ACE 101

Introduction to Agile

Agile Center of Excellence

v_4.0

Welcome!

- In the Google Meets Chat
  - Share 1 thing you liked about the 'Stay at Home'
    - New Hobby
    - New Book
    - New Movie

Introduction

- About Me…
  - Founded ACE in 2017
  - PMP
  - CSPO
  - PSM
  - SPC4
  - Lean Green Belt
  - Proud Daddy!

Agreements

- Resources
  - Slide Deck
  - Jamboard
  - Main Meeting
- Breakouts - Teams / PoC
  - Team 1
  - Team 2
  - Team 3
  - Team 4
- Agreements
- Breaks
- Materials
- Parking Lot
- Retro

Intent and Assumptions

Intent

- This material has been created to establish a common language and framework for all stakeholders to communicate and facilitate the movement of work from ideation to completion in a consistent and efficient manner

Assumptions

- Individuals may perform multiple Roles to ensure all Responsibilities are fulfilled
- Team discretion should be used at all times

Agenda and Objectives

- Project Management Frameworks
  - Waterfall
  - Agile
  - Kanban
  - Scrum
  - Lean
- ACE Playbook 2.0
  - Glossary
  - Roles
  - Ceremonies

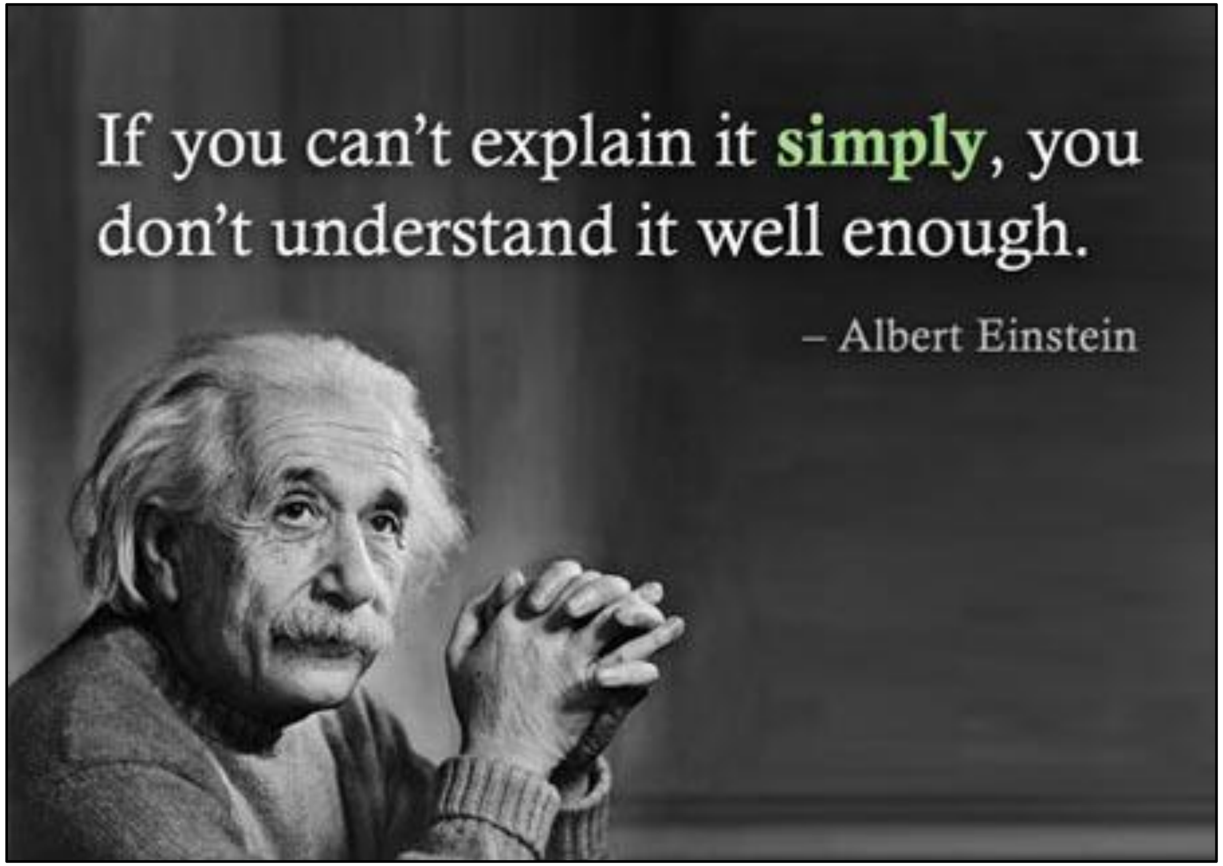
Project Management Frameworks

Agile Center of Excellence

v_1.0

Project Management Frameworks *

- Waterfall
- Agile Manifesto
- Kanban
- Scrum
- Agile @ DISH

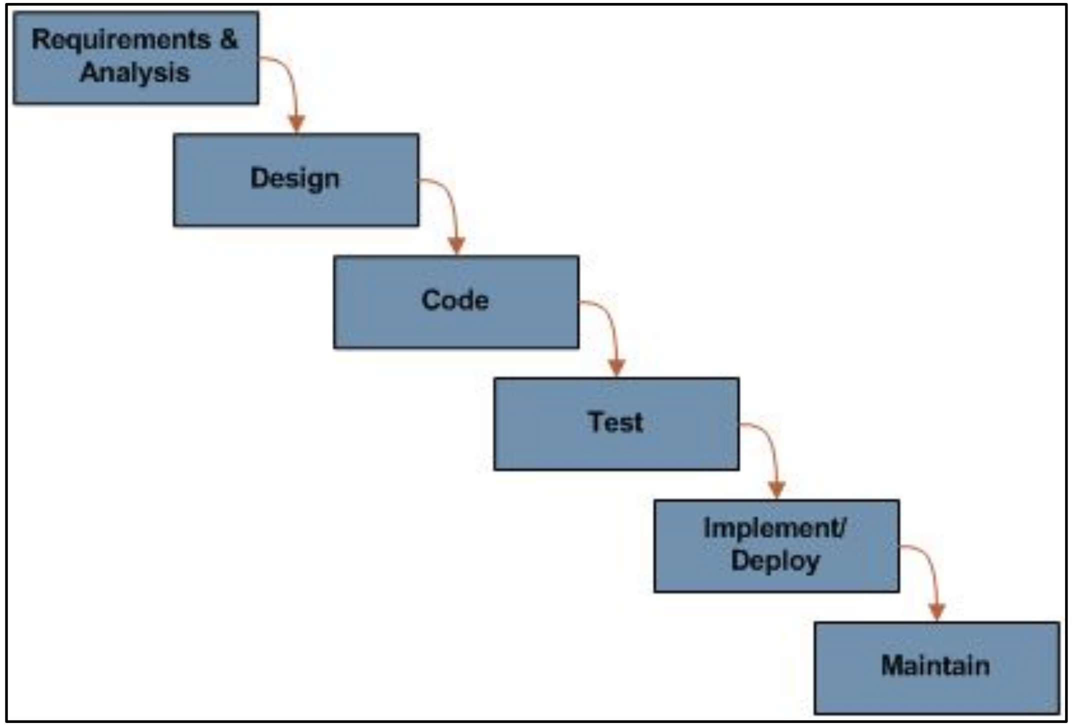

If you can't explain it simply, you don't understand it well enough.
– Albert Einstein

* **Framework** – A system of rules, ideas, or beliefs that is used to plan or decide something

* https://dictionary.cambridge.org/dictionary/english/framework

Waterfall

- Created in 1956 for manufacturing and construction industries
- Sequential, non-iterative process
- Work passes through independent phases before moving to the next phase
- Requirements are known up front
- Limited changes once work begins
- Why is this valuable?
- Where would this be used?



