The Problems 136
C-4.5 Discussion – Your Team 136
C-4.6 Discussion – Make or Buy 136
C-4.7 Discussion – Initial Activities 137
C-4.8 Discussion – what do you do? Which offers Best Value? Which would you choose? 138
PART 2 FROM THE VENDOR PERSPECTIVE 138
C-4.9 Discussion – Listen and Learn 138
C-4.10 Discussion – The Paradigms (20 cents) 139
C-5 Case Study – Hubble Space Telescope – Solutions and Answers 140
C-6 Case Study – The FCC's Decision Puts GPS – Discussion 141
C-7 Case Study – The COVID Recovery Comes Down to – Discussion 142
C-8 Case Study -Five Engineering Disasters That – Discussion 143
C-8.1 The Microsoft Zune 143
C-8.2 The Water of Ayolé 143
C-8.3 Fenchurch Street, London 143
C-8.4 The Russian K-141 Kursk Submarine Disaster 143
C-8.5 "Galloping Gertie," A.K.A., the Tacoma Narrows Bridge 144
C-9 Case Studies – Rapid Fire - Discussion 144
C-9.1 Data Ingest 144
C-9.2 Medical Systems 144
C-9.3 Voting System in the Cloud 145
Appendix D Glossary of Terms 147
Bibliography 153

0.0. Milestone Zero – Before the Beginning

Accomplishments sprout from hard and focused organizational teamwork whether it is a sports team, an exploration of unknown territory, the playing of a great symphony or in the case of Systems Engineering the successful delivery of a system such as an aircraft, a ship, iPhone, construction project or a data center. At the same time, the sciences and innovations have become interdependent for discoveries that are used in Systems. These discoveries have worked their way into our daily lives where cellular communication depends on the data centers, where aerospace depends on high performance computing, healthcare depends on high tech scanning equipment, and attending a concert depends on an acoustic design allowing for the best audience experience. This is the story of how these sciences are engineered to become interdependent systems.

Before starting into Systems Engineering, it is important to focus on the genesis of systems thinking, and the pre-requisites of effective projects and systems. Milestone Zero will offer a quick look back on the pre-20th Century eras that brought us to the use of science, the advent of technology and the application of engineering leading to the 21st Century. Next, I will discuss how to organize for success, the importance and the differences in leadership, management and administration. So we all use the same language, the fonts and basic terms are defined as used in this book, and I'll summarize to what level of understanding this book is written, and to what level of understanding you should have to be successful. But first, before the beginning. I will present three concepts to illustrate engineering over the years 1) The Fifth Era, 2) The Fourth Industrial Revolution 3) Era of Enlightenment.

0.1 We First Learned to Count

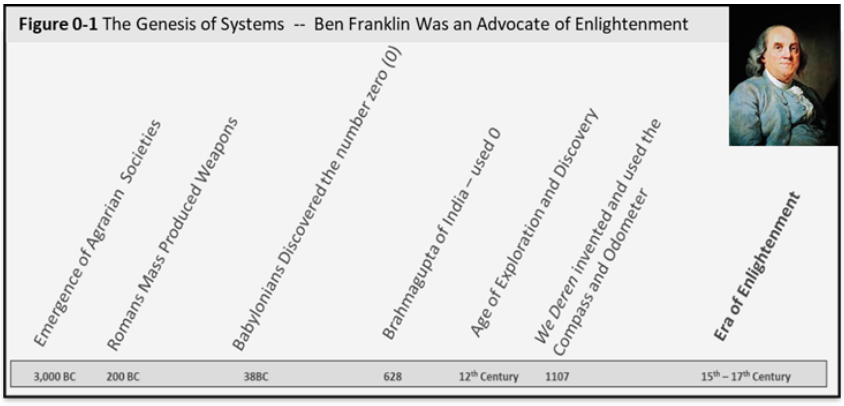
Human progress may be divided into five Era's as described by a book titled Corporate Innovation in the Fifth Era, written by Matthew C. Le Merle and Alison Davis (Le Merle & Davis, 2017). Those eras are;

- Era 1 Hunter Gatherer starting with the origins of humans to about 11,000 BC
- Era 2 Agrarian from 11,000BC to 1400 AD
- Era 3 Mercantile from 1400 ad to 1800 AD
- Era 4 Industrial from 1800 AD to 2000 AD
- Era 5 an unknown, yet named era that started in 2000 AD

Era 5 is marked by a The Digital Revolution and easy access to data, analytics, and computing power. These various capabilities – for quick communications, data and analysis, discovery through modeling and simulation – will converge and affect industries otherwise thought immune to innovation such as transportation, agriculture, and biotechnology. The transition from one era to the successive era was not marked by a specific year or day, but instead these transitions often took years, and the changes were not recognized as a new era. In other words, those who experienced the change would know the change in hindsight. I would say that while most of the world would have moved to a new era, some portions of the world remained in the previous era. We still have Hunter Gatherers, and we still have Agrarian societies in the world. Additionally, the succession of one era was built upon the capability of the previous era. Hunter Gatherers captured and sold their game, Agrarians sold their crops as if it were in the Mercantile. While our focus will be on Era 4, and the emerging Era 5, a quick summary of the earlier eras will help provide context to the Pragmatic Systems Engineer.

Many people believe we are entering The Fourth Industrial Revolution which is marked by the convergence of artificial intelligence, autonomy, internet of things and people's physical lives. The term was used at Davos 2020 for the World Economic Forum (Schulze, 2019). There is a general consensus that our society is entering a yet to be defined era of integrated science and technology, and we've not yet determined the name of this era.

The nomadic tribesmen of thousands of years ago became agrarian upon discovering the science of forecasting, planting, and harvesting. People learned to control or mitigate events by storing and managing resources for villages that could withstand the forces of nature and man. Four thousand to five-thousand years ago the Sumerians invented numbers (Wikipedia, 2020). The Egyptians, Persians and Romans improved the use of mathematics to measure and calculate. The Babylonians reportedly discovered the number zero (0) in the 3 B.C., but didn't know what to do with it until Brahmagupta of India in 628 determined how to use it in mathematics. In 200 BC, the Romans mass produced spears and cross bows for hunting and their armies (Szalay, 2018) (Knighton, 2017). Written languages matured so people were able to convey and record ideas to one another. The 12th Century witnessed great increases in exploration, governance, and conquering nations, and the first compass and odometer was invented in 1107 by Chinese engineer We Deren (Carrick, 2018). The 16th and 17th Century saw the 'Era of Enlightenment' (not considered one of the five Eras) during which societies started to reason, obtain knowledge, use their own sense of community, religion, natural events, reductionist theory and science (History.com Editors, 2020). The Era of Enlightenment[1] occurred during the early days of the Industrial Revolution, which saw the first images of modern engineering and the application of science. Science included the idea of empirical evidence from studies and experiments leading to theories, rather than anecdotal evidence of day-to-day experiences. Reductionist theory states that an entity can be reduced to its smaller parts in the form of a hierarchy, and that the larger entity draws on the resources of these lower parts to provide an emergent behavior (History.com Editors, 2020). While this applied to natural and biological systems, it was the basis for modern system engineering, although the holistic approach offers a healthy intellectual approach to thinking of a system. Reductionist, holistic or a combination of, in any case -- no longer were people subject to the whims of nature.



[1] **Era of Enlightenment** is used in this book as a secular reference. While many theologians or clergy of various beliefs do not favor the Era of Enlightenment, I use it as a major milestone in human thought and culture for science and engineering, as indicated in the text. It is not intended to persuade the reader of a non-secular belief.

0.2. The Organization

An organizational structure should be defined that will suit the goals and objectives of the organization. The organizational structure is built to achieve the mission, goals, and objectives of the business venture. The purpose of the organization and how its constructed are paramount, even more than the needs of those in the organization or the organization will not achieve its maximum performance. The organizational structure, mission, strategy, goals and vision, and policy should be documented and made available to stakeholders, so they know roles and responsibilities of the leadership, and of the team members. From this, the organizational Best Practices, Systems Engineering Management Plans (SEMP), Project Management Plans (PMP) and managing documentation are created. All these documents communicate the structures, goals and plans, standards, timelines, and resource expectations to all stakeholders. Without at least a minimal amount of communication regarding these topics, confusion will prevail.

0.3. Leaders, Managers and Administrators

Leaders, Managers and Administrators are three separate but complimentary disciplines of which none can be dismissed or confused with the other.

- Leaders – inspire others to complete a set of tasks needed to achieve the mission. A leader must have a vision and a goal, and the followers motivated (intrinsically or extrinsically) to achieve the goal. I've described four general leadership styles as synthesized from Paul Hersey and Ken Blanchard's Situational Leadership Model, as outlined below (Hersey & Blanchard, 1969).
    - **Telling (Directive)** – If the team is not self-motivated nor well trained, this is a suitable leadership style. Directive is also needed in emergency situations such as 'Call 911!'.
    - **Coaching** – the leader works shoulder to shoulder to show and tell the followers what must be done and how to do it. The followers are motivated to succeed but not well trained.
    - **Participating** – the followers are trained and motivated and the leader works with the team as a 'hands on leader'.
    - **Liaise Fair** (leader is laid back) – the followers are trained and motivated, and only need the leader to provide the resources and the vision to succeed. Often the leader is not qualified to perform the tasks performed by the team.

A common understanding of the vision and the goal becomes more important for Participative and Liaises Fair leadership style. The concept of Servant Leader becomes applicable in a more, mature team. In technical environments, it is better to have a team that responds to Participative and Liaise Fair, thereby bringing out team innovation, open communication, and the confidence for team members to raise problems early.

If you, as a leader apply the wrong type of leadership then resentment will usually prevail among the followers, so you must know the people you are leading to determine the most suitable type of leadership to apply. A common error is for a newly arrived leader to declare the type of leadership style they apply, and then expect the team to change their work style to meet the leader. Additionally, people tend to become the style they are most comfortable instead of what is needed for the team. Most leaders who were once workers try to continue as a worker (Participating) although this approach is not what the team needs. The good news is followers usually give the leader multiple chances to improve before the team goes awry, flounders or they get another job.

- **Managers** – If your job is to efficiently apply resources to achieve an objective, you are a manager. The objective should be clear and understood, you must have a plan with periodic checks and indicators of your progress. Using the right type of metrics, you must stay on course and adjust the plan as needed. Most of this book will discuss the attributes of managing as an engineer.
- **Administrators** – not to be overlooked, someone must maintain all the documents and records. Someone must ensure timelines are met, resources are controlled, and reports are done on time. This is the job of the Administrator. Good managers need good administrators. In project management, those who perform these duties are often called **Project Coordinators.**

I will use the term 'competent authority' throughout this book and training. Well, what is competent authority? It's a person (sometimes a group) able to decide due to their status in a group, legal authority, their knowledge and expertise, and the information presented to them. A larger organization would have a table of authorities that describe what level of approval is needed to make a decision based on the magnitude of the decision, how it aligns to the strategy, risk and the cost of the decision. Those in competent authority positions are appointed (or should be) by organization senior leadership.

It is best that a leader has at least some management or administrative skill, or that a manager has some leadership skill. It is a common mistake to assign a successful leader to a management position thinking it is the same skill. Again, these are common mistakes, that should these mistakes occur, are easily detected and can usually be corrected.

Trust

It's the foundation of leadership. A term we often use, but to which there is not enough. It's your integrity, your word, the truth. Its confidence your team and stakeholders have in you. Its personal, and it's organizations its capital. It's doing what you say, say what you are doing, and not disappointing your stakeholders. You can do plenty right but without trust, you'll lose customers. But as you build a trusting reputation, business will come your way.

0.4 Effective Communication

Effective communication is the enabler of success. Teams can be highly qualified or motivated but if one part of the team doesn't know what the other part is doing, duplicated work or gaps in work will occur. While everyone on the team is responsible for communication, it is the leader who sets the culture to be open, nonthreatening and innovative. Artifacts[2] of Systems Engineering are used for communication – the requirements express what will be done, the architecture expresses how, organization charter states who will do the work, action items and risks are tasks that must be completed, the test plan is how it will be checked and so on. This allows stakeholders the visibility into the project and system status, plans, cost, and performance. Nonexistent documents, out of date or conflicting documents are poor communication.

[2] Artifact is not only a dinosaur bone or ancient scroll. For Systems Engineer is any accountable item to include a product, plan diagram that needs at least some level of control. This term is used throughout this book and is better defined in the terms in Glossary of Terms.

One more thing about effective communication – the best ideas often come from the quietist person on the team. Seek out that person and coach them to communicate and express their ideas. And remember that up to 80 percent of communication occurs non-verbally – through vocal inflection, facial expression and body movement.

0.5. Its Personal

I will write in first or second person to personalize our communication and will avoid passive third party communication where possible. This is our story.

0.6. Style Guide

We must establish and document a common language in any organization, project or system, business plan or proposal, or people will talk past one another and the effort will be wrought with confusion and misdirection. Recipients of the system may not be able to interpret colors or sound as most others due to even minor impairments of their senses. This common language will be discussed in more detail but for this training and for this book, I've included a style guide to define colors, font, and icons.

0.6.1. Colors

- **Red** – print or graphics in red are generally bad and require immediate attention or are of the most importance. This means a project is far over budget, failure is imminent, and the system is out of tolerance and beyond immediate repair.
- Yellow – the effort is in risk of failure but can be recovered if immediate action is taken to correct it.
- Green – on target to success be it a budget (usually actuals within 5 percent of budget), all problems of the system or project can easily be corrected or tolerated.
- **Blue** – the project or system is performing virtually 'perfectly', but don't get complacent. Whatever you are doing on this project or system can be gleaned out and should be considered for use on other projects if it is applicable. A word of caution – some stakeholders will find 'green' acceptable so any additional effort to make your project or system a perfect blue may not be necessary, and in fact, wasteful.

0.6.2. Fonts

- Call Out Boxes – The sterile presentations of academics often miss the salient points of reality. I try to capture these ideas in 'Call Out Boxes'. 'Confusing Metaphors' below is a Call Out Box.
- *Italics* – for emphasis, titles, names of ships, aircraft, etc
- **Bold** – for emphasis – an important point that you want to stand out.
- Underline – a defined word or expression
- 'Tic Marks' – a statement (direct or paraphrased) by a person, organization in which the literal statement is important to context.
- Web based hyperlinks or URLs are embedded in the text of The Book that leads to cross referenced topics. For example, if you are reading about Functional Requirements and the Section of Verification is referenced, a hyperlink is built in Verification to quickly take you from the Functional Requirements Section to the Verification Section.

Grammar and style formats follow 'The Hodges Harbrace Handbook' (Glenn & Gray, 2010)

0.6.3. Terms

- Project Management vs. Systems Engineering – In this book these are often the same or at least very similar, so unless otherwise indicated the two terms are interchangeable. There is a difference between the two and this will be discussed in Section 1.5. Systems Engineering is a broad set of skills to include Requirements Managers, Architectures, Testers, Human Systems Engineers – all who are considered specialized Systems Engineers. In this Book, the term Systems Engineers will include the specialties unless otherwise indicated.
- Process and Methodology and Model – these three terms have related but different meaning. Process is an action that is performed, or WHAT is being done. Process will not describe HOW it will be done, which is a Methodology (or method for short). How a process is performed is not a Model. Often, terms such as Development Model or Architecting Model mistakenly replace Development Methodology or Architecting Methodology. This Book will not go into this level of detail, but this distinction must be noted. Often the three have come to overlap. Additional information is available in a paper written by Jeff A. Estefan entitled 'Survey of MBSE Methodologies', May 23rd, 2008 (Estefan, 2008).
- Should – the action described is optional but recommended. Not performing the action increases the risk of failing.
- Must – the referenced action is required by policy or law, or the desired outcome will likely fail.
- Can – able to, optional course of action, consideration to improve efficiency
- Will – a desire and intent, that is non-binding (may change your mind) for forthcoming events.
- Provider – generic form of Contractor, Sub-Contractor, Vendor – because this Book is written for a wide audience, various terms may exist for the same organization. Unless stated otherwise, Provider will be used to refer to all these parties.
- Offeror – primarily a proposal word, it is the organization (usually an industry or business) that is providing a proposal in response to a Request for Proposal (RFP).
- The Systems Engineering Book and Training – this Systems Engineering Book and the Systems Engineering Training go hand in hand. Unless otherwise indicated the two terms are synonymous in the training and in The Book. 'The Book' or 'This Book' will be used for brevity.
- Use of metaphors – An effective use of metaphors can allow an abstract idea to become very pragmatic. But misuse of metaphors (or mixed metaphor) can lead to miscommunication and confusion, even resentment. While the metaphor might sound good at the time, only later will confusion be realized. Metaphors that sound especially effective are perpetuated and misused often by those who don't understand the real meaning of what they are saying. I have been careful in the use of metaphors.
- Stakeholders – anyone who contributes to an effort or is affected by the result of the effort is a stakeholder. This may include internal members of the team and external recipients of the system to include those who pay the bill (often the U.S. Taxpayer or the Consumer).
- Team – this term is used to indicate contributing members of those working directly on the project or system.
- The Organization – Everyone is part of an organization that has purpose, rules, provides resources, established agreements. The Organization will have sub-organizations and parent organization. For

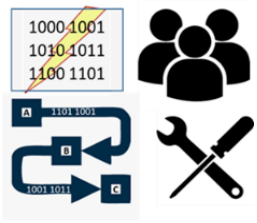
Confusing Metaphors

- Boots on the Ground
- Shovel Ready
- Other Shoe Drops
- Writing on the Wall
- Got Our Arms Around It!
- Brass Ring
- Needle in the Haystack
- Connecting the Dots

the purpose of this Book, The Organization will be the entity delivering value, providing resources, establishing the rules and guidelines, and to which your team feels an allegiance and companionship.

0.6.4. ICONS

As we progress into a particular area of Systems Engineering, I use consistent and meaningful icons. But one that you'll see early is the Project Triad of Cost, Schedule and Performance – all of which may present risk. The Systems icon is represented by People, Process, Tools with input. The lightning bolt illustrates 'Energy' or material, which are needed for the function of the process. The square represents 'way points', or Gate Reviews (thumbs up or down) in which managers decide to progress to the next set of activity or not. More on this later.



It's very unfortunate to embark on a worthy project to create a needed system and to miss the basics of a vision or open communication, even though the right people, processes and tools are in place. It's a waste of human capital and innovation. But these reasons are often why projects fail – because they are flawed before they start. This flaw underscores the importance of systems engineers having a broad background of not only the sciences or engineering, but also of the more liberal arts disciplines needed to thrive in today's world.

0.7. Learning Model

Not to insult you, but I have assumed that some of you have little or no knowledge of system engineering. In fact, you may have a misunderstanding of systems and must 'unlearn' bad information, and are essentially behind those who may have a clean slate. A careful reading of Section 1 The Systems Engineering Fundamentals will help provide clarification. For this Book, Blooms Taxonomy is used to provide the right level of instruction – from basic recollection and memory level to an application level (Vanderbilt University, 2020). See Section 5.4.1 for a description of the Learning Model.

Now that the groundwork has been laid, I will spend the remainder of this Book on System Engineering and Project Management. The foundation of these efforts has been covered in this section, for which without these – the organization, the effectiveness of the people, the leaders, managers and administration, the project and the system are already in jeopardy. That's why I discussed it first. At the risk of sounding redundant, I will make at least three passes through Systems Engineering. Section 1 is the first pass at'30,000 feet' in which I describe the Fundamentals. Section 2 is more detailed but now at mid-level – '5,000 feet'. Section 3 is at 'ground level'.

'Responsibility is a unique concept... You may share it with others, but your portion is not diminished. You may delegate it, but it is still with you... If responsibility is rightfully yours, no evasion, or ignorance or passing the blame can shift the burden to someone else. Unless you can

point your finger at the man who is responsible when something goes wrong, then you have never had anyone really responsible.'

Hyman G. Rickover, Admiral, U.S. Navy
Father of the Nuclear Navy

1.0 Systems Engineering Fundamentals

Systems were built before it was called Systems Engineering. We will start at the origins of System Engineering and then define a system, why perform Systems Engineering, and who performs it. This is the high level – 30,000-foot level. You will see concepts of What, How, Modeling, Behavior, Decomposition and Tracing throughout the entire book.

1.1. The Origins of System Engineering

The origins of System Engineering were born out of the need to build large, complex systems that required various skills and disciplines. As discussed in Section 0, systems existed before the 20th Century, and the need for a 'systems approach' for aviation, shipping, transportation, telecommunication, and space exploration required an integrated, product-based approach. Many people practiced system engineering without calling it system engineering, not knowing the early systems thinkers were blazing a new profession. An early systems approach was, not surprisingly, within the Department of Defense when they created early models using Military Lifecycles, and Specifications (MILSPEC) (Department of War for large ships and aircraft in WWII) for Systems Engineering. Later, in the 1980s Systems Engineering organizations such as the International Council on Systems Engineering (INCOSE) and Industry Best Practices established common approaches for designing and building systems. The early tenants of system engineering emerged and are still valid and evolving to this day.

1.2. What is a System

A system is a set of interrelated parts, each performing a unique function that delivers something of value. Systems have become ubiquitous. They include mass transit, medical records, transportation, utilities, telecommunication, weapons systems, and one could even consider organizations or government as systems. This Book will describe the former – of systems that are designed by people, created of material and energy, that deliver a function to serve people. Because of system complexity – required resources and various functions – successful systems are not spontaneously delivered, so considerable planning must be accomplished to help assure success.

These parts working together (as illustrated in **Figure 1-1**) provide emergent behavior which is the resulting function of the system. Systems are human made from material, energy, data, hardware, software and includes people, process and the tools to design, build and test. Systems are bounded and scoped because there must be limits on the defined problem – systems by themselves cannot solve 'world hunger' or large epic, and often nebulous problems. Bounding a problem and a system prevents the system's behavior from creeping into other systems and their behavior. This would be chaos.

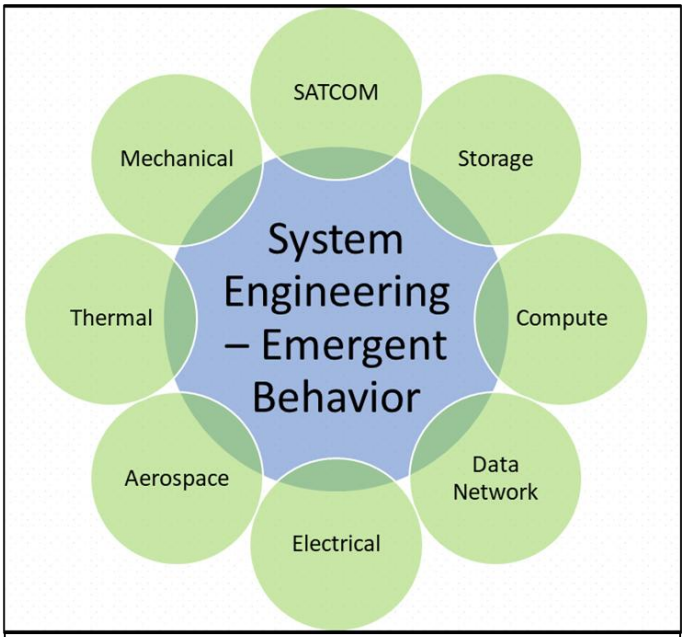


Figure 1-1 Systems Connect Various Engineering Domains to Realize 'Emergent Behavior'

These systems have become seamlessly interconnected to provide unforeseen value of only a few decades ago yet have become more confounding and more complex. If not managed well, these systems will pose epic problems.

The terms System of Interest, System of Systems and Enterprise System are often used in the industry, and often not consistantly. I will define these as follows:

- System of Interest (SOI) – a single entity, and single sponsored system fulfilling a specific purpose, with a single lifecycle, managed by the organizationally sponsored staff.
- System of Systems (SoS) – composed of two or more SOI. While each system is managed and controlled as a SOI, the collection of systems draws resources from one another. One SOI may be considered a 'user' of another system, or a customer of another. The SoS delivers capability that is not delivered by any one SOI.
- Enterprise System – often considered 'classical systems engineering' enterprise suggest that the system delivers a broad set of strategic capabilities, managed by the larger organization, and may be composed of two or more SOI that are under one managed organization. Regardless of how tightly bounded the systems of the enterprise, the Enterprise Systems exists to serve the needs of the one organization.

These terms are still evolving in their meaning so it is advisable for you to learn how your organization will use these terms.

1.3. What is Systems Engineering

Engineering is the use of scientific principles to design and build machines and structures. (Engineering, 2020) Systems Engineering is integrating these creations into a composite set of equipment. A common definition for Systems Engineering is 'an interdisciplinary approach and means to enable the realization of successful systems' (International Council on Systems Engineering (INCOSE), 2020). Put another way, Systems Engineering is solving a complex problem composed of various entities and disciplines by decomposing into small pieces, planning different solutions, modeling the behavior, then solving the problem by constructing the system, while ensuring it will function and perform as planned. This is a long definition and warrants further explanation.

- Complex Problem – Systems Problems are presented differently depending on the circumstances, the time, location, or environment. Most often, they are a new User Need, Mission Requirement or User Requirement in which the user needs a system that will be applied for a new business area, or a new threat. 'Problems' may also be presented as opportunities to apply new technologies to old problems.
- Breaking It Down or Decomposing – It is best to break a problem apart into manageable portions then allow each portion to be solved by a person who is skilled in that problem area. **Figure 1-2** (next page) illustrates several systems that are broken down, and decomposed into sub-systems, and elements. Systems are not only Information Technology Systems but also include a wide range of disciplines, including Mechanical and Civil Engineering. Decomposition lends itself more to a Reductionist view but also applies to a Holistic approach to Systems Engineering.

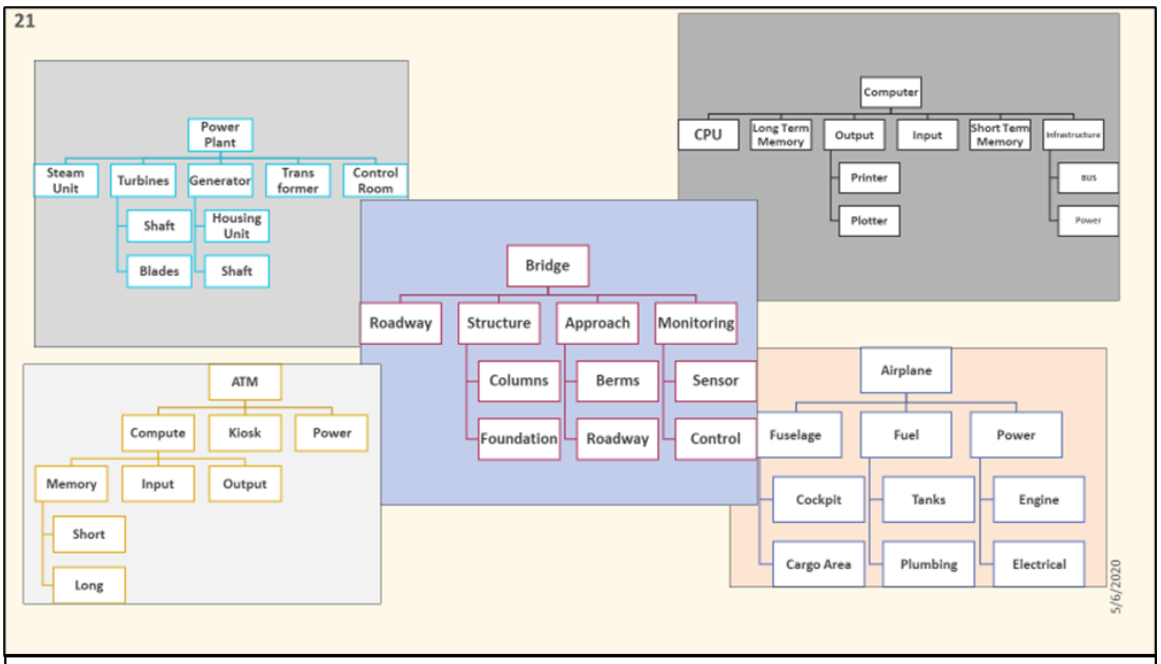


Figure 1-2 Examples of Systems and How Broken Down into a Hierarchy

- Various Entities and Disciplines – Each engineering and technical endeavor are performed with little understanding of other professions, craft or disciplines. The Mechanical Engineer first learns their domain of mechanics, or an Aerospace Engineer first learns about air and space technology. Only later in their careers do they learn, and then apply their technical discipline to an enterprise composed of assorted disciplines or skill entities. The domain to which a person is trained and is performing is considered their domain.

- Modeling Future Behavior of a System – A core concept of Systems Engineering is Modeling and Simulation. This simply means that System Engineers will predict the outcome of events based on how they model the structure in its planned environment. A model may be a wind-tunnel for an airplane (**Figure 1-3**), a closed laboratory for telecommunication system, or even mathematical formulas (**Figure 1-4**). An effective model will forecast behavior and events not thought of by the human mind, such as a Black Swan event (Section 5.5.4), or something that is counter intuitive. Modeling and Simulation are most thought of for systems function and performance, but a model will also be a budget or schedule of events because it predicts the future behavior of the system. Schedules model the future timing and activities of the system, and a budget models the consumption of resources for the system. Mathematics offers languages and formulas as models, and once data is inserted, and the variables take on a value, they also become simulation. Systems Engineers document the expected performance of the model and build computerized models and prototypes to predict and hone this behavior, into a realized system. Models are expected in Systems Engineering or Systems Engineers would haphazardly go about

Figure 1-3 Winds Tunnels are an Examples of Model and Simulation

A=1+2 $\quad ax^2 + bx + c = 0 \quad P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \lim_{x \to p} f(x) = L,$

Figure 1-4 Mathematical Formulas are Models

building in hopes of a successful outcome. These models (laboratories, the designs, architectures, form/fit, prototypes, proof of concept, and the budgets and schedules) are reviewed at key waypoints to allow for a measured approach to progressing in the project.

- Constructing the System – Once the System Engineer defines the discrete parts of the problem, the System Engineer with other engineers will solve the problem by designing, modeling, and simulating several possible solutions. The best solution based on function, cost and schedule is selected, and portions of the system are assembled and integrated while maintaining the waypoints – the Milestones and the Gate Reviews leading to the emergent behavior.
- Emergent Behaviors – System Engineers refer to the function of a system or its parts as 'behavior'. This is the result of the system function, or the output of the system. Emergent behavior is the overall behavior the system, which is greater than any of its parts. If the System Engineer did a satisfactory job on the design, then development and implementation, the behavior of the system should be as expected. At times, the system offers emergent behavior that is not expected in the form of a defect, deficiency or an anomaly. If the Systems Engineer maintained accurate records of the system requirements, the architecture, and design then tracing the source of the defect will be much easier than it would be if these artifacts were not accurate or were unknown.

'All Models are Wrong, but Some Are Useful' – George Box, 20th Century Philosopher (Smart Guy) said this. What did he mean? I believe that we should keep a skeptic eye on a model – a syntax that represents an abstract reality. We tend to oversimplify or lose context in the model, or the model is mis-used. Make a good, useful model but always ask questions – 'what and why, and so what?' (Georgiev, The Start Up, 2019)

1.4. Why Perform System Engineering?

A sound approach of performing any work is to determine what must be done then how the work will be performed. Given a situation to resolve, people tend to think (or should think) of 'defining the problem and the desired result' then how to make the change to the desired result. For complex problems a very methodical approach must be applied as previously described. If the system is complex, the cost is often high, and the system can take extended time and resources of integrated teams before the solution is realized.

Beside the anecdotal evidence, several studies have borne this out. For example, the National Defense Industrial Association, the Institute of Electrical and Electronics Engineers (IEEE), and the Software Engineering Institute of Carnegie Mellon joined to conduct a survey on 148 projects and found that when 'Low' system engineering is applied only 15 percent of these projects were considered successful, 33 percent partially successful and 52 percent failed, as illustrated in **Figure 1-5** (International Council on Systems Engineering (INCOSE), 2011). The situation improved when a 'Medium' amount of Systems Engineering was applied, to a 24 percent of projects considered successful, and the remaining having failed or only partially successful. Those projects applying a 'High' amount of system engineering

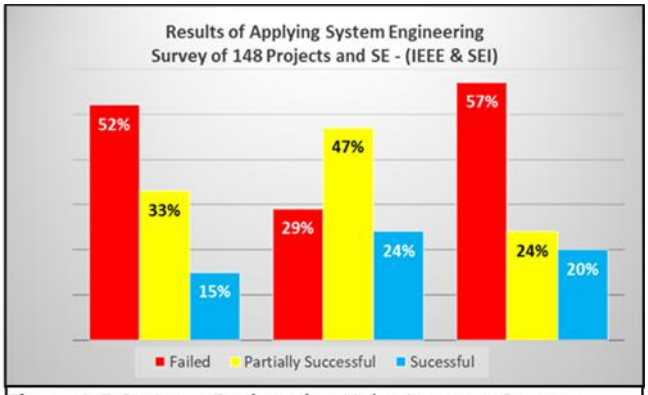


Figure 1-5 Systems Engineering Helps Improve Success

had a success rate of 57 percent, and 24 percent were partially successful when a medium amount of Systems Engineering was applied, and 20 percent were successful. Although the study did not specify the amount for Low, Medium or High, nor did it specify the reasons for the failure (e.g. cost, schedule or performance), the point is by applying enough system engineering the buyer receives a great return on investment.

A study cited by INCOSE and a paper titled, 'The Cost of Doing or Not Doing Systems Engineering' by Eric Honour (Honour, 2013) offers consistent evidence that by applying system engineering, overruns are reduced significantly, as illustrated in **Figure 1-6**. It indicated that if nothing is spent on system engineering (a haphazard approach), there is a 53 percent overrun. If five percent of the project costs are spent on system engineering, then overruns decreased to 24 percent. If 15 percent of the project costs are spent on system engineering, then a remarkable 97 percent of these projects are on or under budget. This survey does not indicate if the projects were ultimately successful or met the requirements, but only if they were within budget.

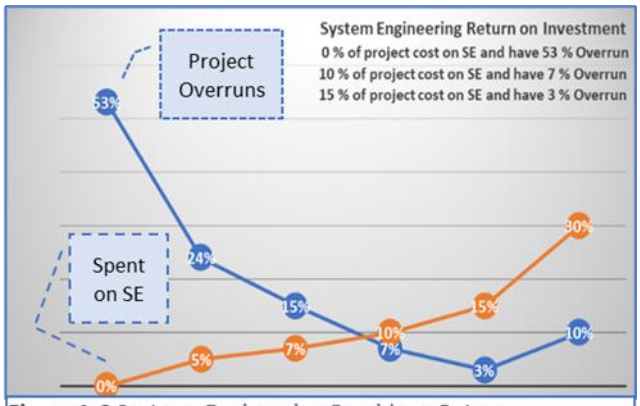


Figure 1-6 Systems Engineering Provides a Return on Investment

Finally, I offer you one more piece of evidence of the effectiveness of system engineering. **Figure 1-7** indicates the cost of delivering and using a system from the start to the finish (Defense Acquisition University, 2020). As you can see, the bulk of cost (on average) occur during Operations and Maintenance (O&M). Systems Engineers must make careful and thoughtful decisions to ensure they understand the problem and they have selected the most effective solution. System Engineers must ensure all parts work together before committing to operations or the system may not perform as expected, adding an inordinate amount of cost and perhaps even impearl those who depend on the system. The thought of a failed mission critical system in its operational environment is sobering to a system engineer.

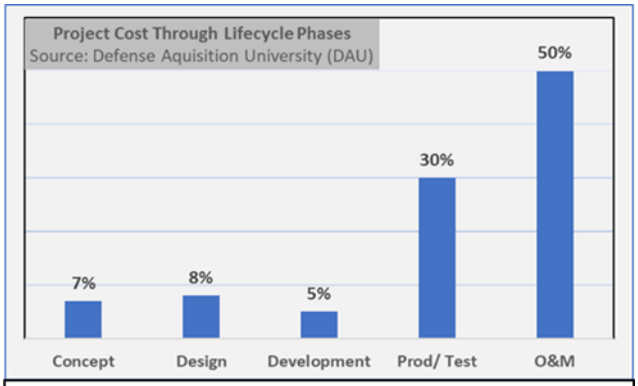


Figure 1-7 Pay for Systems Engineering Early to Help Ensure Success Later

There is no successful business case in haphazardly building a system.

1.5. Who Are System Engineers?

People tend to be Systems Thinkers or have a Systems Perspective. Very few (and no one I know) are fully either one but tend to a blend while favoring one. I bring your attention to **Table 1-1** on the Systems Thinkers and those with a Systems Perspective, and how the two blend. A Systems Thinker is one who thinks 'Bottoms Up' that systems are connected through an ontology that is discovered. Systems Thinkers tend to be holistic thinkers; they look at the entire system, its cause and effect among its parts, and learn of its cascading effects. The person who has a Systems Perspective will look at a system 'Top Down, and those systems have a hierarchy, that it is continually broken down and reduced to its basic parts. With a Systems Perspective, the Systems Engineer will plan, model and simulate the emergent behavior. Both approaches are acceptable, but it is important that Systems Engineers (and others) understand how others might think, and how their own understanding will move from the Perspective to the Thinker, and back again. Finally, there is the Systems Professional who must understand both the Thinker and Perspective and draw on their own experiences and from others. The Systems Professional must think of new ways of developing systems from what may not necessarily be conventional. The Systems Professional will have University Degrees in Systems Engineering, Certifications in Systems Engineering, and Systems Professionals are the custodians of this relatively new profession of System Engineers. Seventy Percent (70%) of the development of Systems Engineers is performing as a Systems Engineer from experience or On-Job-Training. Twenty Percent (20%) of their development is from mentoring, and only ten percent (10%) is from training. So, most Systems Engineers learn by doing.

Table 1-1 Systems Thinkers, Perspective and the Blend

Synthesis ← Some of Each→ Analysis	
Systems Thinkers Holistic, **bottom-up** approach of **synthesizing** the behavior of the small parts, how it contributes to the larger. The ability to understand the cause and effect, cascading events, interconnections of actions and an ontology.	Systems Perspective Reductionist and hierarchy approach, **top-down analysis**, inputs – process – output and the results; abstractions and models, learn and understand the system that includes parts, what is required for each part, anticipating the emergent behavior.
Systems Professionals Built on Thinkers and Perspective – Certifications and University Degrees – the custodian of the Systems Engineering Profession. Systems Professional can determine the best way to deliver a solution, to include the processes, documentation, risks – even if it's a hybrid or new method.	

Typical work of Systems Engineers includes defining the user and operator needs by interviewing, working with and observing, creating architectures and solutions, and leading the team in selecting the most suitable alternative. Systems Engineers make quantitative and qualitative decisions, leading and managing a test process, perform modeling and simulation of possible systems. Systems Engineers monitor and manage systems in operations and ensure smooth transitions to new systems or for retiring systems. Systems Engineers perform risk management, manage schedules and budgets in conjunction with the Project Manager or Project Coordinator. Systems Engineers work with, and lead others such as the Project Manager, Program Manager, domain engineers, testers, requirements managers, senior managers, customers, users and maintainers, and business professionals.

System Engineering is often confused with other professions either because of the similar sounding name or the lack of understanding of what a Systems Engineer does. For example, a System Administrator **is not** the same as a System Engineer. A System Administrator operates an Information Technology (IT) system by adding and deleting users or parts of the IT system, they monitor the system and take corrective action, or they monitor trends and take preventive action. A Technical Administrator or Trouble Shooter **is not** the same as a System Engineer. A smart technical person (albeit impressive at times) **is not** a Systems Engineer who prepares technical and program briefings and reports. Software Programmers or Cloud Engineers **are not** the same as Systems Engineer. These people may be, or become Systems Engineers but those professions, in themselves do not make them Systems Engineers.

1.6. A Stakeholders Point of View

Stakeholders are not only the System Engineer but also the User and Operator, the Project Manager and the Contracts Person (or the Contract Officer Technical Representative (COTR)). It also includes the user, the maintenance person, the software coder and many others but for the point of this Book, we will compare the Perspective of the Systems Engineer, the Project Manager, Contracts Person and the User/Operator. These perspectives are illustrated in **Table 1-2**. The larger perspectives of all stakeholders (often called Viewpoints and Points of View) are discussed in Section 3.1.11. The Language of System Engineers and Section 4.1 will be discussed later. The term user and operator will apply interchangeable throughout this book unless otherwise noted.

Table 1-2 Same Objective, Differing Perspectives

	Systems Engineers	**Program or Project Managers**	**Contracts / COTR**	**User and Operator**
Are my people in the right positions on the same playbook?	Are my engineers working the right tasks?	Are people charging to the right charge number?	Are Contact Line Items (CLINS) overrunning the budget?	Am I paying too much for this system?
Is our timing synchronized?	Will we create our design on time?	Where is the critical path in the schedule?	Will vendors comply with the schedule?	Will it be ready and available when I need it?
Will we cross the goal line?	Have I traced my requirements?	Will the requirements be met?	Will they satisfy the SOW Requirements?	Will it perform and operate as expected?
Do I need a more robust team?	I need someone who can write embedded 'C' !	Do I need to hire another experienced developer?	Is there a position for another full time equivalent (FTE)?	Will I need someone to perform maintenance on it?
What will take me off the game plan?	What may go wrong in My System?	Where is Murphy on My Project?	Have they delivered the Risk Management Plan for My Contract?	I don't want to think about my system too much, I have my job to do.

1.7. And Your Job is ……

We should never forget that although the system will serve people, our work is part of a business. As illustrated by **Table 1-3**, each person at each level has their job and the interactions of the organization must be synchronized and focused.

Table 1-3 What is My Job?			
Org. Level	**Purpose**	**Directive/ Guidance**	**Major Participants**
Corporate, Division, Business Area	Business, Mission – Profit/Loss, Make $$	Policy and Guidance, Strategy, Federal Acquisition Authority (FAR)	Owners, Board, CEO, CTO, Sr Mgt Team
Program	– Delivering Product, Service, Value, & Sustaining It, Profit & Loss (P&L), multiple Projects, O&M	FAR, Program/ Engineering Policy, Cost/Schedule/ Performance and Risk Metrics, Gate Reviews, Enterprise Process, Best Practices, PM Plan, **SE Manager Plan (SEMP)**	PM, Lead Engineer, **Systems Engineering Integration and Test (SEIT**), SMEs, Operations, Contracts, Finance,
Project	Temporary Endeavor -Distinct Start / End, Unique Purpose, Deliver Value	Program/Engineering Plans, Cost/Schedule/ Performance/Risk, Metrics, Architectures, Best Practices, Processes, Gate Reviews, Checklists	SE, Project, Task Managers, Lead Engineer, Developers, VER & VAL, Specialist

As you move through this table note the obvious – more detail is given at the program and project level to ensure accuracy and decrease variance (Remember the Reductionist!). We don't run every project or system differently – we have a standard that allows for a trained person to test any equipment, to develop any system, to track cost and schedule if the method and the tool are the same. Retraining for each project, or system, process variance and uncontrolled activity, differing tools or metrics change the scenery, the dashboard and resources are wasted in continually 'catching up'. There is great value in uniformity leading to improved operations, systems that work, increased morale, better use of resources and of course, a greater profit.

You will note that I highlighted the **Systems Engineering Management Plan (SEMP**). The Systems Engineer is responsible for writing this document. It includes the plan on how the system will be engineered from the start to finish, and calls out the processes used, roles and responsibilities, tool and resources. This is the backbone of running a good program for the Systems Engineer, just as the Project/Program Management Plan (PMP) is for the Project Manager.

I also highlighted the **Systems Engineering Integration and Test (SEIT).** The SEIT is a Senior or Lead Systems Engineer of a program, or a division level who is responsible for all the Systems Engineering activities of that program (or division). The SEIT is commonly responsible for the full range of Systems Engineering activities, but those working these activities are matrixed to the SEIT as needed. In other cases, the SEIT may have the full systems team working directly for them. The SEIT will be especially busy at a program and systems start up but as the system enters operations, their activity decreases and the full staff may not be needed unless there is an upgrade or a problem to resolve with the system. When not needed, the Systems Staff can easily support other programs and systems.

1.8. Useful Paradigms as Mental Models

The Project Management Institute (PMI) uses the triad of **Cost, Schedule, Performance and Risk** in their projects (PMI, 2020). I find this extremely useful as a 'mental model' (or paradigm) when managing or assessing a project. The other paradigm is **People, Process and Tools** which I have seldom found

documented, but something I learned over the years. These paradigms prompt the Systems Engineer or the Project Manager to ask the right questions.

1.8.1. Cost, Schedule, Performance and Risk

Figure 1-8 shows the Cost, Schedule, Performance triad. Costs are associated with budget, projected and actual resources consumed, charging to the right task, compensation and overhead, travel cost and profit or loss. Schedule is associated with a milestone chart, the project Gantt Chart and day to day operational rhythm. Performance is the characteristics and quality of the ongoing tasks of the project, the amount of re-work, and the function of the system. Risk is an event, that should it occur, will have a negative impact on the project or the system. Often a risk in one project area eventually affects the other two areas. If a task is in trouble because of a parts shortage, it will affect cost and may affect schedule due to the potential of working with a new and likely more expensive supplier. The Project Manager is well suited to ask on an ongoing basis, "What may go wrong, and how do I prevent it?" A wise Project Manager knows that 'Murphy is on every Project'[3].

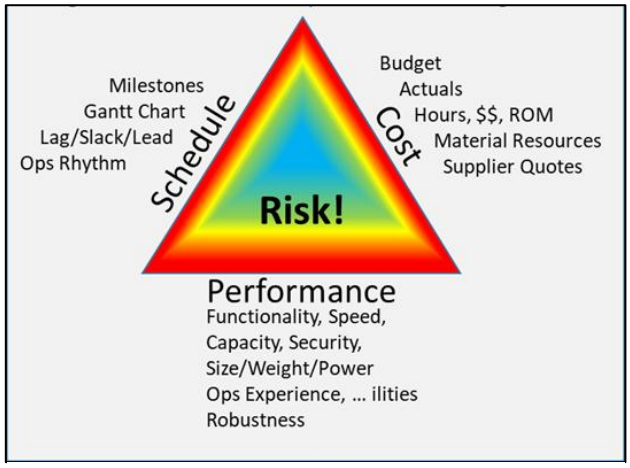


Figure 1-8 Fast and Cheap, or Higher Cost and Higher Performance

Although Risk Management has always been important on projects only recently has it gained the respect needed to make it useful. However, many organizations go through the routine of performing risk management with little or no results. Because of this, I have devoted Section 3.1.7 on Risk, Issue and Opportunity Management.

1.8.2. People, Process, Tools

The Systems Engineer thinks in terms of scheduled tasks and the technical variables affecting the system. A common systems way of thinking is People, Process and Tools (as first discussed in Section 0). **Figure 1-9** illustrates each. The system engineer will check their work by answering these questions.

- What **People** are used on my system? Are they available and trained? Are they cleared? Are they occupied on other tasks? Are they getting the care and the recognition they need? Are they held accountable for their work?
- What **Processes** are used on my system? What processes are applied to design and develop the system? Are processes complete? Are processes producing the needed result? Do I have all prerequisites of the process? Are processes measured? Do the processes add value?
- What **Tools** are used on my system? Do I have the needed labs? Do I have measuring equipment for testing? Do I have software to manage requirements? Do I have software to document my schedule and my drawings? Do I have the needed infrastructure?

[3] Sergeant Murphy is a fictitious World War II Soldier that could not seem to correctly perform his job. It is the basis for Murphy's Law – that if it can go wrong, it will. Each mission was made 'Murphy Proof' because Murphy is on every mission. Murphy's Law has carried over to the civilian world.

I've added a fourth element to this paradigm which I call **Input**. Often overlooked, it includes data, raw material, and energy. In short, processes must have a standard for what the Inputs will be or the process will not function, or tools may not work (or may break). Remember garbage in, garbage out.

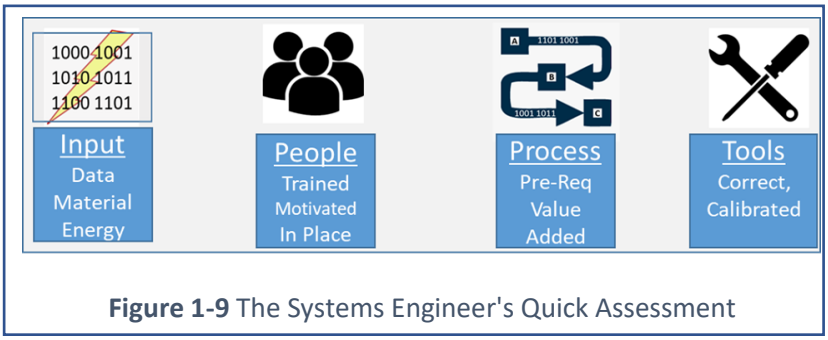


Figure 1-9 The Systems Engineer's Quick Assessment

This People, Processes, Tools, and Input paradigm provides a quick reference to the Systems Engineer validating that the system has what it needs for development and sustainment. It is best to ask these questions early and often to ensure the right people are on the team, the processes are known and applied, and the tools are available. These elements (plus the input as indicated by the lightning bolt and bits) must work seamlessly to deliver a mission solution.

1.9. The Readiness of Technologies (Science and Engineering)

Science and Engineering are two complimentary disciplines. Science is a systematic effort that builds organized knowledge and creates testable observable hypothesis, and theories of physical and natural sciences. Engineering will apply these discoveries by designing and building into structures, machines, vehicles, and electronic systems. Both use similar techniques and terms such as labs, modeling and simulation, behavior, and so on, but the Systems Engineer must not become a Scientist and expect to deliver functioning systems.

Many organizations and projects design and field systems with new and emerging technologies that are on the leading edge of science. Those working these systems are 'first at the gate' in resolving newly discovered problems related to interfaces, security, functionality, training, support, performance, and many other challenges. Hopefully, you can see the relationship from the proceeding section of Cost, Schedule, Performance and Risk, and People, Process and Tools. Introducing new technologies that may have been proven worthy in a lab are not the same as building the new technologies into a system. A Systems Engineer must be diligent on how the emergent behavior reacts to other systems or sub-subsystems, and its environment. A common method to measure the readiness of technology is through the Technology Readiness Level (TRL) (Technology Readiness Assessment (TRA) Guidance, ASD/R&E, TRA Guidance, April 2011). (Assistant Secretary of Defense/R&E, 2011) TRL levels 1 to 9 are illustrated in Figure 1-10. TRLs are a useful maturity model to manage the risk and the opportunity of deploying new and emerging technologies. TRL is a common guide but does not fully account for the risk associated with size, weight and power, it does not account for form and fit, nor does it account for all operational and environmental risks.

