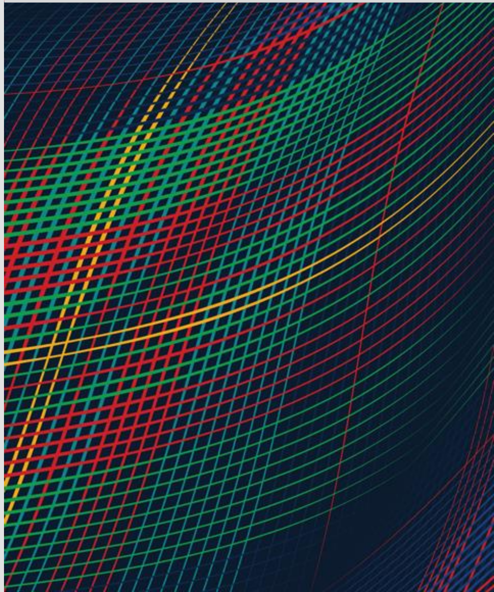
Leadership in Orchestrating Digital Engineering: Integrating MBSE, Agile, DevSecOps, AI, and Simulation for Satellite Mission Success

Tutorial

02/24/2025

Hasan Yasar

Carnegie Mellon University
Software Engineering Institute



Carnegie Mellon University Software Engineering Institute

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM24-0119

Introduction

Hasan Yasar

Robin Yeman

Carnegie Mellon University
Software Engineering Institute

leidos

Objective

This workshop is designed for leaders, program managers, and decision-makers responsible for overseeing satellite missions but who may not be experts in digital engineering technologies. Leaders will learn how to effectively orchestrate key technologies: Model-Based Systems Engineering (MBSE), Agile, DevSecOps, AI, and Simulation into a cohesive strategy to ensure satellite mission success. The workshop includes strategic insights, hands-on leadership exercises, and practical tools for integration.

Agenda

Welcome and Introduction

9:00– 9:30

Strategic role of leadership in satellite Mission

Executive Overview of key technologies

9:30 – 10:30

MBSE-Agile-DevSecOps

Network Break

Overcoming Silos, Leadership Strategies

10:45 – 12:00

Overcoming Silos, Leadership Strategies – Agile Industrial DevOps

Lunch

Exercise

1:00 – 3:30

Bringing it all Together

Network Break

Examples of Technology Integration

3:45– 4:45

AI, DevSecOps, Agile and MBSE Integration Success.

Closing Remarks

4:45– 5:00

Leadership role in System Optimization and wrap up

Logistics

Three Questions

Experience with MBSE, Agile / DevOps, AI?
What organization?
What would you like to learn?

- Introductions
- Three Questions
- Working Agreements

Leadership in Orchestrating Digital Engineering

Set the Stage

Introduction Key Terms

Agile

Project management and product development approach that emphasizes flexibility, customer satisfaction, continuous improvement, and high adaptability to change.

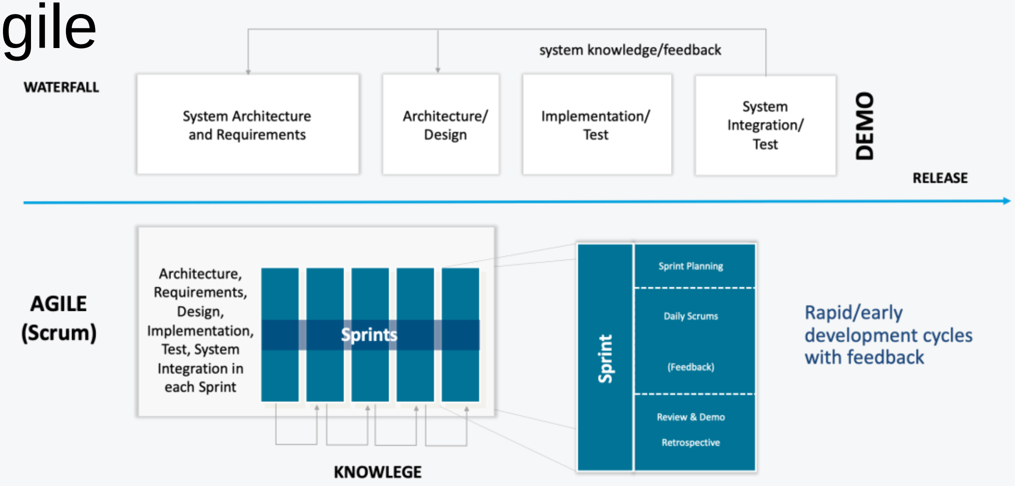
DevSecOps

DevOps approach, integrating security practices within the DevOps process. DevSecOps involves creating a 'Security as Code' culture with ongoing.

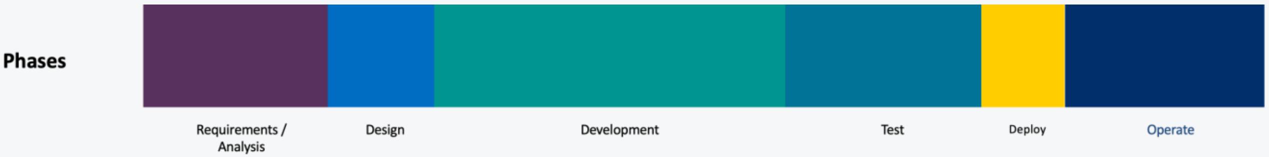
Generative AI

Artificial Intelligence that can generate content from text to images and more

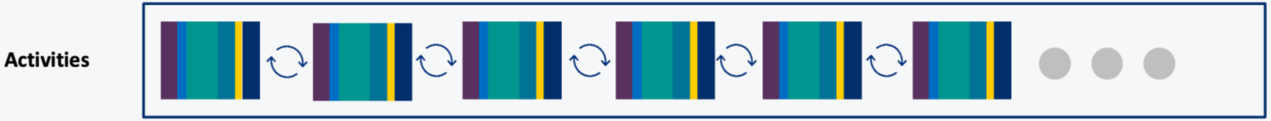
Agile



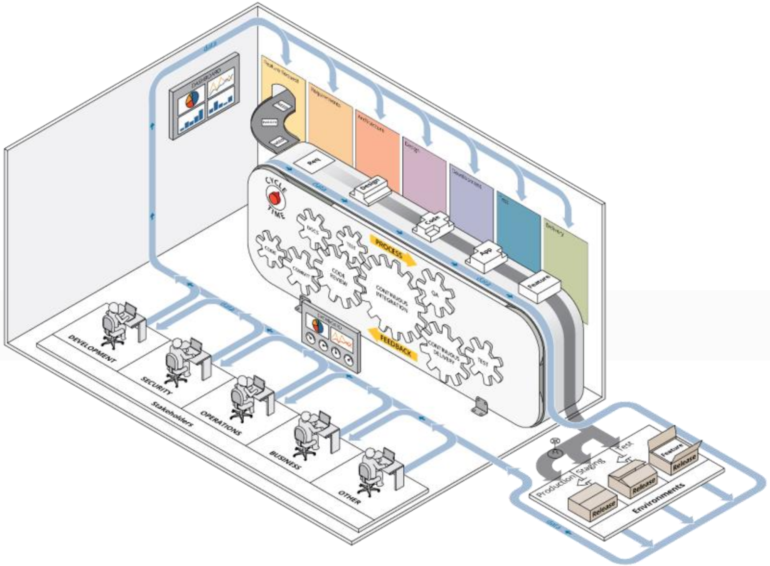
Rapid/early development cycles with feedback



SDLC Phases in smaller batch



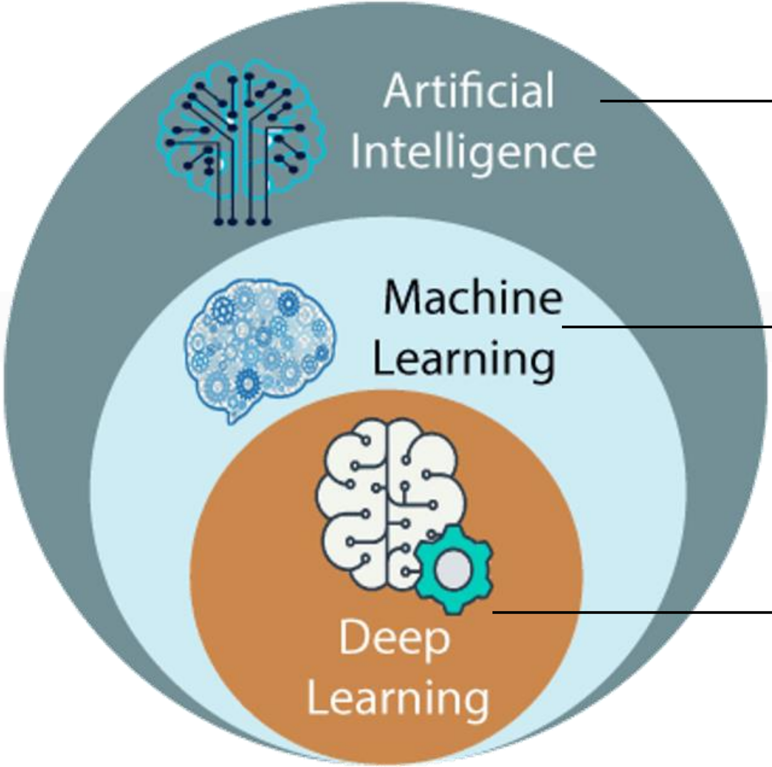
Dev{Sec}Ops for AI



- “**DevOps** is a set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing, continuously improving, and operating software and systems products and services.” [1]
- “**DevSecOps** is a cultural and engineering practice that breaks down barriers and opens collaboration between development, security, and operations organizations using automation to focus on rapid, frequent delivery of secure infrastructure and software to production. It encompasses intake to release of software and manages those flows predictably, transparently, and with minimal human intervention/effort.” [2]
- **DevSecOps for AI** implements the steps for AI model development and deployment with additional security requirements that focus on “trusting the model.”

[1] IEEE 2675 *DevOps Standard for Building Reliable and Secure Systems Including Application Build, Package and Deployment*
[2] *DevSecOps Guide: Standard DevSecOps Platform Framework*. U.S. General Services Administration. https://tech.gsa.gov/guides/dev_sec_ops_guide.

Artificial Intelligence



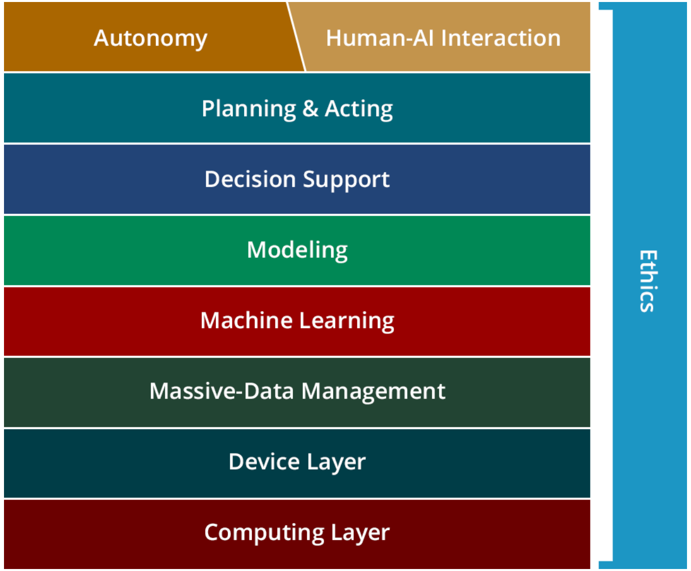
Machines can mimic human behavior including logic

A subset of AI that allows machines to improve at tasks with experience

A subset of machine learning
That enable software to train itself to perform tasks

What is AI systems?

AI systems demonstrate the capacity to achieve forms of tertiary consciousness such as cognitive feedback, self-recognition, and components of self-concept. *


Autonomy

Human-AI Interaction

Planning & Acting

Decision Support

Modeling

Machine Learning

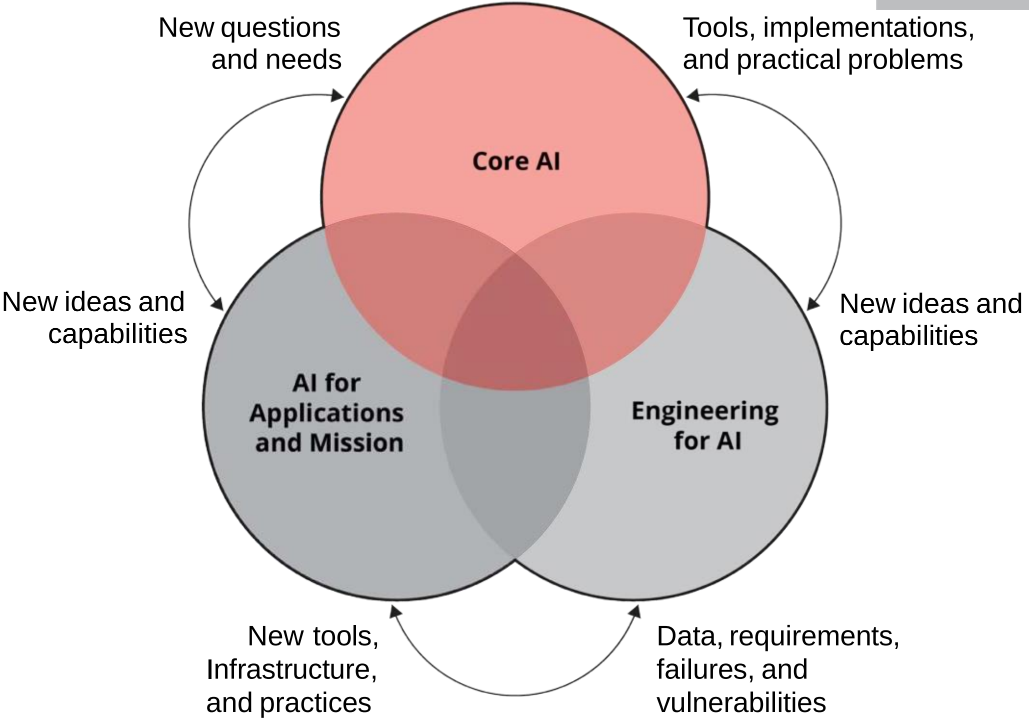
Massive-Data Management

Device Layer

Computing Layer

Ethics

CMU AI Stack


New questions and needs

Tools, implementations, and practical problems

Core AI

New ideas and capabilities

New ideas and capabilities

AI for Applications and Mission

Engineering for AI

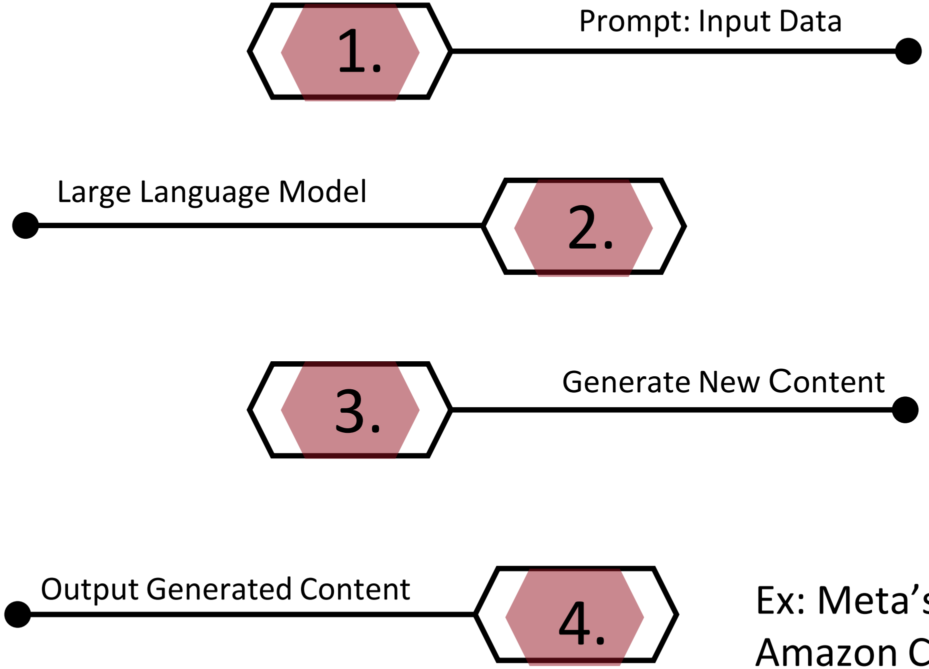
New tools, Infrastructure, and practices

Data, requirements, failures, and vulnerabilities

* https://www.sciencedirect.com/topics/veterinary-science-and-veterinary-medicine/artificial-intelligence-system

Generative AI

Output of new content (text, images, video) in four easy steps

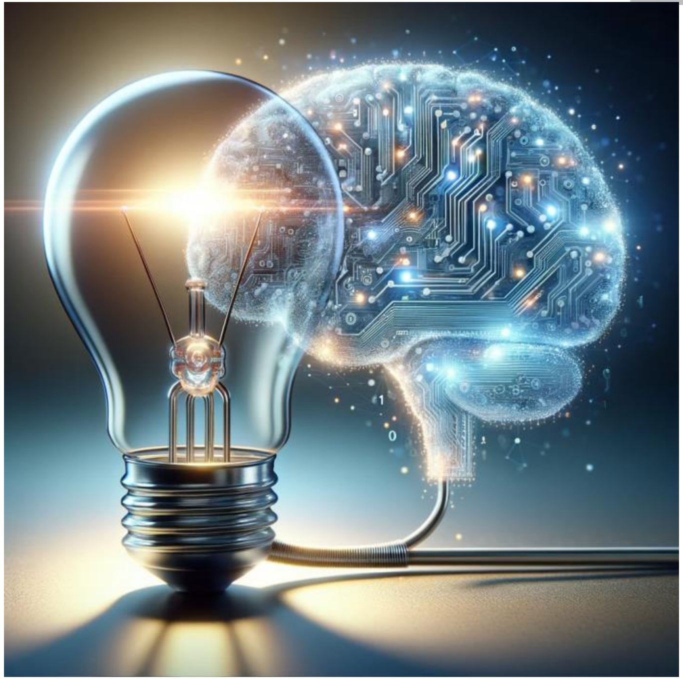


1. Prompt: Input Data

2. Large Language Model

3. Generate New Content

4. Output Generated Content



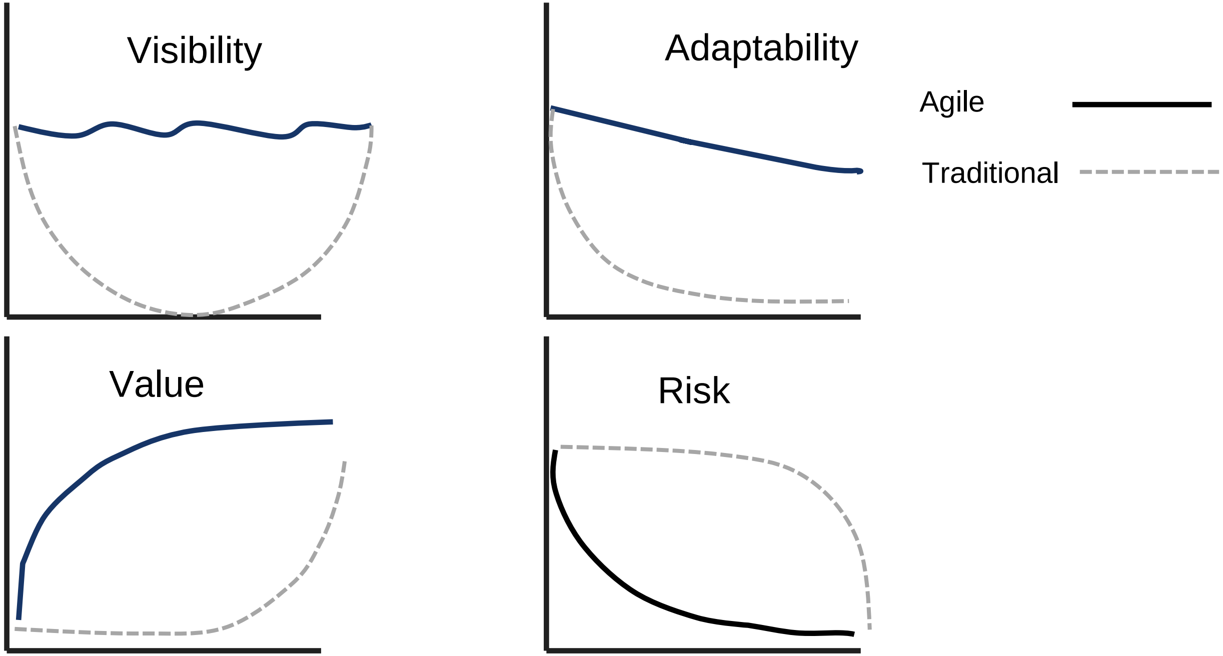
Ex: Meta’s LLaMA, OpenAI’s ChatGPT, GitHub Copilot, and Amazon CodeWhisperer

Leadership in Orchestrating Digital Engineering

Agile 101 foundations

Why Agile

The difference between Agile and traditional



The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiations

Responding to change over following a plan

That is, while there is value in the items on the right, **we value the items on the left more**.

12 Principles (1 of 2)

Carnegie Mellon University Software Engineering Institute

1 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2 Business people and developers must work together daily throughout the project.

3 Welcome changing requirements even late in development. Agile processes harness change for competitive advantage.

4 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

5 Deliver working software frequently, every couple of weeks to every couple of months, with a preference for the shorter timescale.

6 The most efficient and effective method of conveying information to and within a development team is through face-to-face conversation.

12 Principles (2 of 2)

7 Working software is the primary measure of progress.

8 Simplicity—the art of maximizing the amount of work not done—is essential.

9 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

10 The best architectures, requirements, and designs emerge from self-organizing teams.

11 Continuous attention to technical excellence and good design enhances agility.

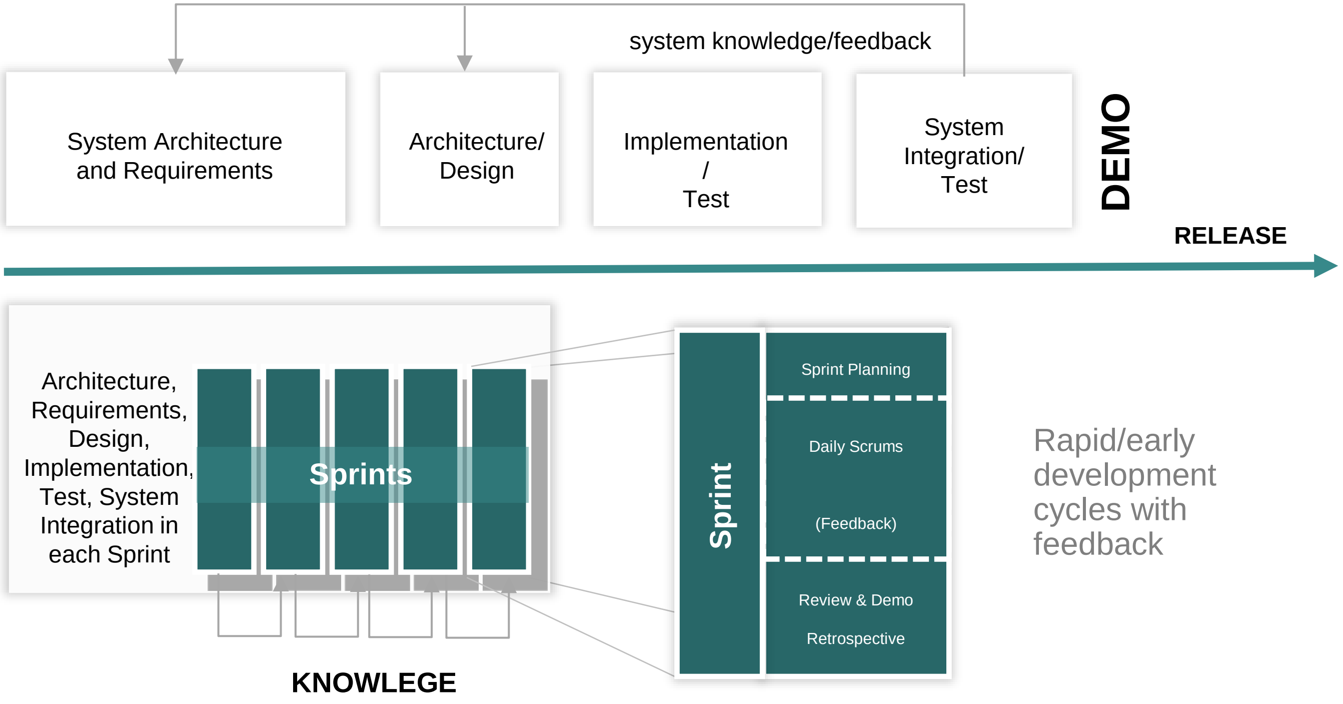
12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile vs. Waterfall

Agile is empirical versus predictive process, how we complete work is only differentiator.

WATERFALL

AGILE



system knowledge/feedback

System Architecture and Requirements

Architecture/ Design

Implementation / Test

System Integration/ Test

DEMO

RELEASE

Architecture, Requirements, Design, Implementation, Test, System Integration in each Sprint

Sprints

Sprint

Sprint Planning

Daily Scrums

(Feedback)

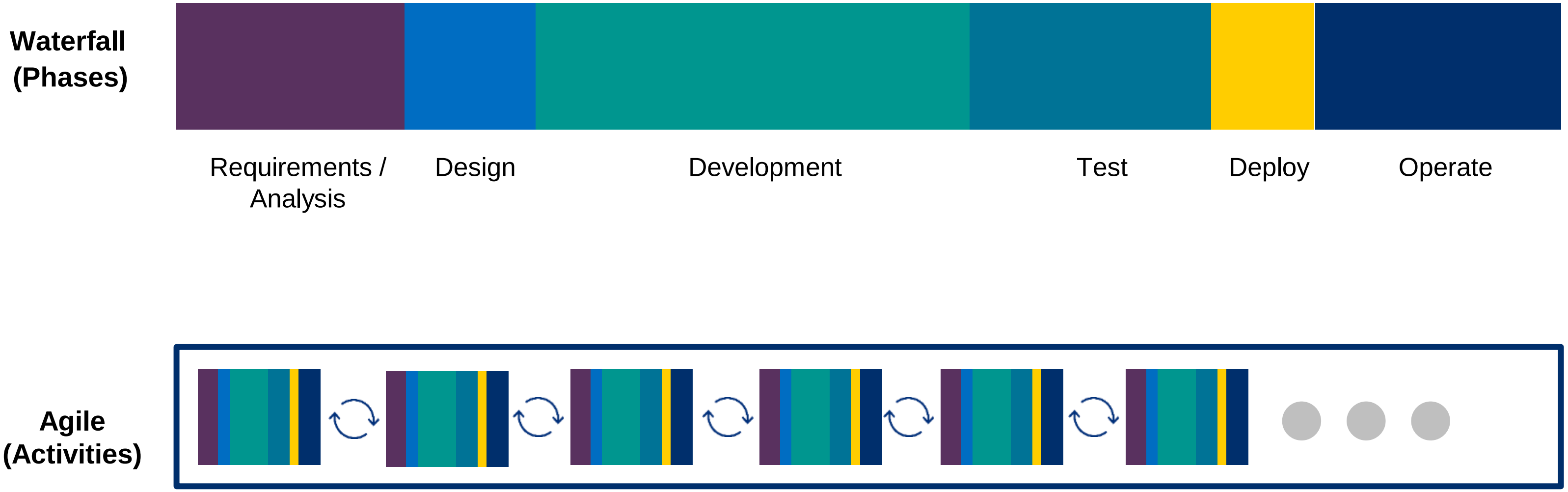
Review & Demo

Retrospective

KNOWLEGE

Rapid/early development cycles with feedback

Waterfall vs Agile Lifecycle

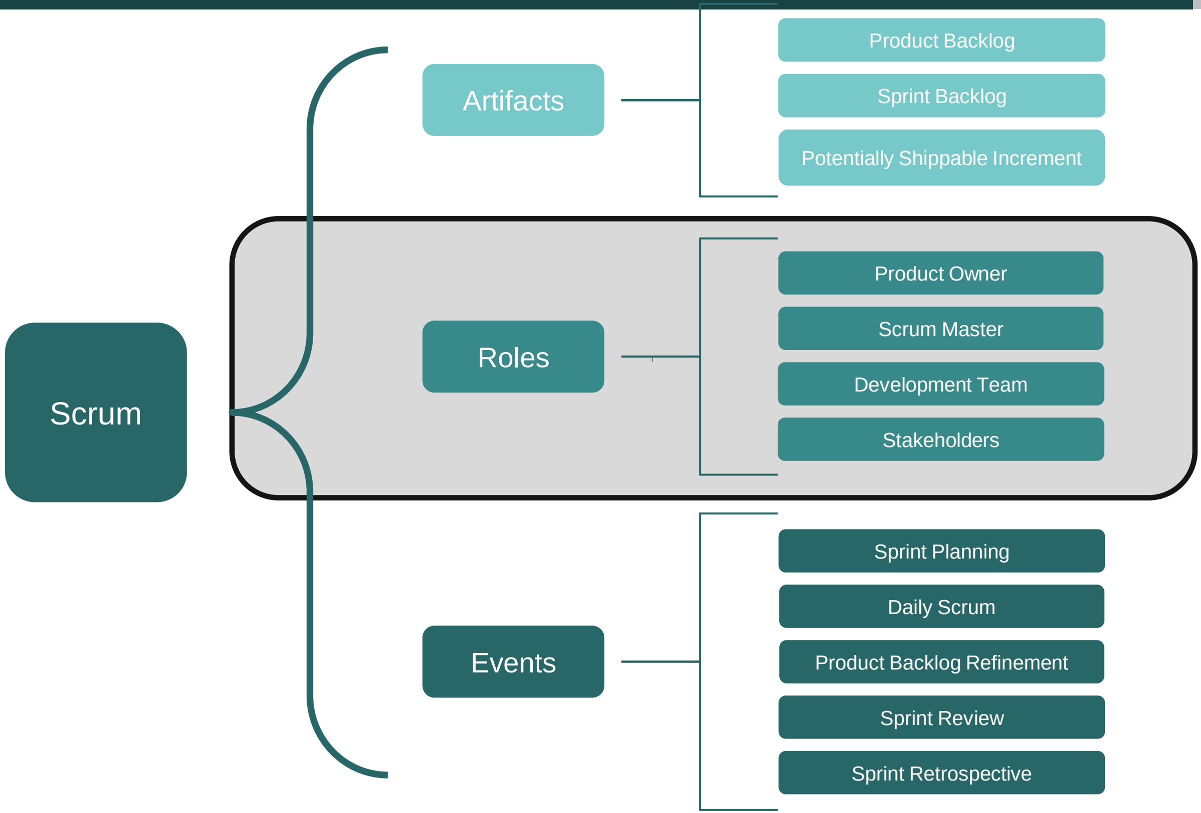


Activities as opposed to phases that have smaller batch sizes and are repeated.

Roles

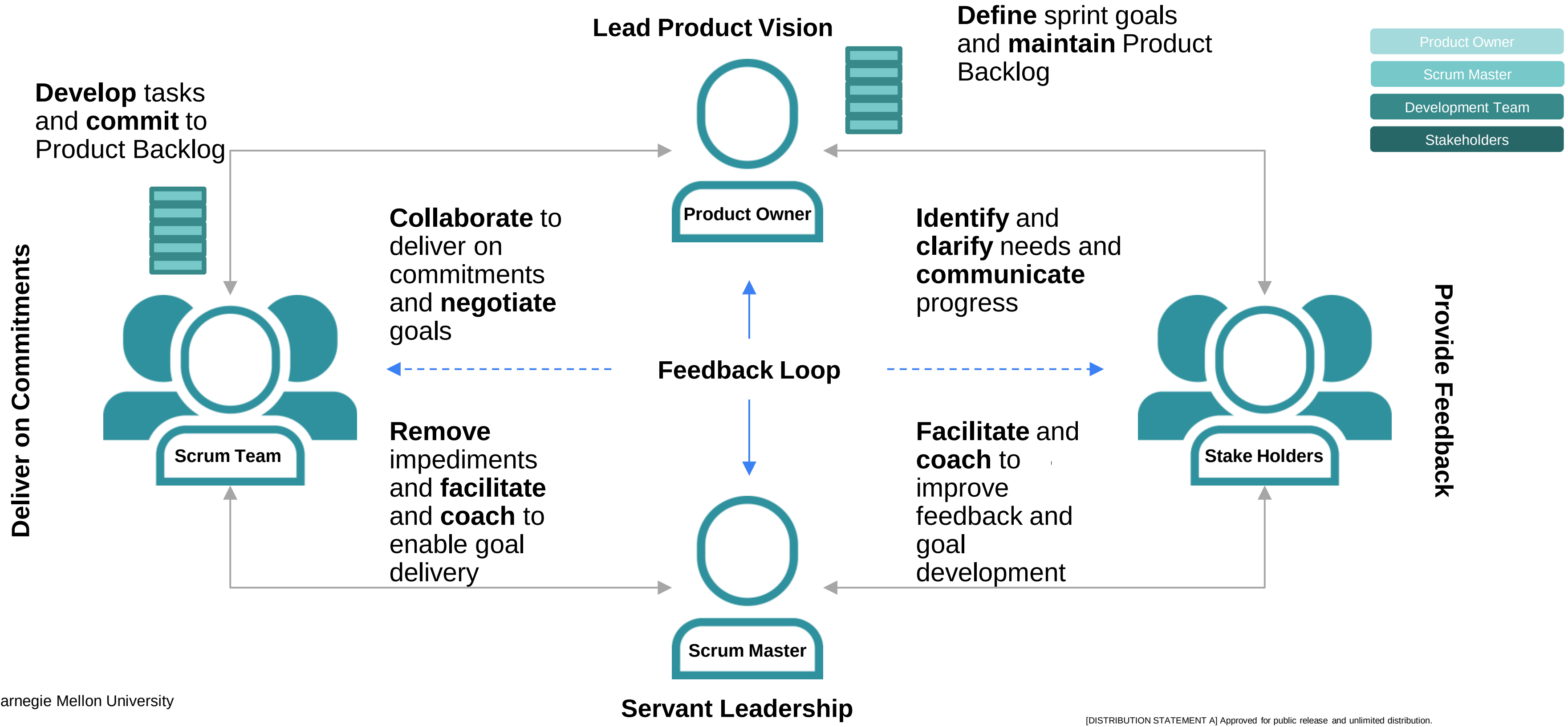
Carnegie Mellon University Software Engineering Institute

There are only three roles internal to a Scrum team, and stakeholders external to Scrum team typically a flat organizational structure.