03 Minutes

Agile Manifesto *

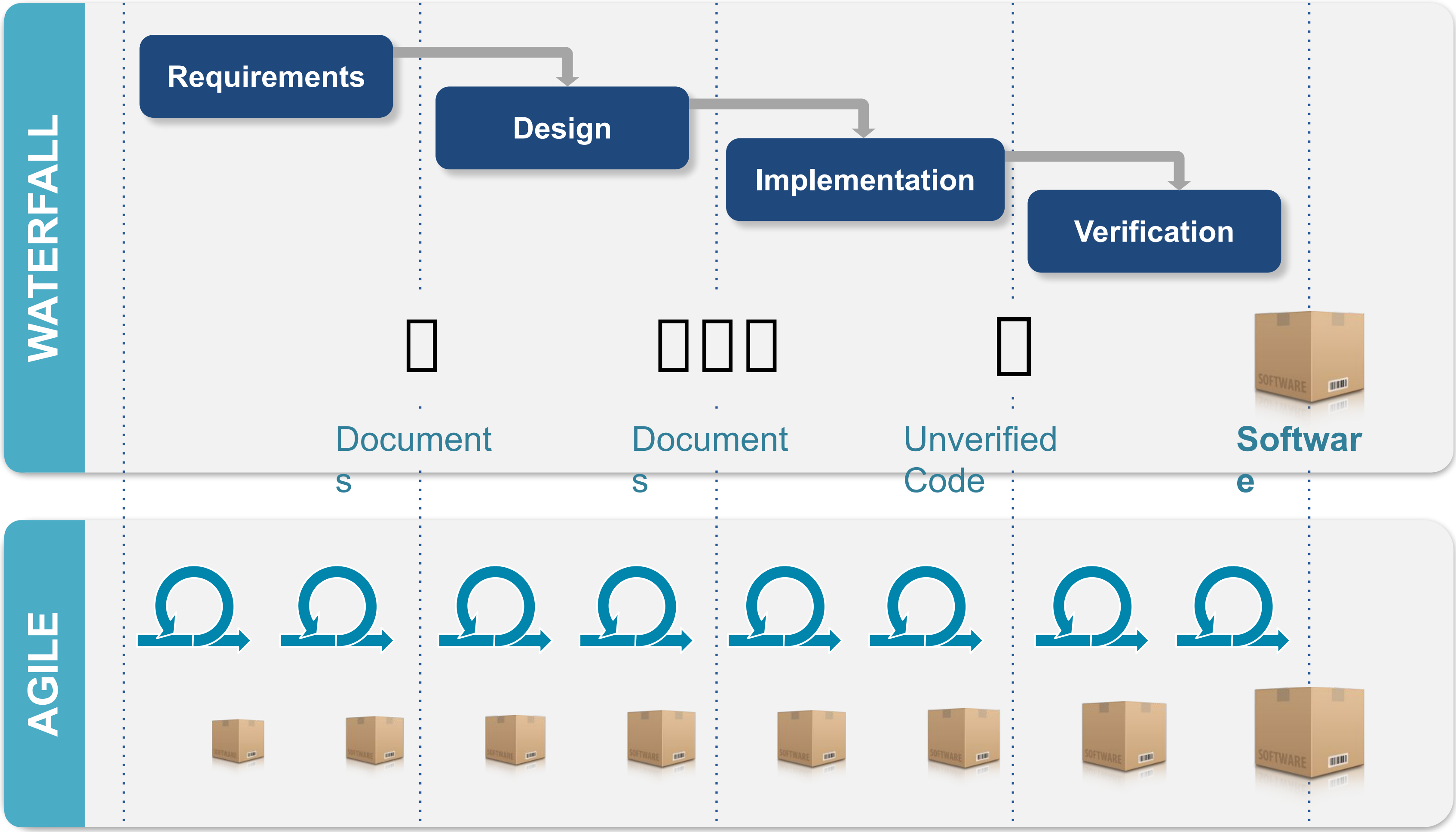
Individuals and Interactions	Over	**Processes and Tools**
Working Software	Over	**Comprehensive Documentation**
Customer Collaboration	Over	**Contract Negotiation**
Responding to Change	Over	**Following a Plan**

While there is value in the items on the right,
We value the items on the left more

* agilemanifesto.org

03 Minutes

What is Agile



WATERFALL

Requirements

Design

Implementation

Verification

Documents

Documents

Unverified Code

Software

AGILE

Agility is the ability to continuously adapt and make improvements to the way you work

