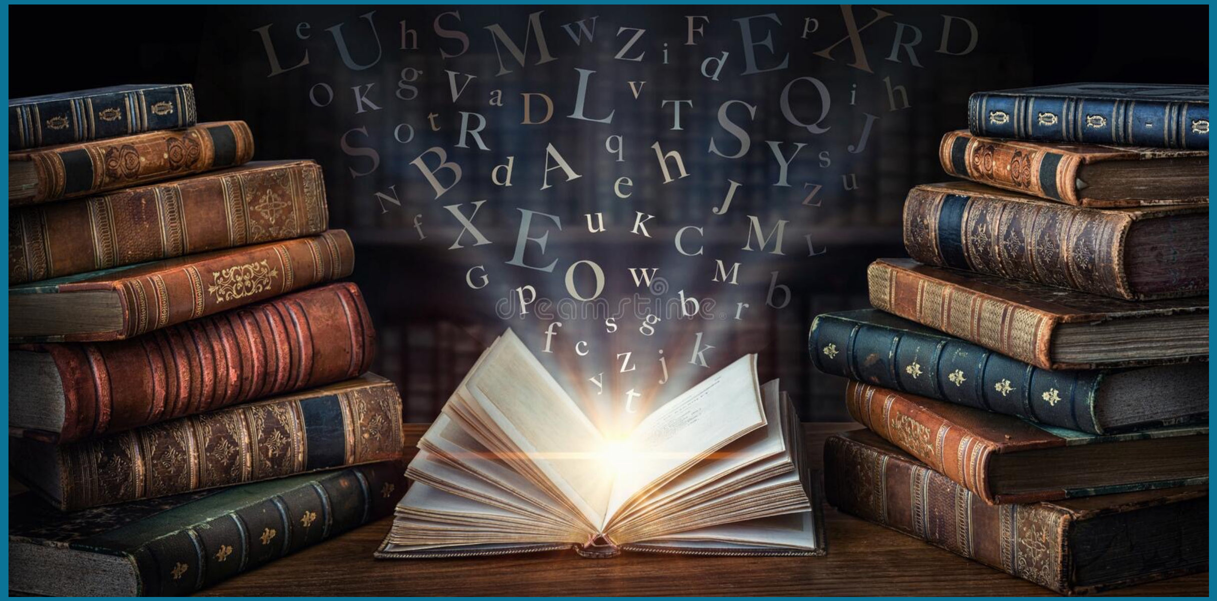
About the author

Expertise spanning over twenty-five years in software engineering with focus on Digital Engineering, DevSecOps, and Agile building large complex solutions across multiple domains from submarines to satellites. She advocates for continuous learning with multiple certifications including SAFe Fellow, SPCT, CEC, PMP, PMI-ACP, and CSEP. She is a Systems Engineering PhD candidate at Colorado State researching best practices to deliver complex safety critical solutions using Agile and DevSecOps.

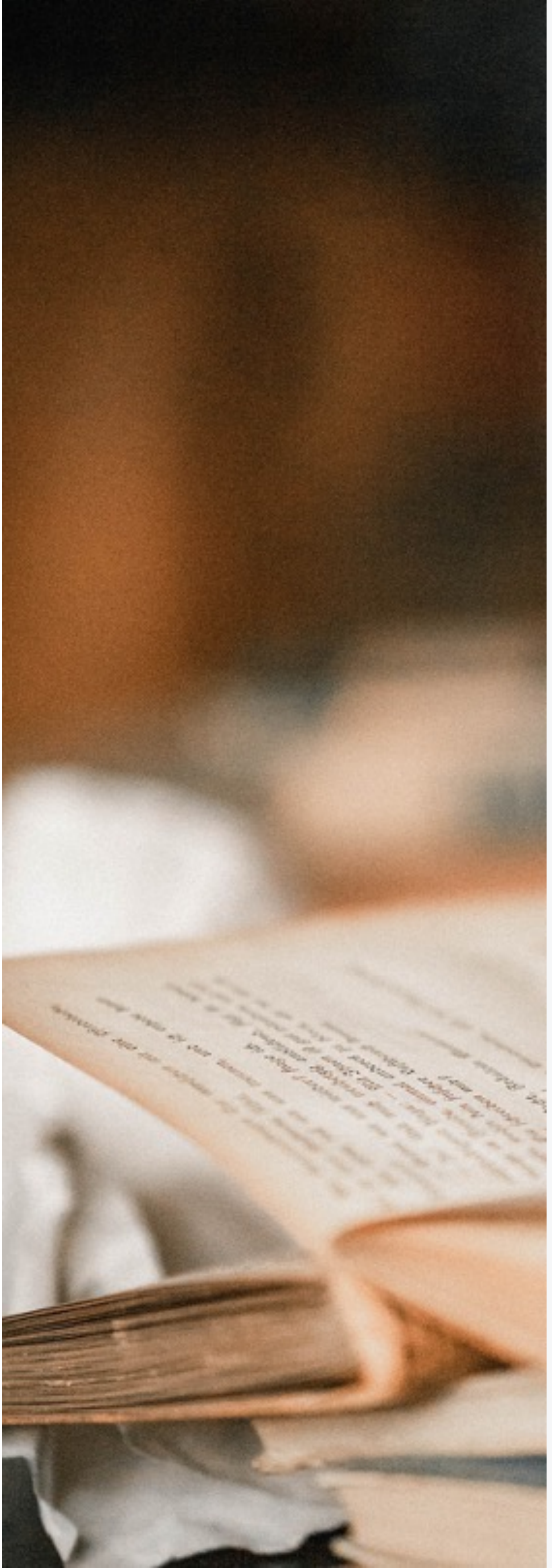


Books are how we install new software on our wetware!!