**Product Backlog Refinement is not a specific Scrum Event*

Key Roles



Lead Product Vision

Define sprint goals and **maintain** Product Backlog

Product Owner
Scrum Master
Development Team
Stakeholders

Develop tasks and **commit** to Product Backlog

Deliver on Commitments

Collaborate to deliver on commitments and **negotiate** goals

Identify and **clarify** needs and **communicate** progress

Feedback Loop

Remove impediments and **facilitate** and **coach** to enable goal delivery

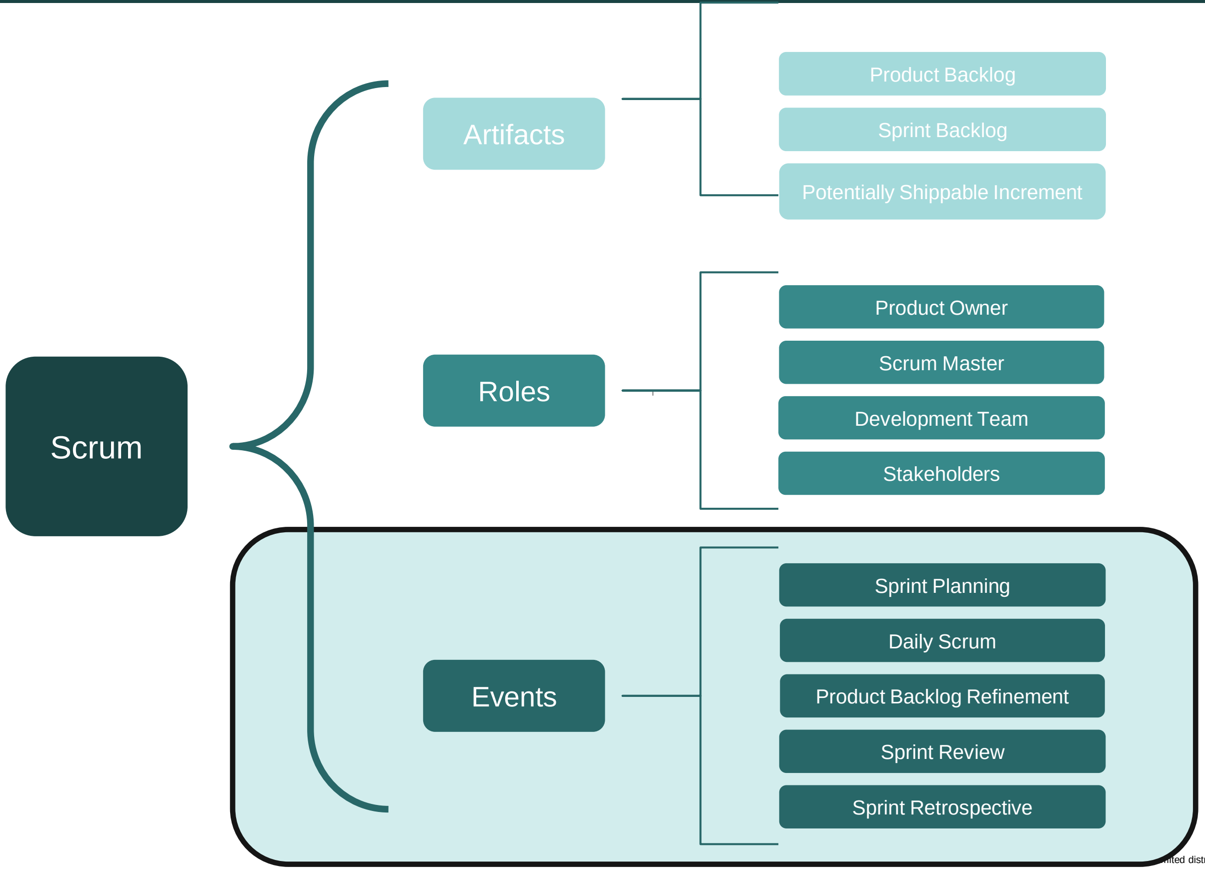
Facilitate and **coach** to improve feedback and goal development

Provide Feedback

Servant Leadership

Events

Product Backlog Refinement is not a specific Scrum Event



Scrum

Artifacts

- Product Backlog
- Sprint Backlog
- Potentially Shippable Increment

Roles

- Product Owner
- Scrum Master
- Development Team
- Stakeholders

Events

- Sprint Planning
- Daily Scrum
- Product Backlog Refinement
- Sprint Review
- Sprint Retrospective

Sprint Planning

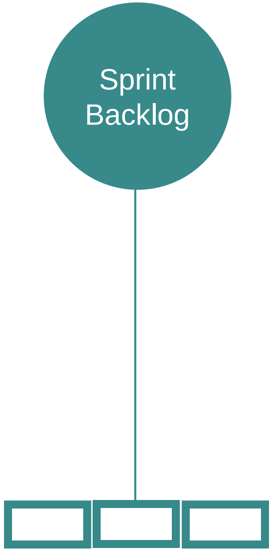
Sprint Planning

Daily Scrum

Product Backlog Refinement

Sprint Review

Sprint Retrospective



Sprint Backlog

The team:

1. Pulls in items from the Product Backlog into the team’s Sprint Backlog
2. These items are typically stories, but they may also be defects or other artifacts your program uses to track work
3. The team commits to *completing* these items in the Sprint.

Sprint Execution

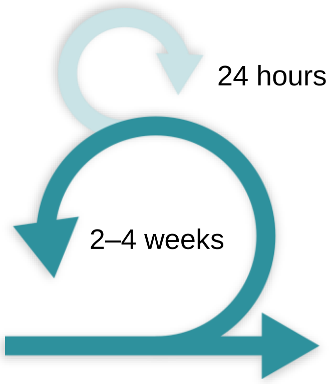
Carnegie Mellon University Software Engineering Institute

- Sprint Planning
- Daily Scrum
- Product Backlog Refinement
- Sprint Review
- Sprint Retrospective

Includes the normal day to day activities of the team: The daily Scrum (every 24 hours)

Communication between team members

Implement Tasks



Sprint Execution

Daily Scrum

Sprint Planning
Daily Scrum
Product Backlog Refinement
Sprint Review
Sprint Retrospective

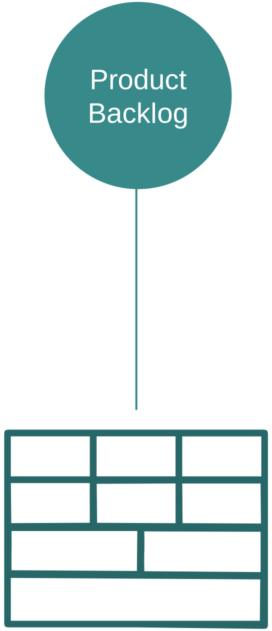
The Daily Scrum is a 15-minute event for the **Scrum Team** to synchronize activities and plans for the next 24 hours.

What did you do yesterday?

What are you planning to do today?

What obstacles are in your way?

Backlog Refinement



Product Backlog

Product Backlog

Backlog Refinement is held during the middle of the Sprint to give the team an opportunity to look forward at the work coming in future

Team should spend time making sure the backlog items are well defined, well understood, and well documented.

Sprint Review

Sprint Planning

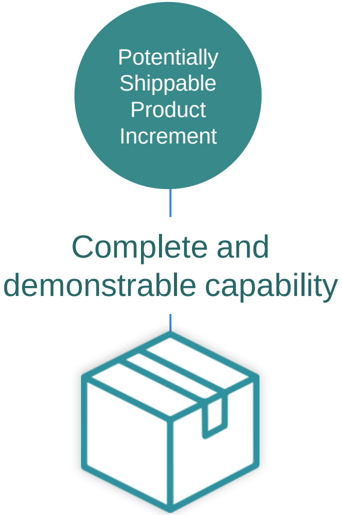
Daily Scrum

Product Backlog Refinement

Sprint Review

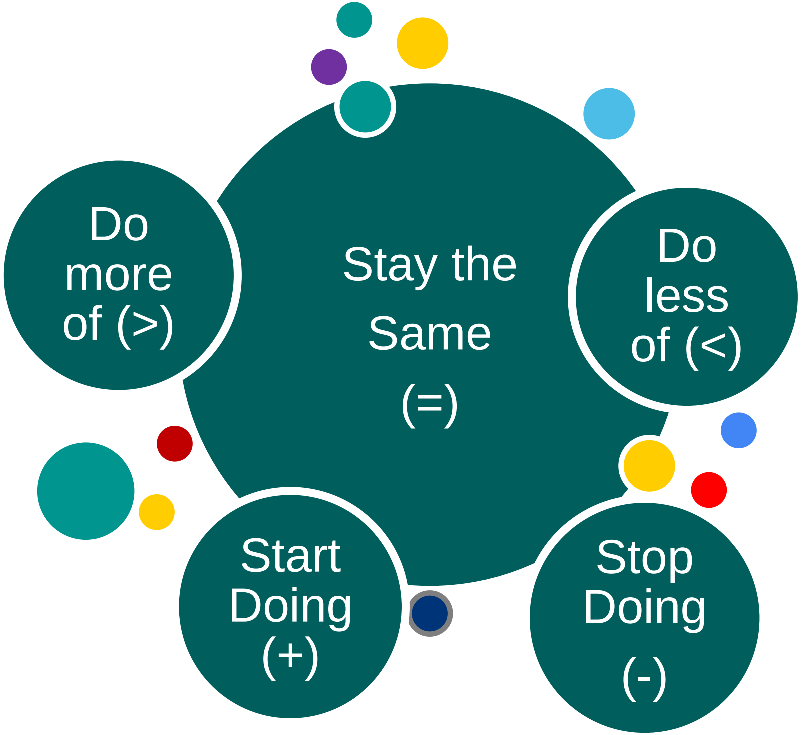
Sprint Retrospective

A Sprint Review is held at the end of the sprint to inspect the increment and adapt the Product Backlog if needed.


Potentially Shippable Product Increment
Complete and demonstrable capability

Potentially Shippable Product Increment

Sprint Retrospective



Sprint Planning

Daily Scrum

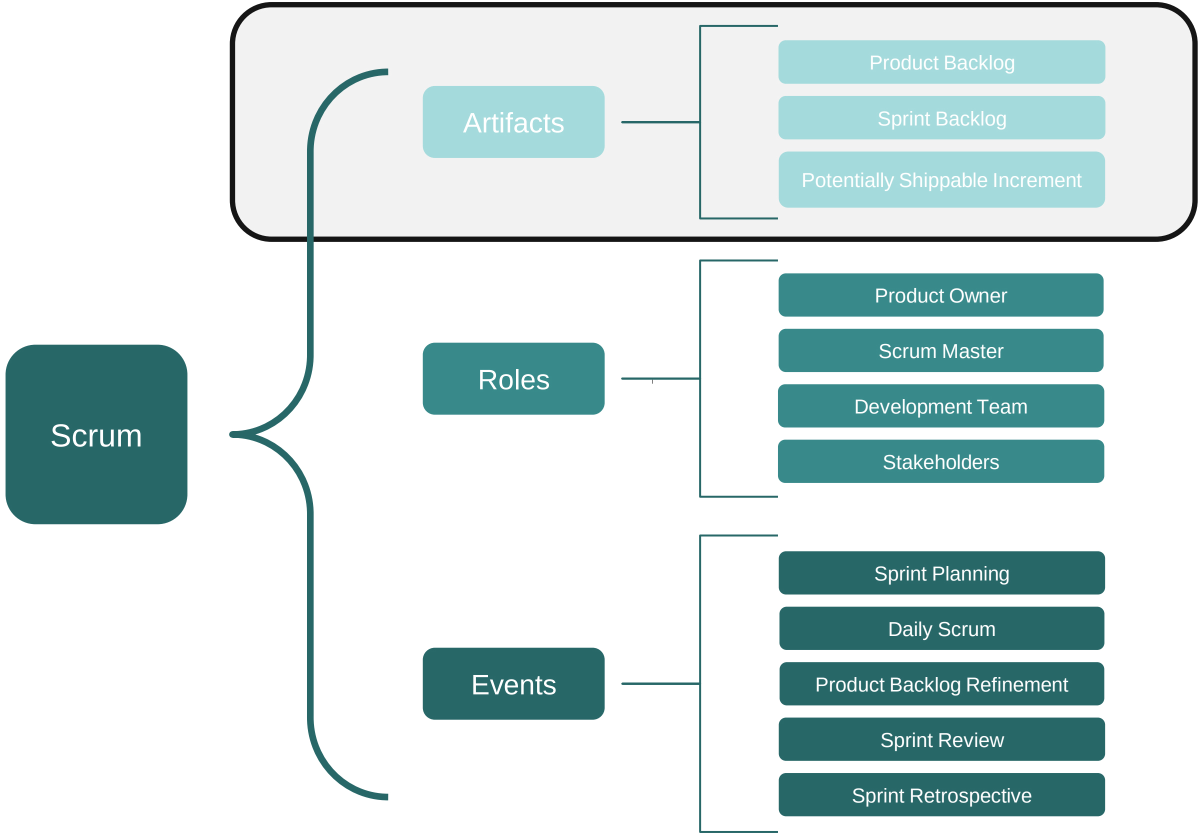
Product Backlog Refinement

Sprint Review

Sprint Retrospective

Artifacts

Carnegie Mellon University Software Engineering Institute



Scrum

Artifacts
- Product Backlog
- Sprint Backlog
- Potentially Shippable Increment

Roles
- Product Owner
- Scrum Master
- Development Team
- Stakeholders

Events
- Sprint Planning
- Daily Scrum
- Product Backlog Refinement
- Sprint Review
- Sprint Retrospective

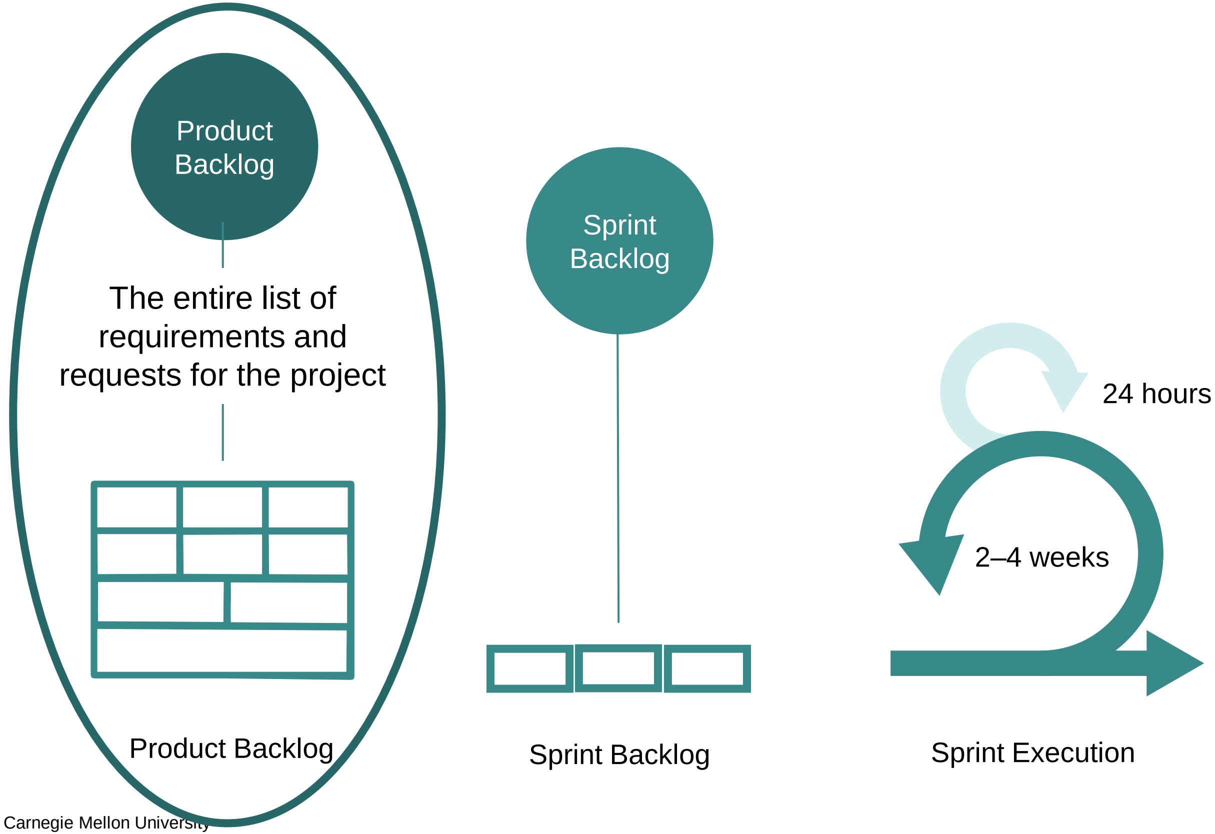
Product Backlog

Carnegie Mellon University Software Engineering Institute

Product Backlog

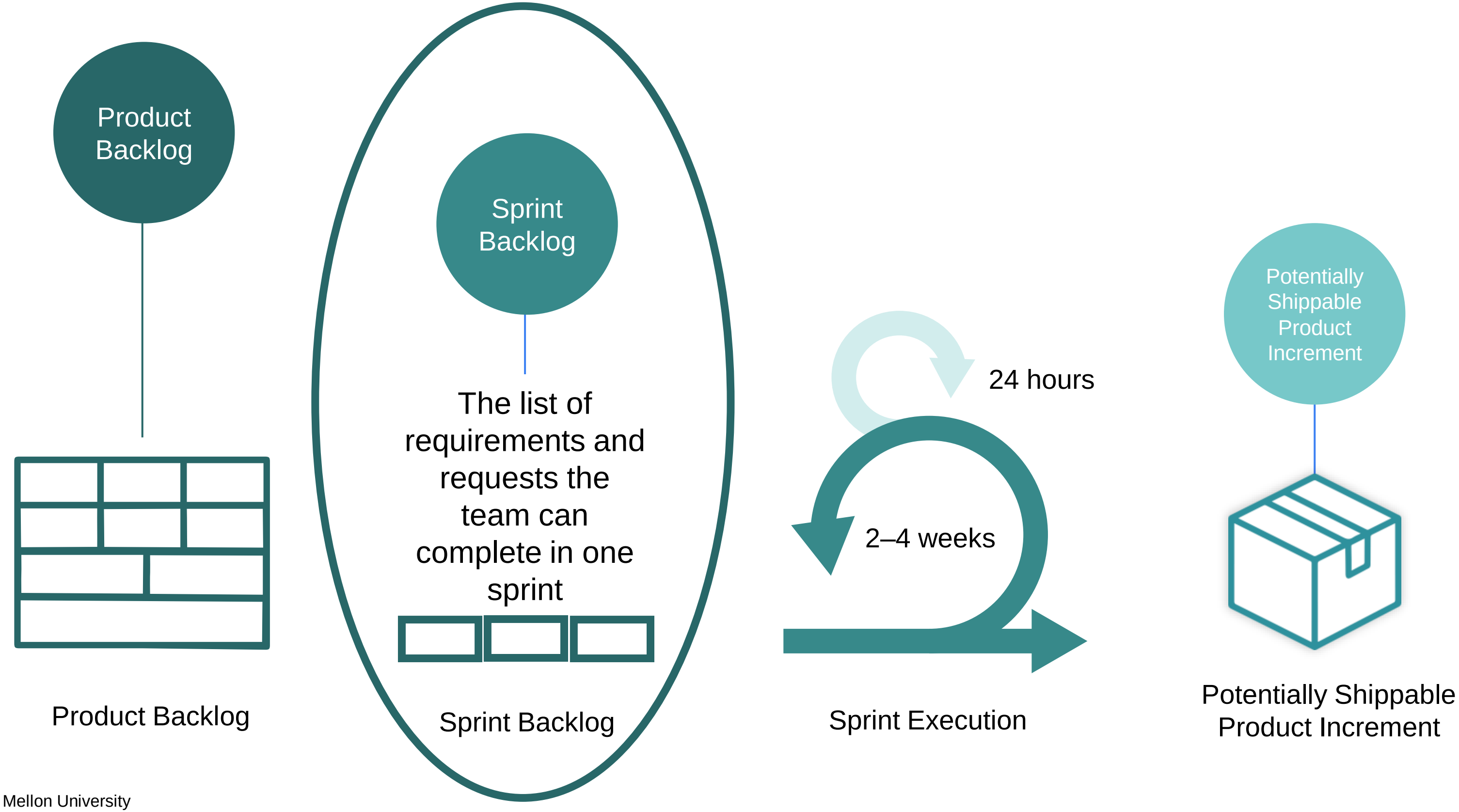
Sprint Backlog

Potentially Shippable Increment



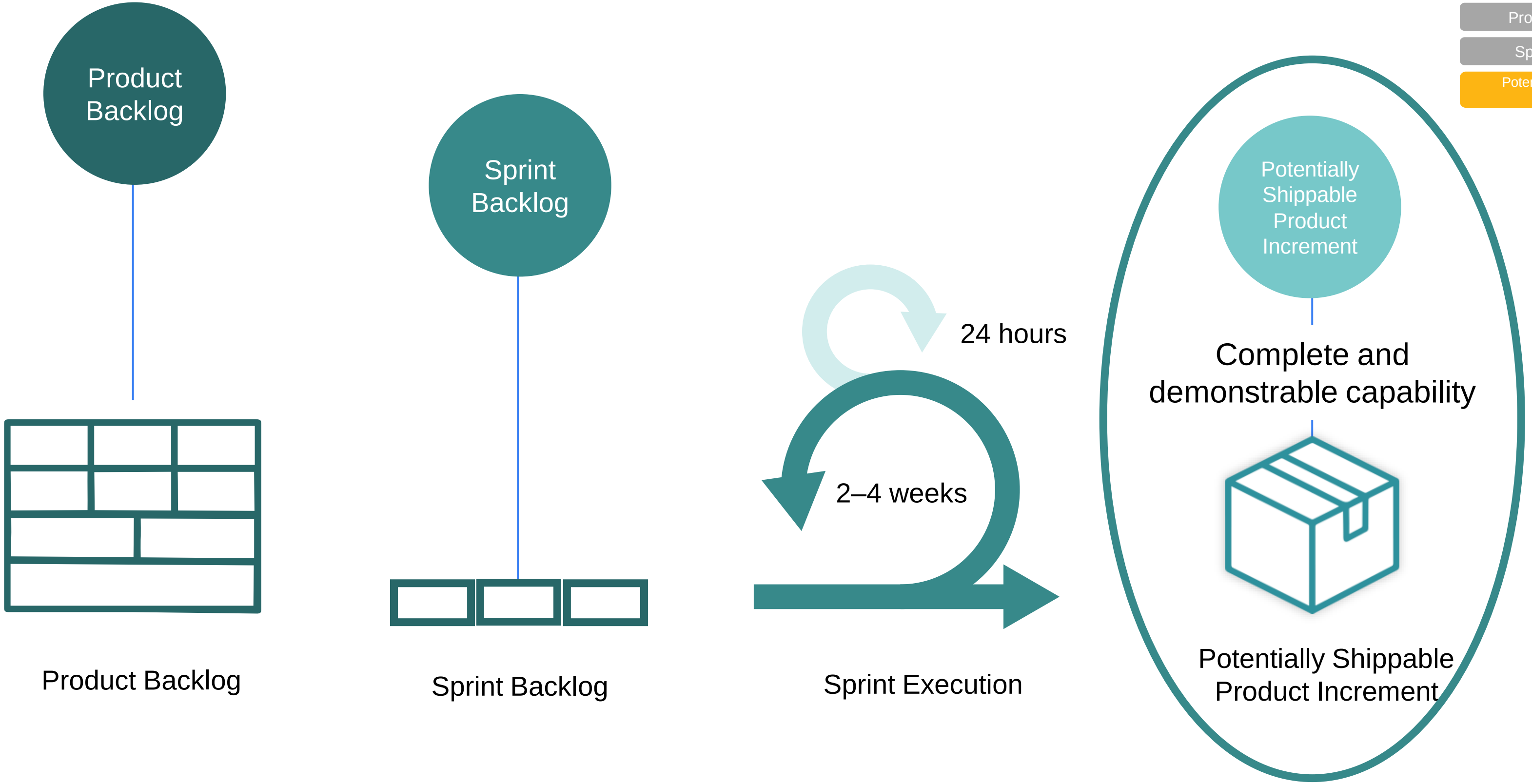
The entire list of requirements and requests for the project

Sprint Backlog



The list of requirements and requests the team can complete in one sprint

Potentially Shippable Increment


Product Backlog
Sprint Backlog
24 hours
2–4 weeks
Potentially Shippable Product Increment
Complete and demonstrable capability
Product Backlog
Sprint Backlog
Sprint Execution
Potentially Shippable Product Increment

Exercise

Carnegie Mellon University Software Engineering Institute

5 Minutes

Group Collaboration

As a group discuss the key differences you see between Agile and Waterfall.

Leadership in Orchestrating Digital Engineering

MBSE *mis*Use-Cases

Carnegie Mellon University Software Engineering Institute

Five Inter-Related Patterns

Model Based *SPECIFICATION* Engineering (MBSE)

Buy the Analysis Model; Receive Only the Diagrams

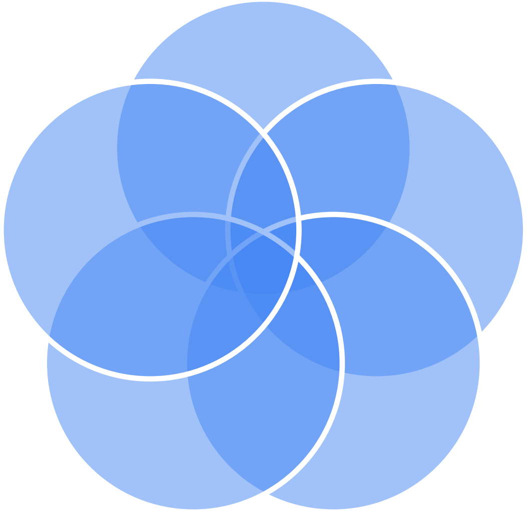
The Science Project for a Few Select Experts

Single Batch Mindset; or Big Modeling Up Front

The Shiny New Thing; Models, Models, Everywhere

In all of these stories, no malice is inferred

- *MBSE is reshaping people's roles*



Model Based SPECIFICATION Engineering (MBSE)

Precisely stated requirements are very desirable

- Failures often trace to poor requirements work

MBSE offers more utility from up-front precision

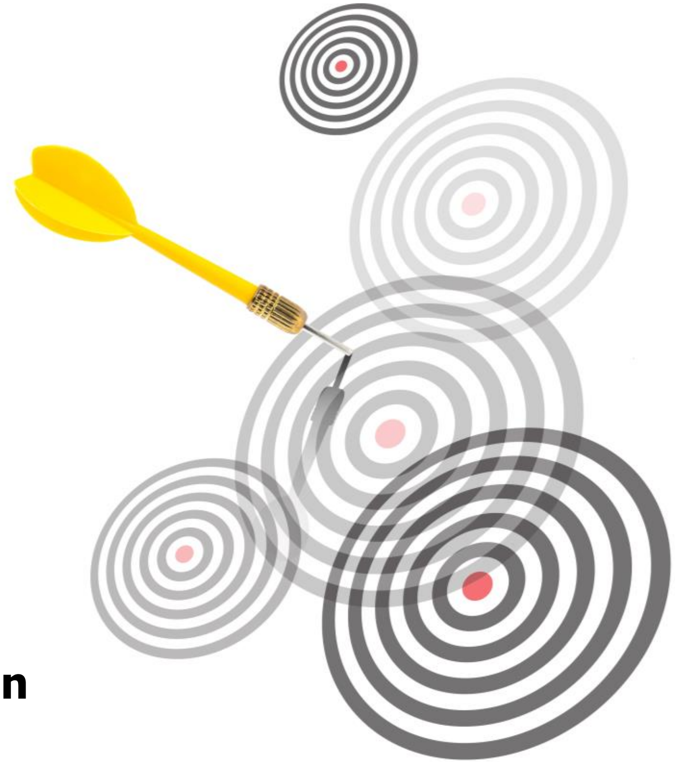
- We can exploit the trade space as we implement

Models are powerful tools with which to reason

- They also aid in communicating our aim

Evolution from descriptive to analytical models

- **Drive iteration in specification and implementation**



Buy the Analysis Model, Receive Only the Diagrams

Modeling activity yields important design artifacts

- Often included in Contract Data Requirement List

Roles of people who receive CDRLs often constrained

- Engineering roles sometimes limited to oversight

MBSE is accepted as an advancement in engineering

- Utility of the model goes deeper than the diagram

Grow a broader population of model users

- **We will build more useful models**

The Science Project

Ealy adopters need less support to use a new technology

- Early trial use yields unique insight for limited audience

Communities of practice can become isolated

- Perceived separation can inhibit broader adoption

Broader adoption requires different forms of insight

- Address State of the Practice and State of the Art

Develop knowledge to move theory into practice

- **Enables much broader technology adoption**

Single Batch Mindset

Architectural layers form a foundation to build upon

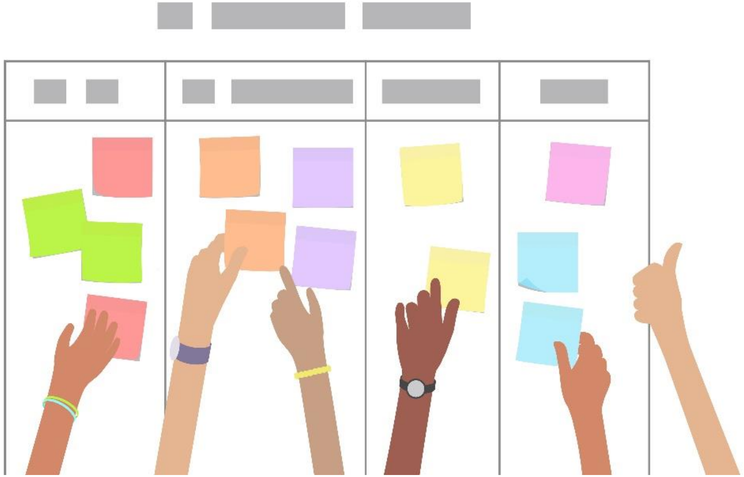
- Baselines promote a shared trade-space

How do we incrementally build the model?

- Can we iterate and learn before it’s “all done?”

Agile tries to solve several problems, including:

- “Nothing is visible until everything is done”



Promote logical decomposition in modeling

- **Iterative evaluation of results drive quality**

The Shiny New Thing

Enthusiasm can drive adoption and progress

- Motivated people make things happen

When the only tool you have is a hammer…

- ~~Everything starts to look like a nail~~
- You can capture everything using SysML

Practice of MBSE is still largely in the early stages

- Some industry participants have robust solutions

Resist over-application of MBSE tools/methods

- **Prevent dilution of the value proposition**

Five Inter-Related Patterns

Model Based *SPECIFICATION* Engineering (MBSE)

- Misunderstanding of intended utility

Buy the Analysis Model; Receive Only the Diagrams

- Mismatch in realization of MBSE intent

The Science Project for a Few Select Experts

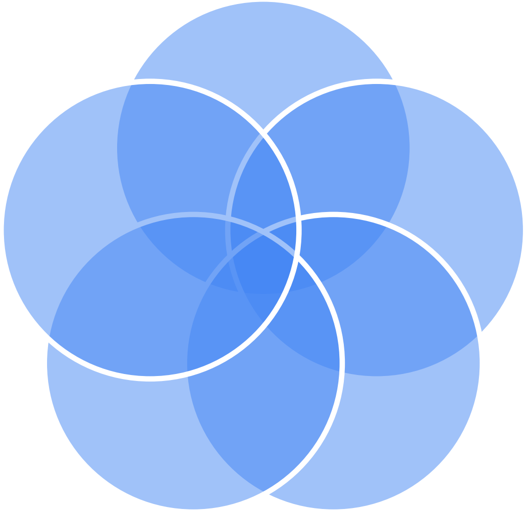
- Missed opportunity for greater value

Single Batch Mindset; or Big Modeling Up Front

- Traditionalism limiting impact

The Shiny New Thing; Models, Models, Everywhere

- Mis-aligned ambition



Leadership in Orchestrating Digital Engineering

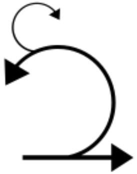
Introduction to Industrial DevOps

Carnegie Mellon University Software Engineering Institute

Background

21st century development approaches, such as Agile and DevOps have benefitted ***small initiatives*** with a ***single team building software*** in their ability to respond to change, reduce product delivery schedules, reduce product cost, increase product quality, and Increase employee morale.

What are the benefits and challenges if we use those same practices at scale for ***large Initiatives*** with ***multiple teams*** who are ***building cyber-physical safety critical systems*** with a scope that includes software, firmware, and hardware development.



Agile – An iterative and incremental approach to project management. It aims to improve project ability to adapt to changing needs empirically.

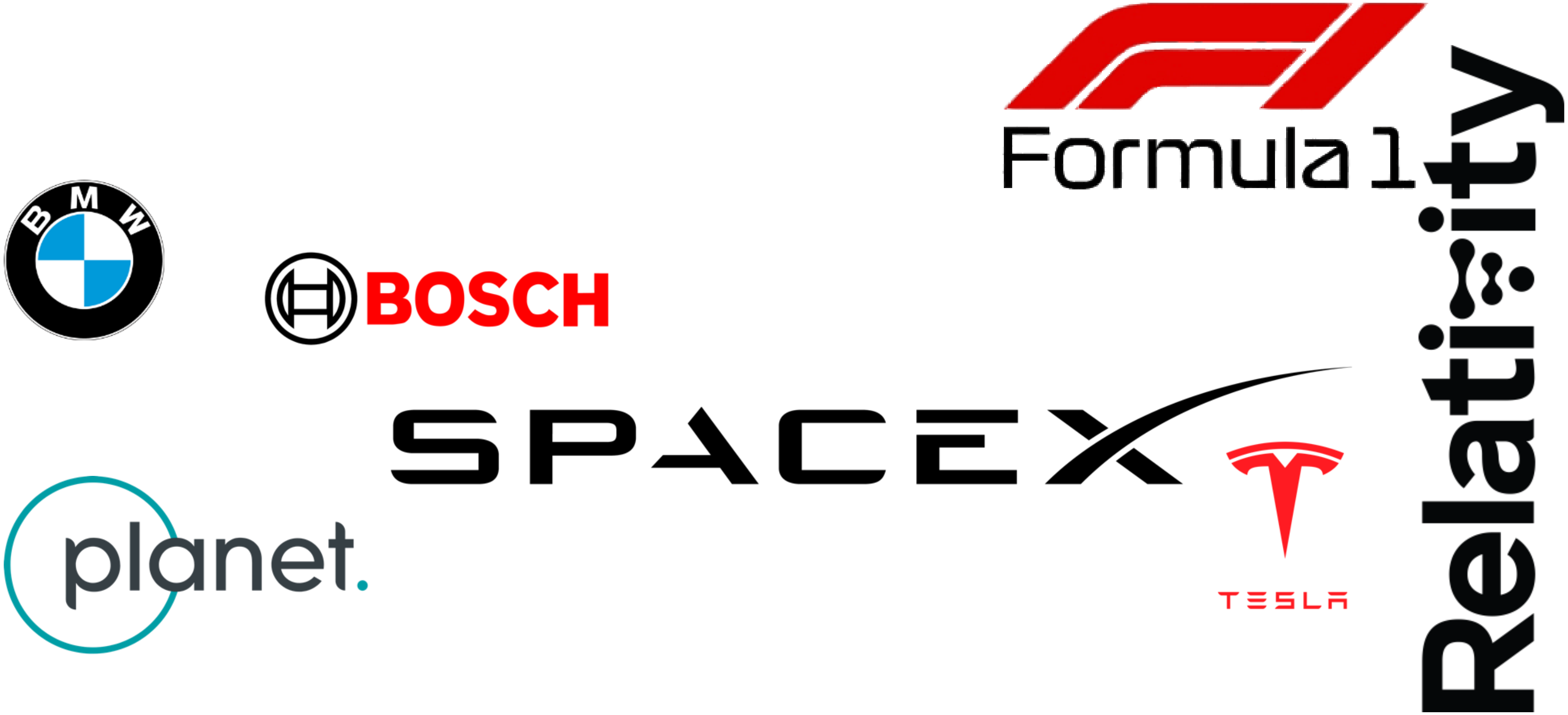
Typically used in software projects

DevOps – A set of practices that combines development (Dev) and operations (Ops). It aims to shorten the systems development life cycle.