* agilemanifesto.org

03 Minutes

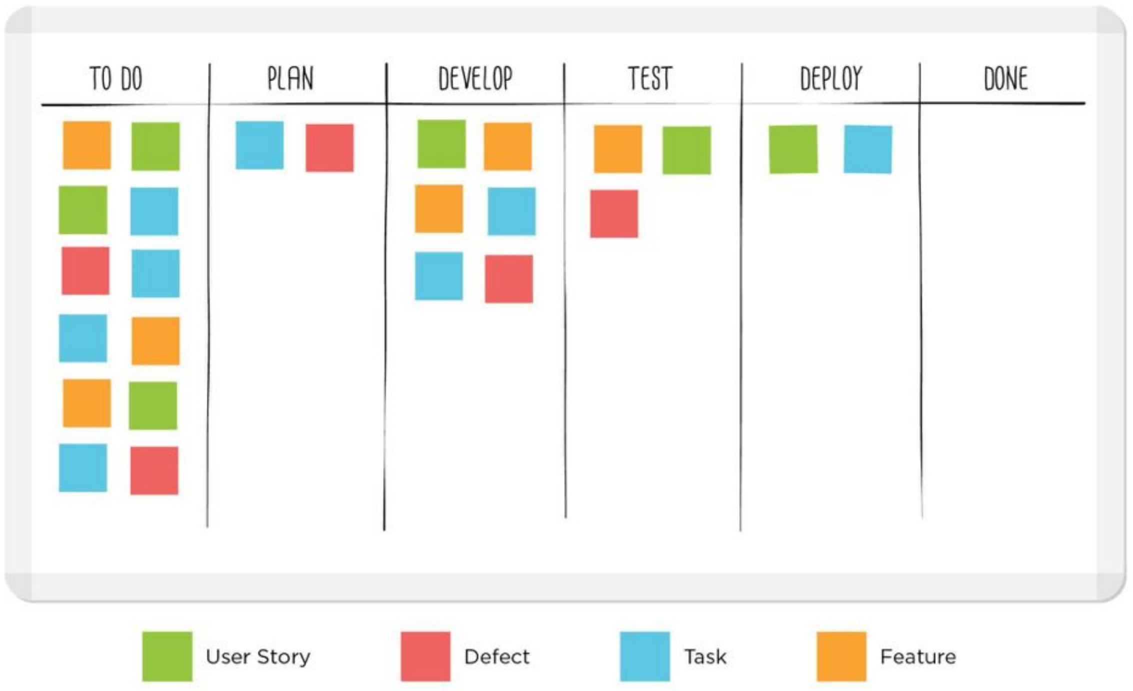
EXERCISE ONE
Kanban Boards

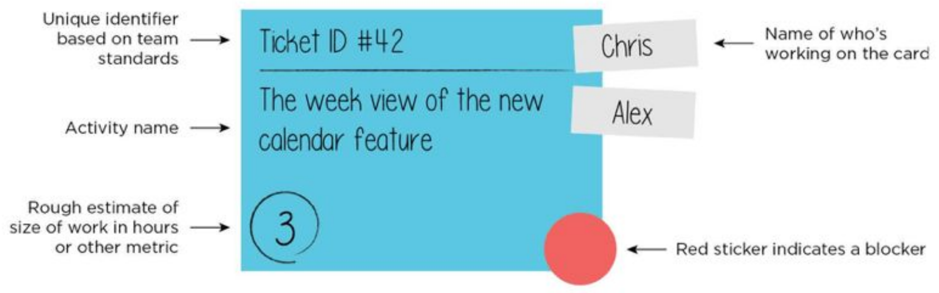
Agile Center of Excellence

v_1.0

Kanban

- 5 Principles
  - Visualize Workflow
  - Make Policies Explicit
  - Limit Work in Progress (WIP)
  - Measure Flow
  - Improve Collaboratively





Unique identifier based on team standards

Activity name

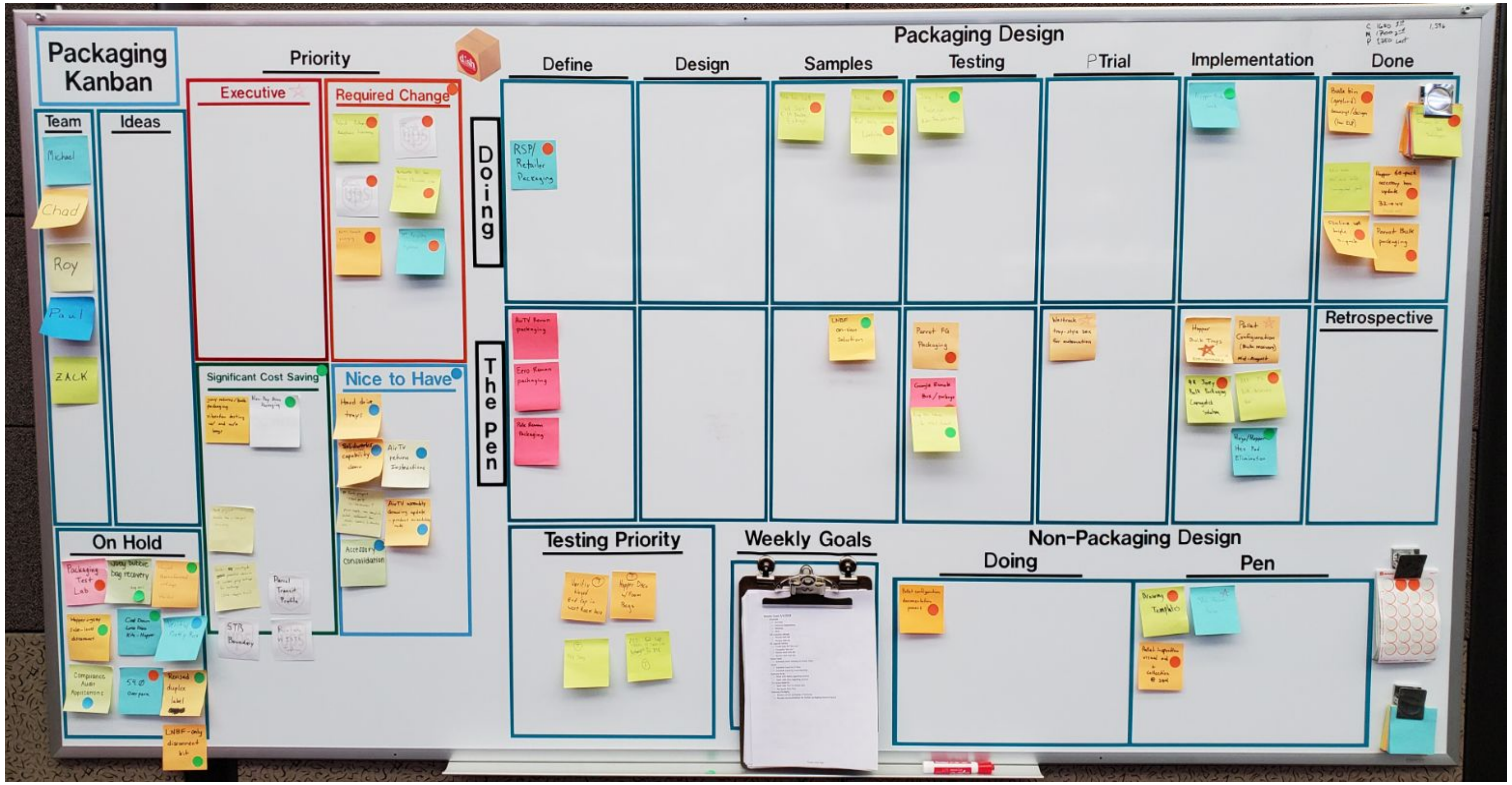
Rough estimate of size of work in hours or other metric

Name of who's working on the card

Red sticker indicates a blocker

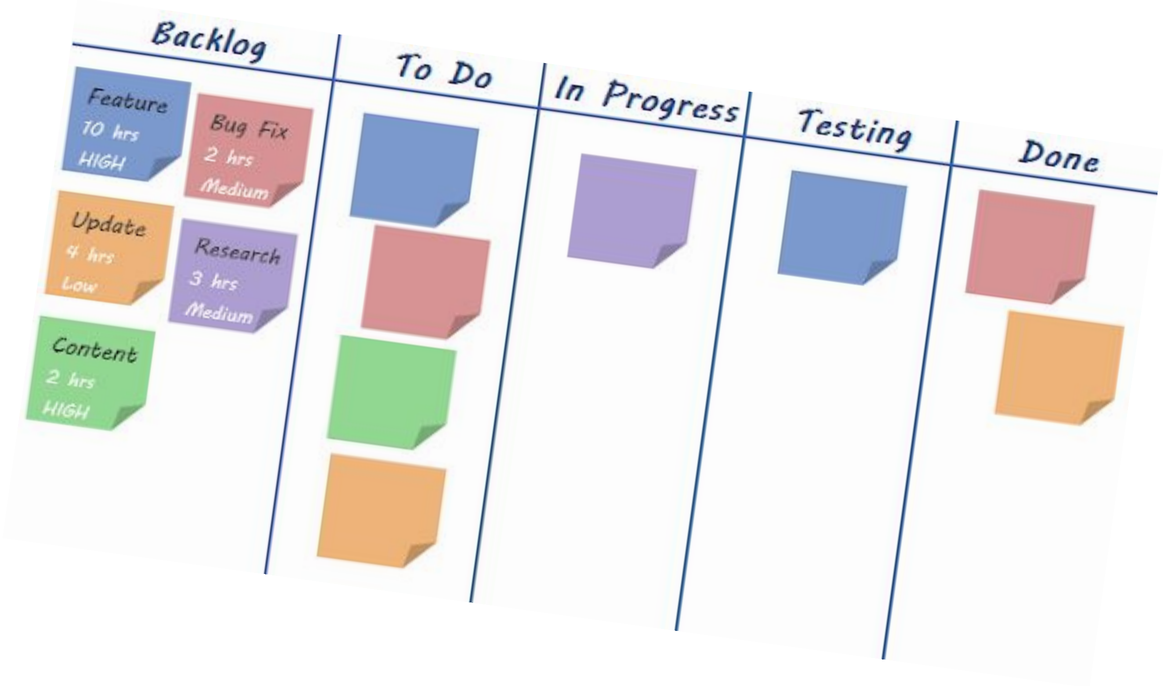
Why Kanban

- Improved communication through transparency
- Limit WIP in order to increase output
- Allow iterative, incremental refinement of the backlog