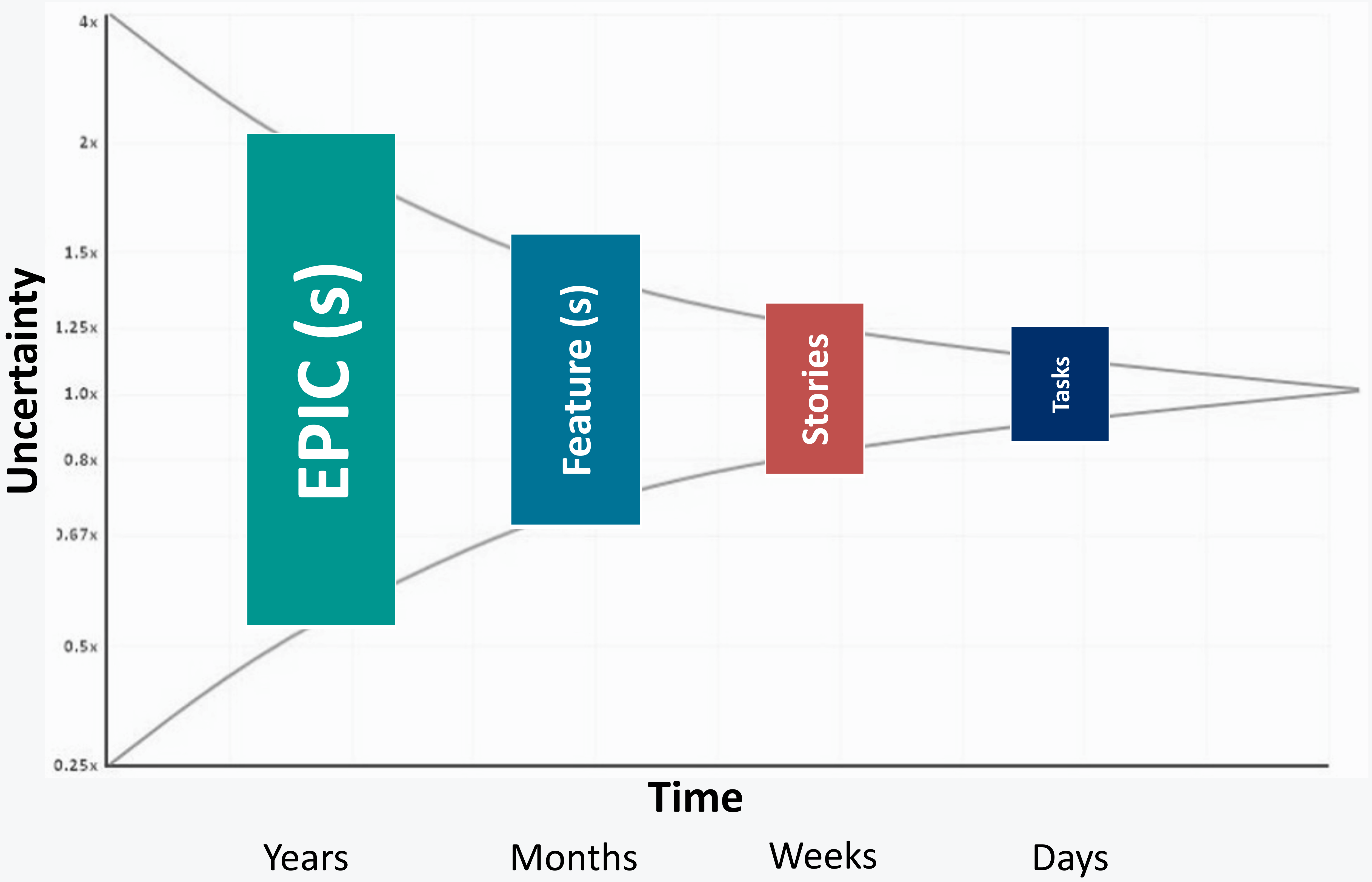
https://www.linkedin.com/in/robinyeman/

Patterns for Splitting Epics and Features

How to decompose your work effectively



Cone of Uncertainty



https://www.linkedin.com/in/robinyeman/

Work Decomposition

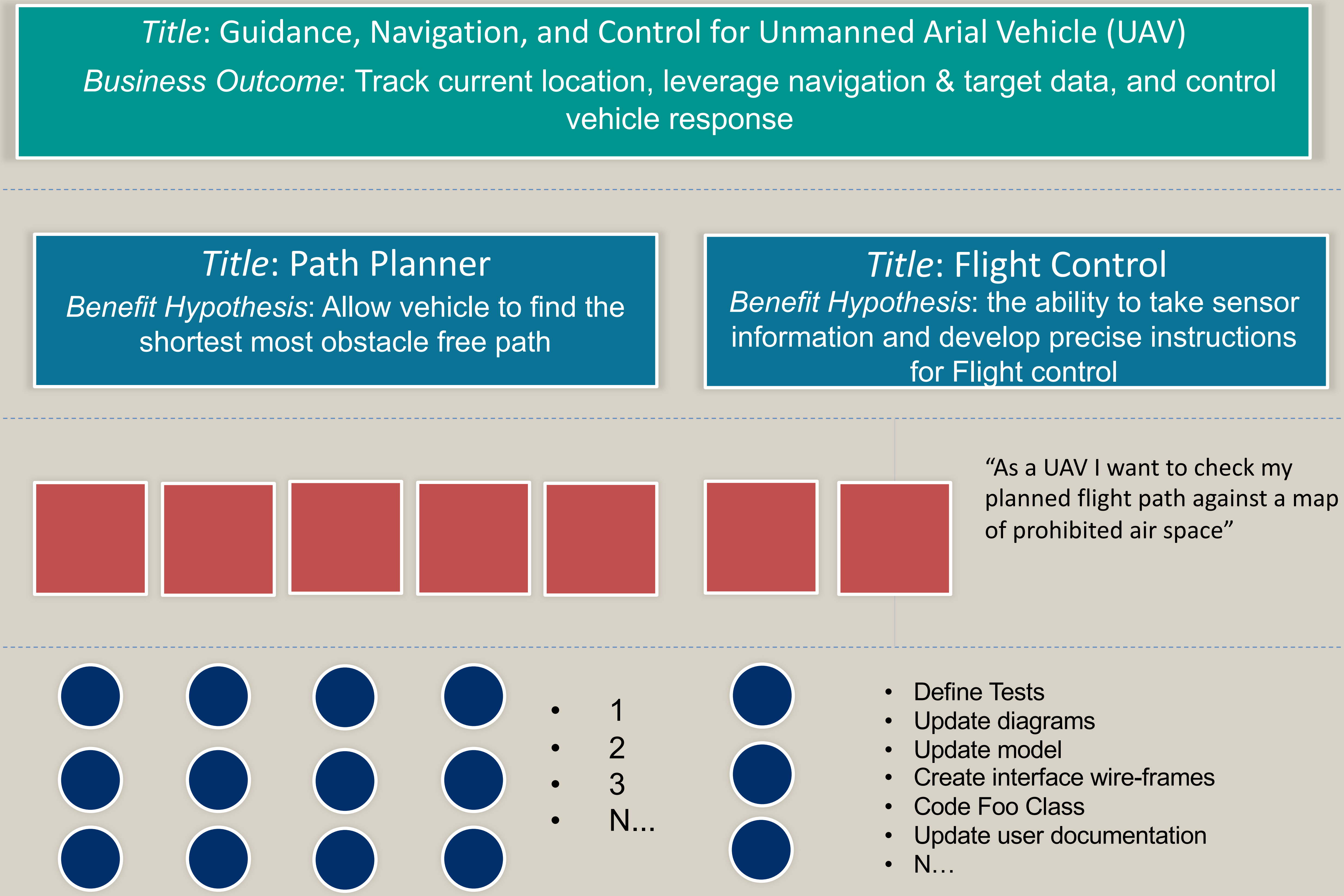
<table>
<tr><td>Years / Team of Teams(s)</td><td colspan="8">EPIC</td><td colspan="8">EPIC</td></tr>
<tr><td>Months / Team(s)</td><td colspan="4">Feature</td><td colspan="4">Feature</td><td colspan="4">Feature</td><td colspan="4">Feature</td></tr>
<tr><td>Weeks / Team</td><td colspan="2">Story</td><td colspan="2">Story</td><td colspan="2">Story</td><td colspan="2">Story</td><td colspan="2">Story</td><td colspan="2">Story</td><td colspan="2">Story</td><td colspan="2">Story</td></tr>
<tr><td>Days / Individuals</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td><td>Task</td></tr>
</table>

Decompose work and then place into time boxes creating modular, validated and verified items of value .

https://www.linkedin.com/in/robinyeman/

Product Backlog Contains

- Product Backlog
- Sprint Backlog
- Individual ”to do list”



Epic

(Story Points, T-shirt Size)

Title: Guidance, Navigation, and Control for Unmanned Arial Vehicle (UAV)

Business Outcome: Track current location, leverage navigation & target data, and control vehicle response

Months - Years

(Business Outcome)

Feature

(Story Points)

Title: Path Planner

Benefit Hypothesis: Allow vehicle to find the shortest most obstacle free path

Title: Flight Control

Benefit Hypothesis: the ability to take sensor information and develop precise instructions for Flight control

Weeks - Months

(Benefit Hypothesis)

Stories

(Story Points)

“As a UAV I want to check my planned flight path against a map of prohibited air space”

Days - Week

(Needed Functionality)

Tasks

(Hours)