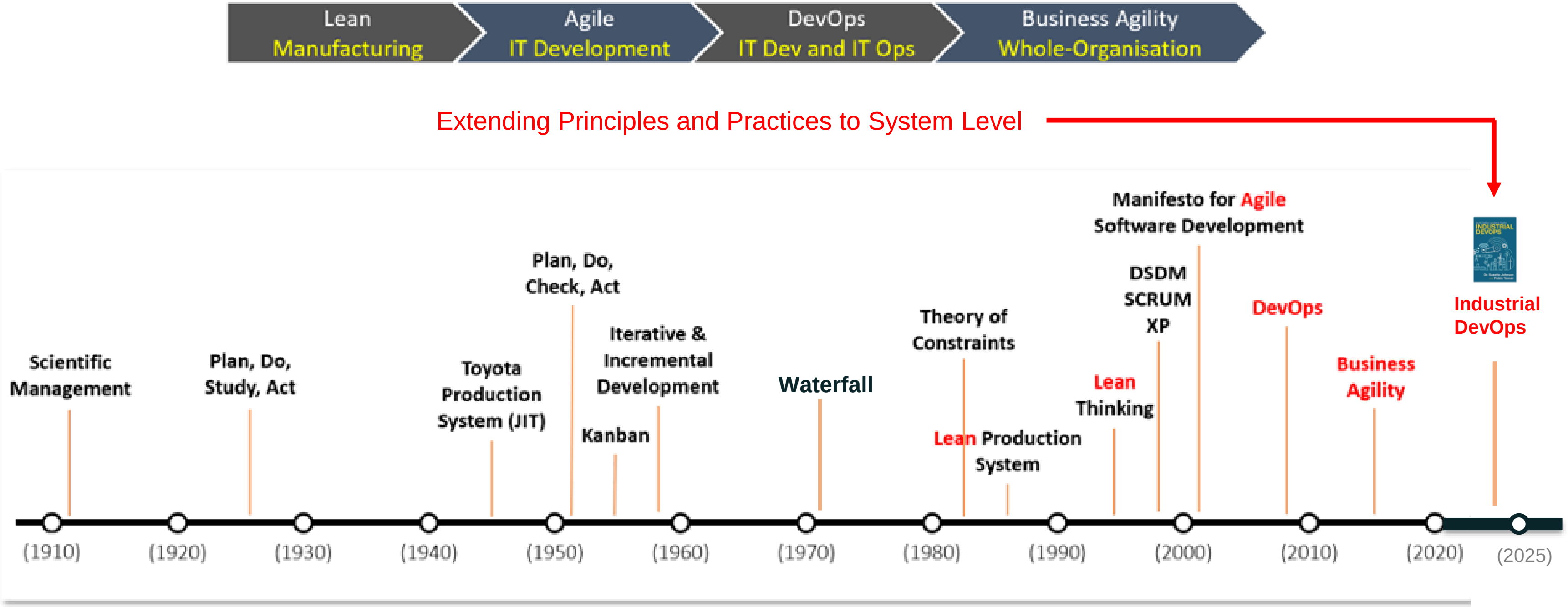
Typically used in software projects

The motivation to migrate to Agile is a demand for ***faster*** *delivery, difficulty managing* ***change****, and increased product* ***complexity****.*

Early Adopters



Evolution



Lean
Manufacturing

Agile
IT Development

DevOps
IT Dev and IT Ops

Business Agility
Whole-Organisation

Extending Principles and Practices to System Level

Scientific Management

Plan, Do, Study, Act

Toyota Production System (JIT)

Plan, Do, Check, Act

Kanban

Iterative & Incremental Development

Waterfall

Theory of Constraints

Lean Production System

Lean Thinking

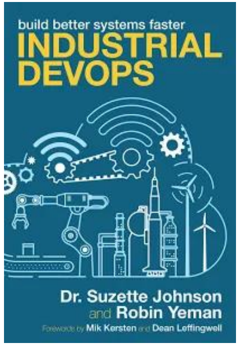
DSDM
SCRUM
XP

Manifesto for Agile Software Development

DevOps

Business Agility

Industrial DevOps

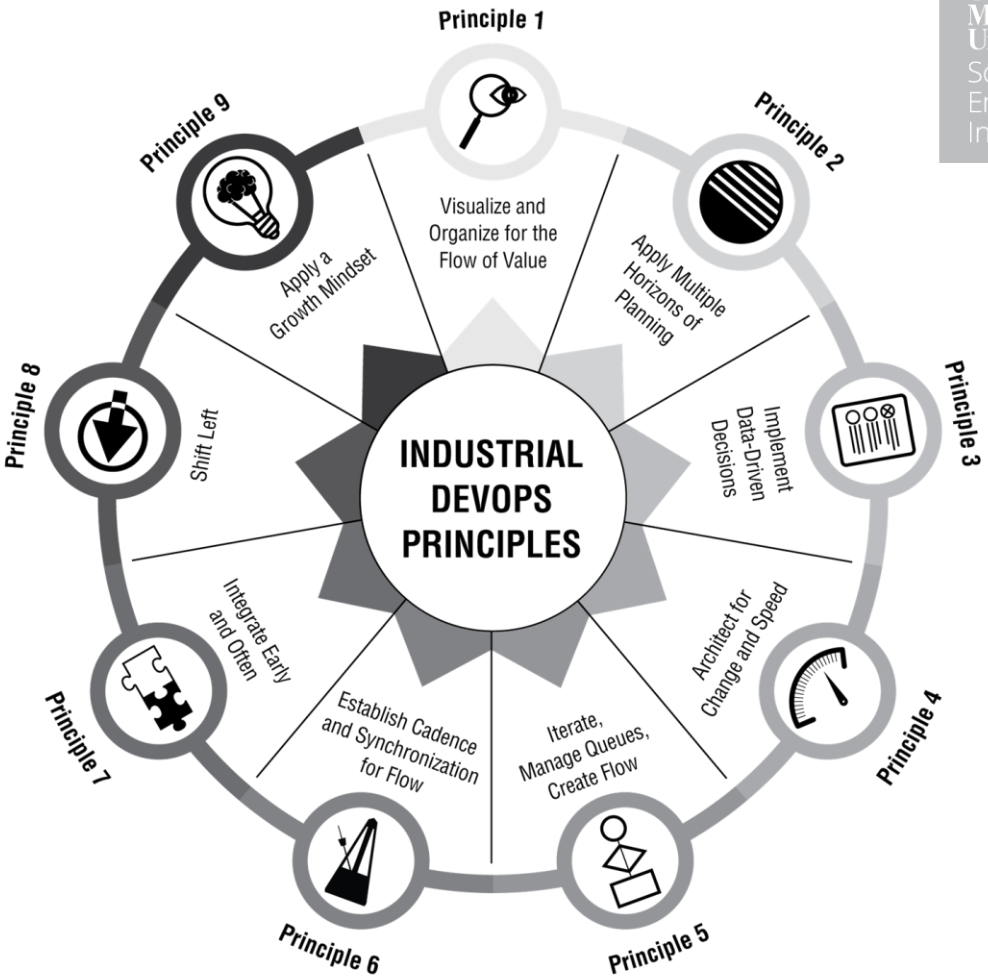


build better systems faster

INDUSTRIAL DEVOPS

Dr. Suzette Johnson and Robin Yeman

Forewords by Mik Kersten and Dean Leffingwell


INDUSTRIAL DEVOPS PRINCIPLES
Principle 1
Visualize and Organize for the Flow of Value
Principle 2
Apply Multiple Horizons of Planning
Principle 3
Implement Data-Driven Decisions
Principle 4
Architect for Change and Speed
Principle 5
Iterate, Manage Queues, Create Flow
Principle 6
Establish Cadence and Synchronization for Flow
Principle 7
Integrate Early and Often
Principle 8
Shift Left
Principle 9
Apply a Growth Mindset

PMO R&D SE HW
FW
SW Test
QA MFG OPS

PMO

R&D

SE

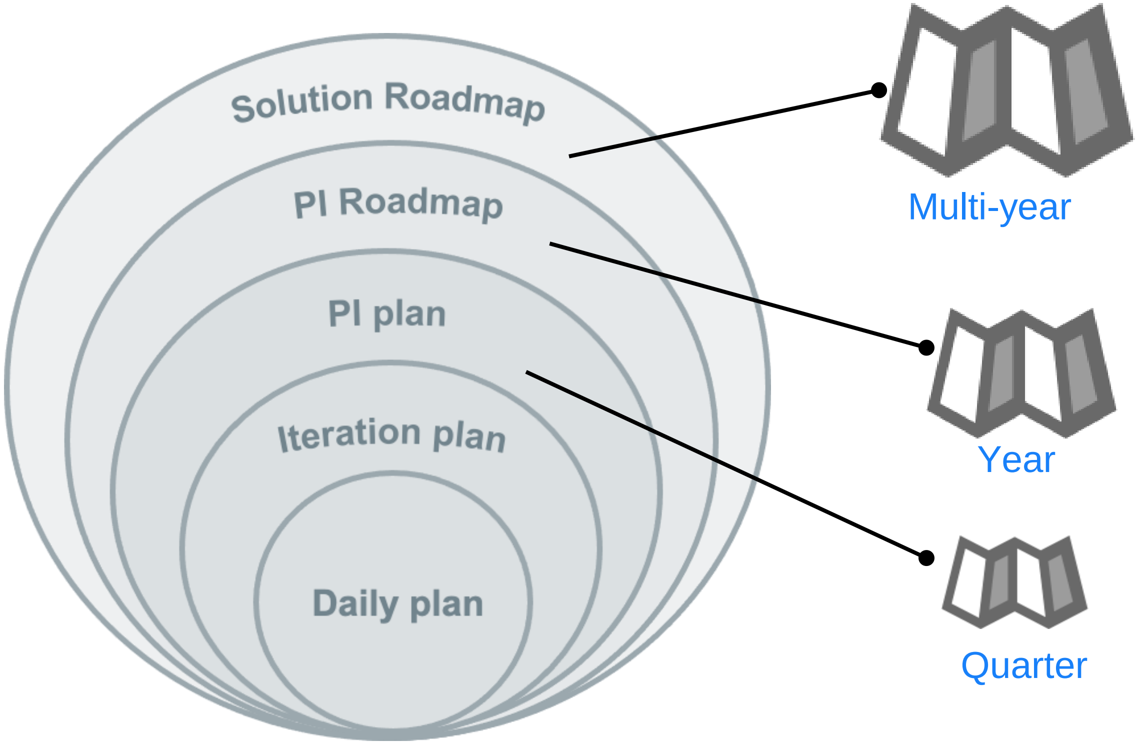
HW
FW
SW

Test
QA

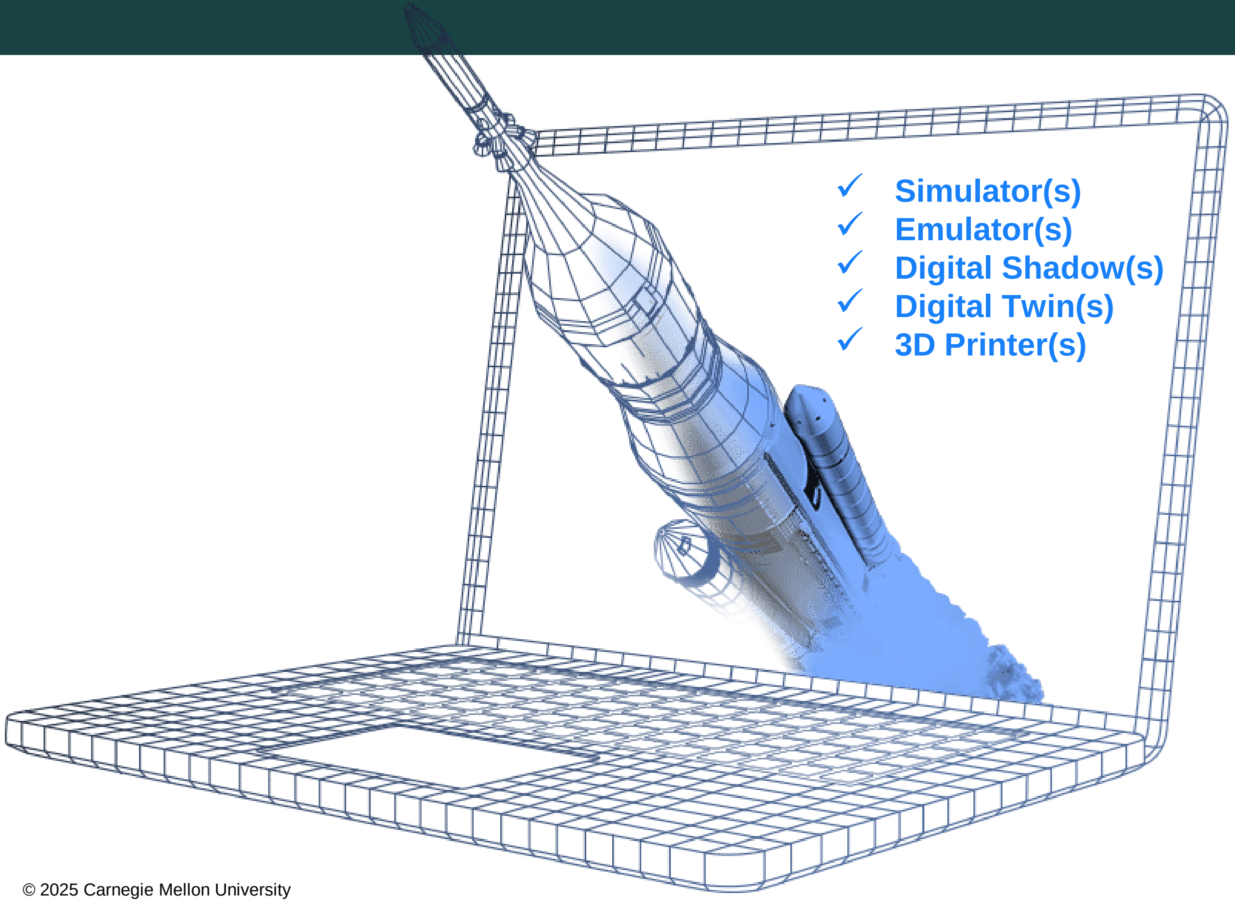
MFG

OPS

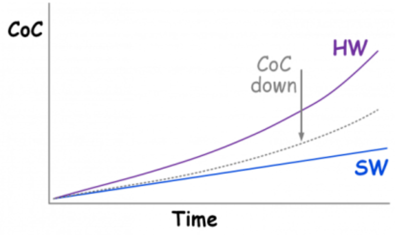
Multiple Horizons of Planning



Implement data driven decisions

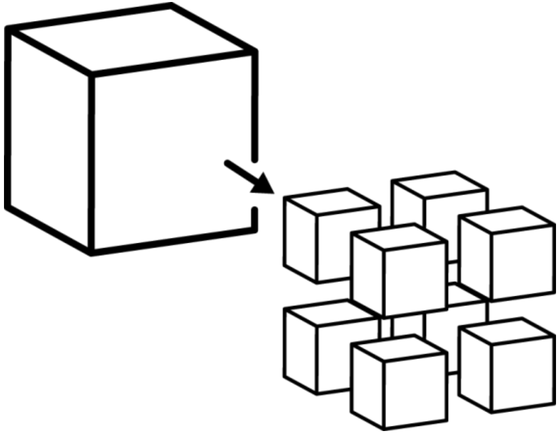


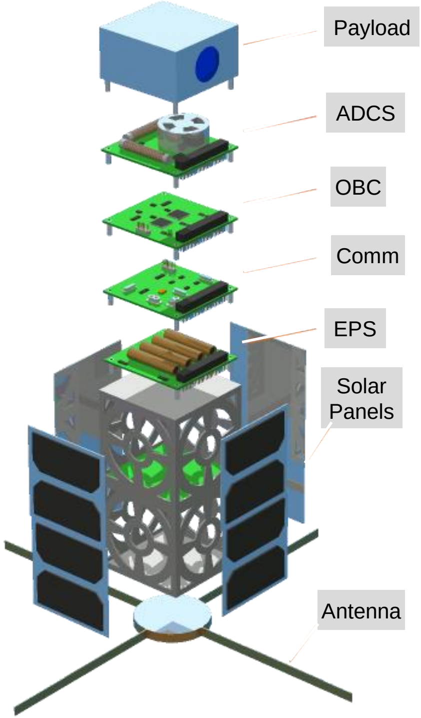
- ✓ Simulator(s)
- ✓ Emulator(s)
- ✓ Digital Shadow(s)
- ✓ Digital Twin(s)
- ✓ 3D Printer(s)

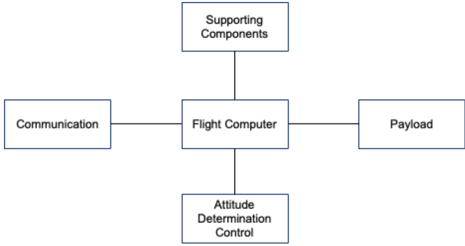


Hardware has multiple Options for fast feedback

Architect for Change and Speed



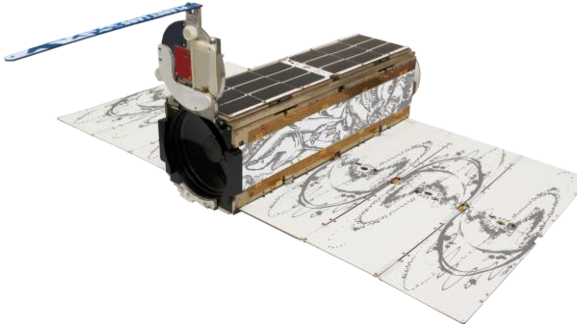


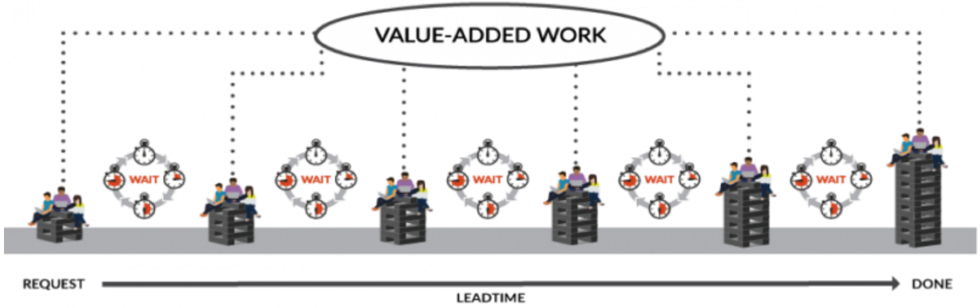


Cyber-physical systems need modular, open architecture with standardized interfaces for both software and hardware

Iterate, manage queues, create flow

Planet Labs frequently releases new spacecraft designs and tests the in space making changes based on the results, they have completed 14 iterations since 2012





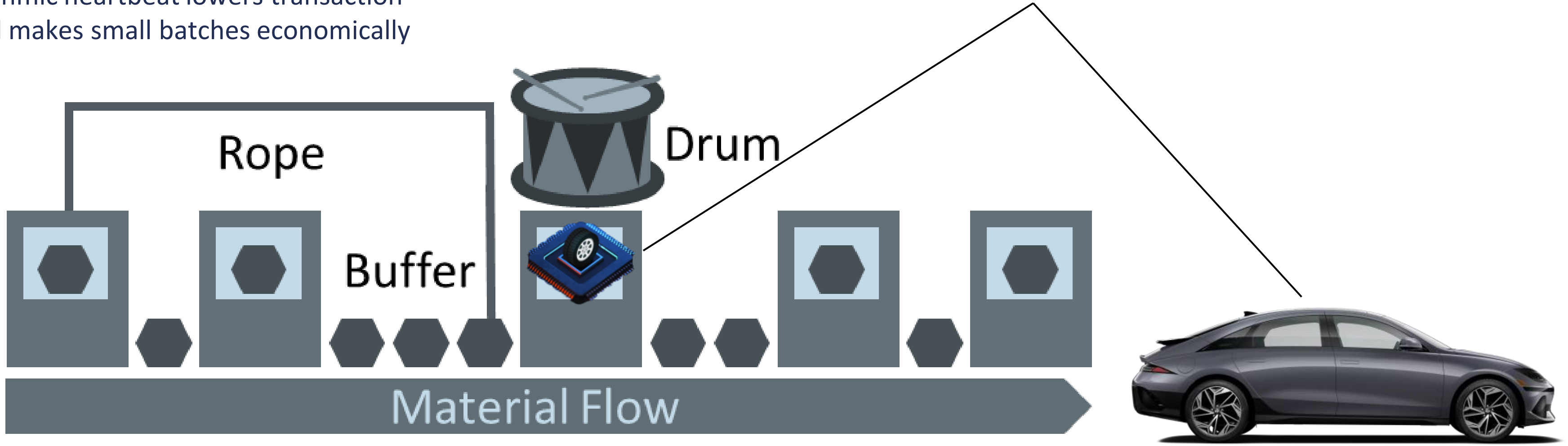
P6 Synchronization

Keeping activities in synchronized reduces waste and maximizes flow around the bottleneck.

Automobile manufacturers drastically cut cadence of vehicle development to subordinate to the constraint of semi-conductor chips.

P6 Cadence

The rhythmic heartbeat lowers transaction cost and makes small batches economically feasible.

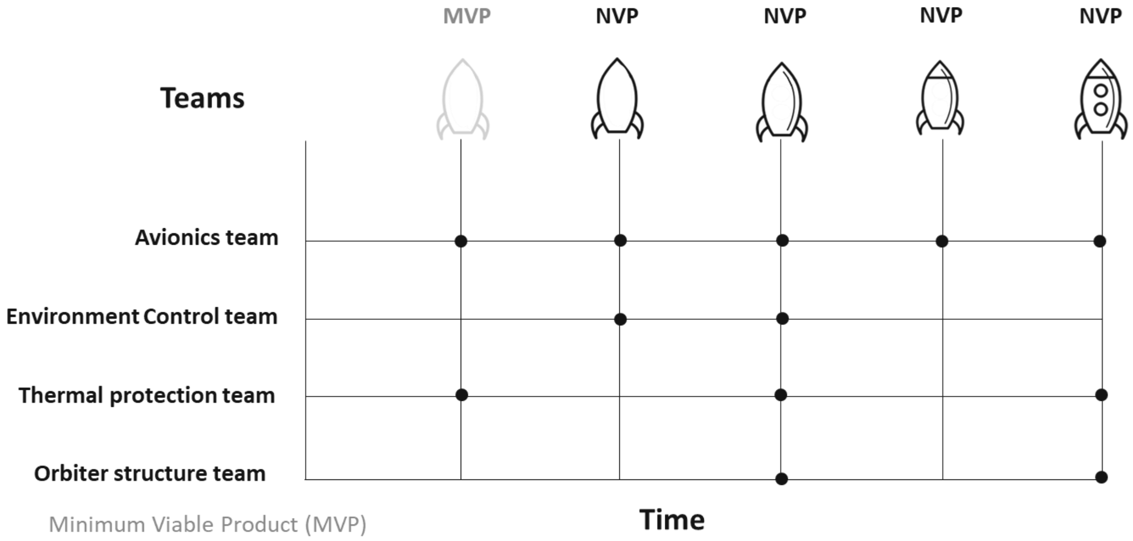


© 2025 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Integrate early and often

Not all elements are going to be integrated at the time, but its critical to integrate as frequent as possible.



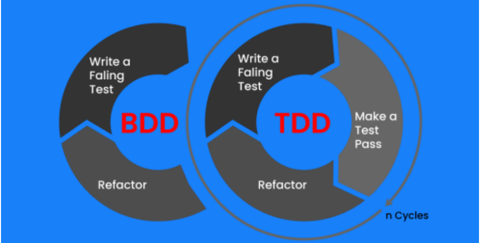
Teams	MVP	NVP	NVP	NVP	NVP
Avionics team	●	●	●	●	●
Environment Control team		●	●		
Thermal protection team	●		●		●
Orbiter structure team			●		●

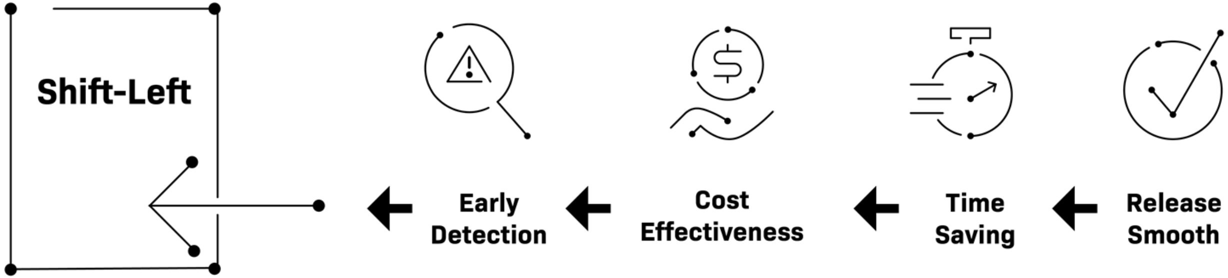
Minimum Viable Product (MVP)

Next Viable Product (NVP)

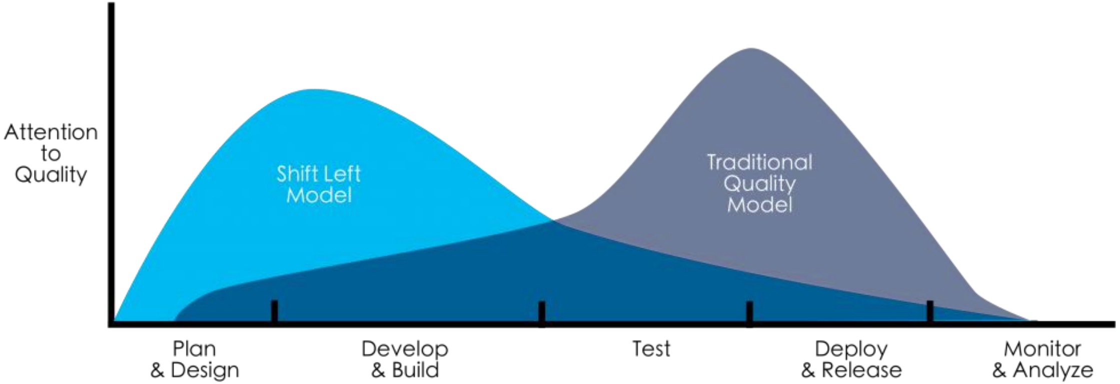
Shift Left

Carnegie Mellon University Software Engineering Institute





Shift Left everything : Test, Data Sets, Security, Safety, and manufacturing.

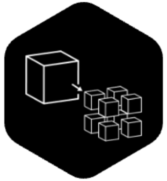


Industrial DevOps Principal Icons

Organize for flow of value

Multiple Horizons of planning

Data driven decisions



Architecture for speed and change

Iterate and improve flow

Cadence and Synchronization

Integrate early and often

Shift Left

Growth Mindset

Reference: Johnson and Yeman. Industrial DevOps. 2023. IT Revolution.

Exercise

5 Minutes

Group Collaboration

As a group discuss the following:

1. Which principle would have the greatest impact on your organization?
2. Which principle would be the most difficult to implement?
3. What would be needed to overcome obstacles?

Key Take Aways

- There are 9 principles to build better systems faster
- There need to manage a rapidly evolving environment
- New adopters are seeing great success

Leadership in Orchestrating Digital Engineering

Managing Complexity with systems thinking (*with MBSE*)

Objectives

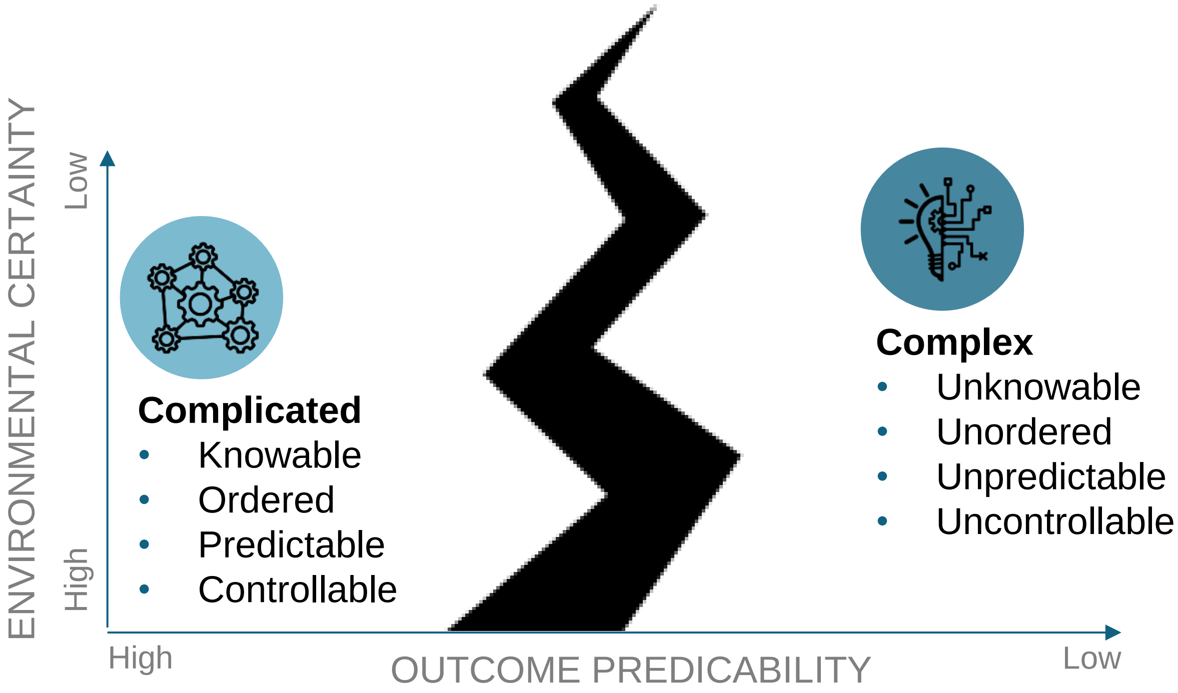
Objectives

- Complex versus Complicated
- Systems Thinking
- The steps to simplify the system

The Complexity Chasm

P3 P4

Cyber-Physical systems are often complex with many unknowns and emergent behavior



Complicated

- Knowable
- Ordered
- Predictable
- Controllable

Complex

- Unknowable
- Unordered
- Unpredictable
- Uncontrollable

Complex Systems

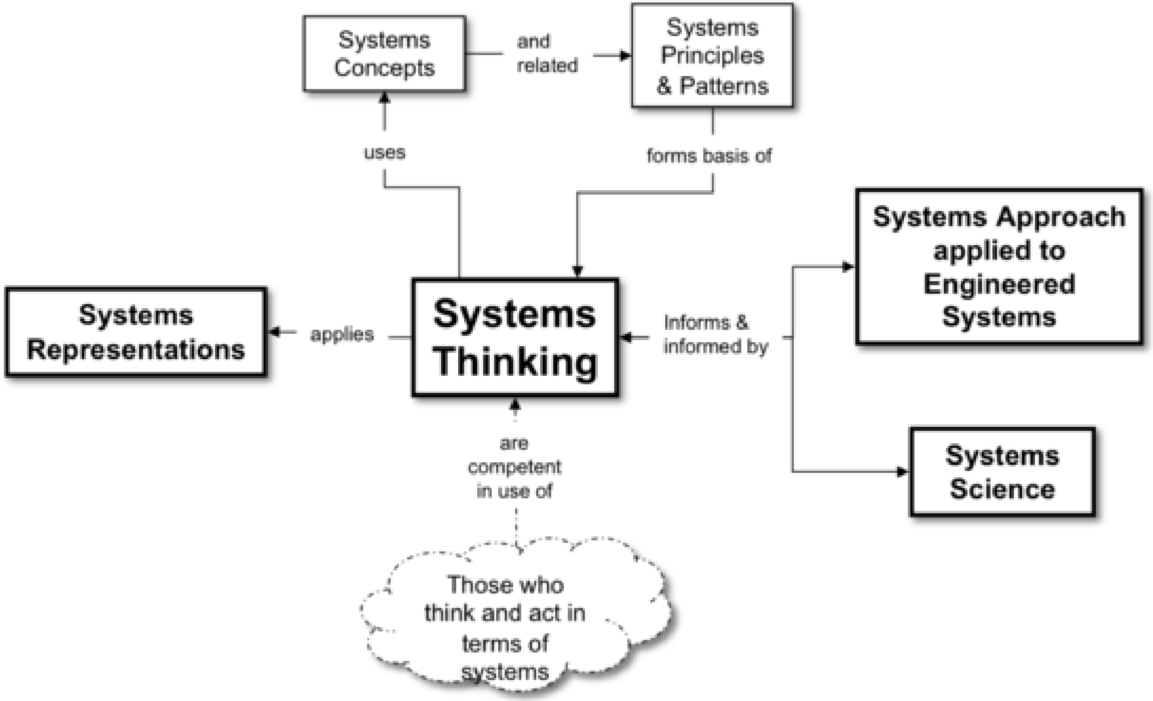
Systems Thinking

P3

P4

Some Principles

- Interconnectedness
- Synthesis
- Feedback loops

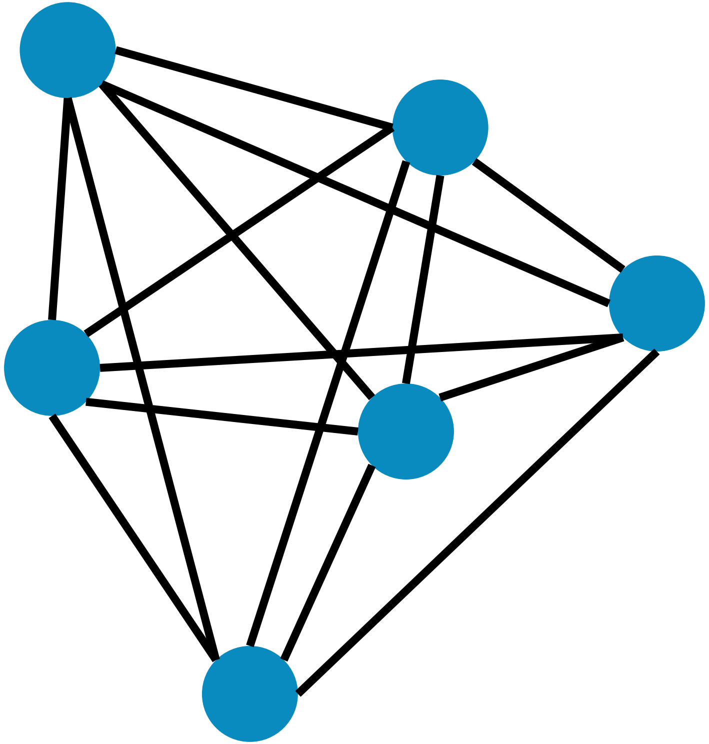


Systems Thinking. (20 November 2023). SEBoK, . Retrieved <citation>22:21 UTC, November 18, 2023</citation> from https://sebokwiki.org/w/index.php?title=Systems_Thinking&oldid=69582.