Building a Kanban Board

1. What are the possible states those tasks can be in
   - These start to establish the columns
2. Are there any process steps related to those states
   - This completes the columns



- Basic Kanban Board Outline

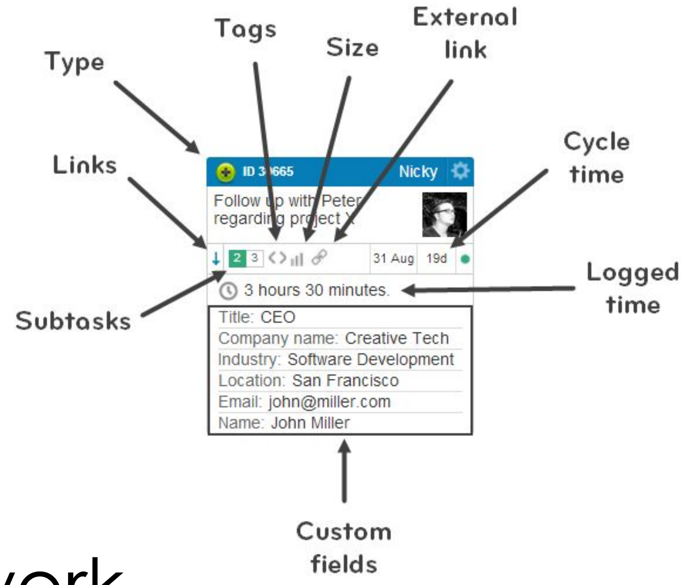
Building a Kanban Board

1. Identify Exit Criteria
   - What is required to move from one step to the next
2. Add queues
   - Forms the basis of the pull system
3. Define Work In Progress (WIP) limits
   - Maximum number of work items allowed at any given time

- Kanban Flow

Building a Kanban Board

- Identify Kanban Card fields
  - What information is important to capture for each work item
    1. Title
    2. Description
    3. Owner
    4. Resources required
    5. Definition of Done / Acceptance Criteria
    6. Other custom fields
    7. Color-coded
    8. Blockers



ID 3[illegible]665 Nicky

Follow up with Peter regarding project X

2 3 31 Aug 19d

3 hours 30 minutes.

Title: CEO
Company name: Creative Tech
Industry: Software Development
Location: San Francisco
Email: john@miller.com
Name: John Miller

- Define ranking system on which to prioritize work
  - Value, time to complete, resources / cost required
- Kanban Card Template

05 Minutes

Kanban Board - Options

1. Add swim lanes
   - By Owner
   - By Work Item Type
   - By Project
2. Add Express Lane
   - What is the toll
3. Holding Area
4. Cancelled Column

Using a Kanban Board

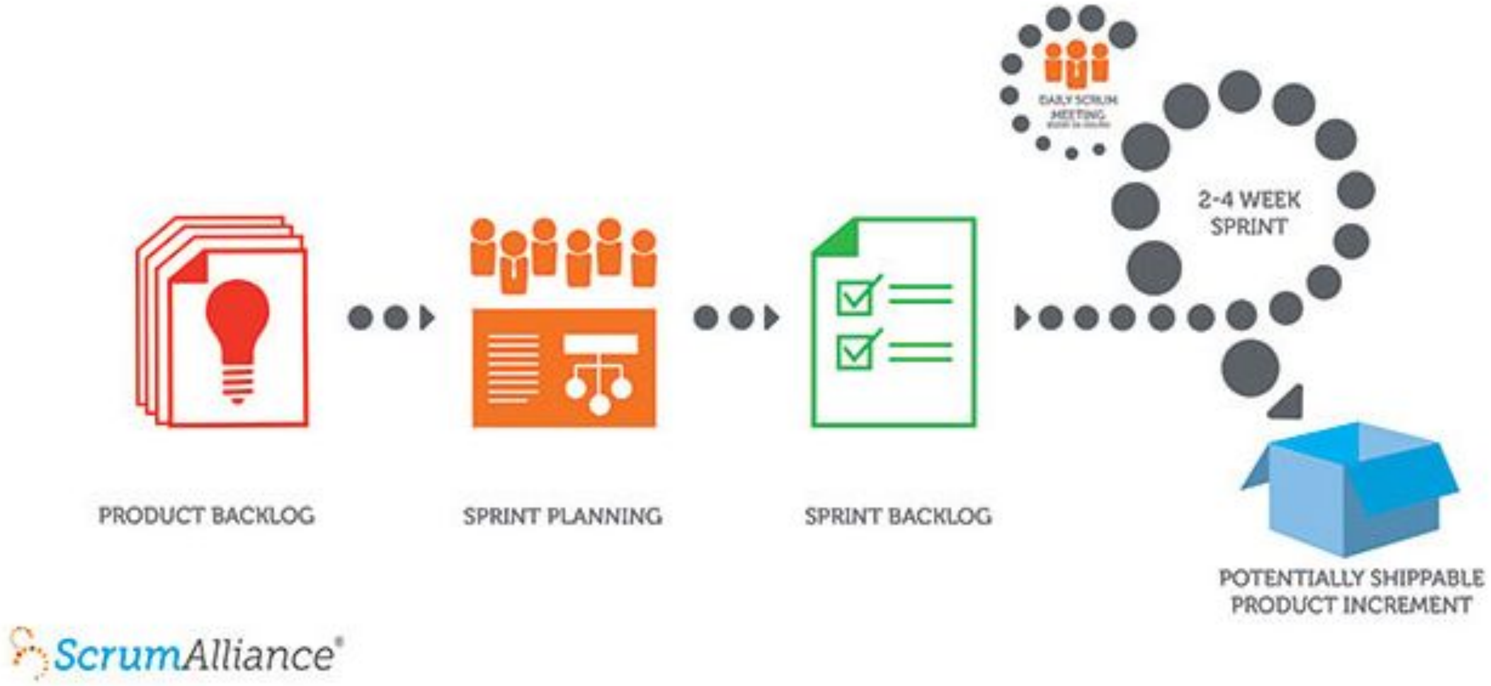
- Hold Planning Meetings
- Stack-rank and prioritize work items
- Identify blocking issues or bottlenecks
  - Stop starting and start finishing!
- Review and update frequently
  - Daily Stand-Up

Break Time

10 Minutes

Scrum

- Definition of Scrum
- Empirical Process Control
- Scrum Values
- Scrum Team
- Scrum Events
- Scrum Artifacts
- Activity
- Next Steps



Definition of Scrum

- What Scrum is and isn’t
  - Scrum Simplified
  - Scrum is lightweight and easy to understand, but when you dig in you will find that it is difficult to master.
  - Scrum is a framework of pillars, values, roles, events, and artifacts, **NOT** a specific process
  - Scrum does not tell you how to do things, it tells you what needs to be done and lets you figure out how to do it.
  - To make it even more confusing, Scrum is not literal; you must modify what it says to match your circumstances.
  - Scrum is a well-balanced framework, all its parts are needed in order to be effective.