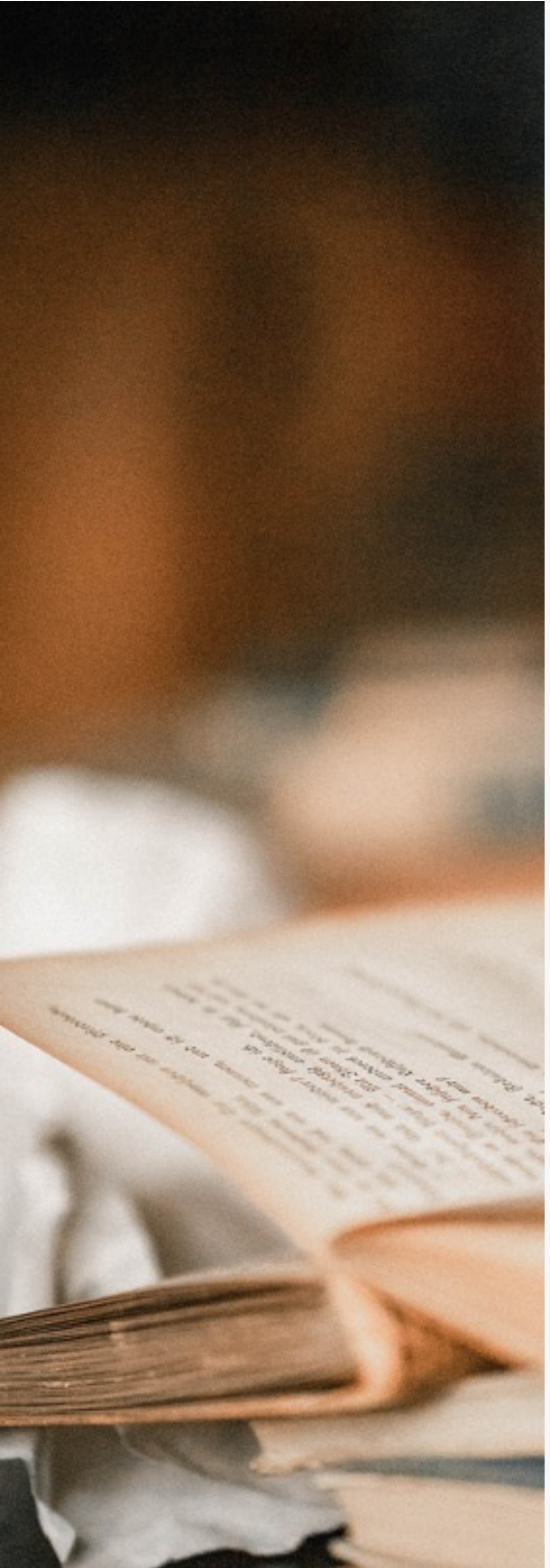
- 1
- 2
- 3
- N…

- Define Tests
- Update diagrams
- Update model
- Create interface wire-frames
- Code Foo Class
- Update user documentation
- N…

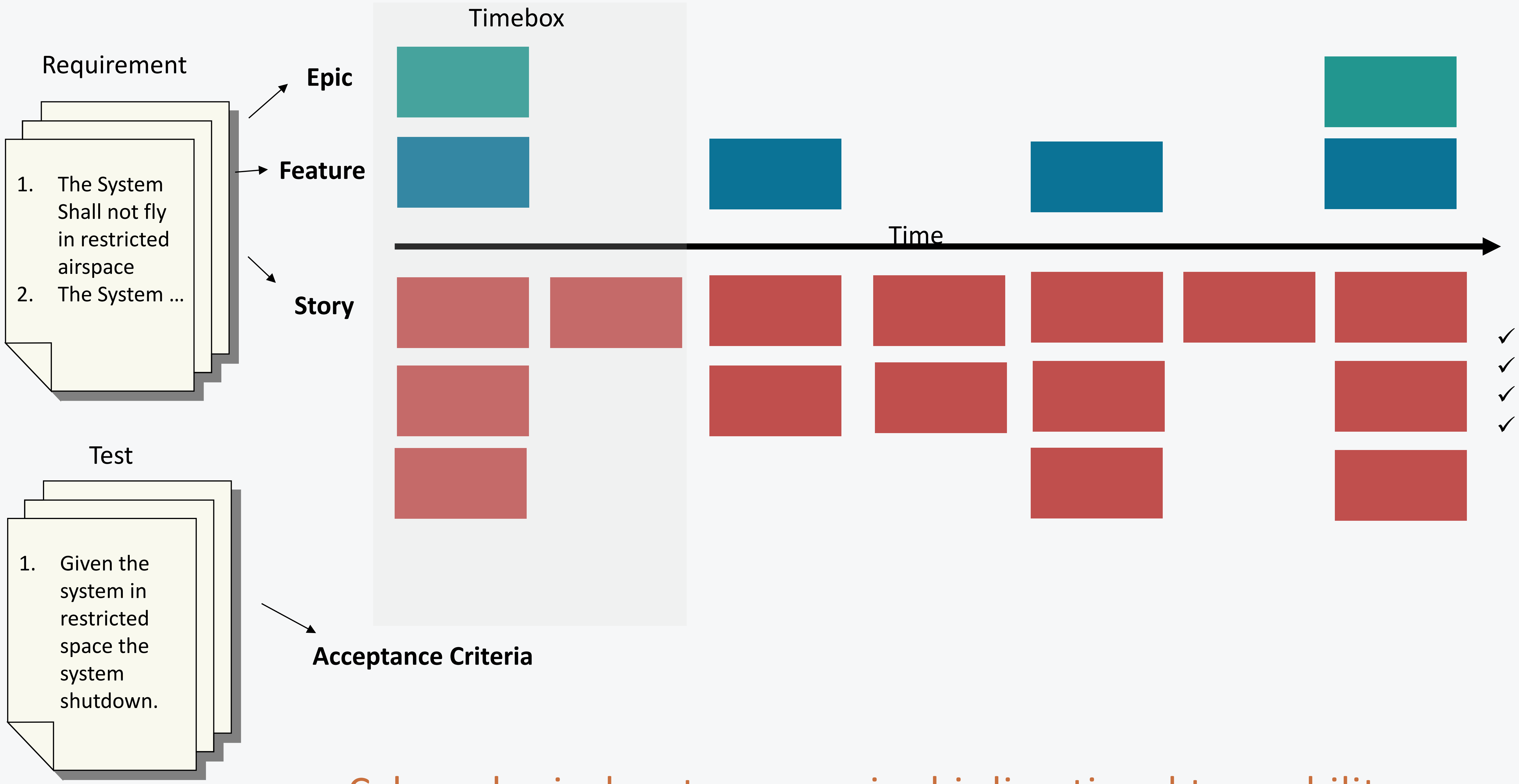
Up to ~8 hours

(daily punch-list)