Interconnectedness

P3 P4

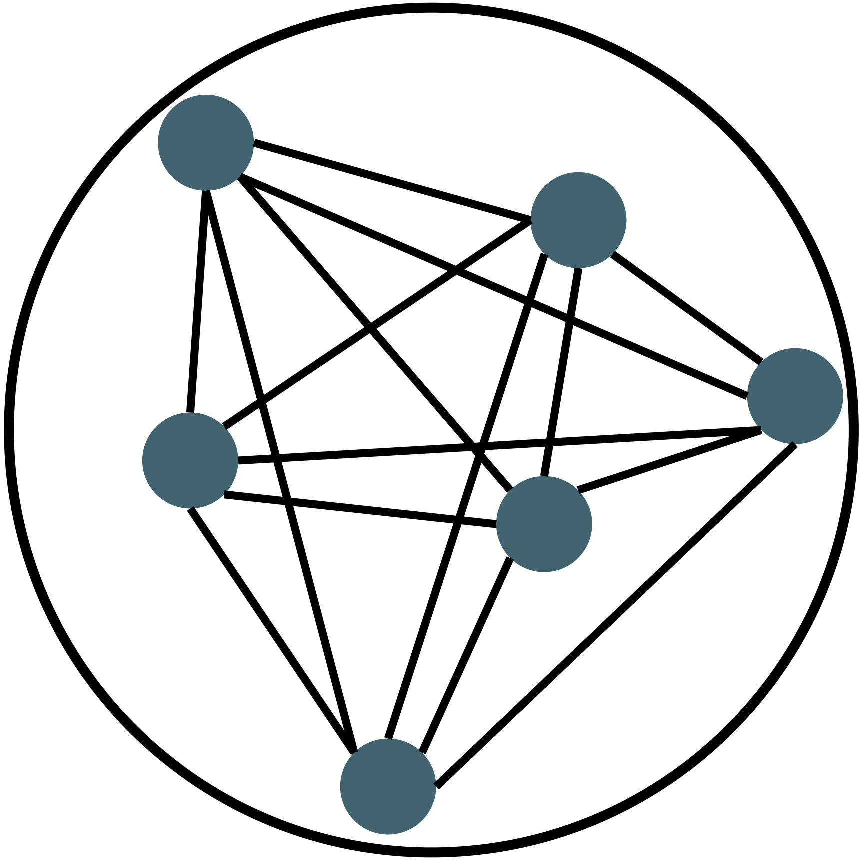
The fundamental understanding that everything is connected.



Synthesis

P3 P4

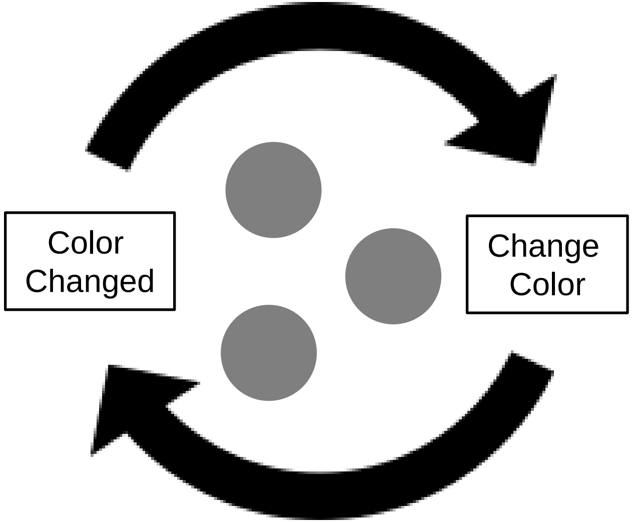
Combination of two or more parts to form a newer whole that helps us understand a larger system



Feedback Loops

P3 P4

Occurs when a change in something ultimately comes back to cause a further change in the same thing

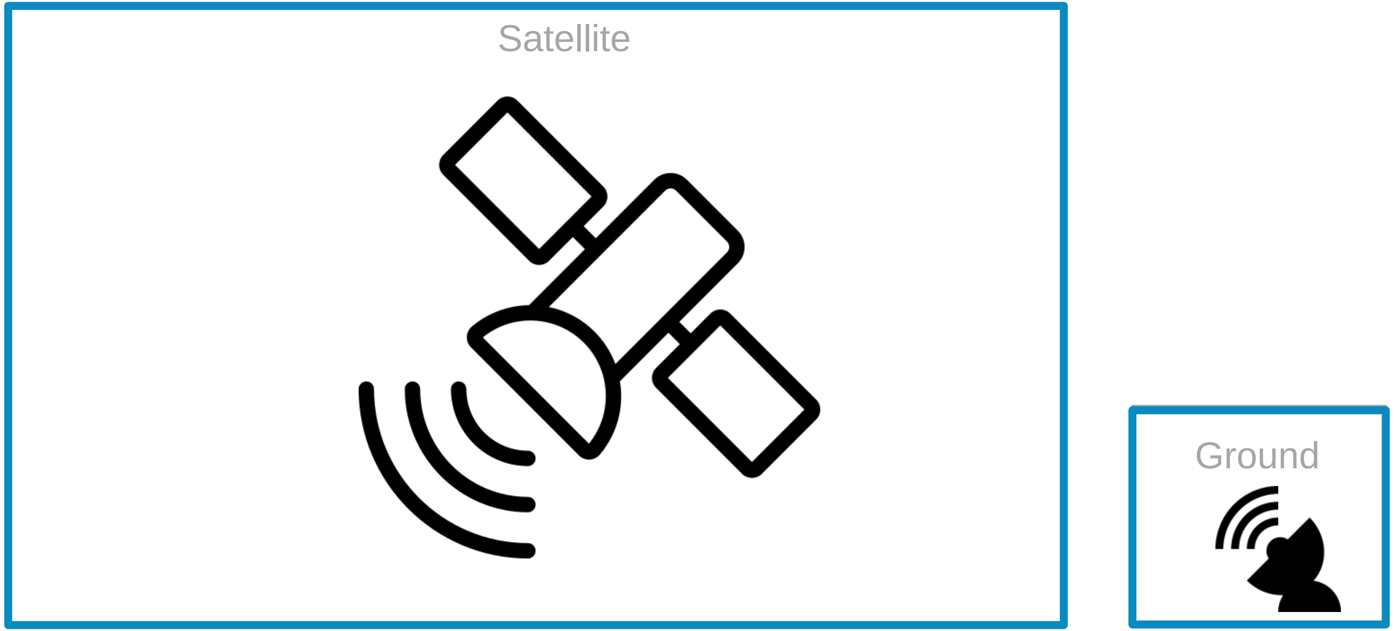


Define System Boundaries

P3 P4

Reduce complexity by bounding the system of interest, what's in and what's outside

Here you can see that the Satellite is a bounded system, and the ground is a bounded system



Synthesis

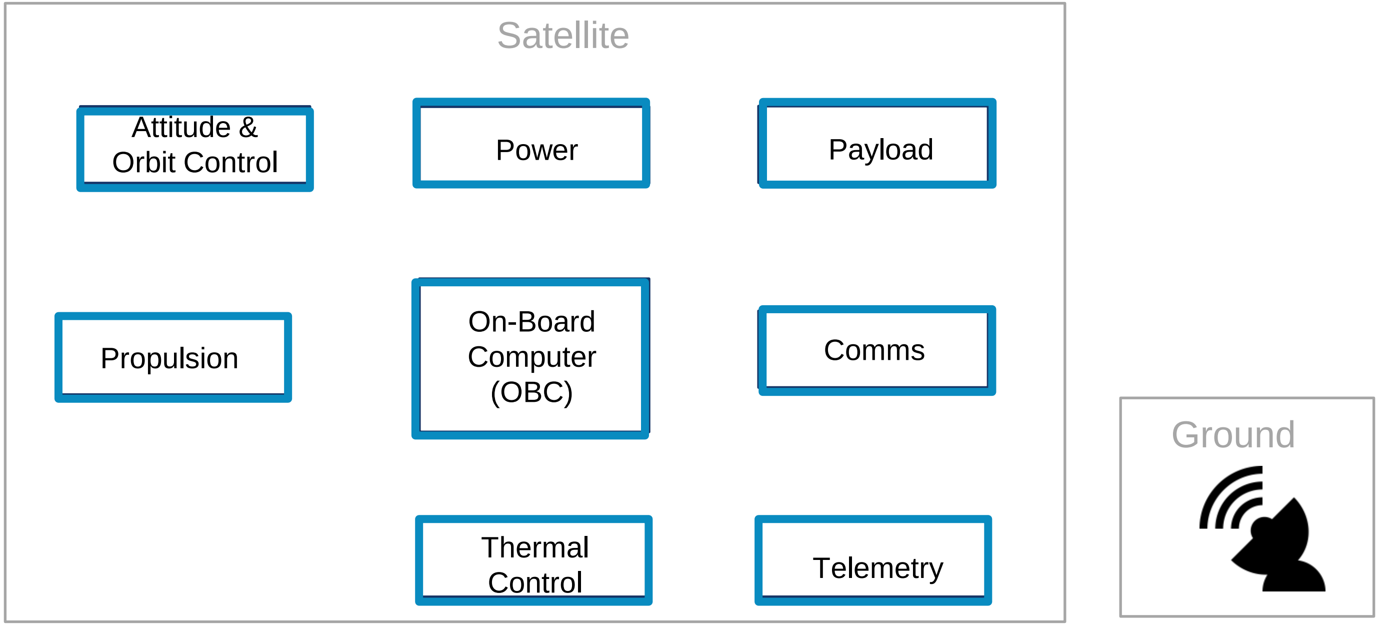
Identify key Components

Carnegie Mellon University Software Engineering Institute

P3 P4

Key Components

1. ***On-Board Computer***
2. ***Power***
3. ***Attitude & Orbit Control***
4. ***Propulsion***
5. ***Thermal Control***
6. ***Payload***
7. ***Communications***
8. ***Telemetry***

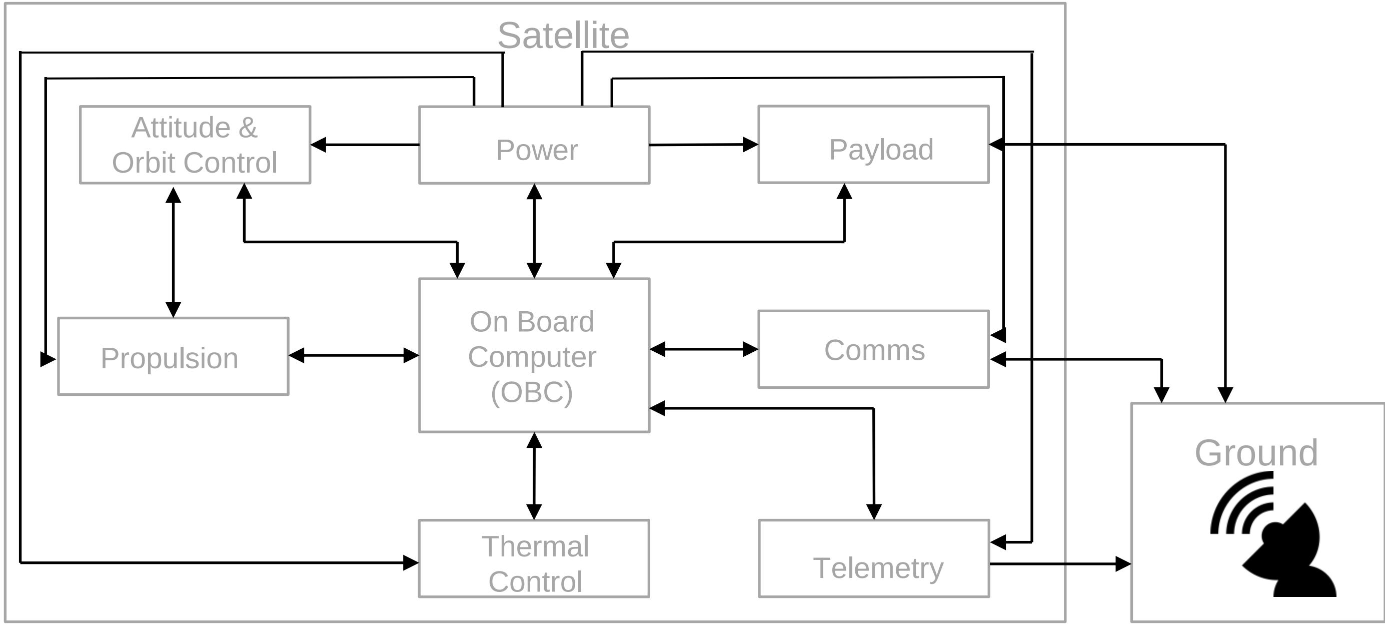


[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Map Interactions

P3 P4

How to our components connect to one another ?



Interconnectedness

Exercise

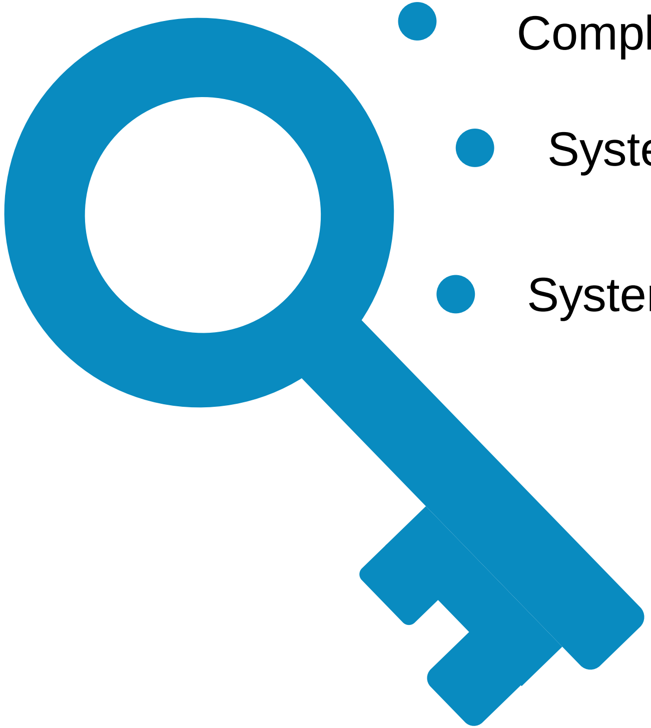
15 Minutes

Group Collaboration

Select a system from one of your contexts

1. Determine whether the system is complicated or complex, and document reasons.
2. Identify which principles from Systems thinking you would use to simplify.
3. Document the steps you took to simplify your system.

Key Take Aways



- Complex versus complicated
- Systems Thinking principles
- System Simplification

Leadership in Orchestrating Digital Engineering

Engineering for safety and regulatory compliance

Objectives

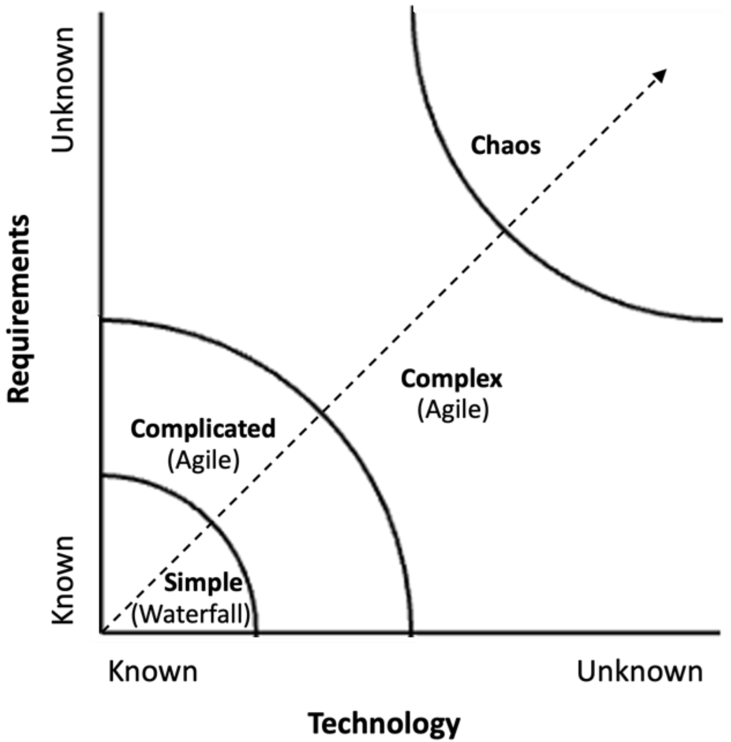
Objectives

- Iterative approach to regulatory compliance
- Integrate Safety into sprints
- Risk Adjusted Backlogs
- Right size documentation and traceability
- Iterative Design Reviews

Agile to manage complexity

P4 P8

Agile is meant to manage uncertainty which make it a good approach to manage safety

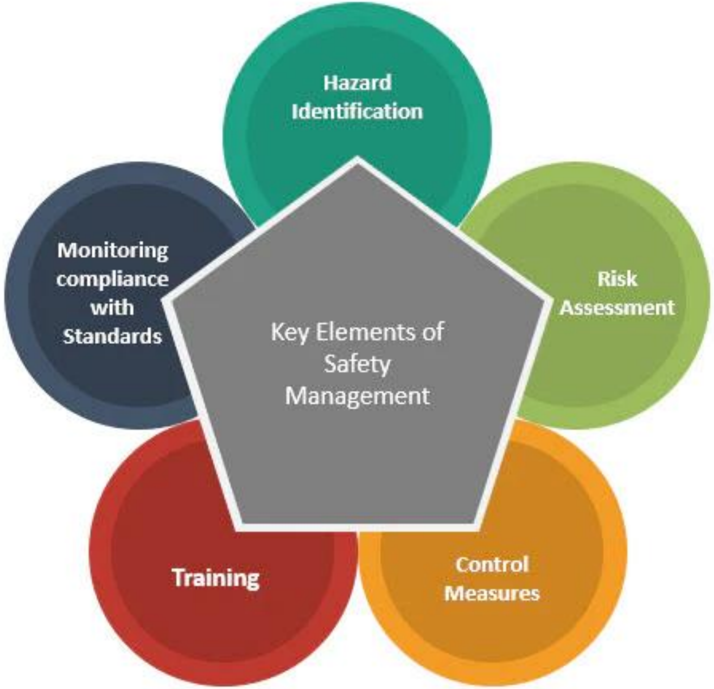


*Developed by Ralph Douglas Stacey in the 1990s (Stacey, 1996). It classifies projects based on two dimensions: the certainty of the requirements and the agreement on the way of working such as technology.

Safety Engineering

P4 P8

Safety in the context of cyber-physical systems (CPS) refers to the state in which the system operates without causing unacceptable risk of physical harm or damage.


Key Elements of Safety Management
Hazard Identification
Risk Assessment
Control Measures
Training
Monitoring compliance with Standards

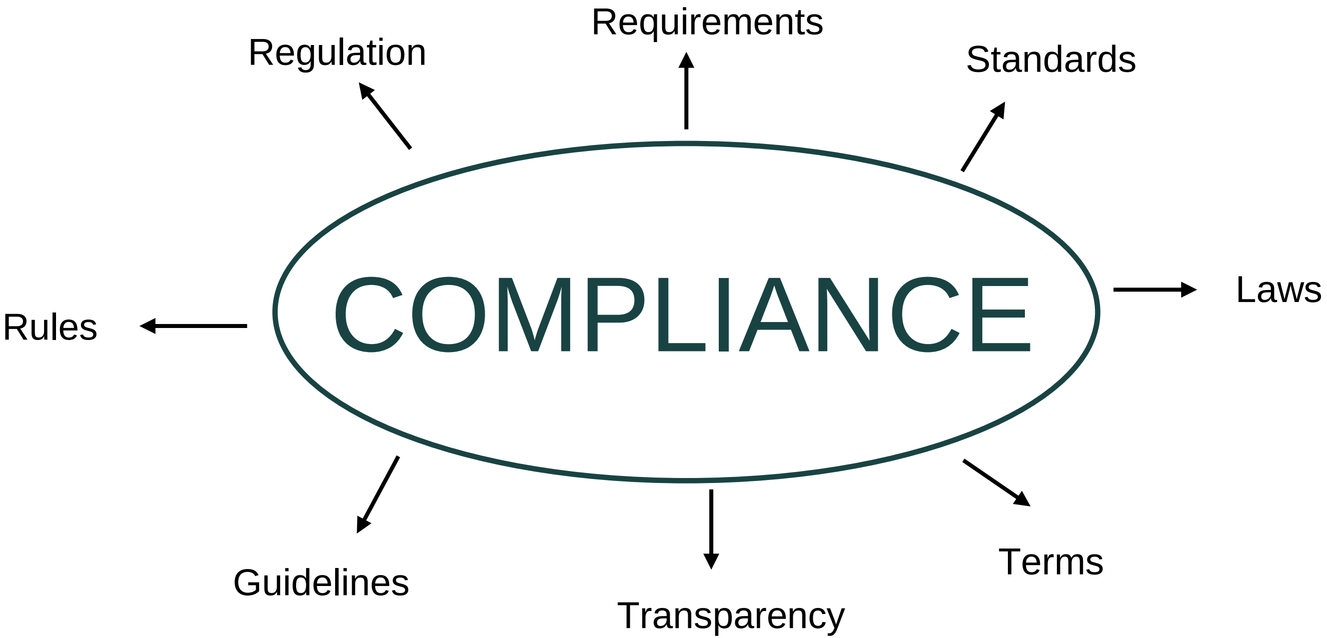
Regulatory Compliance

Carnegie Mellon University Software Engineering Institute

P4 P8

Refers to an organization's adherence to laws, regulations, guidelines, and specifications

ISO26262



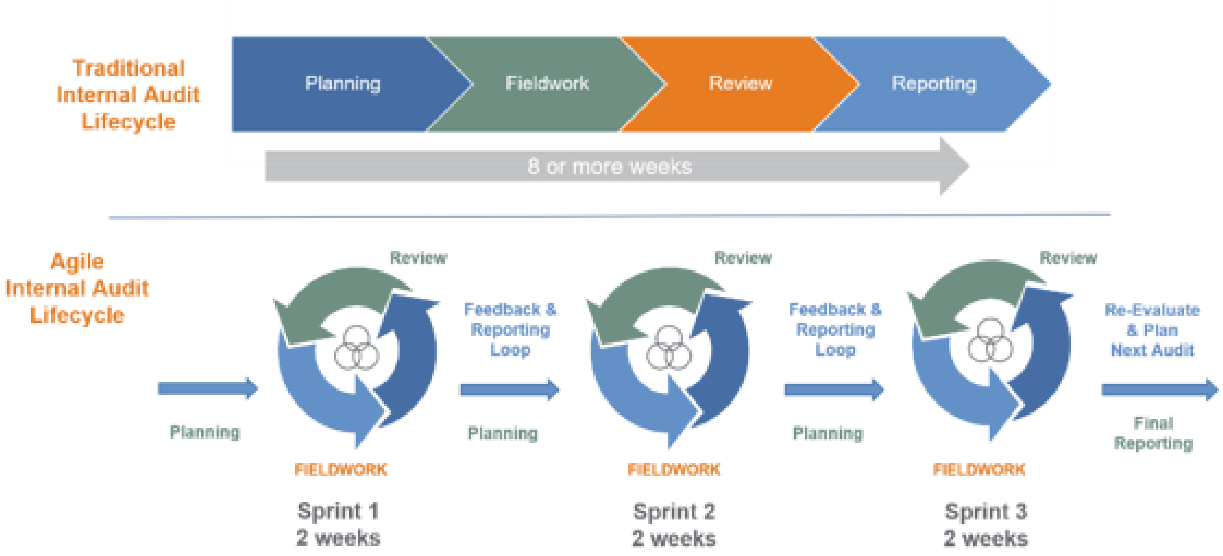
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Monitoring with audits and reviews

Carnegie Mellon University Software Engineering Institute

P4 P8

For best results partner with your auditor and iteratively and incrementally move through audit cycle.

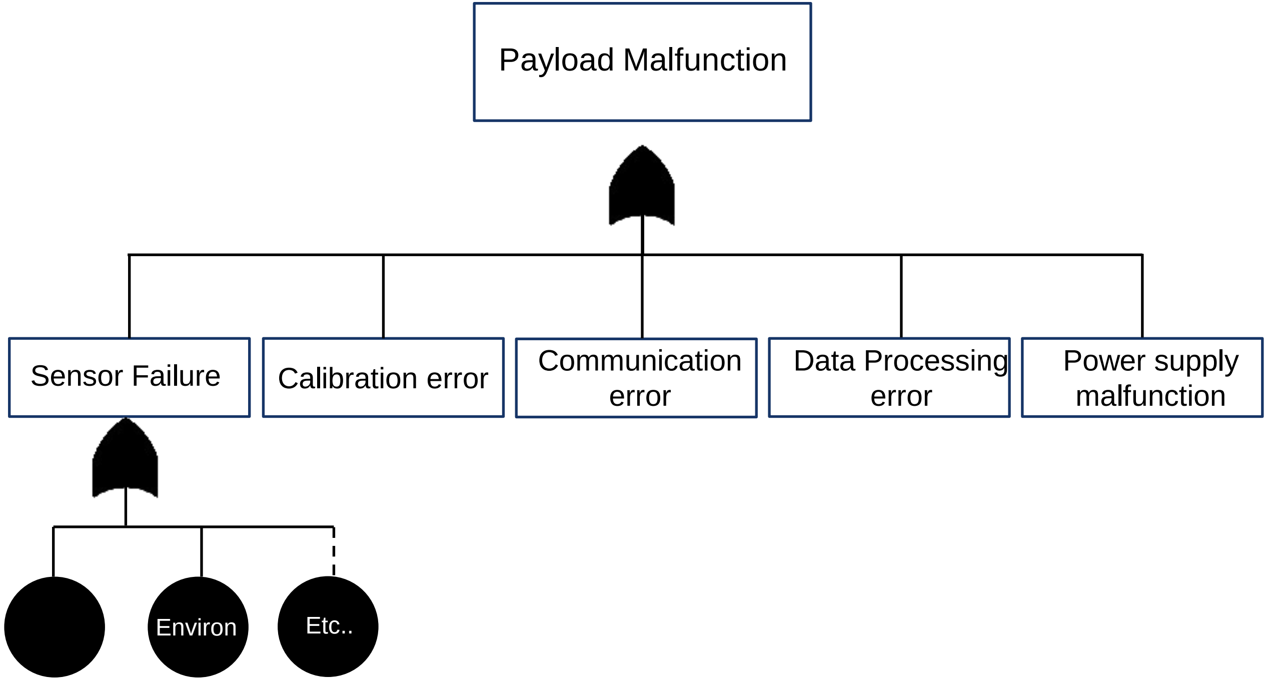


Fault Tree Analysis

P4

P8

Integrate Fault tree analysis into your Agile Ceremonies

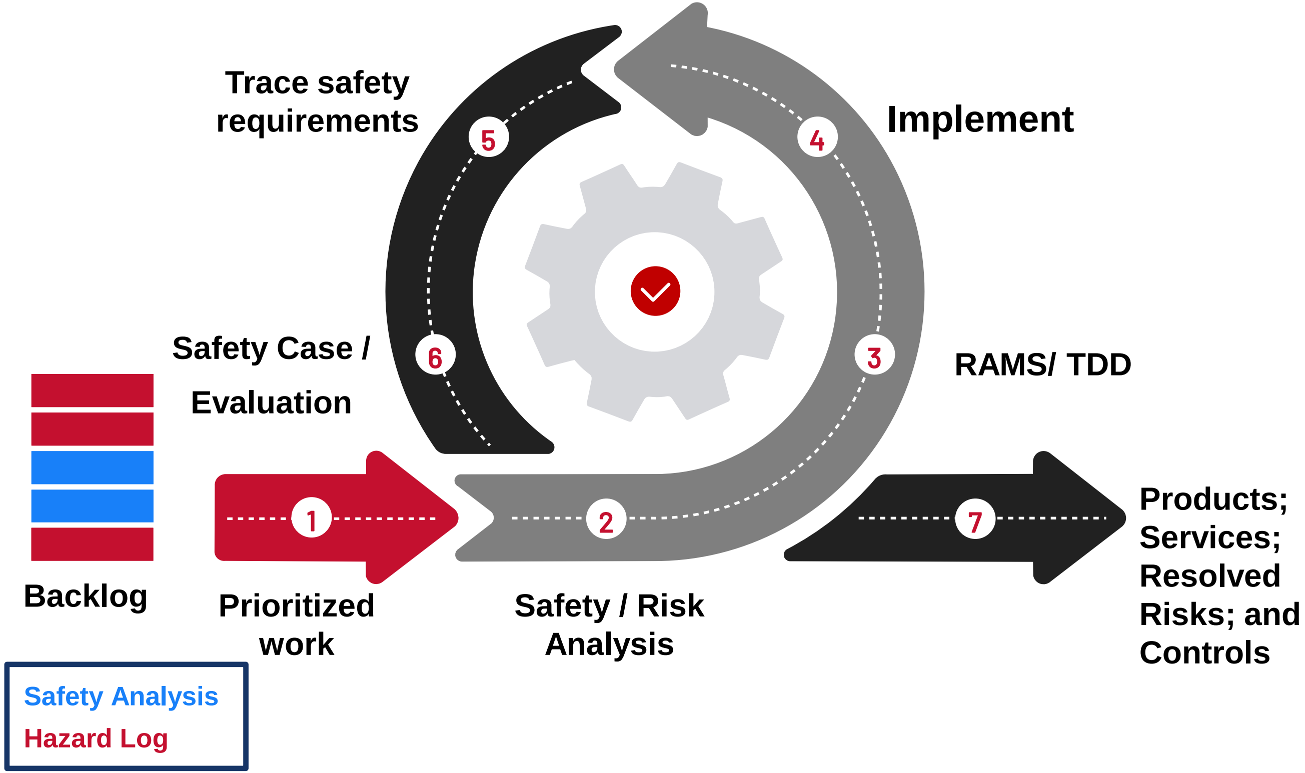


Integrate Safety into each sprint

Carnegie Mellon University Software Engineering Institute

P4 P8

1. Include safety in backlog
2. Emphasize CM
3. Include assessor
4. Perform RAMs
5. Maintain Safety case
6. Safety validation often
7. Automate where possible

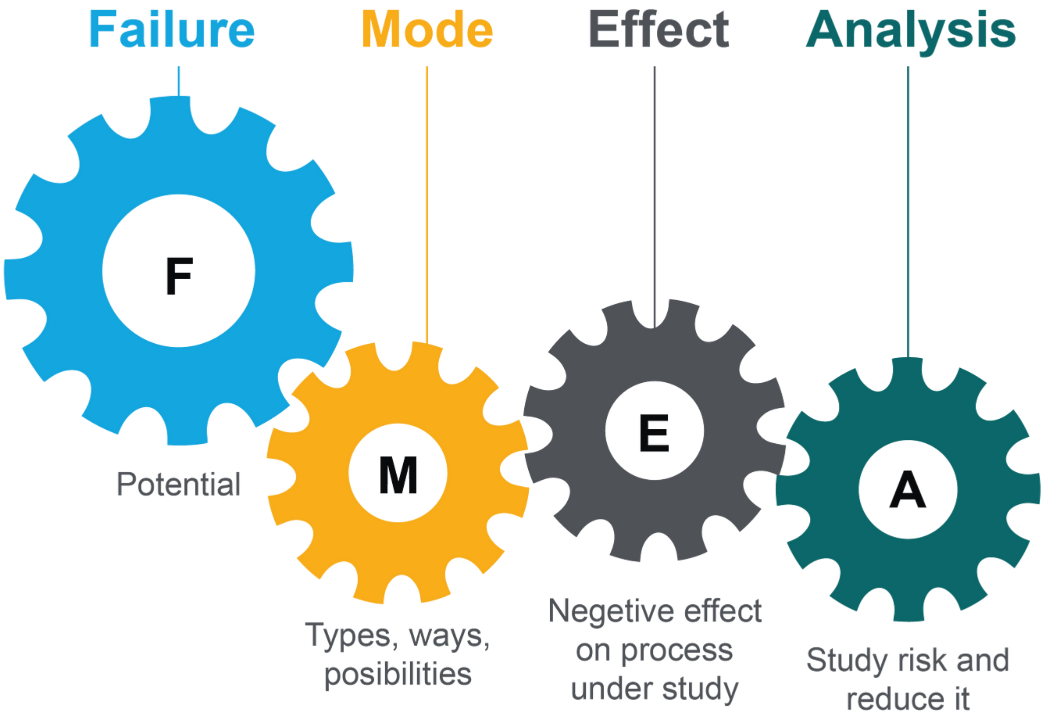


[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

FMEA

P4 P8

The inputs of FMEA can are implemented as enabler user stories; The output of FMEA can be added to your product backlog as risks or as controls



Failure — F — Potential

Mode — M — Types, ways, posibilities

Effect — E — Negetive effect on process under study

Analysis — A — Study risk and reduce it

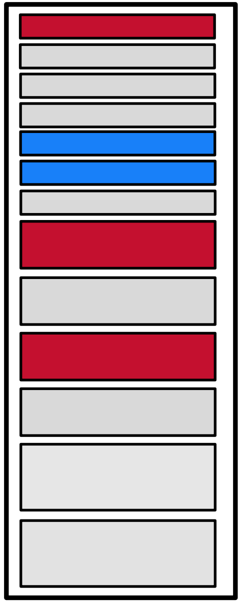
Risk Adjust Product Backlog

Carnegie Mellon University Software Engineering Institute

P4 P8

In Safety engineering risks involve hazards but not all hazard are classified as risks.

Product Backlog

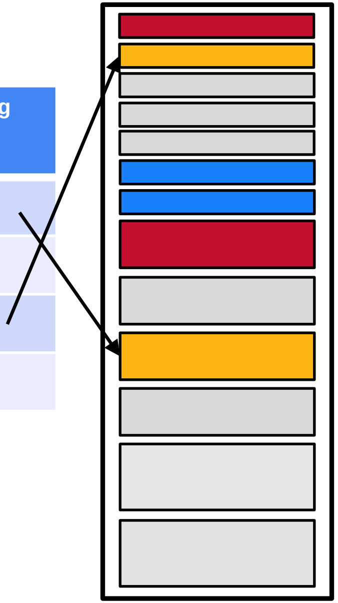


Determine if the mitigations Should be added to backlog.