Empirical Process Control (3 Pillars)

- Transparency
  - Giving visibility to the significant aspects of the process to those responsible for the outcomes *
  - Make work visible (ex. – Kanban Boards, Burndown Charts)
- Inspection
  - Timely checks on the progress towards a Sprint Goal to detect undesirable outcomes *
- Adaption
  - Adjusting a process as quicky as possible to minimize any further deviation or issues *
  - Constantly try small experiments, fail fast, continuously improve

* https://www.visual-paradigm.com/scrum/how-to-maintain-transparency-in-scrum/

Scrum Values *

- Commitment
  - People personally commit to achieving the goals of the Scrum Team
- Courage
  - Scrum Team members have the courage to do the right thing and work on tough problems
- Focus
  - Everyone focuses on the work of the Sprint and the goals of the Scrum Team
- Openness
  - The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work
- Respect
  - Scrum Team members respect each other to be capable, independent people

(1) https://www.scrumalliance.org/learn-about-scrum/scrum-values

Scrum Team *

- Scrum Team
  - Self-organizing, cross-functional team that iteratively and incrementally delivers product to maximize opportunities for feedback
- Product Owner
  - Responsible for maximizing value of Product
  - Sole owner of and accountable for the Product Backlog
- Scrum Master
  - Servant-leader for the Scrum Team, responsible for promoting and supporting Scrum theory, practices, rules, and value to internal team members and external stakeholders
- Development (Dev) Team
  - Professionals who do the work of delivering potentially releasable Increments of product at the end of each Sprint

* Scrumguide.org

Scrum Events *

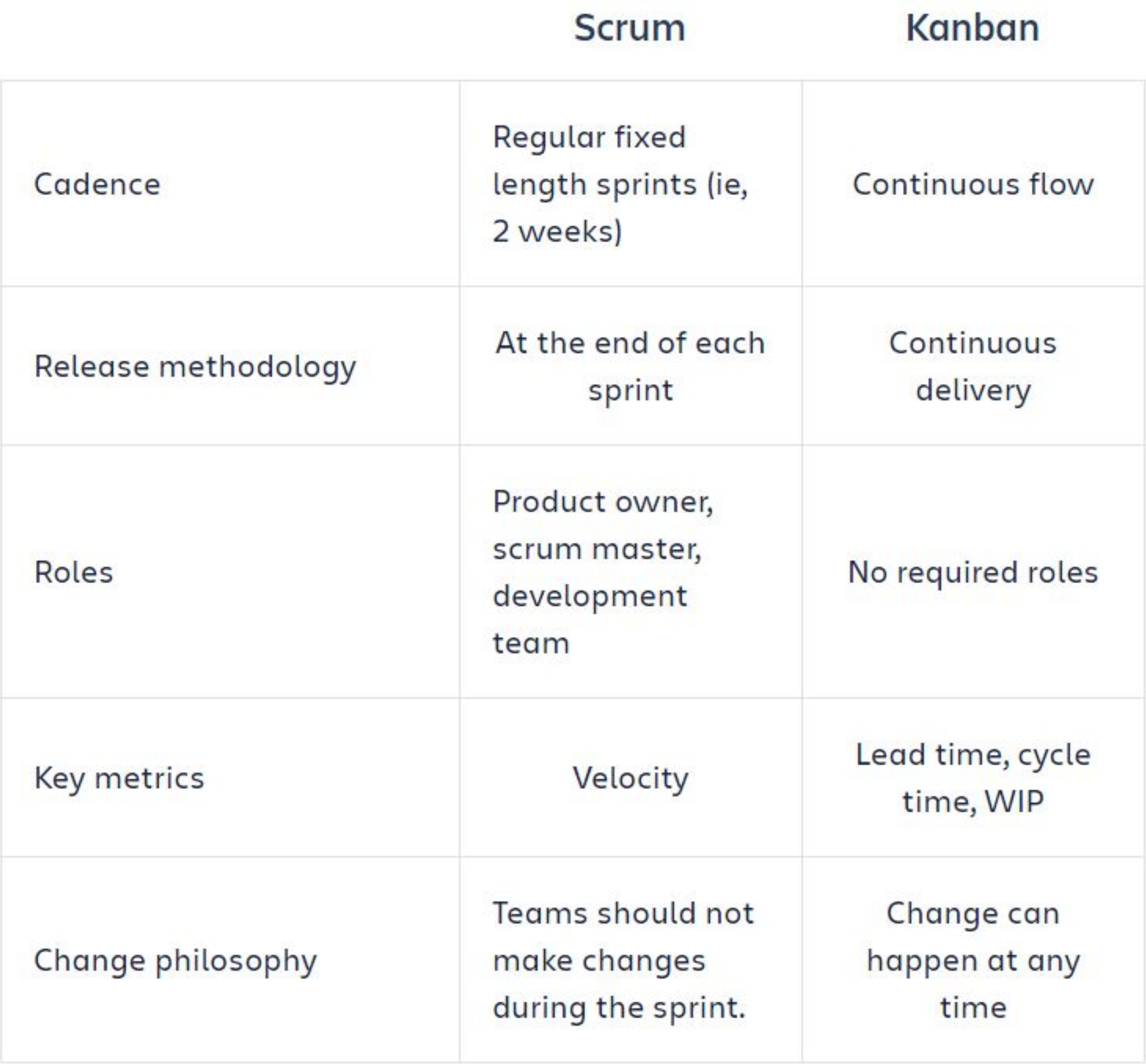
- Sprint
  - Time-box during which a usable, releasable, product Increment is created
- Sprint Planning
  - Event the defines what can be achieved in the upcoming Sprint in the form of a Sprint Goal and a plan on how to achieve the *Sprint Goal*
- Daily Scrum
  - 15 minute time-box for Dev Team to identify impediments and improve communication
- Sprint Review
  - Scrum Team presents completed Increment to stakeholders for feedback
- Sprint Retrospective
  - Team inspects what went well and what could be improved from current Sprint to adapt in next Sprint

* Scrumguide.org

Scrum Artifacts

- Product Backlog
  - Dynamic, evolving list of all features, functions, requirements, enhancements, and fixes know to be needed in the product
- Sprint Backlog
  - Highly visible, real-time, sub-set of Product Backlog items selected for the Sprint, owned solely by the Dev Team
- Increment
  - Sum of Product Backlog items completed during a Sprint
  - At the end of a Sprint, an Increment must be is usable condition and meet the definition of Done