https://www.linkedin.com/in/robinyeman/



Requirements Traceability



Requirement

1. The System Shall not fly in restricted airspace
2. The System …

Epic

Feature

Story

Timebox

Time

Test

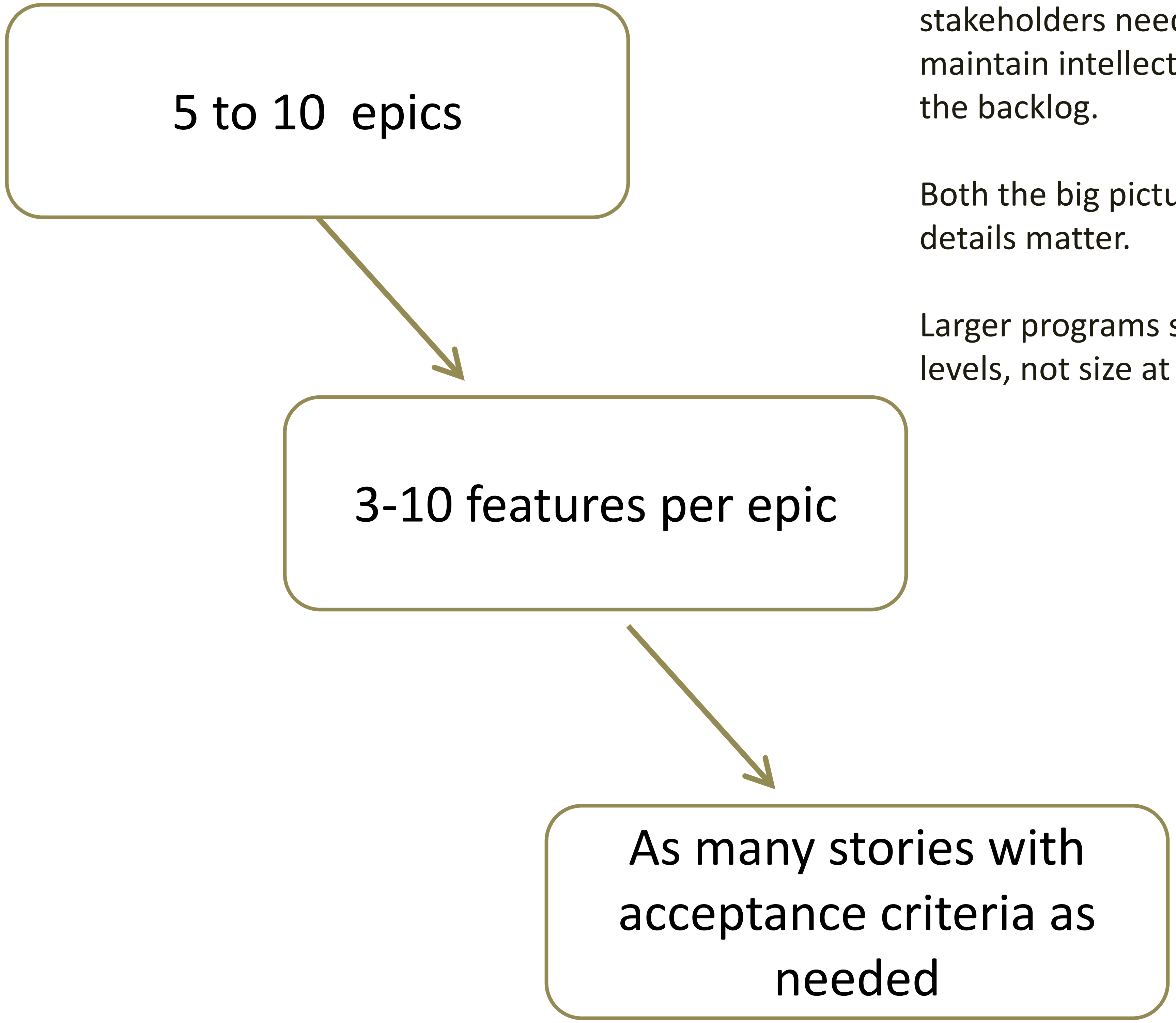
1. Given the system in restricted space the system shutdown.

Acceptance Criteria

- ✓ Description
- ✓ Acceptance Criteria
- ✓ Deliverables
- ✓ Constraints

Cyber-physical systems require bi-directional traceability

Rule of Thumb for midsize



5 to 10 epics

3-10 features per epic

As many stories with acceptance criteria as needed

Engineers, managers and stakeholders need to be able to maintain intellectual control of the backlog.

Both the big picture and the details matter.

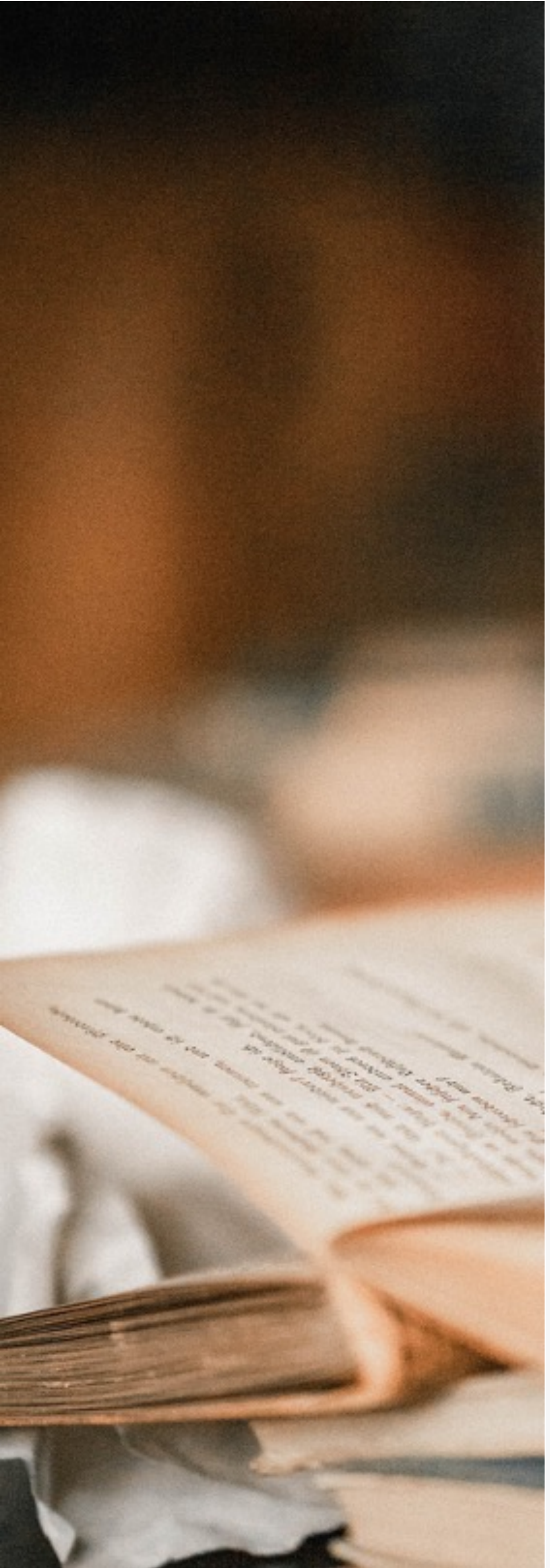
Larger programs should add levels, not size at a given level.

https://www.linkedin.com/in/robinyeman/

Patterns for splitting features

	Pattern	Description
1.	Work flow Steps	Break out all of the steps of the work flow required to deliver value
2.	Business rule variations	Accomplishment of different Business rules
3.	Major Effort	Large effort items can often be split, where the first one is the instantiation of capability and the remaining continue to improve.
4.	Simple / Complex	Capture simplest version of feature and complete remaining to add complexity
5.	Variations in data	Data Variations, such as data sources, complexity, language variants
6.	Data Methods	Split by the user interface itself
7.	Deferring System Qualities	Begin with a simple capability and add the system qualities incrementally
8.	Operations	Order of operations such as CRUD
9.	Use Case Scenarios	Split by goals or scenarios
10.	Breaking out Spikes	Investigate to get information first

https://www.linkedin.com/in/robinyeman/



Workflow Steps

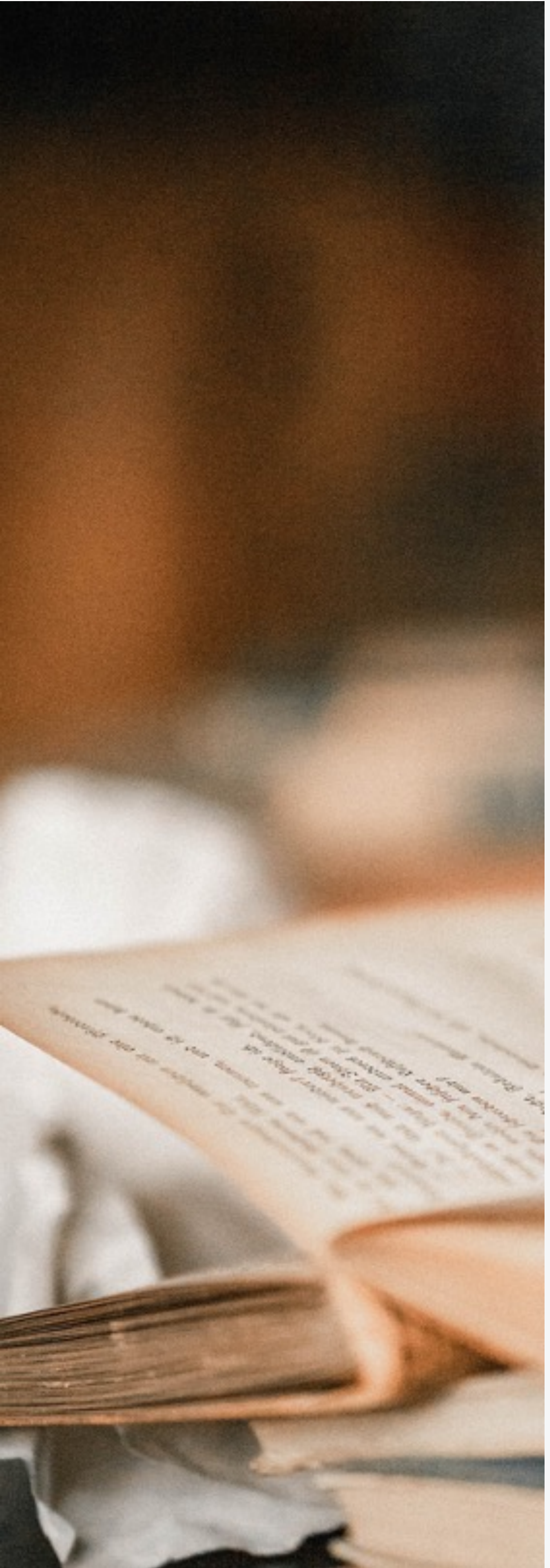
Break out the steps of the workflow required to deliver value