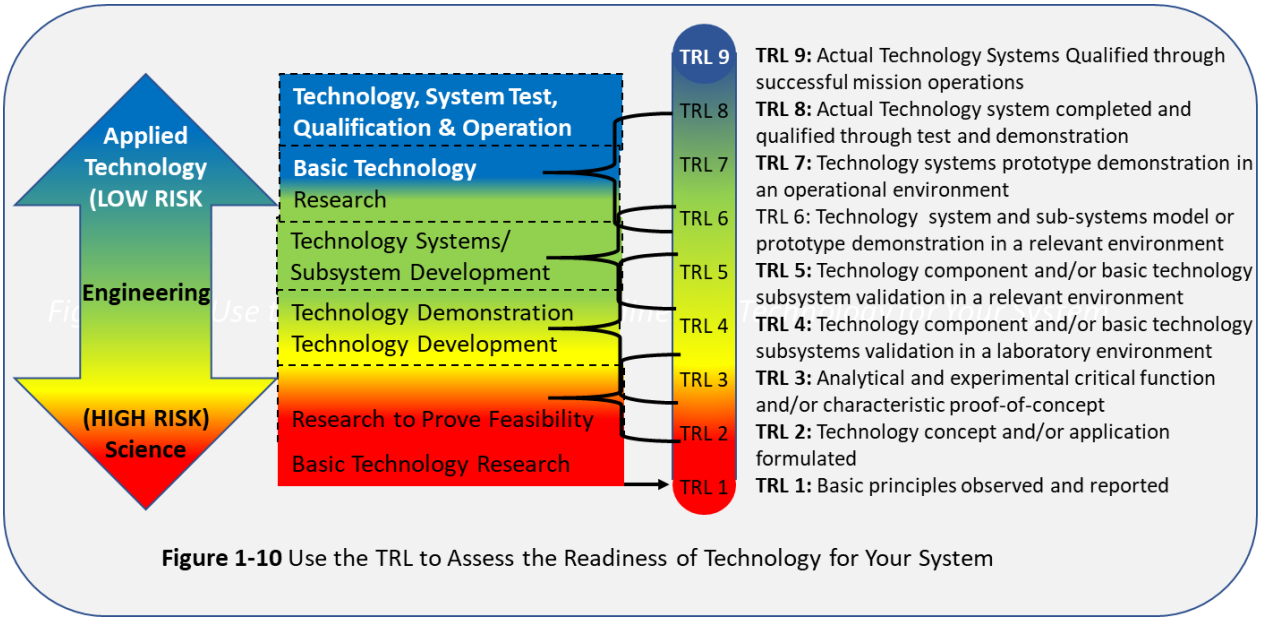
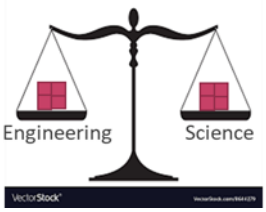


Figure 1-10 Use the TRL to Assess the Readiness of Technology for Your System



Not all systems or products have a TRL, but the guide is useful to assess readiness and is useful to query a supplier on the maturity of their product and system. If you understand how the TRL works, and you or the right expert on your team understand the product of interest, you will know what to evaluate of the product compared to the TRL – in short performing your own informal TRL assessment. A system engineer can ask if the technology works only in a simulated environment (TRL 5 Medium Risk), or if the technology has been deployed and is in use in its operational environment (TRL 9 Low Risk). Then ask the provider for documented evidence that it performs as advertised. If necessary, consider performing your own assessment.

Leading edge is OK – Bleeding Edge is NOT OK

1.10. Summary - Systems Engineering Fundamentals

Modern Systems Engineering really came of age in the early 20th Century because it takes system thinkers and professionals to build big interdependent and complex systems. Systems are decomposed to understandable levels for each domain to solve, then integrated into a greater system that present Emergent Behavior. Project Managers think of the Project Triad of Cost, Schedule, Performance plus Risk, while the Systems Engineer's perspective is People, Process and Tools. But to be safe the Systems Engineer should use both paradigms to quickly identify and prevent or resolve problem areas for the system.

2.0 The Systems Lifecycle

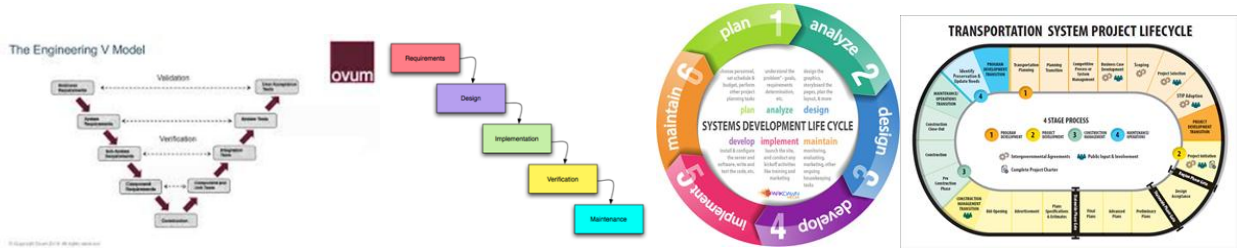
Before starting The Systems Lifecycle (the 5,000-foot level), I'd like to provide these definitions that are used in this and subsequent sections. I will not get stuck on these definitions because there is overlap between the term methodology and models, and even framework, but as for Systems Engineering, it important to understand their meaning.

- Framework – the loosely defined guidelines, offering boundaries, on how a set of activities are performed. A framework offers a repeatable set off methods that may be used on different projects.
- Model – to form or plan – There are two related meanings to a model. One is modeling and simulation as described in Section 1.3 What is Systems Engineering (see Modeling and Simulation) The other meaning is related to lifecycles as a practiced set of activities applied to realize a system. Project models, like systems models, can be measured, show relationships and used to illustrated how something will function – be it a project model or the systems modeling and simulation. A project model should use a framework as the structure in which the model is created.
- Methodology- Orderly practices, techniques, procedures, and rules used by those who work in a discipline. A methodology may offer opportunities for innovation or change but work within the guidelines of the defined methodology. For brevity, the term Method will also be used.

The Systems Lifecycle is one of the topics of Systems Engineering to which I don't necessarily follow convention. Traditionally, there is the Systems Lifecycle which offer a **model** to how a system is envisioned, developed, delivered, and used. INCOSE states there are three types of models 1) Sequential, 2) Evolutionary, and 3) Spiral – of which all three are discussed (INCOSE p35-38) Within a lifecycle there are **methods** for collecting requirements, architecting, and designing, another for development, and for testing, and how to perform the support activities such as documenting the configuration, and performing risk management. These methods and models have come to overlap and intertwine so the development method dictates a lifecycle model. Additionally, if it's called a 'lifecycle' why are many lifecycles laid out in a linear schedule – start to end? It's time to rectify this. How this is done – a cycle, a Systems Vee or a linear schedule is not important but what is important is the right method and model is selected and that stakeholders understand it and use it. As for a **framework** – this is yet another loosely defined but very relevant term that provides guidelines of the prevailing tenants of how a project or system is managed, to which useful structure is provided (processes, decision gates). While these terms are important, just remember, that one of the tenants of good systems engineering is simplifying complex ideas through an abstraction, and ensuring this is communicated to stakeholders.

Unfortunately, many of those working in the systems field will claim any one of the frameworks, models or methods are best – for every project or system. They hold steadfast to their claim as if their favorite way is the only way, the savior for all projects, but they usually have little understanding of alternative models or methods or have had bad experiences with any one of the three. So be patient, don't be afraid to modify the lingo, or the look to avoid an unnecessary disagreement, and then move forward.

Often, organizations create artsy looking business and lifecycle processes that are not practical or useful. Pick the most suitable one or a combination of more than one. A small iconic image of each is provided below but we will go into more detail of how they are used, and their contents.



There are two approaches of the Systems Lifecycle. One is **Incremental** and the other is **Iterative**. Incremental is a measured and scheduled approach. A set of activities are performed according to the schedule and once completed the next set are started. Iterative occurs when some work is performed in more detail or repeated to refine the problem or the answers. Iteration is especially hard to schedule activities because of the continual discovery of new information, but Incremental lends itself more to scheduling the activities. You'll see how these apply in these lifecycles.

2.1. The Systems Vee – A Framework

A concept that Systems Engineers always have in their mind (or should) is the Systems Engineering Vee, depicted in **Figure 2-1**. If you haven't already done so, I highly recommend you become mindful of the Systems Engineering Vee and to continuously use this to link the project as a thread of consistent activities supported by people and process. It links the Operational Concept to the Operational System, and the Functional and System Requirements to Testing. There are 'way points' with the little boxes (thumbs up or down) to assess progress and readiness to proceed into the next phase.

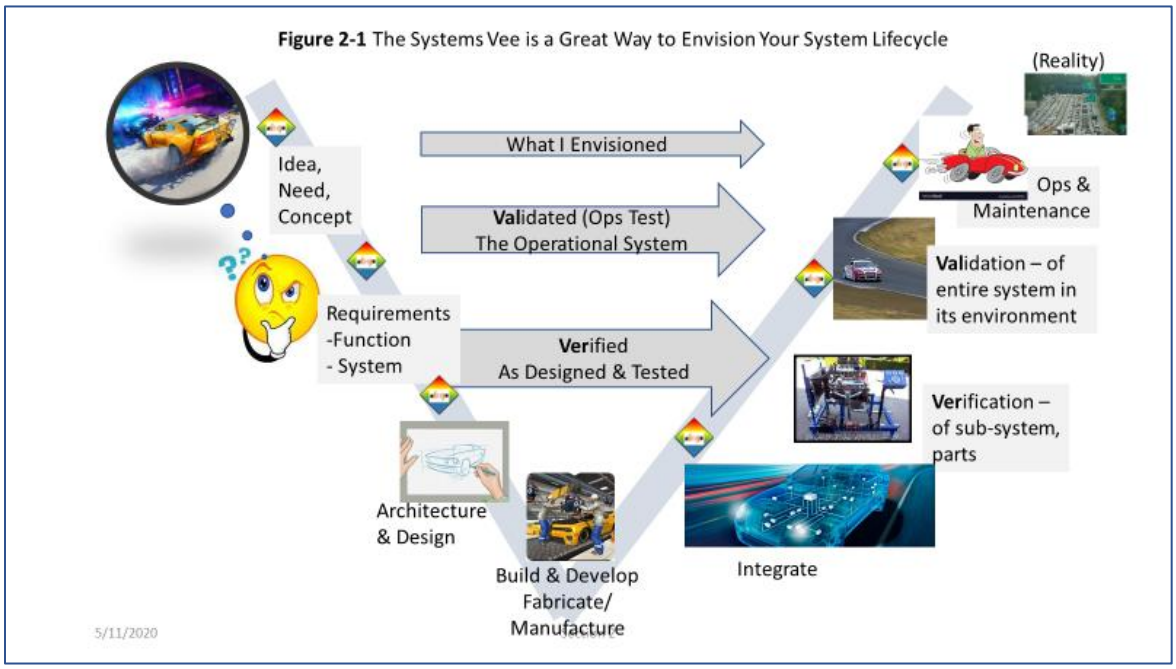


Figure 2-1 The Systems Vee is a Great Way to Envision Your System Lifecycle

The operator has an idea or a need – in this case the 'need for speed', but the operator doesn't quite know what that device is – a rocket, a car, skateboard, or motorcycle. The operator describes what they

want to do – go fast, wind around roads, burn rubber, not waste gas doing so, etc. From this concept and the operator ideas, the Systems Engineer will aid the operator in defining the requirements. Someday, the system will be operationally tested to ensure it meets the expected operator experience – which is the 'Validated' arrow leading to a track test. Each requirement is listed in a ledger as a function, and then synthesized into what part of the system may perform the function. Again, at some point in the future, each one of the requirements are tested to ensure compliance (Verified arrow) – see the engine in the engine test stand. Each part of the system is tested as it is assembled. Once the best alternative is selected – a motorcycle, a skateboard, or a car-development and fabrication start. Frequent testing is accomplished to ensure it will function as needed. Although integration starts during the Architecting and Design activities, it is really performed and tested during the Integration activity. Finally, the Verification is completed, and the Validation is completed, and if it meets all the Decision Gates, it is delivered to the operator.

Figure 2-2 Successive System Vee illustrate continual Release – Each Smaller Vee Requiring the Same Rigor of the Large System Vee

Figure 2-2 illustrates a System Vee interlaced with smaller System Vees representing how a major delivery is accomplished with the Large Systems Vee, and how in between the major deliveries there are smaller version releases. Each of the smaller releases are used to repair defects or add features, and must be carefully managed as a sub-set of requirements, each requiring architecting, developing, Verification and Validation. Only when the release is ready will it be carefully transitioned to an operator. The Systems Vee is a mainstay in the Systems Engineering Industry and is a remarkable model to continually align the needs to the operations, the concept to Validation (Operational Testing), the Requirements to Verification, and what roles the stakeholders play. The Systems Vee does not show the detailed processes to inform the System Engineer of HOW these activities are performed. The detail of HOW comes later in this Book. Additionally, the Systems Vee does not lend itself to a real schedule because it's a framework.

2.2. Transform the Systems Vee to a Linear or Circular Form – Now a Method

By laying the Systems Vee flat (as illustrated in **Figure 2-3**) you should see how the project shapes into phases and a schedule. The Systems Engineer will work their way through each one of these phases and Decision Gates using a plethora of tools and 'tricks of the trade'. The Systems Engineer may select a circular form (**Figure 2-4**) of a lifecycle model depending on the needs of the project, system and organization.

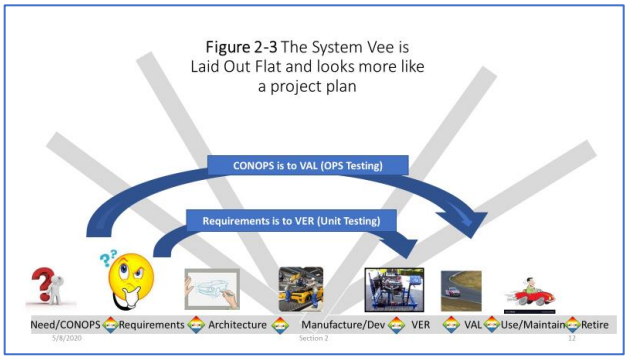


Figure 2-3 The System Vee is Laid Out Flat and looks more like a project plan

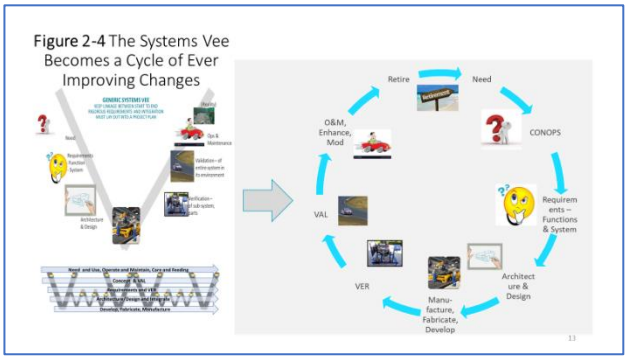


Figure 2-4 The Systems Vee Becomes a Cycle of Ever Improving Changes

A combination of a Systems Vee to start the project will likely transition into a linear or a circular form, or a combination of both. Neither the cycle nor the linear method will provide the needed detail of a schedule, or the activities needed for requirements, verification, validation, architecting, risk, and configuration management, and more.

2.2.1. Circular Lifecycle - Framework

The Circular Lifecycle epitomizes the concept of a life**cycle**, hence there is little confusion of its purpose and what it means. A typical Circular Lifecyle is illustrated in **Figure 2-5**, this one for a software development organization and system. You'll see the six phases plainly described, with the activities of each phase in the center. It is most suitable for existing systems that are augmented with features added and not so much for systems in which the basic CONOPs and requirement have yet to be documented and understood. It is simple and visually appealing for many types of stakeholders to understand, yet it satisfies the needed purpose of managing the project. A cycle schedule is established into a regular operation rhythm so the team can go through the cycle once every three months or so. This model still requires detailed processes to manage the Cost, Schedule, Performance, Risk, Configuration and other processes. The Circular is still 'high level'.

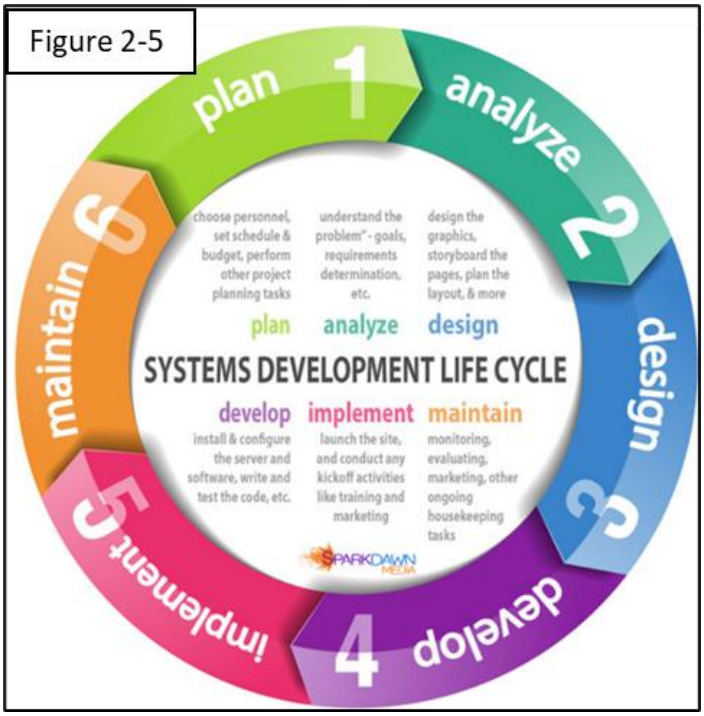
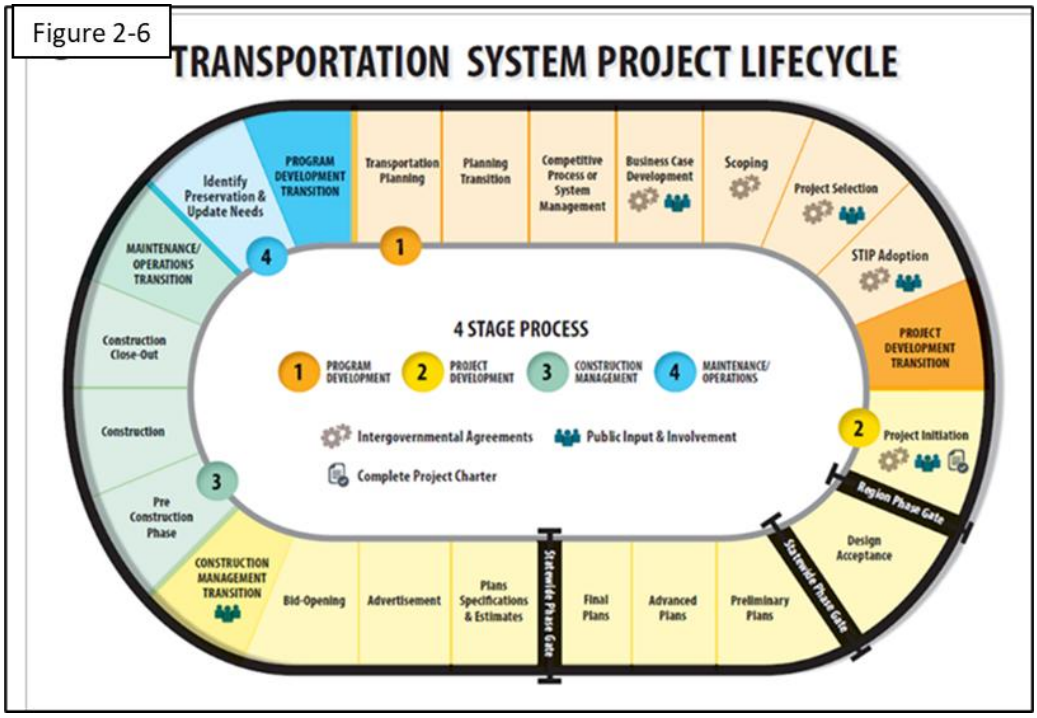


Figure 2-5

2.2.2. Oval Lifecycle - Framework

The only real difference is the oval model from the circular is that it is an oblong version of the circular. But this Oval (**Figure 2-6**) is interesting because it represents all the activities needed to pass through

Figure 2-6

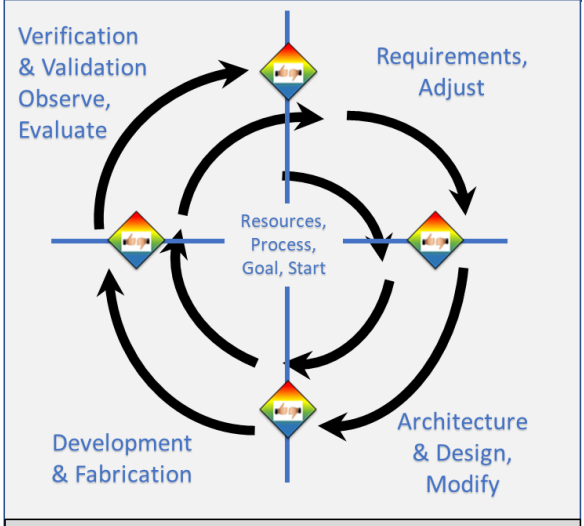


rigorous approvals of the State of Oregon Public Transportation System (Strickler, Kris, 2020). Because it must appeal to the public while offering a model to manage a series of projects, it contains a color code that stands out and is immediately clear. It has the four staged process with the public, environmental, budgetary, and legislative approvals. It contains a solicitation process to hire the most suitable construction company. It requires a transportation strategy and concept – emphasis on new roads or rails, or both for example. It is quite suitable for the use in existing projects that need upgrades and added features, yet a new project may be inserted into the oval. The detailed processes though, are what will bring this to life.

2.2.3. Incremental (Spiral) Method

If an operator has a need but they are not sure what it is, or how best to resolve this need, spiral may be the most suitable model (**Figure 2-7**). The Systems Engineer will work with the operator to best determine WHAT they need and create a set of loosely defined requirements. The system is designed top down in a hierarchy, delivering a general capability at first. The capability is demonstrated to the operator who will approve the next spiral with more detail and added changes. Obviously, spiral requires user participation which users often don't want to do – participate. So, a proxy operator or user (functional expert) may be hired to work on the project team. This allows for requirements and architecture changes (but must still be documented and traced). The Project Manager may have less insight into the budget and actual resources consumed, and it is prone to delay if the operator changes the requirement or is not available (which occurs often). It's often hard to know when the project is finished but spiral can be used to discover new and innovative ideas. Spiral is effective for managing risk of new innovations, to ensure nothing is built into a system that goes to operations before it is ready.

Figure 2-7 Spiral – Develop, Assess, Adjust

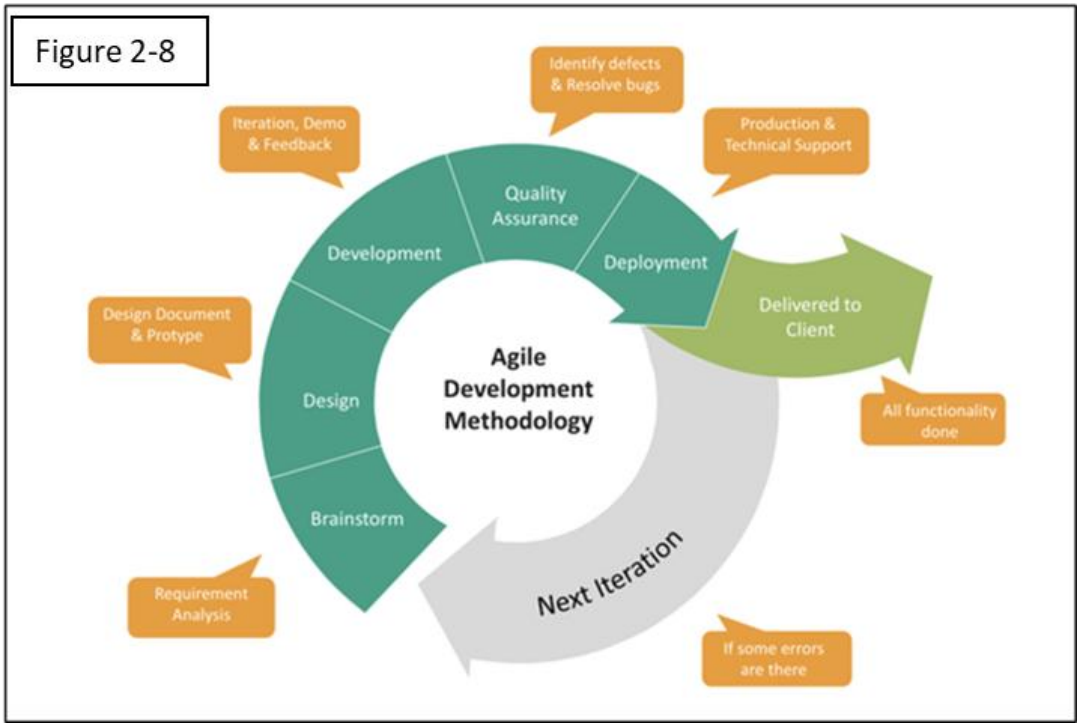


Each Spiral is an Ever Maturing Capability with Continuous Risk Assessment, Control, Adjustments

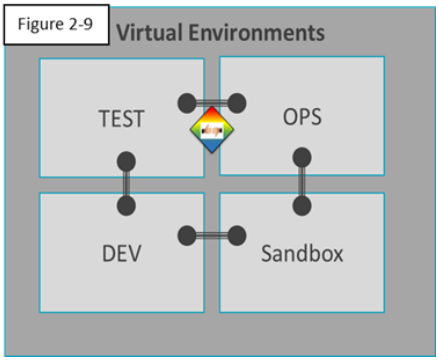
2.2.4. Agile Development Method

Agile started as a software development method but has expanded to hardware and the entire lifecycle. I have also seen it in government (agile government), business operations and more. The idea that Agile will fit all programs tends to distort the great capabilities of Agile. Given the Cost, Schedule, Performance and Risk triad, Agile will 'time box' schedule at a set date (say 60 days) and whatever the team has developed is turned over. Whatever requirements (or user stories) they didn't develop go into a backlog for future development. Because of this, it's hard to imagine something not quite complete or not thoroughly checked out as suitable for going into space or used to monitor a critically ill patient.

Figure 2-8 illustrates how users and Systems Engineers brainstorm ideas, which are then documented into the User Stories. Based on the needs of the User Stories the process moves to design to ensure a viable solution is planned, and in development that this solution is realized. Defects are identified and corrected during Mission Assurance (or Verification and Validation), and then subsequently deployed. The next iteration is planned and the process starts again with a new set of users stories.



Agile has literally hundreds of flavors and versions. Requirements are created in the form of 'User Stories …' that say 'As a User I want to watch television', or 'As a User I want to go fast'. These User stories too, need decomposition and more understanding. Each 'time box' is called a sprint, and several sprints make up an increment. Parallel sprints may occur within an increment. A Scrum Master will facilitate the entire process, and the Product Owner is the proxy User who decides which User Stories will go into a backlog and resolved at each sprint. An Agile team is a composite team of the Scrum Master, the Product Owner, developers, testers and probably someone to document the results. Often Agile is not well documented but it still requires all the activities of a normal lifecycle such as Architecting, Selecting Alternatives, Verification and Validation, Risk Management, Configuration Management, and Integration. It is tempting to focus on development and not so much the detailed processes in Agile, resulting in delay and disappointments.

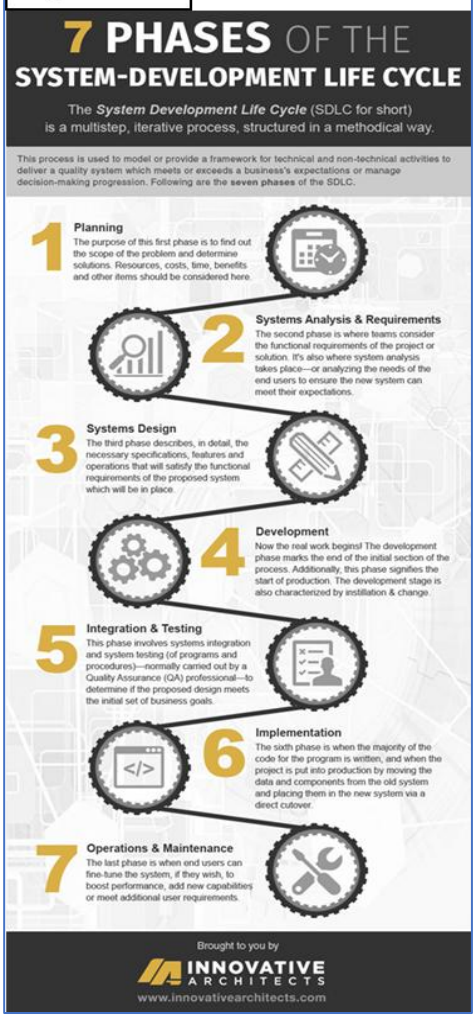


Agile and 'Dev-Ops' go hand in hand. The virtual environments are illustrated in **Figure 2-9**. Although not exclusive to Agile, Dev-Ops (longer name is Development – Operations) is when separate but almost identical virtualized computer environments are configured and development occurs in as a realistic operational environment as possible. So, after the 60-day sprint is complete, and the product is Verified and Validated, it is approved at a Decision Gate and 'promoted' from the Development Environment to the Operational Environment. Other environments may be set up too, such as a Test environment, or a Staging environment or even an experimental environment called 'sandbox'.

2.2.5. Seven Software Phase Model

This Seven (7 Phases) Model (**Figure 2-10)** is another presentation of a linear model but interesting enough, it is in a vertical format. Innovative Architects used this model (Innovative Architects, 2020). Each phase has a specific purpose starting with planning that includes the requirements and scheduling, the system analysis (diagnosing the problem), requirements elicitation and decomposition, and systems design and solutioning that will include the most suitable design option. Next is development and integration and testing (Verification and Validation). A specific phase for implementation implies this activity is risky and requires more attention to detail. As in any other system there is the Operations and Maintenance (upgrades, added features, monitoring), and what is not listed is 'retirement'. Again, this model is easy to understand and would be suitable to show a potential customer that this company can be trusted to develop and deliver software. Given the extensive implementation Phase I would suggest it works well with far reaching and interconnecting systems. The meager amount of testing may not be in favor of delivering a flawless system.

Figure 2-10



2.2.6. Waterfall Model

While the Systems Vee is an excellent illustration the most traditional is the Waterfall Model (**Figure 2-11**). In fact, the Systems Vee can easily transition into a Waterfalls model in the sense it is a cascading set of phases. When one phase is complete the team can move to the next phase when approved at the Decision Gate (see the square thumbs up or down). Waterfalls is excellent for those systems that are inherently risky such as military command and control, aerospace systems or weapon systems. It was the mainstay for System Engineering for most of the 20th Century and has created many great systems. It can be slow and prone to delays but that is because of the associated risk and Decision Gates – to not grant approval until the

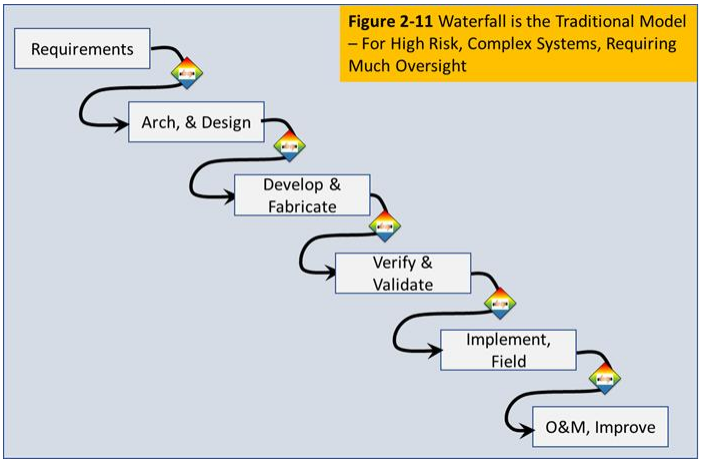


Figure 2-11 Waterfall is the Traditional Model – For High Risk, Complex Systems, Requiring Much Oversight

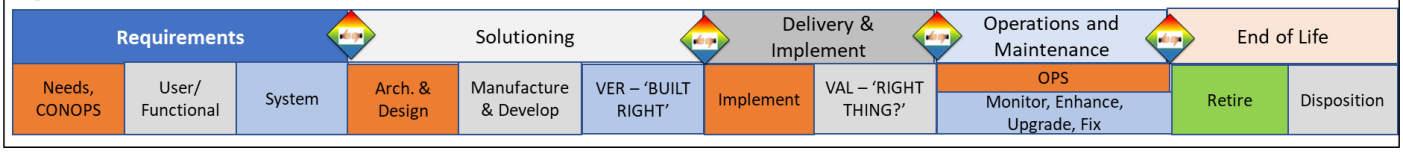
next phase is of acceptable risk. It's hard to rush perfection. Again, the detailed processes are not illustrated here, such as how requirements are managed, how the system is designed, and implemented. This model shows nothing of Integration for example.

Waterfalls has been given a bad rap lately. While it suggests each phase is started only at the approval of a Decision Gate, limited work of the next phase can be performed if granted limited approval, thereby accelerating the schedule. Tasks can be done in parallel and some material, software or hardware can be reused from other similar projects to decrease cost and improve schedule. But once progressing through the phases, stepping back to make change can be costly.

2.2.7. Linear

The Systems Vee easily transforms into a linear systems lifecycle as illustrated in **Figure 2-12**. You can easily see the phases, activities and the milestones. In this generic example there are five distinct phases which are broken down – thus allowing diligent management and measurable progress. Risks are well defined; schedule and budget are matched to the activities. As in the other lifecycle models, recursion and iteration are performed but the Systems Engineer must not step back (recursion) to more than a step proceeding or there is a great risk to the project. That's why before proceeding from a milestone, the Project Manager and Systems Engineer must determine that the tasks required are completed. **It is not bad** to step back from Manufacturing and Develop to Architecting and Design if a minor adjustment must be made. **It is bad** if one must step back from Implementation to User/Functional Requirements to modify a requirement. This latter situation causes a cascading set of costly changes because of re-planning, re-work, and delays. Linear is a great and intuitive visual and is best suited for new systems in which the requirements must be elicited from users, and synthesized. Once discovered, and accepted, there will be little tolerance for change – i.e. the aforementioned cascading changes. As an example, Linear is most suitable for large Command, Control, Communication (C3) Systems, Aerospace or large and complex information technology systems.

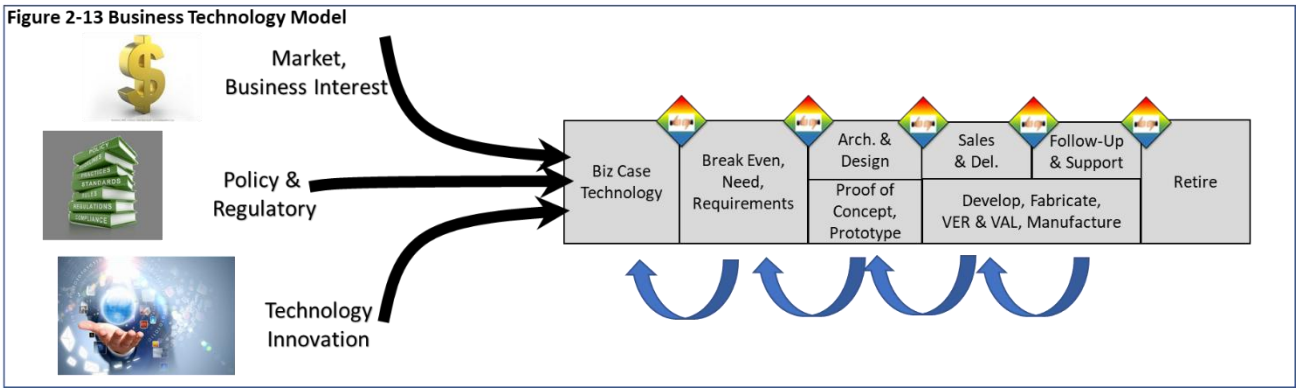
Figure 2-12



2.2.8. Business Technology Model

If you are working in a business that depends on new technologies that you will design and build into new systems, the Business Technology Model (as illustrated in **Figure 2-13**) may be for you. It is a form of a linear model but also includes the market demands, policy and regularity restraints, and the maturity of technology. This is a model that I created to show how a lifecycle should be aligned to the

Figure 2-13 Business Technology Model



interests of the organization, in this case a large business. This model is suitable for large systems which pose risks that may take years to yield a profit. Those three interests are:

- **Market and Business** – Will there be a demand for the system? What will be the price point, and how many can you sell? How long will it take, and are you missing other opportunities that may be more profitable? Will you need to partner with another business? Who are the customers, and what do they need, and what will they need and don't yet know they will need it? How will we sell and deliver the system?
- **Policy and Regulatory** – Will it be permitted based on current or impending legislation? Will an autonomous vehicle be permitted? Is radio spectrum available in which the system will function? Will the system create health or safety issues? Policy and Regulatory ultimately reflect the desires of society although it may not always seem like it.
- **Technology and Innovation** – Recall the technology readiness level in Section 1.9. Will the technology be ready and reliable to integrate into a system? In the case of the radio spectrum, will the antennas, or multiplexers, the up and down converters be technically viable? As for the autonomous vehicle – will the sensors provide the buffer of space and performance margins to allow the vehicle to operate safety?

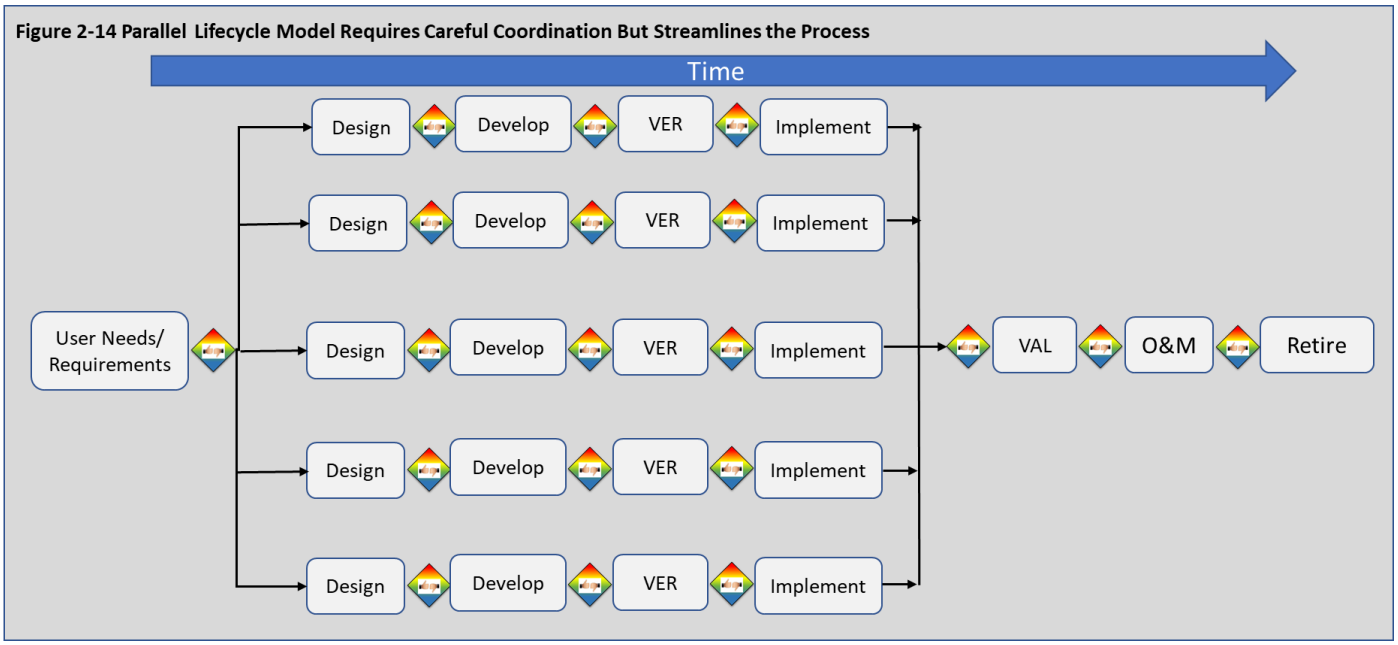
In the case of the linear model, professionals who are experts on these three domains collaborate on these three issues to create a business case, several designs and options are created, modeled and simulated, and plenty of proof of concept, alpha versions and prototyping are performed to assess the risk and readiness. Salespeople must go meet the customer, hear their concerns and return to the organization with more information to improve their system. Once the system is fielded and in operation a rigorous support system must be included with upgrades, repair and improvements - customer says, 'what have you done for me lately?'. As the system is designed, developed, and fabricated the Systems Engineer must know how the system will be disposed someday to include the disposition of broken parts or the entire system. Examples of these systems are 5th Generation Cellular, commercial aircraft and the autonomous vehicles. A common problem I've seen in industry is only two of these three are considered (usually omitting the Technology Considerations) to "go To Market' before it is technically reliable. The Systems Engineer (or the technologist) must make the case to include all three elements.

2.2.9. Parallel

Parallel Lifecycle Model is an excellent way to accelerate the scheduled activities by doing multiple sub-projects simultaneously. This is commonly done on any one of the previous lifecycles but is illustrated

as an example in **Figure 2-14**. It all starts with the Need and Requirements but is parsed into smaller sub-projects – for a large data center one sub-project may be for the building, one sub-project may be for the power, water and networking system, another may be for installing the equipment racks and so on. These parallel activities must be well coordinated and synchronized so when it's time to integrate the systems, the power fits into the racks, the water pipes carry enough water or coolant, the networks will interface with the equipment. It requires detailed configuration and document management, and incremental and well managed progress. If this approach is successful, the system will be realized and delivered much sooner than it would be if the model were a linear model. The requirements must be fairly stable from the start to the end due to the cascading effects of change across multiple paths. Most projects that have a repeatable set of activities will find the parallel quite useful – such as repeatedly building many data centers, or many aircraft, or automobiles.

Figure 2-14 Parallel Lifecycle Model Requires Careful Coordination But Streamlines the Process

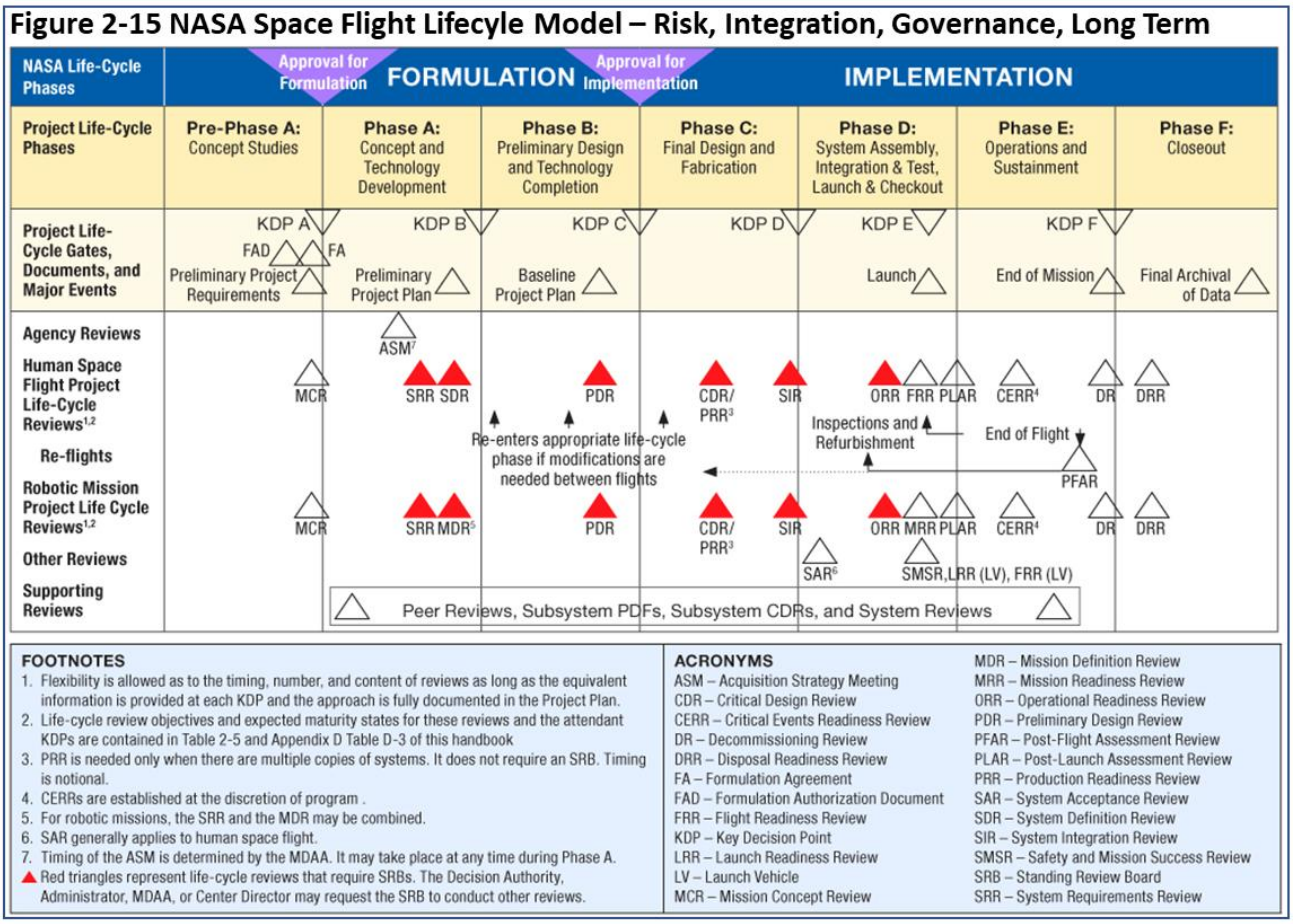


2.2.10 National Aeronautics Space Administration (NASA) Space Flight Lifecycle Model

The last two Lifecycle Models are more complex because they are used in mission critical and complex systems. People may quickly perish if not done correctly, and there is a large amount of needed capital to design and build and operate the system. These lifecycles include integrated activities to policy makers, budgeting, and the public. **Figure 2-15** is a Lifecycle for the National Aeronautics and Space Administrations (NASA), Formulation and an Implementation Phase (Hirshorn, Steven R., 2007) that is then broken down into seven sub-phases. Each phase is carefully planned to capture and maintain the focus for the U.S. National Interest and Goals related to space missions. Note how NASA will carefully evaluate the readiness and usefulness of technology and how it may be used for the NASA mission. They work with industry to create the most viable solution for NASA. Note the numerous and rigorous Decision Gates and reviews performed with differing interests – human space, robotic, agency reviews and approval – only proceeding when the risk is acceptable. Also, note how terms and icons are defined so all stakeholders can view this and quickly understand how this lifecycle works. This lifecycle will also embed forms of circular, parallel, and even agile although these models are not seen within this

lifecycle. This is a large chart on a small page so for more detailed look please go to https://nodis3.gsfc.nasa.gov/displayDir.cfm?Internal_ID=N_PR_7120_005E_&page_name=Chapter2

Figure 2-15 NASA Space Flight Lifecyle Model – Risk, Integration, Governance, Long Term



FOOTNOTES

1. Flexibility is allowed as to the timing, number, and content of reviews as long as the equivalent information is provided at each KDP and the approach is fully documented in the Project Plan.
2. Life-cycle review objectives and expected maturity states for these reviews and the attendant KDPs are contained in Table 2-5 and Appendix D Table D-3 of this handbook
3. PRR is needed only when there are multiple copies of systems. It does not require an SRB. Timing is notional.
4. CERRs are established at the discretion of program .
5. For robotic missions, the SRR and the MDR may be combined.
6. SAR generally applies to human space flight.
7. Timing of the ASM is determined by the MDAA. It may take place at any time during Phase A.

▲ Red triangles represent life-cycle reviews that require SRBs. The Decision Authority, Administrator, MDAA, or Center Director may request the SRB to conduct other reviews.