Risks / Opportunities

Risk	Likelihood	Impact	Mitigation	Mitigation Cost	Mitigation Size	Backlog
Risk 1	**10%**	**$250,000**	**M1**	**$10,000**	**Feature**	**Yes**
Risk 2	**50%**	**$10,000**	**M2**	**$10,000**	**Feature**	**No**
Risk 3	**25%**	**$100,000**	**M3**	**$2500**	**User Story**	**Yes**
Risk 4	**50%**	**$5,000**	**M4**	**$3000**	**User Story**	**No**

Risk Adjusted

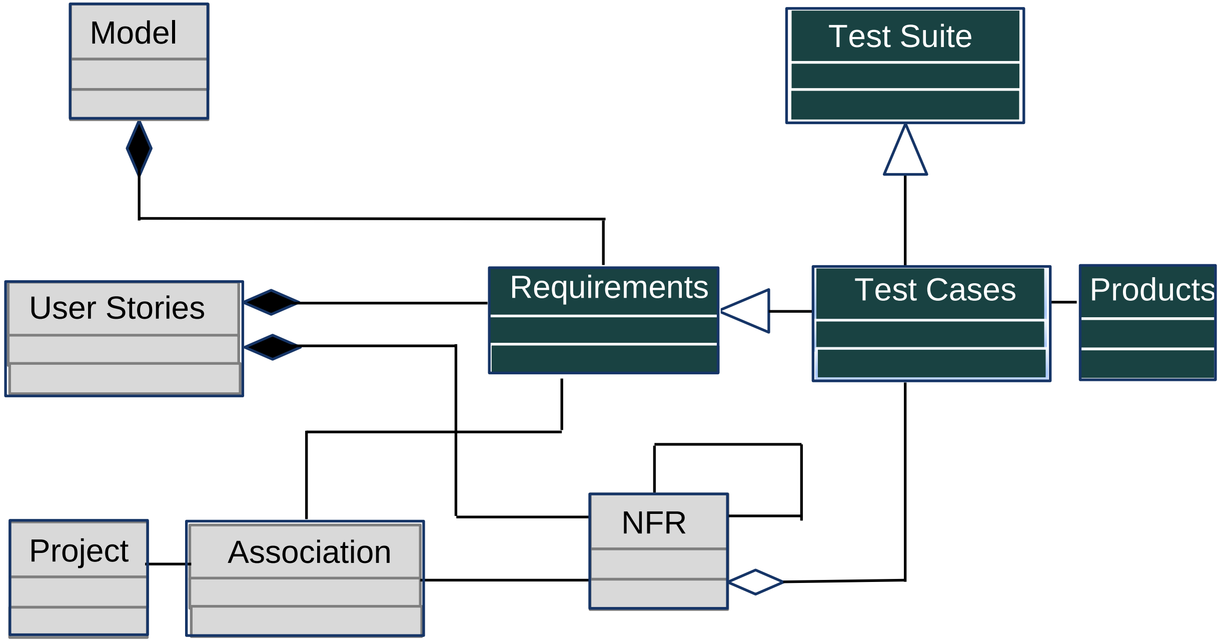


Prioritize Mitigation against other items in product backlog

Documentation and Traceability

P4 P8

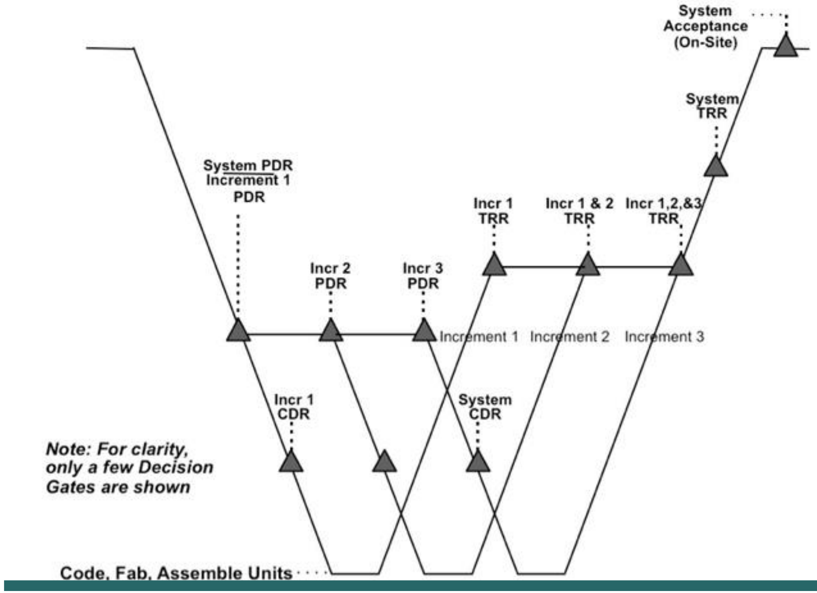
Traceability is required for Safety-critical programs and right-sized documentation



Incremental Design Reviews

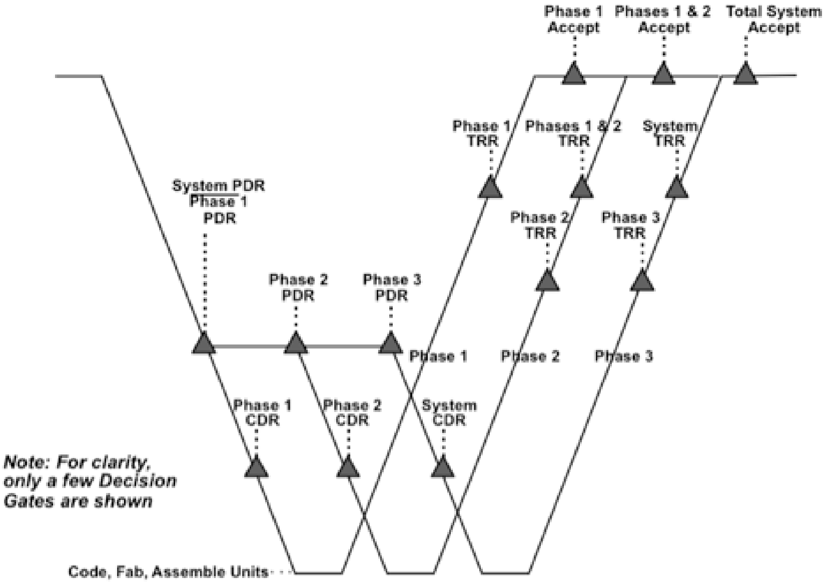
P4 P8

Incremental Development with single Delivery



Note: For clarity, only a few Decision Gates are shown

Incremental Development with multiple Deliveries



Note: For clarity, only a few Decision Gates are shown

SEBoK. (2023, November 20). *System lifecycle process models: Incremental*. Guide to the System Engineering Body of Knowledge. https://sebokwiki.org/wiki/System_Lifecycle_Process_Models:_Incremental

Exercise

15 Minutes

User Story

As a satellite operator, I want the onboard systems to automatically detect any faults in critical subsystems (like power or propulsion) and switch to backup systems without delay to prevent mission failure.

Group Collaboration

As a group develop the implementation plan and tasks to satisfy the fictional user story.

Key Take Aways

- Iterative approach to regulatory compliance
- Integrate Safety into sprints
- Risk Adjusted Backlogs
- Right size documentation and traceability
- Iterative Design Reviews

Leadership in Orchestrating Digital Engineering

Build Security into baseline - DevSecOps

Objectives

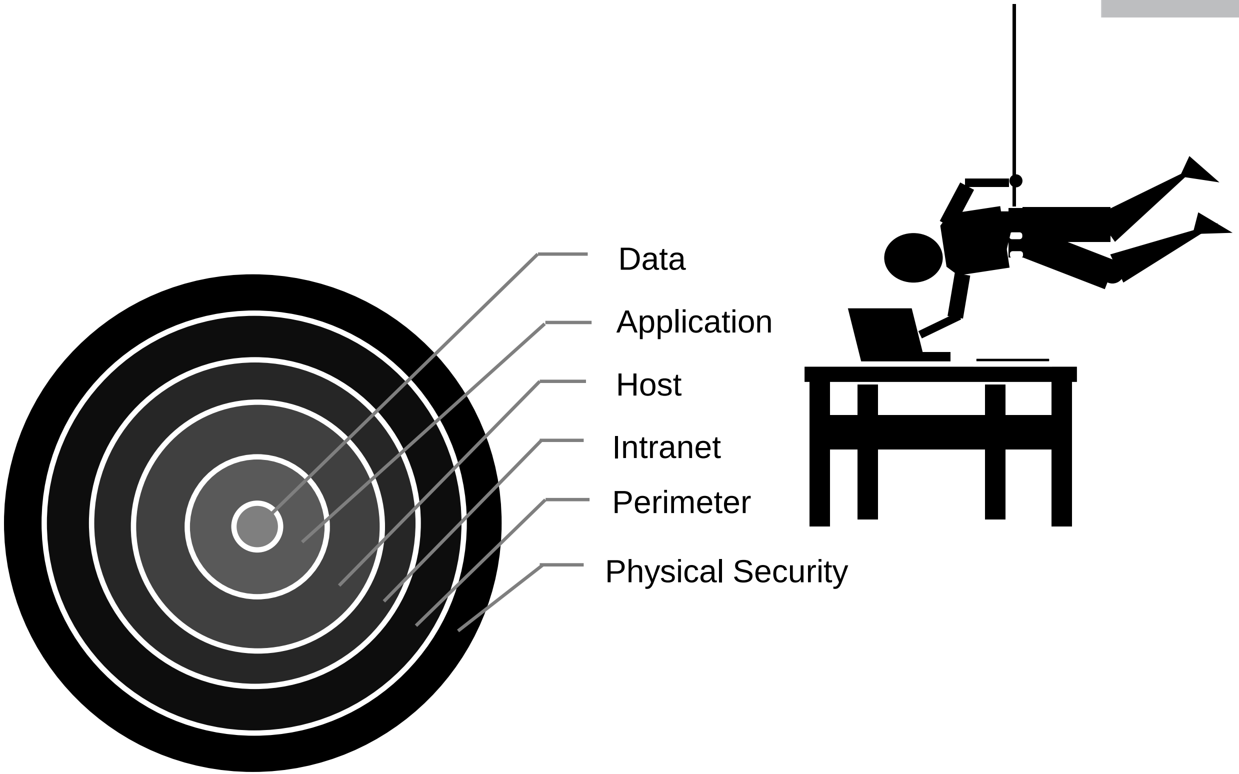
Objectives

- Shift Left Security
- Threat Modeling
- The continuous security highway

Architecting Security

P4 P8

We need to build in defense in depth with a focus on zero trust



[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Securing the Baseline

P4

P8

People, Process, and Culture

- BDD
- Hacker Persona’s
- Threat Modeling
- Game Day’s
- Chaos Engineering
- Site Reliability Engineering
- FMEA (Failure mode and effects analysis)
- Observability & Instrumentation

Engineering Tool Examples

CHAOS MONKEY!

Gremlin

Terraform

cucumber

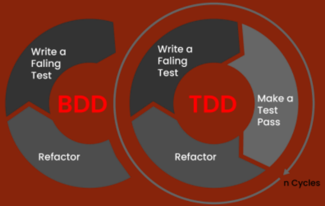
splunk>

new relic.

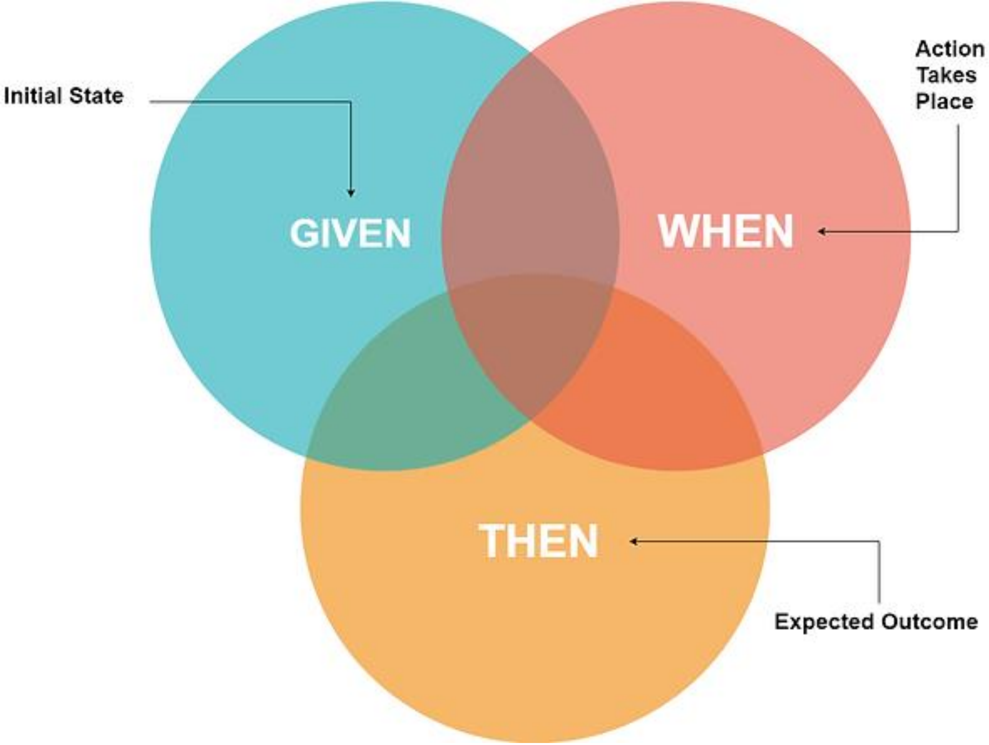
Threat Dragon

Shift Left Security

P4 P8



Begin with test to build executable requirements

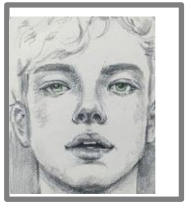


Hacker Persona (s)

P4 P8

Insider Threat

Chuck Careless Developer



Skillset:

Degree in computer science with less than five years experience. Explores the latest technology at home with the ability to code in multiple languages

Identification:

Real Name: Charles Diavol
Alias: Charles 123

Motivations:

- ➢ Wants to maximize delivery of software
- ➢ Wants access to use the latest tech and libraries
- ➢ Reduce workload of perceived overhead work

Frustrations:

- ➢ Governance and compliance that slows him down
- ➢ Ever-growing technical debt
- ➢ Legacy technology

Advanced Persistent Threat

Annie APT

Skillset:

Highly trained and skilled in cyber attacks of all kinds. Effective social engineer. Skilled at evading detection.

Identification:

Real Name: Annie Alvarez
Handles: Triple Pez, 3Pez, Pez

Motivations:

- ➢ Use highly effective attacks, including social engineering
- ➢ Gain Trust, Develop relationships through social media
- ➢ After compromise, remain undetected to meet objectives

Frustrations:

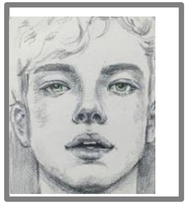
- ➢ When I exploit a target without enough privilege to move forward with my objectives
- ➢ Security controls that block outbound communication

Chuck’s User Stories

P4

P8

Chuck Careless Developer



Skillset:

Degree in computer science with less than five years experience. Explores the latest technology at home with the ability to code in multiple languages

Identification:

Real Name: Charles Diavol
Alias: Charles 123

Motivations:

- ➢ Wants to maximize delivery of software
- ➢ Wants access to use the latest tech and libraries
- ➢ Reduce workload of perceived overhead work

Frustrations:

- ➢ Governance and compliance that slows him down
- ➢ Ever-growing technical debt
- ➢ Legacy technology

User Story

As a Developer I want check-in features quickly so that I can go move on to something else.

User Story

As a Developer I want avoid administrative work so that I can code which is more fun!

Annie’s User Stories

P4

P8

Annie APT

Skillset:

Highly trained and skilled in cyber attacks of all kinds. Effective social engineer. Skilled at evading detection.

Identification:

Real Name: Annie Alvarez
Handles: Triple Pez, 3Pez, Pez

Motivations:

- Use highly effective attacks, including social engineering
- Gain Trust, Develop relationships through social media
- After compromise, remain undetected to meet objectives

Frustrations:

- When I exploit a target without enough privilege to move forward with my objectives
- Security controls that block outbound communication

User Story

As a Annie APT I want to eavesdrop on company X and obtain sensitive information that can be sold.

User Story

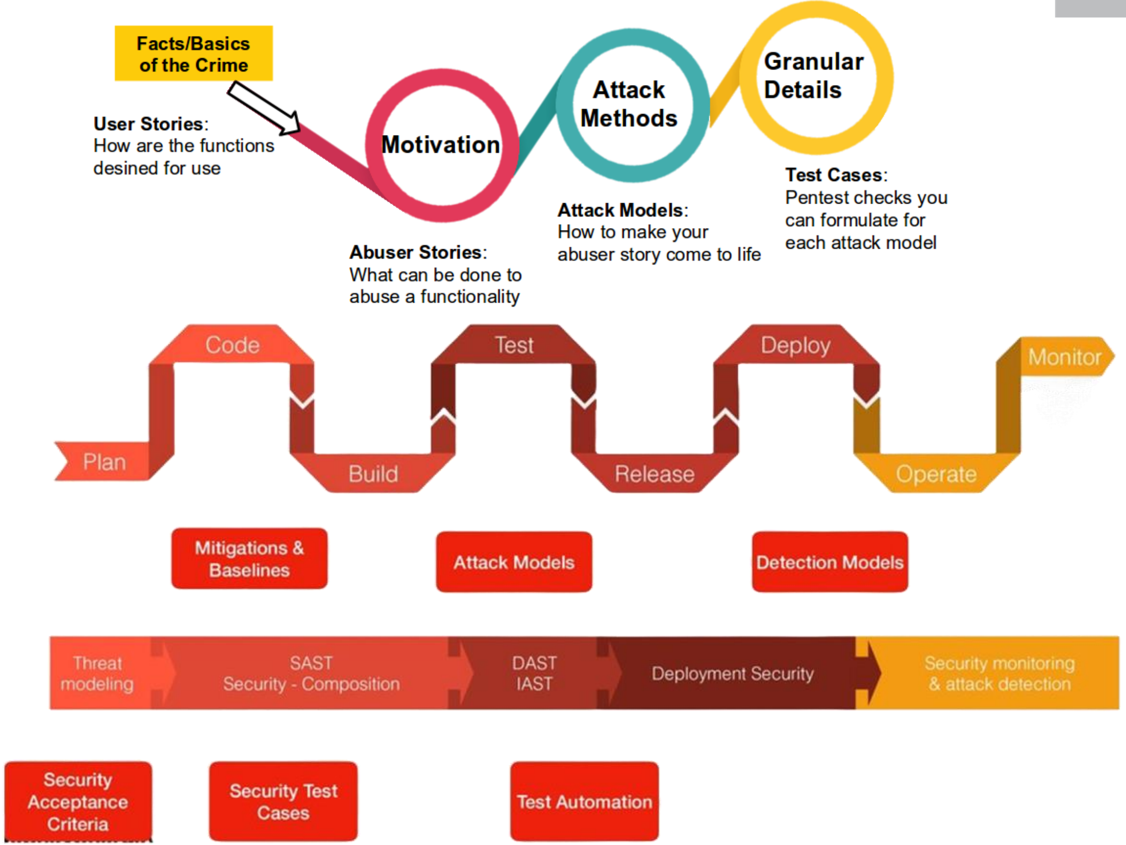
As a Annie APT I want to upload ransomware so that I can extort victims to further my political agenda

P4 P8

Leverage an Agile approach to threat modeling to increase safety

- Identify Assets
- Define the Attack Surface
- Decompose the System
- Identify Attack Vectors
- List Threat Actors
- Analysis & Assessment
- Triage
- Controls

Threat Modeling



Facts/Basics of the Crime

User Stories: How are the functions desined for use

Motivation

Abuser Stories: What can be done to abuse a functionality

Attack Methods

Attack Models: How to make your abuser story come to life

Granular Details

Test Cases: Pentest checks you can formulate for each attack model

Mitigations & Baselines

Attack Models

Detection Models

Threat modeling → SAST Security - Composition → DAST IAST → Deployment Security → Security monitoring & attack detection

Security Acceptance Criteria

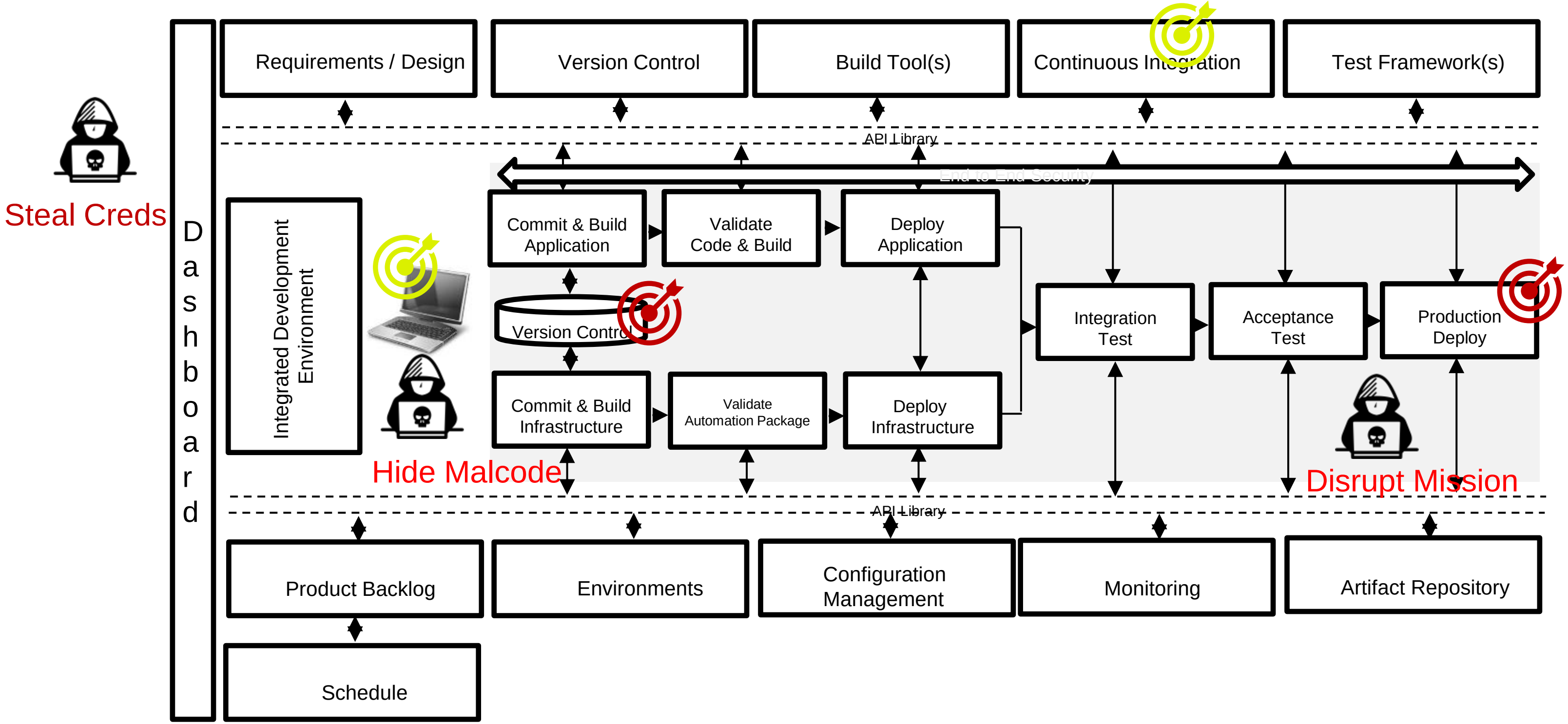
Security Test Cases

Test Automation

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Example Threat model for a pipeline

P4 P8

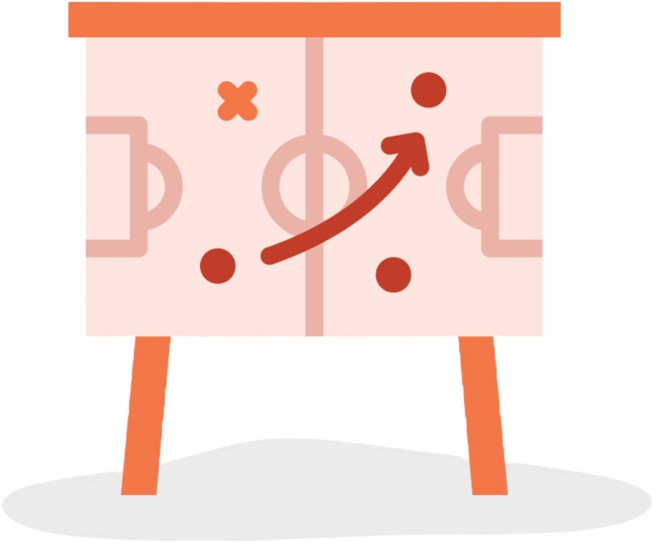


Game Day

P4 P8

1. ***Select Game** (Active Failure or Team Table-top)*
2. ***Identify Team***
3. ***Create Key Roles** (game runner)*
4. ***Identify System***
5. ***Prepare chaos***
6. ***Execute Game Day***
7. ***Retrospective***

A timebox of 2-4 hours set aside for a team to run one or more chaos experiments on a system or service, observe the impact, then discuss the technical outcomes.

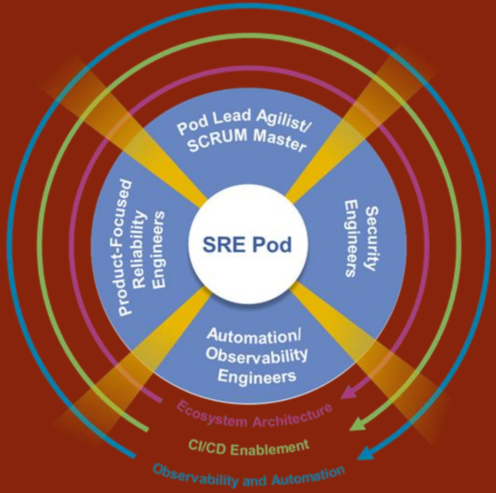


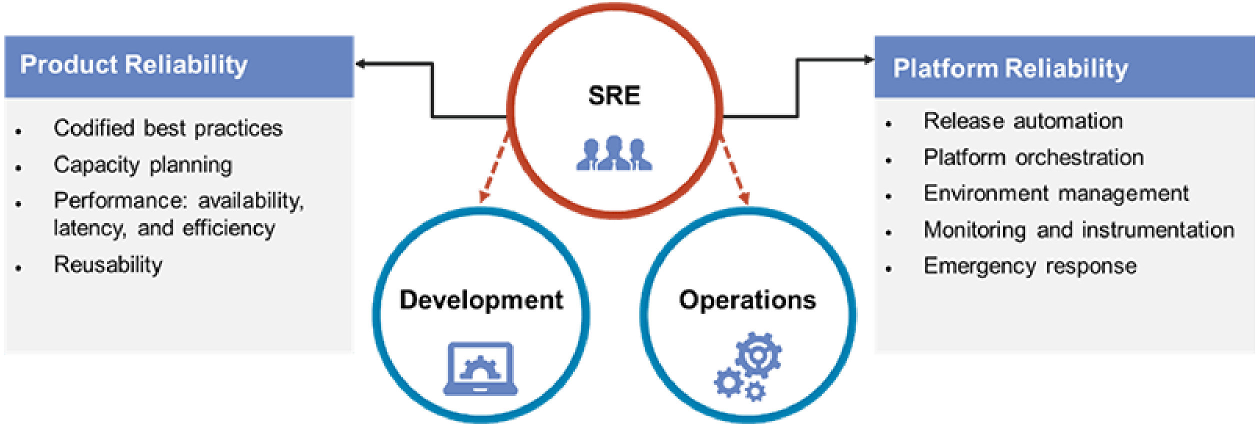
Site Reliability Engineering

P4 P8

SRE Pods

1. *Autonomous Team*
2. *Cross-functional*
3. *Aligned around value*
4. *3-9 People*
5. *Uses Agile Practices*
6. *Evolves over time*


Pod Lead Agilist/ SCRUM Master
Product-Focused Reliability Engineers
SRE Pod
Security Engineers
Automation/ Observability Engineers
Ecosystem Architecture
CI/CD Enablement
Observability and Automation



Product Reliability

- Codified best practices
- Capacity planning
- Performance: availability, latency, and efficiency
- Reusability

Platform Reliability

- Release automation
- Platform orchestration
- Environment management
- Monitoring and instrumentation
- Emergency response

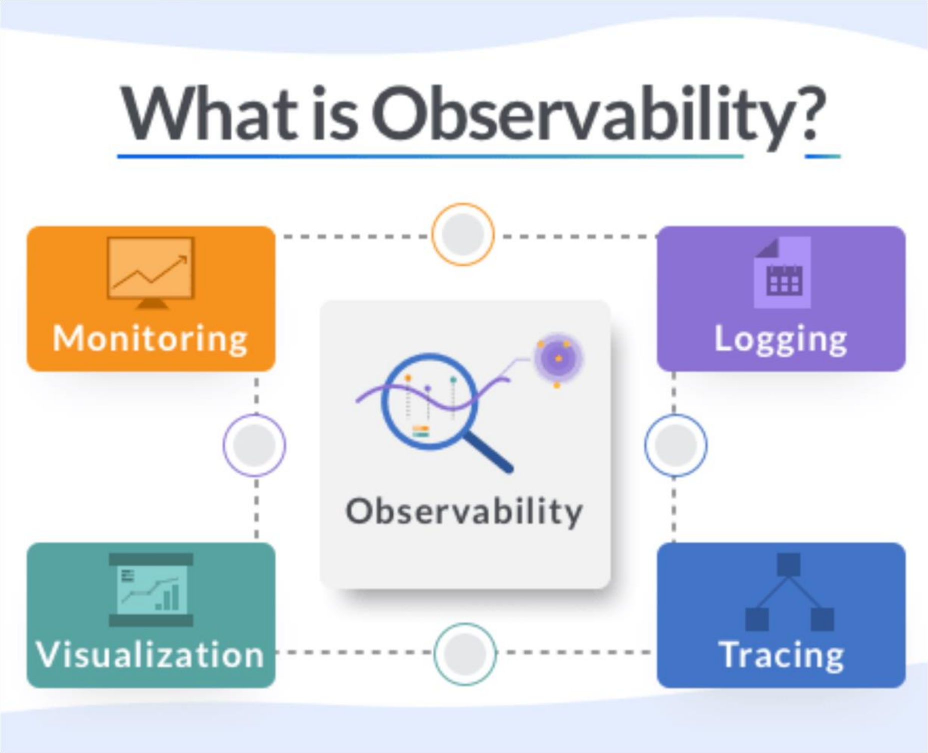
https://us.nttdata.com/en/blog/2022/june/site-reliability-engineering-and-agile-pods

Observability

P4 P8

Instrument the system for Observability

Intentional development required to obtain observability



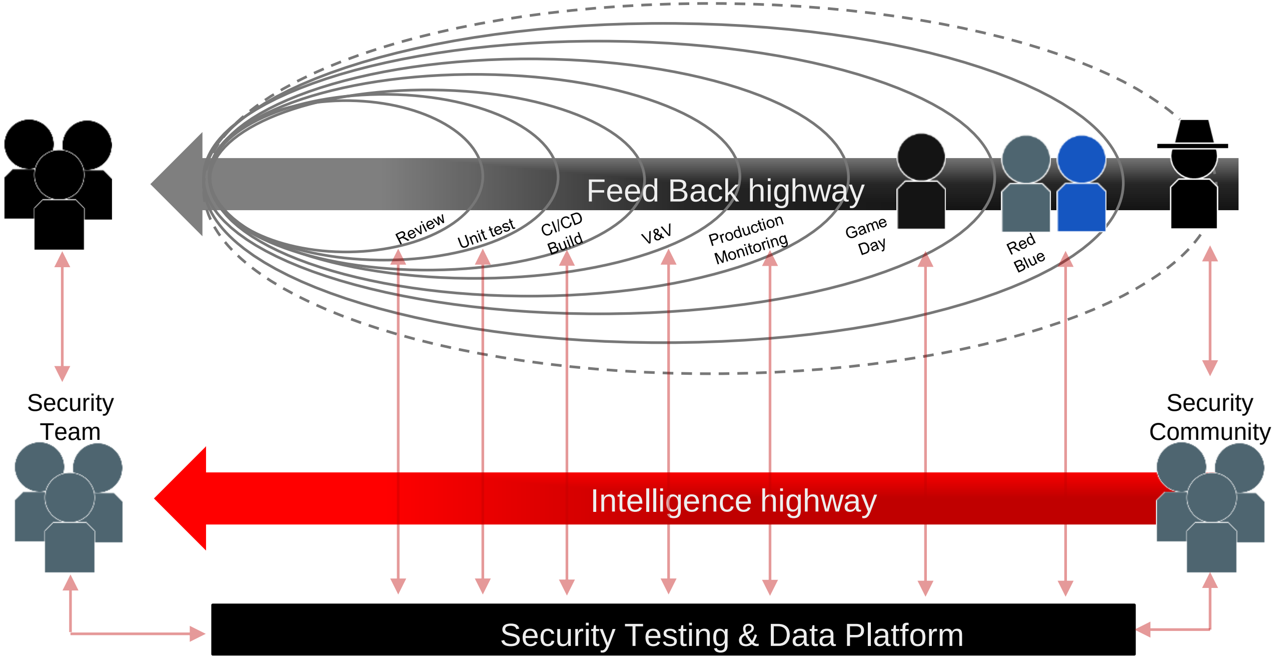
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

The Continuous Security Highway

P4 P8

Carnegie Mellon University Software Engineering Institute

DevSecOps: Seamlessly integrate security into the implementation pipeline; ensuring everyone takes responsibility while continuing to shorten feedback loops



Example Threats

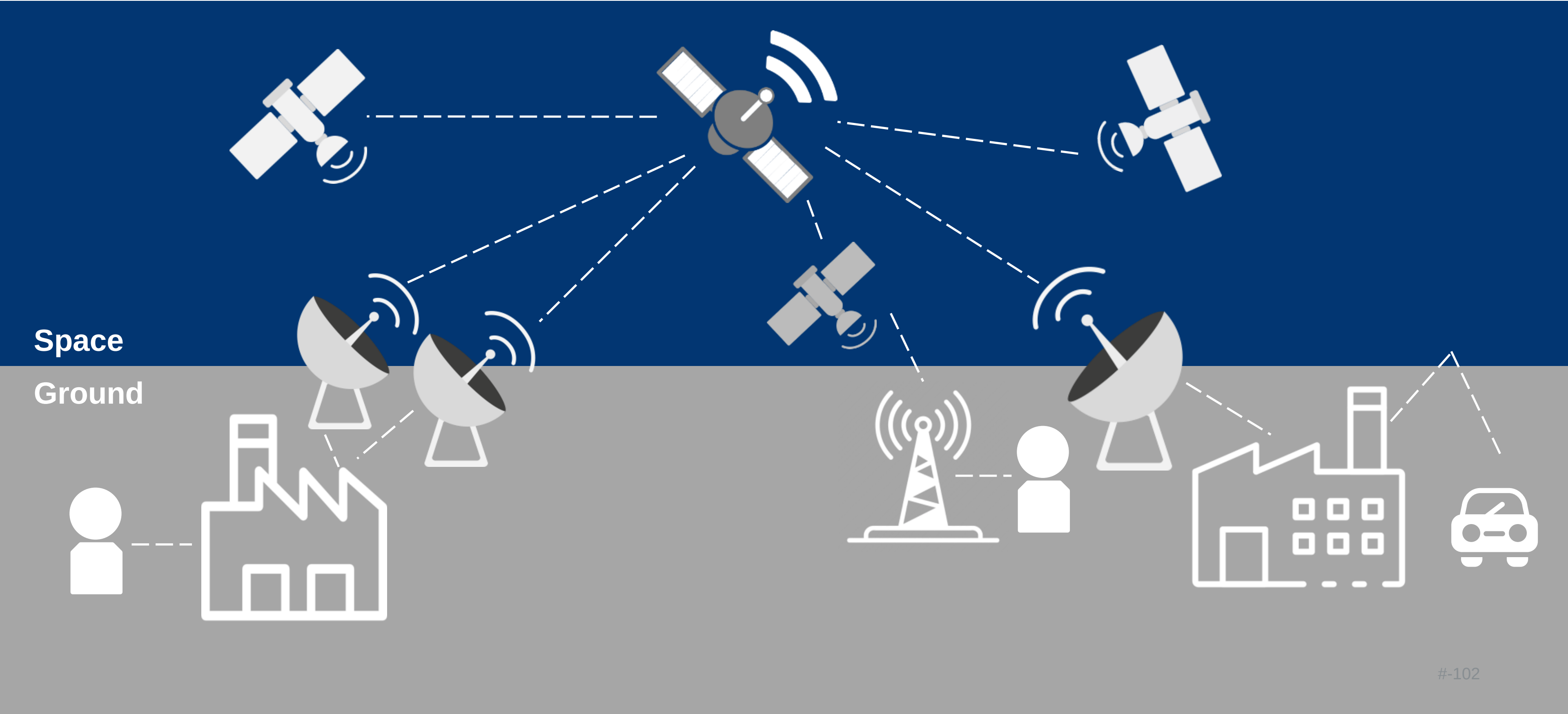
P4 P8

Link Segment	Space Segment	Ground Segment	User Segment
C2 Intrusion	GPS Interference	Attacks to IC / OT Systems	Loss of Network Connect
Malware/ransomware	Spoofing Jamming	Supply Chain	Compromised transactions
Denial of Service	Space Debris	Malware / Ransomware	Supply Chain disturbance
Remote Code Execution	Space Weather Interference	Remote Code Execution	PNT Interference
Man in the middle	Anomalous Behavior	Terminal Hacking	GPS Interference
Spoofing / Jamming			

These are not exhaustive and all-inclusive the list are some examples

Threat Model Exercise

P4 P8



Exercise

15 Minutes

Group Collaboration

As a group perform an abbreviated threat model on the satellite system.