Feature: **Flight Path Planner**

Workflow Steps:

- **StartUp**
- **FlightPlanObtainGPSLoc**
- **FlightPlanSetWayPoint**
- **FlightScheduler**
- **ShutDown**

User Story: **As a UAV Pilot I want to run startup sequence for drone so that I can interact with UAV to plan flight.**



Business Rules

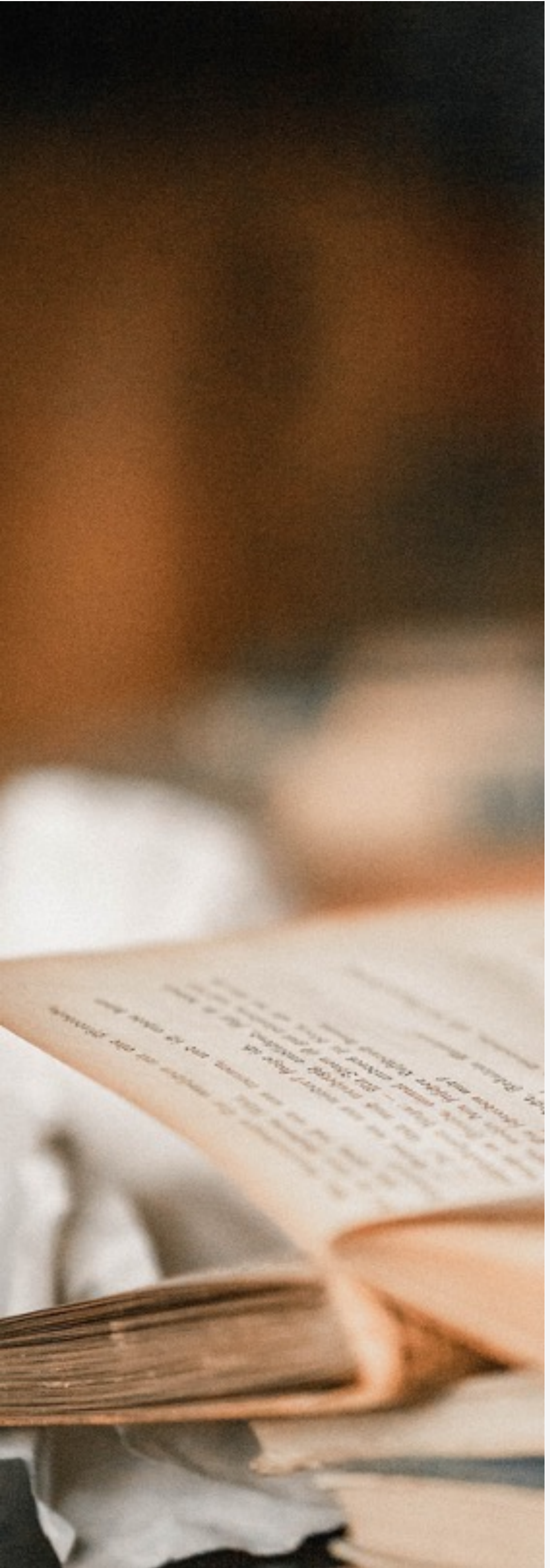
Accomplishment of different business rules

Feature: Flight Path Planner

Business Rules:

- FlyZone
- SafetyThresholds
- Permit
- LocationCharacteristics
- Timing
- SetWayPoint
- ParameterAsignment

User Story: As a UAV Pilot I want to view non restricted Fly Zones so that I can plan a safe flight for my UAV.



Major Effort

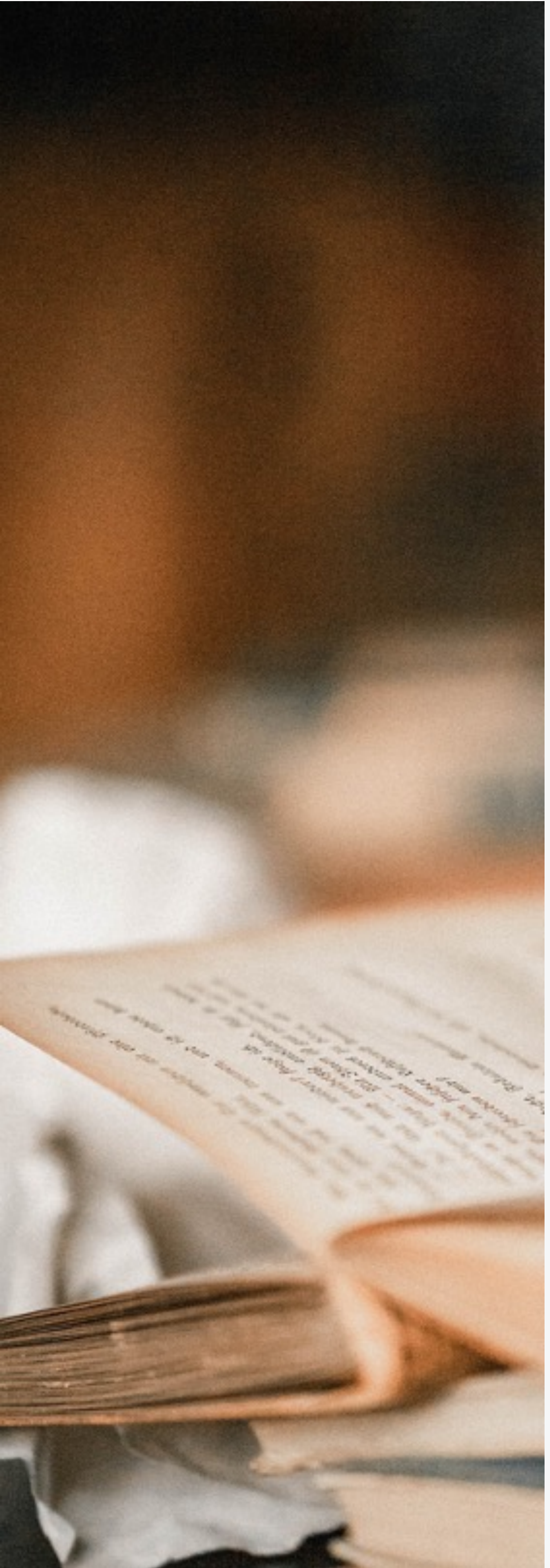
Large effort items can often be split, where the first one is the instantiation of capability and the remaining continue to improve.