ACRONYMS

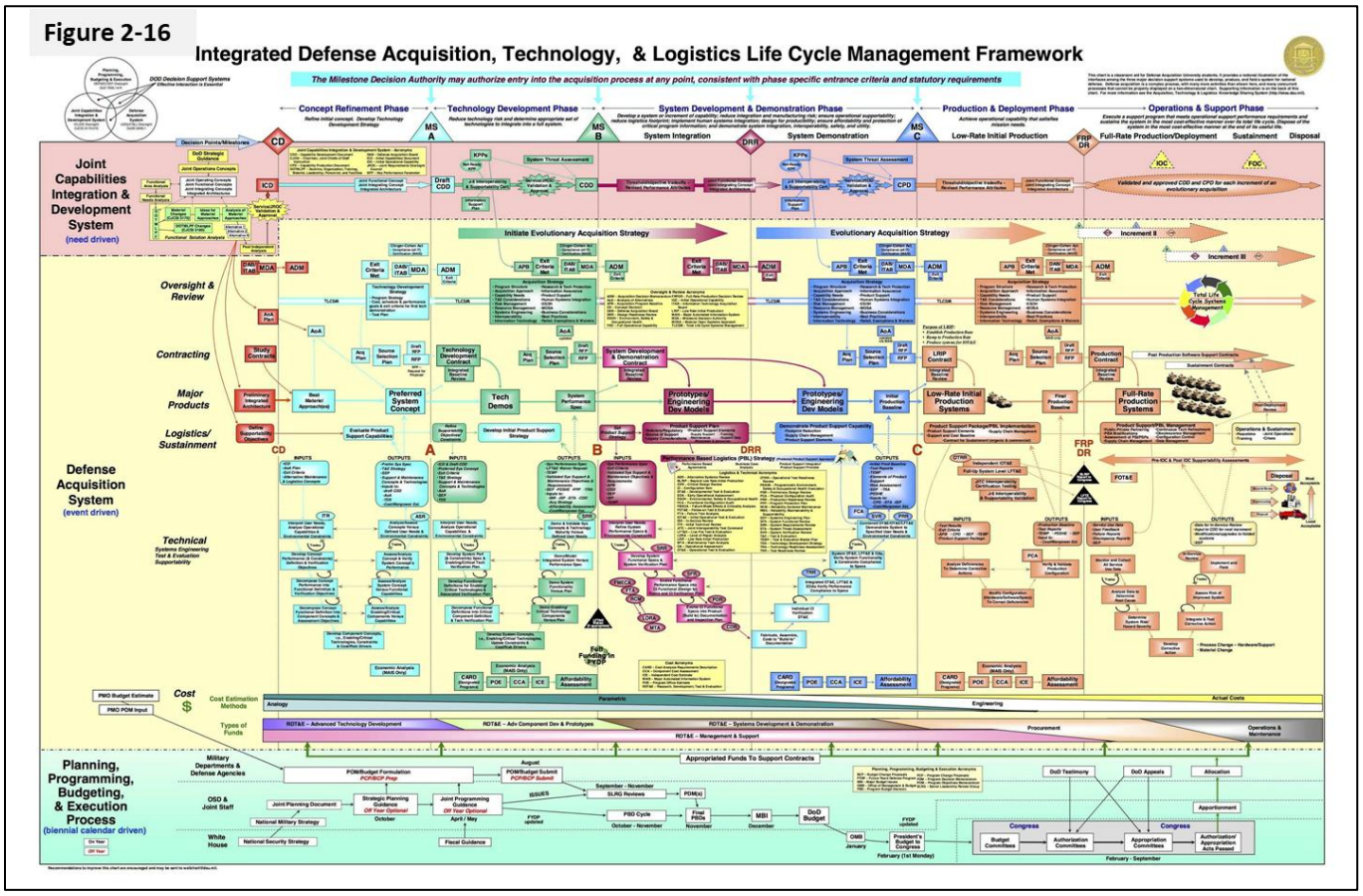
ASM – Acquisition Strategy Meeting
CDR – Critical Design Review
CERR – Critical Events Readiness Review
DR – Decommissioning Review
DRR – Disposal Readiness Review
FA – Formulation Agreement
FAD – Formulation Authorization Document
FRR – Flight Readiness Review
KDP – Key Decision Point
LRR – Launch Readiness Review
LV – Launch Vehicle
MCR – Mission Concept Review
MDR – Mission Definition Review
MRR – Mission Readiness Review
ORR – Operational Readiness Review
PDR – Preliminary Design Review
PFAR – Post-Flight Assessment Review
PLAR – Post-Launch Assessment Review
PRR – Production Readiness Review
SAR – System Acceptance Review
SDR – System Definition Review
SIR – System Integration Review
SMSR – Safety and Mission Success Review
SRB – Standing Review Board
SRR – System Requirements Review

2.2.11. Department of Defense Lifecycle Framework

The most complex is the Integrated Defense Acquisition, Technology, & Logistics Life Cycle Management **Framework** as illustrated in **Figure 2-16** (Kobren, 2017). It's not called a model because of the vast, epic size of the lifecycle that rightfully includes the many interests of our national defense – from the policy makers, the current and future security threats, budget authorities, contractors and of course the U.S. Taxpayer. Rumor is that if every step and process is used in this lifecycle it would take 47 years to deliver a fighter plane. But not every step must be performed, and many steps are done in parallel. The F-35 Multi-Role Fighter Plane started as a program based on technology from the stealth fighter in 1995 and then it first flew in December 2006. It went into service in July 2015 and will go into full rate production in 2021. Many are to be sold to our allies which will subsidize the cost to the United States. The F-35 is designed to include upgrades and modifications that account for changing threats and new technology, and it is expected to fly well to 2070 – seventy-five years and $1 Trillion after it was first designed. All weapons systems, command and control, and support systems must be well integrated and interoperable, and supportable to deliver the maximum affect. That is why this Lifecycle is so detailed – the cost is great

and our national security depends on it. For a close up view of this lifecycle you may go to https://www.dau.edu/training/career-development/logistics/blog/Updated-DoD-Acquisition-Life-Cycle-Wall-Chart

Figure 2-16



2.3. Summary - Lifecycle Models, Methods and Frameworks

In Section 2 I've presented a mix of Lifecycle and Development Models, Methods and Frameworks. Because the three tend to overlap I did not present separately. By now you should see how the Systems Vee provides a good visual of relating the need to operations, and requirements to Verification and so on. A Systems Engineer can transform the Systems Vee into circular lifecycle or lay it flat depending on the organizational needs and the type of systems they are developing. Lying flat is most suitable for newer type systems that may not yet exist, but for existing systems requiring a continual upgrade the circular may be best. It is not unusual for an organization to use a hybrid or a mix of a linear and a circular lifecycle. In any case, approvals by various stakeholders are needed, and these approvals must be inserted in the right part of the lifecycle. Complex lifecycles are needed such as used by NASA or the Department of Defense for complex, and integrated systems that cost billions of dollars and have a wide range of stakeholders. In any case, it is imperative to maintain effective records of events and decisions made, of plans in the form of requirements, schedules, integration, agreements, and risks so that as stakeholders come and go on projects, continuity is maintained, and the newly arrived people don't mistakenly take a wrong turn because they didn't know about a previous decision. This can be costly and may kill a project or system. In this next section, I will introduce the next level down – below a lifecycle. Processes are the detail needed for people to follow that make it possible to deliver an effective system.

3.0. Processes

In Section 0 Milestone Zero – Before the Beginning, I spoke of the genesis of System Engineering, how to set up effective organizations, leaders, the roles of managers, administration and project coordinators, and effective communication. Without these attributes in place, the organization will not succeed. Section 1 described that WHAT, WHO, WHY, and the paradigms of Cost, Schedule, Performance, Risk and People, Process and Tools (the 30,000-foot level) – all of which offer a quick assessment on how well system and project is is managed. Section 2 provided a discussion Lifecycles (5,000-foot level) – pick the most suitable and use it, or a combination. Section 3 is on Processes, and this is the bread and butter for System Engineering. Processes are at the 'Ground Level'. Without a uniform set of processes, it will be hard to succeed.

At this point I'd like to recognize the **International Council on Systems Engineering (INCOSE)** and the contents of the *Systems Engineering Handbook*, much of which I have elaborated or provided real life experiences. In this section I discuss processes that INCOSE has created, but I give more specific formats and frameworks for the reader to apply.

When I look at a project or a system in which I plan or evaluate the system and project status, I look for these processes:

- Manage Project
- Manage Risk
- Configuration Management/Document Management (CM/DM)
- Manage Requirements
- Verification (VER)
- Validation (VAL)

If these processes do not exist, it is likely there are problems on the project and the system. In this section I will describe the purpose of processes, provide a template and provide a generic example of each of the processes listed above. Additionally, I will also provide examples for:

- Architecting
- Technical Exchange Meeting (TEM)
- Managing Defects
- Integration
- Manage Schedule
- Conduct Trade Study
- Project Management Review (PMR)
- Transitions
- Agreements
- Knowledge Management (KM)

Your organization will require processes like these but will be different in detail or even name.

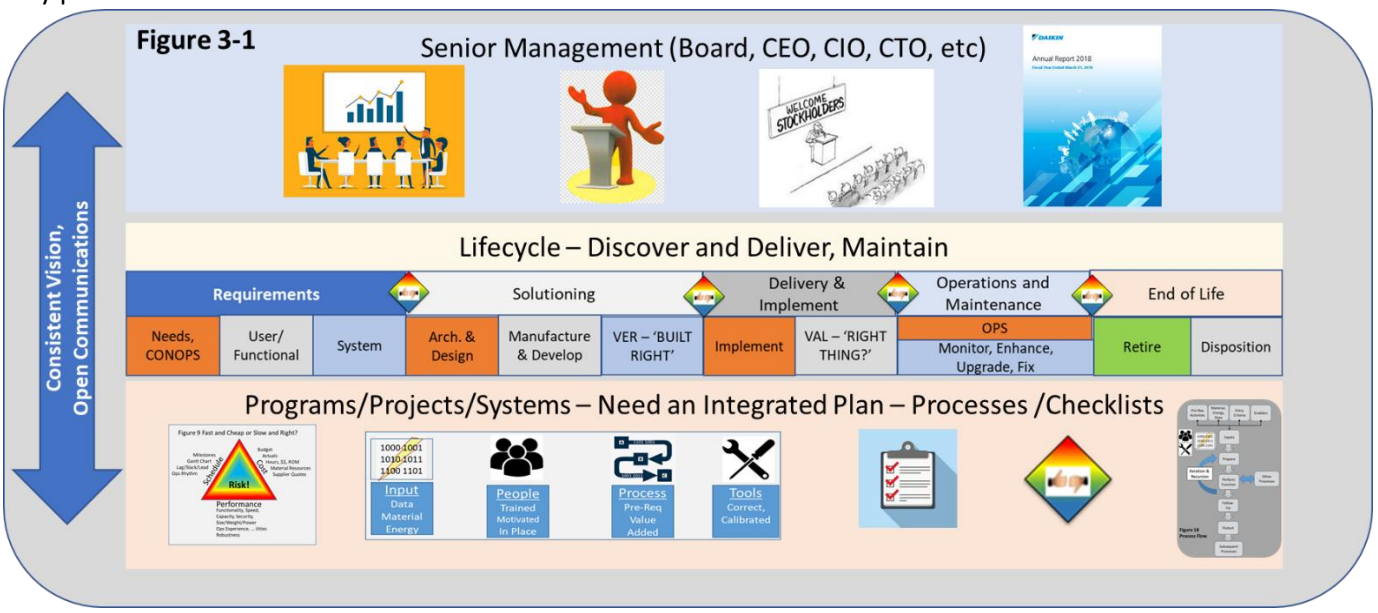
3.1. Why Process

We are creatures of habit. We tend to perform with more accuracy and efficiency when using a consistent process. Think of your process when preparing for your day – shower, eat breakfast, kids off to school, pack lunch and out the door. These activities have become well synchronized with other household members to prevent conflict, overlap and inefficiencies. But mix this up your routine one day and its likely you'll miss something like your lunch, be late for work, or you'll have a disgruntled family member. Why not do the same for your work? A process that is synchronized to other processes and activities will help ensure the recipients of your work will receive what they need and when the need it – and vice versa. By applying a regular process, roles and responsibilities, the needed skills and tools are better planned. Gaps and overlaps in task, and efficiencies are discovered and corrected. And understanding the process will allow the Systems Engineer to not perform a process if not needed or to

apply one when it is needed. Processes creation and upkeep is a collaborative effort, so that the stakeholders involved document and approve their activities. Documented processes should go into CM/DM and baselined so only approved changes are made. Additionally, I tend to not print processes but refer to them online to ensure that each process is current and accurate, plus to allow for quick navigation by using hyperlinks.

Most documented processes are the culmination of years of experience by many subject matter experts and therefore, processes have capital value. They contain 'tricks of the trade' and offer a body of knowledge that delivers value. Therefore, most corporate and industry processes are considered company proprietary and not easy to find on the internet. Processes include the flow down of the company's strategy and interests, the framework of how policy, regulatory and legal issues are performed at the day-to-day level. Processes can be (and often must be) certified for the International Standards Organization by an auditing organizing. When writing a process keep in mind the expertise of those performing the process, and the needed tools. Ensure the process is written at the adequate level with enough detail, but not too much. I have found that if one follows a process and the outcome is still flawed, the individual who followed the process is not liable – because they complied to the process.

Figure 3-1 shows the alignment to the cooperate interests to the day-to-day activities as reflected in the processes, and perhaps as detailed in a checklist. At the corporate level, executive manages are interested in delivering to the stockholders, creating the strategic business plan, and ensuring compliance to laws and regulation. This is transitioned down to the divisions and programs that reflect the interests of the corporate level through a lifecycle and other supported documents. The day to day work on the systems and programs are in the form of the processes, the project plans, architectures and testing activities, and the operating instructions. Without the alignment of these three (sometimes four or five) levels of the organizing, the team will never gain the needed focus to optimize success. That is why processes are needed.



A good set of processes apply to multiple branches of the organization. We often think of processes as mainly applying to engineering and Project Management, but most processes have outputs that are inputs to supply management, security, contracts or finance, and often to other contractors as dictated

by mutual agreements. In turn, another branch of the organization is depending on the inputs from other branches, and sometimes other contractors. Once these processes are mature, people come to know them quite well – and need not continually read how to perform their job. However, processes must not collect dust – all employees must periodically review to ensure they understand and to submit changes to improve or update.

3.1.1. Decision Gate Review Process

Milestones are markers or signposts indicating progress, and points of completion and achievement. These checkpoints are used on all projects but will vary in formality based on system and project risk. For large government systems and programs, Milestone Review Authorities are assigned at the executive or flag level, but this Book will not apply the Milestone Review Authority. **Checkpoints** are indicators of when a task or artifact is due based on a schedule. For the purpose of this book, I will use the term **Gate Approval** (Gate for short), which is when the management team agrees to continue or not continue the project.

The Project or Systems Management Team is composed of pre-designated experts and managers as outlined in the Organization Chart the SEMP or the PMP. These exports and managers convene at critical points in the project and assess the project and system readiness, or if the system is still needed, and grant (or not grant) approval to continue into the next portion of the lifecycle. 'Whoever's available' to attend a Gate Review leads to managers not aware or knowledgeable of the system or project status, groupthink, decisions made under peer pressure and incomplete information. While the Gate Review will have a pre-established criterion, executive organizational management reserves the right to override the decision. Each Gate Review should be named according to the nature of the review, and to allow clear communication of its purpose and the result.

The amount and scheduling of Gate is dependent on the risk of the project and each phase, the amount of resources you expect to consume in the next phase, the duration of the phase, and the maturity of the team. Systems that may pose great risk or complexity should have more Gate Reviews, where simple and lower risk system may need few. **Table 3-1** offers only a high-level understanding of the type of gates you should select on your project, and to what detail you should consider.

For example, track along the row called Functionality. At Gate 1 Project Start the functional need is defined, along with topics related to security, and if the function is permitted by law. The Systems Engineer and their team examine Lessons Learned and Risks of other, similar projects so these lessons can be applied, and the risks are preempted for your new system or project. At Gate 2 Preliminary Design Review (PDR) (in the Functionality Row) the questions are related to conceptional design; can it meet the requirements and the 'ilities', will the system perform to capacity and speed? Later at Gate 3 Critical Design Review (CDR) the drawings, tracing to requirements, results of any prototyping, TRL are reviewed. Is the system still needed and will the customer see it through? At Gate 4 Fielding, the results of Verification (VER) & Validation (VAL) (or V&V) are reviewed and assessed, asking if the system is compliant, was the right system built, and did we build the right thing? You will see the same progressive detail for all attributes of a project and for whatever Gate Reviews are established for your project and system. Gate 5 Retire asked if the replacing system has taken the functionality, if there are environmental or security issues with packing and shipping or disposing of the system. Any one of the rows can be chosen and the interest related to that topic can be traced from the start to the end.

Table 3-1 Gate Reviews and Their Focus - Progressively Detailed Through Each Gate					
Approval Gate	**Gate 1 Project Start**	**Gate 2 PDR Pre-Design Review**	**Gate 3 CDR Critical Design Rev.**	**Gate 4 Fielding Review**	**Gate 5 Retire**
Scope	Consistent to Charter	Trade Studies Experience Outsourcing	Agreements Made, MOA	Agreements Ready	Dependent Users
Function-ality	Defined Need, Regulatory, Data Sensitivity	Deliver Need, Compliance, Data, … ilities, Performance, KPIs,	Requirement, … ilities, Compliance, Approval, Drawings as Evidence	User Exp, Need, Compliance, Resolved, Interfaces	Connectivity, Environmental, Security
Cost	Notional Budget, Availability of Resources	Phased Budget, Availability of Resources, Licenses	Est. at Complete/Est. to Completion (EAC/ETC) Prepared Committed Resources,	Travel, Export Issues, Shipping, Transition Plan	Packing and Shipping, Destruction
Schedule	Milestones	Gantt Chart, Development Method, Draft Project Plan/SE Plan	Detailed Gantt, Development Method, Project Plan/SE Plan	Delivery Schedule, Transition Plan	Transfer or Decommission vs 'pulling the plug'
Process	Known Processes	Processes Identified and Documented, in Use, Checklists	Process Results Identified, Completed Checklists	Mature, re-used processes, not the first time this is done, checklists	Retention of needed documentation – as evidence of compliance
Communication	Listening to Stakeholders/ Users, Who needs to Know What, Establish Project Comms.	Baseline designated artifacts in a known location, Informing and listening to Team and Stakeholders, 'Grease Communication', Outline Education and Training, Procedures	Baseline artifacts, control changes, informing and listening, 'Greasing', Prepare and Conduct Training, Write Procedures	Notify external stakeholder and dependencies, daily and frequent updates, Inform of Changers/ Awareness, Train and Educate	Capture old documentation, inform stakeholders and dependent users
Procurement	Supplier Feasibility, Non-Disclosure Agreement Alternative	Most suited – cheap and slow, or expensive and fast, trusting provider, ISO, sub-contracts,	Back Up Procurement if needed, open comms with existing sub-contractor, SLA, ISO Audits	Travel, availability 24/7, requirements flow down, communication	Not becoming complacent, Complete and Close Contracts and Invoicing, terminate licenses, close agreements
Demand of System	Is there a need? Others providers	Still a need? Viable substitute	Capacity to meet demand, cost recovery (profit). Checklists	Check each user site for demand changes. Adjust performance as needed. Follow Up entering O&M.	Verify demand is transferred seamlessly, inform users and other SE.
External Issues	Political, Environ-mental, Availability of Material, Bandwidth, utilities, survey external threats	Plan for external Issues, high level identification of issues, utilities redundancy (BU Power), slack, margin, insurance	Detail of issue planning, include into Risk Plan, observe external condition	External networks or data available, last check before fielding. Travel, Import/ Export or data restrictions	Treaties and Alliances for data, service. Tacit agreements.

Any meeting (Decision Gate, Reviews, Working Group) must have an agenda, a list of items to review, a purpose, decisions to make, minutes and action items, risks and mitigation, attendees, and the venue. Source documents are provided (and previewed) in advance of the meeting by those attending. Key documentation must be preserved in a controlled location (SharePoint, GitHub, Shared Drive, etc) for later reference and to maintain the common understanding of key events. If not documented, then the event or task 'did not happen'. While maintaining and controlling these documents may be regarded as an administrative activity, the Systems Engineer cannot allow an administrative action to jeopardize the system development and later operations, not to mention wasted resources.

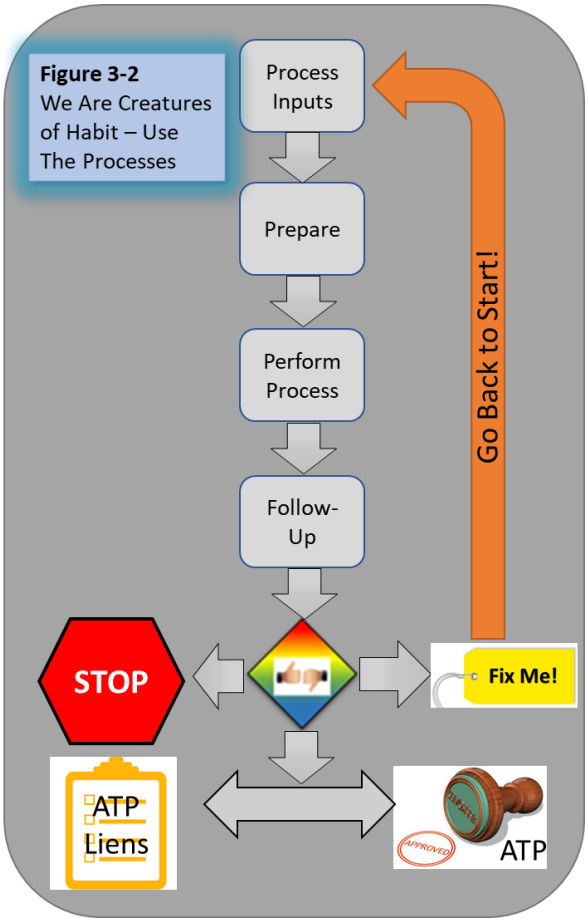


Figure 3-2 We Are Creatures of Habit – Use The Processes

Figure 3-2 shows the generic process for conducting a Decision Gate. The process will have inputs such as artifacts, proceeding processes, or other Gate Reviews. Once the Inputs are completed you may prepare by reserving a room, inviting the right people, ensuring any presentations are ready and meeting room equipment works. Documentation to preview should be (must be) sent out or provided to the decision makers in advance so they have time to read and ask questions, and to get answers. 'Greasing It' means to informally solicit the desired result in advance to make the process goes more quickly, and to reduce surprises at the actual Gate Review. Next conduct the meeting. Ensure all follow up actions are complete such as minutes and action items documented and sent out. The Gate Review will have only one of four possible outcomes, none of which are 'bad' unless done for the wrong reason. The four alternative outcomes of a Gate are;

1. **Approval to Proceed (ATP)** – proceed to the next set of work, with no carry over work (or liens).
2. **ATP with 'liens' or To Be Determined (TBD) or To Be Resolved (TBR)** – proceed to the next set of work, but complete the documented actions that may not have been completed in the previous phase
3. **Return** to the previous phase and perform **incomplete or defective work**. This is illustrated by the yellow tag.
4. **Cancel (Stop) the program** – the risk is too great, system is no longer needed, there are better alternatives, or the business case is no longer viable.

Besides the Gates used as an example in **Table 3-1,** other important decision gates include:

- Initiate Project – approval to start a project, given budget, need, milestones
- Requirements Review – periodic and careful review of requirements, traceability, need
- Preliminary Design Review (PDR) – conception design acceptable, alternatives
- Critical Design Review (CDR) – based on PDR – detail, meet requirements, performance, ready to 'light the fuse'. This is called **Critical** for a reason

- Test Readiness Review (TRR) – prepared to conduct VER or VAL – test plans, test cases, VER and VAL Process ready
- Operational Readiness Review (ORR) – given VAL results, assess if system ready to go to ops – and how will this be done?

The worst outcome is to grant ATP for a faulty system that later, problems are discovered after resources were wasted or the system has had a catastrophic failure.

3.1.2. A Process - Use a Common Game Plan

Processes are one third of the project and systems activities (think people, **process,** and tools). It draws in the resources to complete activities. A bit more on processes is discussed next.

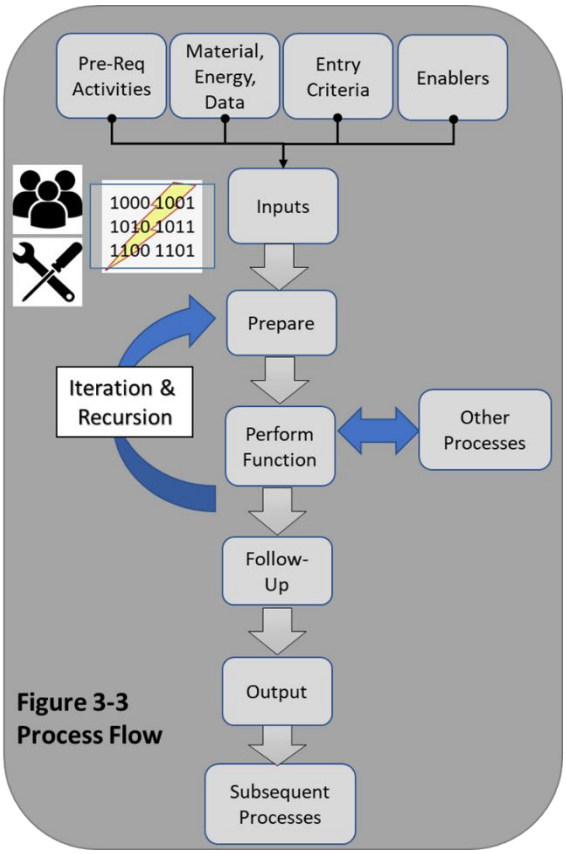


Figure 3-3 Process Flow

Figure 3-3 illustrates a generic process flow. The illustration does not show all steps described below but it captures the main points. Your organization may have more or less detail to your processes depending on the needs of the organization and the systems you are building.

- Inputs – the proceeding set of activities or resources that are needed before a process may begin. Inputs may be resources, data, tools, or control. One process input is a proceeding process output.
- Enablers – the means to which a process is performed.
- Control – governs the accomplishment of a process.
- Entry Criteria – the work that must be completed and the conditions that must be provided before a process may begin.
- People – those performing the process. It is assumed that those performing the process are trained and qualified.
- Tools – workbench, software (and data), measuring tools, hand tools, lab, etc., needed for the process to perform.
- The Process Function – the series of work activities employing resources, performed by people, to achieve a result in the form of an Output.
- Steps – most simple and lowest form of work performed. Checklists may be used to ensure critical steps are completed. Steps are the HOW processes are performed.
- Other Process – may be embedded into the main process.
- Output – the succeeding set of activities or resources that are constructed or created as a result of the completed process. Outputs of a process are the Inputs of another Process and must be completed based on the Entry Criteria of the succeeding process.
- Exit criteria – the work that has been completed, to the defined criteria, specification or new behavior or condition. Only when the exit criteria are met is the process complete.

Measured processes allow for the discovery of an inefficient step (something not tested correctly, wire not connected tightly, or eliminating an unneeded step, etc.) that can be quickly identified and improved. Processes may be tailored in a planned fashion and according to what is suitable to the needs of the system and to the project. The System Engineer must not react to last minute, unthought shortcuts. A process is also successful if the right tools are available. Falling back on People, Process, and Tools Paradigm – the three work together for a successful result. If any fail, then you are risking project or system failure.

3.1.3. Process Template

Take a few minutes to review this template so that when you review the examples, you'll more quickly understand the examples.

<table>
<tr><td>Name: Process Name – Make it Active/Verbal</td><td>Purpose: Why are You Performing This Process</td></tr>
<tr><td>Inputs and Prerequisites: Items, Processes, Activities, Documents Used in the Processes, or as Inputs, Data Sets, RF Feed</td><td>Information: General Tips / Information, special safety, OSHA Can embed an illustrated process flow. Process Templates May have a Block Flow Diagram</td></tr>
<tr><td>Related Processes/Tasks: Proceeding, Sub-sequent, Parallel and embedded activities and processes</td><td>Tools: Software, Hardware, Facilities, Lab, Testing & Measuring Equipment  </td></tr>
<tr><td colspan="2">Roles & Responsibilities 
• Project Manager – will always be responsible for the success of the project
• Systems Engineer – Compliance/ Participation - only needed processes performed
• Specialist - Requirements Manager, CM/DM Manager, Subject Matter Experts (SME), etc.
• Quality Manager – ensure people, process, tools performing, delivering to expectation.
• Project Coordinator – assist PM in maintaining the schedule, budget, project documents.</td></tr>
<tr><td colspan="2">Process Activities
1. General Sequence of Activities – starting with plan and process, include training and awareness so team know what is being done, their roles
2. Describe Each Step – briefly – it is assumed those performing the task are trained to the level needed to perform the tasks.
3. Document What was done, any success, failures, suggested improvements – so others will know esp. those performing subsequent processes.
4. Finish – Close files, put away tools, clean up, report any issues, etc.</td></tr>
<tr><td>Output and Results: Expected Results, Artifacts – if not completed the process may not be finished. Some outputs depend on the degree the process was performed. Not every artifact may be needed.</td><td>Change Log
(printed copies may not be current)<table><tr><th>Ver</th><th>Date</th><th>By</th></tr><tr><td>.1</td><td>04/01/20</td><td>BLS</td></tr><tr><td></td><td></td><td></td></tr></table></td></tr>
</table>

And keep in mind that even the examples are quite general and can apply to differing types of systems, and industries. A process may also contain a simple block flow diagram, like what is illustrated in **Figure 3-2 or Figure 3-3.** A change log must be included for each documented process, with the recommendation that the printed version is subject to uncontrolled changes and may not be current. The use of hyperlinks enables quick navigation of the set of documented processes. Make processes

visually appealing with subtle colors or modest artwork – in this case I included icons for People and Tools.

3.1.4. Manage Project

The Project Manager is responsible for the success or failure of the project. That's why they are the Project Manager. As such, they estimate the type and the amount of resources needed for the project, they ensure effective communication among the project team and stakeholders, they apply metrics and make adjustments to the project, and they manage the associated risks that occur to the project.

The Project Manager performs the project activity, but the Systems Engineer must work closely with the Project Manager to ensure deliverables are on time, costs are collected, schedules are met and risks are identified. Others who assist in this process include schedulers, finance personnel, contracts and subcontracts, Project Controller and Quality.

Typical tools and methods include the use of a project Gantt Chart (for scheduling) with Gate Reviews and phases, tasks, including dependencies slack and lag time, and the organization group assigned to each major task. An operational rhythm must be established and documented that includes regular events by week or by month. Project resources are managed (often by a spreadsheet) of a budget to actuals (Estimate at Completion (EAC) and Estimate to Completion (ETC)) which are the amount to be spent on the project, and the amount needed for the remainder of the project (in that order) other processes used in conjunction to **Manage Project** are **Manage Schedule**, **Project Management Reviews**, **Configuration Management/Document Management**, and of course **Risk/Issue/Opportunity Management**. But frankly the Project Manager will have a whole plethora of processes in their tool kit to draw. Closing a project related to systems will include the technical relocation of interfaces, ensuring all deliverables are delivered and include final invoicing, customer acceptance documents, closing contracts, charge numbers, and taking care of people by helping then find new jobs.

Managing a project almost requires clairvoyance to predict events and taking action to avoid, or to capture an opportunity (see Risk, Issue and Opportunity Management). Managing Projects include having a budget or schedule reserve 'in your hip pocket' to account for pop up risks, and to stage engineers or technical specialist in case quickly needed. This involves the ability to look outward but focus inward to the project. Large programs will likely have a separate process to 1) Start a Project 2) Manage a Project 3) Closing a Project. Finally, the Project Manager must always draw on the paradigm of Cost, Schedule, Performance and Risk.

Here's a good time to talk about **Work Breakdown Schedules (WBS)** – which is a decomposition of work to perform on the project. Each activity is called out by an indexed number, broken down into sub-numbers to the point that each task and sub-task is understood and measurable, and assignable to a workgroup. The work to perform at this level is often described in a **Work Package** (a more detailed description of the work, who will perform, requirements etc). It is helpful to use a verb form of the work to ensure its understood. A separate row can be used to document the team to perform the work and the forecasted resources. The WBS must come under peer review and acceptance, and it will be used to assign charge numbers, manage resources. A WBS is instrumental in creating a schedule, by placing the task in a logical sequence, with start, stop dates, and linked to other dependent tasks. The WBS is critical is estimating the needed resources by first going top down (decompose) the bottom up

(estimating resources). You'll see the WBS is called out throughout the processes. Other methods such as agile use similar decomposition but not necessarily in a WBS format.

<table>
<tr><td>Name: Manage Project</td><td>Purpose: Plan, Organize, Control, Direct Resources to achieve the expected outcome</td></tr>
<tr><td>Inputs and Prerequisites: Approved Project</td><td>Information: Ongoing Process – start to end of project, many embedded activities; PPT: Cost/Schedule/ Performance & Risk, WBS</td></tr>
<tr><td>Related Processes/Tasks: Manage Risk, VER & VAL, Manage Schedule, Manage Cost, Create Architecture</td><td>Tools: MS Word, Excel, Tableau, Project Now!  </td></tr>
<tr><td colspan="2">Roles & Responsibilities 
• Project Manager – success of the project – prepare and execute project plan, build team, awareness and training, deliverables
• Systems Engineer: Establish & Comply to SE Management Plan, CM/DM, Risk Management, Verify Enterprise Architecture, Solutioning, Quality, Continual Improvement
• Quality Manager – ensure people, process, tools performing and delivering to the expected standard.</td></tr>
<tr><td colspan="2">Process Activities
1. Plan Project – PM creates project plan – include Roles & Responsibilities, Scope, Assumptions, Schedule/Ops Rhythm, Budget, Deliverables. Open and assign charge numbers, establish, and enforce agreements with contractors and providers.
2. Technical Activities – Systems Engineer – creates and approves SE Management Plan, other Systems Plans, sets up and manages systems tool (CM/DM and Requirements, Architecture Tools), build team
3. Ops Rhythm – establish an ongoing patter of meetings/activities – use checklists, action items, PMR, TEMs to monitor and adjust resources to ensure quality and completion
4. Record Activities – after an activity or process is completed, record in minutes, reports, test results, and place into CM/DM as proof of compliance.
5. Communicate and Awareness – vertical and horizontal - formal – via staff meeting, TEMS, PMRs, few emails. Informal by face to face 'walk around'. Create and deliver formal training for major changes or new tools.
6. Close of Finished Project – phase down activities, reward/help team find new jobs, transition needed activities/interfaces, ensure all deliverables delivered, customer feedback obtained, close contracts, KM, close charge numbers</td></tr>
<tr><td>Output and Results: Managed tasks, completed tasks accepted by recipient, functioning and tested system</td><td>Change Log
(printed copies may not be current)<table><tr><th>Ver</th><th>Date</th><th>By</th></tr><tr><td>.1</td><td>04/01/20</td><td>BLS</td></tr><tr><td></td><td></td><td></td></tr></table></td></tr>
</table>

3.1.5. Manage Schedule

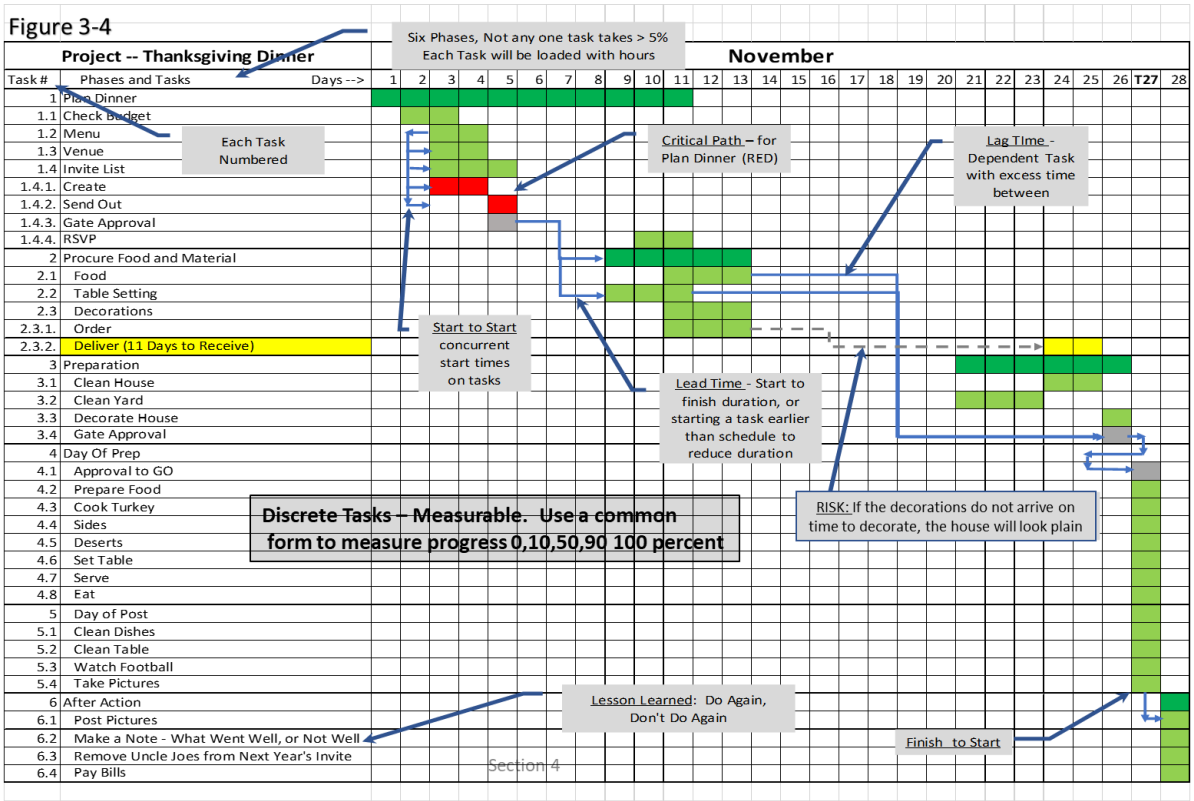
A solid and well understood schedule is critical to project and systems success. A schedule is both a problem-solving tool and a communication tool to the team, so team members know their role, and when to perform their tasks. At project inception the schedule will likely only include milestones (eventually will be called gates). Before starting the project, the schedule will become a more detailed document of tasks, and processes and expected duration, dependencies, lag and lead time. Without a schedule the project is in great risk of failure. A typical project schedule artifact is the Gantt Chart as illustrated in **Figure 3-4**, which includes a sample below to which I will reference. Those assisting the Project Manager are the Systems Engineer of course, the Scheduler, Risk Manager, those responsible for the deliverables, and Quality.

<table>
<tr><td>Name: Manage Schedule</td><td>Purpose: Establish and maintain a manageable project schedule.</td></tr>
<tr><td>Inputs and Prerequisites: Approved Project, Project Plan, WBS, Deliverables, Roles /Responsibilities, SOW</td><td>Information: Schedule may be Gantt Charts, Ops Rhythm, or Increments (for Agile).</td></tr>
<tr><td>Related Processes/Tasks: Manage Project, PMR</td><td>Tools: MS Project, Primavera  </td></tr>
<tr><td colspan="2">Roles & Responsibilities 
• Project Manager – success of the project - on time, as expected and to cost
• Scheduler – create and update schedule based on system and project needs. Identify and help treat risks
• Systems Engineer: Ensure tasks/ deliverables in the project plan. Lead / organize differing tasks to ensure integration, identify risks.
• Quality Manager – ensure people, process, tools performing and delivering to the expected standard.</td></tr>
<tr><td colspan="2">Process Activities
1. Establish Schedule – Schedule from solicited project (from proposal) is updated and Re-baselined from the proposal.
2. Schedule Review Work to Perform – SOW, Work Breakdown Structure (WBS), List of Deliverables and with project/systems team determines sequence and dependencies, durations – drafts schedule
3. Lean Out Schedule – first run at schedule is incomplete – will take iteration and recursion to ‘streamline’ and ensure completion.
4. Baseline Schedule –once suitable and acceptable place in CM/DM but prepare to make controlled changes at next PM Meeting.
5. Update Schedule – make approved changes, show completion.</td></tr>
<tr><td>Output and Results: Baselined Schedule or Ops Rhythm, Updated Schedule, Completed Scheduled Tasks.</td><td>Change Log
(printed copies may not be current)<table><tr><th>Ver</th><th>Date</th><th>By</th></tr><tr><td>.1</td><td>04/01/20</td><td>BLS</td></tr><tr><td></td><td></td><td></td></tr></table></td></tr>
</table>

Common terms used for scheduling, and as referenced in **Figure 3-4**, are;

- Critical Path – a set of dependent tasks through the schedule in which a delay in any one of the tasks will cause a schedule delay. These critical path tasks should be subject to scrutiny so they do not pose a risk that would delay the delivery or increase the cost of the system. Upon discovering the critical path, Systems Engineers must be astute to the 'secondary' critical path which may emerge and cause an additional and likely not observed, risk.
- Lag Time – upon completion of the one scheduled task, the subsequent dependent task is delayed a set amount of time. This allows resources to be applied on other tasks, or for a task to be completed in synchronization with other tasks.
- Slack Time (Float Time) – the amount of time a scheduled activity may be delayed without affecting the schedule of any subsequent tasks.
- Lead Time – starting a task earlier then needed. Lead time is an opportunity to accelerate the scheduled completion of the project.
- Concurrent and Parallel tasks – two or more tasks may be performed to expedite the activities on the schedule. Essentially, the Systems Engineer is dividing the team so one part of the team can work on a set of tasks, and the other work on a second set of tasks. This is common and needed so not every task is done in serial (finish to start relationship), but the risk is to join the results of the task together at the finish. Periodic progress checks between the tasks will help reduce this risk. The concurrent taskings may occur on virtually all portions of the lifecycle and associated processes.

Figure 3-4



Event Driven – a project or a series of tasks in which the schedule is controlled by an outside entity, as in a contingency plan or emergency plan. Once the event occurs, it kicks off a set of project or system activities. Event driven is not intuitive to how we think - we must control activities but at times, we are at the whims of externalities and natural phenomena.

Each scheduled task must have an 'owner' who will ensure the assigned task is being worked and is complete. The System Engineer with the Project Manager should have a consistent way to measure task completion. A common criterion I use is:

- 0 percent – The task owner has accepted the task but has not yet read, studied or planned how to complete the task.
- 10 percent – The task owner has read, understood and has developed a plan to complete the task.
- 50 percent – the task owner is in the process of completing the task.
- 90 percent – the task owner has completed the task, but the task output has not yet been accepted by the recipient.
- 100 percent – the recipient of the task has received and accepted the output of the task.

Using this method (or whatever method you choose) and ensuring all use the same method will offer higher confidence that the stated progress is accurate. It's too easy for someone to say 'I'm finished!', but it's not.

Any one task should not exceed ten percent of the total project. In fact, any one task should probably not exceed five percent. The reason is that if the task volume (schedule and resources) are too great, the Systems Engineer and Project Manager will lose insight to how complete the project really is because of the potential great variance of that one, single task.

A project schedule (i.e. Gantt Chart) can be messy because these charts are working documents. If printed and posted to a wall there should be a healthy dose of pen marks and notes with lines and arrows, and stains (blood, sweat and tears). It's a thinking document and it models the future schedule of synchronized activities, causing much coordination and thought. But once the schedule is updated and agreed, it is baselined and goes into document control. Only coordinated and approved changes may be made after it is baselined.

To Gantt or Not Gantt

I learned the hard way that Gantt Charts work well in solving schedule problems for a team. But I do not recommend showing the entire Gantt Chart at a PMR or executive-level manager meeting. Although you may be proud of your team's hard work, too often the executive managers, who infrequently see such detail, tend to dig down and ask too many questions, try to 'fix it' and second guess, and time is taken away from other important topics.

3.1.6. Risk, Issue and Opportunity Management

Risk is an event that, should it occur, will cause damage to the project or system. Risks are categorized according to their **likelihood** of occurring and the **impact** of the event if it occurs.

Every member of the project and systems team has a role in risk management, to include:

- Systems Engineer – Working with the Project Manager and Systems Team, the Systems Engineer will create a system design and solution that is of acceptable risk, and balances the Cost, Schedule, Performance Paradigm.
- Project Manager – Risk will always be on the mind of the Project Manager as if to continually ask 'what can go wrong' Project Managers must have the ability to look past the visible signposts and indicators to events that have not yet unfolded. Besides this, the Project Manager ensures the team

is trained and following the processes. The Project Manager will assign the Project Risk Manager and the Risk Owner for each risk.

- Enterprise Risk Manager – the person who manages all risk from various programs, and ensure they are captured, documented, and resolved. There will be a lesson learned, and lesson applied into the organizational processes to pre-empt the same risks from occurring elsewhere.
- Project Risk Manager - creates a project risk plan, manages the risk process to include vetting real risks from those which are nebulous, simply a 'Go Do' or something that is not difficult.
- Contracts and Sub-Contracts – to outsource risks for risk transfer, and to ensure risks are known by the sub-contractors, and to the customer.
- Risk Owner – person assigned to create and manage a technical plan to reduce the risk leading to closure.
- All Project/Systems Team Members – responsible for reporting risks that they may observe.

Name: Manage Risk	**Purpose:** Capture / Manage Risk, preventing it from becoming an Issue
Inputs and Prerequisites: Risk Plan, Risk Repository, Trained Team, ICD 503, ISO/IEC 27000 Information Security Management	**Information:** Risk is an event, should it occur, will cause damage to the project or the system.
Related Processes/Tasks: Start Project, KM, Manage Project, Manage Requirements	**Tools:** Spreadsheet, PowerPoint 

Roles & Responsibilities

- **Project Manager** – success of the project, ensure processes are followed, balancing cost, schedule, performance and risk.
- **Systems Engineer -** create a design and solution of acceptable risk, delivering capability
- **Risk Manager -** capture, track and manage risk, compliance to the process
- **Enterprise Risk Manager:** Document Risk and Mitigation in repository and make this knowledge available to those in the organization.
- **Risk Owner -** Risk is assigned to someone to manage to lower the Likelihood / Impact.
- **Quality Manager –** ensure people, process, tools performing and delivering to the expected standard

Process Activities

1. **Identify Risks** – all stakeholders can identify risk, and report to the SE, PM or Risk Manager
2. **Document Risks** – Risk Manager vet risk (if accepted) into a common risk ledger and chart. PM assigns risk owner.
3. **Treatment** – Risk Owner create and perform treatment plan.
3. **Risk Monitoring** – Risk Manager provides oversight of the process, ensure compliance.
4. **Risk Closure -** Execute plan to reduce impact /likelihood of risk – to closure. Complete risk documentation.

Output and Results: Documented Risk, Risk Treatment Plan, Triggers, Closed Risk, Identified Issues

Change Log
(printed copies may not be current)

Ver	Date	By
.1	04/01/20	BLS

3.1.6.1. Risk Management – One Step at a Time

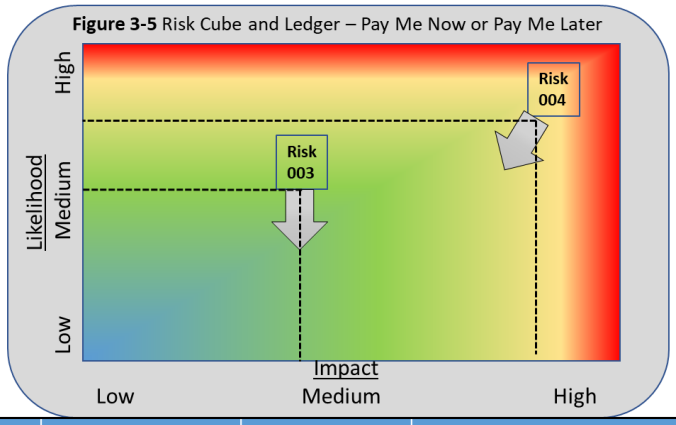
Often risk management is not well planned nor executed on a system or project simply because it is not considered important, or it is not well understood. Risk statements should be discrete and manageable, and not nebulous that only a few understand. A risk is not something that is 'just hard to do' – there is a method to capturing and quantifying or qualifying the risk so it can be measured and managed. Three simple risk statements will be:

- If Data-R-Us Company cannot hire four more data engineers, we will not be able to ingest the Big Medical Data Repository by the scheduled date.
- If the Cost of Silver reaches 75 cents per gram, the cost of the circuit boards will exceed our budget.
- If the Miami Fl does not implement flood mitigation, then a Category 4 or 5 Hurricane will flood the city.

3.1.6.2. Risk Documentation – an Example

Figure 3-5 illustrates a Risk Cube and a Risk Ledger, such as the fictional risk shown for Peter Drucker. The X Axis of the cube is the weighted value of **likelihood** of the risk occurring – between Low to Medium to High. The **impact** of the risk, should it occur, is expressed in a weighted value along the Y Axis – Low to Medium to High. These ratings are usually subjective and can be hard to quantify so carefully assess the level of risk without embellishing, or without underestimating. The intersection of the likelihood and impact are plotted on the cube, and those risks in the yellow or red zone will be tracked and managed to work them down into Low impact and Low likelihood of occurring, and the closure. Drucker and Warren Buffet's risks are plotted as examples. Each risk will be documented in the ledger with the number, risk name, risk statement, owner (who will treat the risk), action taken to treat, and may also include 'triggers'.

Figure 3-5 Risk Cube and Ledger – Pay Me Now or Pay Me Later



Risk	Name	Owner	Mitigation
003		Peter Drucker	
If … Then Risk Statement will go here			
004		Warren Buffett	
If …. Then risk statement will go here			

Risk impact and/or likelihood may be High, Medium, or Low as follows:

- High – will have mission critical impact by severely impacting or halting the mission. Immediate action is needed to prevent the risk from becoming an issue. Subject risk is likely to occur and become an Issue.
- Medium – Will hamper or degrade mission performance and may temporarily stop mission performance, increase cost and cause delay in deployment. It can usually be resolved through normal Systems Engineering activities.
- Low – an inconvenience, will not stop mission, cause moderate delay or cost increase.

3.1.6.3. Risk Treatment

There are various methods to treat risk, and organization and some industry Risk Management Best Practices may have their own set of terms (i.e., treatment, reduction vs manage) to which stakeholders should adopt.

1) Transferring risk to a third party who may be an expert in the topic at risk
2) Mitigate by creating a plan to decrease the impact of the risk, should it occur.
3) Avoiding any changes to the project plan by extending the schedule or adding budget
4) Accepting if the risk impact and likelihood are low

Common activities to reduce risk are:

- Management Reserve (MR) for budget and schedule – as previously stated
- Prototyping – determining if a desired capability is achievable before committing resources
- Verification and Validation (V&V) – confirming that a system will function and perform as expected before progressing to the next stage of the project.
- Unit Testing – confirming software modules perform, as the system is developed.
- Component Testing – verifying that a sample of components perform as expected before integration into the system. First Unit test will test only the first unit of a batch. For critical no fail systems, every component will be tested.
- Outsourcing – Assign the risk to an expert (internal or external contractor) who are most qualified to mitigate the risk.
- Back Up or Contingency Contract – if the first contractor or vendor is not performing, acquire a backup contractor. This will be expensive and should be considered when performance and schedule are paramount. A Quick Reaction task order may also be used.
- Similar Systems – the Systems Engineer should evaluate risks and how managed in similar environments and apply those mitigations to their system.
- Architectural Designs – may be created to reduce risk, such as an open architecture or pre-staged design. See Section 3.1.10 Create Architecture.
- Back Up Plans – example: migrating a spare data set or establishing alternative network connections.
- Continuity of Operations (COOP) – a backup site ready to assume operations if the primary site or system should fail or be compromised. **Warm backups** require little preparation time to assume operations, and **hot backups** essentially run in parallel and can immediately take control.
- Pre-staging Equipment or Spares – In case the primary fails, especially for equipment that is prone to failure.
- Work 'at Risk' – perform work, procure material before payment is arranged or approved. Working 'at Risk' will allow progress on schedule (usually critical path) before formally approved (usually at a Decision Gate) but will likely still require approval of senior or executive management. The level of approval will depend on the managements level of approving authority. Potentially, the organization may procure material or perform work that they may not get paid, but are willing to take the risk.
- Trigger – an event established in the risk handing plan that should that event occur, it will start actions to handle the risk – such as hire a third party, extend the project, acquire added resources, conduct an alternative (likely more costly) task.