1. Who are Hacker Personas?
2. What are the threat vectors?
3. What are potential controls?

Key Take Aways

- Shift left for security with hacker persona and tests
- Threat modeling
- Continuous security highway

Leadership in Orchestrating Digital Engineering

Managing lead times through automation –CI/CD

Carnegie Mellon University Software Engineering Institute

Objectives

Objectives

- Role of automation for CPS
- CI/CD Pipelines for CPS
- DevOps for CPS
- Digital Twins

Role of Automation for CPS

P5 P7

1 **Design and Simulation**
- Automated Design Tools
- Modeling and Simulation

2 **Manufacture and Assembly**
- Robotic Assembly
- Additive Manufacturing

3 **Testing and Quality Assurance**
- Automated Test Equipment
- Non-Destructive Testing

4 **Launch Operations**
- Automated countdown and Launch sequencing
- Autonomous Flight Safety

5 **Orbital Operations and Control**
- Autonomous Navigation
- Automated Fault detection response

6 **Maintenance and Servicing**
- Robotic Servicing
- Automated health monitoring

7 **Decommissioning and Deorbiting**
- Automated Deorbiting Systems

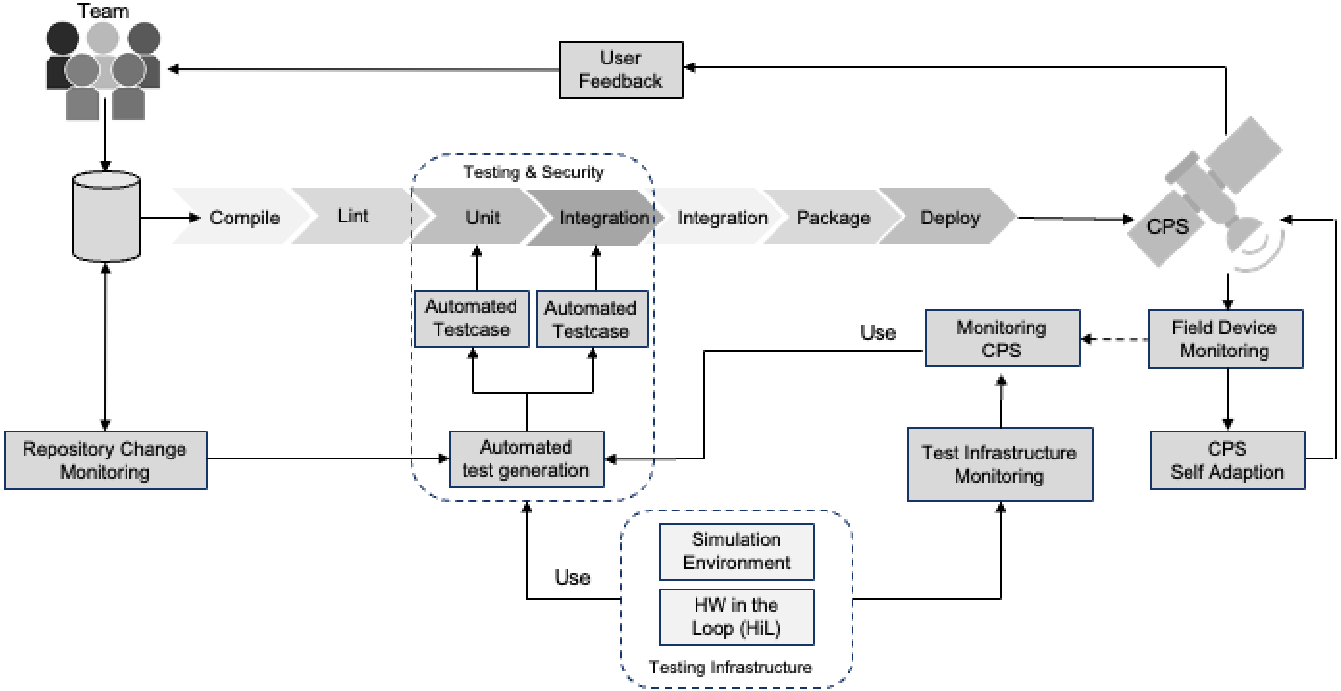
8 **Data Analysis and Mission Planning**
- Automated data processing
- Mission Planning Tools

Extensive automation increases productivity while reducing human error

CI/CD for CPU

P5 P7

Continuous Integration is a process that continually merges a system's artifacts and configuration items from all stakeholders on a team, into a shared mainline to build and test the developed system



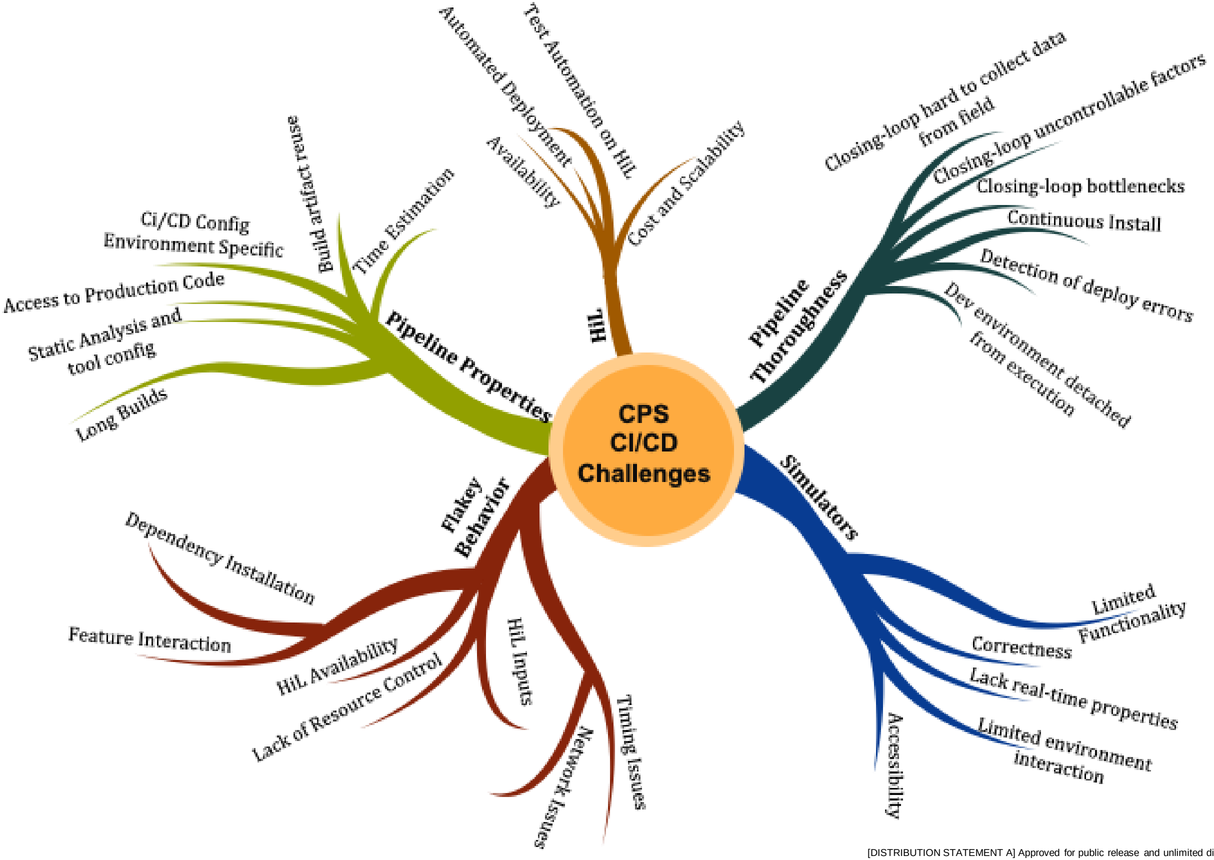
Challenges of implementing CI/CD

Carnegie Mellon University Software Engineering Institute

P5 P7

There are multiple challenges which are grouped below:

1. *Pipeline Properties*
2. *Pipeline Thoroughness*
3. *HiL*
4. *Simulators*
5. *Flakey Behavior*



DevSecOps for CPS

P5

P7

Create

- Document security Considerations
- FMEA
- Design Schematics
- Modeling
- Iterate based on simulation results
- Develop SW and FW for HW interaction

Plan

- Threat Modeling
- Security Requirements
- Safety Risk Assessment
- Feasibility Studies
- Features (SW / HW)

Release / Deploy

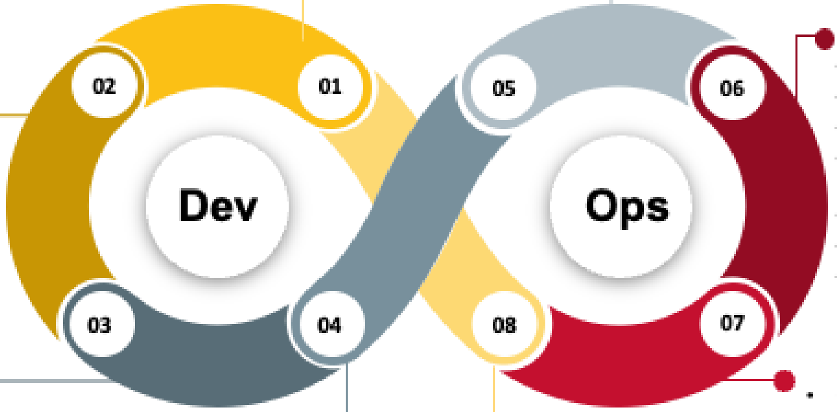
- Deploy Cyber Security Monitoring tools
- Deploy process for tracking safety and security issues
- Full-Scale Production
- Deploy Final Software Versions
- Provide documentation

Configure

- Instantiate automated security checks
- Encrypt sensitive data
- Train operations team
- Provide diagnostic tools to troubleshoot HW
- Provide detailed config guides
- Turn on Feature Toggles
- Set up monitoring Dashboards

Verify

- Automated Test (SAST/DAST)
- Safety Impact Analysis
- Fault Injection / Chaos engineering
- Test Prototypes
- Electrical and functional Circuit board.



Pre-Prod

- Security Reviews / Pen Test
- Pilot Runs to validate safety measures in real-world conditions
- Finalize manufacture / Assembly
- Stage release candidates for SW

Monitor

- Real-time safety / security monitoring
- Use IoT Platforms /predictive maintenance
- Conduct A/B Testing
- Collect user feedback

Operate

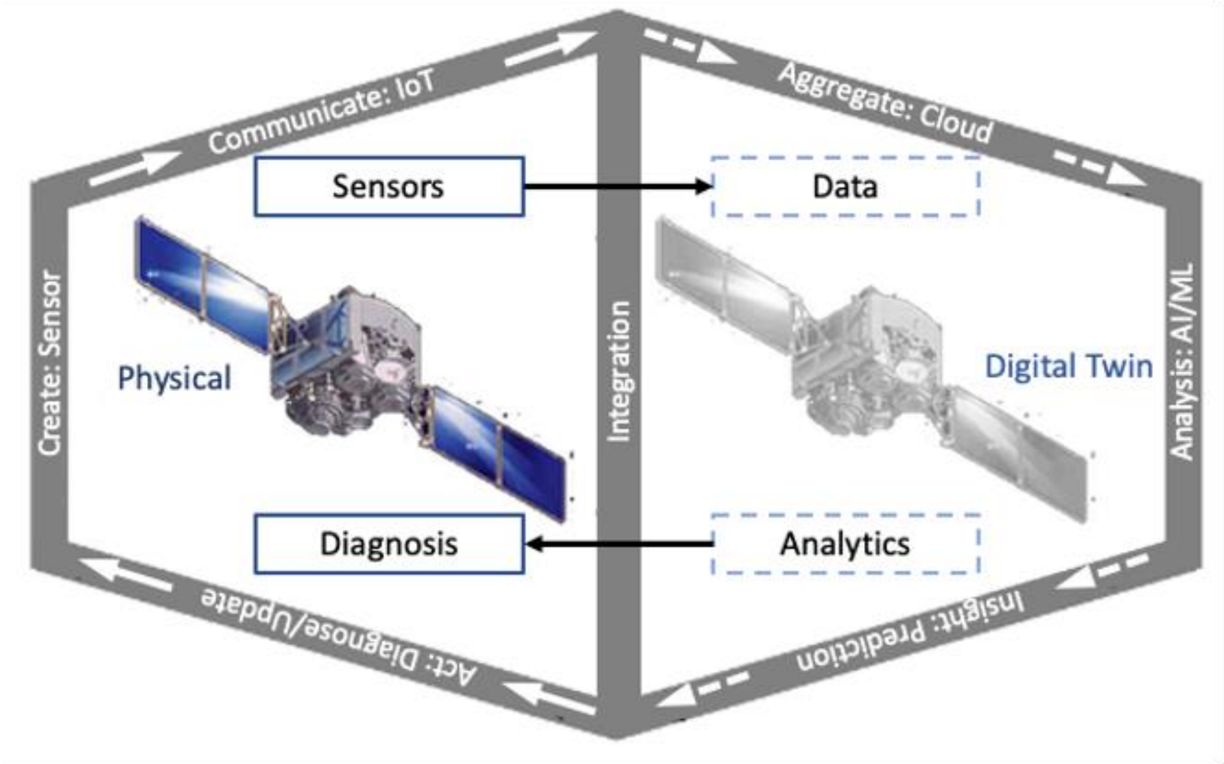
- Conduct security audits
- Monitor system for breaches
- Conduct regular safety reviews
- Implement system to manage spares
- Manage Field operations
- Roll out Patches / Updates

Digital Twin

P5

P7

Validate the product before it exists in the real world and Iterate regularly

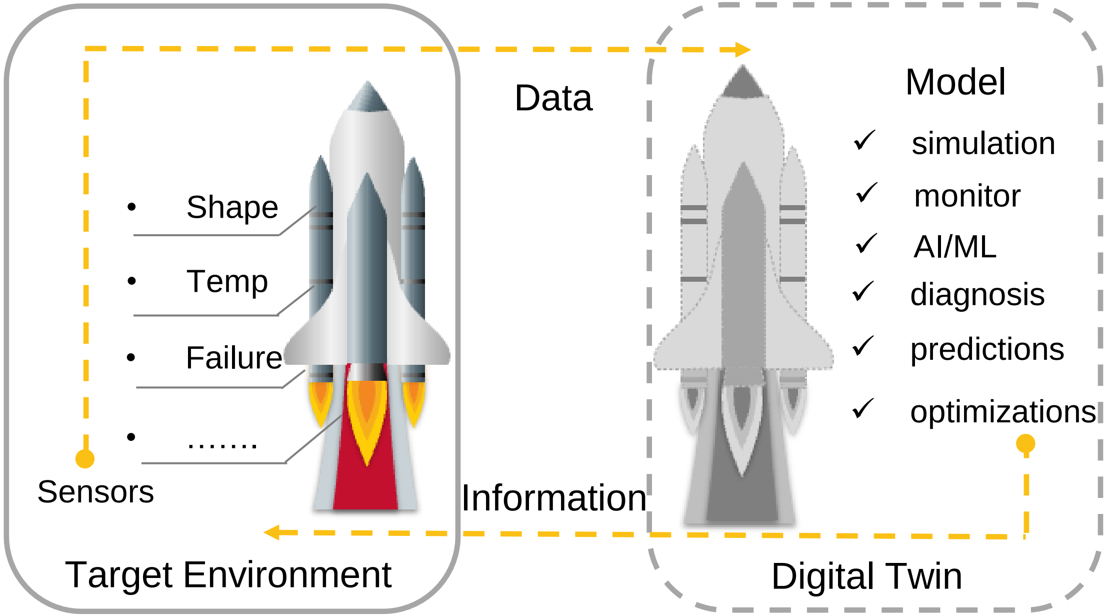


[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Benefits of Digital Twin

P5 P7

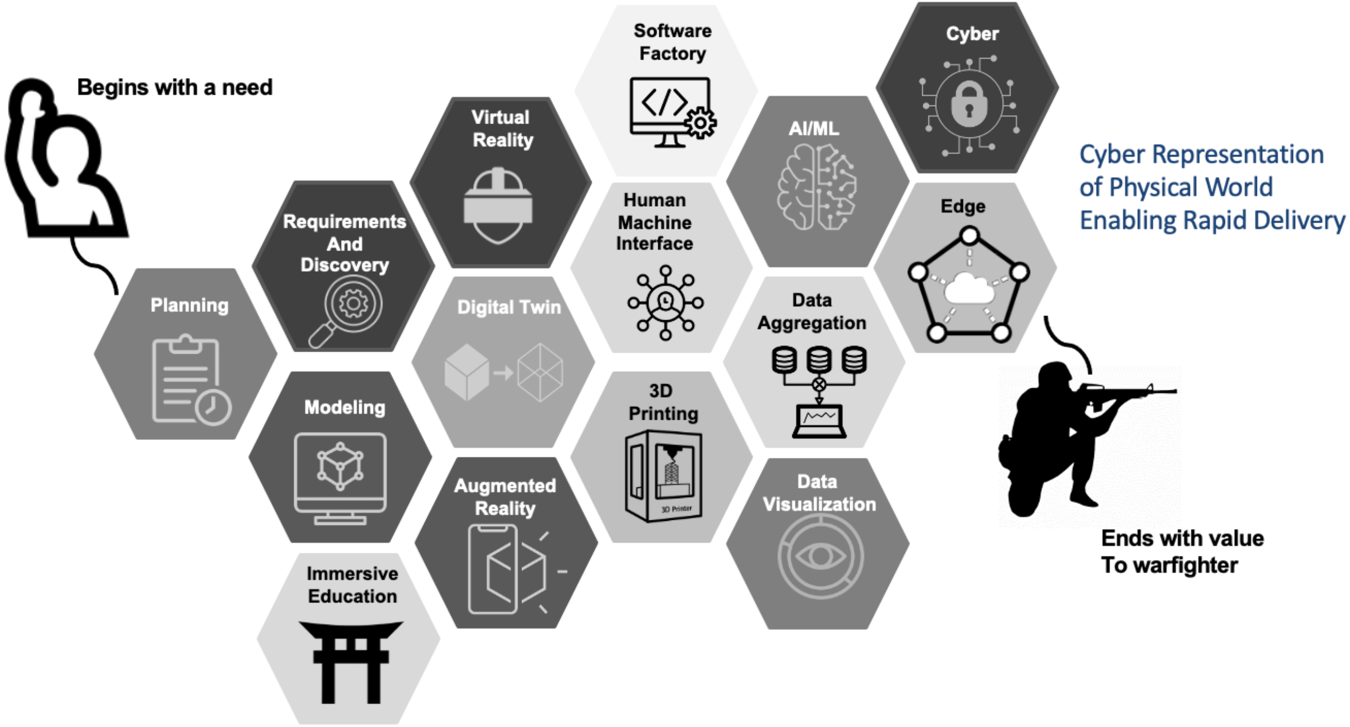
1. ***Design Optimization***
2. ***Future proofing***
3. ***Time / Cost savings***
4. ***Operational Efficiency***
5. ***Predictive Maintenance***



Digital Engineering Value Stream

P5 P7

Use all the tools in your toolbox.



Cyber Representation of Physical World Enabling Rapid Delivery

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Key Take Aways

- Role of automation for CPS
- CI/CD Pipelines for CPS
- DevOps for CPS
- Digital Twins

Leadership in Orchestrating Digital Engineering

Exercise– Challenges for large-scale, safety-critical, cyber-physical systems

Carnegie Mellon University Software Engineering Institute

Objectives

Objectives

- Understand the current challenges for CPS
- Knowledge of complexity in CPS
- The barrier with the greatest difficulty

Exercise

5 Minutes

Group Collaboration

As a group identify **5** potential **challenges** you see with implementing Industrial DevOps in you organization be prepared to share

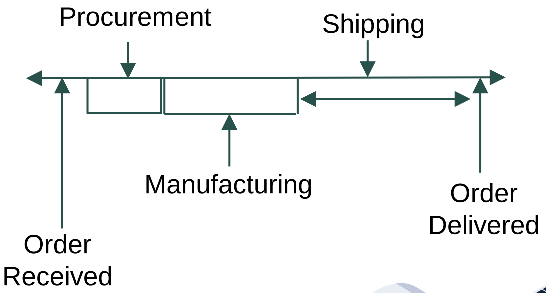
Map our challenges to Industrial DevOps Principles to see what Module we address.

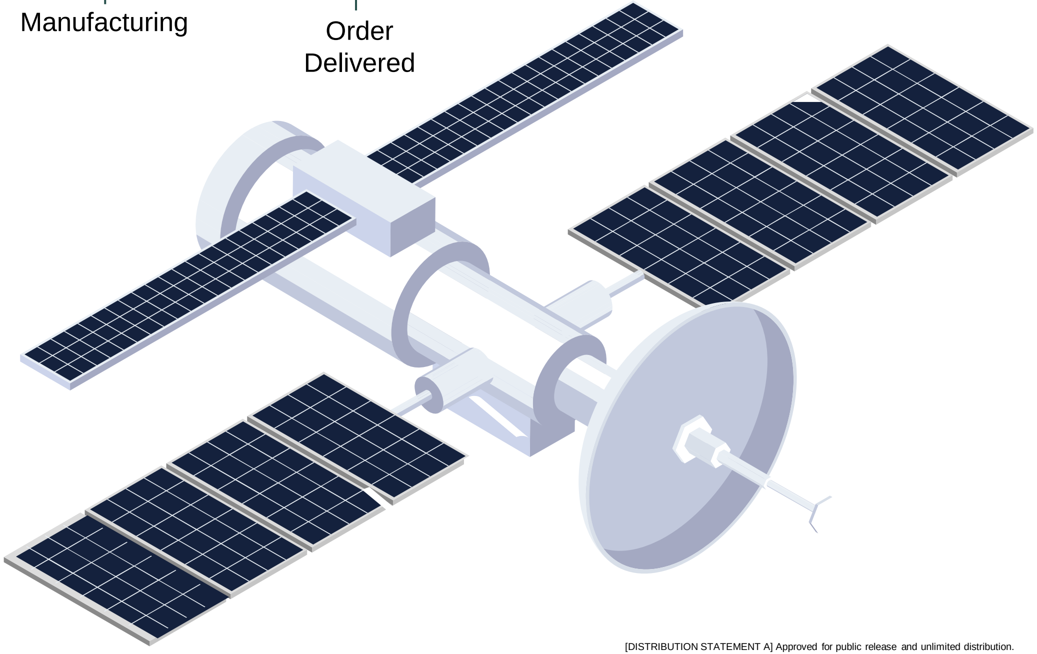
M = Module that addresses Challenge

Challenges	P1 - Organize for the flow of value	P2- Apply Multiple Horizons of planning	P3 - Implement Data Driven Decisions	P4 - Architect for Change and Speed	P5 - Iterate, Manage Queues, Create Flow	P6 - Establish Cadence and Synchronization	P7 - Integrate Early and Often	P8 - Shift Left	P9 - Apply a growth Mindset
Long delivery times / limited prototype iterations					M6		M6		
Unique integration and test requirements					M6		M6		
Technical complexity and Interdependencies			M3	M3					
Complex risk management			M3	M3					
Large security attack surface				M5				M5	
High stakes safety and reliability				M4				M4	
Regulatory and compliance hurdles				M4				M4	
Supply chain and vendor management	M7	M7				M7			
Collaboration and coordination at scale	M7	M7				M7			
Culture									M8

Long Delivery Time

While software can often be incorporated into product daily, typically cyber-physical systems have components and sub-assemblies from suppliers



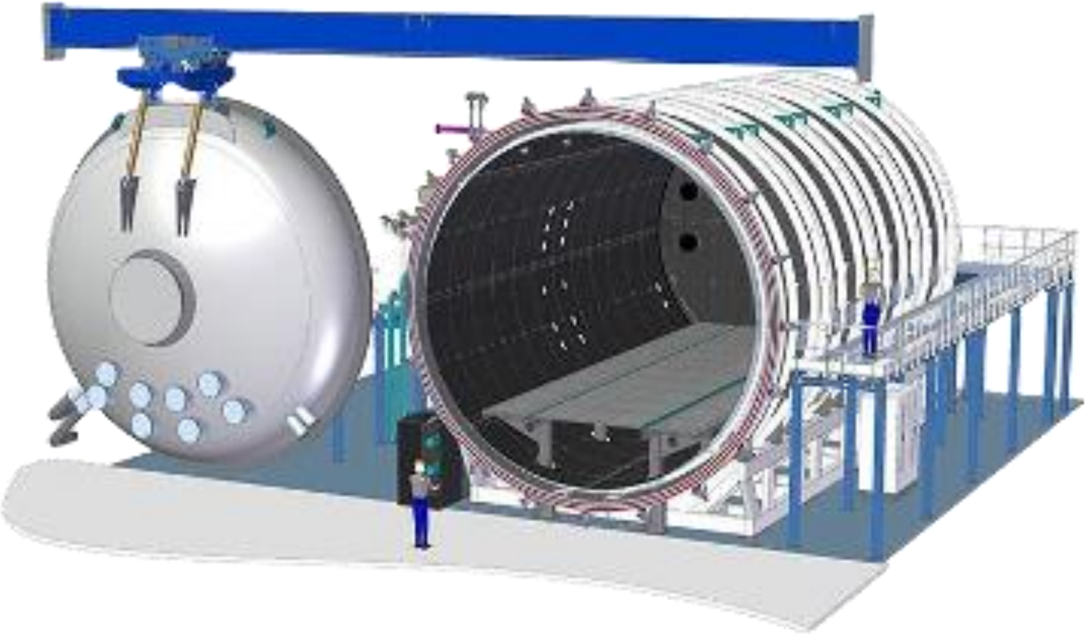


Unique Integration and Test Requirements

Carnegie Mellon University Software Engineering Institute

Cyber-Physical systems can have unique and expensive test equipment

Very large Thermal Vacuum chambers cost millions of dollars and the tests can take months



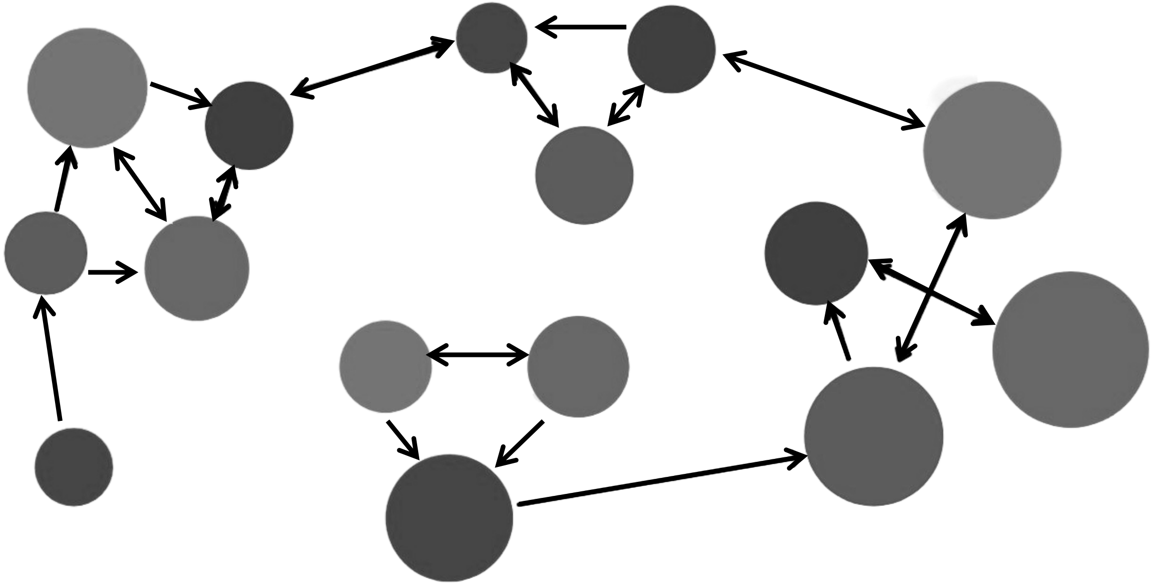
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Technical Complexity and Interdependencies

Carnegie Mellon University Software Engineering Institute

Can you imagine all the Dependencies in GPS Satellite constellation

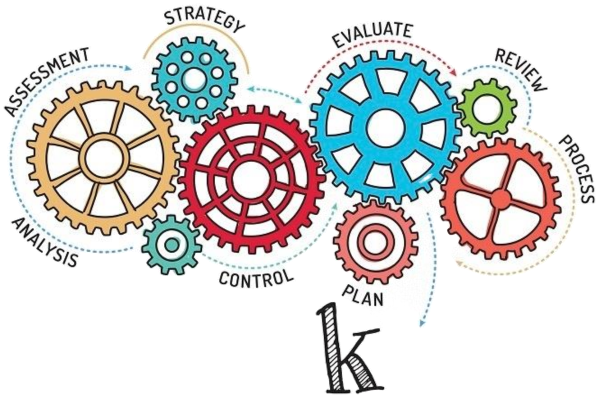
Cyber-Physical systems have many dependencies
and are often systems of systems magnifying those dependencies.



[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Example GPS Risks

- Solar Interference
- Frequency Crowing
- Signal Degradation
- Jamming
- Spoofing
- System under attack (Cyber/physical)

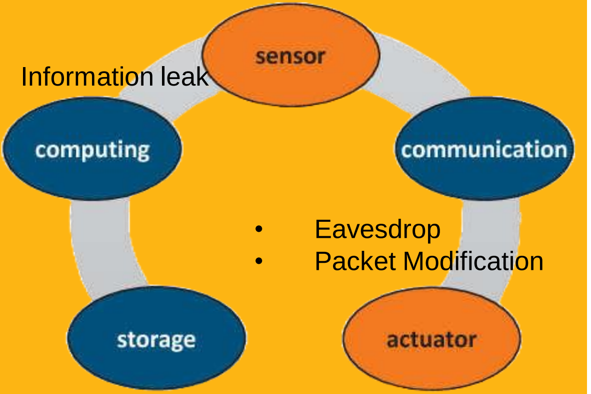


Attack Surface

Cyber-Physical System Attack

- Sensor Spoof
- DoS

Information leak

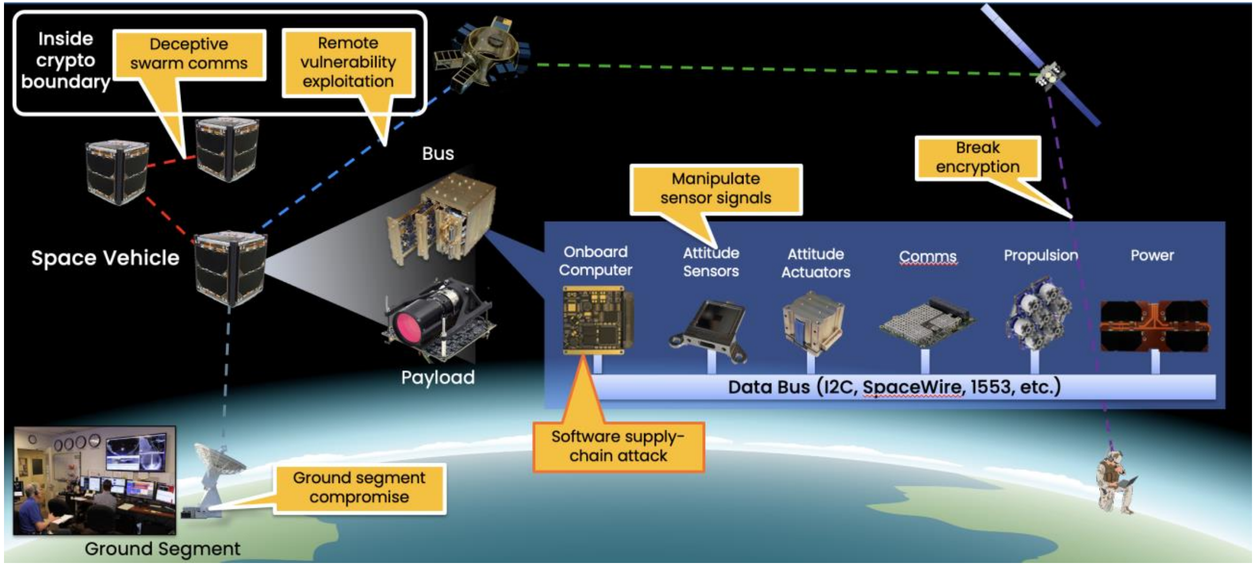


- Eavesdrop
- Packet Modification

Storage Mod

- Authentication Failure
- Deadline miss

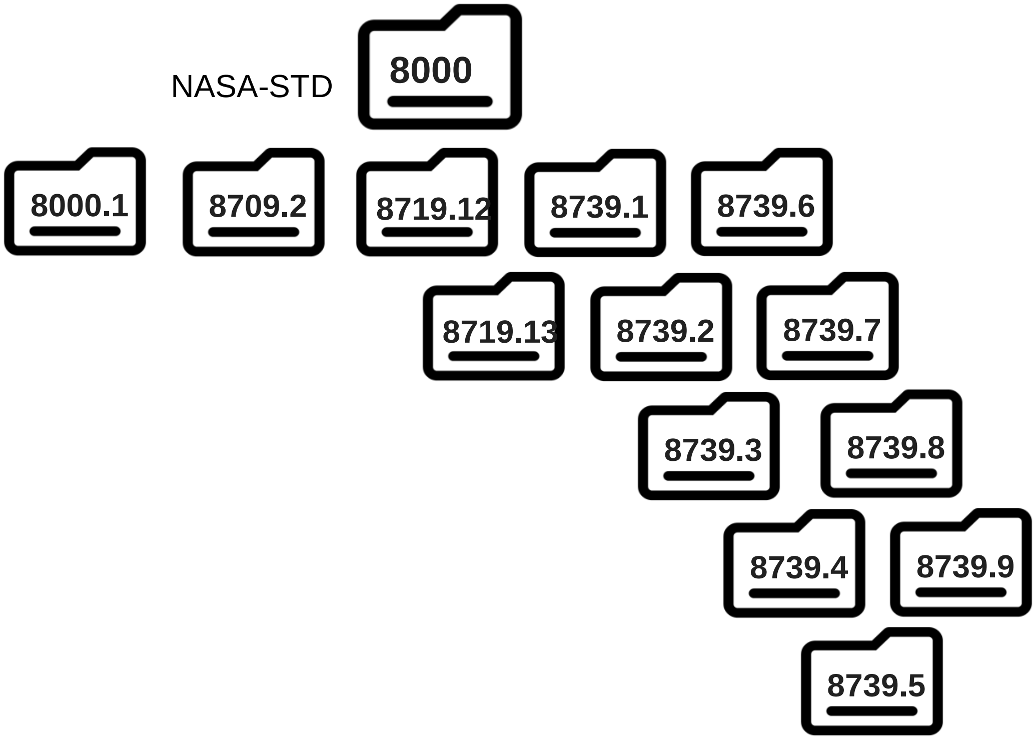
Once we connected everything, they attack surface magnified.