Feature: Flight Path Planner

Major Effort:

- Initialization
- ManualFlightPlanMaP
- AutoFlightPlanMap
- AutoFlightPlanMapless
- AutoFlightPlanMultiSensor
- ShutDown

User Story: As a UAV Pilot I want the ability to manually input a flight plan so that I can fly my UAV



Simple / Complex

Capture simplest version of Feature and complete remaining to add complexity

→

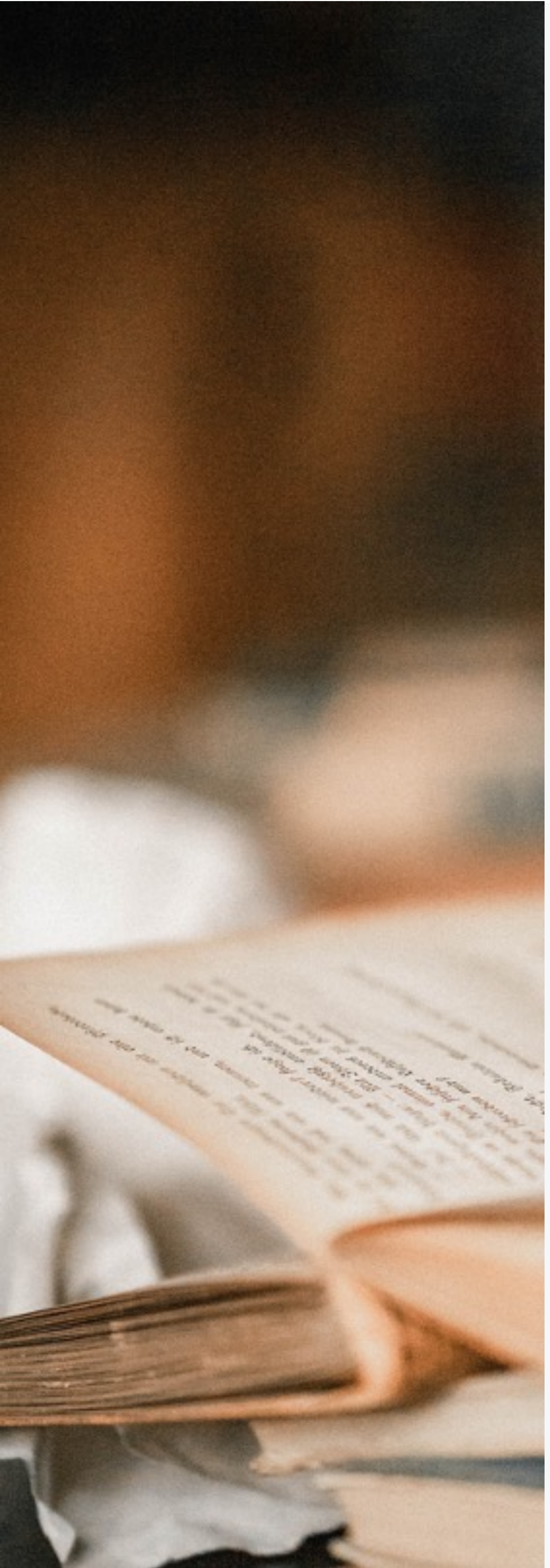
Feature: Flight Path Planner

Simple Effort:

- StartUp
- InputPath
- StorePath
- PlanningTechnique
- AutoFlightPlanMultiSensor
- ShutDown

User Story: As a UAV Pilot I want the ability to input a path into the system so that I can begin planning my route.

https://www.linkedin.com/in/robinyeman/



Variations in data

Data Variations, such as data sources, complexity, language variants

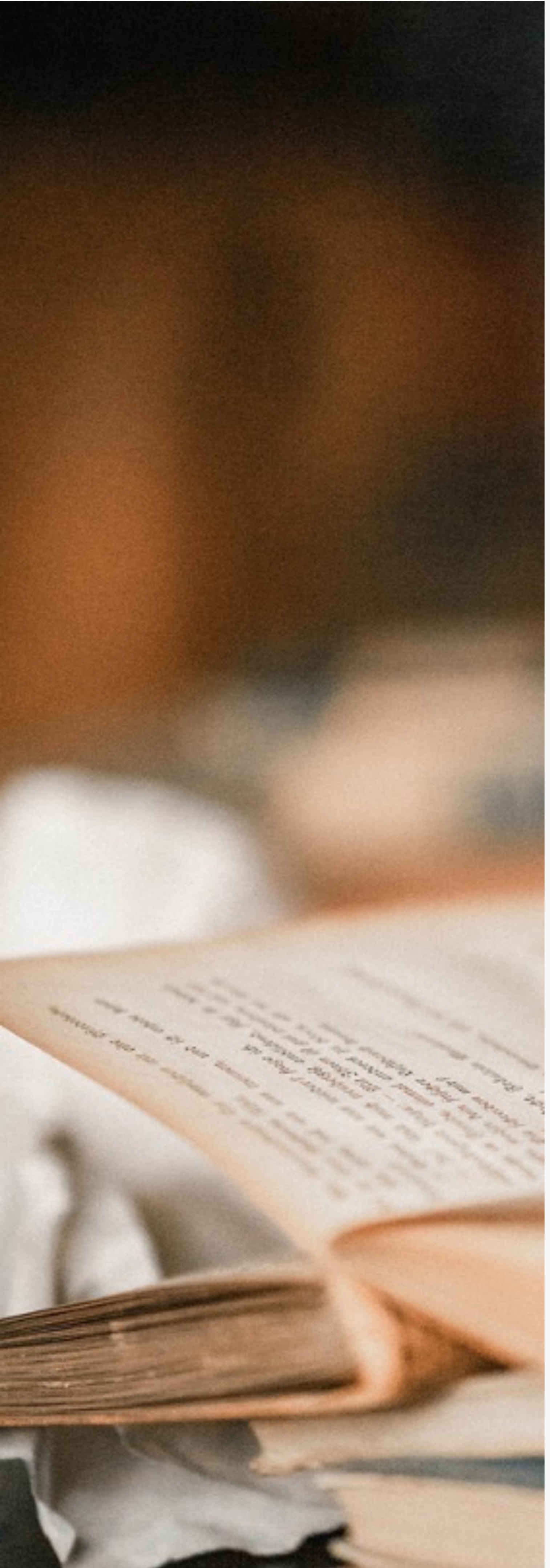
Feature: Flight Path Planner

Variations in Data:

- VehicleData
- MissionData
- EnvironmentData
- FlightCoordinateData
- RegulatoryData

User Story: As a UAV Pilot I want input coordinate data so that I schedule my flight path.

https://www.linkedin.com/in/robinyeman/



Data Methods

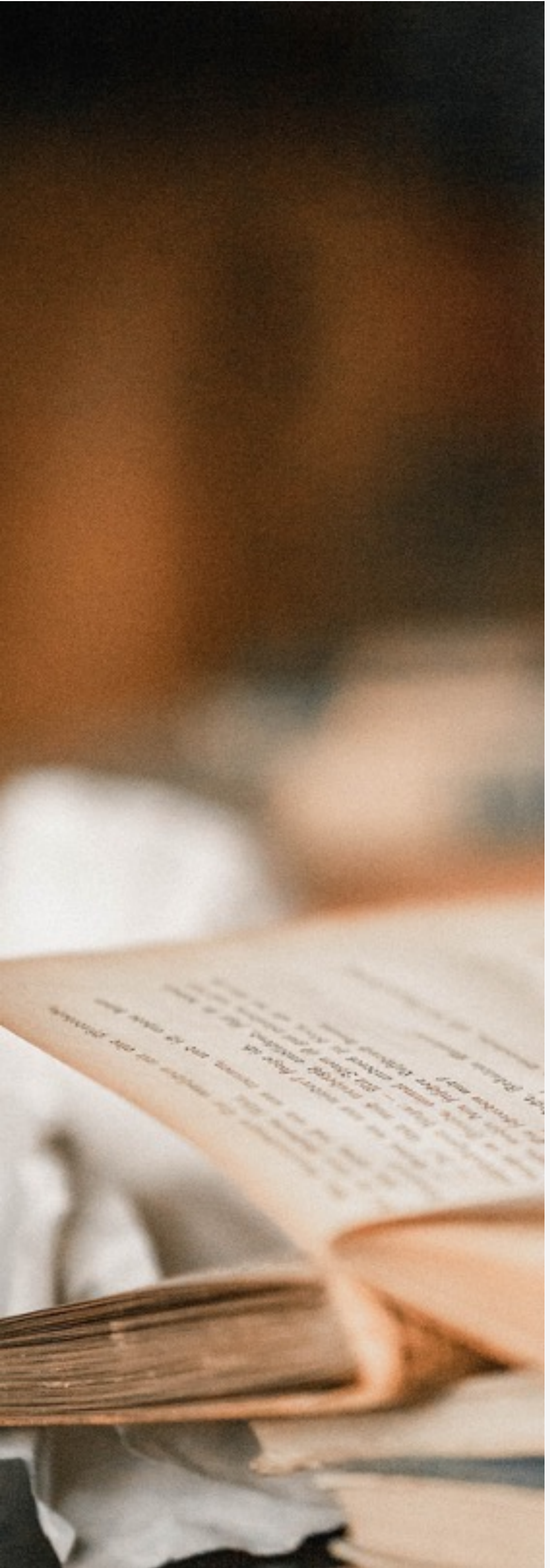
Split by the user interface itself.