Top Ten Quotes on Risks – Collected by Richard Branson (Branson, 2016)

10. "The biggest risk is not taking any risk … In a world that's changing really quickly the only strategy that is guaranteed to fail is not taking risks." – Mark Zuckerberg
9. "If you don't play you can't win." – Judith McNaught
8. "Life is inherently risky. There is only one big risk you should avoid at all costs, and that is the risk of doing nothing." – Denis Waitley
7. "Why not go out on a limb? Isn't that where the fruit is?" – Frank Scully
6. "A ship in harbor is safe, but that is not what ships are built for." – William G.T. Shedd
5. "When you take risks you learn that there will be times when you succeed and there will be times when you fail, and both are equally important." – Ellen DeGeneres
4. "There is freedom waiting for you,
   On the breezes of the sky,
   And you ask "What if I fall?'
   Oh, but my darling,
   What if you fly?"
   -- Erin Hanson
3. "I am always doing that which I cannot do, in order that I may learn how to do it." – Pablo Picasso
2. "Don't be afraid to take a big step. You can't cross a chasm in two small jumps." – David Lloyd George
1. "Two roads diverged in a wood … I took the one less travelled by, and that has made all the difference" - Robert Frost

https://www.virgin.com/richard-branson/my-top-10-quotes-risk

3.1.6.4. Issues and Opportunities

An **issue** is a risk that has been realized. Issues imminently threaten a project, system or a mission/business venture, or that damage has already occurred, and will continue to occur until the situation is resolved. In most cases the damage is not recoverable, such as lost aircraft or propriety information obtained by a malicious third party.

Opportunities are unplanned (often fleeting) events that, if captured and exploited, will decrease cost and schedule, or enhance capabilities to a system. Opportunities often pose risk, and should the opportunity not be correctly captured, may lead to an issue. An opportunity must be documented, an owner assigned, and a plan established to take advantage of the opportunity.

3.1.7. Requirements Management

Requirements start with a need to complete a mission or a business venture before it is known how the capability will be satisfied. The user will define WHAT needs to be completed or the desired end state in these three, and progressively detailed forms:

- **Statement of Need (or Mission Need)** – a simple direct problem statement describing what condition or state is needed to satisfy the function. It may be to procure supplies, to transport data, or to make sick people well. Statements of Needs are often strategic documents that will take years to deliver and will encompass numerous systems and programs.
- **Concept of Operations (CONOPS)** – presented almost in a story format with drawings, the CONOPS describes what the solution will do. The writers of CONOPS are often tempted to explain HOW the

system shall perform its function, but by doing so they may unnecessarily preclude innovative solutions. The CONOPS will help the Operator and the Systems Engineer discover Functional Requirements as they tell the operational story of the needed system. CONOPS often include drawings to help explain WHAT the system will do. Department of Defense Architectural Framework (DoDAF) provides a more structured method to capture the CONOPS. Storyboarding is synonymous to CONOPS except Storyboarding is done visually and is a continual state of improved draft formats, until the story is understood. Often, CONOPs are not completed because once the

Name: Manage Requirements	**Purpose:** Analyze, Determine, Document WHAT is Needed and to what Specification
Inputs and Prerequisites: Bounded and Scoped Problem, New Mission or New Business Operation, Enterprise Architecture, Regulatory & Compliance, ICD 503, ISO/IEC 27000 Series - Security	**Information:** Trace Requirements from Source to Testing, to When and How Delivered
Related Processes/Tasks: Start Project, Systems Analysis, CM/DM, Manage Project, Manage Risk, TEM, Create Architecture, VER & VAL	**Tools:** DOORS 

Roles & Responsibilities 

- **Project Manager** – success of the project
- **Systems Engineer:** create a design and solution of acceptable risk – balances Cost, Scheduler,
- **Performance** – to deliver the desired capability
- **Operations/User –** communicate need and operational requirements to Requirements Manager and SE
- **Requirements Manager:** Solicit, decompose and synthesize, track each requirement from the source to when delivered.
- **Quality Manager –** ensure people, process, tools performing and delivering to the expected standard

Process Activities

1. **Document Problem** into a Needs Statement
2. **Write and Document** the CONOPS
3. **Write** Each **Functional/User** Requirement
4. **Identify Key Performance Parameters, MOEs, MOPs**
5. **Decompose** Requirements and Document
6. **Gain Approval** of Functional/User Requirements
7. **Document/Decompose/Synthesize Systems Requirements**
8. **Gain Approval** of System Requirements/Document into Tools

Output and Results: Needs Statement, CONOPS, Functional/User Requirements, Systems Requirements, Key Performance Parameters, MOPs and MOEs	**Change Log** (printed copies may not be current)

Ver	Date	By
.1	04/01/20	BLS

story is understood and the requirements are discovered (as a problem-solving document), it is put

on the shelve and the team moves to the next step. And this is a mistake because others may join the project who would find a CONOPs extremely useful, other similar systems may need to interface with your system (that would require for them to read your CONOPs), and if the system changes scope, the CONOPs should reflect this change.

- **Functional (User)and System Requirements** – requirements are singular and discrete statements on what the system will do, and how it will perform.

Those performing this process in addition to the Systems Engineer include a Requirements Manager, Functional Expert or Customer (who knows the operations community), Contracts and Sub-Contracts if contract modifications are needed, and Configuration Management to baseline the requirements once approved, and of course Quality. The Project Manager will work closely with all these experts to ensure completion of the process.

3.1.7.1. Types of Requirements

- User Requirements – WHAT the system will do from a User Perspective, an individual statement, "The User shall…" contains discrete description of the function. User requirements are design agnostic.
- Functional Requirements – WHAT the system will do – similar to User Requirements. The term Functional Requirement tends to capture not only what the user does but also the function of the system. These functions are performed by a user who is a person or the function of a machine. Functional requirements are design agnostic.
- User Stories – Similar to functional users but provides a contextual understanding by stating 'As a User I want to ….', or 'As a Customer I want to ….'. User Stories replace User and Functional Requirements for Agile, or Scrum.
- System Requirements come from Functional Requirements and read in the form "The System shall…" System Requirements are not for the User but are for the architect and the developer and describe the WHAT the system will do. Although still design agnostic, the solution will slowly start to emerge on HOW the behavior will be delivered.
- Security Requirements – Security Requirements are what prevents unauthorized access to the system, unauthorized viewing of the contents or of the system, unauthorized modification of contents, or prevents an outside party from denying service to legitimate users. Security Requirements authenticate the source and contents of data that allows recipients the confidence that the data or function is legitimate. For government these requirements are outlined in the *Intelligence Community Directive (ICD) Number 503 Intelligence Community Information Technology Systems Security Risk Management, Certification and Accreditation* (Risk Management Framework (RMF)). (OD&I, 2008). Commercial systems can and should apply *ISO/IEC 27000 Information Security Management* (ISO/IEC, 2020). Security requirements really are functional regulatory requirements and can be expressed as so. The important thing is to include these requirements early so they are built in and not simply retrofitted later as an afterthought, at much greater cost. Additionally, most security requirements are in the schedule critical path and by not starting these tasks on time, delay of the project delivery is likely. One more thing on Security Requirements; I've seen where someone wrote a risk that security requirements will cause a delay in delivering the system. Actually, of course, security is risk prevention. **Security is your friend.**
- Performance Requirements – related to speed, duration, and throughput.
- Key Performance Parameters (KPP) – a requirement that is so important that if not met, the system is not considered successful.

- Technical Performance Measurements (TPM) – specific measurements used to assess design and build progress. TPMs are few and are apply criteria that provide insight to the Systems Engineer on the systems state-of-health.
- Interface and Integration Requirements – the expectation that function, data, control, or power are passed from one system, sub-system or element to another, and subsequently through the entire system. Also, that form and fit among two or more entities are compatible.
- Size, Weight, Power – ability to fit into a defined area, budgeted weight, power consumption over a given period, and heat emissions usually for aerospace systems, or any type of sea or land vehicle.
- Regulatory Requirements – Requirements to governance, Federal Acquisition Authority (FAR), policy and laws, security requirements such as ICD 503 Information Technology Systems Security Risk Management, and Federal Communications Commission (FCC) for spectrum allocation, or data compliance related to privacy and retention, Health Insurance Portability and Accountability Action (HIPAA) and other regulatory requirements. These are only a few Regulatory Requirements.
- … ilities – requirements regarding the use, maintenance, modification, upgrades of the systems. … ilities may be hard to measure and test but are often the most important requirements of a systems (Derived from *The INCOSE Book*). (International Council on Systems Engineering (INCOSE), 2011) … ilities include:
    - Availability – the system is ready to use, when and where needed, and that the system will perform as expected.
    - Usability – ease of use – simple, clear, intuitive, layout and feel to the operator – relates to ***Human Systems Integration***.
    - Maintainability – the ease of performing the maintenance on a system, to include the cost, time needed to perform maintenance, and minimal disruption to operations.
    - Scalability – how well the system is designed and built to allow for future growth. A scalable system will have plenty of potential for volume and growth.
    - Reliability – how well the system will perform with no failure, or if a failure does occur how quickly the system will recover. Reliability is often measured by ***Mean Time Between Failure (MTBF***) and **Mean Downtime (MDT).**
        - MTBF= [(start of uptime – start of downtime)/number of failures]
        - MDT= [(start of downtime – start of uptime)/number of failures]
    - Supportability – maintenance, time to repair, monitoring

'… ilities' are requirements of the system that are most salient yet, and are often hard to satisfy but without these, the system may not be considered successful. The methods and models called out in Section 4.1 are very useful in capturing these requirements.

- Measures of Performance (MOP) and Measure of Evaluation (MOE) – measurement of system operation in its environment – to be discussed in Section 3.1.13.2 Validation.

- Most Important Requirements (MIR) – MIRs are common in the capture and proposal world, before a Request for Proposal (RFP) is released, capture teams glean the MIRs from customers, by attending industry events, industry information, old proposals and other such sources. Once the RFP is released the customer has now told potential solicitors the requirements. If the Capture Team did a good job, the MIRs will match the requirements in the RFP, and the MIRs are no longer needed.

3.1.7.2. Requirements Activities

Requirements are **derived** as more detail is discovered and defined, from the user or from analysis. The Systems Engineer must document why the derived requirement is needed. As the requirements are synthesized, **Functional Analysis** is performed by the Systems Engineer to group like requirements into a **hierarchy**. This is the first visual towards a solution as one can see the sub-systems, elements etc. As the analysis and synthesis occurs the system will become more balanced so not to demand too much on one element and little demand on other elements. The hierarchy is used to create the **System Specification** which will show the detail down to each Configuration Items (CI) as is described in the CM/DM Process.

3.1.7.3. Tracing Requirements

Figure 3-6 is a simple requirements traceability table (may be called Requirements Verification Traceability Matrix (RVTM)) which is an end-to-end tracing of requirements – from the source, to user-to-system requirement, what feature of the system delivers that requirement, and the test case, how tested and the result. This process threads its way through the entire project and interrelates with other processes such as V&V and CM/DM. There are many computerized tools such as IBM's Rational Dynamic Object-Oriented Requirements System (DOORs), that are used to manage requirements but a simple spreadsheet may be suitable (IBM, 2020).

Figure 3-6 RVTM - Trace Requirements From Source to Solution

From The User Requirement or Regulatory Compliance – these are decomposed to a discrete level

One **Functional** to Many System Requirements – for the developer

Part of the Design Satisfies the Requirement

VER - Test is written to the require ment

Requirements Traceability Matrix – Follow the Thread from source Functional Requirement to System Requirements, to Test Cases							System Requirement	Design Component	Test Case
Req #	Priority	Source	Statement	Date Entered	Person Entering	Comments			
XXX.001	H	CONOPs 3.1	The User Shall …	5 Jan 2020	I Newton		The System Shall ….	A12 XXXX	
XXX.002	H	CONOPs 3.1	The User Shall …	5 Jan 2020	I Newton		The System Shall ….	A12 1121	
XXX.003	H	FCC xxx	The User Shall …	5 Jan 2020	I Newton		The System Shall ….	B12 1454	
XXX.004	M	FAR	The User Shall …	5 Jan 2020	I Newton		The System Shall ….	B15 5678	
XXX.005	L	Style Guide	The User Shall …	5 Jan 2020	I Newton		The System Shall ….		

3.1.7.4. Requirements Decomposition

The initial set of requirements are the Functional Requirements or User Requirements – WHAT the system will do for the user, from the user perspective. User requirements are expressed in the form of "The User shall…" repeatedly, each stating one single operation of the user. The first round (draft) of a user requirement often contains two or more ideas, may be nebulous, and this is OK. Initially, the user

must get their ideas 'down on paper' and into the RVTM. In conjunction with the Systems Engineer though, the User eventually will have one discrete operation for each 'The User shall … ' Statement.

For example, if a user needs to call their sales agent:

'The User shall contact the Salesman,' may be divided into;

- The User shall enter the telephone number.
- The User shall select the telephone number from the directory.
- The User shall prepare a text message.
- The user shall send the text message.

Without decomposing, the system **cannot** easily be designed and built, or tested without the ability to use a directory, or without the ability to make a call (Developer says," you only said you have to SEND a text, not receive them!"). Again, the requirements do not dictate or suggest a design.

A finished set of Functional Requirements will be understood by the stakeholders, written from the perspective of a user (from the operator, maintainer, manager, or auditor, etc.), but do not provide a technical solution.

The Systems Engineer will create System Requirements – or WHAT the system will do to deliver the required function. A system requirement will read 'The System shall…' – each expressing one function of the system, and each traced to a User or Functional Requirement. Creating system requirements may force the Systems Engineer to consult with the user for clarification and detail of a User Requirement (see **recursive**). But once completed, the system requirement will be thorough and descriptive enough to start the design. The requirements table illustrated in **Figure 3-6** Requirements Verification Traceability Matrix (RVTM) will expand to include the Systems Requirement, and eventually the design feature and the test case for VER & VAL.

The Truth About Requirements

If you have some experience in Systems Engineering, you may now be thinking that although the CONOPS and the Functional Requirements or User Story are to describe WHAT the system will do, they often describe HOW the system will perform the functions. This WHAT versus HOW is challenging under any circumstance in the systems world. The more a CONOPS or Functional Requirement describes HOW, the fewer options the designers have for innovative ideas of new technology or processes. A Functional Requirement that dictates a solution to build a high-speed horse and buggy will preclude the use of the automobile. A requirement that dictates a fixed phone with a 30-foot cord would eliminate a wireless phone. There is no easy answer to this except the user and operator, and the systems engineer must have an open dialog that fosters innovation into WHAT the user needs and HOW a system will function.

I would be remiss if I didn't mention the impact of cloud computing (see Section 6.3 Three Generations of SE). A new user requirement may quickly emerge that can be provided by special high-speed processing, quick storage, data analysis, desk top applications, artificial intelligence or other

applications, all available in a cloud. Instead of performing an extended Requirements Process, Users may tap into the Cloud and acquire the needed service to resolve their issue, provide an answer and once completed return the service to the provider. This evolution occurred within a 20-year period and has provided great IT power to the fingers of many users.

3.1.8. Configuration Management/ Document Management (CM/DM)

CM/DM is identifying and documenting key attributes of a system and program. The CM/DM Board (as defined in the CM/DM Plan or the SEMP) will declare the current or future baseline, and those who need it save the baseline document in a repository for easy access. CM/DM starts at the beginning of a project and goes the entire duration of a project and extends to every portion of a system.

A CM/DM person writes the CM/DM plan, and in turn will manage the plan and the process. The CM/DM person (CM Manager) will have a technical understanding of the artifacts created and how the project and system functions. The CM Manager may be assisted by other CM Specialist, plus is involved in virtually every activity and person that creates an artifact requiring control and accountability. They need not attend every meeting or review, but they must ensure the systems and project team comply to the process. Quality, as usual, will provide oversight.

Good CM/DM can save the project team and system function from miserable failure. At the origin of a project the teams are small, but as the team grows and the complexity of the system grows, so does the coordination and awareness of the system or project. Systems are not static; but instead, systems change to suit emerging requirements, for new technologies and to resolve problems. A dynamic and responsive CM/DM Plan must support this approach. The plan (and its associated processes) must identify the tool to use, those who will perform CM/DM, how to submit and approve any changes, and a naming convention of the artifacts under control.

Name: Configuration Management/ Document Management (CM/DM)	**Purpose:** Document and Approve Characteristics and Attributes of the System and of the Project. Control/Manage baseline
Inputs and Prerequisites: Approved Project, Artifacts to approve, approved tool CM/DM Plan, CM/DM Process	**Information:** Trace Requirements from Source to Testing, to When and How Delivered
Related Processes/Tasks: Start Project, Manage Project, Manage Risk, VER & VAL	**Tools:** CM/DM Repository, Chef, Puppet, CM Now! 
Roles & Responsibilities  • **Project Manager** – success of the project, ensuring compliance to process • **Systems Engineer** – Ensure Compliance, Capture and Document Configuration Items, Manage Change, Awareness • **CM/DM Manager** – Write CM/DM Plan, Manage CM/DM Process, Manage/Control Change to Process/ Configurations, Awareness • **Quality Manager** – ensure people, process, tools performing and delivering to the expected standard	

Process Activities

1. **Plan for CM/DM** – write or verify Plan and Process are suitable, Create Configuration Items (CI) Schema, train team
2. **Capture CI and Artifacts** – team proposed base documents – approved at CM/DM Board, enter into the Repository, Baselined
3. **CM/DM Changes** – as changes are proposed, consider risk, requirement, improvement or defect, VER/VAL, and how change will be made, and awareness, fallback. Calorize Change as needed from Major, Minor, or Administrative or something similar
4. **Capture Minutes** from CM/DM Board and enter into CM/DM
5. **Make Change** – make change according to approved plan, for major changes perform during least amount of risk to mission. Document changes, monitor, 'fall back' if needed
6. **Conduct CM/DM Audits** – Physical accuracy of Baselined CI as documented to the 'as is'.

Output and Results: Managed Configuration, CI Baselined and Approved, Traceable and diagnosable configuration

Change Log
(Printed Copies May Not Be Current)

Ver	Date	By
.1	04/01/20	BLS

A few definitions, provided below, are helpful.

- Baseline – the collection of items that identify the system configuration, that are approved by competent authority. This usually applies to the current and in use baseline (as built), but a future baseline may be defined (TBD). Often previous baselines are maintained and documented in case the current baseline fails and is required to fall back to the previous baseline. Baselines not only include technical items but also schedules, budgets, meeting minutes and briefings, and are determined by *competent authority*.
- Configuration Item (CI) – a critical artifact of the system that must be uniquely identified and documented with which only approved changes are made.
- Artifact – an item subject to CM or DM – may be a drawing, minutes, or a product.

Configuration Items that range from a low-level part, software, data, hardware up to the entire system are managed by form of a version number. For software, data and documents its usually a number separated by two or three decimal points. It looks like this:

- XX.YY.ZZ where XX is a major release. YY is a minor release to repair or upgrade, or add a minor feature. ZZ may be a patch or a minor enhancement.
- Large hardware systems, and systems of systems usually go by block number, i.e. Block 3 or Block 4. This is especially prevalent in aerospace systems or ground based weapons systems.

If you think about it, even U.S. Navy ships come in a class. For example, the first laid keel of a type of a ship is the class of the subsequent ships of that type – such as the **Nimitz Class** or the **Arleigh Burke Class.** Military planes carry the model such as the **B-52 F Stratofortress Bomber** versus the now retired B-52 D.

The level of approval for a release or a block depends on the level of risk associated with each, and the amount of control provided by the release authority.

The schema to which your system is managed must be defined early in the program and documented in the SEMP and/or the CM Plan, and outlined in the Project Management Plan (PMP).

Typical items that may come under CM/DM are:

- Schedules
- Contract Documents
- User Requirements Document (URD) that contain SON, CONOPS, User Requirements
- Requirements Traceability Matrix (RTM)
- Project Plans
- Risks, Opportunities
- Meeting Minutes
- Budgets
- System Architectures and identified Configuration Items
- Test Plans
- Test Cases
- Test Results
- Agreements such as ICD, APIs, MOEs, SLA, Charters
- Operator and Maintenance Manuals
- Systems Engineering Management Plan (SEMP)
- Form and Fit Dimensions
- Weight Specifications
- Performance – Required and Actual
- Customer Feedback

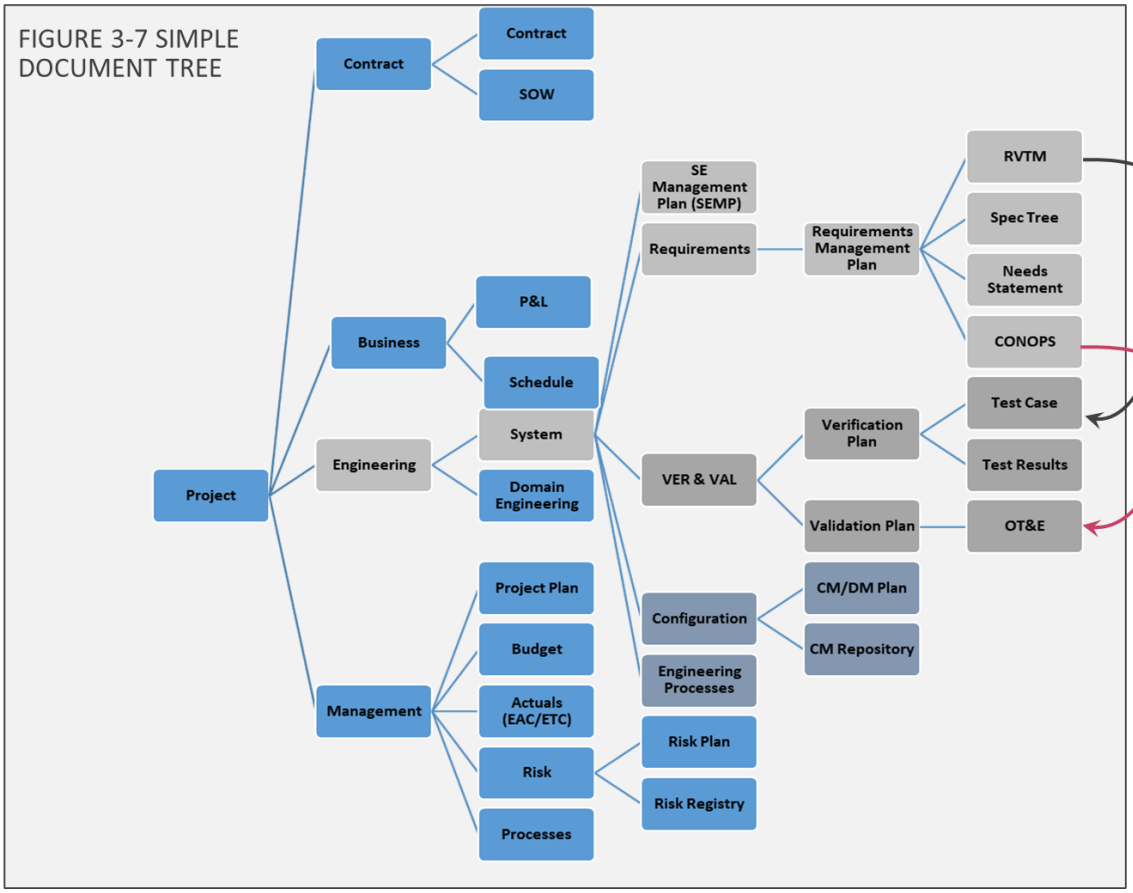
Not everything needs to come under CM/DM. My rule of thumb for CM/DM is:

1) Can the system be reproduced with the baseline documentation?
2) Is the cost of putting the CI under CM/DM less than the mission impact if the CI were not known?
3) Will an uncontrolled change to the system NOT cause mission failure?

If the answer to any one of the questions is NO, **then manage the configuration.**

A document tree is a list of documented that may be needed to support a system or to manage a program. The document tree will aid the Systems Engineer is mapping the needed documents, how they are related, the document hierarchy and who should be responsible for that particular document.

A simple document tree is illustrated in **Figure 3-7**. You'll see how the gray boxes call out the Systems Engineering documents.



3.1.9. Integration

I did not create a specific process for integration, but it is very much related to other processes. For example;

- Requirements that are mutual and dependent – may provide a service to the other.
- Architectures define the interfaces and the integration to the detail a CI can be assigned to it, and the interface must be tested.
- Development – to build in integration and document as outlined in the plan and in the architecture.

Many systems fail simply because they are not interoperable – they don't work together. While they may function for a short time, at some point the system will fail. If the system was not properly integrated, it will fail to pass data, control, connect or that the elements fail to separate when commanded. While an interface is the connection and mating of two components or organization, integration is the interconnection of the entire system, and to and from other systems. **Figure 3-8** illustrates **Interfaces** – the bulb and socket fitting together, that enable **Integration** of the entire illuminated city. If an interface fails, it will affect integration – once again the devil is in the details. Interfaces and Integration must be managed and controlled. For this discussion we will use both terms interchangeably although there is a difference.

Good integration requires careful design and a suitable level of CM/DM. Because of this, the CM Manager is involved to ensure each interface is documented and baselined. Testers create test cases

related to integration, and for a complex system a third party may be used to manage integration. Quality ensures these activities are performed correctly, and the result is to expectation.

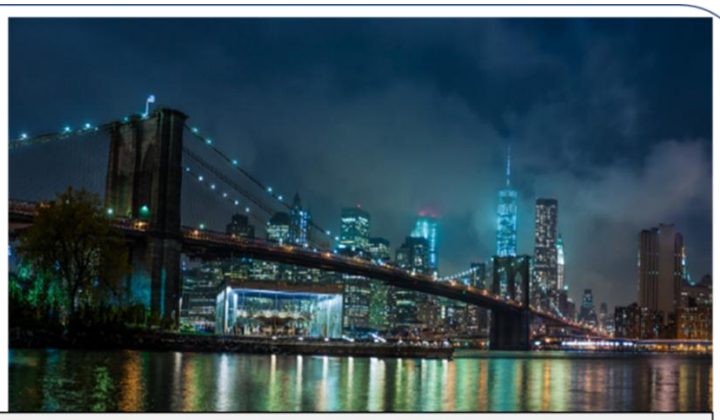


Figure 3-8 Interfaces – two objects fitting together, and Integration is the entire System Working Together

The best way to perform integration is to plan for it early in the lifecycle, to perform and improve integration. Integration should be defined in the system architecture in the form of **Interface Control Documents (ICDs), Application Programming Interfaces (APIs), Request for Comments (RFCs)** by applying standards outlined in bodies such as the *Institute of Electrical and Electronics Engineers International Telecommunication Union* (IEEE, 2020)*, or International Standards Organization (ISO)* (ISO/IEC/IEEE 42010:2011, 2017). Integration between teams should be agreed and documented in the form of Charters (see 3.2 Other Processes-Agreements), the SEMP, Memorandum of Agreements, and a definition of Roles and Responsibility. The responsibility of integration must be assigned to someone, and it should be a documented system requirement. For critical systems, a third party may be assigned to ensure all interfaces are thoroughly resolved and are included in the Verification and Validation Process.

Some of the integration tools include simply creating an architecture that illustrates the inputs and outputs. Other's methods include documenting the library of Interface Control Documents (ICDs) and to what systems, sub-systems and elements they are assigned. One that I will touch on is the N2 Diagrams.

N2 Diagrams illustrate elements of a function and their output, and corresponding input to another function. The outputs extend from the vertical side of the element (represented by a box). An empty box has no inputs or outputs. The use of N2 helps ensure all interfaces are accounted. TRW created N2 Diagrams in the 70s for complex software, but these were eventually expanded to include integration of hardware (Burge, 2011). **Figure 3-9** illustrates a simple N2 in which data is ingested

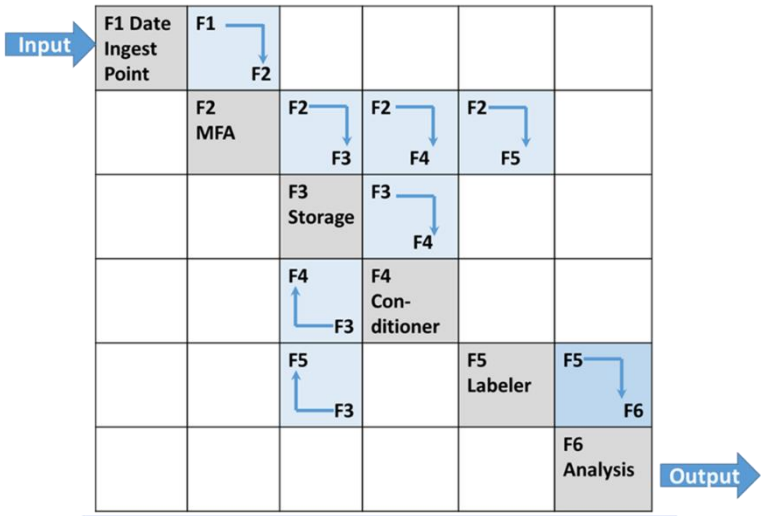


Figure 3-9 an Example of an N2 Chart – Leave No Outputs Behind

into a Cloud (for Cloud Computing), authenticated (MFA[4]), stored and contained, labeled and made available for analysis. There is one storage function (F3) that is updated from F4 Conditioner, and F5 Labeler. The output is rendered to the user from F6.

Faulty integration and interfaces are a major reason for problematic systems. Time and diligence must be applied to ensure integration is well defined, documented, verified, and correctly implemented.

3.1.10. Create Architecture (Architecting)

To this point I've made references to 'perspective' – everyone has at least one and it changes. The perspective is resolved in the Enterprise Architecture, a framework to determine WHO are the Stakeholders, WHAT are their interests, WHERE do these elements present themselves (including the stakeholders), HOW will the system function to deliver to their needs. These perspectives are laid out in a matrix with the X Axis listing the What, How, Where, Who, When Why and the Y axis listing the stakeholders. Where the X and Y meet is the perspective of that stakeholder regarding the issue of What, How, etc.

Prior to 1998 these perspectives (and interests of various stakeholders) were overlooked, other related systems were overlooked, and systems were severely stove piped. Budgets were overrun, systems were not interoperable, and for information technology there was little confidence on how much was spent. The Clinger-Cohen Act of 1998 mandated the use of an Enterprise Architecture based on the **Zackman Framework for Enterprise Architectures** (Zackman, 2008). Each national security organization required a Chief Information Officer that ensure compliance to the Act. Eventually, this idea was also rightfully implemented in other parts of government and civilian industry. The Enterprise Architecture is a strategic document that doesn't often change, but it does require flow down of perspectives to System and Design Architecture. Much of what Systems Engineering is today – the architectures, the languages of Systems Engineers, the methodologies described in Section 4.1 are a result of The **Clinger-Cohen Act of 1998**. But we are not finished – many organizations struggle to ensure the perspectives are really understood in their organization and reflected in their architectures. Once the Enterprise Architecture is reviewed that Systems Engineering and Architect will move to the next level – the Systems Architecture and Systems Design.

[4] **MFA is Multi Factor Authentication** - uses two or more sources to verify the identity of a person or an object. It may be two passwords, one password the other a finger scan, or a finger scan and answer a question only you would know, or any multiple sets of identification.

Name: Architecting	**Purpose:** Technical Solution, most suitable solution that satisfies the requirement and need
Inputs and Prerequisites: System Requirements, Risk Management, Trade Study, TEM, VER & VAL	**Information: Language of System Engineers,** Defined Notation and Vocabulary, Bounded System
Related Processes/Tasks: Start Project, Manage Project, Manage Risk, VER & VAL, CM/DM, Modeling & Simulation	**Tools:** PowerPoint, Excel, CORE, M&S Tools 

Roles & Responsibilities
Project Manager – success of the project
Systems Engineer: Create a compliant solution, suitable value – balance cost, schedule, risk
Architect – work closely with operations, requirements manager to understand need, requirement, create alternatives, aid in the selection of alternatives
Quality Manager – ensure people, process, tools performing, delivering to expected standard

Process Activities

1. **Review and Understand** Need, CONOPS, Functional, System Requirements. Review Enterprise Architecture.
2. **Parse 'like' requirements into a hierarchy** – so all security requirements in one set of function, user interface in another set, processes function into another, for aerospace – navigational in another set – this is defining sub-systems, elements
3. **Start High Level and Work Down** – System Architecture, Sub-System, etc. Created needed flow diagrams, Date Structures, Monitoring and Control, KPP, MOE and MOP
4. **Define Interfaces** – one system interconnects to others – standards, agreements, etc., – leads to integration.
5. **Document and Notate** – common and agreed to 'language of systems engineers. Estimate cost (BOM).
6. **Evaluate several alternatives** – Use TEMS and Trade Studies, full value of team to estimate risks, costs, performance. Make suitable selection
7. **Record** the 'to be' future architecture, use version # – use DM/DM Process
8. **Make Changes** – as updates proposed include in the 'to be' and perform same set of activities.

Output and Results: Baselined Architectures, Flow Diagrams, Bill of Material (or list of items), Managed Risks

Change Log
(Printed Copies May Not Be Current)

Ver	Date	By
.1	04/01/20	BLS

System Architectures are defined and bounded to WHAT will suit and satisfy the functional and system requirements. It is transitioning from WHAT the system will do, to HOW it will do it. A common set of architectural templates with common icons, notation, descriptions, format, etc., must be in the Architectural Document.

The Systems Engineer is intimately involved in the architecture process by ensuring requirements are traced to a design feature and the best alternative is selected. Depending on the approach taken, you may need an expert in one of the languages discussed in the subsequent Section 4-1 **The Language of**

System Engineers - UML, SySML, IDEF, etc. Systems Engineers, Requirements Managers and Architects often become quite fluent in these languages because the common language reduces confusion and allows you (as the Systems Engineer) to spend less time trying to recall the language and more time to understand the solution. **CM/DM** and **Quality**, again, are involved in this process.

I have used a tiered approach to visualize how architectures are created. This tiered approach is not an industry standard but is only used here to help describe the three levels of architectures. As will be discussed in the next section, **Enterprise Architecture** is a strategic organizational design offering differing perspectives – depending on one's position of the system – the user has one perspective, the maintainer, the business entity, buyer yet another. (Recall Section 1-5 and Table 1-1 Who Are Systems Engineers) The Enterprise Architecture is useful in creating the next tier – The **System Architecture** in that it provides scope, has identified perspectives of various stakeholders needed for the Systems Architect. The third tier is the **Architectural Drawings** which contain the needed detail for development. A **Technical Design** provides the most basic level of detail calling out specific hardware and software to allow the building or assembly of the system, costing with a Bill of Material (BOM), troubleshooting, updates and pulling in the necessary services provided by Cloud Computing.

3.1.10.1. Approach to Creating an Architecture

Architects use the System Requirements and the CONOPS to create an architecture. This is done in a top-down structure, to illustrate the top-level look at the system and sub-systems and relationship to other systems. The system is continuously decomposed according to the requirements with each level showing more level of detail. Typical architectural documents include:

- Process Flow
- Interfaces
- Data Structures
- System Monitoring and Control
- Implementation of Policy – restrictions, auditing, etc
- Illustrated Entities
- Narrative Descriptions of the drawings
- Defined Notations, ICONs
- Transition – from the legacy system to the new system
- Title and Version Control Number

3.1.10.2. Use Your Processes

Requirements are matched to each element to allow for tracing through the system. Systems Analysts will aid on throughput, capacities, load balancing and reliability of the system and will work with the Architecture to adjust. **Risks, Issues and Opportunities** are captured for each alternative. The Architect is not too proud to reach back to lessons learned or architectures of other system – thereby saving time, resources and perhaps using components that are already available to the organization. Several alternatives should be considered based on the previously established selection criteria based on **Cost, Schedule, Performance and Risk**. The results of **Modeling and Simulation** will aid the Systems Team in conducting a **Trade Study** of the most suitable alternative.

Iteration and **Recursion** are common during the Architecting Process in that the Architect revisits the real meaning of a requirement so they may design into the system. The Architect will dig deeper to understand and solve a problem only to discover that another solution of another part of the system is dependent. Key requirements such as **Key Performance Indicators, Measures of Performance** and **Measures of Evaluation** – all of which are previously documented in the Requirement Process – must be carefully factored into the Architecture.

Does Your System Talk to You?

Throughout this Book I referred to a system's 'behavior', and to Iteration and Recursion. When modeling and simulating a system, or when troubleshooting a problem, the Systems Engineer is (in a sense) having a dialog with the system. The Systems Engineer formulates a question based on observation and system behavior, 'asks' the question and evaluates the reply back. Alistair Cockburn of Agile Software Development referred to this dialog which I found very interesting. So, when performing your job as a Systems Engineer, remember that you are having a dialog with your system. Listen to It. (Cockburn, 2007)

The Systems Engineer and the Architect will lead a series of **Technical Exchange Meetings** to collaborate on the alternatives and the most suitable solution. While being an Architect tends to be a solitary profession, it is also a team exercise to ensure the project team and stakeholders are reasonably comfortable with the solution and are ready to progress. The engineers commonly use all levels and all types of drawings to ensure a sound solution, but the managers of the program will usually only review the top-level systems and sub-systems drawings.

If you are a newcomer to Systems Engineering (and Project Management) you will now realize how these activities evolve from a set of requirements to a design, the business activities, build and deliver, and why a methodical process of system engineering as needed.

3.1.11. Conduct Trade Study – It's Good to Have Options

Trade Studies/Alternative Analysis are used when given two or more alternatives, then conducting the research of the alternatives based on cost, schedule, performance, and the value offered to the stakeholders. The established criterion is used to select the most suitable alternatives. Criteria are usually weighted in value to what is most needed and desired for the system. The Alternative Analysis process is often used to select the best course of action – such as to buy, build, modify or maintain the status quo. Systems Engineers and Systems Analysts use modeling and simulation, prototypes, public

and private publications, and knowledge obtained from other similar systems to help make their decision. Virtually anyone on the project may use the Trade Study Process. The 'earned wisdom' or knowledge equity of the studies should be baselined into an Enterprise Library for future use on other projects or systems.

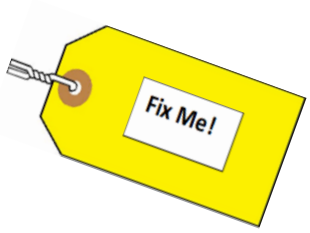
<table>
<tr><td>Name: Conduct Trade Study</td><td>Purpose: Given a situation with two or more alternatives to choose, make the most suitable selection.</td></tr>
<tr><td>Inputs and Prerequisites: Alternative and criteria for selection.</td><td>Information: Two or more alternatives to choose. May be a contractor, product, service to select. This process may be performed by any stakeholders during any time of the lifecycle.</td></tr>
<tr><td>Related Processes/Tasks: Manage Requirements, Architecting, Manage Risk, CM/DM, Modeling & Simulation, Decision Tree Analysis</td><td>Tools: Excel and Common Desktop Apps  </td></tr>
<tr><td colspan="2">Roles & Responsibilities 
• Project Manager – success of the project
• Systems Engineer: Ensure criteria established, weight factors, critical issues identified, dates, times, etc.
• Person Conducting Trade Study – often the contract/sub-contract manger, but also the SE or any stakeholder
• Quality Manager – ensure people, process, tools performing and delivering to the expected standard</td></tr>
<tr><td colspan="2">Process Activities
1. Trade Study Alternatives – the selection that must be made, the pros/cons, critical info for date/time, performance - weighted factors.
2. Peer Review – to ensure Trade Study Alternatives are correctly established – modify as needed.
3. Conduct Research – without making the assessment, consider most viable alternatives by research, trade shows, modeling and simulation, performance data, and conduct analysis. Score the results. Sophisticated Trade Study may 'mask' the product or service provider to reduce bias.
4. After Data is Collected – score the results and peer review, check.
5. Conduct Selection – recommend – decide contingent on level of authority (Table of Authority)
6. Document the Results – in project plan, system plan, architecture, in KM and ensure relevant stakeholders aware.</td></tr>
<tr><td>Output and Results: Product or Service Source is Selected, Action Items, Updated KM</td><td>Change Log
(Printed Copies May Not Be Current)<table><tr><th>Ver</th><th>Date</th><th>By</th></tr><tr><td>.1</td><td>04/01/20</td><td>BLS</td></tr><tr><td></td><td></td><td></td></tr></table></td></tr>
</table>

Figure 3-10 illustrates a simple trade study to fulfill the 'Need for Speed'. You'll note how the requirements are defined and prioritized using a 1 to 5 scale. The capability or value delivered by each is also scored on a 1 to 5 scale, with 5 being the highest and 1 the lowest. These are multiplied and summed with the results along the bottom. Often, the results are different than what you would expect. While your heart is set on the Corvette, the BMW is the better solution.

Figure 3-10 Simple Trade Study for the Need for Speed													
(5 Superior, 4 Sat, 3 Can Mod, 2 Marginal, 1 Doesn't Have)													
Requirement	**Priorty**	**Corvette**		**Mustang**		**Charger**		**Porsche**		**BMW**		**Tesla**	
Sustained Speed	5	5	25	4	20	4	20	5	25	4	20	4	20
Acceleration	4	5	20	4	16	4	16	4	16	4	16	5	20
Curve Speed	4	4	16	3	12	3	12	5	20	5	20	5	20
Comfort	3	3	9	3	9	3	9	3	9	4	12	4	12
Safety	3	4	12	3	9	4	12	5	15	5	15	4	12
Fuel Economy	3	1	3	1	3	1	3	1	3	1	3	4	12
Price	5	4	20	4	20	4	20	3	15	4	20	2	10
Transmission	2	4	8	4	8	4	8	5	10	5	10	5	10
Color	3	4	12	4	12	5	15	3	9	3	9	4	12
4 Seater	2	1	2	3	6	3	6	1	2	3	6	4	8
Air Conditioning	3	4	12	4	12	4	12	4	12	5	15	2	6
Wheels	4	4	16	4	16	4	16	4	16	4	16	4	16
			155		143		149		152		162		158

3.1.12. Manage Defects and Improvements

Most (if not all) systems have an escalating problem resolution structure. For large systems, the level of maintenance is 1) **Organization Level Maintenance, 2) Intermediate Level Maintenance, and 3) Depot Level Maintenance** (Bell & Defense, 2007) – which often culminate in major upgrades or repair of systems, sub-systems, and components.



Escalating maintenance also applies to smaller systems, as it enters operation and maintenance (starting with Validation), a process to collect defects and potential improvements must be in place. These defects and potential improvements must include the date, time, location, errors codes and computer dumps, configuration information, mission impact (priority level), and function at the time of the problem. Defective parts are removed, tagged, and isolated from the rest of the inventory to prevent mistaken reentry of a faulty part into the system. Flawed or defective computer environments are locked down and images (dumps, configuration, etc.) taken to aid in trouble shooting. Accident scenes are taped off to restrict entry. Typically, the users notify a service desk to report the problem. Simple problems are usually resolved on the spot but more complex problems are elevated through several layers.

Name: Manage Defects & Improvements	**Purpose:** Operations and Maintenance (O&M) – process to capture defects or potential enhancements to a system.
Inputs and Prerequisites: Operational System includes Users, Maintainers, Support. Usually considered to have met IOC, KM and Other Similar Problems	**Information:** Defects or Potential Improvements, Predict Issues or improvement - Predictive Analyses
Related Processes/Tasks: Trade Study, TEM, CM/DM	**Tools:** Service Now! HP Help Desk, Excel, Telephone/Email 

Roles & Responsibilities 

- **Project Manager** – success of the project
- **Operations Manager** – if project is delivered and is in O&M, the Ops Manager may fulfill the job of a PM.
- **Systems Engineer:** Frequently review performance metrics, reported defects, trends, ensure process compliance.
- **Service Desk Manager** – trained workforce, with correct and usable tools, timely completion and resolution.

Process Activities

1. **User/Operator Has Problem or Observe Potential Improvement** – call, send email, report via online – date/time, name, contact number, system behavior, error codes, operational impact must be documented and reported
2. **Service Desk** – Completes documentation (Ticket), assesses critical level (1,2,3,4), initial analysis and troubleshooting, resolve on the spot if able using other similar documented problems. Isolate or cordon off the defective area.
3. **Elevate the Problem** – until it is resolved – may require system modification using Manage Requirements, V&V, CM/DM
4. **Make Change or Correction** – Awareness to informing user, to more complex changes may include training.
5. **Document Results** in CM/DM and KM

Output and Results: Updated or Closed 'Ticket', KM Updated, CM/DM Completed

Change Log
(Printed Copies May Not Be Current)

Ver	Date	By
.1	04/01/20	BLS

A typical escalation process is;

- Level 0 - automated or Self-service such as password reset, knowledge base lookup.
- Level 1 – filters Help Desk Calls and basic support, troubleshooting, use of a documented knowledge base of previous and known questions. 'Tickets' are used to make a documented record of the call. If needed and the problem is not resolved at Level 1, it goes to Level 2.
- Level 2 – resolve break, fix and configuration problems, hardware, and software repair. May be 'touch' maintenance where a technician is dispatched.

- Level 3 – generally used to resolve infrastructure problems. Synonymous to Depot Maintenance.
- Level 4 – complex problems that will require an external specialist to resolve.

Each Level 1, Level 2, Level 3 and Level 4 call or request is (using a ticket system) documented with the caller's name, contact, date, time, nature of problem, etc., and tracked to closure. A well-managed system will use metrics from these tickets to continually improve the system.

Information Technology Service Management (ITSM) (formerly Information Technology Infrastructure Library (ITIL) has a great model for capturing IT needs, requirements, improvements, and customer relations (Freshworks, 2020). ITSM is discussed in more detail in Section 4.2.8.

3.1.13. Verification and Validation (VER&VAL) (V&V)

Verification and Validation are the periodic 'checks' throughout the system lifecycle to confirm the system is being built to meet the requirements. In either case, VER and VAL must be linked to a requirement - be it a functional requirement for Verification or a Use Case from the CONOPS for Validation. Although these are two separate processes the processes are closely related, yet have distinct differences, and are usually two separate processes – one for VER and another for VAL.

High risk and complex systems will have a very robust set of V&V activities, but less complex or less risky systems will have fewer V&V activities. Like any other activity and process, the V&V processes must be factored into the lifecycle and the project plan. VER and VAL share methods in how a system is 'checked'. These methods to perform VER and VAL are;

- Test – controlled conditions, real or simulated.
- Inspection – visual or dimensional exam.
- Demonstration – correct operations against observable characteristics.
- Analysis – modeling and simulation.
- Analogy or similarity (type) – evidence of similar element.
- Simulation – performed on models.
- Sampling – statistical sample.

(International Council on Systems Engineering (INCOSE), 2011)

Verification and Validation Teams should be organizationally separated from the Systems or the Project Team and should remain as a 'neutral' party from the organization. Verification Team may be part of the program, but Validation Teams (who perform operational testing) are usually proxies to the customer. (Hence Independent Verification & Validation or IV&V).

Name: Verification and Validation (VER & VAL) or V&V	**Purpose: Verify (VER)** the system and its parts are designed, built and delivered as outline by the requirements **Validate (VAL)** that the system to be delivered will meet the expected operational need, in its expected environment.
Inputs and Prerequisites: CONOPS, Functional Requirements as documented in requirements registry, Risk Management, Architecture, O&M Process, Defect Management Plan & Process, Test Readiness Review (TRR). IOC for VAL	**Information:** VER and VAL are normally two separate processes but are combined here for succinctness and due to many similarities. VER and VAL start early but become front and center during and right after development.
Related Processes/Tasks: Manage Risk, CM/DM, Modeling & Simulation	**Tools:** PowerPoint, Excel, CORE, Test Tools 

Roles & Responsibilities

Project Manager – success of the project

Systems Engineer: Ensure all requirements are traced to a test case, and ready to take output of test results.

Requirements Manager – all requirements in repository, traced from source to solution

Verification Manager – create a test case for each requirement, procure resources to conduct testing

Validation Manager – create test plan/ scenarios for system performance in expected environment - CONOPS.

Quality Manager – ensure people, process, tools performing and delivering to the expected standard

Process Activities

1. **Prepare for Verification**– work with Requirements Manager and Operations to create test case for each requirement,
2. **Ensure Resourced** – test tool, capture test results, test environment, test product, testing time, data. Prepare a method to observe and capture the results so they may evaluate.
3. **Conduct Awareness** – according to plan and process – awareness may be simply read, checklist or formal training.
4. **Capture and Analyze Test** – if complete, accurate – conduct test again as needed. Capture problems or issues
5. **For Validation** – create Use Cases or Scenarios to perform based on the system in its operational environment. Work with Operations Community to ensure training is performed, maintenance, support activities in place.
6. **Evaluate and Assess** – after the results are captured, assess compliance and suitability to the pre-established criteria of the requirements.

Output and Results: Test Plan, Test Cases, Test Results, Defects and Improvements, Results of KPP, MOPs/MOE.

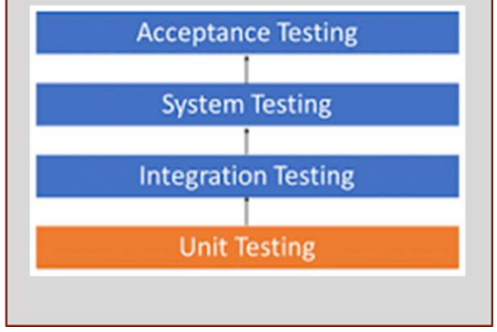
Change Log
(Printed Copies May Not Be Current)

Ver	Date	By
.1	04/01/20	BLS

3.1.13.1. Verification (VER) – 'Did We Build It Right?'

The Systems Engineer or the Requirements Manager will perform Verification of the system to ensure it meets the requirements. All requirements are entered into a Requirements Verification Traceability Matrix (RVTM) which was discussed Section 3.1.8 Requirements Management. As the system is developed, Verification becomes more formal as the Testers perform a robust set of verification activities. **Figure 3-11** Levels of Testing is a simple illustration of how a system is Verified and Validated, from bottom-up Unit Testing (code testing often done by the developer), and as the system is assembled Integration Testing then System Testing. Finally, there is Acceptance Testing which is more Validation.