Kanban vs. Scrum

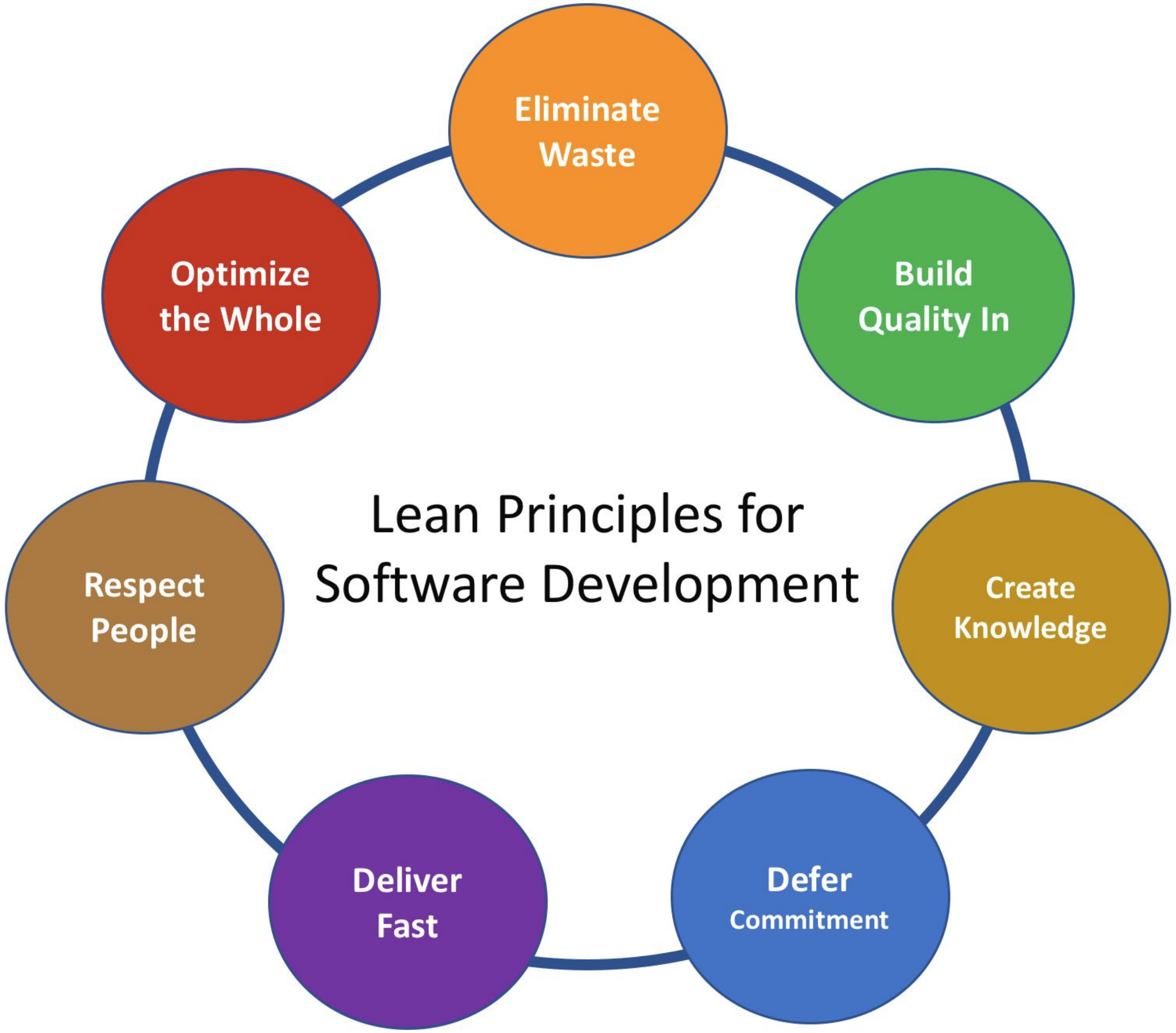


	Scrum	Kanban
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint	Continuous delivery
Roles	Product owner, scrum master, development team	No required roles
Key metrics	Velocity	Lead time, cycle time, WIP
Change philosophy	Teams should not make changes during the sprint.	Change can happen at any time

* agilemanifesto.org

Lean Software Development

- 7 Principles *
  - Eliminate Waste (TIMWOOD)
    - **T**ransportation
    - **I**nventory
    - **M**ovement
    - **W**aste
    - **O**verproduction
    - **O**verprocess
    - **D**efects
  - Build Quality In
  - Create Knowledge
  - Defer Commitment
  - Deliver Fast
  - Respect People
  - Optimize the Whole



* https://leankit.com/learn/lean/principles-of-lean-development/

ACE Playbook 2.0

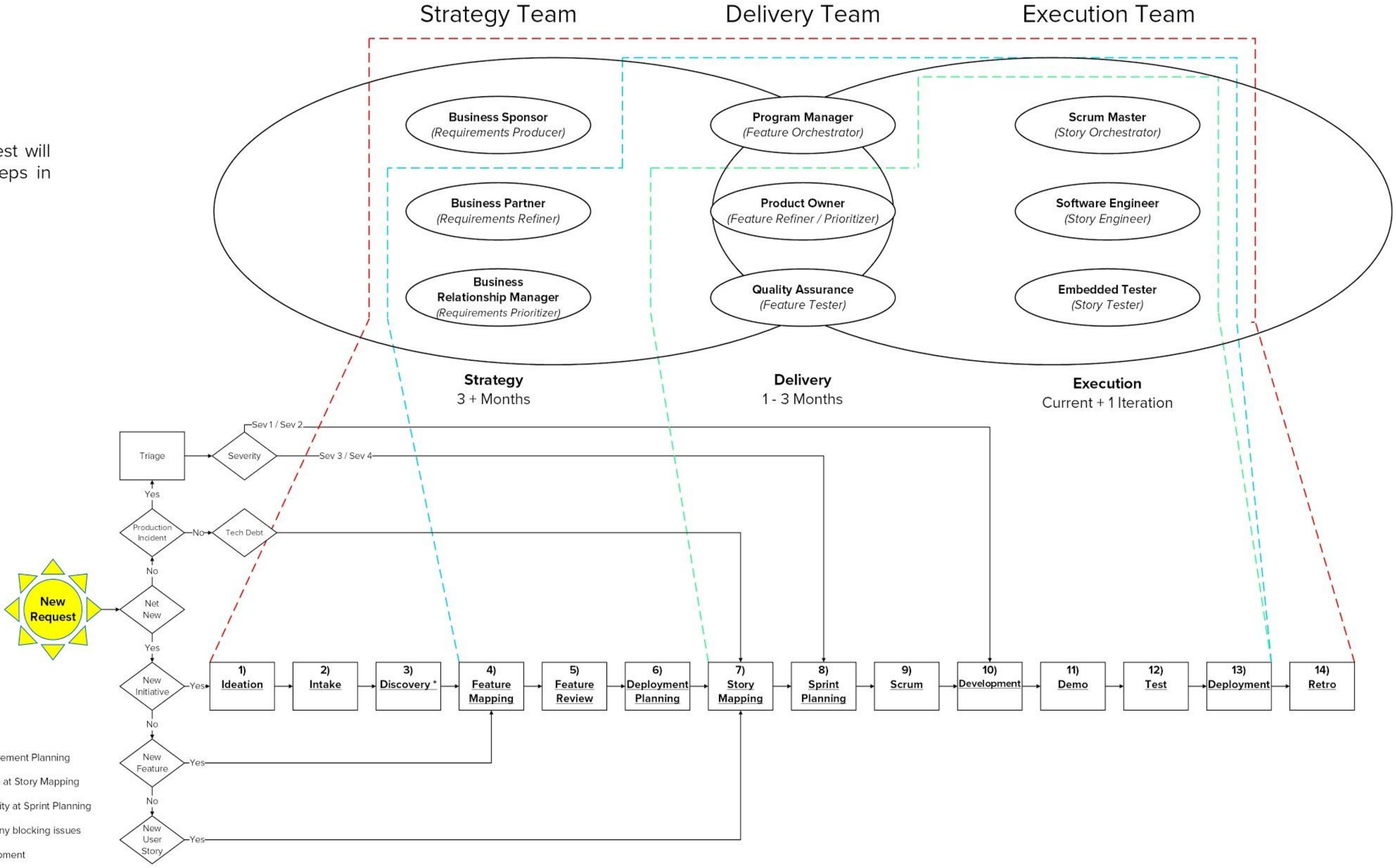
Index

Overview:

- Software Delivery Framework
- Glossary
- Roles
- Responsibility Matrix
- Ceremonies

Software Delivery Framework

The type, size, and complexity of a request will determine who is impacted and what steps in the process need to be followed



Legend

🔴 **New Initiative**
- Impact 1+ Product
- Impact 1+ Team
- Deployed in 1+ Milestone

🔵 **Existing Initiative**
New Feature
- Impact 1+ Team
- Deployed in 1 Milestone
- >300 Hours of Development

🟢 **Existing Initiative**
Existing Feature
New Story
- Impact 1 Team
- Deployed in 1 Milestone
- >20 Hours of Development

Process Steps by Project Phase

🔴 1) New Projects created in Ideation

🔴 2) Impacted Teams identified at Intake

🔴 3) Requirements identified in Discovery

🔴 🔵 4) Features defined in Feature Mapping

🔴 🔵 5) Feature Scope accepted in Feature Review

🔴 🔵 6) Incremental Feature delivery priority finalized at Production Increment Planning

🔴 🔵 7) Required Stories identified to meet Feature Acceptance Criteria at Story Mapping

🔴 🔵 🟢 8) Execution Team commits to top priority Stories based on capacity at Sprint Planning

🔴 🔵 🟢 9) Review of what has been completed, what is in progress, and any blocking issues

🔴 🔵 🟢 10) Execution Teams completes committed Stories during Development

🔴 🔵 11) Features are presented for approval at Demo

🔴 🔵 12) Features pass TQA / UAT / DIT testing during Test

🔴 🔵 13) Code Deployed to Production

🔴 14) Teams utilize Retros for continuous improvement

Glossary

- **Acceptance Criteria** - Specific conditions, with clearly defined pass / fail results, that must be met, before a Feature can be Accepted
- **Artifact -** Theme, Initiative, Feature, User Story, or Task
- **Assumption** (*a.k.a. Pre-Condition)* - Specific criteria that must be met before the end user is eligible to achieve the desired outcome identified in the Requirements or Acceptance Criteria
- **Backlog** - Stack-ranked, prioritized list of work items to be completed
  - **Portfolio Backlog** - Project / Initiative artifacts prioritized by the Business Sponsor or Business Partner
  - **Product Backlog** - Feature artifacts prioritized by the Product Owner
  - **Team Backlog** - User Story artifacts prioritized by the Scrum Master
  - **Iteration Backlog** - User Story artifacts for a specific team, for a specific Iteration, prioritized by the Scrum Master
- **Bug** - Error found in the Development environment
- **Business Case** - Issue or opportunity to be solved and expected outcome the solution will provide
- **Capacity** - Maximum amount of work a team is capable of delivering in a given time period
- **Customer Journey Mapping** - Exercise of visually documenting the process a customer goes through to achieve the desired outcome specified in the Requirements or Acceptance Criteria
- **Cycle Time** - Time between when work begins on an Artifact and when work is complete on an Artifact
- **Defect** - Error found in the Test environment
- **Definition of Done** - All Acceptance Criteria has been met, Testing is complete, code is deployed into Production, and all Artifacts are updated
- **Dependencies** - Completion of one Artifact is reliant upon the completion of another Artifact
- **Deployment** - Artifact has met the Definition of Done and is deployed into the Production environment

* Recommended if appropriate

Glossary

- **Dev-Ready** - Both the Business and IT agree upon the Value Statement, Scope, and Acceptance Criteria of an Artifact and there is sufficient detail for development to begin
- **Feature** - Independently deployable functionality that satisfies the Definition of Done for the Requirement
- **Impacted Teams** - Individual teams that will be required to complete work in order to meet the Definition of Done for the Artifact
- **Incident** - Error found in Production
- **Initiative** - Group of one or many Features that collectively deliver a specific outcome for the end user
- **Iteration** *(a.k.a. - Sprint)* - Pre-defined consistent time box, during which Impacted Teams develop and deliver User Stories, typically between 1 - 3 weeks
- **Lead Time** - Time between when an Artifact is created and when work is complete on an Artifact
- **Level of Effort** *(LoE)* - Estimated time (+/- 25%) required to deliver an Artifact
- **Milestone** - Used to track the completion of a set of functionality
- **Release** - Pre-defined consistent time box, during which Impacted Teams develop and deliver Features, typically between 1 month - 1 quarter
- **Requirement** - Desired outcome, output, or functionality requested by the Business on behalf of the End User
- **Return on Investment** (*RoI*) - Benefit *(Sales / Savings)* divided by the Expense *(Labor / Resources)* required to deliver an Artifact
- **Roadmap** - Strategic plan that identifies the key priorities major milestones required to deliver specified outcomes
  - **Portfolio Roadmap** - 6 - 18 month plan that identifies critical Business outcomes, typically at the Project or Initiative level
  - **Product Roadmap** - 1 - 6 month plan that identifies critical Product outcomes, typically at the Feature level
- **Roadmap** - Desired outcome or functionality that a specific Artifact will deliver

* Recommended if appropriate

Glossary

- **Spike** - A type of User Story used to track the work to research a solution to a Feature or Story when the Impacted Team does not know how to meet the Definition of Done
- **Story Point** - Abstract metric used to measure the complexity, risk, and time required to deliver a User Story. Is used as the baseline to determine a team's Capacity and Velocity. Modified Fibonacci Sequence is one of the more popular methods
- **Task** - Smallest unit of work, owned by an individual, and typically no smaller than 2 hours and no larger than 4 hours
- **Technical Debt** - Rework incurred for the future due to implementation of the easier, but not the best solution in the present
- **Test -** Ensure the execution of the Artifact satisfies the Acceptance Criteria and Steps to Test
  - **Automated** - Test cases are executed automatically with the assistance of a software tool
  - **Manual** - Tests cases executed manually by an individual
  - **Regression** - Ensure that deployment of new Feature does not negatively impact existing functionality
  - **Functional** - Ensure the deployment of new Feature satisfies the functional specifications of the Feature
- **User Story** - Unit of work for a specific Impacted Team that satisfies specific Acceptance Criteria within a Feature, completed within one Iteration

* Recommended if appropriate

Glossary

- **Value** - Objective measure used to prioritize an Artifact based on:
  - Business Rank
  - Return on Investment
  - Dependencies
  - Legal Requirements
  - Market Conditions
  - Other Factors
- **Value Statement** - Describes the intended outcome or value received by the end user as a result of a specific action
  - As a ______*(End User)*
  - I want to ______*(Action Performed)*
  - In order to ______*(Outcome / Value Received)*
- **Velocity** - Rolling average number of Artifacts delivered over a given number of Iterations *(User Stories)* or Releases *(Features)*