MIT Lincoln Laboratory. (n.d.). Space systems cyber-resiliency. MIT Lincoln Laboratory. https://www.ll.mit.edu/r-d/projects/space-systems-cyber-resiliency

High Stakes Safety and Reliability

Space vehicles need to comply with extensive regulatory criteria



NASA-STD

8000

8000.1

8709.2

8719.12

8739.1

8739.6

8719.13

8739.2

8739.7

8739.3

8739.8

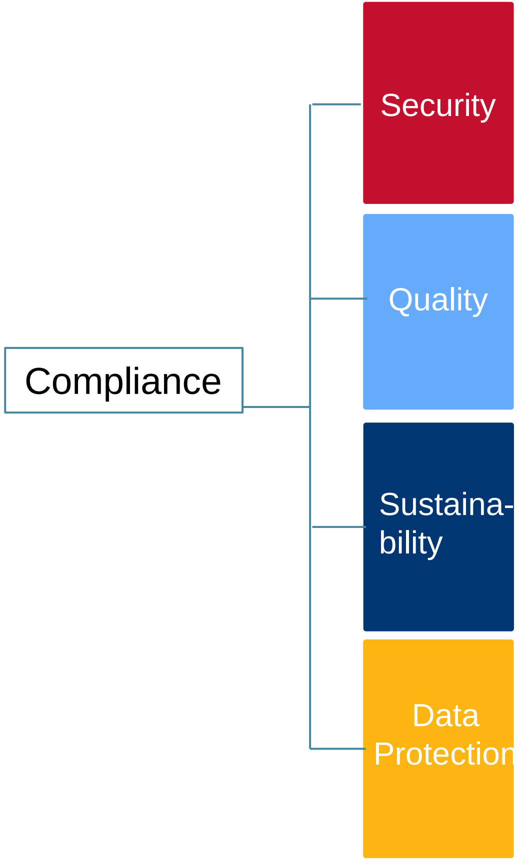
8739.4

8739.9

8739.5

Regulatory and Compliance Hurdles

Safety and Security dictate multiple regulatory standards for cyber-physical systems.



ISO 27001

Requirements for Information Security Mgt System

TISAX

Assessment and exchange mechanism for Information Security

ISO26262

Road vehicles functional safety of electrical and electronic systems

IATF 16949

Technical spec for development of QMS based on 9001

ISO9001

Specifies requirements for Quality Mgt System To demonstrate consistency.

ASPICE

Framework for development process of system / software in automotive industry

ISO 14001

Specifies requirements for environment management for businesses

ISO 45001

Gives guidance for creation and maintenance for occupational health and safety

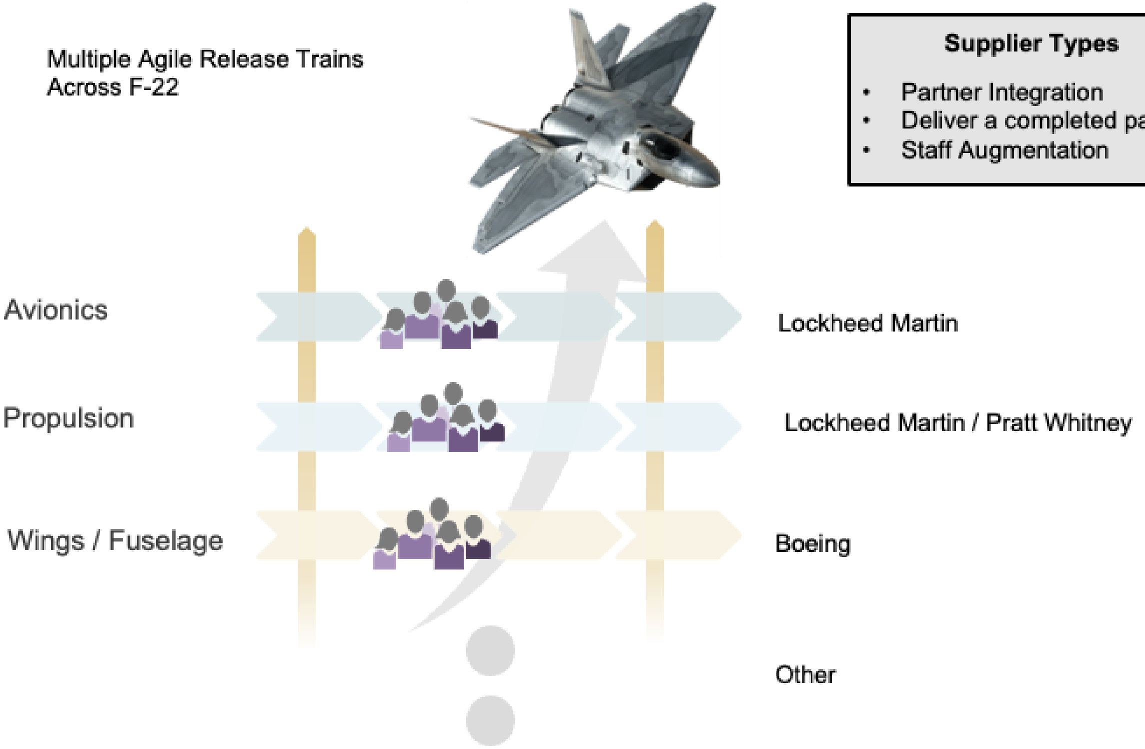
GDPR

Addresses responsible transfer of personal data inside and outside the EU

Supply Chain and Vendor Management

Carnegie Mellon University Software Engineering Institute

Cyber-Physical Systems have extensive supply chains

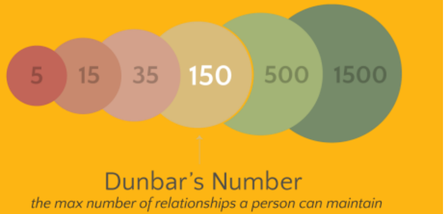

Multiple Agile Release Trains Across F-22
Avionics
Propulsion
Wings / Fuselage
Lockheed Martin
Lockheed Martin / Pratt Whitney
Boeing
Other

Supplier Types

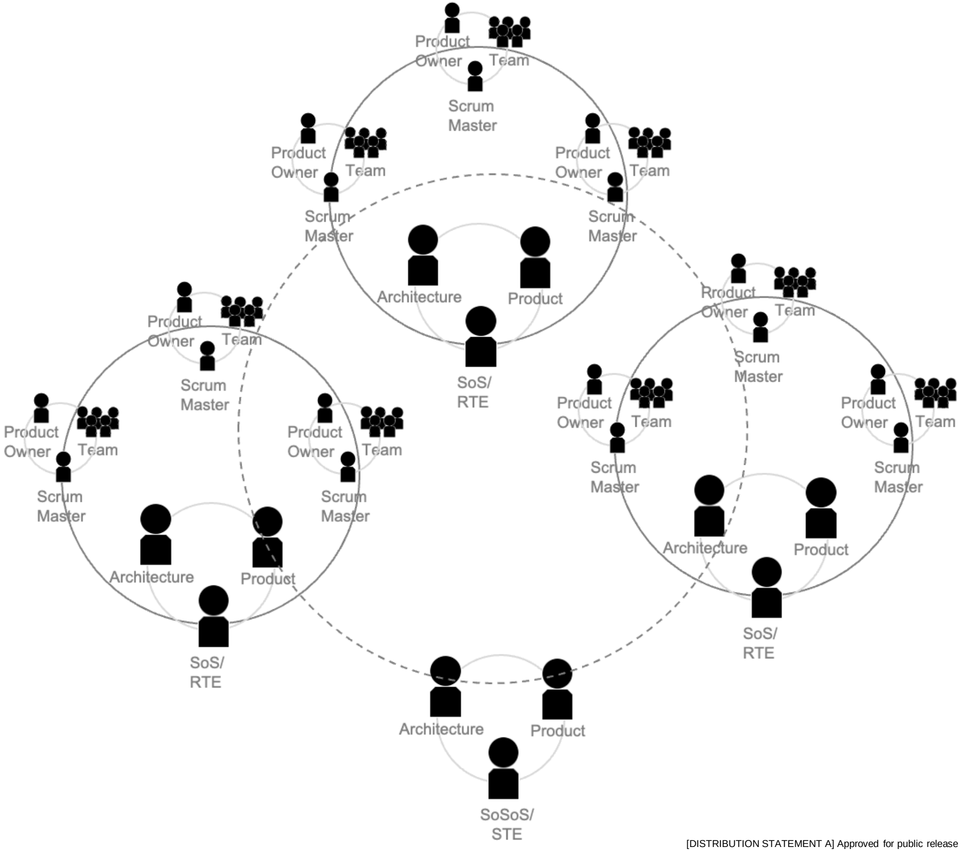
- Partner Integration
- Deliver a completed part
- Staff Augmentation

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Collaboration and Coordination at Scale



As organization grows
The communication
and collaboration
degrades



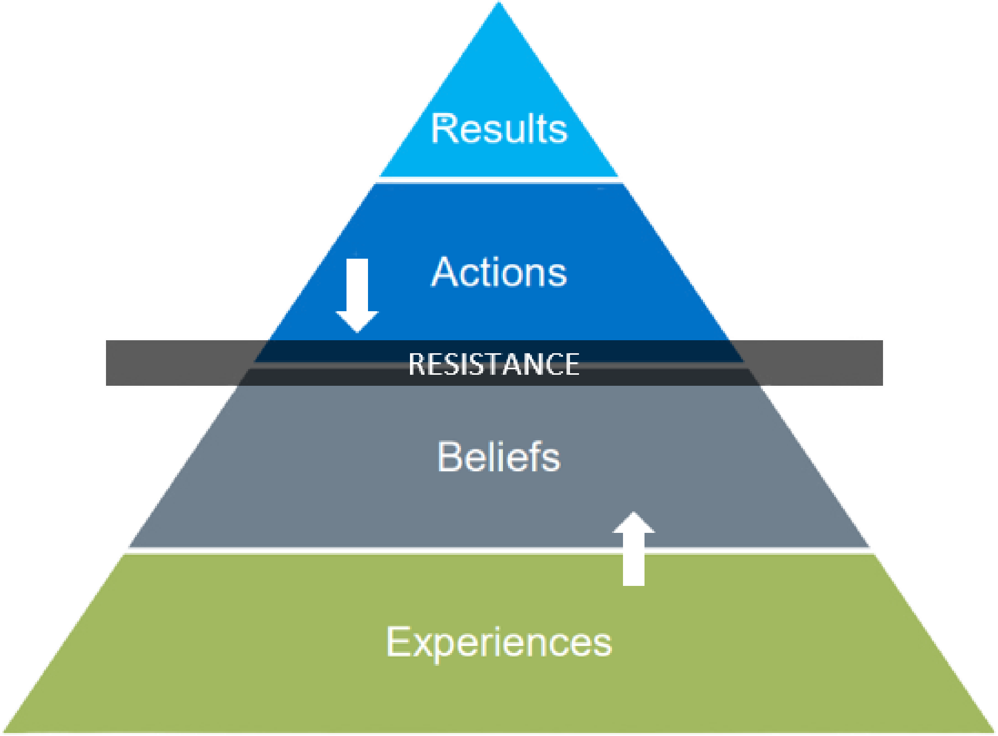
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Culture

The is a shift in culture required to apply Industrial DevOps to the organization.

Change can only come by having different experience.

Greatest Barrier



SOURCE: CHANGE THE CULTURE, CHANGE THE GAME:
The Breakthrough Strategy for Energizing your Organization and Creating Accountability for Results

Exercise

5 Minutes

Carnegie Mellon University Software Engineering Institute

Let Compare Notes, did you have additional challenges that were not noted here.

Key Take Aways

- There are multiple challenges in cyber-physical
- What is complexity
- The greatest barrier to address

Leadership in Orchestrating Digital Engineering

Collaboration and Communication

Carnegie Mellon University Software Engineering Institute

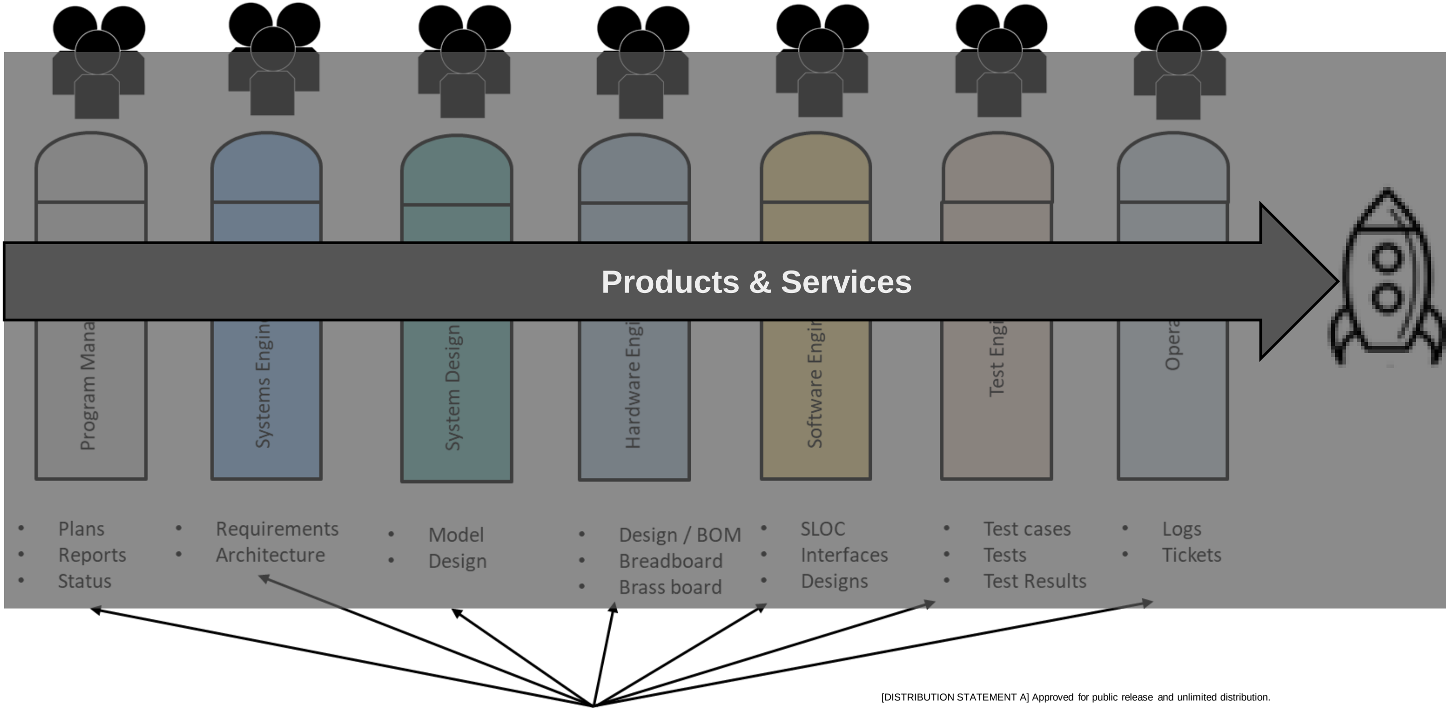
Objectives

Objectives

- Organizing for Value
- Predictive to Empirical Planning
- Delivery through Cadence and Synchronization

Organize Around Value

Communication follows the organizational structure


Program Mana

Systems Engin

System Design

Hardware Engi

Software Engi

Test Engi

Opera

Products & Services

- Plans
- Reports
- Status

- Requirements
- Architecture

- Model
- Design

- Design / BOM
- Breadboard
- Brass board

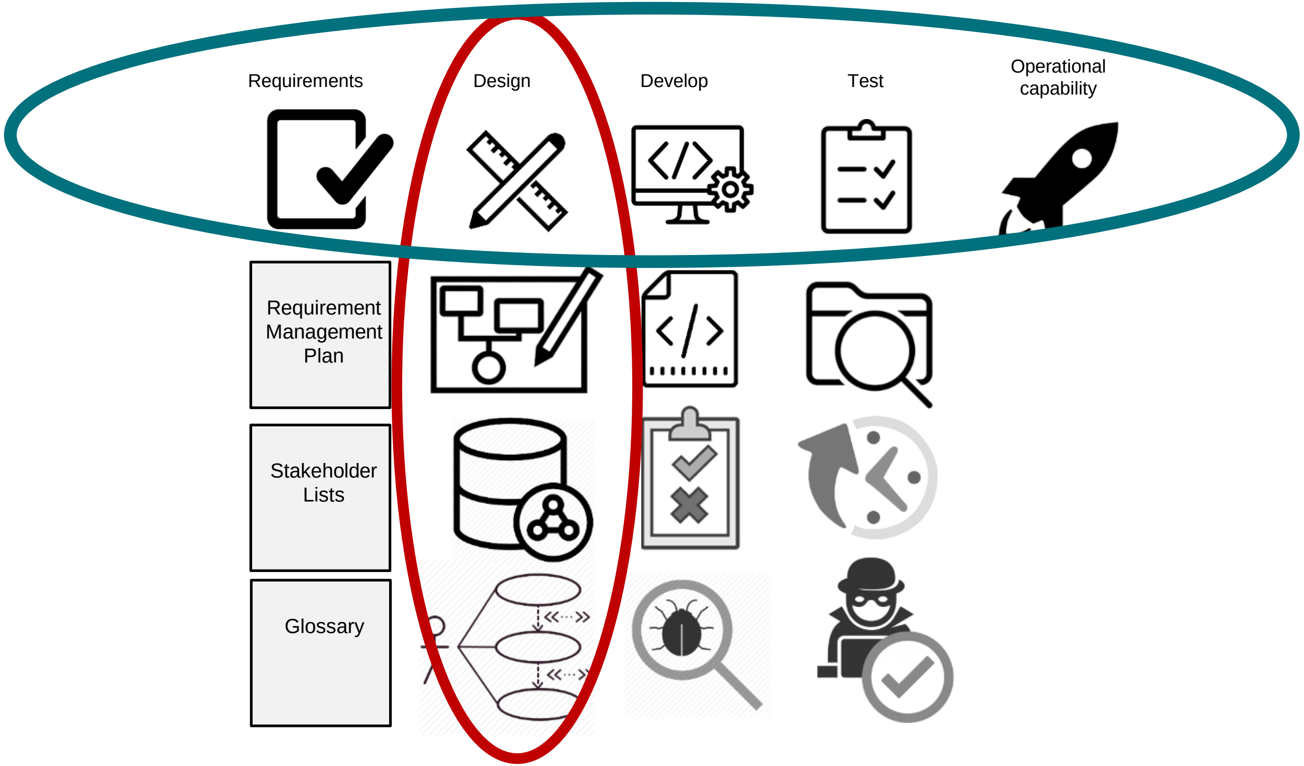
- SLOC
- Interfaces
- Designs

- Test cases
- Tests
- Test Results

- Logs
- Tickets

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

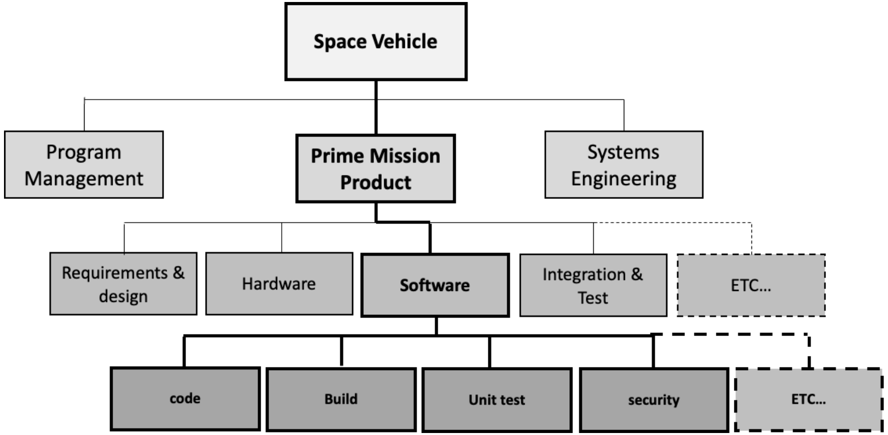
Path to Value

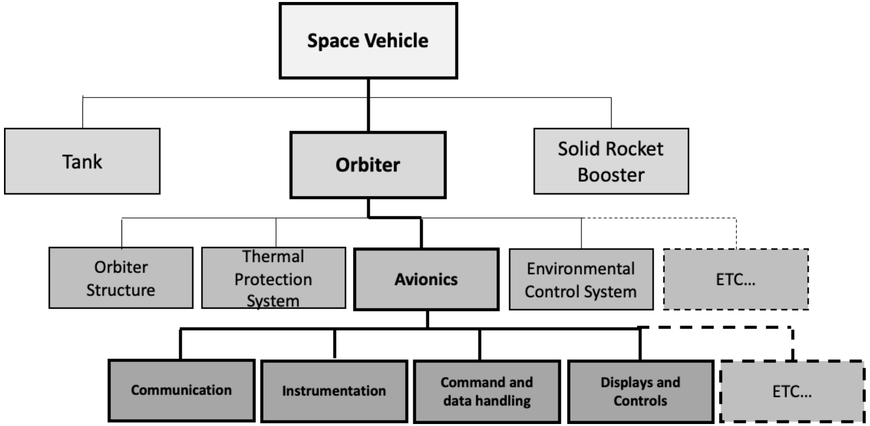


Focus and measure against business value, not inputs

Difference in organization structures

P1 P2 P6 P9



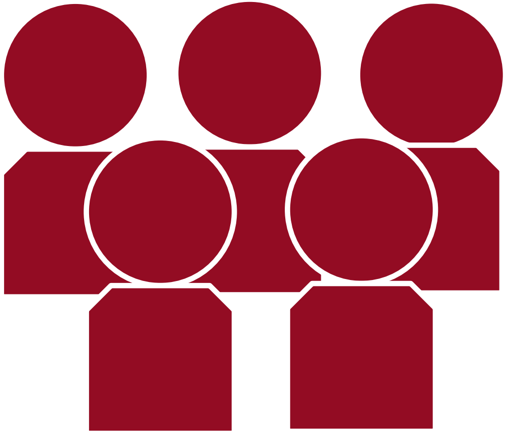


Functional Decomposition– Decomposing a system by function or skill type.

Product Decomposition– Decomposing a system by the products and business outcomes

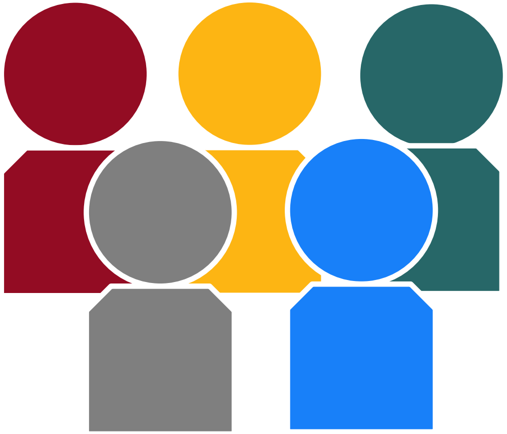
Cross-Functional

Cross-functional teams reduce handoffs which increases throughput and quality!



Cross-Functional Team

Build Solutions

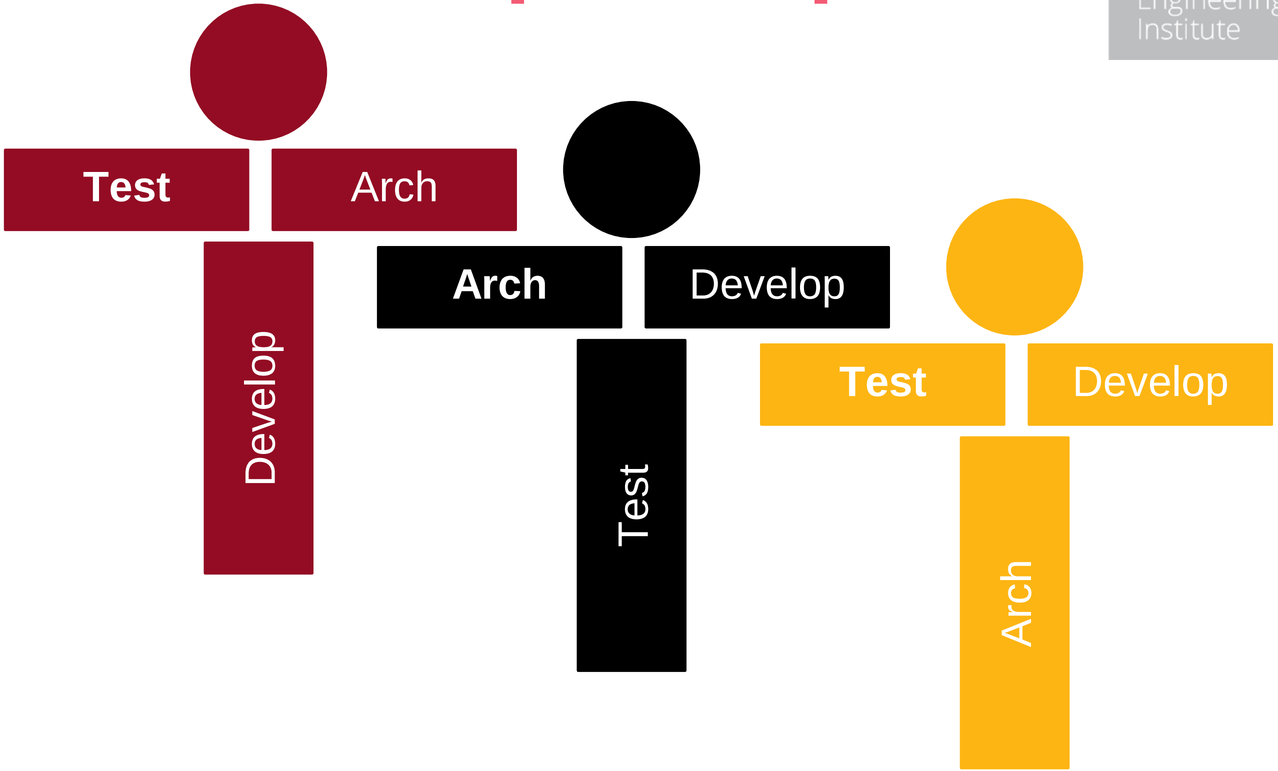


Software Team

Build software

T-Shaped People

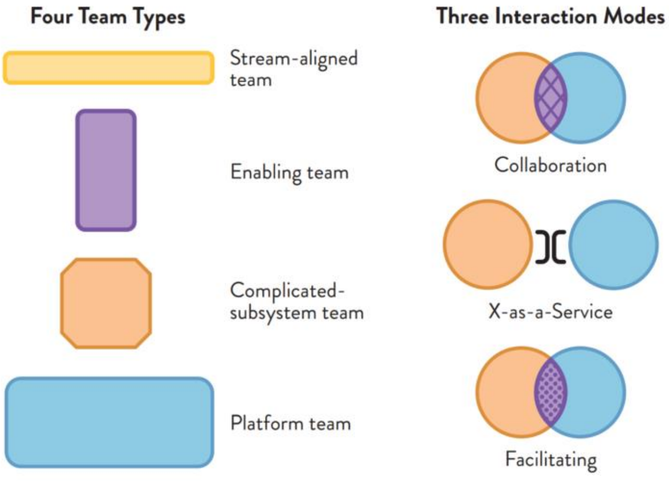
People who have depth in one or more areas and breadth in others.



A team player - who wants to and does contribute to the success of the team and the project

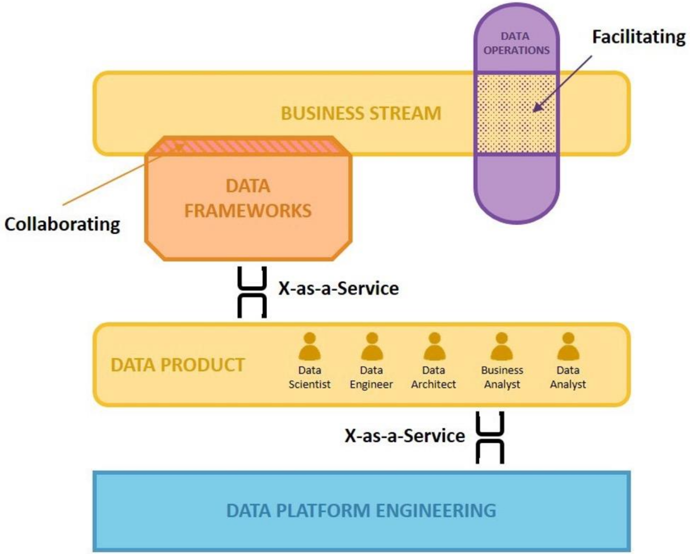
Types of Team

P1 P2 P6 P9



© Matthew Skelton and Manuel Pais from *Team Topologies*

Example applied



Apply multiple horizons of planning

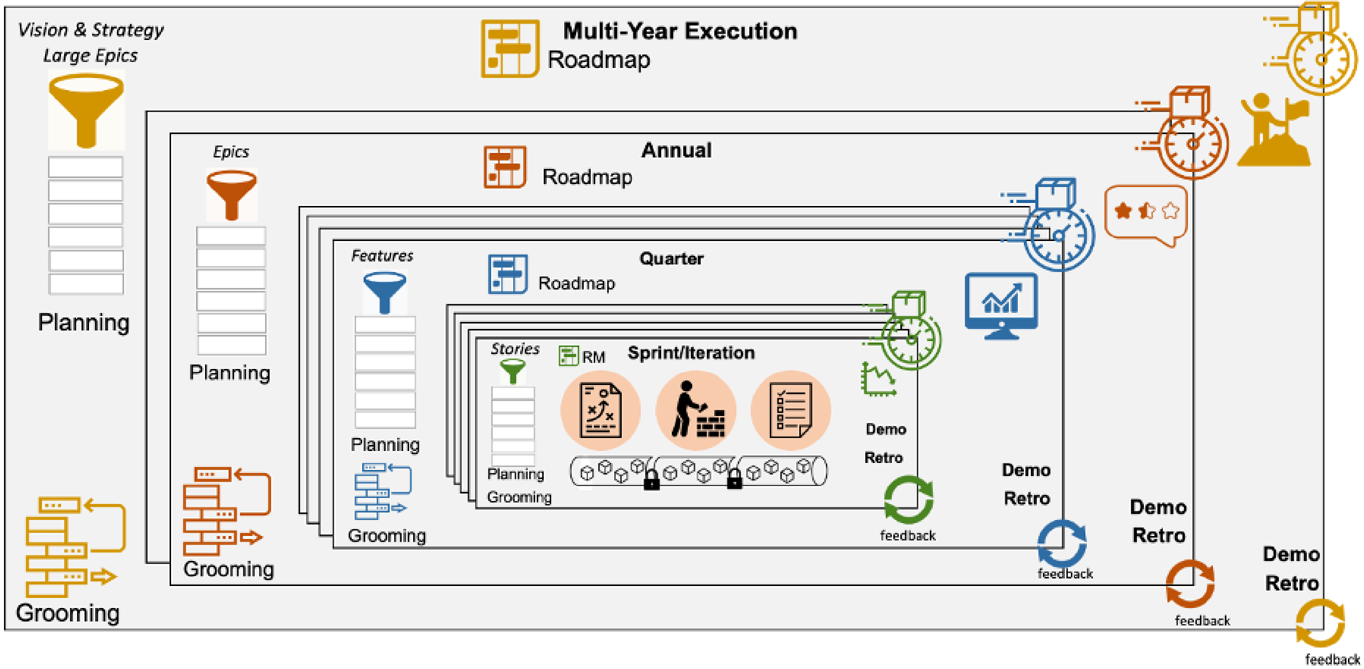
Carnegie Mellon University Software Engineering Institute

P1 P2

P6 P9

“Plans are useless, but planning is indispensable.”

-Dwight D. Eisenhower

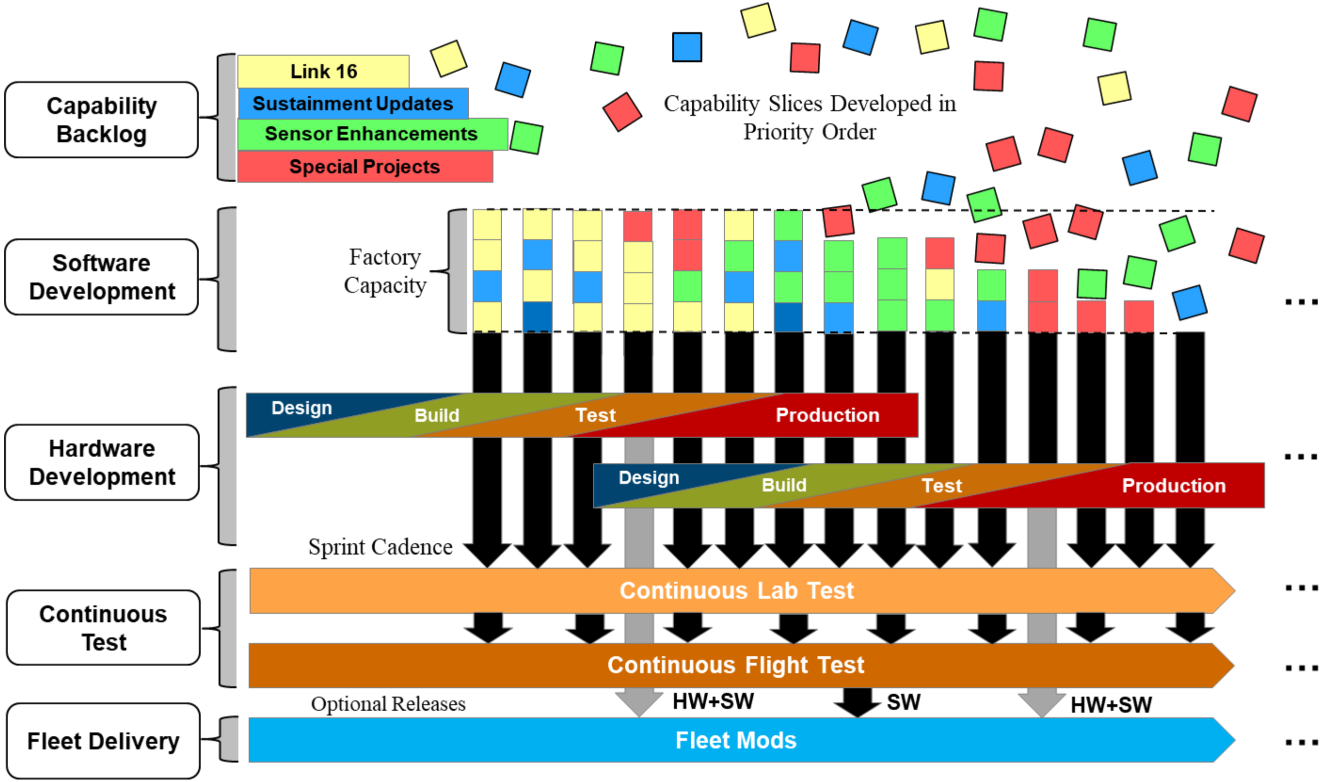


Moving from predictive to empirical

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Cadence and Synchronization

P1 P2 P6 P9



Capability Backlog

Link 16

Sustainment Updates

Sensor Enhancements

Special Projects

Capability Slices Developed in Priority Order

Software Development

Factory Capacity

Hardware Development

Design Build Test Production

Design Build Test Production

Sprint Cadence

Continuous Test

Continuous Lab Test

Continuous Flight Test

Fleet Delivery

Optional Releases

HW+SW

SW

HW+SW

Fleet Mods

...