Figure 3-11
Levels of Testing



3.1.13.2. Validation (VAL) – 'Did We Build the Right Thing'

Validation is performed against the CONOPs or Use Case, within the expected environment by the operators who will use the system, maintained by those who will fix it, on the infrastructure it will reside, and with no external help as would be expected in a mission or business setting. Validation is often performed through **Operational Test and Evaluation (OT&E),** used to validate the operations and support of the system in its intended environment. Validating organizations spend significant resources planning for Validation by learning the system and its design, and to how best to Validate the system. Test scenarios are prepared matching the operational and support expectations. This will include functionality, performance, stressing the system in the range of specified environments and load, usually performed in incremental and increasing levels, while evaluating the User and **Operator Experience**. The major areas of Validation - Operational Test and Readiness – as discussed in Section 3.1.7.1 Requirements Process) are:

- **Measures of Performance (MOP)** – the physical and functional attributes relating to the system operation, measured, or estimated under specific testing or operational environmental conditions. MOPs measure the design and performance requirements.
- **Measure of Evaluation (MOE)**– the overall operational success criteria from the users' point of view. MOEs are design agnostic. MOEs are derived from MOPs.
- **Key Performance Parameters (KPP)** – a requirement that is so important that if not met, the system is not considered successful
- **Technical Performance Measurements (TPM)** – specific measurements used to assess design and build progress. TPMs are few and are select criteria that provide insight to the Systems Engineer and PM on the project's state-of-health.

These terms are paraphrased from the *INCOSE Handbook, 4th ed., pp 133-135* (International Council on Systems Engineering (INCOSE), 2011).

The system will meet **Initial Operating Capability (IOC)** so a small set of the system is available for Validation. Failure to validate not only wastes money but may jeopardize the operator's safety and security as a result of the unexpected emergent behavior, or the lack of expected behavior. Once acceptance is complete, subsequent systems are installed and built out to **Full Operational Capability**

(FOC), and the system enters **Operations and Maintenance** and will satisfy the operational need as described in the CONOPS.

Verification and Validation Professionals put the 'stamp of approval' on a system. They are recommending approving to the Decision Gate Authority to approve the system for operations. A flawed system that is granted Approval to Proceed (ATP) for operations before it is ready can easily result in a catastrophic failure and even loss of life.

3.1.14. Conduct Project Management Review (PMR)

Periodically, the status of the project (or program) is reviewed and assessed to ensure the activities are progressing as planned, and to allow the project to make any necessary adjustments. A PMR is similar to a Gate Review, but the PMR usually occurs each month. With the presentation, there is usually a written report or at least a set of slides. The Project Manager presents the PMR to the customer, and those participating are the managers for each area to include Requirements, CM/DM, V&V, Finance and Contracts, and of course Quality.

The Project Manager will prepare the metrics in conjunction with the Business Team. These metrics will include the **Estimate at Complete (EAC),** and **Estimate to Complete (ETC**), and any other financial metrics such as **Earned Value (EV)** (PMI, 2020)**.**

- EAC is the total amount of budget spent from the start to the end of the project.
- ETC is the total amount of budget spend from a point it time of the project to the end of the project in its remaining time.
- Earned Value aligns the metrics of cost and schedule to the completion of work into a common index. An EV of 1 means the budget amount has been spent, at the scheduled time, and all the planned work is completed as scheduled. The project is on time, meeting all requirements to date, and is on budget.

The Schedule is also reviewed, as are the user and customer satisfaction metrics. **Risk, Issues and Opportunities** are reviewed, and direction given, if so needed. The Project Manager will brief the ongoing activities, state what deliverables have been delivered, material on order and received and the plan for the next period. Another key topic is staffing, especially on those systems with heavy staffing requirements. This topic is discussed to project losses, vacancies, gains, and often customers desire to know why there may be vacancies or why there is a high turnover. Budget and Schedule variances are noted, and adjustments may be made at the direction and approval of the customer. Tickets, defect, and potential new features are reviewed and may enter the lifecycle as a new requirement. Action Items and Minutes are taken and distributed after the PMR is complete, and these minutes are baselined into a DM repository.

For Cost Reimbursable Contracts – the customer takes special interest in how the program is run to include variances, staffing, risks, etc.

For Firm Fixed Price – the contactor or system provider bears the cost and schedule risk, so the Customer PMRs will brief. However, there will be organizational PMRs (may be part of the organization's **Business Management Review (BMR))** in which these topics are discussed.

Name: Conduct Project Management Review (PMR)	**Purpose:** Communicate the stakeholders of the project and systems status, plans and illicit stakeholders in making decisions.
Inputs and Prerequisites: Project Plan, Schedule, Budget (EAC/ETC), Risks, Deliverables, Staffing, Action Items	**Information:** Includes Cost, Schedule and Perform, People Process, Tools and Risks. May be Project or Program Management Reviews, conducted approximately monthly.
Related Processes/Tasks: Manage Risk, CM/DM, Modeling & Simulation	**Tools:** PowerPoint, Projector, Computer, Room 

Roles & Responsibilities 

- **Project Manager** – success of the project, lead the PMR, prepare presentation, collect information
- **Contracts Manager** – to ensure compliance to contract, and to answer questions related to the contract.
- **Systems Engineer:** Collect information from SMEs, Tech Team and other engineers, synthesize and provide to PM
- **Finance/Contracts** – provide information and reporting to the PM to include in PMR. Attend PMR.
- **Other Expertise** – depending on project and systems activities, others who may attend are V&V Manager, Requirements Manager, Task Managers, Supply Managers, Staff/HR, Usually Quality Manager attends.

Process Activities

1. **Compare Communication -** PM, SE and others periodically compared communication from stakeholders and customer to ensure all topics (emerging, new, ongoing) are captured. Review formal documentation of same.
2. **Project Manager** – extend 'data call' to relevant stakeholders – pertaining to the PMR. Check Contract to ensure required topics are included. Include suspense date when due back. Best to put into the Ops Rhythm.
3. **Prepare for PMR** – send out agenda, read ahead, reserve facility, check to ensure equipment functions, have a note taker. Often the PMR will include a written report – be sure the report and the briefing are consistent.
4. **Conduct PMR** – 1) Introduce and Purpose 2) Accomplishment 3) Cost (EAC/ETC), Schedule, Performance 4) Risks, 5) Plans or Requested Changes/Questions 6) Staffing 7) Review Actions
5. **Complete PMR** – Record Meeting and Actions Taken – obtain approval and put into CM/DM

Output and Results: Record of Meeting, Action Items, change requests.

Change Log
(Printed Copies May Not Be Current)

Ver	Date	By
.1	04/01/20	BLS

Some PMRs have become low profile and routine enough that only a few people attend, and it may only take a few minutes. In some cases, PMRs do not even occur. However, 'leaning out' a PMR to save time, but will overlook critical items is not a good idea. If the system is performing as expected and the project is going well than a boring PMR is good.

Don't Be a Watermelon

I've attended many a management and technical reviews where the project or system looks good on the outside but is wrought with problems on the inside. The metrics and charts don't always tell the truth. Eventually, those problems on the inside reveal themselves in the undesired emergent behavior, schedules slippages and cost overruns. Hiding these problems demonstrate a lack of sincerity with ourselves and the recipients of our system. Your organization should have the core values to disclose problems and resolve as a team, - early, and have the leading indicators to identify and prevent problems – Don't be a watermelon.

3.1.15. Conduct Technical Exchange Meeting (TEM)

A TEM may occur at any time in the lifecycle by anyone deeming it necessary. TEMS may be very structured and formal, or they may be almost impromptu and informal. In any case, TEMs should have an identified problem and purpose, the right people should attend, problem to resolve can openly be discussed (often on a white board), and the results recorded and agreed. White board talks that solve a problem attended by only a few of the right stakeholders, of which the results are not known are of little value and cause confusion. The Systems Engineer should sanction all TEMs, not that the System Engineer needs to attend, but the Systems Engineer needs to ensure there is an agenda, problem to solve, the right people attend. Relevant Subject Matter Experts should attend as well.

TEMs are where problems are resolved and often, TEMs are the seeds of progress where the best brainstorming occurs, respected opinions are heard, and inhibitions and restraints are often released. But because of their value, the process should be documented, and the results captured and shared, and factored into to the system solution. Conducting a TEM can be spontaneous (which may violate the operations rhythm theme). But given the choice of working a problem when the opportunity is right, versus waiting a few days for all the right people during which the problem may be overcome by other issues, hold the TEM.

Name: Conduct Technical Exchange Meeting (TEM)	**Purpose:** Resolve a technical or programmatic problem.
Inputs and Prerequisites: Risk Management, Requirements, Defects, Architectures, Manage Defects, Action Items	**Information:** TEMs cover a range of issues, and start at the beginning, go through all development to O&M and even to retirement and system disposition. May be quite formal but often less formal is better.
Related Processes/Tasks: Manage Risk, CM/DM, M&S, Architecting	**Tools:** White Board, PowerPoint, Excel, Modeling & Sim. Tools, Meeting Room 

Roles & Responsibilities 

- **Project Manager** – success of the project
- **Systems Engineer:** Scope and scale, bound problem to resolve. Ensure problem, alternatives and solution are documented and included into the system and project plans.
- **Requirements Manager** – all requirements in repository, traced from source to solution
- **Any Engineer, Technician, Manager** – given a problem to resolve may call a TEM with peers, to resolve.

Process Activities

1. **Define the problem** – document, bound the problem, determine key participates.
2. **Coordinate with Management or Peers** – prepare agenda, problem statement, send invites. Establish decision criteria.
3. **Draw Problem, Background, Restraints** – collaborate and brainstorm, sketch problems and potential solutions.
4. **Conduct Trade Study or Assessment** – include problem statement, selection criteria, assumptions, alternatives
5. **Conduct Modeling and Simulation** – as needed, and as part of the TEM and Trade Study
6. **TEM Make Recommendation** – given this info from Trade Study, Modeling and Sim, and White Board
7. **Systems Engineers and Team Make Selection** – contingent on their level of authority (Table of Authorities)
8. **Incorporate into Plan** – get needed change approval or Use CM/DM process, and awareness to team

Output and Results: CM/DM, KM, Action Items, Record of Meeting

Change Log
(Printed Copies May Not Be Current)

Ver	Date	By
.1	04/01/20	BLS

3.2. Other Processes

I was unable to call out all the process you may need to deliver a system, or to manage a project. The ones I detailed (generically) are the ones you'll most likely need but I've summarized a few more that may be critical to success, depending on your organization and your system.

Establish Agreements are agreements between two or more parties. For stakeholders working on a project within the organization have at least implicitly agreed to use processes and comply to organizational policy, those outside the organization have not, and agreements must be established. These agreements are in the form of Statement of Work, Task Orders, contract modifications,

Memorandum of Agreements (MOA), Interfaces Controls, performance, how data or propriety information will be handled and also, **Service Level Agreement (SLA)** on how problems are managed, and improvements are made. Often agreements are overlooked because Roles and Responsibly are not well defined, which goes back to the importance of Section 0. Agreements must also call out how data and information, and specifically licenses are handled. For example, if intellectual property (IP) is provided to a developer how will this IP be handled and then deleted when no longer needed? How will software licenses be transferred when the contract is completed? (which is part a transition activity). However, the Project Manager and Systems Engineer should also be aware that the more detail and reports that are requested the more costly the effort in that resources are consumed to create and deliver this detail.

Transitions occur when a legacy contractor departs the project, and a new contractor is taking over. It occurs when a new contract or effort is starting, or a system is expanded in scope. It may also occur in the merging or breakup of an organization. Larger transitions should be treated as a project of their own, with a **Transition Manager** and a staff, a project plan and associated processes, and checklist that are signed by applicable parties when completed. Plans for transition must be built on consensus between the parties. All parties must avoid 'scoring points' against an opponent at the cost of the mission or business – customers know this and don't forget. Managers and Leaders again, must think in terms of Cost, Schedule, Performance and Risk, and People, Process, Tools. Transitions must account for how material, processes, property, and software licenses are transferred. Those departing the project should have help in finding new jobs and getting the due recognition as they depart, and those arriving should be part of the transition plan that includes on-boarding, orientation, where and how each person fits in the organization, the organization's mission, and their role in the mission.

Knowledge Management (KM) - the organizational experiences and lessons are documented and then applied back on the organization. Knowledge is documented in the form of processes and checklists, and through undocumented **tacit and tribal knowledge** of the team – in that 'we just do it that way based on experience'. Its best to draw on this knowledge and document it so the entire team has the same advantage of the experienced team members. Lessons Learned must be part of KM. Knowledge and lessons should also be documented, and there should be a method for the team to submit lessons during a project (while fresh on their mind) as well as after the project (or a phase) is completed. There is nothing more frustrating than having teams repeat the same mistakes from one task to the next simply because the lesson was not encoded into a process change. Knowledge is of capital value, or corporate equity and should be treated as such.

> 'Its fine to celebrate success but it is more important to heed the lessons from failure'
> Bill Gates
> (Gates, n.d.)

Customer Feedback can be formally captured by way of a survey or perhaps a public statement. Any deficiencies must be captured, and a plan is established to improve, and the good comments must be applied on other projects. I have found that some of the most useful feedback comes simply by informally listening and observing the customer. The Systems Engineer and Project Manager must be patient, carefully asking questions and when you least expect it, they will tell you what they really need – such as '*we are losing five people next month*', or '*the director is combining our two originations next fiscal* year', or '*our budget is being increased by 25 percent*'. Take these clues, then plan, and offer a helpful solution to the customer. This is of great value (Section 6.2. on Value) to the

mission, the customer, and your business. Help solve their problems before they even know it's a problem.

Continual Improvement (may be called Continuous Improvement) is discussed more so in Section 5.4.1 but if you have a process, it should include how you can optimize your system, your project, and the business. Not only does it include Continual Feedback but also set of measurements and metrics to find vulnerabilities, bottle necks and risks, opportunities to improve and to accelerate the completion of a task. Continual Improvement is not only preventive, it is also predictive – again which is discussed in Section 5.5.3 but because it is often a formal documented process, I didn't want to overlook how it should be documented.

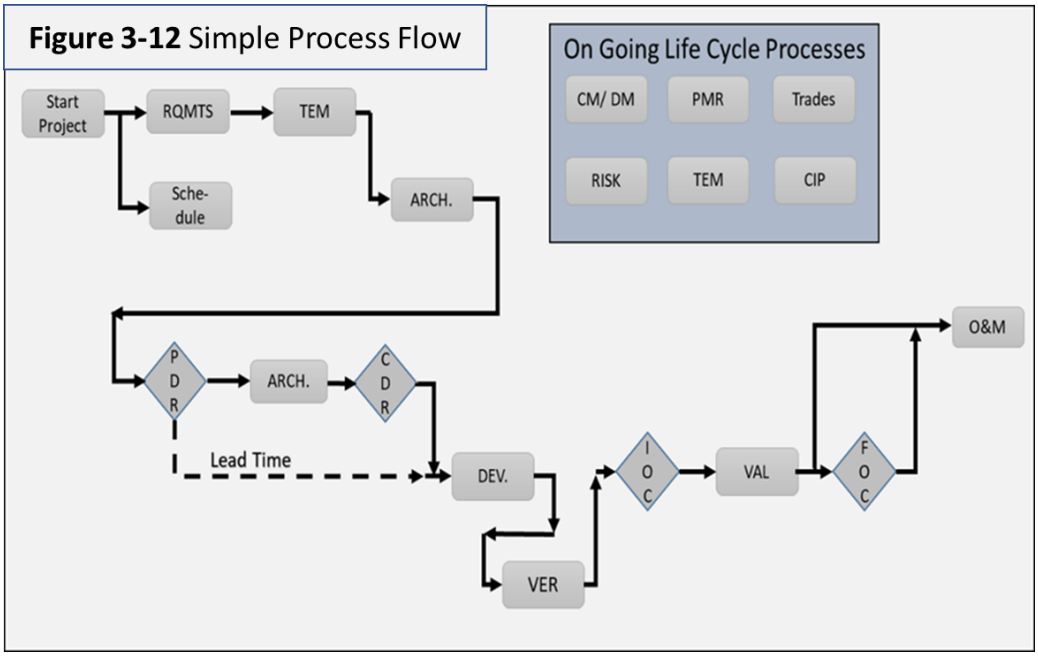
3.3. Best Practices

Organizations and industry groups have established 'Best Practices' that are applied on their projects, systems, organization and industry. The range of Best Practices are from documented procedural direction to loosely stated recommendations. Some Best Practices apply to an industry segment at large (Environmental Best Practices, HR Best Practices) as a framework and guiding structure for an industry or an organization. My take is (not to be cynical) Best Practices are subjective as to what is defined as 'Best' without the rigor of oversight, peer review and continual improvement. But Best Practices are not a substitute for documented processes of a business and engineering organization. By stating organizations follow 'Best Practices' they may be able to claim a routine and repeatable set of activities, that actually shortcut the documented and measured process. If Best Practices are based on organizational policies and plans or augmented by documented processes as defined by sound Systems Engineering, they will probably work. But then why not just write a process?

3.4. Summary - Putting It All Together

To wrap up Section 3 I've created a simple illustration of how the processes and the gates work together. You can see the sequence as illustrated in **Figure 3-12** of the processes as you would expect in chronological sequence, starting with the process 'Start Project'. The output of the process is the input to the next process, and some processes are initiated within other processes. Other processes are repeated (iteration), and yet other processes span the entire lifecycle as illustrated in the upper right corner. Referencing process are expected in SEMPs and PMPs. Online documents allow the use of URLs to move quickly through the lifecycle, and ensure the documents are current.

Figure 3-12 Simple Process Flow



A Rant

I will challenge any team or person who states **they do not have processes**, or **they don't need processes**. Everyone has a process, so the questions are 'Are processes consistent across the team? Are processes performed the same by each person every day? Do you have one process, and others have another process, and do you expect everyone else to go out of their way to do it your way?'

4.0 Say What You Mean, and Mean What You Say

Systems Engineering is a very abstract discipline in which complex problems and ideas are captured and must be expressed to other engineers, those in support functions and to managers. Additionally, these ideas can become lost or confused if everyone expressed these problems in their own language. Could you imagine building a house from different types of drawings, where the structure is represented one way, the plumbing another, and electrical yet in another way? How would one capture what the customer desires when they may not fully know themselves, or to ensure all the pieces connect, or the Systems Engineer starts assembling the system only to find out they don't fit? It happens! Thankfully, there are the Languages of Systems Engineers, or the Language of Engineers in general. Not only do these languages convey complex ideas, but they are also thinking and problem-solving tools that allow collaboration, modelling and simulation, measurement of the results, exposure of risks and conveying these ideas to one another to allow the most suitable way ahead. The first part of this section (4.1) will present only a few of these languages, all presenting the various perspectives that are defined in the **Enterprise Architecture**. Section 4.2 will include the main organizations that create and maintain these standards, the languages, and methods.

If there is a problem with these languages, standards, and frameworks it is that there are many from which to choose, and if one engineer is fluent in one, they may not be fluent in another. The other problem is that organizations tend to make their own flavor from these languages, almost as if it's a dialect. I find this a minor problem though, compared to an organization that uses no framework, standard or common language. Some methods or frameworks do well in capturing, synthesizing and decomposing requirements and in creating an architecture, and even Verification and Validation, yet others partially apply any. Successful Systems Engineer use these methods and models singularly or together, drawing on the strengths of each. These traits are pointed out.

4.1. Languages for Systems Engineers

The common languages discussed are.

- Department of Defense Architectural Framework (DoDAF)
- Integrated Definition Methods (IDEFx) – six diagrams – block flows, functional, data, ontology to name a few
- Uniform Markup Language (UML) – 15 drawings for Software Development
- Systems Modeling Language (SysML) – nine drawing types for Systems Engineering and Software Development.
- U.S. Military Standards – MILSPECs, Handbooks, Standards, Performance Specs, Detail Specs.

4.1.1. Department of Defense Architectural Framework (DoDAF)

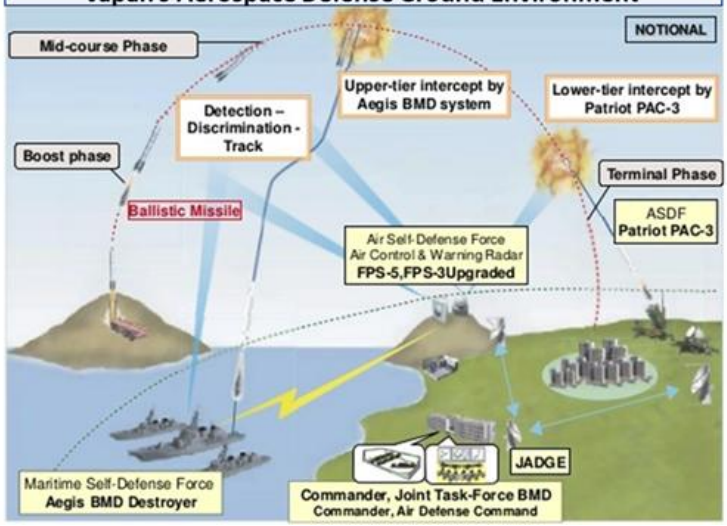
Department of Defense Architectural Framework (DoDAF) includes multiple Views, such as Operations, the Network, and Data, etc. (DoD CIO, 2010). This extensive selection of documents is illustrated in **Figure 4-1.** System Engineers will often use an Operational View (commonly called an OV1) which is simply a drawing of the operational environment. **Figure 4-2** called Japan's Aerospace Defense Ground Environment Operational Architecture is an example. One may view this drawing and quickly understand the purpose and function of the system. Many Systems Plans, Architectures and briefings contain an OV-1, unbeknownst to the viewers that they are viewing a DoDAF drawing. There

are other views such as OV-2 Operational Node Connectivity, System Interface IV-1, and more. Besides linking between system and enterprise, DoDAF helps stakeholders have a consistent understanding of the system and will help find the salient requirements of a system and ideas on how to satisfy those requirements.

Figure 4-1 DoDAF Drawings (V1.5 & V2.0)

DoDAF V1.5 \ DoDAF V2.0	Operational Viewpoint	Systems Viewpoint	Services Viewpoint	All Viewpoint	Standards Viewpoint	Data & Information Viewpoint
AV-1				AV-1		
AV-2				AV-2		
OV-1	OV-1					
OV-2	OV-2					
OV-3	OV-3					
OV-4	OV-4					
OV-5	OV-5a, OV-5b					
OV-6a	OV-6a					
OV-6b	OV-6b					
OV-6c	OV-6c					
OV-7						DIV-2
SV-1		SV-1	SvcV-1			
SV-2		SV-2	SvcV-2			
SV-3		SV-3	SvcV-3a, SvcV-3b			
SV-4a		SV-4				
SV-4b			SvcV-4			
SV-5a		SV-5a				
SV-5b		SV-5b				
SV-5c			SvcV-5			
SV-6		SV-6	SvcV-6			
SV-7		SV-7	SvcV-7			
SV-8		SV-8	SvcV-8			
SV-9		SV-9	SvcV-9			
SV-10a		SV-10a	SvcV-10a			
SV-10b		SV-10b	SvcV-10b			
SV-10c		SV-10c	SvcV-10c			
SV-11						DIV-3
TV-1					StdV-1	
TV-2					StdV-2	

Figure 4-2 Operational View 1 (OV1) of Japan's Aerospace Defense Ground Environment



DoDAF Details may be found at https://dodcio.defense.gov/Library/DoD-Architecture-Framework/

4.1.2. Integrated Definition Methods (IDEFx)

Integrated **DEF**inition Methods (IDEF*x*) (IEEE, 2020) is a set of modeling languages that aid the Systems Engineer, and mainly the Architect, to flush out requirements, synthesize and model into a design. IDEF may be considered a requirements tool plus an architectural design tool because of the iterative nature. IDEF came about in the 80s from a set of Air Force projects. IDEF allowed heterogenous modules in an architecture so differing systems could draw on those modules as needed. IDEF's notations of arrows, lines and squiggles have specific meanings and the System Engineer must learn their meaning. These symbols are illustrated in **Figure 4-3**, along with a sample IDEF3 and IDEF1. Once IDEF is understood, the use of IDEF saves time, allows for reuse on other systems, and reduces errors. IDEF is consistent to Enterprise Architecture in that there are stakeholder perspectives although it may not specially call them out. A few of the IDEF*x* type drawings are:

- IDEF0 - Function Modeling – this is the most common IDEF drawing, and often the only IDEF drawing used. It includes the inputs as Controls, Inputs and Mechanisms to go into the Function, and the resulting output. That output leads to another function block with the same type of inputs. IDEF0 inherently illustrates a decomposed and time-based set of functions. An example of an IDEF0 – Functional Modeling Diagram is illustrated in **Figure 4-4.**
- IDEF1 - Information Modeling – illustrates the structure, relationship and semantics of information.
- IDEF1*X* - Data Modeling – modeling of meta-data mainly for IT
- IDEF2 - Simulation Modeling – interfaces but later repurposed for timing and synchronization of system entities.
- IDEF3 - Process Description Capture – relationships between objects, entities.
- IDEF4 - Object Oriented Design – for component-based systems, to enough detail to allow development.
- IDEF5 - Ontology Description Capture – objects, their naming conventions, and relationships.
- IDEF6 - Design Rationale Capture – ensure purpose in creating enterprise-oriented systems by asking questions about the design, alternatives, and why alternatives are selected.

Examples provided below are not clear and easy to read but are presented for you to get at least a notion of IDEF. More detail on IDEF*x* may be found at https://www.idef.com/idefo-function_modeling_method/

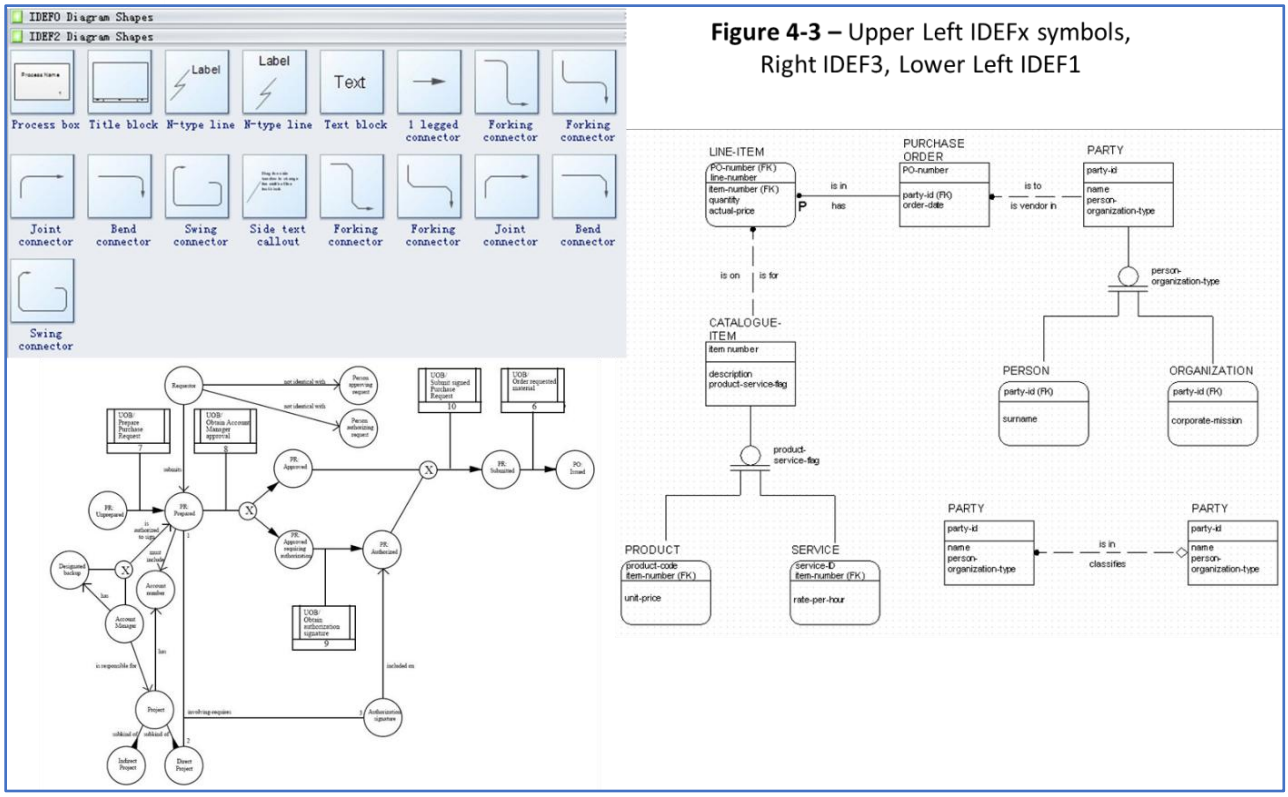
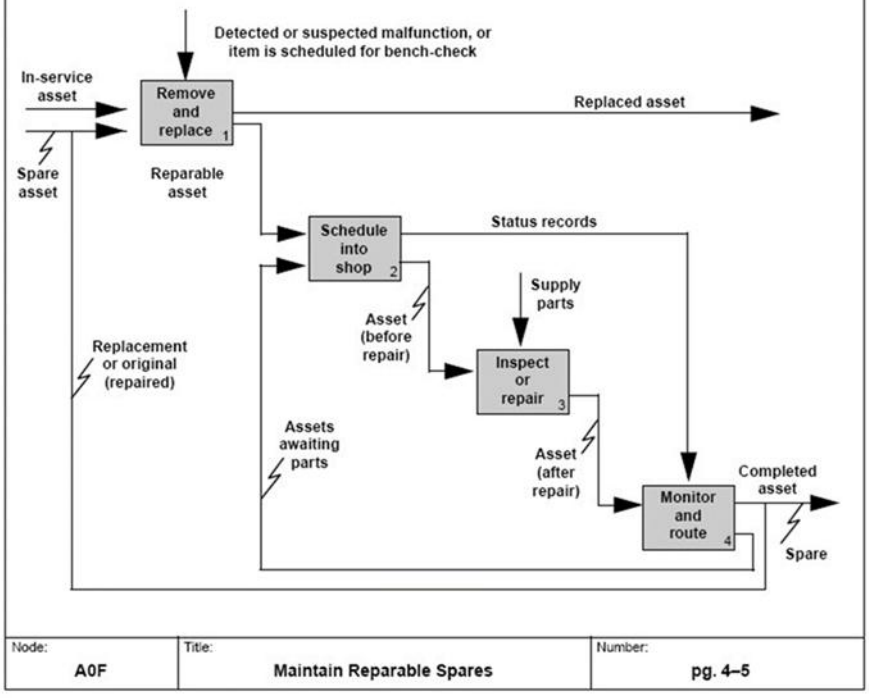


Figure 4-3 – Upper Left IDEFx symbols, Right IDEF3, Lower Left IDEF1

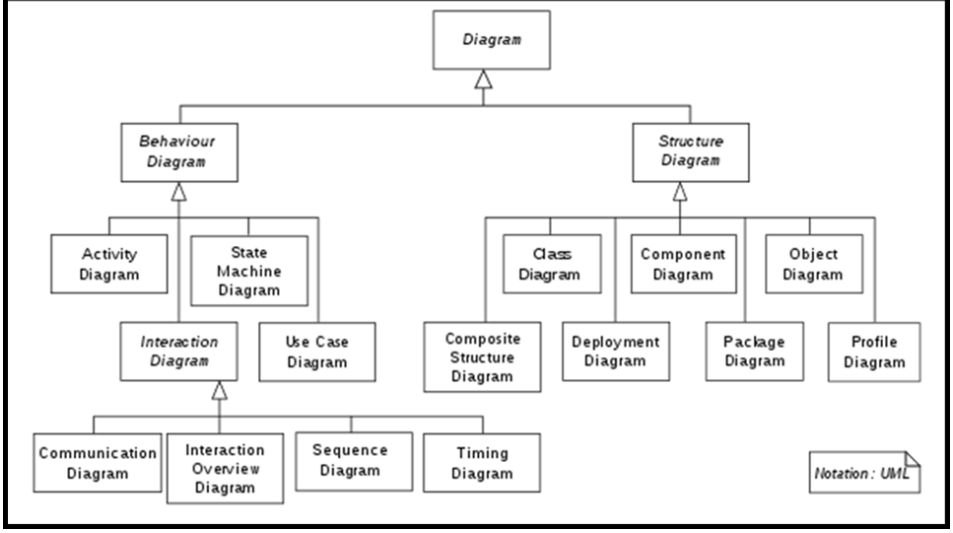
Figure 4-4 IDEF0 Functional Model – Illustrates Flow – Inputs, Outputs, Controls, Mechanisms



4.1.3. Uniform Markup Language (UML)™

Rational Software was the origin of UML (UML, n.d.)in 1994, when a small team created a standardized modeling language for Software Engineering. The focus at the time was more object-oriented programming. UML is a graphical representation providing a set of diagram formats and works as a model that has two different views 1) static with objects, attributes, operations and relationships, and 2) Dynamic (or behavior). **Figure 4-5** illustrates the two main types of language with a total of 14 diagrams. (left is Behavior, right side is Structure). UML has matured over the years from version 1.X to today's UML Version 2. The UML standard is maintained by the Open Management Group (OMG).

Figure 4-5 Uniform Markup Language has differing representations depending on one's perspective

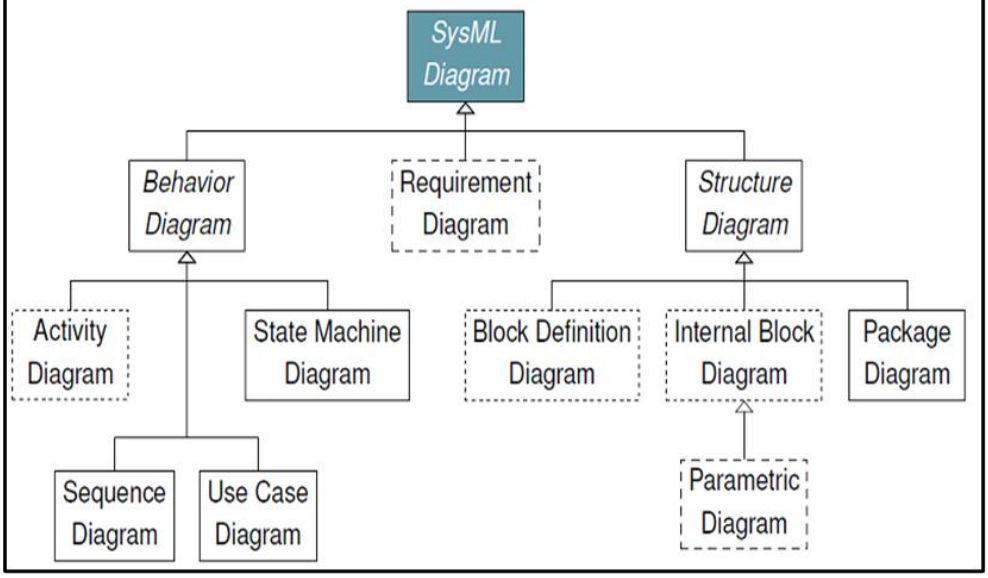


4.1.4. Systems Modeling Language (SysML™) (SysML, n.d.)

SysML grew out of UML as general-purpose modeling (simpler) language for not only Software Engineering but also for Systems Engineering. It is used for requirements analysis, design and for Verification and Validation. The specification for SysML is maintained by the International Council on Systems Engineering and the Object Management Group. SysML has nine diagrams that show a range of elements and activities to include Block definition diagram, Internal block diagram, Package diagram, Use case diagram, Requirement Diagram, Activity diagram, Sequence diagram, State machine diagram, Parametric diagram. **Figure 4-6** illustrates the layout of SysML.

- SysML is useful for Human Factors and Usability Engineering of people and machine interface. Often, those using SysML observe how people work, then document, draw and determine how a human function may be performed more efficiently and safely.

Figure 4-6 Systems Modeling Language – similar to UML but for Systems Engineers



For additional reading on UML and SysML go to
https://en.wikipedia.org/wiki/Object_Management_Group

4.1.5. U.S. Military Standards

When I think of architecture and design, I usually fall back on Military Standards, or specifically Military Specification (MILSPEC). (Wikipedia, 2020) Before IDEFx and DoDAF, and the other abstractions, there was MILSPEC. Upon research I found over 80 documented MILSPECs or MIL-Handbooks for differing topics, including designs. I would encourage you to survey this list so you'll have a notion of the list contents and where to go for more information.

It may be found at https://en.wikipedia.org/wiki/United_States_Military_Standard

Many of the standards for System Engineering have been subsumed by more commercial and open standards such as ISO 15288 and INCOSE. But the U.S. Military Standards are alive and well.

4.1.6. Architectural Description Language - ISO/IEC 42010

> 'The nice thing about standards is that there are so many of them to choose from.' – *Andres S. Tanenbaum, Professor of Computer Science, Vrije Universiteit, Amsterdam*

(ISO/IEC/IEEE 42010:2011, 2017)is an architecture description composed of four viewpoints plus concerns (note consistency with DoDAF), and a common terminology to document and convey the design. The four conceptual models are;

- Foundations – composed of about 20 concepts ranging from differing architectural views, and models, even correspondence.
- Architectural Description – includes the architectures and identifies stakeholders, and more viewpoints
- Architecture View – subset of Architectural Description – explains one of each viewpoint, stakeholders.
- Architecture Viewpoint – Sub-set of Architectural Description – of one or more stakeholder concerns, one architectural view and one or more models
- Concern – concerns are derived or expressed from one of the previous

More information about ISO/IEC may be found at https://en.wikipedia.org/wiki/ISO/IEC_42010

4.1.7. C4 Model

C4 is especially useful for decomposing requirements, the functional and system architecture, and interfaces (C4 Model, n.d.). C4 too, applies standard notation and syntax to ensure the abstractions are documented and communicated. C4 is software and IT focused, and thoroughly provides differing aspects and detail of an architecture as previous discussed. The four levels are;

- Level 1 Context Diagram – the Big Picture, used for non-technical people.
- Level 2 Container Diagram – a separate runnable/deployable unit that executes and stores data.
- Level 3 Component Diagram – portions that make up the container, such as security for authentication, database, or a single application.
- Level 4 Code – uses Uniform Markup Language (UML) to illustrate various types of diagrams of the system and its parts.

C4 includes a set of augmenting diagrams such as Landscape and a Dynamic Diagram used to offer more detail and understanding.

More information may be found at https://c4model.com/

4.1.8. Summary - Systems Engineering Languages

This list includes only a few of the methods, models, frameworks – the Language of Systems Engineers. And as pointed out, some are useful in almost the entire lifecycle and others are useful in only a portion of the lifecycle. These methods, models and frameworks may also be used together on the same system as long as it does not cause a conflict (i.e. C4 and SysML). Additionally, all capture the concept of a perspective (View and Viewpoints) as outline in the Enterprise Architecture. **Table 4-1** offers a summary of the ones we discussed.

Table 4-1 A Few of The Systems Engineering (SE) Languages and their Features

Method	Used For	Works Well With Y-Works Well, N=Doesn't Lifecyle Phase: Requirements	Architecting	Decision	Develop	VER	VAL
DoDAF	All Systems	Y	Y	N	N	Y	Y
UML	Software	Y	Y	N	N	N	N
SysML	All Systems	Y	Y	N	N	N	N
MIL-STD 498	Software	Y	Y	Y	Y	Y	Y
C4	Software but works all SE	N	Y	Y	N	N	N
ISO/IEC 15504	SW Dev. Stds & Framework	Y	Y	Y	Y	Y	Y

4.2. Guiding Bodies and Standards Bodies

Someone must establish and document a standard of the framework, languages, methods and models. Thank goodness, there are plenty – so we can choose one. The System Engineer should not only be aware of these organizations, but work actively to stay abreast of changes, to participate in the professional groups and to help influence their activities.

4.2.1. Defense Acquisition University (DAU)

DAU is the premier acquisition school in the United States binding the rigors of acquisition of systems with program management, Systems Engineering to mission and operations. DAU is a degree offering university available for employees associated with the United States Government. It also offers many short term and online classes. DAU is headquartered at Fort Belvoir and has graduated over 181 thousand students (DAU, 2020).

4.2.2. Project Management Institute (PMI [(T)])

PMI[(T)] is a not-for-profit organization that develops standards, conducts research, provides education, publish Project Management guides and documents, offers networking opportunities for the purpose of quality Project Management and associated disciplines (PMI, 2020). PMI offers professional certifications including the Project Management Professional (PMP). PMI publishes several guides including the **Project Management Body of Knowledge**. (PMBOK). The PMBOK standards have been adopted by the International Standards Organizations (ISO), and the American National Standards Institute (ANSI). PMI was established over 55 years ago and now has over 1.5 million certified professionals in over 300 Chapters worldwide.

The nine certifications one may attain through the PMI are:

- Project Management Professional (PMP(R))
- Certified Associate in Project Management (CAPM®)
- Agile Certified Practitioner (PMI-ACP®)
- Program Management Profession (PgMP(R))
- Disciplined Agile Scrum Master (DASM)
- Portfolio Management Professional (PfMP(R))
- PMI Professional in Business Analysis (PMI-PBA)
- PMI Agile Certified Practitioner (PMI-ACP(R))
- PMI Risk Management Professional (PMU-RMP(R))
- PMI Scheduling Professional (PMI-SP(R))
- Portfolio Management Professional (PfMP®)

More information may be found at https://www.pmi.org/

4.2.3. International Council on Systems Engineering (INCOSE) (T)

INCOSE is a not-for-profit professional association offering a framework and methodology used for Systems Engineering (INCOSE, 2020). INCOSE hosts professional development and certifications based on years' experience, written or verbal exam. INCOSE has 18,000 members of which almost 4200 hold one of three certifications, and there are more than 65 chapters worldwide.

More information may be found at https://www.incose.org/

There are three progressive levels of certifications provided by INCOSE. These are:

- ESEP(R) – Twenty plus years of documented experience and supported by three references, sustained technical leadership, plus meet an interview board.
- CSEP(R) – five years of documented experience across multiple area and supported by three references. Pass the written exam.
- ASEP(R) – pass a written exam, zero years of experience but should be working as an SE. May be an engineering student.

INCOSE has developed a solid framework and is consistent to industry standards, and in particular ISO/IEC 15288:2008. This book and training are based on INCOSE.

4.2.3. International Standards Organizations (ISO) - 9000 Quality Management System

ISO a family of quality management system standards to help organizations meet the needs of their customers, and to ensure compliance to statutory and regulatory requirements (ISO, 2020). The areas of ISO-9000 include requirements, design and development planning, communication, documentation, and record keeping, customer satisfaction and continual improvement. AS 9100 is the Aerospace application of ISO - 9000, and has standards that are more stringent, such as reverse tracing of requirements and corrective action of faulty systems. Organizations that are ISO-9000 Certified or AS 9100 certified

have achieved a level of performance that proves their ability to deliver high quality products or service. Certification is accomplished by an approved external 3rd party auditor. There is no formal recognition of "ISO Compliant" as sometimes advertised by organizations. ISO Compliant usually means the organization conducted a 2nd Party (internal audit) and determined by themselves, that they were following all the standards of ISO. PMs, Systems Engineers and customers and government contracting organizations will likely experience contractors who have ISO certifications. Knowledge of this subject may prevent issues from occurring on your system or contract.

4.2.4. Capability Maturity Model Infrastructure (CMMI)

Carnegie Mellon is the origin of CMMI with their initial focus on software development, but it branched out into Systems Engineering and operational disciplines (CMMI Institute, 2020). CMMI offers a maturity rating that is assessed and awarded by a 3rd party auditor. It takes considerable investment to start this process but once in place, efficiencies are realized in decreased waste, and continual improvement.

The model has two approaches:

- Maturity Model – a common maturity across all elements of organizations activities. Level 1 and 2 normally mean inconsistent and unmeasured processes. Level 3 means repeatable processes are in place and in use. Level 4 means the processes are being measured to find weaknesses, and potential improvements. Level 5 is the highest maturity, which means processes are measured and continually improved and optimized. Level 5 focuses on preventing problems through predictive analysis and causal analysis. Organizations should consider the cost benefit analysis of achieving the more costly Level 4 and Level 5, and the virtually flawless systems it is expected to deliver (Space Systems, Command and Control). Lever 3 may be suitable for those organizations not striving for perfection.
- Reference Model – when only several of an organization's processes have shown to be mature. Other processes have not yet been proven to be mature.

4.2.5. Institute of Electrical and Electronics Engineers (IEEE)

IEEE is a worldwide standards-setting body composed of representatives from various national standards organizations (IEEE, 2020). IEEE promotes worldwide proprietary, industrial, and commercial standards.

4.2.6. American National Standards Institute (ANSI)

ANSI is an American organization through consensus, establishes standards related to software, information technology, and telecommunication and coordinate these standards with international bodies, so American products may be used internationally (ANSI, 2020).

4.2.7. International Telecommunications Union (ITU)

ITU is an organization chartered by the United Nations that established telecommunication standards allowing systems to interoperate (ITU, 2020).

4.2.8. Information Technology Service Management (ITSM)

(Formerly Information Technology Infrastructure Library (ITIL)) – Framework to design, develop, deploy and operate information technology systems (Wikipedia ITSM, 2020). Includes service design,

Service Level Agreements (SLA), capacity, continuity, transition, security, configuration, and other areas. PMs, Systems Engineers and Government employees will likely come across vendors who use ITSM and a basic understanding of ITSM is essential.

4.2.9. Security Management - Cybersecurity

I will discuss two frameworks to Security Management, both of which are like the other. In fact, they share knowledge and ideas. One is for U.S. Government Systems and the other for commercial systems. Information security has a far greater reach than twenty years ago. It not only applies to ground-based IT Systems but extends to the aerospace, autonomous vehicles, artificial intelligence and robotics, utilities, transportation infrastructure, wireless and virtually any object that needs confidentially, integrity and robustness. Let's look at these two standards.

4.2.9.1. ICD 503 Intelligence Community – Information Technology Systems Security Risk Management – Certification and Accreditation

ICD 503 includes a description of the Risk Management Framework (RMF) which is Six Step Process to identify, mitigate, design, build and test, then monitor risk associated with Confidentially, Integrity, and Reliability. Although RMF applies to government systems or systems containing national security information, the principles apply to commercial and private systems as well. ICD 503 is part of cybersecurity. ICD 503 is the 'bread and butter' for the Information Systems Security Manager (ISSM).

4.2.9.2. ISO/IEC 27000 Series – Information Security Management Systems

ISO/IEC 27000 was established in the 90s and has been continually updated and approved. It provides a framework composed of a set of the 27000 series that cover a common set of terms, security techniques, code practice, monitoring, measurement, analysis and evaluation, audits on any platform having IT. ISO applies a three-staged maturity model with stage 1 as preliminary and informal review of security practices of an organization, Stage 2 more detailed and formal compliance audit, and the Stage 3 – the Ongoing follow-up and reviews.

4.3. Summary of Section 4

It's important, and in fact it's completely necessary for Systems Engineers and engineers in general to work and communicate in a common language. In Section 4.1 I discussed a few of these languages and how they are used, and their traits and capabilities. Not only are these necessary to communicate with stakeholders but necessary in discovering and solving the problem. These languages, frameworks, methods and models are created and managed by organization called out in Section 4.2. There are many of these organizations because some provide guidance, others are standards bodies, and all covering a wide range of engineering domains.

5.0 Supporting Skills – The Tricks of the Trade

Throughout this book I referred to more skills, techniques and even 'tricks of the trade'. In this section I will touch on these related skills because the Systems Engineer needs help from not only the domain engineers, but also subject matter experts more directly related to systems engineering. I will discuss Quality and their role, which I called out through all the documented processes.

5.1. More Skills

5.1.1. Systems Analysis

Systems Analyses is the study and documenting of how systems function. Systems Analyst learn and discover methods of performing these tasks, within an automated system or a manual system, to include performance, throughput, robustness, failure mode and recovery, load balancing. This enables the design of a system, or to improve existing systems.

5.1.2. Operations Research

Operations Research (OR) is a skill that is rigorously applied on defense systems, due to their critical nature. OR analytically solve advanced problems, capability development, mission assurance – statistical analysis, throughput, and load balancing, max/min of a range of data. Over recent years, and in the coming years I see great opportunity for the application of Operations Research in the commercial and private industry.

5.1.3. Usability Engineering/Human Systems Integration

Usability Engineering/Human Systems Integration are the study of how people and machines interface so effective and seamless operations are created and built into the system. This includes colors and indicators, alarms, and placement of operator controls. Those performing Usability Engineering and Human Systems Integration will find SysML quite useful in preparing the unstated but observed requirements of an operator.

5.1.4. Human Factors Engineering

Human Factors Engineering is the psychological and physiological principles to engineer and design products, processes, and systems – to reduce human error, increase productivity and enhance safety and comfort – human interfaces with a device.

5.1.5. Environmental, Occupational Health, and Safety (EOHS)

EOHS is a discipline and a profession that evaluates the conditions of a working environment, then creates and implements work practices. These practices are compliant to Federal Requirements and ensure safe and healthy working environment for the employees, customers, visitors, and that don't cause harm to the community.

5.1.6. Thermal Engineering

Thermal Engineers design and assess the thermal energy created and omitted by a product or a system, at various operating capabilities. Designing the system for most effective performance in conjunction with thermal generation is of most importance. These types of systems are often used in closed and restricted (small space) areas or in aerospace systems where power is budgeted, and the omission of heat is restrained.

5.1.7. Information Systems Security Managers (ISSM)

ISSMs are especially important experts for government systems, who manage the development, testing and operations of security requirements and features as outlined the Risk Management Framework and ICD 503 Information Technology Systems Security Risk Management, Certification and Accreditation and ISO/IEC 27000 Information Security Management Systems (ISMS) (McConnell, John (V Adm), 2008), (ISO/IEC, 2020). The role of ISSMs may be found as cyber security specialist, security engineers in civilian and commercial industry, and are of growing importance as systems become more integrated, and as threats increase. Much or the work performed by ISSMs is defined in Section 4.2.9. ISSMs work at the Program Level or higher in the organization to provide the needed clout, and the autonomy to carry out their job.

5.1.8. Critical Support

Others who support the project, system and the business but are not detailed in this book are Financial Analysis, Contracting, Cost Estimators, Accounts Payable, Account Receivable, Supply Manager and other business specialist. Without these, the project and the system would fail.

5.2. Quality Management

This section is derived from ISO 9000 Quality Management (ISO 9000, 2015).

You will have noted the 'omnipresence' of quality while discussing the processes and as outlined in Section 3.1 Why Processes. I waited until last to touch on Quality not because its least important but because of its unique role in a systems organization.