* Recommended if appropriate

Roles

- **Business Sponsor (BS) /** Requirements Producer
- **Product Manager (PdM) / Business Partner (BP) /** Requirements Refiner
- **Business Relationship Manager (BRM) /** Requirements Prioritizer
- **Program Manager (PgM) /** Feature Orchestrator
- **Product Owner (PO) /** Feature Refiner & Prioritizer
- **Quality Assurance (QA) /** Feature Tester
- **Scrum Master (SM) /** Story Orchestrator
- **Software Engineer (SE) /** Story Engineer
- **Embedded Tester (ET) /** Story Tester

Responsibility Matrix

ROLES R = Responsible P = Participates	Ideation			Refinement			Development						Test			Deploy	
	Define Initiative	Refine Initiative	Prioritize Initiative	Define Feature	Refine Feature	Prioritize Feature	Define User Story	Refine User Story	Prioritize User Story	Develop User Story	Test User Story	Accept User Story	Test Feature	Demo Feature	Accept Feature	Deploy	Retro
Requirements Producer	R	P*	P*	P*										P	P*		
Requirements Refiner	P*	R	P*	P	P*	P								P*	R		
Requirements Prioritizer	P	P	R	P*	P*	P*								P*			
Feature Orchestrator			P*	P*	P*	P*			P*					P*		P	
Feature Refiner / Prioritizer		P*	P*	R	R	R	P*	P	P*					R	P	P	
Feature Tester				P*	P	P	P*	P*	P*				R	P*		P*	
Story Orchestrator					P	P	R	R	R	P*	P*	R	P*			R	R
Story Engineer							P*	P	P	R	P	P*	P*			P*	P
Story Tester							P	P	P	P	R	P*	P*	P*			P

* Recommended if appropriate

Ceremonies

Participants / (*Owner)*

- ***Responsible for setting up the meeting, inviting the correct people, and ensuring the Entry Criteria is met***
- Involved Party

Entry Criteria:

- Specific conditions which must be satisfied before the Ceremony may begin

Key Activities:

- Specific activities that must be completed by participants

Exit Criteria / Output:

- Specific outcome or deliverable that is produced at the conclusion of the meeting

Metrics:

- Measurements to determine if the Ceremony was successful

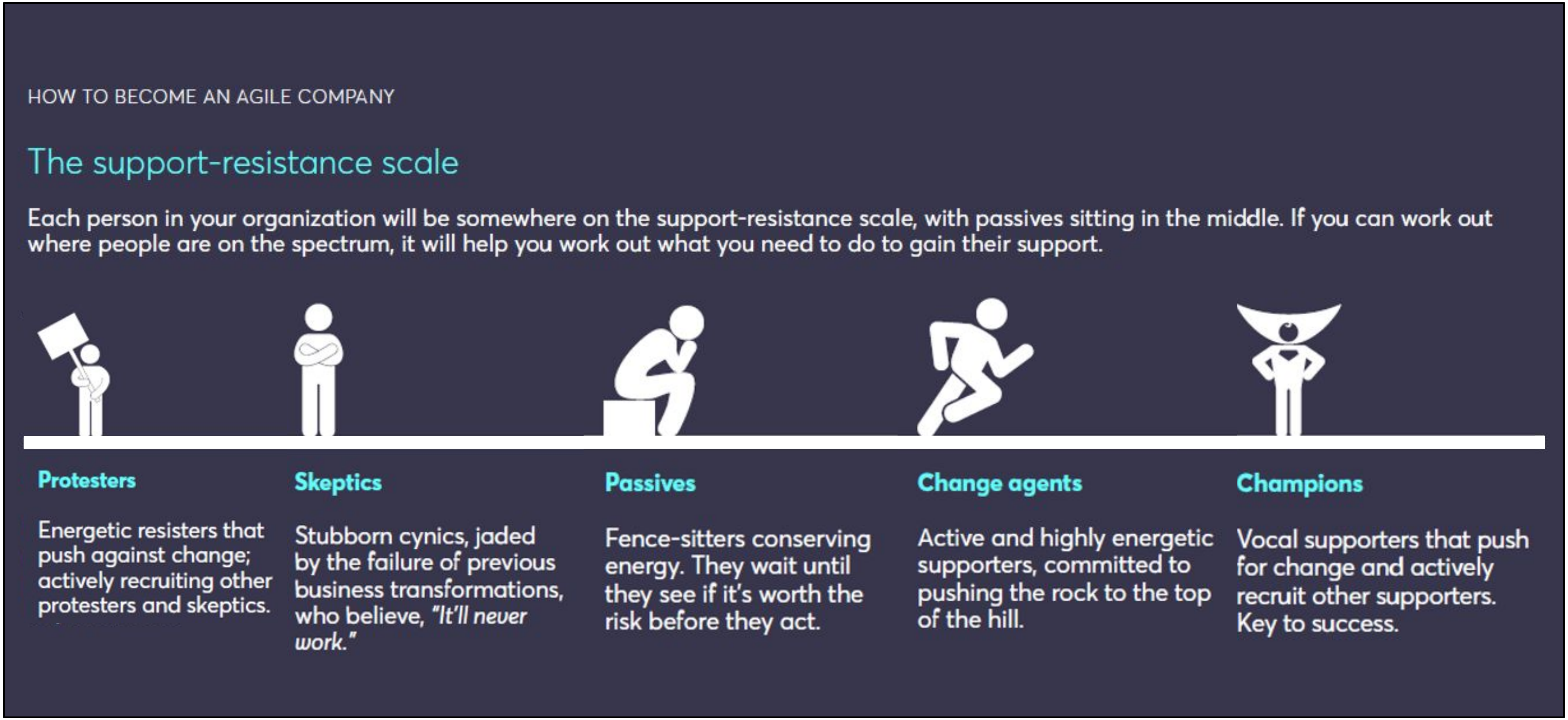
LET'S HAVE A LITTLE PREMEETING TO PREPARE FOR THE MEETING TOMORROW.
WHOA! DO YOU THINK IT'S SAFE TO JUMP RIGHT INTO THE PREMEETING WITHOUT PLANNING IT?
© 1998 United Feature Syndicate, Inc.(NYC)
OKAY, LET'S GET THIS PRELIMINARY PREMEETING MEETING GOING.
YOU THINK YOU'RE FUNNY, BUT YOU'RE NOT.

* Recommended if appropriate

What's Next...

Agile Training

- ACE 101 - Intro to Agile
- ACE 102 - Roles and Responsibilities
- ACE 103 - Ceremonies and Activities
- ACE 201 - Reporting Metrics
- ACE 202 - Rally Dashboards
- ACE 203 - Sprint Planning, Daily Stand-Up, Demos, Retros
- ACE 301 - Scrum
- ACE 302 - Feature Writing
- ACE 303 - Lean Development



HOW TO BECOME AN AGILE COMPANY

The support-resistance scale

Each person in your organization will be somewhere on the support-resistance scale, with passives sitting in the middle. If you can work out where people are on the spectrum, it will help you work out what you need to do to gain their support.

Protesters

Energetic resisters that push against change; actively recruiting other protesters and skeptics.

Skeptics

Stubborn cynics, jaded by the failure of previous business transformations, who believe, *"It'll never work."*

Passives

Fence-sitters conserving energy. They wait until they see if it's worth the risk before they act.

Change agents

Active and highly energetic supporters, committed to pushing the rock to the top of the hill.

Champions

Vocal supporters that push for change and actively recruit other supporters. Key to success.