Feature: Flight Path Planner

Data Methods:

- KeyboardCommandPlan
- TouchScreenCommandPlan
- AutonomousRulesCommand
- AutonomousAICommand

User Story: As a UAV Pilot I want be able to input flight path through keyboard so that I can generate flight plan.



Deferring System Qualities

Begin with a simple capability and add the system qualities incrementally.

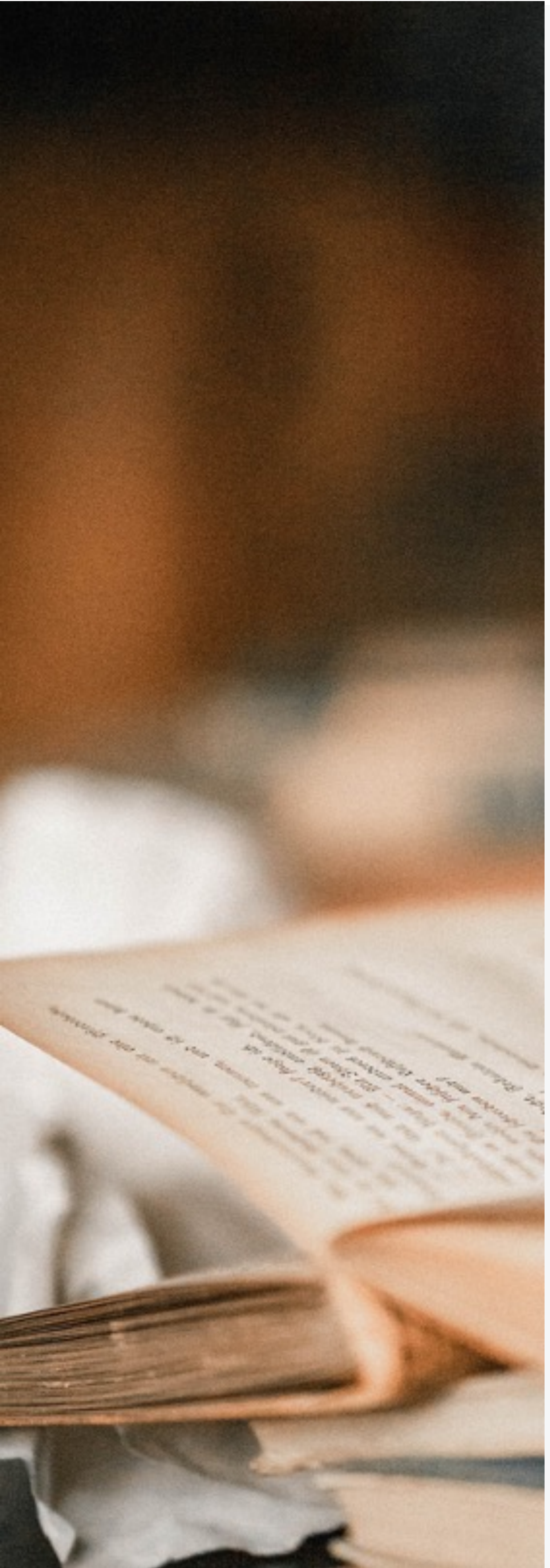
Feature: Flight Path Planner

System Qualities:

- LocalPathDeterminGen
- LocalPathNonDeterminGem
- GlobalPathDeterminGen
- GlobalPathNonDeterminGen

User Story: As a UAV Pilot I want to be able top input a local pre-determined path so that I can fly my UAV locally.

https://www.linkedin.com/in/robinyeman/



Operations

Order of operations such as CRUD.

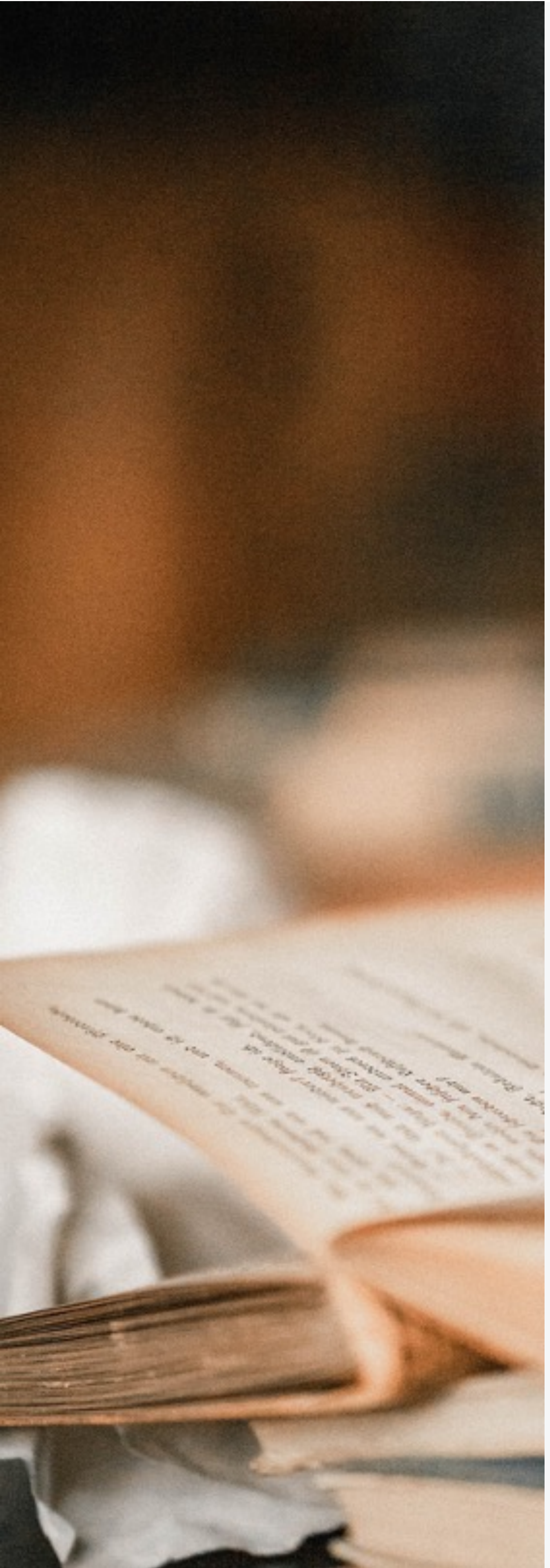
Feature: Flight Path Planner

Order of operations:

- CreateFlightPath
- StoreFlightPath
- ReadFlightPath
- UpdateFlightPath
- DeleteFlightPath