5.2.1. Everyone Wants Quality

The Executive Management Team will have a Quality Policy that is known and understood by all stakeholders. Project and systems team members are responsible for quality from the start of the project to the very end of the project. This includes quality processes, directives, tools, and labs. Quality is not simply the Quality Manager's job but instead the Quality Manager performs periodic checks on the quality of processes, tools, labs, personnel training, to ensure compliance.

5.2.2. Quality is Defined by the Customer

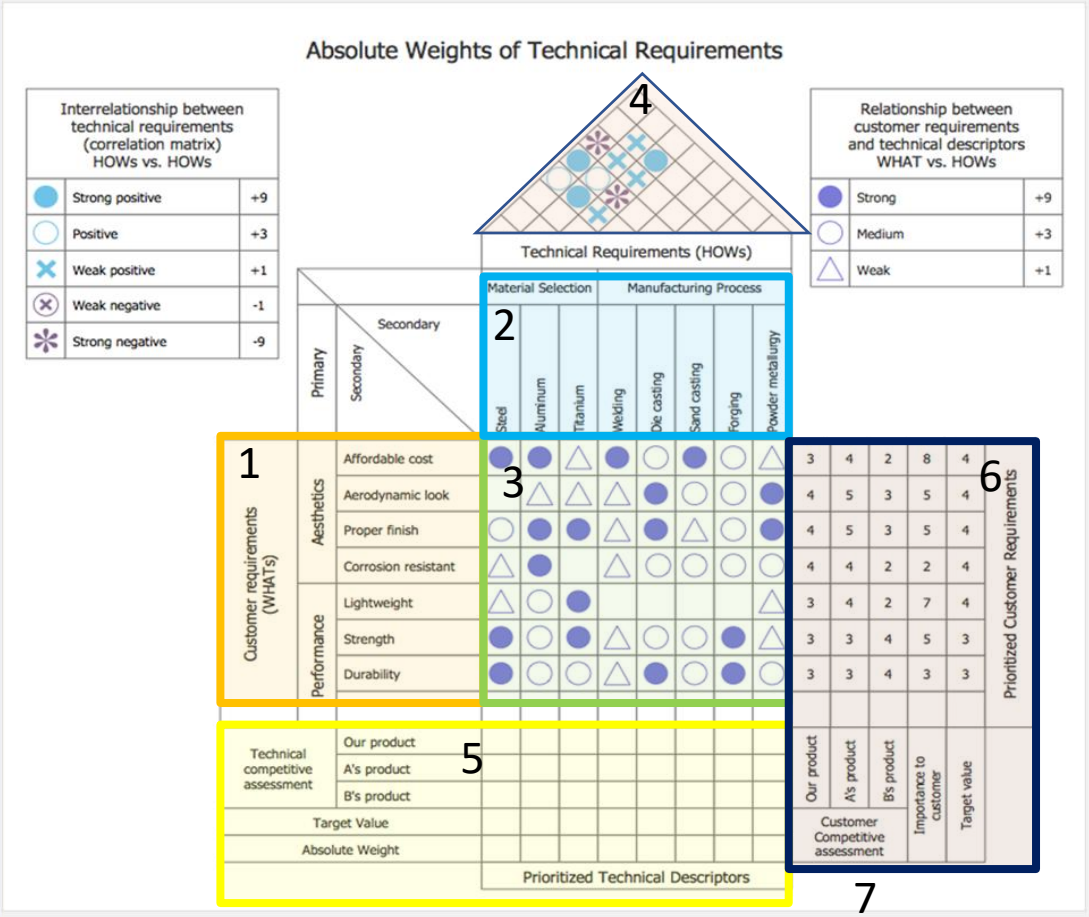
It meets quality standards when the customer says so – sounds harsh but it's true. That's why Systems Engineering must always have the 'voice of the customer' in their ear, and to keep the customer involved as much as the customer's desire. Quality Managers review records (as proof of compliance), conduct audits and trace to the source to ensure accuracy and completeness. These records are available for the customer to review if they choose.

- Quality Control verifies that a process was performed, or a part was created after its completed.
- Quality Assurance checks a process or activity to verify that its going (present tense) as expected.

5.2.3. House of Quality – The Voice of the Customer

Few tools, or methods exist to the extent of the House of Quality, or Quality Functional Deployment (QFD). This is a multi-dimensional model that is used to align the importance of customer needs to what an organization can best deliver, to what the competitors can deliver. It may be used to evaluate a service, product, system, or all of this. It looks complicated but simply takes a few minutes to figure it out. It starts with the 'Voice of the Customer', which are not only the need and requirements but the tacit concerns they will have. It links the most important desires to a feature in the system and

Figure 5-1 House of Quality
Aligns the Voice of the Customer to Systems Engineering



1. Far Left – add Voice of Customer – what they want
2. Control Factors – what we (the provider/seller) controls
3. Center Box - Relationship between Customer Requirements and Technical Descriptors – this is what the customer wants (WHATs) compared to what you can deliver (HOW) - create and agree to a value – see upper right corner of illustration
4. Top – The Roof – cross ref each in the cell that they meet and rate (each control factor) and scale how well they compliment (the relationship) each. See scale in box on uppe left
5. Importance Factor (along bottom) multiply each cell in center to weight and put the value on Importance Factor Row – the higher the most you must focus on because that's what you can control, and what is important
6. Competition Rating – rate how your competition does in each of the 'Voice of the Customer' Create a rating such as 1 fail, 2, marginal, 3 Sat, 4 Excellent, 5 Outstanding
7. You may or should have multiple competitors

the related concerns. **Figure 5-1** illustrates a simple **House of Quality** with color coded boxes to guide the instruction. It states the requirements on the left (Y Axis), such as affordability, aerodynamic look and so on. Where these features are built in are indicated by a solid circle, or a not filled in circle, or an X as indicated in the upper left corner. These attributes prioritized far right axis and those features are built in accordingly. You may never see the House of Quality but if used properly it is a powerful tool to link the requirements and 'ilities' to what the customer actually receives. The assessment of your capabilities, the competition and what the customer needs are subjective, open for debate but don't get hung up on specifics or one issue – debate and come to a consensus. You may consider a customer survey to help strengthen your objectives. The result will give you a better idea on your strengths and weaknesses, and how to beat a competitor to deliver value to a customer.

More information on the House of Quality may be found here https://asq.org/quality-resources/house-of-quality and https://www.youtube.com/watch?v=u9bvzE5Qhjk

5.2.4. Just Give Me Quality!

It's not unusual for those procuring a system to not understand why a special person or group is responsible for quality. Many people often assume 'why would you deliver something without quality?'. Pioneers of Quality such as W. Edwards Deming and Joseph M. Juran noted the importance of an independent third party to ensure process and product Quality Assurance and Quality Control prevent variance, provide oversight and for those performing the process are astute to the accuracy of their work (Deming, W. Edwards, n.d.).

5.2.5. Quality Manager

Although quality is everyone's job, a Quality Manager and staff are usually assigned to engineering and customer-oriented organizations to ensure People, Process, Tools are meeting standards to deliver as specified. The Quality Staff must not work for the project or the program manager because of the bias and influence they may have from the program. The Quality Manager and the staff work for a Senior Manager from the organizational level. This is like Independent Verification and Validation (IV&V), or security staff which also must maintain autonomy from the program. Quality Managers are not testers, they don't develop, they don't manage requirements. Quality managers perform audits and related work as outlined in the International Standards Organization (ISO) or Capability Maturity Model Integration (CMMI) or similar quality framework. Performing any of these activities while performing as a Quality Manager is a conflict of interest.

5.3. Model Based Systems Engineering

Model Based Systems Engineering (MBSE) enables Systems Engineers to model and simulate domains of systems and to re-use artifacts of these domains to integrate and build systems. For these domains, the function, input, output, and interfaces are already defined. The emergent behavior from that domain is known, and the larger system has a need to tap into and use that emergent behavior. Through modeling and simulation, the emergent behavior becomes an input to a needed system until suitable integration is completed. This approach reduces the overall work expended on the Systems Engineering lifecycle and reduces risk. SysML and UML are very useful to MBSE. MBSE is an initiative from an **INCOSE Working Group** in 2007 (Estefan, 2008).

5.4. Techniques

A few techniques applied by the Systems Engineer and the supporting specialist are touched on next.

5.4.1. Continual Improvement

Continual Improvement was discussed as a process, but I reserved this section for the detail on how this activity is performed.

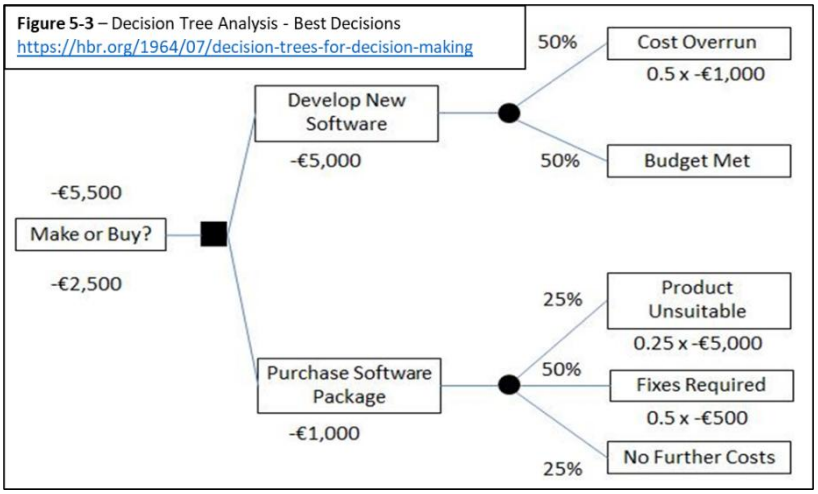
Figure 5-2 Continual Improvement

As illustrated in **Figure 5-2,** People, Process and Tools may be improved when a noted defect or an opportunity is discovered (ASQ, 2020). Continual Improvement starts with **identifying** potential efficiencies or corrections through metrics disclosing bottlenecks or single points of failure, reported defects, extended process time, risk, and potential opportunities. The change is **planned** by ensuring the change and its desired outcome are understood, designing, and modeling the change, and by creating an awareness or training plan (depending on scope) for the change to ensure stakeholders are prepared. The change is **executed**, at each step, monitoring progress, 'unfreezing' the current method, making the change, then 'refreezing' the new method. Have a fallback plan in case it does not work. Finally, **review** and monitor the change to verify it is actually working as planned. This process (called 'closed loop') must be managed to ensure only viable changes. As illustrated in the figure, the red fringe of the cycle is unmanaged or variance in the processes, introducing risk. Technology and policy improvements may offer opportunities. Each person on the project and systems team, plus the user community, will be part of continual improvement. Someone is assigned to manage this process.

A common approach to continual improvement is **Six Sigma**, which was invented by Motorola and applied by General Electric starting in the 90s (Wikipedia Six Sigma, 2020). The concept is to allow as little variance in a process and the product as possible so that 99.99966% of processes and product are identical. Detecting variance is especially important in repeatable processes that are found in a manufacturing activity, where variance can easily creep into a system before it is detected. If there is variance the process and product should be thoroughly examined to discover the anomaly, and corrective and preventive action applied. The tenants of Six Sigma are the defined hour goals, know how to measure and analyze to improve it, and control the process to reduce and eliminate variance. Many people have become certified in Six Sigma. https://en.wikipedia.org/wiki/Six_Sigma.

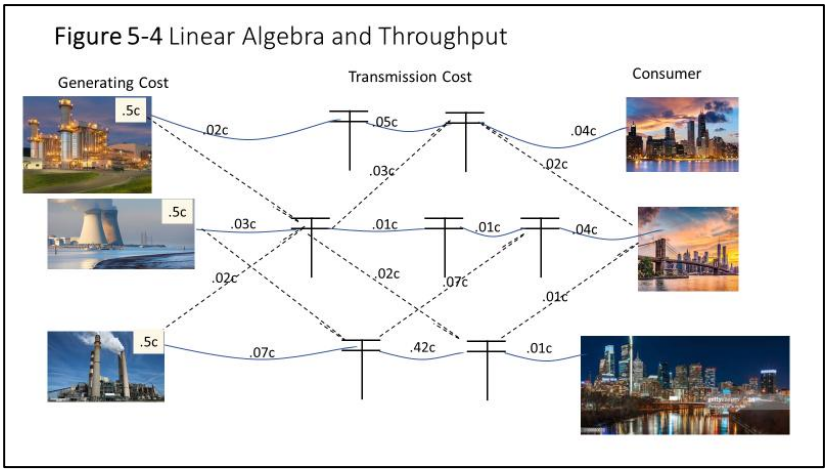
5.4.2. Decision Tree Analysis – probability of a chain of events occurring

The Systems Engineer cannot 'predict the future', or can they? No, but a Systems Engineer and Systems Analyst can use a Decision Tree to anticipate the most probable outcome of a series of events. Its best done with a small team of subject matter experts for making a series of decisions on a particular course of actions. Decision Tree Analysis is very useful in deciding to build or buy a system, or to venture into a new business area, or to assess a series of opportunities that may be fraught with risks. Often the result is different then what would have otherwise been expected because it considers the unintended consequences. The use of Decision Tree Analysis requires a decision criterion before the process starts, discovering as many alternatives as reasonably possible, learning of potential outcomes based on the past performance, technical limitation, governance (remember the business lifecycle), and removing emotion from the analysis, which I acknowledge is not easy. A simple example is provided in **Figure 5-3** in which an organization is considering a new software solution. (Magee, John F., 1964) Should they make or buy? If they buy the product, it may not be suitable (25%), or it may require fixes (50%), or it may actually work (25%). If they develop it themselves there is a 50% chance of an overrun. The assumptions are from experience and market information of the success of a particular provider. This method can be flawed if the inputs are flawed (remember the Section on Processes?).



5.4.3. Linear Algebra

Linear Algebra is an especially interesting solution to problems that often uncover counterintuitive issues. It peels back the layers of a problem that one may not have known, discovering inefficiencies, bottle necks and single points of failure. Linear Algebra can become quite complex, but it leads to machine learning and predictive analysis. The example illustrated in **Figure 5-4** for the usage of Linear Algebra are given three electric power plants, and three sets of transmission lines (some cross to other lines), and three cities, which would be the most efficient for the cities? Assume the power is sold as different cost by each plant and the transmission of power is a different cost, as listed per megawatt hour. It may also be reversed so the power companies can



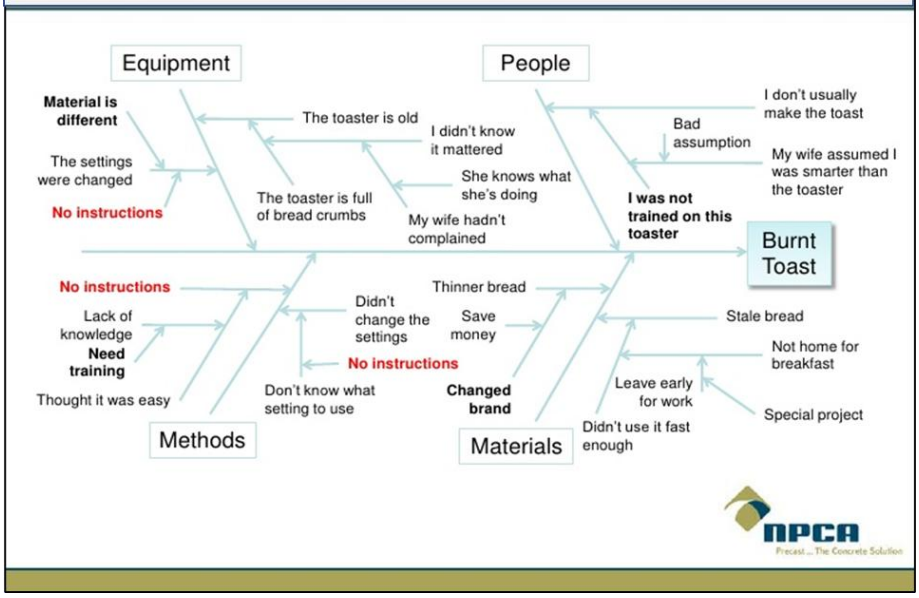
determine how maximize their profit. Other uses for Linear Algebra are throughput of data on a system, water transmission and a shift schedule in a retail shop, auto or aircraft traffic volumes, deconflicting of events, data, and network load balancing.

5.4.4. Root Cause and Corrective Action (RCCA)

RCCA is used to find the source of a problem by tracing it back, step by step, to the cause, then determining the best form of correction.

A 'Fish Bone' Chart is an illustration of applying an RCCA. As illustrated in **Figure 5-5**, it is used to break a problem down into smaller portions until the one, or two causes of the problem are discovered. The RCCA process should be led by an RCCA coordinator who knows how the process works and has the discernment to focus on the problem while leading a small team. The team should consist of a small group of those who have direct knowledge of the problem and potential causes. If too many people are involved, the group becomes unwieldy, will lose focus and some valued participants may shut down and say very little. Also its best to have participants that are not emotionally committed or are too close to the problem.

Figure 5-5 RCCA – How Did the Toast Get Burnt!



An example for Burnt Toast is describe below, with the corresponding diagram (NPCA, 2010). RCCA should be used in conjunction with the continual improvement cycle described in Section 5.4.1.

1) Define the observed problems. Scope the problem as to the system and the time but be conscious that whatever problem is discovered, and its root cause may also be occurring on other similar systems.
2) Gather data and evidence of the problem, and characterize as "What," "When," "Where," "Who," and "How," leading back to "Why," and the source of the problem. Collect trouble tickets, logs, computer dumps and images, times, dates, configuration and user's observations, from normal state through the problem state to recovery – the lifecycle of the problem.
3) The Burnt Toast example lays out the events according to the paradigm of People, Process, Tools and Material is added. This will help the team from overlooking any events. They can be discarded later if not part of the cause.
4) Slowly, at least one common source will be disclosed. Most complex problems are from two or more sources that would not otherwise normally occur if they occurred singularly.
5) Clearly state the source(s) of the problem and then identify ideas on how to resolve the problem. This is your corrective action.

6) Test the corrective action by observing the new behavior, updated documentation, vetting with others, rechecking sources, etc. Create a change plan that may include a modification, updated instruction, training, etc.
7) Once the change is made, refreeze and re-baseline as needed. Observe to ensure the problem or another problem does not occur. For the Burnt Toast, the source of the problem was "No Instructions," but there were contributing factors, such as his wife left for work early and he had to make toast, the bread was thinner and would more easily burn, plus he simply operated the toaster with little regard to the of the result
8) Share your knowledge – Check other systems, operators, or organizations/projects for similar problems so they may avoid the same problem. For the Burnt Toast, who else would buy that brand, or the type of bread, or not have a backup plan on the operations of the toaster.
9) File the results of the RCCA in your KM Repository.

Someone Knew It Was a Problem

From my experience of performing after action reports I have observed that people involved in the project or system knew of the problem. Usually, a middle level worker with credible experience saw a systemic shortcut, a poor decision, undue risk, the wrong person – and knew there would likely be an emerging problem that could threaten the outcome. And this includes when they reported it yet senior leadership failed to understand the issues and take corrective action, as if the problem would magically go away. This is why I wrote Section 0 Milestone Zero first.

5.4.5. Poised to Deliver

A common question (from the formal training and discussions) is how a Systems Engineer, or anyone, can use processes or a development model and not get bogged down? How should I plan for an emergency, or not be inhibited by process when someone needs a system now? Often, people state their need as Quick Reaction Capability (QRC). Here are some suggestions.

- Open Architecture – that can easily (and quickly) accept interfaces, commercial products.
- Indefinite Delivery/Indefinite Quantity (IDIQ) Contracts – existing contract with a series of task orders that can be quickly enacted, without having to go through an extensive acquisition process.
- Anticipate potential needs and requirements and stage a provider with potential designs.
- Providers can have supply agreements already established (often critical path activities).
- Providers could have some candidate designs completed if given requirements.
- Stage equipment, keep hot and warm spares.
- Concurrent design and development – require close team coordination so the pieces can integrate.
- Use mature technologies (See Section 1.9 Technology Readiness Level 6 to 9) that requires little testing and preparation time.
- Move activities out of the critical path early by completing these tasks such as security accreditation.
- Keep signature authority and approval authority to a minimum by delegating and empowering. Many an hour and day have been lost awaiting an executive signature that was probably not really needed.

- Best Practices and Processes – confirm that they are current and efficient, so when used it not like the first time they were ever used.
- Model Bases Systems Engineering (MBSE) – modeling to existing system capabilities, so it may be used instead of building new. See Section 5.3 on MBSE.
- A trade-off of cost, schedule, performance, and risk is made. Expecting an accelerated schedule may cost more or increase risk. Expecting low risk and high performance will influence schedule. Delegating signature authority will mean giving a subordinate manager the "big picture" so that they know what to approve or not approve.

5.5. How People Think and Learn

Part of your job as a Systems Engineer is knowing how people think and solve problems. No one person approaches a problem the same as others (as pointed out by in Section 1 Systems Thinker and Perspective) but there are some common attributes to consider.

5.5.1. Learning Model

Bloom's Taxonomy – I find Bloom to be an extremely helpful guide on how thoroughly one needs to know and apply a particular topic (Vanderbilt University, 2020). Bloom allows for focused communication, and in the case of this Book, a guide to how well the System Engineer should know their topic. I recommend this taxonomy to my fellow Systems and Program, Project Managers and Leaders in their efforts to coach, teach, mentor and train. These levels are:

- Remember – recall facts and basic concepts
- Understand – explain ideas and concepts
- Apply – Use information in new situations
- Analyze – Draw connections among ideas
- Evaluate – Justify a stand or position
- Create – Produce new or original work

I attempted to write this Book to the Understanding and Applying Level of Understanding, with some Analysis. More information on Bloom may be found at https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

5.5.2. Reasoning

This short section will aid the reader in how people reason, or how they should approach a problem. Those trained in networking tend to follow a flow from beginning to end (holistic), while those more attuned to the computer approach will isolate a problem starting at the top and working down (reductionist). Knowing and accepting this will aid in working as a team to solve complex problems. Here are more examples.

Inductive Reasoning is a logical process in which multiple premises, all believed true or found true most of the time, are combined to obtain a specific conclusion. One will define a hypothesis, collect data about the problem and derive a conclusion. The data will support the conclusion. Inductive reasoning is considered 'bottoms up' logic.

Deductive Reasoning is a logical process in which a conclusion is based on the concordance of multiple premises that are generally assumed to be true. If one accepts the previous hypothesis, then one will accept the conclusion. Deductive reasoning is sometimes referred to as top-down logic.

Where deductive reasoning proceeds from general premises to a specific conclusion, inductive reasoning proceeds from specific premises of a general conclusion.

Deterministic and Stochastic - Deterministic is when a range of results (answers or output) are expected and anticipated in solving a problem. Stochastic is performance on large data sets, and the resulting output will have a greater range of alternatives. (Deterministic vs. stochastic models, 2013)

Methods of Processing

- Deterministic
  - Empirical
  - Knowledge
  - Driven
  - Process Driven
- Stochastic
  - Input Random
  - Output Variable
  - Ex: Monte Carlo

(NCSU, 2013)

More information on Deterministic and Stochastic may be found at https://www4.stat.ncsu.edu/~gross/BIO560%20webpage/slides/Jan102013.pdf

5.5.3. Predictive Analysis

Systems Engineers once performed corrective action and lessons learned after a system failed. Often there was system damage or a mission failure. Soon came preventive action that was performed as scheduled intervals or when early signs indicated a harmful event may occur. Granted, we struggle in our systems in advancing from corrective to preventive and now to predictive. Those who have moved to **Predictive Action** or **Predictive Analysis** are realizing cost savings, performance improvements, and they often shape events to their advantage because they can anticipate an outcome, and they can shape it to exploit it. We already spoke of many of the tools that aid us in Predictive Analysis such as Decision Tree Analysis, Knowledge Management, Deterministic Systems and so on. But before we move to Section 6 and before we progress into the 21st Century too far, we must understand the importance and usefulness of Predictive Analysis.

One can anticipate problems using leading indicators by observing machine temperatures (Thermo Engineering), vibrations for mechanical systems, system behavior, irregular timing and so on. Once these attributes are placed into a large data system with many other records of systems behavior, one can anticipate events, or predict an outcome. This is consistent to Determinist Processing. Aircraft today, for example continually send back performance data to a central location while in flight, to notify pilots of a problem that may occur, logging durations, capturing temperatures, vibrations and aircraft behavior long before a problem evolves. This data is compared to other data and notifies engineers of an emerging problem before it occurs. This is Big Data Analytics which is now come of age. The Systems Engineer must understand Big Data Analytics to be successful in this decade. By merging sensor data, performance data and comparing to similar systems, a system can 'learn' to perform with little human control. Predictive Analysis is used in **Artificial Intelligence** and **Machine Learning** that allow autonomous machines to function.

5.5.4. Artificial Intelligence

AI sort of snuck up on us. What I mean is some sort of AI emerged in the 2nd generation of systems engineering when systems were delivered aiding a user to resolve a problem. Most impressive of these were the real time systems that a preconfigured action was activated, or a notice was dispatched from a remote sensor. What is different with today's AI is that it comes to you on your handheld device or workstation, and users naturally interact with AI, requiring no training or orientation. I cannot think of any other agent or application that adopted how the user interfaced instead of a user having to justify the cost of a new application, get the training, loading and configuring only to have disappointing results. The AI we now have cannot do anything a person cannot do except it does it much more quickly – ask it a question and it will provide a narrative or graphic answer.

Depending on which book you read, or YouTube you watch, there are three kinds of AI 1) Artificial Narrow (Weak) AI -that will perform basic predefined functions. 2) is General AI or AGI – knowns as Strong AI, which take previous learning to perform tasks without human assistance and 3) Super AI – which can think, reason, learn and make judgements and possess cognitive abilities surpassing human being (https://www.ibm.com/think/topics/artificial-intelligence-types) Super AI is still theoretical.

We have embedded AI into factories, transportation, medical and science, military, weather, space craft and countless other applications. It's our responsibility to ensure it is managed and used for goodness, and it does not cause harm.

5.5.5 Managed Services

While not a new phenomenon, organizations have found their ability to manage services they do not specialize in as unwieldy, a burden that takes time away from their core mission. If you ran an insurance company, do you want to spend resources on managing a large IT system? If you ran a manufacturing organization, you want to focus your innovation on building things. Outsource these services to specialist in which you rely, that gives you the time for your job. And a specialist that understands your business and can help systems become more innovative and efficient in achieving your mission. Managed Services should have a **Service Level Agreement** (ref 4.2.8 ITSM) that will ensure you get the levels of up time and responsiveness needed to achieve your job.

5.5.6. Black Swans

We've all experienced the proverbial Black Swan – an event so unexcepted and usually dreadful, and no one saw it coming. There are real Black Swans, but few have seen them. People think of White Swans and know that perhaps, out there somewhere is a Black Swan. Suddenly one day, a Black Swan floats by leaving you in total awe. The same is true with improbable events that society thought would never happen. Can you think of any? Often people think of the Terror Attacks of Sept 11th 2001 as a Black Swan events, or the recent COVID 19 as a Black Swan but after they occur, in hindsight we really 'should have seen them coming', as it is said. There were warnings and we really knew that Black Swans existed, but our limited imaginations felt these events improbable. Perhaps our assumptions are not correct, perhaps we used Determinist instead of Stochastic, or we had the inability to be creative. But for whatever case, we should consider the Black Swan events. Artificial Intelligence and Machine Learning can help us see the Black Swan.

5.6 Summary of Section 5 – Supporting Skills

You will have lots of help as a Systems Engineer – it's the nature of the job to integrate differing engineering domains and supporting skills to make the systems work. Some of these skills are;

- Systems Analysist – problem solver, determine how a system works, improve performance
- Operational Research – analytically solve hard problems – mission assurance
- Usability Engineering/Human Systems Integration – human to machine interfaces
- Human Factors Engineering – reduce human errors, increase productivity, safety and comfort
- EOHS – biological, chemical, physical factors affecting human health and environment

Quality is the responsibility of all those working on the project and system, from the inception to when the system is retired. The job must be completed to the expected level of performance, an adhere to all safety, security, and health requirements. The Quality Manager and the Quality Staff provide oversight into quality by conducting audits and training, improvements and ensuring the voice of the customer is included in all parts of the system. Finally, the Systems Engineer has a plethora of tools and methods they may employ to solve problems, improve efficiencies and to ensure effective communication. A few of these methods are;

- Quality Functional Deployment (QFD) – Voice of the Customer
- Continual Improvement – continual 'Identify – Plan – Execute Change – Review'
- Decision Tree Analysis – make complex decisions several moves ahead
- Linear Algebra – Throughput, Max/Min Profit, Efficiencies
- Root Cause and Corrective Action (RCCA) – trace a problem to its sources
- Blooms Taxonomy – to what level someone must understand a topic
- Deterministic and Stochastic and the Black Swan Events

6.0 What's Next in Systems Engineering

I start this section on a discussion by introducing technology and systems to a business market, and how this was done in the past and how it may be successful for future ventures. **Section 6.2** compliment **Section 6.1** by explaining how to build a business using Systems Engineering. **Section 6.3** describes the three generations of Systems Engineering. **Section 6.4** describes the Global Digital Grid that has become known as the Digital Information Eco-System. I took some liberties in this area based on my observations, training and research.

6.1. The Triad of the Systems Business

One of my professors at The George Washington University (Dr. Gerald Brock) presented the venture of new technologies into the consumer world as having a mutually agreeable paradigm of technology, an economic demand, and policy that allows or enables the business (Brock, 1999). When the three of these converge, business will emerge and flourish. The Venn Chart in **Figure 6-1** illustrates this concept.

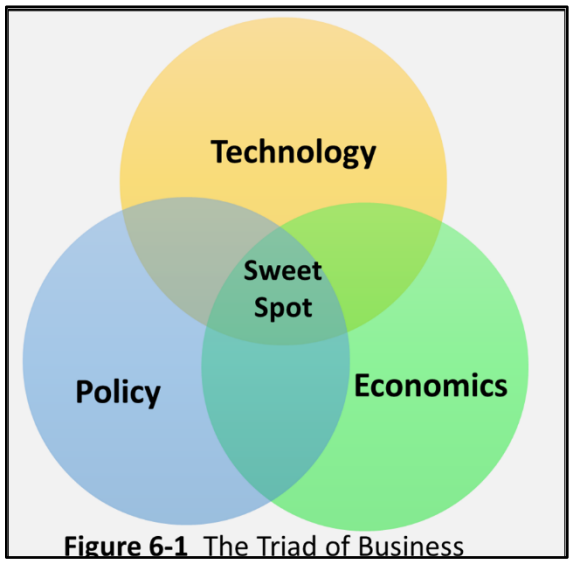


Figure 6-1 The Triad of Business

Technology as made possible by science and engineering – physics and science continually make discoveries that can be transformed into usable systems. Rules and theorems for mathematics and science have been around for years but only through science have we made it plausible for higher processing speeds, energy management, artificial intelligence and more. For example,

- Nyquist – (Shannon's' Law) – given a wave form, enough of the form's pattern is sampled and stored, so the waveform can be reconstructed later and at another location. Introduced in the early 20th Century, this Theorem or Law is the basis for modern high-speed telecommunication.
- Bayes' Theorem – Probability of an event occurring based on prior knowledge. This is key in 'Big Data'.
- Maxwell's Equations – foundation for classical electromagnetism, optics, electrical circuits. Latter part of 19th Century.
- Deterministic versus Stochastic – expected output within a range vs unknown output (The Black Swan).
- Alan Turing's Theory of Computation – the basis for modern computing and encryption.

Policy as reflected and encoded by the will of society. A policy protects businesses, consumers and its citizens from safety, security, economic and health hazards. International and National Organizations monitor and help create policy. Examples are:

- Federal Communications Commission (FCC) – for managing electromagnetic spectrum, preventing interference and general spectral chaos. The 1982 Telecommunicator Act (97th Congress of the United States of America, 1982) deregulated the industry and started the spectrum auction (in 1994) paving the way for cellular communication devices.
- Federal Aviation Authority (FAA) that regulates air routes, aircraft and the operations of aircraft.

- The aforementioned Section 00 Enlightenment Era in which monarchies fell and the rigor of science ushered in a new age of technology.

Economic demand as reflected by limited resources and information, that people will exchange a portion of their own resources for an improvement in their condition. The amount exchanged and the amount they're willing to receive is subjective, and elastic. This relates to the business case for technology. People employ their skills to create wealth is a 'win-win' for all who participate. This phenomenon was first written by Adam Smith's 'The Wealth of Nations' and is the basis for Western economies today. (Smith, 1776).

In short, the technology may exist in a lab and there may be a demand for the technology, but policy may restrict its use. Or the policy allows it and the technology is available, but the price is not low enough to appeal to a buyer. Or the theories exist but there is not a machine that can perform at the speed and capacity required to put the theory in practice. The three elements must be realistic and enacted before technology can be made available and purchased by a consumer. Because of this, we are now seeing newer and greater innovations become a reality that we could not even dreamed a few years ago. Policy is less restrained and encourages the unleashing of technologies that are powerful enough to apply the old theorems of large numbers and statistics, that in turn hit the affordable price point. It is prudent for today's Systems Engineer to understand not only how to design and develop systems, but they should also know the other influences in making system available to society.

Systems Engineering is evolving from a rigid design process to model based, various development processes to improve agility and decrease risk, and now to a breathtaking speed that's transforms innovation to market. For example, there was no need for the iPhone until one was invented. Once the price point was hit, people started to buy them and then discovered what they could do – sort of backwards to determine the requirements first. Now we need iPhones.

Numerous capabilities (which I will briefly discuss in the next section) have accelerated the Systems Lifecycle process by going from requirements to operations in minutes instead of days. The use of virtualization (invented in the 60s) is now, commonly used. The use of high speed and autonomous vehicles offer drones to observe environment or deliver packages. Vast amounts of data are collected and used to anticipate systems and human behavior, and to deliver an instant service or agent – such as calling an Uber, pointing out a product one may desire, or anticipating a target. Is there a mission need or user requirements or will someone just 'do it'? How will this system be Verified and Validated? Will policy protect people's privacy and theft of information? Will technology be applied by those who intend to do us harm? Just because we can instantly communicate or deliver products, or services – should we? These are exciting times for a Systems Engineer and for Policy Makers, but these technologies are wrought with societal risks

6.2. Systems Engineering in Growing a Business

A special area of interest to me is applying System Engineering practices and processes into new business activities, especially as related to government procurements. I've considered capturing new business as especially challenging, and my slogan is 'capture and proposals are the convergence of sports and business'. Its competitive and not for the weak hearted.

As for how this relates to System Engineering, I have seldom seen adequate Systems Engineering in a government acquisition or proposal process. As a result, poor work and writing is provided by non-experts, customers are disappointed, contractors are stressed under the paradigm of delivering a product, under oppressive cost restrictions, and hardly on time. So instead of reviewing the entire acquisition process and the Systems Engineering Processes I will present a handful of 'rules' to follow by both the procuring party and the contractor. Here goes.

1. The procuring organization should be clear with WHAT they want and allow the contractor to determine the most suitable way to deliver. Many Statements of Work dictate HOW a system is to be built which eliminates some viable and valuable solutions.

2. The procuring organization should allow the contractors (well before they prepare a bid) to know the big picture. Providing this context will allow a greater understanding and better ideas.

3. The procuring organization should provide functional (or user) requirements that are decomposed, measurable, etc. Allow the contractor to create the system requirements, and to change the architecture or to create it from scratch. If it is not possible for the contractor to write systems requirements, allow for the system requirements to be easily changed.

4. Contractors – some are so large they don't know who else in their company may be bidding – figure it out and speak with one voice. Get the right people on the job.

5. Contractors - start early – very early. If you think you know what your potential customer may need create a couple notional alternative solutions so when you get the real requirements you will have at least thought some of it through. Recall Section 3.1.7.1 and Most Important Requirements.

6. By SOW or DRFP – the Contractor should have completed trade studies to select the best teammates, the most suitable solution and will have had what is essentially a PDR or even a CDR. A few adjustments and modification are expected as more is learned once the Draft and Final RFP are provided. Don't expect to start the new business process when the DRFP is released – it's too late.

7. DON'T – if the contractor is behind schedule during the proposal phase, don't just throw together a solution and throw a price on it to win – its high risk, likely has fluff and hopefully you'll lose so you don't have to waste the buying organization's time or money. Your bogie (or price to win) may not be possible. Its OK to say No Bid.

8. Contractors – build your team early and have team management commitment – so that when you need someone for just a few days, they are available and not unexpectedly assigned to another job and out of reach, or as I've seen, laid off.

9. Maintain Team continuity – retain the same Capture and Proposal team members as much as possible throughout the campaign and proposal process. Ensure those coming on the proposal are briefed on the RFP, themes, strategies, etc., Some people will come and go as needed so this is especially important for them. If someone is not working out, find a replacement. Proposals are not a proving ground.

10. Be thorough in describing your solution, staffing, and pricing to include realistic work breakdown structures (WBS) and Basis of Estimates. This adds confidence to your price and your ability to

execute. Know what you are bidding – low price technically acceptable, or Best Value, or Past Performance assessments and structure your bid accordingly.

A few more words on Systems Engineering in Business. Know what you will deliver to your customer and of what value it will be to their interests. Value is not simply profit or completing a task or turning in a deliverable. It is determining the root of their need, anticipating how you can satisfy it, and preempting the need for them. Value is learning and understanding the tacit needs that you are not necessarily expected to deliver, but you do deliver. It's preventing the 2AM phone call, a product that easily integrates and is scalable to sizes greater than originally required, a solder returning home safe and sound, and having the resources in reserve that others may not have thought.

6.3. Three Generations of Systems Engineering

I've observed everchanging transitions over the years, and most make me ask 'so what'. But in hindsight I can see what transitions such as virtualization, regionalized data centers, open standards, software reuse, common interfaces and autonomy have merged and became useful, in fact revolutionary in our journey of making systems built by humans, to serve humans. I am not a futurist nor can I fully evaluate the value of emerging technologies (few can) but here is my take. It would serve you well to understand and be part of these changes and to not necessarily stay 'old school'. I have observed what I would call the three generations of Systems, mostly in hindsight, to which I will present to you now.

6.3.1. First Generation of Systems Engineering -- 1920s to 1970s

Systems were built primarily from a Reductionists, and hierarchy approach, with the focus on the emergent behavior. It included a Top Down, Decomposition of the requirements that worked well for systems and capability that did not yet exist. This resulted in many great capabilities that have kept democracies safe and our populations reasonably comfortable. But systems were not easily interoperable – they were stove-piped, a one of a kind. Any interfaces or integration was done by a device that was uniquely built to provide an interface and it often failed. It was not unusual for the interface to be an air gap – where data or function was manually transferred from one system to the other – which is slow and prone to errors. Many systems were unnecessarily duplicated and cost rapidly increased from the duplication.

6.3.2. Second Generation of Systems Engineering – 1970s to 2000s

The idea of a holistic approach started as early as the 1968 with the advent of Virtualization – so that a computer environment (image) could be loaded on different processors and memory units as long as the environment met certain configuration standards. This streamlined upgrades, testing, and development but it didn't really catch on until the 90s. **Service Oriented Architecture** (SOA), which required a large communication bus, loosely coupled defined interfaces and an assorted set of services came about in the 90s. SOA saved lifecycle time in that users could select the type of service they needed. Additionally, rapid processing speeds, greater capacity of memory and improved use of spectrum opened new opportunities to more rapidly develop systems based on existing capabilities. Re-use of code, methods, establishing of frameworks, Model Based System Engineering aid accelerated the development and interface of systems. Aerospace platforms, Entertainment, electronic communication was all affected by these advances. The Second Generation of Systems Engineering gave us a peek at a Digital Eco-System and an impeding change in integrated systems.

6.3.3. Third Generation of Systems Engineering – we are in it

Cloud Computing emerged in the first decade of the third millennium. Organizations (Amazon, Microsoft, Oracle and many more) had excess data storage they leased out to other organizations and individuals as needed. The concept evolved into services such a desktop application, data transport, analytics tools and several different types or short term and long-term storage. I never really had confidence on what organizations claimed they spent on information technology in the Second Generation due to the decentralized procurements, misuse of software licensing and what some may call 'rouge' budgets. But with Cloud Computing the user pays for what they use, and after they are done, they turn it back in the Cloud Provider. The elasticity saves money, allow for quick allotment of needed resources and in short – goes from requirements to operation in minutes and not weeks or days. This is phenomenal.

Figure 6-2 Digital Information Eco Systems - Only the Strong Will Survive, and Beware of Predators

Cloud Computing is now well established in the industry, and I expect many more years of evolving 'shared use' with a wider range of services that we have not yet realized. Now this capability is tethered to far flung devices – sensors measure traffic, volume, the environment, to observe and devices held in your hand (known as 'phones') but provide an enterprise of application at your fingertips.

The idea of a 'global system of systems' is revealing itself as a 'Digital Information Eco-System' (Audrertsch, 2005) as illustrated in **Figure 6-2**. Wikipedia provides a brief, non-sales description of the Digital Ecosystem (Wikipedia, 2019). https://en.wikipedia.org/wiki/Digital_ecosystem. National Institute of Standards and Technology (NIST) has a collection of forums and articles on the Digital Ecosystem (National Institute of Standards and Technology, n.d.)https://www.nist.gov/fusion-search?s=digital%20ecosystem&start=30&sort=relevance

Data Center Infrastructure Management (DCIM) applies predictive analysis to anticipate problems and infrastructure demands. DCIM joins IT operations with Infrastructure Operations (heating, cooling, water, power requirements) to anticipate surges in demand, provide load balancing, offer efficiencies for disk and processing demands. DCIM identifies and eliminates bottle necks, threats and risks to the system using a record of similar systems behavior. This takes specialized hardware for analytics, monitoring, detection supported by an operations center. DCIM is an example of the use of predictive analysis and the convergence of what is otherwise differing engineering disciplines – in this case Civil, Mechanical, Thermal and Electrical Engineering, and Computer Science.

The Third Generation of Systems Engineering (that I estimate has started about 2000) is consistent with the Fifth Era as described by in the book *Corporate Innovation in the Fifth Era* (Le Merle & Davis,

Application, Services and Agents – there's a difference

I'll take a chance here. We once had standalone software like FORTRAN, COBOL and Assembly Language. Bound within this software were most of the components (utilities, services) needed to run the program. These languages gave way to structured Pascal, 'C', Ada and more in which common services were called to perform functions like printing, data calls, inputs, etc. But it was still controlled by the written program. Then came the next generation where software-initiated Agents that would go off into the digital world and perform functions that were autonomous to the program that started it – almost as if it took on its own life. Why is this important? It demonstrates the autonomy of computer and networked systems, that how no one person knows the far-reaching effects of systems, and the challenges of maintaining control of Agents that may kick of unintended actions, or collect and transport sensitive information.

2017) that I had referenced in Section 0 Milestone Zero. The Fifth Era is an emerging era in which digitalizing, knowledge gain, autonomy, integration into a broad set of life experiences, and bio-medical is occurring. My observation is that the previous eras took more capital for one group or individual to make a profitable transition. To procure factory equipment it took an investment. To procure a network system and mainframe, it took a loan or stockholders. This capital demand, and the associated risk acted as barriers to entry for ideas and innovations that could have been realized years earlier had the capital been available to these groups or individuals. Conversely, successful individuals and stockholders in these eras could accumulate large amounts of wealth and capital that could then be used for more innovation. As we enter the Fifth Era, I suspect the barriers to fielding innovation will decrease with the omnipresent world of digitalization, and the decreasing cost of technology. But it's too early to know for sure.

6.4. The Digital Information Eco-System

I find the term Digital Information Eco-System[5] especially interesting in that it implies there is some form of this system in which only the strongest survive, and the weak or less useful wane. It suggests there is an equilibrium where the behavior of one object is subject to the behavior of other objects – that it may do harm or that 'goodness' may emerge. A Digital Information Eco-System in fact, has these qualities. It is contained within a defined yet distributed environment, it has various sub-community's and is self-organizing, it scales up or down as needed and the system is robust – it can survive due to its various services and its distributed nature. Think of how many applications you may select to forecast the weather or present the news – if it doesn't serve you, you discard it and get another. If a part of the system fails in a critical mission such as data collection from an autonomous airborne mission over hostile territory, there is a backup site or failover on the Digital Information Eco-System. If your medical doctor is performing remote medicine on you the doctor will have all your information, similar type patient information and a better diagnoses and treatment. The doctor can

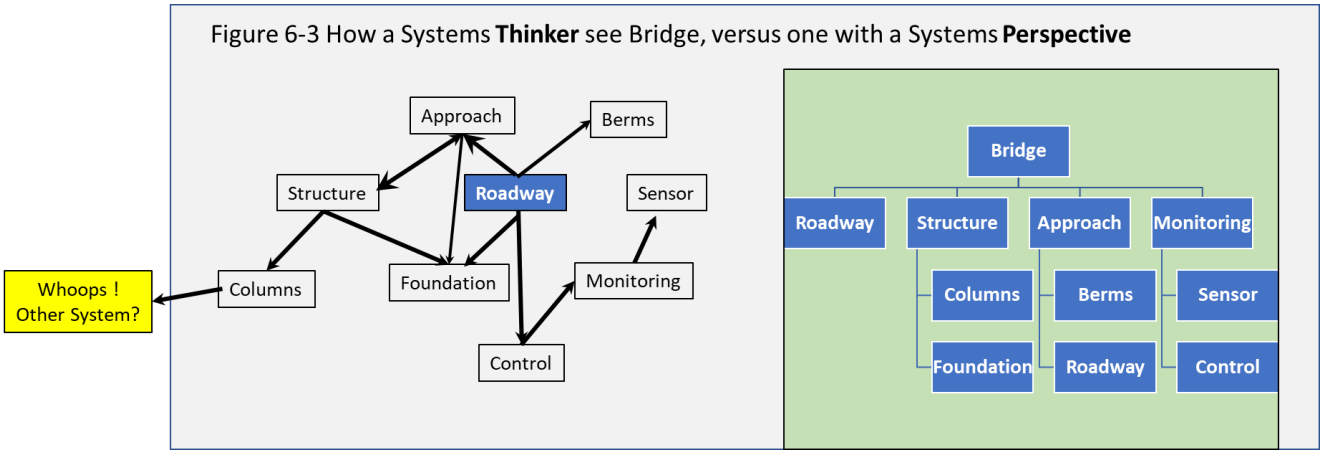
[5] The industry commonly uses the term Digital Information Eco-System but once this new paradigm becomes understood and in use the term may change. In other words, we've not yet settled on a name.

even see and hear you. Granted, it's not the same as a physical exam but it's a great improvement for many – the access to health care, and potential decrease in its delivery.

Don't do anything in the Digital Information Eco-System that you would not do in the physical world. Protect your system from intruders by compartmentalizing your valuable resources so that if one compartment is compromised, the others are still intact. Add some randomness to the architecture so that if one compartment is penetrated, the intruder cannot easily duplicate the design of other compartments, or sub-systems. Use passwords and authentication to protect objects, sub-systems, elements from unauthorized entry and spoofing, and denial of service. Identify and authenticate the source of information to prevent a third party from inserting bogus information. We are only starting our journey on the Digital Information Eco-System, and we have much to learn. Turning away from these problems, having a Pollyanna approach or uncontrolled open system will only aggravate the problem and threaten your system.

The internet of things (IoT) is the omni-presences of communication devices, sensors and controls often ubiquitously located, collecting events, and returning this data back to the Cloud. The data is used to predict events through statistical analysis as previously discussed. As 5th Generation Cellular Communication emerges (which is at least ten times but up to 100 times the capacity of LTE 4th Generation), these technologies complement one another to help make up this smart grid of a Digital Information Eco-System. Technologies such as **block chain** will authenticate transactions between two or more parties to allow transfer of currency. Block chain may occur surreptitiously thereby screening the user's identity from oversight, often scrupulous individuals. Block Chain is an example of the use of the triad – that technology exists, there is a business case, yet policy has not yet determined how it should be applied for the public good. Artificial Intelligence allows machines to accumulate and apply information to continuously improve their function. These are only a few of the features that we now see and will continue to emerge in this Digital Information Eco-System. Many questions abound to how we will control this and ensure it delivers function that serve people. Will we have the forethought to understand the cascading events such as when we connect a system without modeling its behavior first such as is illustrated in **Figure 6-3**. Remember the system is bounded but if a needed service is available outside our system, the temptation may be great for a holistic systems thinker to just connect the systems.

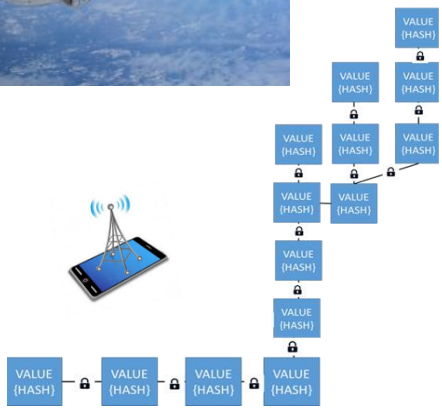
Figure 6-3 How a Systems **Thinker** see Bridge, versus one with a Systems **Perspective**



The 20th Century was age of Systems Perspective – those systems were a hierarchy created from the start – a top-down way of designing a system. Yes, there were Systems Thinkers but when few real systems existed, we needed the top-down Reductionist. The 21st Century and its innovations are now leaning more to the Systems Thinker, holistic thinking and the Digital Information Eco-System that has been built, and will continue to be built with interconnected systems, data centers, 5G & 6G Cellular, Internet of Things, Sensors, Big Data, Analytics and the topics we have been discussing. These things are here, and more are on the way. As Systems Engineers, we need to understand and go beyond the hierarchy way and think holistic. We must know what we are doing, why we are doing it, its consequences, allow for a measured approach, and understand the often-massive risk we will face if we get this wrong. It is an exciting time to be a Systems Engineer but also, a sobering time.



6.5. Summary of The Pragmatic Systems Engineer



We've started our journey before the first millennium, when few people even knew of science and engineering, and when cultures and societies began to evaluate their condition and environment. They then began to forecast and model it, adjust, and then capture its energy to improve our condition. Big changes occurred in the **Era of Enlightenment**, with the acceptance of science and self-determination. In the 19th and 20th Centuries, people began to build real systems, by using a systems process. People are **Systems Thinkers** – that systems are created bottom up and relational, and with a **Systems Perspective** – top down in a hierarchy with emergent behavior. Systems Professionals are recognized as the custodians of this new engineering discipline, and as ones who create new methods, models, and frameworks to suit the times, the organization, and the needs of the system. Systems our not created spontaneously but through planning based on **Cost, Schedule, Performance** and managing **Risk**. Systems Engineers must never forget the **People, Process and Tools** Paradigm and the corresponding **inputs** to the system of energy or data. System behavior is continually modeled, and only when it is acceptable will the Systems Team, in hand with the management team proceed to the next step or phase.