Exercise

15 Minutes

Group Collaboration

As a team use one of your context and sketch current organizational structure. After you sketch the "AS IS" take some time identify if there are changes you would make and document reasons.

Key Take Aways

- Organizing for Value
- Predictive to Empirical Planning
- Delivery through Cadence and Synchronization

Leadership in Orchestrating Digital Engineering

DevSecOps and AI

DevSecOps Practices that Hold True in AI Development

people, process, pipeline

automation of building, testing, and deployment

small version deliveries

ATO and other security requirements

post-deployment monitoring and data transfer

full stakeholder engagement

New Considerations for DevSecOps

The overall focus is on AI model development, training, testing, validation, and operationalization.

Stakeholders now include AI engineers, data collectors, data scientists, labelers, model evaluators, end users, operators, and operational environment SMEs.

Sets of pipelines automate AI model development, operationalization, and data curation.

Output from the development pipeline typically takes the form of an AI model in various stages of training and testing.

Post deployment primarily monitors AI model accuracy.

The pipelines include testing for bias and subjectivity in the AI model.

The Three Main DevSecOps Pipelines

There are three main pipelines for creating a complete development, deployment, and monitoring system.

Data curation: ingests raw data and converts it to a usable format for AI model development.

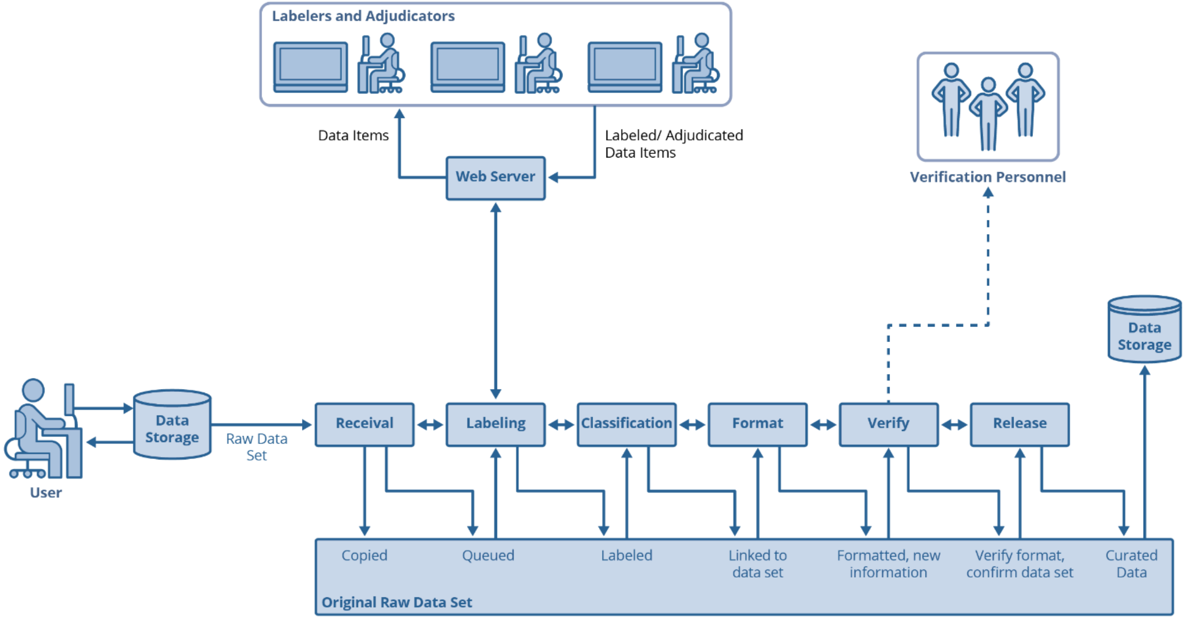
Model development: applies tools for coding, training, testing, and validating a model in staging and operational environments.

Operationalization: packages a completed AI model in a usable form as a component of some larger operational system.

Data Curation Pipeline

The pipeline

- inputs raw data and outputs curated data.
- implements each step to label, classify, and format data.
- may need personnel to label raw data.
- should run before and parallel to dev pipeline.



Verification Phase

This phase assures a correctly formatted data set.

More importantly, it validates the following:

- The data is objective.
- The data is compatible with expected use cases.
- There are no individual assumptions.
- Adversarial AI is absent.

The phase could benefit from both human and machine inspection.

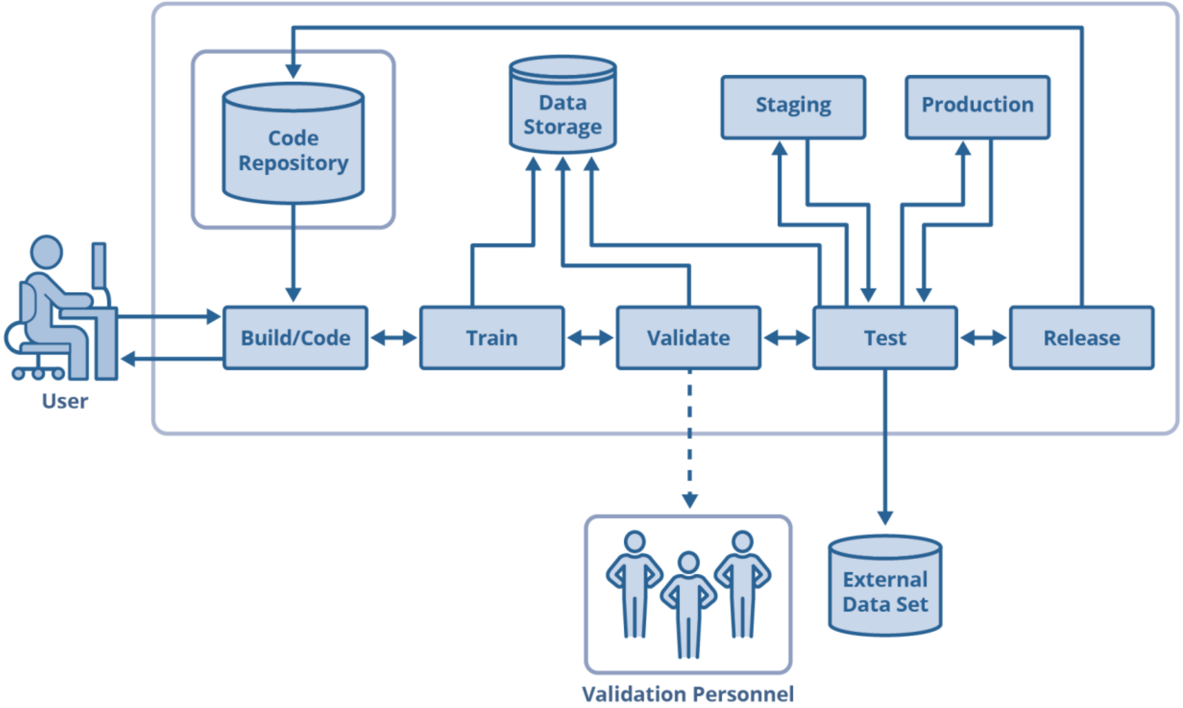
Consider setting parameters with automated checking to ensure you meet all the points above.

This phase supports Shift-Left testing → finds issues in data before its use in AI model development.

AI Model Development Pipeline

The pipeline

- builds model implementation code.
- trains and validates models.
- includes feedback loop at each step.
- uses processed data from curation pipeline via common storage.
- outputs a trained and tested AI model.



AI Model Operationalization Pipeline

The pipeline

makes models that are usable in the real world.

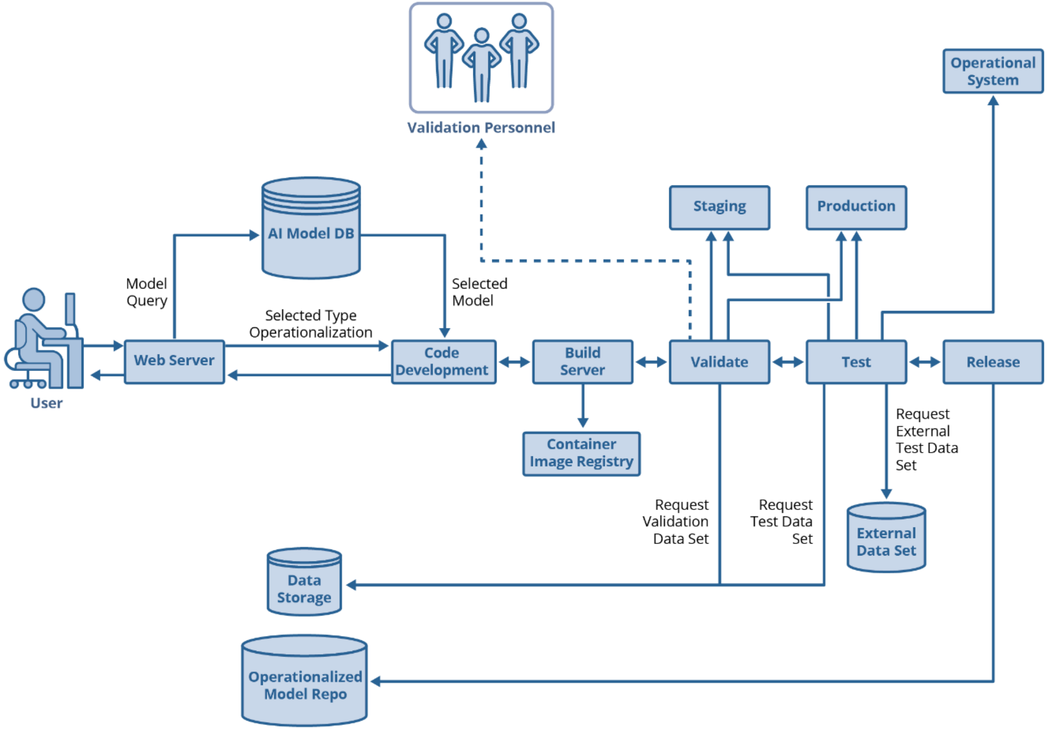
packages in a deployable artifact.

may need data curation pipeline.

tests in operational environments and systems.

supports OT&E.

provides public methods for data ingress, prediction, and continuous monitoring.



Setbacks in Operationalization Efforts

Unable to deploy models: *60% of the models that data science teams have spent months developing, testing, and verifying don’t make the leap from the data science team into operations.**

*https://gritdaily.com/why-60-percent-of-machine-learning-projects-are-never-implemented/

Post-Deployment Monitoring

Monitoring

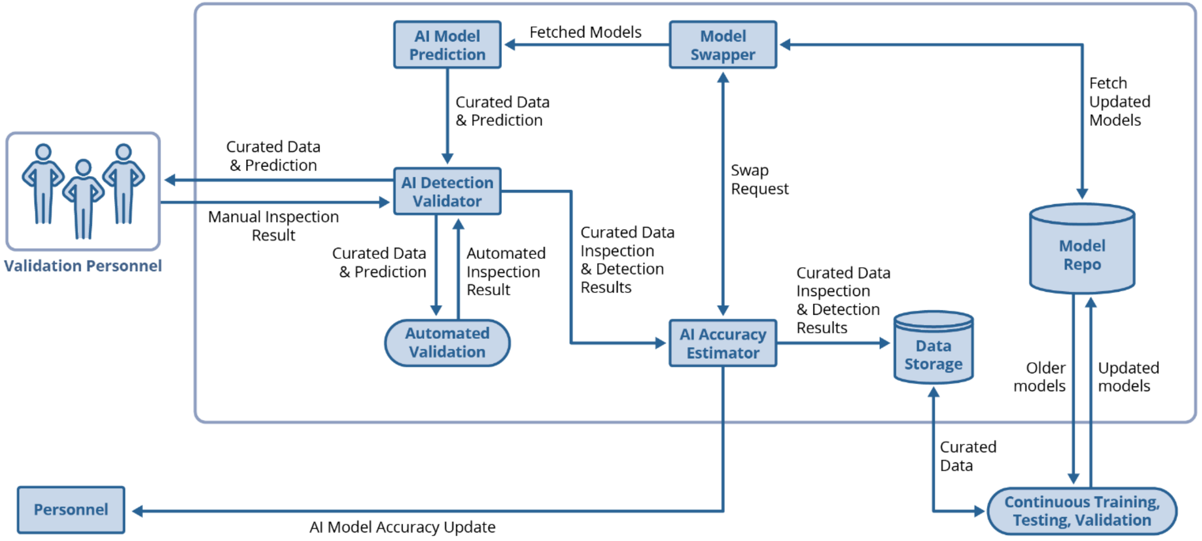
works with operationally deployed models.

validates each model prediction.

constantly trains models with real-world data.

can dynamically swap models.

may include personnel for prediction validation.



Validation Model Testing and Monitoring

This phase occurs in development, operationalization, and post-deployment monitoring.

The phase helps establish trust in the model.

During development and operationalization, this phase creates and runs test cases that check the model’s decision making for the following:

- bias, subjectivity, and assumptions
- incompatibility with target population
- other issues that indicate a lack of objectivity or relevancy to target population
- the potential presence of adversarial AI

Supports Shift-Left testing → identify improper functionality prior to deployment.

Validation of the items above may require data set review, recreation, model retraining, and personnel.

Build AI System with DevSecOps. *(DevSecOps for AI)*

CREATE

- Model Selection
- *Training Time (Blocker)*
- Build "ilities" in
- Iterative Parameter Tuning Configuration & Management
- Scalable Oversight
- Regularization

PLAN

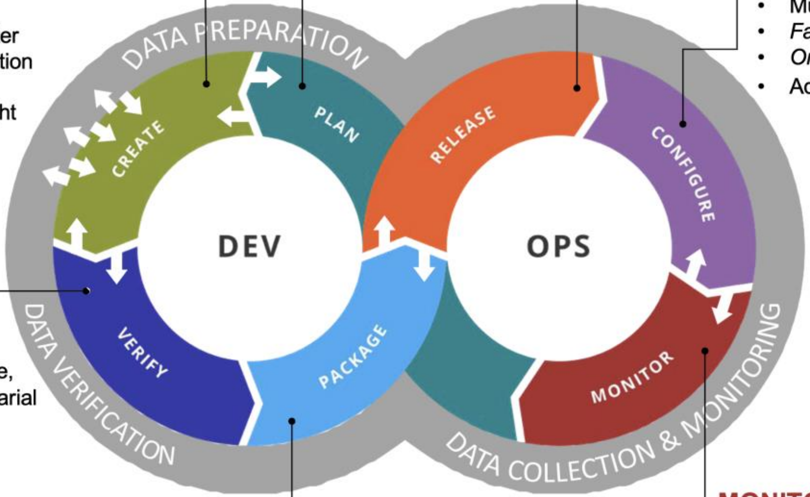
- *Metrics*: Performance, Robustness,…
- Architecture Decisions
- Data Preparation

RELEASE

- *Deployment Models*
- Compute Infrastructure
- Mission Risk

CONFIGURE

- Multiple Algorithms
- *Fallback/Rollback*
- *Online Learning*
- Adaptation



VERIFY

- Inspection & Introspection
- Testing: Performance, Robustness, Adversarial
- Security & Privacy
- *Push Back*

PACKAGE

- Integration into larger complex system
- *Algorithmic Agility* (multiple algorithms)
- Architecture Decisions
- Data Preparation

MONITOR

- Adversarial Attack
- Concept Drift
- Detection and Prevention
- "Safe" Operating Boundaries

Major Considerations:

1. Data is necessary throughout the AI/ML DevOps process; data dependencies are hard to track, assess and analyze
2. Emergent behavior in deployment model and from online adaptation
3. Modeling and training can block the speed of creation
4. Verification can push the DevOps cycle backwards at multiple stages

Leadership in Orchestrating Digital Engineering

Infusing AI into SWE, DevSecOps and MBSE

Carnegie Mellon University Software Engineering Institute

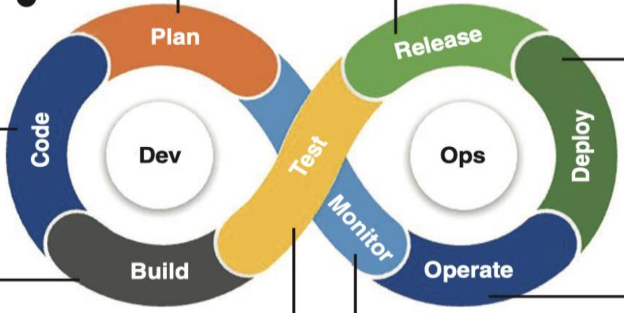
Infusing AI into SWE and DevSecOps *(AI for DeSecOps)*

Code
- Architectural Design
- GAI-Based Pair Programming
- Code and Unit Testing Generation
- IDE Secure Code Vulnerability Solution
- ML-Assisted Code Review Selection
- AI-Assisted Code Review
- AI-Enabled Collaboration
- Suggestive Refactoring

Plan
- Natural Language Requirements Gathering
- NLP Requirements Analysis for Inconsistency and Ambiguity
- GAI Epic and User Story Generation
- Effort Estimation Using Neural Networks
- GAI-Assisted Threat Model Policy Identification

Release
- Compliance Validation
- Reinforced Learning-Based Models Generate Deployment Scripts
- AI-Enabled Failure Analysis
- Release Risk/Success Prediction
- AI-Driven CI/CD Workflow Automation

Deploy
- Dynamic Environment Provisioning and Deployment Optimization
- Real-Time Rollback
- AI-Assisted Log Aggregation
- ML Anomaly Detection
- GAI Deployment Scenario Simulations



Security Is Infused into All Actions and Activities

AIOps Engines Provide Correlation and Predictive Monitoring

Build
- Aggregated Merge Request Impact Analysis
- GAI-Based Identification of Security Vulnerabilities
- ML Algorithm Optimized Build Times
- AI-Assisted Security Vulnerability Detection
- Software Composition Analysis

Test
- Natural Language Test Case Generation
- Test Data Generation
- AI-Enabled Test Effectiveness Predictions
- End-to-End Functional Test Execution
- Intelligent Failure/Self Healing Testing
- NLP-Based API-Based Contract Definition
- Intelligent Test Execution

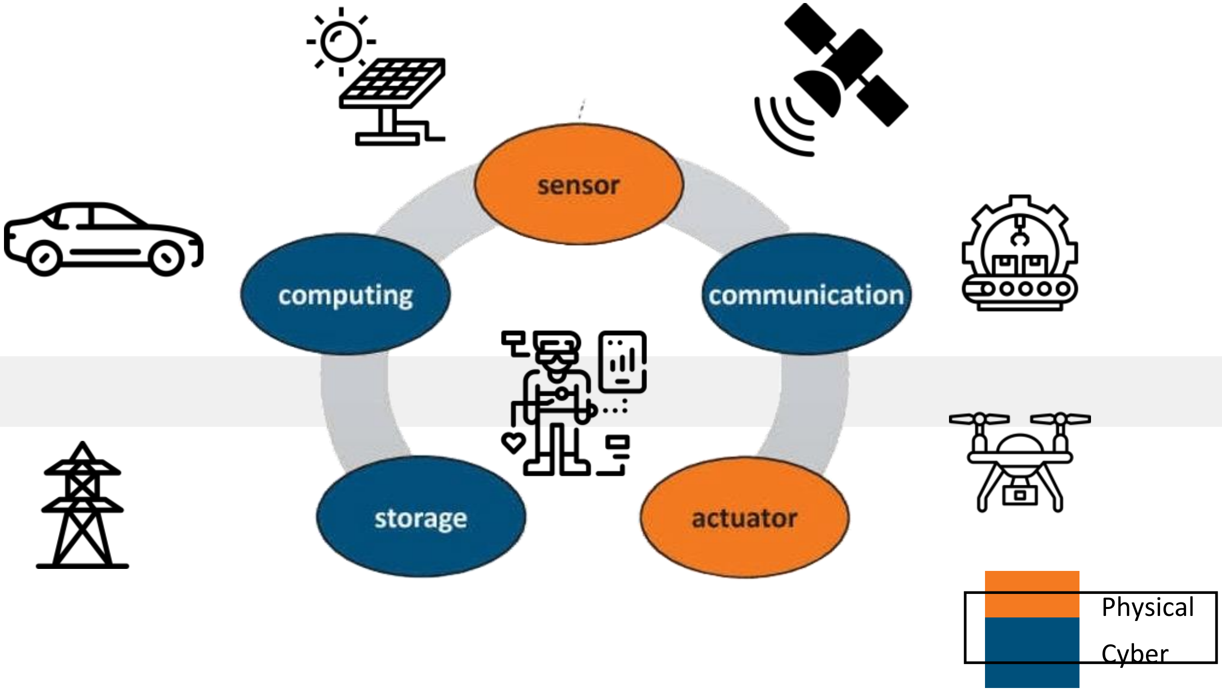
Monitor
- Event Correlation
- False Alarm Filtering
- Self-Healing Techniques
- Root Cause Analysis
- Observe System Performance
- Usability Patterns
- Monitoring

Operate
- Deterministic AI-Based Ticketing and Support Allocation
- AI-Based Self Healing Decision
- LLM Integration for Virtual Assistance
- GAI/GPT-Powered Knowledge Bases

Role of AI in Autonomous Vehicles and ADAS (Case Study)

- Most of the application of AI in autonomous vehicle technology and ADAS (Advanced Driver-Assistant System) features with AI.
- Various types of AI models used, such as machine learning for predictive maintenance, computer vision for object detection, and neural networks for decision-making processes.
- ADAS built with the Synergy of Agile, DevSecOps, and AI

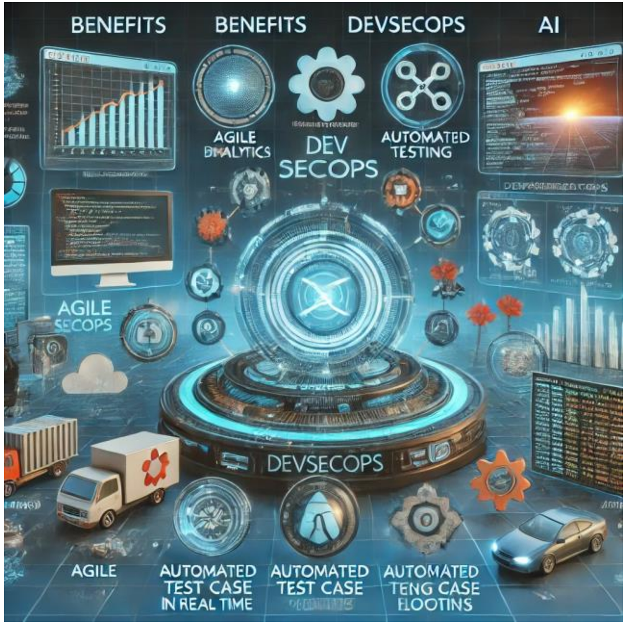
Automotive Cyber-Physical System (CPS)



Most of these systems are software defined and hardware enabled

Benefits of Adopting These Technologies

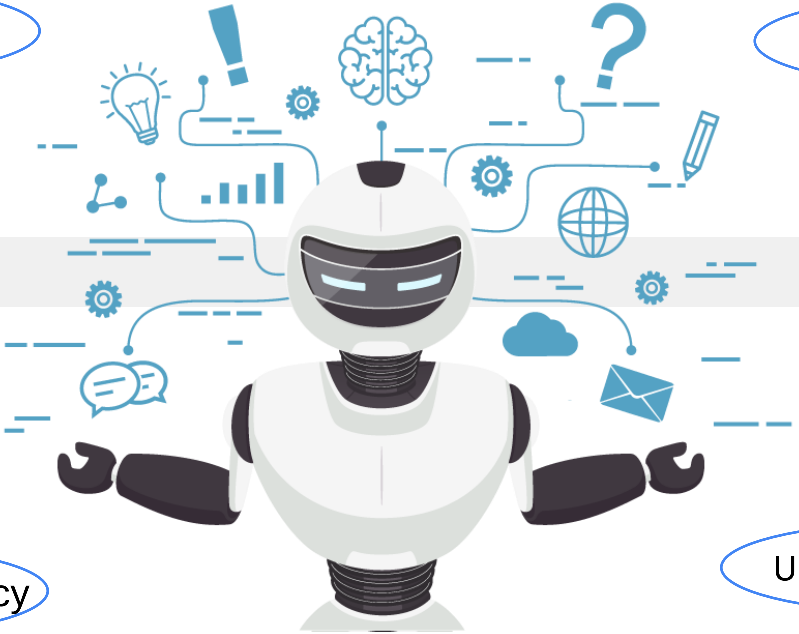
- Increased efficiency and reduced time to market.
- The use of AI in automating testing procedures.
- Improving feedback loops through AI analytics.
- AI’s role in automated test case generation and execution.
- Enhancing Agile sprints with AI-driven testing tools.
- DevOps and AI for real-time bug tracking and resolution
- Automating testing procedures with of AI.
- Improved safety features through continuous testing and updates.
- Enhanced ability to respond to customer needs and market changes.
- Using AI for automated test case generation and anomaly detection.



Benefits AI for CPS

Reduced Error

Learn Faster



Faster Decisions

Unbiased Analysis

Increased Efficiency

Unbiased Decisions

Reshaping Testing with AI

- The use of AI in automating testing procedures.
- Improving feedback loops through AI analytics.
- Enhancing Agile sprints with AI-driven testing tools.
- DevOps and AI for real-time bug tracking and resolution
- Using AI for automated test case generation and anomaly detection.
- AI has a signficant role in automated test case generation and execution.

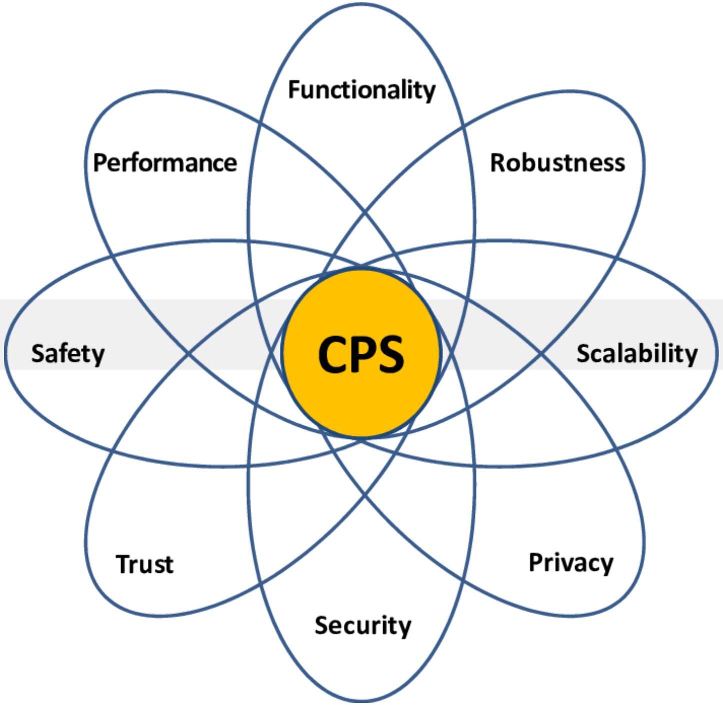
SW Testing is reshaping with AI
Ral-time Testing
Agaires Sesting
SevOps
Real-Time Age Spitns
Real-time Bug tacking with Analyia
Real-tore Bug tracking with anaımal aıts
sevo7s
DEVOPS
DevOps

Innovating Quality Assurance

- The impact of AI on quality metrics and standards.
- Using predictive analytics to identify potential areas of failure.
- Continuous quality monitoring through machine learning.
- Apply Ai with Digital Twins.
- AI further augments the digital twins by leveraging advanced algorithms and machine learning techniques to manage and analyze data, uncover patterns, and drive intelligent automation

AI AI
CONTINUOUS QUALITY
Predictive Driven Quality Assurateng
AI
Digital Twins
Digital Twins
ITAL TWINS
Digital TWINS
Continuous Quality Noitoring using Foisitons
CONTINUIS MONITOORING USING

Challenges


Functionality
Performance
Robustness
Safety
CPS
Scalability
Trust
Privacy
Security

Key Challenges in Integration

- Key technical and organizational challenges.

  Potential challenges in integrating Generative AI with Agile and DevOps.

  IP Right, resource, tools and platform
- The need for specialized skills and cross-functional collaboration.
- Regulatory challenges and safety concerns.
- Highlight the potential for increased cybersecurity risks.
- Address the complexity of integrating AI within Agile and DevOps frameworks.

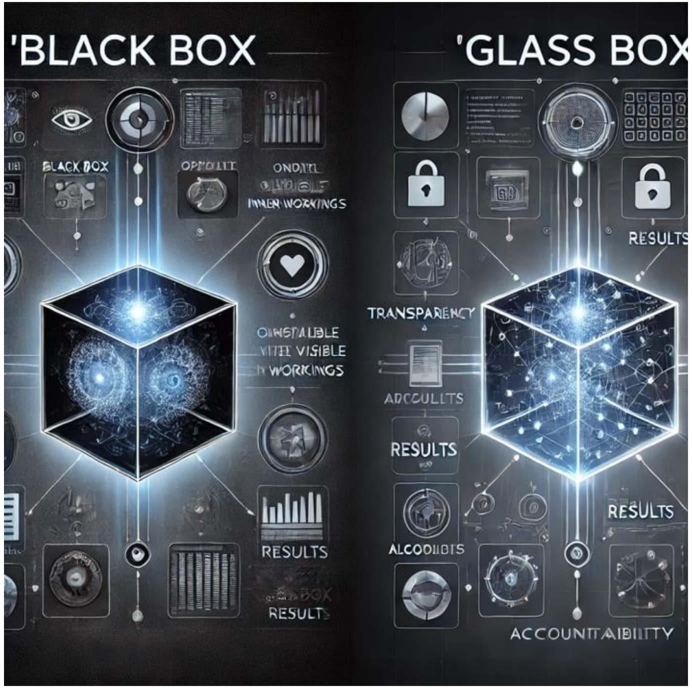
Transparency AI assistance in CPS Systems

- Increased role of GAI in software development bring issues on :transparency, ethics, and bias in AI.

the "black box" problem in AI, where decision-making processes are not visible.

transparency is critical for trust and accountability in software systems.

Misusage of AI supported decision and recommendation

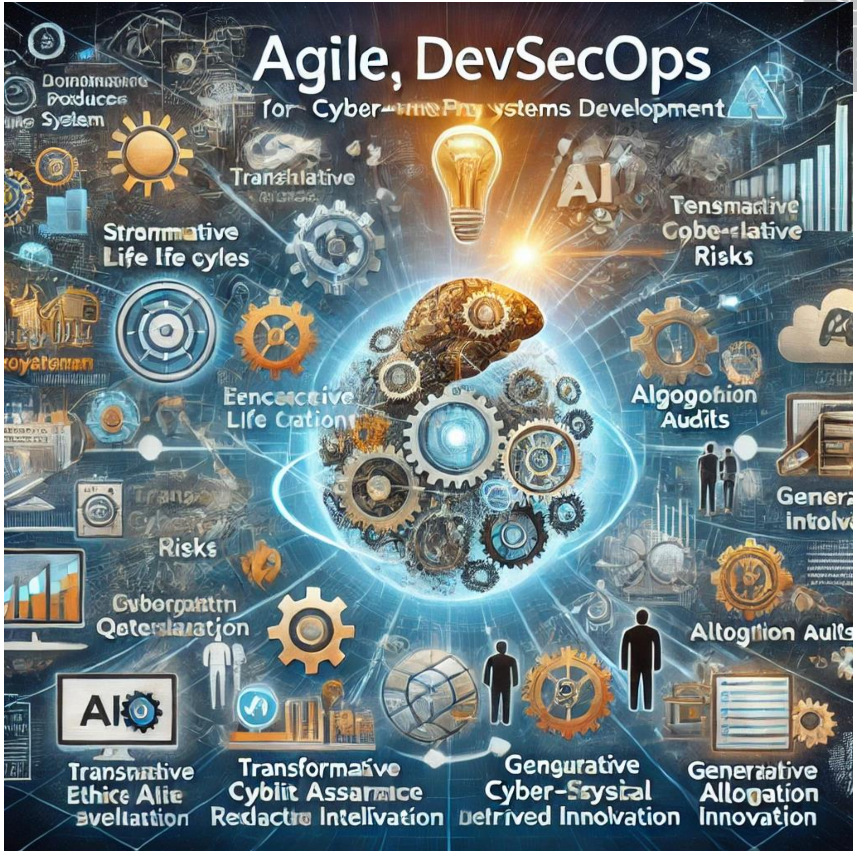


Ethics and Bias Challenges

- The ethical considerations when developing and deploying AI systems, such as privacy concerns and the potential for misuse.
- The prevalence of bias in AI, how it arises from training data and algorithmic design, and its implications.
- Offer strategies for mitigating bias and ensuring ethical compliance, such as diverse data sets, algorithm audits, and inclusive design practices.

Synergies in between these concepts

- Integrating these elements can be ***Transformative*** in system development
- Find a balance between these concepts by:
- *Trust but Verify*
- *Keep Human in the loop*
- *Apply core SWE principles*
- *Apply traceability across SDLC including Operations*


Agile, DevSecOps
for Cyber-ᴛʜᴇPʜy ystems Development
Dointinning Poduces Sylem
Tranhlative
AI
Stronmative Life lfe cyles
Tensmartive Cobe-slative Risks
Encecctive Life Cation
Algogohion Audits
Risks
Genera intolv
Cyborgantin Qeterslaution
Altogition Aults
AI
Transmative Ethice Alie Bvellattion
Transformative Cybilt Assarnice Reclactro Intellvation
Gengurative Cyber-Ssystal Detrived Innolvation
Generative Allogation Innovation

Ref: all images were created by DALL-E

References

DevOps:	https://www.sei.cmu.edu/go/devops
DevOps Blog:	https://insights.sei.cmu.edu/devops
Webinar :	https://www.sei.cmu.edu/publications/webinars/index.cfm
Podcast :	https://www.sei.cmu.edu/publications/podcasts/index.cfm

Q & A

ASK AWAY!

Thank you

Hasan Yasar

Technical Director

hyasar@cmu.edu