User Story: As a UAV Pilot I want create a flight plan so that I can determine where to fly my UAV.

https://www.linkedin.com/in/robinyeman/



Use Case Scenario's

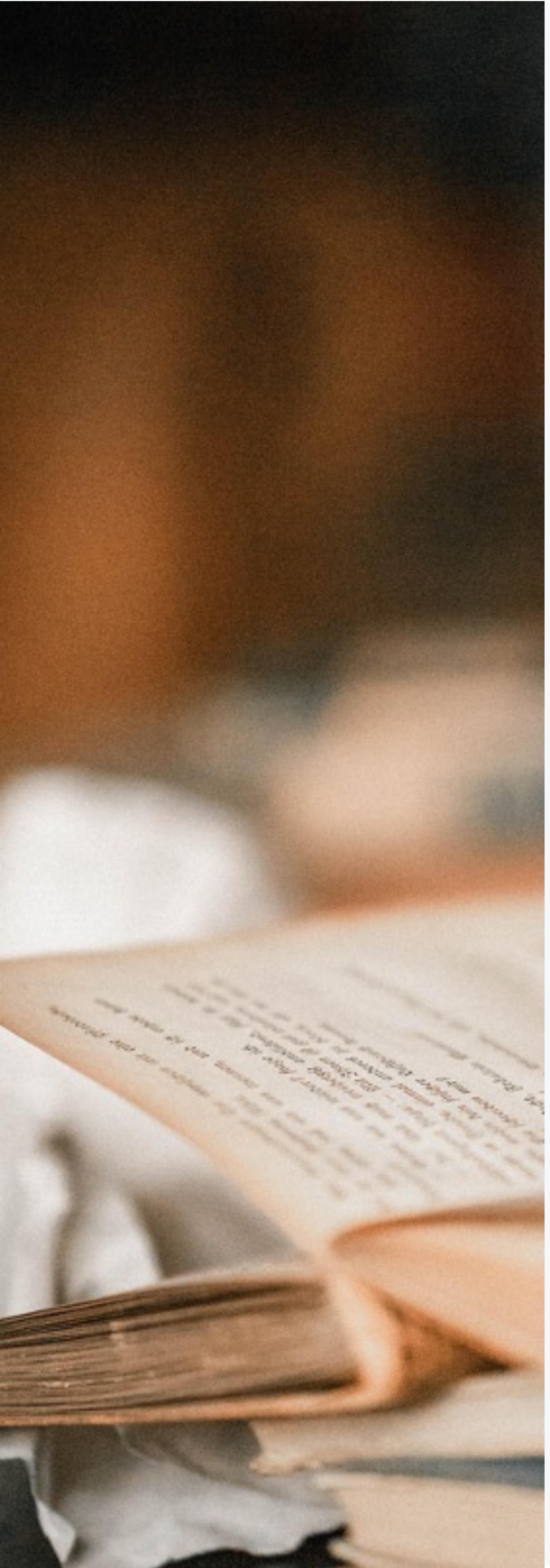
Split by goals or scenarios

Feature: Flight Path Planner

Use Case

- MissionPlanning
- WayPointGeneration
- TrajectoryFollow
- Sensing
- DecisionMaking

User Story: As a UAV Pilot I want generate multiple way points so my UAV can fly autonomously



Break out Spike

Investigate to get information first.

Feature: Flight Path Planner

Spikes:

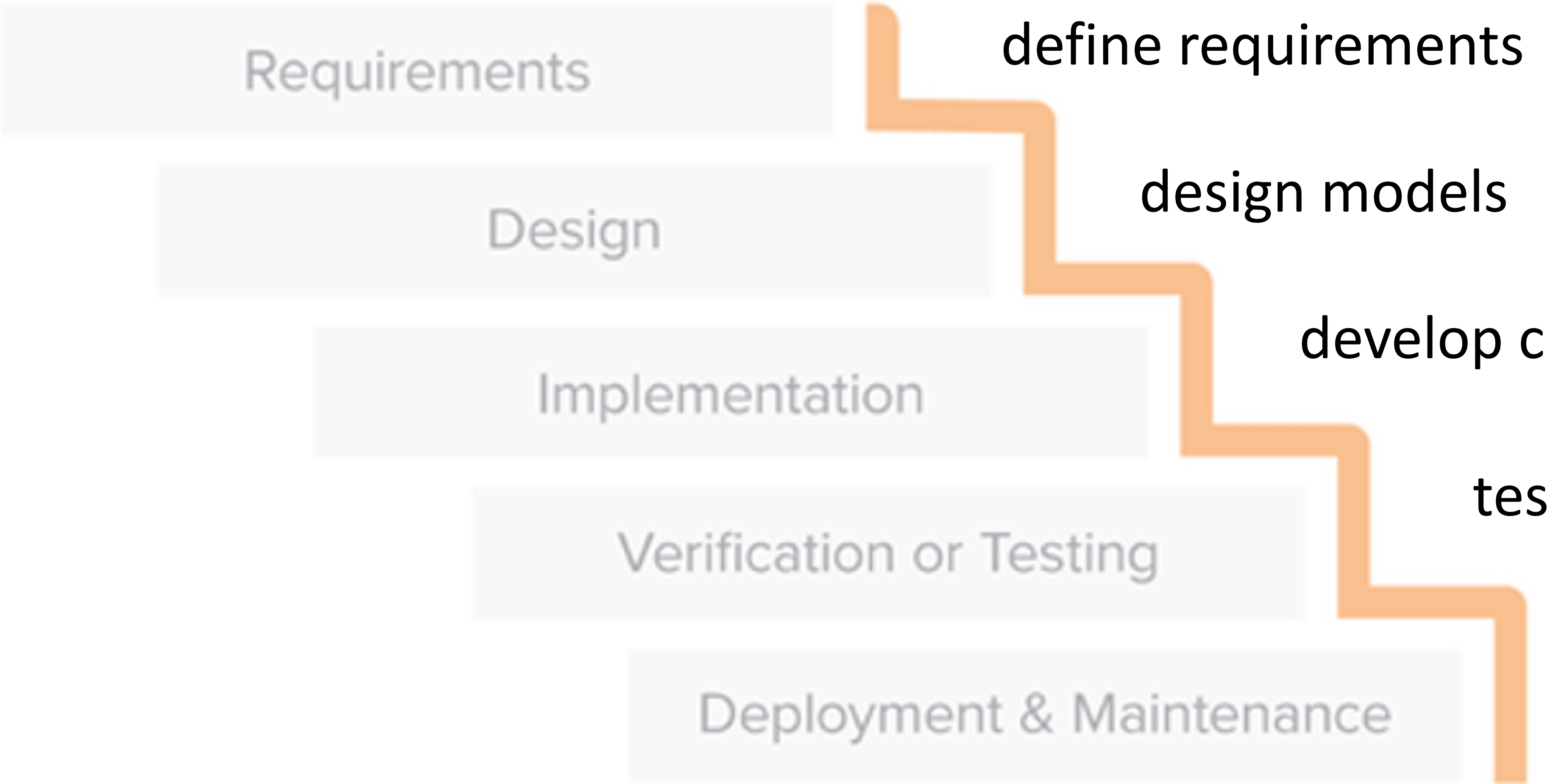
- ResearchOptimalPath
- ResearchTimeVariable
- ResearchSensorOptions
- ResearchMulitObjectivePath
- ResearchSwarming

User Story: As a UAV Pilot I want select the optimal path so that I can minimize risk to my UAV.

https://www.linkedin.com/in/robinyeman/

Ways features are Not split

As a user I want to.....



Requirements

Design

Implementation

Verification or Testing

Deployment & Maintenance

define requirements

design models

develop capability

test capability

deploy capability

So that I can deliver a big bang solution

https://www.linkedin.com/in/robinyeman/

The goal is to learn faster

Through validated delivery of value!!!

Delivering in the digital Age is different than the industrial age.

https://www.linkedin.com/in/robinyeman/

Questions