Systems Engineers must have leadership and management traits. The organization must be well focused – top to bottom, with defined roles and responsibilities. There must be a consistent and open form of communication, where all team members count, and the organization delivers value to the customer and to its stockholders.

As systems become more dependent, interfaced, and global the Systems Engineer is more challenged and must be more astute than ever. Not only must they now apply the traditional reductionist and hierarchy approach to systems, but they now must know and embrace the holistic and relational way of systems. Without understanding these, management and realization of systems will get out of

hand, unwieldy, and at risk of catastrophic failure. Systems cannot take on a life of their own – they are developed by people to serve people, and this cannot change. The future looks bright for systems, but it will take new and innovative methods to develop systems; methods that are not overbearing yet aggressive enough to deliver new, and future capability.

Finally, in my observation we are at the cusp of global enterprise of assorted and integrated systems that is unprecedented in our world history. We need to grab it, manage it and use this global system that we built as humans to serve humans – the essence of Systems Engineering.

Godspeed.

Appendix A Systems Engineering Checklist

1. **Remember the foundation** – organization, leadership, management, vision, purpose roles and responsibility, effective communication, everyone counts. Recognize your weaknesses and work to improve, let others perform better than you can, and vice versa. Compliment your team in public, counsel in private.
2. **Think People, Process, Tools** – don't take any of these for granted – work hard to synchronize their activities, take care of all of them. Give your team what they need to succeed.
3. **Think Cost, Schedule, Performance and Risk** - most suitable approach is not always the least expensive, or the fastest. Remember Sergeant Murphy.
4. **Have a plan that is suitable to your project and system** – what are you doing and why are you doing it? Plenty of methods and models to choose or make a hybrid.
5. Consider the top-down **reductionist** and the **holistic** approach – both are effective if used properly – both require a plan, requirements and objective, resources, managed risks, accurate documentation and records
6. **Special Attention to the 'Digital Information Eco Systems'** – know system boundaries, maintain control of your system and project, your apps and services. Know interfaces, and your agreements with others.
7. **Risks of the Digital Information Eco-Systems** – Do in the Virtual World as You Would Do in the Physical World – firewall entries, compartmentalize the system, subsystem, don't have too much risk in any one compartment and encrypt and authenticate all entries
8. **Re-Use** – Code, Scripts, Material, Plans, Lessons Learned, Processes that work. Use and build on to existing capability – think MBSE and Model and Simulation.
9. **Encourage but manage innovation** – especially in the 3rd Generation of Systems Engineering – don't allow things to stall out because of thwarted motivation but don't let things cascade beyond your control. Think the Biz, Tech, Policy Venn Chart

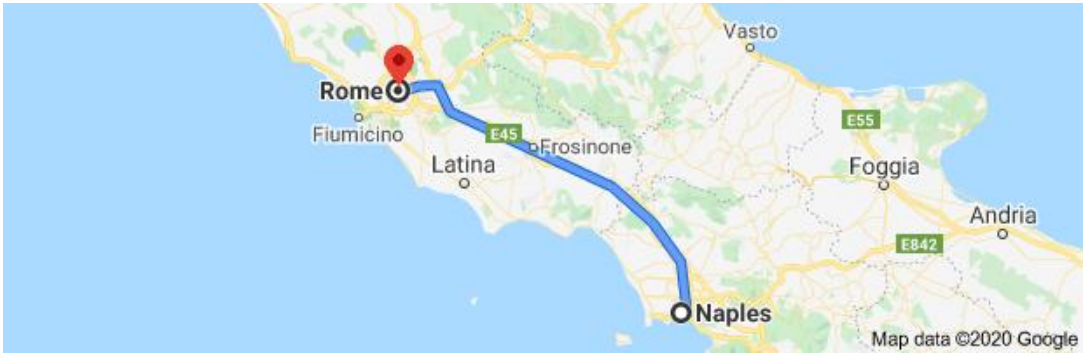
Appendix B. Case Studies – Presentations and Solutions

B-1 Case Study – Driving in Italy

15+ minutes – group or overnight individual

User Need → Functional/User Requirements → System Requirements

Need: I live in Naples Italy and need transport that I can use as transportation with my family of four. I want a clear view of the vast views of Naples. Some intersections I will encounter require a 90 degree turn inside of 20 feet, and I must have a 270-degree view of the road and surrounding area due to pedestrians. About once per month I travel the Autoroute (E45) to my cousin's house in Rome with my family of four, and three mid-sized pieces of luggage. It's a 226 KM drive. Because gas is expensive and stations are infrequent on the Autoroute, I expect to drive this duration without stopping for gas. I am big person and I have a bad back, and must modify my seat position about every 30 minutes.



What are the functional requirements and the system requirements? Think of at least five or so. What are the most important requirements?

B-2 Case Study – Volts and Bolts

30 to 60 minutes – group or overnight individual

Volts and Bolts (V&B) is a U.S. Based Company that designs, builds and delivers small electric utility vehicles. These vehicles are forklifts for warehouses and outside storage areas, pickup trucks for local driving that can carry up to 1 ton and two passengers, and small electric vehicles (golf carts) for inside and outside facility transportation. These vehicles are certified as safe for these purposes by the National Transportation Safety Board and those vehicles that travel the public roads are inspected annually. The Executive Managers of V&B want to expand into electric powered longer-range transport trucks that a carry 20 tones for 500 miles before recharging is needed. The current batteries are Hybrid Electric Automotive (NiMH) batteries that are sold to V&B by **Stray Sparks**, who have performed well and have built a trusting relationship with V&B. Stray Sparks states they can deliver a longer-range battery and has offered several options as outlined below. The first on the list is the one used now for the local transport – the Hybrid Electric Automotive.

You are the Chief Systems Engineer and you have been asked to perform an assessment of the pros and cons, that will support the business case.

- How would you go about this assessment? Think of how you can use the Systems Engineering topics to make a recommendation to the Executive Leadership.
- Which battery would you choose? Why? What should you do before choose a battery?

Battery Type and Description
Hybrid Electric Automotive
Large, dry-cell batteries, NiMH and Li-ion most common, sealed, rechargeable. Formats: Large pack of small cells. Common Uses: Hybrid and electric automobiles. Safety: Non-spill-able
Steel Case
Large, flooded cell batteries, lead acid, vented, rechargeable. Formats: Individual cells within a steel casing. Common Uses: Forklifts, industrial machinery, motive power Safety: Spill-able, lead is a toxic heavy metal, contains acid electrolyte that is corrosive if spilled
Lead Acid
Medium to large, flooded cell batteries, vented, rechargeable. Formats: Hard case, rectangular, multi-cell. Common Uses: Automobiles, motorcycles, boats, outdoor power equipment. Safety: Spill-able, lead is a toxic heavy metal, contains acid electrolyte that is corrosive if spilled. Lead acid battery recycling: With a 99% recycling rate, lead acid batteries are among the most recyclable batteries. Every part of the battery is recycled and the lead usually goes back into making new batteries.
VLRA
Description: Medium to large, maintenance-free, vented (VRLA), rechargeable. Formats: Hard case, rectangular, multi-cell. Common Uses: Automobiles, motorcycles, boats, wheelchairs, emergency lighting, uninterruptible power supply. Safety: Non-spill-able, lead is a toxic heavy metal, contains some acid electrolyte

B-3 Case Study – Corp of Discovery – Transport

1+ hour group, or overnight individual

Background

https://en.wikipedia.org/wiki/Lewis_and_Clark%27s_keelboat

On February 28, 1803, President Thomas Jefferson was granted approval from Congress for a visionary project, an endeavor that would become one of America's greatest stories of adventure.

Twenty-five hundred dollars were appropriated to fund a small expeditionary group, whose mission was to explore the uncharted West. Jefferson called the group the Corps of Discovery. It would be led by Jefferson's secretary, Meriwether Lewis, and Lewis' friend, William Clark.

Over the next four years, the Corps of Discovery would travel thousands of miles, experiencing lands, rivers and peoples that no Americans of Europena origin had ever had before.

Most of this area was unknown and unmapped, thought to be a short distance to the coast

Mission: Discover the newly purchased Louisiana Purchase. Learn and document plant and animal species, landmass, people, navigational routes, people. Return and brief the President and Congress.

Assumptions:

- Path to the West Coast **generally** unimpeded West of the Dakotas. A mountain range existed but not of great challenge to cross.
- Missouri River – able to take a boat up the river – depth and water's current generally permissible.
- Supplies, weapons, ammo can carry on the boat. Boat will have a small cannon to repel the savages or wild beasts[6].
- Portage boat as needed and carry around shallow areas, during floods, ice, shortcuts, rough waters
- Permission to travel outside U.S. Boundaries (Passports) not needed.

[6] Savages and Wild Beasts – terms used in the early 19th Century.

Constraints:

- Opportunity to use the river for transport was between April and June when water levels high, but before it froze by Winter.
- Rivers are at a depth of two feet West of the Dakotas for several miles, then return to a depth of five or more feet.
- Territory west of Bismarck ND is generally uncharted.
- Spanish held territory may limit access to Pacific. Spanish consider the Corp of Discovery an Act of Aggression
- Savages in Northern Dakotas not friendly
- Wild Beasts will try to eat them

Transport Requirements: Horses, Walking, and a boat for the Missouri River. Boat is put in at Pittsburgh, float down to St Louis, loaded with cargo and crew and go up the Missouri River. At some point, boat is removed and carried to the upper rivers of the West (the Columbia River), and floated down to the Pacific Ocean. Boat will be used for return trip in similar manner. If available, animals may be procured along the trip to assist in carrying cargo.

Transport is the need; the System is the Boat - Your assignment is:

1. Define the Mission Need
2. Provide five user requirements
3. Provide at least five system requirements
4. Sketch a basic system hierarchy
5. What are the interface and integration challenges?
6. How would these be Verified?
7. How would they be Validated?

Present to the class – ten mins each group

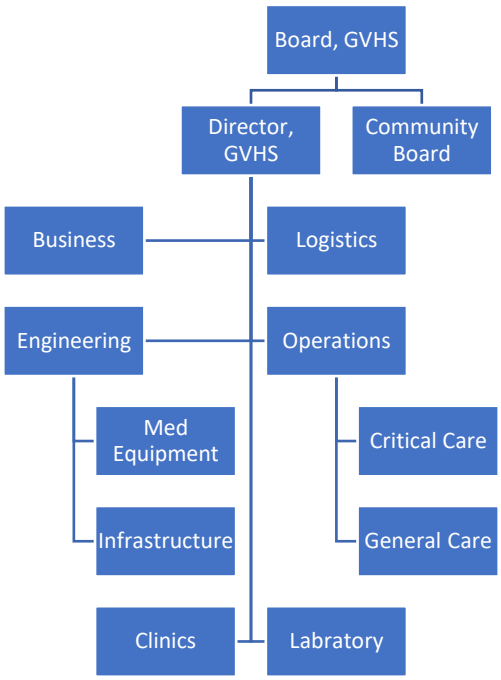
B-4 Case Study – Green Mountain

This is an epic Systems Engineering Problem that contain many of the real problems encountered by Program Managers and Systems Engineers. It is divided in two parts. Part 1 is from the View of the Customer. Part 2 is from the View of the Vendor. The entire problem will take several hours and should be performed as teams. You may break portions of the problem off to shorten the Case Study and to focus on the areas most needed.

B-4.1 PART 1 – From the GVHS Perspective

You are a lead system engineer for the Green Valley Health Services (GVHS). The GVHS has one main hospital of 500 beds and a second smaller hospital of 200 beds, and it has a cluster of ten clinics throughout the metropolitan area of Green Valley, a mix of heavy industry and technology city of 500,000 people. The current operations rely heavily on paper and manual processes, all of which are prone to loss, compromise, and it inefficient. Additionally, privacy laws driven by HIPPA and electronic medical records are driving innovation. Employees swap paper and information almost continually with other clinics, labs, practices, drug stores and hospitals, and you realize GVHS is often the bottle neck in this information flow. So, the 21st Century is knocking at the door of Green Valley.

GVHS structure is composed as follows.



- The GVHS Board is composed of five senior level executives who do not work for GVHS but perform oversight to ensure policy compliance, and that GVHS is serving the public. The board meets once each quarter or as needed.
- The Community Board is composed of ten respected citizens of the community, to ensure that GVHS is serving the community, and that any issues are quickly corrected.
- Business conducts invoicing, billing of patients, all Medicare and Medicaid Claims, Insurance Claims, and they ensure payment of capital equipment and material. They take input for the

annual budget from each department that will be approved or not approved by the Director, and the Board.

- Logistics works closely with Business in that they procure all expendable supplies, capital equipment.
- Operations is the core of GVHS – they diagnose and administer care directly to the patients. This is what most outside people see of a hospital.
- Laboratories collect samples, perform analysis of the samples and report the results.
- Clinics – there are ten clinics that are open to patients during the day and composed of the full realm of routine, and outpatient medical care. Those needing full time care are admitted to one of the two hospitals.

Key Roles are their names are:

- Director – Dr. Grace
- Operations – Dr. Winter
- Logistics - Ellie
- Business -Abby
- Engineering - You
  - Engineering Infrastructure – David
  - Engineering Medical Equipment - Ronin
- Labs – Buck
- Chief of Clinics – Billy Bob

The Dr Grace (Director) has set a strategic goal of implementing electronic medical records in 18 months, and virtual eliminating paper and manual billing and record keeping. You have been assigned as the project lead. Dr Grace wants the first report on 30 days on your approach, team, estimated cost and technology. How do you start this?

B-4.1 Discussion – Project Start Up – How do you start a project?

You obtain a representative from each directorate (those listed above). In some cases, you think you may be getting the 'B Team' but you don't really have a choice. Your first meeting with the team went well, as expected – lots of enthusiasm for doing something new, and all part of the 'storming and norming' and greeting process.

You have work to do. You must determine the requirements. You must add more detail to the schedule. You must determine if you will 'make or buy', you must determine the lifecycle management and development plan. You know there are many vendors and other hospitals out there who have electronics medical records. What should you do?

B-4.2 Discussion – Initial Planning – What activities are performed for initial planning?

The CONOPS (in brief) looks like this. Included are the problems that are encountered.

> A sick person arrives at the hospital several ways. One is the Emergency Department for urgent or emergency care. First responders coordinate with the Emergency Department to verify there is room before bringing the patient to that hospital. They also coordinate if GVHS is the most suitable hospital to take the patient.

The ten clinics operate mostly by appointment. The patient must fill out paperwork upon arrival which takes about 15 minutes and is often incomplete, because patents do not recall the name of medicines, insurance information or have contact info with them for the other doctors they may have seen.

In all cases, the paperwork must eventually be completed. Part of the risk is submitting a patient that doesn't has complete forms, or they complete the forms with erroneous information often leading the poor diagnoses on a problem with insurance. Body scans, MRIs, Bones Scans, laboratory results etc are obtained and again, prepared and submitted primarily via paper form.

Once the patient enters a care phase, either in the clinic or in the hospital, the doctor has available to them the information of the patient, usually what was provided by the patient, or screening info provided by a nurse and in paper form. Many doctors have found this information to be unreliable and have held back in their diagnoses and treatment, simply because they cannot rely on this information. Many patients return later because they were not properly treated the first time. A few have been mis-diagnosed and given improper treatment and have caused harm to the patient. Some notes from medical professional have been found in the rubbish and notes stuck to a wall or white board available for all (including other patients) to see.

B-4.3 Discussion – CONOPS – What are the challenges and issues as outlined in the brief CONOPS?

CONOPS – Continued

As a person is treated (either in the hospital or at a clinic) their treatment is documented on paper by the medical professional conducting the treatment. The medical record 'follows the patient' as they move through the system. The medical professional schedule is (also manual) is done manually. The doctor or medical professional arrive at their place of work for the day (or morning) and go from room to room with a typed list of who to see. Often there are conflicts between when two medical professions arrive to see or treat the same patient at the same time, so impromptu deconfliction must occur which not only causes additional delays but at times hard feelings and confusion. Billing is done on a Point of … Principle. Each time resources are expended on a patient a slip is completed and signed by the provider and sent to billing. Points include 1) Points of Care 2) Points of Diagnoses 3) Points of Treatment 4) Points of Collection (labs, scans), 5) Points of Food. As resources as expended (such as bandages, medicines, etc) resources are not replenished due to logistics not knowing they are running short. Considerable retail purchases have made up for the shortages but at higher cost, and its unknown the true consumption of resources. There have been many complaints by patients that they were billed for items they never received but naturally few complaints for items received not billed. Some people have two lunches and some get no lunch. The latter is a loss by GVHS and they assess there is a loss of up to ten percent of revenue due to lost, incomplete and unsubmitted slips.

END OF CONOPS

B-4.4 Discussion – The Problems

By now you can see a plethora of issues with GVHS, and why the Dr Grace wants to implement a more efficient system. What are the main areas to which you would start to formulate requirements? List a few – up to about five. How many user requirements will there be?

- Operations – care is not documented, lost records, risk of mistreatment, food service and medical treatment is prone to error.
- Logistics – unknown consumption, unable to maintain sufficient stock, depleted items requires large retail purchases a much great expense.
- Engineering – didn't say this but if equipment breaks or is out of calibration, they may not know it. Would you trust this hospital with what you know now?
- Communications – slow, inaccurate both internally, and to other health care systems. Most pharmacies and other clinics, and insurance companies have compatible le systems that allow for quick and accurate exchange of information.

Your team (one rep from each department) has embraced these issues and have a great desire to fix these problems. But this will not be without challenges. Dr. Winter of Operations is 'King of the Hill' and tend to blame everyone else. Ellie of Logistics is very enthusiastic but rightfully complains when she hears comments such as logistic ran out of bandages again, or logistic is forced to dispatch their people to buy supplies locally, at a costly bulk. Abby in the Business office doesn't pay bills on time because they don't know logistic purchased items that need paid. Now some vendors have threatened to cut them off. Buck, who represent Labs does nothing but complain about the others but has no solutions, and cannot quantify the problems at hand. Billy Bob, Chief of Clinics is quiet and reserved but is observant and if asked his opinion has great ideas. Some patients have actually been lost at the hospital – when a doctor visits them, they are not where they are expected because they were moved to another part of the hospital. Some miss meals or get two meals due to the confusion. Some may not be getting the needed treatments or medicines.

As the leader of this matrixed group what should you do? How do you handle these personalities so you can focus on what needs done?

B-4.5 Discussion – Your Team –What are the 'people' challenges and tasks?

After considerable synthesis, you now have a set of user requirements. It's time to sketch a system hierarchy and what the systems, sub-systems may look like. You have a decision to make – to buy or build? Sketch a System Hierarchy, and outline a how you would make the decision to buy or build.

B-4.6 Discussion – Make or Buy – What and How do you do this?

- Sketch a System Hierarchy
- Sketch Out a Trade study for Buy or Build – what are the criteria?
- Include Risk, Issue, Opportunity Management
- Estimate the cost of doing any one of these

You, Ronin, Dr Winter and Abby seem to have a solid understanding of the needs. You work on the trade study and continue to mature it. You decide to visit other health care facilities to see how they do it. Meanwhile, chaos continues back at GVHS and there are several lawsuits in the works due to poor

treatment and unpaid bills. So, Dr Grace is putting on the pressure – she now wants this done in one year instead of 18 months.

Given the trade study (buy or build) what would you do?

You pre-brief the team on the results of the trade study and your recommended course of action. There is a robust discussion and ultimately a consensus of the best course of action. Billy Bob says little but after talking to him one on one he recommends doing this in phases and not one big flip the switch in which its all or nothing. Had you not spoken to Billy Bob one on one you may not have thought of this. You brief Dr Grace on your approach including the phased approach and she asked a few pressing questions and said she has considered expediting this by a couple months – now down to ten months. From talking to other senior people at GVHS, you suspect she will have you do this – shorten the schedule by two more months. You also learn that budget is not a major issue – that this messed up process must be fixed or GVHS may not stay in business as it is now. So, she has agreed that GVHS should buy and not build. She asked for a detailed plan at the next meeting in two weeks. She will brief the board.

B-4.7 Discussion – Initial Activities

What do you do?

You prepare and send out an RFP with the requirements, the contract items, deliverables, schedule, and a hierarchy. You desire a quick and quality solution and are willing to a bit more for this. You hope to use a system that exist elsewhere but replicating it and implementing it into GVHS. Here is a summary of the five offers. Note that this is only for implementation and not sustainment. Sustainment will be another part of the contract – Contract Line Item (CLIN).

Name	**Cost ($)**	**Schedule (IOC) From Start**	**Meets Requirement**	**Value**	**Risk**
Healthy Tech	5,700,000	6 months	Exceeds – Cloud Based	High, cost may be less if used less in out years	Used in a similar hospital
Big Sky Health Solution	4,950,000	8 months	Exceeds Client Server	High – scales up, intuitive,	RF may interference Ronin works part time for Big Sky
Johnson Tech	5,500,000	5 months They are inherently on-board w/GVHS as the incumbent IT provider	Yes, and the favorite of the staff	Medium Value - Known, do OK work now. Not sure they can scale up to what is needed, can do quickly.	Local Company, you know them and they now do some work for GVHS such as payroll, billing, email, scheduling

B-4.8 Discussion – what do you do? Which offers Best Value? Which would you choose?

PART 2 FROM THE VENDOR PERSPECTIVE

You are the Chief, Systems Engineer for a company called Healthy Tech, a company in the state of Westellvania that provides electronic health systems to clinics and hospitals. Your company has over 100 clients across three states. Your capabilities include paperless health records, documentation through the entire lifecycle of a patient, ingesting old paper records into the online system, integration to billing and accounting to ensuring timely and accurate business functions that are synchronized to the operations. But your sweet spot goes beyond records. Your company also has electronic beacons throughout the hospital that allow wireless communication for the medical staff and to each patient. Each patient has a RF device on their wrist so their location will always be known. So, there is few if any, lost lunches or missed appointments, or mis-prescribed medicines. Healthy Tech has one main office and three remote offices staffed with technicians to perform routine and touch maintenance to any one of its clients, and to provide training, status reviews and simple customer contact (How goes it? What do you need?). You have moved to a cloud bases solution using AWS but this is seamless to your clients, except they can send and receive data via their portable handheld device and have the same data and info as if they were on a desk top computer. You have taken on many clients that use paper records, transactions, etc and reluctantly moved them into the 21st Century but going as much online as possible. You've done this by instilling confidence, understanding their operations, their strategy and goals and slowing relieving them of the burden of a record keeping and data system. If there is one thing, you'd like to improve it is your cost. But as technology improves you believe the price will decrease over time. Besides, your competitors have a reputation of poor quality, delays, and cost overruns.

You are approached at a seminar on digital Health Systems by a lead Engineer of GVHS named <Name Here>. You were impressed with his enthusiasm and engineering knowledge but thought he needed some coaching on what it would take to automate and go online. You could tell he was under pressure to deliver by the Senior Staff, and in fact found out later that GVHS was having quality problems with lost records, poor care and at times lost patients. How do you handle this situation?

B-4.9 Discussion – Listen and Learn – What and How?

The visits went well. They are prepared to write and send out the RFP. They will want a project and systems plan for the entire lifecycle – from needs, requirements, to development and testing to delivery and to the start of O&M. Here are some of the high-level requirements.

- Each medical professional will document each transaction with the patient at the time of transaction.
- Each medical professional will have immediate access and knowledge (but at varying discretion depending on their need) of the patient's medical record.
- Each patient and their designated care person (family or friend) will have immediate access to the status of their patient.
- The current location and the schedule for each patient will be available and made known to the medical professional who needs to know.
- The business office will have immediate access and knowledge of the transactions for each patient.

- The business office and medical professionals will have automatic pricing for planning and for actual for each patient. (to develop the Best Value treatment plan).
- Medical Professionals and Business Office will have metrics to indicate trends in patient health, volume, costs and risks.
- The Business Office will prepare billing and manage billing of patients and insurance.
- Engineering will have access to the availability and performance of medical equipment to prevent outages or to ensure needed throughput (that there is enough equipment).
- Engineering will have access to building infrastructure capability to assess readiness, prepare for surge, prevent failures in HVAC, Water, Power and other infrastructure.

What processes would you need for this project and system? What are the KPPs? Think of the various PM, SE and Business Paradigms – what should you consider and how are these managed?

B-4.10 Discussion – The Paradigms (20 cents) – KPPs, Processes, the PM and SE Paradigms, the Business Venn Chart – sketch these out

You should know enough about this project to talk it through to the end. What problems may you encounter? What can you do to help ensure success?

End of Part Two

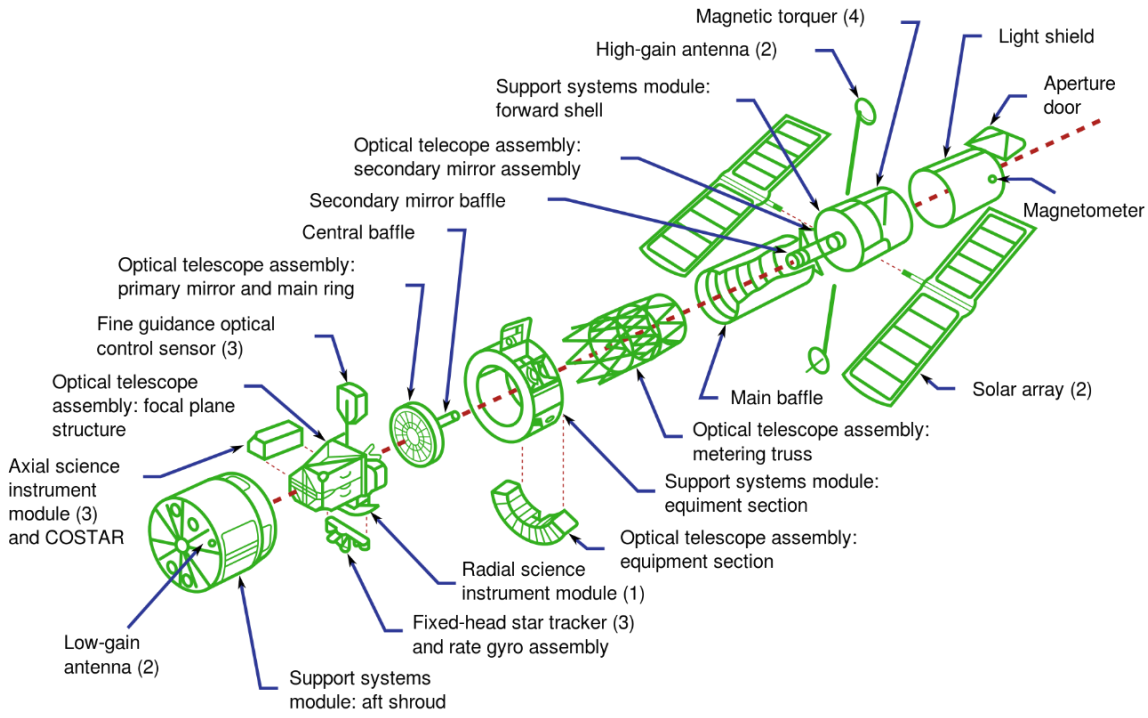
B-5 Case Study – Hubble Space Telescope (HST) (Wikipedia, 2020)

Overnight or One Hour Class – Group presents 15 mins each

https://en.wikipedia.org/wiki/Hubble_Space_Telescope#Proposals_and_precursors

- **Purpose** – Using a 2.5-meter mirror and four main instruments, HST is designed to observe ultraviolet, visible and near infrared regions of EH spectrum. It is designed to capture extremely high-resolution images and deep view into space, enabling breakthroughs in astrophysics and determine the rate of expansion of the universe.
- **Background** – HST was built by the NASA with contributions from European Space Agency. Marshall Space Flight Center (MSFC) was responsible for the design, development and constructions, and Goddard Space Flight Center would control the spacecraft in flight. MSFC contracted the optics to Perkin-Elmer to build the Optical Telescope Assembly and Fine Guidance Sensors for the space telescope. Lockheed was hired to construct and integrate the spacecraft and its elements, to include the integration of the

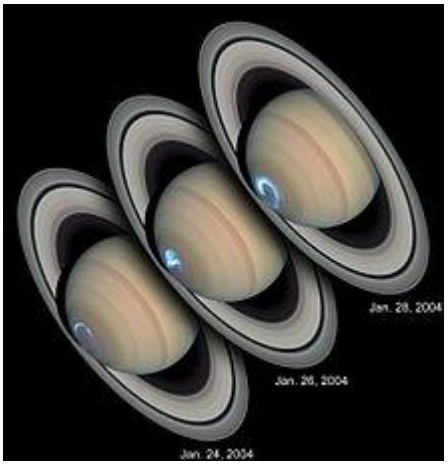
telescope. Planned cost for the entire system was $400M.



- **Design and Development** –
  - The design for the telescope included two hyperbolic mirrors. While this design provided good imaging performance this type of mirrors was difficult to construct. The mirror needed polished to an accuracy of 10 nanometers or about 1/65th of the wavelength of red light.
  - Perkin-Elmer intended to use customer-built and extremely sophisticated computer-controlled polishing machine to grind the mirror to the require shaped. But NASA demanded that Perkin-Elmer contract the work to Kodak for a backup mirror. So, the primary and the backup mirror construction proceeded.
  - Kodak and another company called Itek had previously submitted a proposal to work jointly in building the mirror and that they would check each's work, but NASA rejected the proposal.
- **Development and Delays**
  - Perkin-Elmer purchased the blank mirror (before polishing started) from Kodak. In 1979, using their computerized grinding, they started construction. The mirror was laid on 130 rods to best simulate the varying amount of force from space, and to maintain the correct size and shape. Cost and schedule delays occurred and to save money, NASA stopped the backup lenses being created by Kodak. Launch date for the HST was delayed to April 1985, again March 1986, and again to Sept 1986.
- **Launch**
  - The HST was finally launched in entered into space in On April 24, 1990, Space Shuttle *Discovery* successfully launched it during the STS-31 mission. The cost to date was now $4.7B.
- **Problem and Cause**

- Ground stations received flawed and fuzzy images even after the HST was commissioned and adjusted for space flight. Analyst determined that the lenses were polished incorrectly – and that the outer edge was too flat by about 2200 nanometers. The results were a symmetrically flawed image, meaning that most of the images that NASA and scientists had hoped to receive were now impossible. The Allen Commission was stood up to determine the problem and its cause, and from interviews and records, the commission determined Perkin-Elmer used two null correctors, meaning that the mirror was precisely polished but with the wrong dimensions. Additionally, some of the engineers and technicians at the time questioned the process, and observed inconsistent test results, and suggested it would lead to a faulty mirror but their queries and comments went unheeded. There was a schedule to meet.

- **Correction**
  - HST was designed for in space upgrades and maintenance at regular intervals. The first effort was to correct the defect of the HST by a spacewalk removing a component the captured the image and replace it with a component that would account for the symmetric flaw of the lenses. This was performed by STS-61 in Dec 1993. The new image was corrected. Other problems occurred that were not necessarily from the misaligned lenses, and these issues were corrected in subsequent missions.
- **Discoveries**
  - Ultimately, the HST capture unprecedented images from the beginning of time, of far off cosmos and some images that fifty times fainter the presently known. It greatly increases our understanding of mass and star formation, and were able to distinguish previously unseen structures and spectra.



Questions and Discussion

- Was the Hubble Space Telescope a Success?
- What were the problems with HST at each phase?

- Given the Lifecyle what were the problems – Requirements, Design, Development, Integration, VER & VAL
- Cost, Schedule, Performance, Risks (what was more important)
- What were the problems with HST in the areas of People, Process, Tools?
- What should have been done?

Read about the James Webb Space Telescope (JWST) for the next generation of space telescopes.

https://www.jwst.nasa.gov/

B-6 Case Study – The FCC's Decision Puts GPS at Risk By Mark Esper

Editorial WSJ, May 6th 2020 (Esper, 2020)

Read in advance – casual guided discussion – 30 mins

Every day, tens of millions of Americans rely on the Global Positioning System. We use it for location features in cellphones, navigation for vehicles and aircraft, and financial and commercial transactions, including ATM withdrawals. And every day, the Defense Department and our colleagues across government use GPS to protect and serve the public by coordinating global trade, banking and transportation, as well as tracking terrorists and other threats to U.S. national security.

A recent decision by the Federal Communications Commission, however, will degrade the effectiveness and reliability of this critical technology. On April 20, the FCC announced its approval of Ligado Networks' application to create a cellular network by repurposing a portion of radio spectrum adjacent to that used by GPS. The power and proximity of Ligado's ground emissions on this spectrum will drown out GPS's space-based signals. If you've ever tried to talk to a friend while standing next to the speakers at a rock concert, you get the point.

In announcing its recent decision, the FCC rehashed Ligado's old arguments, wrapped in new language, to say that the company has made changes and the FCC has set conditions to ensure GPS won't be affected. Don't be fooled. The sheer number of cases of interference combined with the difficulty of attribution will make enforcement nearly impossible, not to mention expensive.

Independent testing and analyses conducted by nine federal departments and agencies show that allowing Ligado's proposed system—including its proposed modifications—to operate near the GPS spectrum would cause harmful interference to millions of GPS receivers across the U.S. The FCC's decision will disrupt the daily lives and commerce of millions of Americans and inject unacceptable risk into systems that are critical for emergency response, aviation and missile defense. Further, it will stunt innovation in GPS; people won't use the system if they can't depend on it everywhere, all the time. For these and many other reasons, 13 federal agencies, along with leaders from a range of industries, called on the FCC to deny the Ligado request.

Ligado claims it is the solution to America's 5G woes, but its proposed license modification isn't really about 5G. There is no evidence that the company has a technically viable 5G solution. This is about one company changing the rules to maximize the value of its spectrum, and the cost to Americans is too great to justify.

The Defense Department recognizes that 5G technology is vital to maintaining America's strategic and economic advantage over its competitors. We strongly support President Trump's call for the U.S. private sector to lead the way, and we're moving quickly to develop opportunities to share mid-band spectrum, a finite resource. As demand outpaces supply, spectrum sharing holds the key to U.S. dominance in 5G. The Defense Department will dedicate millions of dollars to test 5G technologies at military bases, while promoting collaboration among government agencies, academia, and allied countries to advance to a 5G solution.

We need a comprehensive, whole-of-nation approach to develop technologies that affect so many. Disregarding the concerns of industry and government—objections grounded in hard data—the

FCC's Ligado decision is a shortsighted giveaway that will disrupt our way of life and potentially cost the American people billions of dollars.

The first and most sacred responsibility of government is to protect and defend its people. GPS allows us to pinpoint 911 calls, launch precision airstrikes, prepare our forces for combat, and otherwise act to safeguard health and well-being. Interfering with the accuracy and reliability of GPS risks the safety of the American people and undermines national and economic security. America deserves a better alternative. Mr. Esper is U.S. secretary of defense.

What is the frequency band for 5g?

The 5G frequency band plans are much more complex, as the frequency spectrum for sub-6 GHz 5G spans 450 MHz to 6 GHz, and millimeter-wave 5G frequencies span 24.250 GHz to 52.600 GHz, and also include unlicensed spectrum. Additionally, there may be 5G spectrum in the 5925 to 7150 MHz range and 64 GHz to 86 GHz range.

Radar-frequency bands according to IEEE standard[12]

Band designation	Frequency range	Explanation of meaning of letters
HF	0.003 to 0.03 GHz	High Frequency[13]
VHF	0.03 to 0.3 GHz	Very High Frequency[13]
UHF	0.3 to 1 GHz	Ultra High Frequency[13]
L	1 to 2 GHz	Long wave
S	2 to 4 GHz	Short wave
C	4 to 8 GHz	Compromise between S and X
X	8 to 12 GHz	Used in WW II for fire control, X for cross (as in crosshair). Exotic.[14]
K_u	12 to 18 GHz	Kurz-under
K	18 to 27 GHz	Kurz (German for "short")
K_a	27 to 40 GHz	Kurz-above
V	40 to 75 GHz	
W	75 to 110 GHz	W follows V in the alphabet[*citation needed*]
mm or G	110 to 300 GHz[note 1]	Millimeter[12]

1. ^ The designation mm is also used to refer to the range from 30 to 300 GHz.[12]

Questions and Discussion

- How is policy effecting the use of technology?
- What did the government do to obtain more information on the risk of Ligado's request?
- What are the risks, and who is affected? State a risk statement
- How is the Venn Diagram of Economics, Policy and Technology interwoven into this discussion?
- Which one of the three should this be an issue?

B-7 Case Study – The COVID Recovery Comes Down to Engineering by Guru Madhavan

WSJ Editorial, May 6th 2020 (Madhavan, 2020)

Read in advance – casual guided discussion – 30 mins

Reopening the country in the midst of a pandemic is akin to charging an enemy position at the top of a hill. Recovery and rebuilding will test us at every step with the risk of losing hard-won ground. But an old military insight can provide us with surer footing: amateurs talk tactics, professionals discuss logistics. Epidemiology established the right methods for fighting individual battles with Covid-19, from hand-washing to social distancing, but now it's time to talk recovery logistics.

Much of the conversation will come back to engineering, which historically has advanced public health far more than medical care has. Sanitation, water supply, electrification, refrigeration, highways, transportation safety, body scanning and mass production area few examples. It's easy to overlook how these technologies improve health outcomes, so consider one that's an obvious part of many Americans' lives today: the bandwidth necessary for telework.

In 1917, the Danish-born engineer Agner Krarup Erlang—a bookish red bearded bachelor—published formulas to manage "teletraffic" better. Erlang worked at the Copenhagen Telephone Co. and was vexed about capacity problems from call congestion. Even with centralized switching, phone calls took around 15 minutes to be connected during peak load. If a network had more capacity than demand, service providers lost money. With less capacity, the calls would drop as surely as the patience of the callers. The queuing conundrum had been treated separately as a physics, economics or psychology problem, but it became a systems-engineering problem in 1918 when the global influenza pandemic broke out, months after Erlang's publication.

By the early 1920s, about 1 in 8 people in the U.S. had a phone connection. Phones rang with announcements, weather reports, schedules and even lullabies for babies. Phones made social distancing possible during the pandemic. People conducted over-the-phone get-togethers, a habit being revived now as virtual meetings, hangouts and happy hours proliferate. The logistics became dependable, thanks in part to Erlang's work.

Today many people take for granted the systems engineering needed to maintain our far more complex internet and mobile networks. As with many vital services, only the malfunctions garner attention.

But this and many other overlooked engineering systems will be vital to the logistics of reopening. Consider the stable supply of water, phone, groceries, and internet. Think also of electric power grids, servers, data centers, gas systems, waste processing, express product delivery. Resources that are now primarily consumed by residential loads will see shifts back to the commercial sector as operations reopen. These demands will need to be nimbly managed against Covid-specific needs: how to mass-produce testing kits, cleaning supplies, effective vaccines, and affordable therapies, as well as how to deliver them in billions under a tight schedule. While the Covid-19 episode is novel in many ways, the engineering logistics required for recuperation are standard but require flexible supply chains.

Consider how engineers like Margaret Hutchinson improved penicillin from a petri-dish discovery in 1928 to its lifesaving use in World War II. Feats of manufacturing and logistics allowed the production of 400 million units by spring 1942. By fall of 1945, 650 billion units of the antibiotic were available every

month for civil and defense purpose. (A dose consists of thousands of units.) The production protocols standardized then are still in use today.

Sir Ronald Ross, the polymath Nobel-winning physician who discovered mosquito-borne malaria transmission, once said that it takes at least a decade to understand a new idea. The coronavirus pandemic has compressed that calendar. We are faced with an evolving microbe, and it will require a deft response. With dozens of simultaneous emergencies, each with its own tempo and temper, it's been a story within a story nested in another: a narrative of long-distance relationship between our lives and livelihoods; a war tale of primitive approaches versus a sophisticated pathogen; a chase sequence involving testing and treatments; a comedy on conference calls and cloud computing; a love-hate plot on the future of oil-powered commuting; and a tragedy on the loss of our confidence and competence.

Separated, specialized approaches to remake our health, economy and civics will guarantee the next breakdown. Let's engage engineers and adapt industry practices for federally organized logistics to pave the way out of this pandemic. This is an essential service.

Mr. Madhavan is a biomedical system engineer and author of "Applied Minds: How Engineers Think."

Questions and Discussion

Engineering is the use of scientific principles to design and build machines, structures, and other items, including bridges, tunnels, roads, vehicles, and building (see Section 1.3 Systems Engineering). In the case of COVID Recovery what are the engineering and analysis disciplines that must be enabled and used to obtain the goal of a COVID 19 Recovery

- What role does telecommunication (today and a century ago) have in how we coped with Stay at Home?
- How do you think the Stay at Home will affect how we apply Systems in the future?
- Given the Economies, Policy and Technology Venn Chart what elements of the 1918 Pandemic and COVID 19 are present?
- In this article, describe what is science and what is engineering?
- Why does it take a decade to understand a new idea? Does the Black Swan have something to do with this?
- What is the Systems Engineering problem that Mr. Madhavan writes?
- Why does it take a decade to understand a new idea? Does the Black Swan have something to do with this?

B-8 Case Study – Five Engineering Disasters That Could've Been Prevented with Systems Thinking, From Worcester Polytechnic Institute

(Monat, Jamie; Gannon, Thomas, 2018)

Each ten minutes, individually or as a group

- How would Systems Thinking Helped Prevent These Issues?
- Would a Systems Perspective have helped prevent these issues?
- What were the external factors that were overlooked?

What is Systems Thinking?

According to the latest paper by WPI professors Jamie Monat and Thomas Gannon, it's an approach to engineering that incorporates the full context of the problem at hand.

And it might have helped avoid some of the most notorious disasters in engineering history.

Here's a summary of five catastrophes Monat and Gannon contend wouldn't have happened if those involved had engaged in Systems Thinking a little more actively.

B-8.1 The Microsoft Zune – Discussion

What happened: Typically cited more often in marketing classrooms than as an engineering case study, the Zune is still the butt of jokes in pop culture – notably in 2017's Guardians of the Galaxy 2 – for not having the iPod's aesthetic appeal or "cool" factor. In fact the Zune, according to Slate tech columnist Farhad Manjoo, was "perfectly fine" as an isolated piece of equipment.

Unfortunately, users don't experience *anything* as "an isolated piece of equipment" anymore, especially not a pre-streaming audio playback device that, at the time, required users to maintain their own media library across an ecosystem of equipment and services. Microsoft didn't adequately situate the Zune within the context of a full *User Experience System,* while Apple's various iPod rollouts did. By leveraging systemic benefits like intuitive and stylish design, common parameters across multiple devices, a library of available music licensed for downloading, and an easy-to-understand pricing scheme, Apple made short work of Zune, which lasted only 5 years before being discontinued.

B-8.2 The Water of Ayole' – Discussion

The newly constructed water supply infrastructure of a small West African village broke down after three years, forcing residents to use parasite-infested river water.

What happened: The rural village of Ayolé, Togo relied on the Amou River as a water source, exposing residents to guinea worms, tiny parasites that cause excruciating pain. Government and international aid organizations responded to the crisis by digging and installing new wells. After a few years of regular operation, the wells shut down.

How? The village was simply not equipped to handle the normal wear and tear on their new infrastructure. There were no spare parts available, no technical expertise on hand to help fix or maintain the pumps, and no money to pay for repairs.

'Throw it over the wall' overlooked O&M

B-8.3 Fenchurch Street – Discussion

The curved façades of this high-rise office building focus the sun's reflection off its windows and into a concentrated "death ray."

What happened: Designed by Rafael Viñoly and completed in 2014, the parabolic shape of this 38-story London office complex reflects a huge swath of sunlight onto a small area at street level for several hours each day, resulting in storefront temperatures exceeding 200°F. An automobile was partially melted, and a reporter fried an egg on the sidewalk. The thermal behavior caused locals to nickname the building "the Fryscraper."

B-8.4 The Russian K-141 Kursk Submarine Disaster – Discussion

During a training exercise in August of 2000, the greatest tragedy in the history of the Russian Navy resulted in the loss of 118 crewmen.

What happened: A leak of hydrogen peroxide (H_2O_2) from one of the ship's torpedoes reacted with contaminants in the torpedo tube, triggering an explosion of the ship's ammunition. The submarine flooded and sank within minutes, condemning the few crew members who survived the initial blast to a horrific fate.

B-8.5 "Galloping Gertie" A.K.A., The Tacoma Narrows Bridge – Discussion

Everybody's seen the iconic footage of this well-known engineering disaster.

What happened: Wind shears surging through the Tacoma Narrows exerted extreme *aerolastic tortional flutter* (a synonym for "wobbling like a real-life cartoon") on this ill-fated suspension bridge. Slight sways increased the amount of surface area exposed to gusting winds, acting as a force multiplier that twisted the bridge further and required greater elasticity for it to return to its original shape.

The oscillations were exacerbated by two major factors: a deck construction not stiff enough to dampen the twisting, and vortices downwind of the bridge that essentially turned it into a giant flapping airplane wing. After a cable finally snapped, Gertie had its last gallop, collapsing only two years after its initial 1938 construction.

B-9 Case Studies – Rapid Fire

B-9.1 Data Ingest

Ingest of Data – You are the Chief, Systems Engineer for a company that provide a Public Cloud service contained on one zone. You are a startup and relatively new to cloud computing. A customer has approached you and said they have 50 terabytes of data that needs stored. The data will be refreshed two times per day, some data will be aged out based on a criterion of how often accesses, other data will go to long term storage. There are 300 users of the data. An additional ten TB will be added per month which will also need refreshed. The data primary consists of normalized data but as new data enters it must be tagged and labeled (extracted, transformed, loaded). The tools to access and analyze the data will also migrate with the data. There will be continued development work on the tools after the data is migrated. This is how the customer explains it to you because they are not extremely technical, but they think they are.

Review Section 3.0 Processes and Section 2.0 The Systems Lifecycle for reference.

- What processes would you consider in this activity?
- If this were AWS, what services would you consider?
- What type of Lifecycle Model is most suitable?
- What processes would you use for this project?
- How would you charge the customer for this activity?

B-9.2 Medical Systems

You are the Lead Systems Engineer for several medical providers to use a common data set of customer business records, medical records, pharmacy records and insurance information. The problem is that each medical office has the same information on each patient but in differing formats, often not current and prone to errors. This is a 'first of its kind' in which organization will attempt to eliminate redundant data and allow quicker access from users for what they need. However, this will have serious security requirements because not everyone can see all data. For example, Nurses and Medical Technicians cannot have access to billing information, the business office cannot have access to personal medical notes of the doctor, the pharmacy technician cannot have access to the doctor's notes on the patient, but the pharmacist can have access to this data. To make this more challenging, there is federal and state law on privacy and data access. Recent incidents are on the rise of hackers gaining access to systems containing this information. How, as the Lead System Engineer, would you approach this project?

1) Think – what are the requirements? - Jot a few of them down.
2) What are the Key Performance Requirements?
3) How could you protect this system, the data? Think Confidentiality, Integrity, Reliability
4) What type of contract would you arrange?
5) What type of lifecycle(s)?
6) What agreements are there?
7) Address the Cost, Schedule, Performance and Risk topics.
8) Address the People, Process, Tools issue – what would you use,

B-9.3 Voting System in the Cloud

Voting System – mini-CONOPs – the voter will register three months prior to the election period, and then arrive at the proper voting station, present their identification to the poll officials, use their voter id to case one vote for each candidate of their choice, or skip a candidate, ensure their vote is private ad anonymous, that votes are quickly tallied at the local, state and federal level.

- What are (some) of the functional requirements?
- Give an example or two of a KPP.
- What are the requirements for Confidentiality, Integrity, Reliability?
- What Processes should you use?
- What are the policy and technical implications of this system?
- Why is it called a System, and not a Voting Machine?
- Sketch a hierarchy of the voting system.

Appendix C Solutions and Answers to Case Studies

C-1 Case Study - Driving in Italy - Solutions and Answers

Solution and Discussion

#	Functional	System
1	The User shall turn 90 degrees in 20 feet at 20 MPH	The system shall turn wheels at 45 degrees
		The system shall have 500 pounds of torque resistance on the vehicle steering.
2	The User shall have a 270-degree clear unobstructed view from the auto.	The system shall have a 240-degree spherical windshield.
		The system shall have a video augmentation of the front view of 15 degrees either side.
3	The user shall carry their family of four passengers.	The system shall have a rear seat of six feet wide.
4	The User shall stow their luggage equaling four large suitcases each 20 pounds	The system shall have a trunk area in front of the cab area of the vehicle to allow for two 3 feet by 2 foot by 2.5 feet boxes, each weighing 20 pounds.
		The system shall have a trunk area in back of the cab area o the vehicle to allow for two 3 feet by 1 foot by 2.5 feet boxes, each weighing 20 pounds.
5	The User shall drive to Rome in four hours without refueling.	The system shall have a fuel capacity of 45 liters.
		The system shall get 85 Kilometers per liters.
		The system shall get 55 Kilometers per per liter in congested traffic (five stops per hours, each 2 minutes).

C-2 Case Study – Volts and Bolts – Solutions and Answers

Answers and Discussion to Volts and Bolts

- Mission Need and CONOPS – state the mission need for your customer and for V&B
- Imagine what this system will do in the form of a CONOPs – use a day in the life of one of these trucks. How would you prepare it for use, charge it, load it, drive, recharge, off-load it? Is it approved and is it safe?
- People – who drives it, maintain it, design and build it – do you have the expertise? How is driving it different? What if it breaks and needs maintenance?
- Process – how would you go about capturing the requirements, and develop it? What might be an operational test you could perform? - track run for 500 miles in differing conditions, or run it on the interstate for 500 miles
- Tools – this is a larger electric vehicle and may require different tools. What tools would go with the truck in case in broke down? Would the driver know how to use the tools? How would the driver notify someone of they broke down along the interstate? Also, what tools are used to build and test the truck?
- Cost – what is the breakeven? Are there other providers? What are the alternatives and options?
- Schedule – how long will it take to get approvals, to build it, get to market? Will others beat you to market?
- Lifecycle Models – what might be the most suitable lifecycle? Why

Discuss the Venn Chart of Business, Economics and Policy

C-3 Case Study – Corp of Discovery – Transport – Solutions and Answers

Corp of Discovery - Discussion - What Happened:

One of the first federal contacts, went to Jacob Myers of Pittsburgh. Didn't go so well – Myers had trouble keeping builders employed, and sober. Myers 'interfered' with their work. Lewis was very dissatisfied with the progress – boat was behind schedule and of course, overbudget. But it met the requirement.

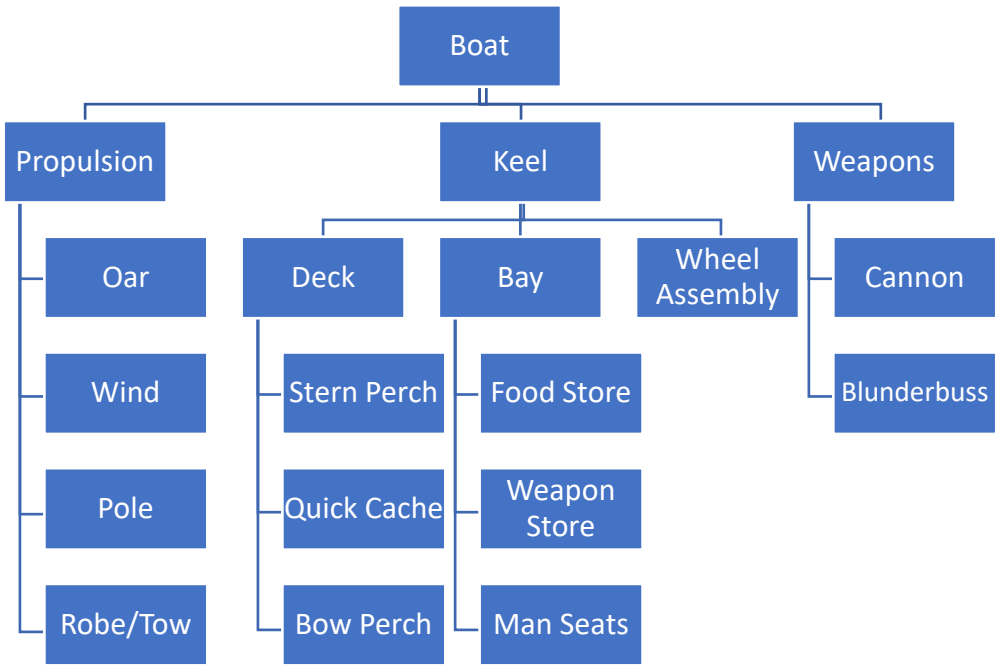
Boat made it to what is now Bismarck ND and loaded with scientific artifacts, rocks, animals, a savage and returned to St Louis. Boat was not used to get to the top of the Missouri River as envisioned. Small cut outs were used to go down the Columbia once over the Bitterroot Mountains. Mission took almost two years.

Requirements Tracing – derived from reading about the boat			
F SN	**Functional**	**S SN**	**System**
1	The User Shall propel by wind.	1.1	The system shall propel by a sail.
2	The User Shall propel by oar.	2.1	The system shall have eight-man operated oars (four on each side)
3	The User Shall propel by pole.	3.1	The system shall have one, 40-foot pole extending from the stern

4	The User Shall pull the boat by towlines.	4.1	The System shall extend two 50-foot ropes from the left and right side of the bow.
5	The User Shall transport the boat past rough waters.	5.1 5.2	The system shall portage the boat by 20 men for one mile
6	The User Shall transport the board past water of less than two feet in depth.	6.1 6.2	The system shall have attachable wheels.
7	The User Shall observe for obstacles that will damage the boat.	7.1	The System Shall have a five-foot elevated perch on the bow.
8	The User Shall operate a cannon capability of a 20-pound ball.	8.1	The System shall contain a 200-pound cannon mount on the stern.
		8.2	The System shall contain an ammo cache within reach on the mount of the stern.
9	The User Shall carry a 500 pounds of food cargo.	9.1	The System shall have two rodent resistant 50 foot by 20 foot by 5-foot food storage.
10	The User Shall carry a 2000-pound ammunition cargo.	10.1	The system shall store two water proof ammo caches (one on port, other on starboard).
		10.2	The system shall two men to pass ammo from the caches to the quick cache on the perch.
11	The User Shall carry ten men on the boat.	11.1	The system shall have five seats seat on the port side, and five on the starboard side.
12	The User Shall command the boat from an elevated position on the boat.	12.1	The system shall have a ten-foot, cloth covered cabin next to the cannon perch, on the stern.

- Go back and ask questions, get detail – Iteration.
- Go deeper – breaking it down more – Recursion.
- Observe how the boat could be built and delivered, and accepted but not suitable for the user.

Systems Hierarchy



Integration Challenges:

- Passing ammo from Weapon Store to Quick Cache
- Propulsion System – Oars from Keel
- Propulsion System – Sail Mount
- Perches – unobstructed observation
- Cannon – unobstructed shooting

Validated by departing Pittsburg and going down the Ohio River to St Louis before the actual journey started.

General Characteristics (as built)

Type: Galley	Tonnage 12 t	Length 55 feet
Beam: 100 inches	Height 32 feet	Draft 3 feet
Propulsion: Squire Rigged Mail Sail, Headsail, 20 oars	Sail Plan: 1 mast	Capacity: 666
Crew: 1 captain as commander, 1 sergeant as helmsman, 1 sergeant as centerman, 1 sergeant and private as bowmen, 22 private oarsmen		
Armament: 1 swivel gun and 2 blunderbusses		

C-4 Case Study – Green Mountain – Solutions and Answers

This is an epic Systems Engineering Problem that contain many of the real problems encountered by Program Managers and Systems Engineers. It is divided in two parts. Part 1 is from the View of the Customer. Part 2 is from the View of the Vendor. The entire problem will take several hours and should be performed as teams. You may break portions of the problem off to shorten the Case Study and to focus on the areas most needed.

C-4.1 PART 1 – From the GVHS Perspective

The Dr Grace (Director) has set a strategic goal of implementing electronic medical records in 18 months, and virtual eliminating paper and manual billing and record keeping. You have been assigned as the project lead. Dr Grace wants the first report on 30 days on your approach, team, estimated cost and technology. How do you start this?

C-4.1 Discussion – Project Start Up

- Clarify the objective
- Review a Mission Statement
- Review the Enterprise Architecture
- Get a cost bogie
- Create a Milestone Schedule
- Form your Team
- Create a CONOPS or Use Story Boards – probably both
- Explore Potential Technology

This demonstrates that project and systems are often driven by those who are not systems engineers, so sometimes you are not able to follow a normal SE process. Exploring Potential Technologies may be a bit like solutioning before you understand the user requirements, but the Director wants you to have some ideas, in 30 days.

You obtain a representative from each directorate (those listed above). In some cases, you think you may be getting the 'B Team' but you don't really have a choice. Your first meeting with the team went well, as expected – lots of enthusiasm for doing something new, and all part of the 'storming and norming' and greeting process.

You have work to do. You must determine the requirements. You must add more detail to the schedule. You must determine if you will 'make or buy', you must determine the lifecycle management and development plan. You know there are many vendors and other hospitals out there who have electronics medical records. What should you do?

C-4.2 Discussion – Initial Planning

- Revisit and verify the CONOPSs
- Determine tools needed to manage the project – such as requirements management, configuration management, schedule tools
- What processes would you start with – Requirements Management, CM/DM, Risk, Schedule ← but you don't have documented processes!!!!
- Create User Requirements (first round)
- Devise Key Performance Parameters

- Conduct research on what is in the market.
- Start the QFD or House of Quality ← this will help tell you if you can do this 'in house'
- Establish an Ops Rhythm with your team, and how you will communicate to stakeholders
- Establish a process to track funding and schedule

C-4.3 Discussion – CONOPS

- What are the problems here?
- Privacy Issues
- In-efficiencies and repeat work
- Unknown amount of people harmed by mis-diagnoses and mis-treatment.
- Completing forms – incomplete and incorrect info
- Unpaid bills – due to lack of insurance info (some referrals need pre-approval or are not covered)

. END OF CONOPS

C-4.4 Discussion – The Problems

- What are the problems?
- Lost revenue, confusion, inefficiency, manual system
- Lack of metrics for resources.
- Unable to forecast patent load.

C-4.5 Discussion – Your Team

- Realize that people are different and have different skills. Some may not want to be there and if they don't perform, get a replacement. To do that you must have specific incidents where they faulted and how you tried to correct it. Remember the leadership styles of directive, coaching, liaise faire, etc.
- Create an environment of accountability but not blaming. Start with documented goals and objectives, start risk management, create agendas and keep minutes, and action items.
- Solicit ideas from the team but guide them toward capturing the problems, and how to solve. Be a visionary and be an inspiration.
- Make sure you vision and goals are consistent to Dr Grace – and repeat these goals and the vision to the team.

C-4.6 Discussion – Make or Buy

- Sketch a System Hierarchy
- Sketch Out a Trade Study for Buy or Build, and the Pros and Cons
- Include Risk, Issue, Opportunity Management
- Estimate the cost of doing any one of these

Need & Feature	Company A	Company B	Company C	Make
Privacy				
Robustness				
Cost				
Schedule				
Functionality				
Integration				
Scalability				
TRL				
Risks				
Red – None Compliant, Yellow – Partially Compliant – Marginal, Green – Compliant – meets but does not exceed standard. Teal – Fully Compliant and Exceeds standard in most areas. Blue – Exceeds in all areas, has great value				

You will need more detail in each of the Needs and Features

You will need to assess the ability for each to deliver, and the risk with each.

If Make - You will need to hire developers and testers, which is not your core business.

C-4.7 Discussion – Initial Activities

What do you do?

- Include what you learned from other hospitals in your requirements
- Verify requirements are complete and make sure they are documented.
- Create a Project Plan
- Create a Systems Engineering Plan
- Ask for White Papers from three of the most favorable vendors.
- Draft an RFP that will go to the public.
- Make sure you inform the public (via public affairs) on what you are doing – remember to keep it simple and understandable. And be positive.

Name	Cost ($)	Schedule (IOC) From Start	Meets Requirement	Value	Risk
Healthy Tech	5,700,000	6 months	Exceeds – Cloud Based	High, cost may be less if used less in out years	Used in a similar hospital
Big Sky Health Solution	4,950,000	8 months	Exceeds Client Server	High – scales up, intuitive,	RF may interference Ronin works part time for Big Sky
Johnson Tech	5,500,000	5 months They are inherently on-board w/GVHS as the incumbent IT provider	Yes, and the favorite of the staff	Medium Value - Known, do OK work now. Not sure they can scale up to what is needed, can do quickly.	Local Company, you know them and they now do some work for GVHS such as payroll, billing, email, scheduling

C-4.8 Discussion – what do you do? Which offers Best Value? Which would you choose?

What are the pros and cons of each? Who would you select, and why?

PART 2 FROM THE VENDOR PERSPECTIVE

C-4.9 Discussion – Listen and Learn

Ask probing questions and listen. Be respectful and understanding.

Once some level of confidence is maintained ask if you can come visit their facility for a firsthand look. The confidence will go both ways – that they start to trust you, and you trust that they are sincere in their need to make a change, or else you will waste your time.

Upon the visit, - take an engineer or a business analysis, and a functional expert with you. Do not come of condescending but more inquisitive and supportive. Find out where they are in the lifecycle – ready to buy, prepare requirements, etc. If they are too far along (with an RFP out) you may be jeopardizing your potential by conducting a visit that your competition could later say gave you an advantage.

Have them visit your facility and the lab. Do not bore them with a power point presentation (keep in brief) but show them the lab, our solutions, data on how you've helped others. Now this is tricky – bring them to the point that you can deliver something that the others cannot – make them captive to your solution so that when an RFP is floated you will have features that others cannot achieve.

The visits went well. They are prepared to write and send out the RFP. They will want a project and systems plan for the entire lifecycle – from needs, requirements, to development and testing to delivery and to the start of O&M. Here are some of the high-level requirements.

- Each medical professional will document each transaction with the patient at the time of transaction.
- Each medical professional will have immediate access and knowledge (but at varying discretion depending on their need) of the patient's medical record.
- Each patient and their designated care person (family or friend) will have immediate access to the status of their patient.
- The current location and the schedule for each patient will be available and made known to the medical professional who needs to know.
- The business office will have immediate access and knowledge of the transactions for each patient.
- The business office and medical professionals will have automatic pricing for planning and for actual for each patient. (to develop the Best Value treatment plan).
- Medical Professionals and Business Office will have metrics to indicate trends in patient health, volume, costs and risks.
- The Business Office will prepare billing and manage billing of patients and insurance.
- Engineering will have access to the availability and performance of medical equipment to prevent outages or to ensure needed throughput (that there is enough equipment).
- Engineering will have access to building infrastructure capability to assess readiness, prepare for surge, prevent failures in HVAC, Water, Power and other infrastructure.

What processes would you need for this project and system? What are the KPPs? Think of the various PM, SE and Business Paradigms – what should you consider and how are these managed?

C-4.10 Discussion – The Paradigms (20 cents)

- You would decide the KPPs, GVHS would but what do you think they should be?
- What processes would you use? Most of them plus some. You'll need to add detail to the ones outlined in this Book.
- PM
  - Cost – projected vs actual – be sure the make it detailed enough to track, Consider all suppliers. **Sketch a budget and the type of artifacts and metrics you would use.**
  - Schedule – all milestones, PDR, CDR, Test Readiness Review, and all actions detailed enough to a discrete and measurable level. DO tasks in parallel to meet schedule even if it cost a bit more – remember schedule is more important than cost. **Sketch out a schedule.**
  - Performance – how well the team is performing, the system will function and is functioning. Performance is paramount – not much room for failure.
- Systems Engineering
  - People – who will you need on the team, and when will you need them?
  - Process – answered above
  - Tools – to capture requirements, do CM, RIO – to support your processes. Tools for testing

- The Policy, Technology and Economics paradigms – all three exist because have done this before but as you get into wireless you will have be sure you are using certifying equipment that doesn't conflict with other signals in the hospitals. You have to be sure the system is secure – do a risk assessment and get certified by third party.
    - Policy is driving much of the automation – medical facilities are getting relief for implementing on line medical records, and much of this is required for Medicare and by insurance companies. Also, policy will mandate the type of privacy needed, and what electronics may be used in hospitals.
    - Technology – wireless systems, cloud computing, high speed networks, encryption
    - Economics – is technology of such a price that it makes it profitable for a medical facility to use it, and for a company to sell it?

You should know enough about this project to talk it through to the end. What problems may you encounter? What can you do to help ensure success?

End of Part Two - GVHS

C-5 Case Study – Hubble Space Telescope – Solutions and Answers

- Was the Hubble Space Telescope a Success?
    - Subjective but most people would say yes. Cost was underestimated and extensive schedule delay, some of which were not NASA's fault, due to the Challenger explosion. Once the images started to return people tended to forget about the cost.
- What were the problems with HST at each phase?
    - Phase – Need and Requirements – reasonable, unknown if requirements were discrete and traced
    - Phase – Design – two hyperbolic mirrors hard to construct. There was a backup plan but it's OK to do hard things – allow enough resources to get it done, and measure progress.
    - Phase – Construction, including procurement – Proposed by Kodak to work with Itek to check each other's work but rejected by NASA which was probably a mistake.
    - NASA insisted PE contract with Kodak – not sure if the caused part of the problem or not but seems NASA should have let Kodak execute the way they proposed. Delays may have caused NASA to become more involved in the contractor decision making. Delays and overruns cause tension from executive management and they become involved and force bad engineering decisions. Some of the employees expressed concern about the process and that it may be faulty, but apparently were dismissed. Those goes back to company culture.
    - Phase – Testing (V&V) – Verification was curtailed due to schedule issues. Verification should have been done by an independent organization. Validation – appears to not have been fully ground tested. Only once in space and commissioned were the problems really noted.
    - Phase – Launch and Ops – Launch was delayed so HST went into a long-term storage, which added $$$ to the project cost. The in-space repair was one of NASA's best moments. Once in ops people tended to forget that HST was once a lemon.

    - Phase – O&M – like the component replacement plan. Assume software could be pushed up to the space craft. HST has outlasted it expected life.
- Cost, Schedule, Performance, Risks (what was more important)
    - Cost – overrun due to unexpected complexity, ignored problems and didn't appear to manage risk
    - Schedule – pressure to meet schedule probably hindered development and testing
    - Performance – Problem was in the making of the lens. There seemed to be no quality control and periodic checks.
- What were the problems with HST in the areas of People, Process, Tools?
    - People – engineers and techs appear to be qualified. Sr Management didn't pay attention. Metrics (checks) unknown of they were available.
    - Process – no periodic checks, Ver was flawed.
    - Tools to do the job probably OK. Although this was a complex project, it was doable.
- What should have been done?
    - More open culture
    - Risk Management
    - Better Project Estimation – suspect Sr Exec Managers wanted to keep cost down so it would be approved by Congress.

C-6 Case Study – The FCC's Decision Puts GPS – Discussion

Questions and Discussion

- How is policy effecting the use of technology?
    - First, check your sources, fact check, look up the FCC's standing on this, and search for Ligado
        - https://ligado.com/press/ligado-networks-supports-fcc-order-new-filing/
        - https://www.fcc.gov/sites/default/files/house-asc-letter-to-commissioners-05072020.pdf
        - https://www.ntia.doc.gov/press-release/2020/ntia-files-petition-reconsideration-fcc-grant-ligado-license-modification
        - https://www.defense.gov/explore/spotlight/protecting-gps/
        - https://blog.knowlescapacitors.com/blog/an-introduction-to-the-5g-frequency-spectrum
    - Ligado's proposal and the FCC's approval of modifying bandwidth allocation near GPS will interfere with GPS – decrease quality and reliability.
- What did the government do to obtain more information on the risk of Ligado's request?
    - Independent Testing and Analysis by nine government agencies determined that this modification proposes risk to GPS – but doesn't say the impact or likelihood.
- What are the risks, and who is affected? State a risk statement
    - If the spectrum use is modified then it will have a negative effect on the operations of GPS, thereby causing interference and damage businesses who need it.
    - Damage to military ops, commercial use and also business loss.
- How is the Venn Diagram of Economics, Policy and Technology interwoven into this discussion?

    - Appears to be used against each other – that policy is for the public good, but technology indicates it causes harm, or doesn't cause harm. Non-advocates of this mod imply that the economies (and profit) are superseding Policy and Technology.
- Which one of the three should this be an issue?
    - Mostly Policy but all three are an issue. Seems one party is making the economy the predominant issue of policy and technology. Technology – could the signal be better processes and filters to prevent bleed over?

C-7 Case Study – The COVID Recovery Comes Down to – Discussion

Engineering is the use of scientific principles to design and build machines, structures, and other items, including bridges, tunnels, roads, vehicles, and building (see Section 1.3 Systems Engineering). In the case of COVID Recovery what are the engineering and analysis disciplines that must be enabled and used to obtain the goal of a COVID 19 Recovery

What role does telecommunication (today and a century ago) have in how we coped with Stay at Home?

- Over the phone get togethers on party lines
- Virtual meetings and visits
- Online shopping
- Television
- Web Sites and Social Media

How do you think the Stay at Home will affect how we apply Systems in the future?

- Work at home, virtual meetings, encryption and VPNs,
- Higher Bandwidth and more at home professional networks.
- Reduction of office space demand more online shopping and less brick and mortar

- Given the Economies, Policy and Technology Venn Chart what elements of the 1918 Pandemic and COVID 19 are present?
    - Economies – government put money in it, the value of treatment and
    - Policy – Phase 1,2,3 testing still in place but expediated, although there is(was) temptation to bypass

- In this article, describe what is science and what is engineering?
    - Engineering - Sanitation, water supply, electrification, refrigeration, highways, transportation safety, body scanning and mass production
    - Science - The queuing conundrum had been treated separately as a physics, economics or psychology problem, but it became a systems-engineering problem in 1918 when the global influenza pandemic broke out, months after Erlang's publication. Erlangs formulat for queuing $\mathbf{E} = \lambda h$
- What is the Systems Engineering problem that Mr. Madhavan writes?
    - Once a solution is available (a feat of science) it's up to engineers to distribute and administer, and record the treatment and vaccine.
- Why does it take a decade to understand a new idea? Does the Black Swan have something to do with this?

- Not sure it really takes a decade buy it takes time for people of accept something outside their mindset, even when empirical evidence suggests this potential event is likely.
- These are Black Swans

C-8 Case Study -Five Engineering Disasters That – Discussion

C-8.1 The Microsoft Zune

You might not think of the failed attempt at competing with Apple's iPod as a "disaster," but it cost $289 million. Imagine if that money had gone to the Bill & Melinda Gates Foundation instead.

How Systems Thinking could have helped: Microsoft may have cemented a better reputation as a hardware company by building the Zune as merely one functional component of a complete *User Experience System*, rather than a stand-alone device. Instead of eliciting a chuckle every time somebody says "Zune."

You might not think of the failed attempt at competing with Apple's iPod as a "disaster," but it cost $289 million. Imagine if that money had gone to the Bill & Melinda Gates Foundation instead.

CONOPS, Integration

C-8.2 The Water of Ayolé

How Systems Thinking could have helped How Systems Thinking would have helped: Stakeholders eventually applied System Thinking after the initial well-building project treated the water issue as merely an engineering problem. Togolese extension agents trained villagers in well maintenance and repair, the local hardware store established a supply chain for repair parts, and the village's women organized an agricultural production and sales system to help pay for the parts. This uncovers one of the more important lessons of Systems Thinking: problems are best solved when incorporating the interrelationships between engineering, socio-economic conditions, logistics, and users themselves

CONOPS, Integration

C-8.3 Fenchurch Street, London

The curved façades of this high-rise office building focus the sun's reflection off its windows and into a concentrated "death ray."

Lessons Learned, the environment it is built in

C-8.4 The Russian K-141 Kursk Submarine Disaster

How Systems Thinking would have helped: System Thinking incorporates advance planning for the eventual "Controllers" and "Maintainers" of a system, which in this case would have been the cash-strapped Russian Navy of the early 2000's. The risk associated with hydrogen peroxide propulsion of torpedoes were known and well documented, but the cost of removal or cleanup proved prohibitive. Rather than triggering an alert to scale back naval activity or decommission submarines containing H_2O_2, the danger was simply ignored.

People Trained, **Process** – for normal ops, and an emergency, **Tools** – to extinguish and mitigate the initial explosion also the **culture** – cavalier. Refusal to accept help from NATO Country until it was too late, system was not well maintained and could not mate with rescue sub

C-8.5 "Galloping Gertie," A.K.A., the Tacoma Narrows Bridge

Everybody's seen the iconic footage of this well-known engineering disaster.

How Systems Thinking would have helped: "Environment" is a key input to Systems Thinking, and in this case, that environment involved predictable forces acting on system components with enough force to cause structural failure. Cost concerns led to cutbacks on the initial design, which called for trusses that would have prevented the collapse. But failure to evaluate the interdependence of system components ultimately doomed the bridge

Cost – went cheap, Schedule --, Performance – failed. Verification of each component and unit

Science should have detected the physics of wind. Engineering would have applied physics to build a reliable structure.

Rhetorical Question – did someone know this would happen and they were ignored?

C-9 Case Studies – Rapid Fire - Discussion

C-9.1 Data Ingest

For discussion

- Project Planning
    - Requirements Management, Schedule, Risk/Issue/Opportunity, CM/DM, VER and VAL, Integration, Agreements, Transition ← may of these may be combined into a repeatable Standard Operating Procedure
- If this were AWS, what services would you consider?
    - EBS, Glacier,
- What type of Lifecycle Model is most suitable?
    - Start with Linier and a Project Plan with schedule), then circular
- What processes would you use for this project?
- How would you charge this customer for this activity? How would you figure this out?
    - Have a cost model – because you'll be repeating this.
    - Cost Model – initial set up payment, payment by the volume successfully ingested, monthly fee for O&M – break this in to CLINS
    - Cost Reimbursable to share the risk

C-9.2 Medical Systems

- 1Think – what are the requirements? Jot a few of them down.
    - The Patient
        - The user shall input their personal data
        - The user shall change their data.
    - The Doctor
        - The User shall read records of other doctors
        - The User shall input notes from the patient visit.
        - The User shall prescribe medicines
    - Pharmacist
        - The User shall have access to the doctor's notes

        - The User shall answer questions for the patient pertaining to the prescribed medicines.
    - Pharmacy Technician
        - The User Shall fulfill prescriptions
        - The User Shall read previous orders of the patient.
        - The User Shall charge the patient for the order.
        - The User Shall charge the insurer for the order.
- What are the Key Performance Requirements?
    - The orders must be accurate in the dosage, volume, frequency
    - The system will not be infiltrated, and data will remain private, and not altered.
- How could you protect this system, the data? Think Confidentiality, Integrity, Reliability
    - Confidentiality – data will remain private and accessible only by those authorized
    - Integrity – the data will not be changed by unauthorized people
    - Reliability – the system will have a Mean Up Time of 99.99 percent, and not be down for any more than five minutes at any one outage.
- What type of contract would you arrange?
    - Cost Plus if requirements change and its medium risk. If requirements are dynamic
    - Fixed Cost if very stable, can 'cookie cutter' the system (Large, Med, Small) or quantify the user demand and data demands.
- What type of lifecycle(s)?
    - Agile, but if the lifecycle is something of value to a potential customer make it simple and presentable
- What agreements are there?
    - Contract, Sub-Contract, an independent tester (third party) for VER and VAL, Procurement of material
- Address the Cost, Schedule, Performance and Risk topics.
    - Cost – see Contract Arrangement –
    - Schedule – put this in to a rhythm, and integrate with other customer activities
    - Performance – is paramount – don't go cheap
    - Risk – if data set is new the data extraction, transform and load may not work as well,
- Address the People, Process, Tools issue – what would you use,
    - People – users will need some training to access and use the system – which should not change that much if going from client server to a cloud, Technicians – trained in cloud to the needed level, observe and understanding leading indicators

C-9.3 Voting System in the Cloud

- What are (some) of the functional requirements?
    - Deliver a Voting System that may be used in local, state and federal elections. It must be: secure, confidential, provide voter anatomization, verify voter identity, allow one vote per voter for one candidate in each category, allow for quick tally but polling station, by county, by state.
- Give an example or two of a KPP.
    - Voter integrity, one vote per person, privacy, speed
- What are the requirements for Confidentiality, Integrity, Reliability?

    - The user shall not have their vote disclosed.
    - The user vote will not change once the vote is cast.
    - The user shall not experience a delay beyond one minute once the voting process has started.
- What Processes should you use?
    - All of them - Manager Project, Manage Schedule, Architecting and Design, Risk/Issue/Opportunity, Requirements Management, PDR, CDR, – plenty of VER and VAL
- What are the policy and technical implications of this system?
    - Policy – multi-day voting? Voting ID, Acceptance of Electronic Voting, Cultural Change that is encoded into policy
    - Technical - Equipment approved for voting (iPhone?), secure (encryption, authentication)
- Why is it called a System, and not a Voting Machine?
    - It includes multiple elements such as a voting booth/machine, the device that collects the votes at the polls, device the checks off a voter so the vote once, device the confirms the identify of a person, device that transmits the voter to the region/county, reporting data, security and encryption, etc
- Sketch a hierarchy of the voting system.
    - See above

Appendix D Glossary of Terms

Glossary of Terms	
Term	**Meaning**
"… ilities"	Developmental, operational, and support requirements a program must address (named because they typically end in "itility" – availability, maintainability, vulnerability, reliability, supportability, etc.). (2.8.1 Requirements)
Acceptance Test	A criterion agreed by two or more parties that if passed will allow a system, or a portion of, to be transferred from the supplier to the acquirer. For SED, this is most easily understood by the Vendor as supplier, the government as the acquirer, and the test as an operations test. (ATR, IATR)
Agile	An adoptive development method using a composite team of developers, user representative, team lead, and testers with a backlog of user stories, time lined into scrums and sprints. Use when requirements not fully developed and can adapt to changing requirements. For SED type projects, this is mostly applied in the form of agile software development, User Stories and backlogs, scrums, increments, scrum master, etc. (2.2.4 Agile Development Method)
Alpha Version	a version of a piece of software that is made available for testing, typically by employees of the company that is developing it, before its general release.
Application Programming Interface (API)	The agreed value passes between software programming elements (routines, data-structures, objects, variables, calls, sub-routines, functions, etc.). Well-defined APIs enable easier passing of values and communications for software development and functionality, and provide a solid building block for scaling software programs. The definition of the APIs is commonly delivered with the software. APIs are under configuration management. Also, see Interface control Document (ICD). (IATR)
Approval to Operate (ATO)	ISSM's approval to operate the system – see ICD 503.
Artifact	A product of the Systems Engineering process that has a distinct characteristic. Artifacts are identified and usually require some form of control or observation, to include configuration or document management. Examples of an artifact are drawings, documents, physical interfaces, parts, components. (3.1.8. CM/DM)
Baseline	The approved set of configuration items and their value. For example, the user requirements are base lined so only approved changes are made, and so the stakeholders know which baseline to use.
Basis of Estimate (BOE)	A detailed description of hours and cost of performing a task, usually in a proposal. A BOE is completed according to how a WBS is created.

Glossary of Terms	
Term	**Meaning**
	The BOE will include the type of labor, duration, amount of labor in hours. There are differing methods to creating a BOE, such as parametric or bottoms up.
Black Hat	A review in which a team of subject matter expert's role play as the opponent, in an effort to find flaws or weaknesses in the team approach to a system, an operation or a project. The review team will debrief the stakeholders with their observations and recommendations. The Black Hat usually occurs between the Pink Team and the Red Team.
Change Management	A formal board that approves proposed changes to a system or project, to include how the changes are made.
Configuration Control	See Configuration Management
Configuration Item (CI)	An artifact such as a drawing, requirements, an interface, test plan or any artifact that should it change (version or definition) would cause considerable harm to the system or its operation. The CI may define the "Form, Fit or Function" of a system or its parts. Therefore, these artifacts require control and only approved changes are made. See Baseline.
Contract Office Technical Representative (COTR)	The person responsible for the technical execution of the contract and who provides oversight and direction to the Vendor. The COTR is usually the Project Manager.
Control Gates	A decision point that comes from a set of work, in which the applicable level of authority makes the decision to proceed, pause, stop the project or re-do a portion of (or all) the proceeding tasks of a phase. The Gate Review contains Gate Slides with criteria that must be met before proceeding.
Critical Path (Schedule)	The longest set of dependent tasks throughout the schedule. A delay in any one of the tasks in the critical path will cause a schedule delay. These critical path tasks should be subject to close scrutiny so they do not pose a risk that would delay the deliver or increase the cost of the system. Upon discovering the critical path, Systems Engineers must be astute to the "secondary" critical path, which may cause an additional and likely not observed risk. (See Lag Time, Float Time and Lead Time)
Decision Package	The documented result of a Milestone Review that is approved by competent authority.

Glossary of Terms	
Term	**Meaning**
Decompose or Decomposition	Breaking a problem down into smaller problems, to the point that each smaller problem is discrete, measurable, and expressed in the form of a user requirement. (Systems Vee, REQ)
Defect Report (DR)	Documenting an anomaly of a problem with the system, assigning a unique identifier, and managing the defect to resolution. Any stakeholder may submit DRs during any phase of the lifecycle. DRs become more prevalent during the Integration and Test Phase but continue through O&M to Retirement of the system. Defects may also be called Deficiencies. (STR/ATR)
Deficiency Report	See Defect Report
Deliverables	A task provided to a second party, the recipient, as agreed upon in format and characteristics. A deliverable is usually a task with supporting documentation that is stated in the contact, and provided from the vendor to a customer. It may be a Contract Deliverable (CDRL)
Department of Defense Architectural Framework (DoDAF)	A set of architectural drawings from various "viewpoints" that join needs of all stake holders in a set of composite drawings. Drawings include the Operation View, the Network View, and the Data View. (4.1.1. DODAF)
Dependencies	In general terms, a dependency is an action that must be completed before the subsequent action is completed or started. In scheduling, the dependencies may be start to start, or finish to start. Systems that receive inputs from other systems also have dependencies of those other (external) systems.
Engineering Change Proposal (ECP)	A request by a stakeholder to make a technical change to a system. This is synonymous to a Defect Report until Phase 2 of SELP. ECPs are common language used by most vendors.
Float Time (Slack Time)	The amount of time a scheduled activity may be delayed without affecting the schedule of any subsequent tasks. (See Lag Time, Critical path and Lead Time)
Gold Team	A review usually in the form of an in-brief and review of a project or an operation, to validate the approach, and to provide first hand understanding to the reviewers. Reviewers will provide feedback and tips, but ultimately the Gold Team's purpose is to approve the activity. The Gold Team is the final review of the color reviews. (See Red Team, Black Hat)
Incremental	Completing a set of tasks or activities one step at a time, only proceeding when the one step is completed.
Initial Operation Capability (IOC)	Approval to start using the system. IOC is a milestone (called Review) but has no tasks except to dispatch a cable as directed by the CLB to

Glossary of Terms	
Term	**Meaning**
	the stakeholders to declare that they may use the system. (See Full Operational Capability [FOC])
Integration	A collection if interfaced components or elements each presenting their emergent behavior to the greater system. Interfaces make Integration possible.
Interface Control Document (ICD)	A document that describes the interfaces(s) to a system or subsystem such as the inputs and outputs of a single system or the interfaces between two systems, subsystems, or other elements of a system. The ICD is a key artifact in Systems Engineering because it defines and controls data passed between two parts of a system, or between two systems. The ICD may refer to software (often expressed in an API), hardware, data structures, firmware, or other type of system element. It may define software values or hardware values (electrical physical connections, etc.). An ICD does not describe the behavior of characteristics of the information it is passing, only the values, format, etc. IEEE and ITU often express universal interfaces that may be called ICDs. ICDs are base lined and under CM. (IATR)
Issue	The output of an action in which an event that will cause system or project failure must be resolved A risk that has occurred and must now be triaged or treated urgently to prevent further damage to the project or the endeavor. Issues are to be documented and managed (usually aggressively) to closure.
Iteration	Breaking a problem down to its basic form and solving the problem at each discrete level. See decompose. In software, iteration is repeating the same process until a defined condition (If, Then, Else) or (Until).
Information Technology Service Management (ITSM)	policies, organized and structured in processes and supporting procedures – that are performed by an organization to design, plan, deliver, operate and control information technology (IT) services offered to customers.[1] (Wikipedia ITSM, 2020) (4.2.8. ITSM)
Lag Time	Upon completion of one scheduled task, the subsequent dependent task is delayed a set amount of time. This allows resources to be applied to other tasks, or for a task to be completed in

Glossary of Terms	
Term	**Meaning**
	synchronization with other tasks. (See Critical Path, Float Time and Lead Time)
Lead Time	Starting a task earlier then needed, that is not the critical path. Lead time is an opportunity to accelerate the scheduled completion of the project (See Lag Time, Float Time and Critical Path)
Life Cycle	A series of phases with embedded milestone reviews. The Lifecycle starts with conceptualizing the system, documenting the requirements, (decomposing and integration), designing and developing the solution, testing and fielding, then decommissioning and disposing when no longer needed.
Market Research	Conducting extensive study of a problem and how it may be solved, usually by vendors. The problem statement is usually of wider issues to be resolved than the normal routine problem from a market survey. Market research is an effort to find new vendors, new solutions, and innovation that has not yet been discovered.
Market Survey	Conducting a limited set of research usually with a defined scope and with a set of traditional providers. The market survey responses are evaluated for innovation, capabilities and rated accordingly.
Milestone	A decision point in the Lifecycle Process during which a set of criteria are met and approved by competent authority, before proceeding.
Performance	The speed to which a product performs, or the quantity to which it successfully operates, or the capacity of data stored or transported.
Pink Team	A peer review of subject matter experts, that validates and confirms the understanding of the problem related to the operation, project, or system, and may verify the approach taken to resolve the problem for the stakeholders. The peer review team will provide a debriefing with their findings. The Pink team is the first in a series of color reviews.
Recursive	The activity of repeating a previous step or task, as more information becomes known about the task.
Red Team	A peer review in which subject matter experts are briefed, and read the approach taken to a project or an operation, to validate the correctness or the approach. The peer review team will debrief the stakeholders early enough in the activity to implement and include any findings or recommendations. The Red Team is the second peer review of the color reviews.
Request for Information	A query to another party, usually a potential vendor or contractor, in which the person sending out the RFI is interested in learning capability of a technology and resources to perform a contract or project.
Requirements	A documented need for a stakeholder, usually the user or operator. Requirements may also be dictated by regulatory input such as

Glossary of Terms	
Term	**Meaning**
	security, frequency management, or governance. Requirements must be traced from the origin through development, the component to the test case, and the test results.
Rough Order of Magnitude (ROM)	As estimate, usually provided by a vendor, that expresses the resources that are needed to complete a project or a task. The ROM is usually expressed in the form of dollars but may also be in hours. An ROM should include the basis (past similar projects, bottoms up, quotes, parametric, etc.), and risks and assumptions.
Segment	Part of the system that has discrete requirements and interfaces, that is testable. This may be a sub-system, element or component.
Service Level Agreement (SLA)	An agreement between two separate parties to what level of services is expected, the process to resolve problems, and the measurements to evaluate performance. SLAs are usually related to technical performance in information systems where the recipient has a need and expectation for network connectivity, performance and backups, for which they pay the provider. SLAs are part of ITSM.
Statement of Work (SOW)	A document that describes the work to be performed by the Vendor, to include the expected deliverables, project elements and system elements. The SOW is considered section J of a Request for Proposal, and is written by the COTR.
System Requirements Specification	The composite and documented description of system requirements, interfaces, performance, functionality.
Work Breakdown Structure (WBS)	Decomposition of work, a set of tasks to a discrete and measurable level. Each descending level represents a more detailed set of work until the work is measurable and can be assigned a labor category. (PMBOK)

Bibliography

97th Congress of the United States of America. (1982, May 21). *The Telecommunications Act of 1982 (H.R. 5158, 97th Congress): Provisions and Controversies.* Retrieved from EveryCRCReport: https://www.everycrsreport.com/reports/IP182.html

Andrews, E. (2015, February 5). *9 Things You May Not Know About the Ancient Sumerians*. Retrieved from History Channel: https://www.history.com/news/9-things-you-may-not-know-about-the-ancient-sumerians

ANSI. (2020, May 9). *About ANSI*. Retrieved from American Nations Standards Institute: https://www.ansi.org/

Armstrong, P. (n.d.). *Blooms's Taxonomy*. Retrieved from Venderbilt University Center for Teaching: https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

ASQ. (2020, May 1). *American Society of Quality*. Retrieved from American Society of Quality: https://asq.org/

Assistant Secretary of Defense/R&E. (2011, April). Technology Readiness Level (TRL) (Technology Readiness Assessment (TRA) Guidance. The Pentagon, Virginia, USA.

Audrertsch, T. (2005). The Digital Ecosystesm Research Vision: 2010 and Beyond. *Semantic Scholar*.

Bell, J., & Defense, D. S. (2007, March 10). *Acquisition Notes.* Retrieved from http://acqnotes.com/wp-content/uploads/2014/09/DoD-4151.18-H-Depot-Maintenance-Capacity-Utilization-Measurement-%E2%80%93-10-Mar-2007-Change-1.pdf

Body, I. S. (2020). *ISO*. Retrieved from https://www.iso.org/home.html

Branson, R. (2016, March 17). *My top10 quotes on risk*. Retrieved from Virgin: https://www.virgin.com/richard-branson/my-top-10-quotes-risk

Brock, G. D. (1999). Trachtenberg School of Public Policy & Public Adminstration . *TCOM Class Presentation* . Virginia, U.S.A.: The George Washington University (Columbian College of Arts and Sciences).

Burge, D. S. (2011). *The Systems Engineering Tool Box.* Warwickshire: Burge Hughes Walsh.

C4 Model. (n.d.). *C4 Model for Visualising Software Architeture - Contezxt, Containers, Components and Code*. Retrieved from C4 Model: https://c4model.com/

Carrick, P. (2018, November 5). *Who Invented the Magnetic Compass?* Retrieved from https://historydaily.org/who-invented-the-magnetic-compass: https://historydaily.org/who-invented-the-magnetic-compass

CMMI Institute. (2020, May 9). *CMMI*. Retrieved from Capability Maturity Model Integration: https://cmmiinstitute.com/

Cockburn, A. (2007). *Agile Software Development - The Cooperative Game (2nd Ed) .* Boston: Pearson Education, Inc.

DAU. (2020, May 9). *About Us*. Retrieved from Defense Acquisition University: https://www.dau.edu/about

Defense Acquisition University. (2020, May). *Life Cycle Cost*. Retrieved from DAU: https://www.dau.edu/acquipedia/Pages/search.aspx?k=cost%20of%20doing%20system%20engineering#k=systems%20engineering%20cost

Deming, W. Edwards. (n.d.). *The W. Edwards Deming Institute*. Retrieved from https://deming.org/

(2013). *Deterministics vs. stochastic models.* Raleigh: NC State.

DoD CIO. (2010, August). *The DoDAF Architecture Framework Version 2.02.* Retrieved from U.S. Department of Defense - CIO: https://dodcio.defense.gov/Library/DoD-Architecture-Framework/

Engineering. (2020, June 26). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Engineering

Esper, M. (2020, May 6). FCC's Decision Puts GPS at Risk. *Wall Street Journal*, p. 14.

Estefan, J. A. (2008, May 23). *Survey of MBSE Methodolgies.* Retrieved from IAN Sommerville: https://iansommerville.com/software-engineering-book/web/spiral-model/

Freshworks. (2020, April). *What is ITSM?* Retrieved from Fresh Service: https://freshservice.com/itsm

Gates, B. (n.d.). Retrieved from BrainyQuotes: https://www.brainyquote.com/lists/authors/top-10-bill-gates-quotes

Georgiev, G. (2019, Nov 5). *"All Models Are Wrong" Does Not Mean What You Think It Means*. Retrieved from TheStartup: https://medium.com/swlh/all-models-are-wrong-does-not-mean-what-you-think-it-means-610390c40c9c

Georgiev, G. (2019, Nov 5). *The Start Up*. Retrieved from Medium: https://medium.com/swlh/all-models-are-wrong-does-not-mean-what-you-think-it-means-610390c40c9c

Glenn, C., & Gray, L. (2010). *The Hodges Harbrace Handbook (17ed).* Boston: Wadsworth.

Hersey, P., & Blanchard, K. (1969). *Management of Organizational Behaviour: Utilizing Human Resources.* Englewood Cliffs: Prentice-Hall.

Hirshorn, Steven R. (2007). *NASA System Engineering Handbook.* Washington, DC: National Aeronautics Space Administration .

History.com Editors. (2020, February 21). *History Channel*. Retrieved from Enlightement: https://www.history.com/topics/british-history/enlightenment

Honour, E. C. (2013). *Systems engineering return on investment.* Adelaide: University of South Australia - Defece and Systems Institute - School fo Electrical and Information Engineering.

IBM. (2020, May 11). *Overview of Rational DOORS*. Retrieved from IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/SSYQBZ_9.5.0/com.ibm.doors.requirements.doc/topics/c_welcome.html

IDEF - Integrated DEFinition Methods (IDEF). (2020, April). Retrieved from IDEF0 Funtion Modeling Method: https://www.idef.com/idefo-function_modeling_method/

IDEF. (n.d.). *IDEF Family of Methods*. Retrieved from Integrated DEFinition Methods (IDEF): https://www.idef.com/

IEEE. (2020, April). *Institute of Electrical and Electronics Engineers*. Retrieved from IEEE Advancing Technology for Humanity: https://www.ieee.org/

INCOSE. (2020, May 9). *INCOSE - About Systems Engineering*. Retrieved from International Council on Systems Engineering: https://www.incose.org/about-systems-engineering/about-systems-engineering

Innovative Architects. (2020, July 10). Retrieved from http://local.innovativearchitects.com/

International Council on Systems Engineering (INCOSE). (2011). *Systems Engineering Handbook - A Guide for System Life Cycle Processes and Activities.* San Diego: SE Handbook Working Group.

International Council on Systems Engineering (INCOSE). (2011). Systems Engineering Handbook - A Guide for System Life Cycle Processes and Activities. In I. SE Handbook Working Group, *Systems Engineering Handbook* (p. 5). San Diego.

International Council on Systems Engineering (INCOSE). (2020, April 4). *Collaboration Portal "Connect"*. Retrieved from https://www.incose.org/

ISO. (2020, May 9). *ISO Home*. Retrieved from International Council on Systems Engineering: https://www.iso.org/home.html

ISO 9000. (2015). *International Standards Organization (ISO)*. Retrieved from ISO 9000 Family Quality Management: https://www.iso.org/iso-9001-quality-management.html

ISO/IEC. (2020, May 11). *ISO/IEC 27001 Information Security Management*. Retrieved from International Standards Organization: https://www.iso.org/isoiec-27001-information-security.html

ISO/IEC/IEEE 42010:2011. (2017). *International Stndards Organization*. Retrieved from Systems and Software Engineering - Architecture Descriptions: https://www.iso.org/standard/50508.html

ITU. (2020, May 9). *About ITU*. Retrieved from International Telecommunication Union (ITU): https://www.itu.int/en/Pages/default.aspx

Kant, I. (1996, June 28). *What is Enlightenment.* Retrieved from Columbia University - Sources of Medieval History: http://www.columbia.edu/acis/ets/CCREAD/etscc/kant.html

Knighton, A. (2017, Nov 19). *How The Spear Transformed Warfare - From Ancient Times To The Age of Gunpower*. Retrieved from War History Online: https://www.warhistoryonline.com/instant-articles/spear-transformed-warfare-mm.html

Kobren, B. C. (2017, May 11). *Updaged DoD Acquisition Life Cycle Wall Chart*. Retrieved from Defense Acquisition University: https://www.dau.edu/training/career-development/logistics/blog/Updated-DoD-Acquisition-Life-Cycle-Wall-Chart

Le Merle, M. C., & Davis, A. (2017). *Corporate Innovation in the Fifth Era.* Corte Madera, CA: Cartwright.

Madhavan, G. (2020, May 5). The COVID Recovery Comes Down to Engineering. *Wall Street Journal*, p. 14.

Magee, John F. (1964). *Decision Trees for Decision Making*. Retrieved from Harvard Business Review: https://hbr.org/1964/07/decision-trees-for-decision-making

McConnell, John (V Adm). (2008, Sept 15). *ICD 503.* Retrieved from Director of National Intelligence ICD 503 Intelligence Community Information Technology Systems Security Risk Managment, Certification And Accreditation: https://www.dni.gov/files/documents/ICD/ICD_503.pdf

Monat, Jamie; Gannon, Thomas. (2018, Sep 12). *Applying Systems Thinking to Engineering and Design.* Retrieved from Worcester Polytechnic Institute : https://www.mdpi.com/2079-8954/6/3/34

National Institute of Standards and Technology. (n.d.). Retrieved from Digital Ecosystem - Search Results: https://www.nist.gov/fusion-search?s=digital%20ecosystem&start=30&sort=relevance

NCSU. (2013). *Deterministic vs. stochastic models.* Retrieved from NS State University - Dept of Statistics: https://www4.stat.ncsu.edu/~gross/BIO560%20webpage/slides/Jan102013.pdf

NPCA. (2010, March). *Corrective Action and Root Cause Analysis*. Retrieved from Slideshare: https://www.slideshare.net/

OD&I. (2008, September 15). *OD&I Documents.* Retrieved from Office of the Director of National Intelligence: https://www.dni.gov/files/documents/ICD/ICD_503.pdf

PMI. (2020). *PMI Entry Page*. Retrieved from Project Management Institute: https://www.pmi.org/

Santayana, G. (n.d.). *Wikiquote*. Retrieved from George Santayana: https://en.wikiquote.org/wiki/George_Santayana

Schulze, E. (2019, Jan 17). *CNBC*. Retrieved from Everything you need to know about the Fourth Industrial Revolution: https://www.cnbc.com/2019/01/16/fourth-industrial-revolution-explained-davos-2019.html

Smith, A. (1776). *The Wealth of Nations.* Scotland: Strahan, William; Cadell, Thomas.

Strickler, Kris. (2020, July). *Oregon.gov*. Retrieved from Oregon Department of Transportation: https://www.oregon.gov/odot/About/Pages/Library.aspx

Surg, J. C. (2014, January 25). *US National Library of Medicine, National Institues of Health.* Retrieved from PubMed: https://www.ncbi.nlm.nih.gov/pubmed/24406559

SysML. (n.d.). *What is SYSML*. Retrieved from Object Managment Group (OMG): https://www.omgsysml.org/index.htm

Szalay, J. (2018, September 18). *Who Invented Zero*. Retrieved from Live Science: https://www.livescience.com/27853-who-invented-zero.html

UML. (n.d.). *UML Profiles, Specifications, Training*. Retrieved from Unified Modeling Language: https://www.uml.org/

Union, I. T. (2020, April). *ITU - Committed to connecting the world*. Retrieved from https://www.itu.int/en/Pages/default.aspx

Vanderbilt University. (2020, March). *Center for Teaching*. Retrieved from https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

Wikipedia. (2019, November 27). Retrieved from Digital Ecosystem: https://en.wikipedia.org/wiki/Digital_ecosystem

Wikipedia. (2020, March 12). *Babylonian Mathmematics*. Retrieved from https://en.wikipedia.org/wiki/Babylonian_mathematics

Wikipedia. (2020, April 25). *Hubble Space Telescope*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Hubble_Space_Telescope

Wikipedia. (2020, April 17). *Reductionism (ontological)*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Reductionism

Wikipedia. (2020, February 21). *United States Military Standard*. Retrieved from WIKIPEDIA: https://en.wikipedia.org/wiki/United_States_Military_Standard

Wikipedia ITSM. (2020, April 8). *IT Service Management*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/IT_service_management

Wikipedia Six Sigma. (2020, March 15). *Wikipedia*. Retrieved from Six SIgma: https://en.wikipedia.org/wiki/Six_Sigma

Woolsey, James. (2020, May 7). *About DAU*. Retrieved from Defense Acquisition Universty (DAU): https://www.dau.edu/about

Zackman, J. A. (2008). *The Concise Definition of The Zackman Frameworks by: John A. Zachman.* Retrieved from Zackman International Enterprise Architecture: https://www.zachman.com/about-the-zachman-framework