

Mastering Agile Systems Engineering for Agile Practitioners

Nate Crews ncrews.btcs@gmail.com

System Engineering Instructor & Enterprise Agile Transformation Coach

California Institute of Technology

Center for Technology and Management Education

Agenda

- Agility Background
- Agile Systems Engineering
- Agile Systems Engineering Techniques
- Agile Systems Engineering Frameworks
- Summary/Questions

Dr. Nathaniel P. (Nate) Crews Ph.D

- 23 Years in Industries using Systems Engineering
- 20 Years Working and Teaching Agility
- Taught over 10,000 hours of Quality Instruction
- Over 55 Certifications on PM, Leadership & Agility
- Senior Instructor/Technology Consultant
- System Engineer & PM Instructor at Cal Tech
- Project Management Professional (PMP)
- Certified Agile & Scrum Professional (ACP & CSP)
- Certified Business Analyst & System Engineer
- SAFe 6.0 Program Consultant (SPC6)
- Executive Masters in Technology Management, Wharton School of Business
- Doctorate in Management & Organization Development
- Approach: Applicable, Practical, Engaging & Fun
- Interests: Life Long Learner, Sports, Karaoke



Email: ncrews.btcs@gmail.com

Industries require the development of increasing complex products to support their customers



Automotive



Transportation



Medical Equipment



Defense/Energy



Banking & Finance



Aerospace



Commercial Products



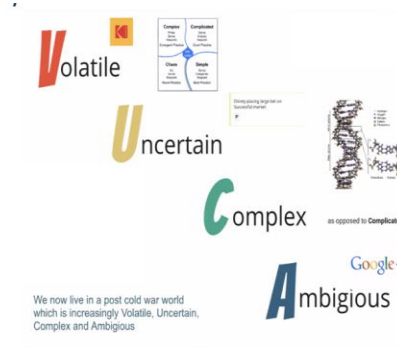
Software

An integrated set of elements, subsystems, or assemblies that accomplish a defined purpose; these elements include products (hardware, software, and firmware), processes, people, information, techniques, facilities, services, and other support elements (INCOSE Systems Engineering Handbook, 2010)

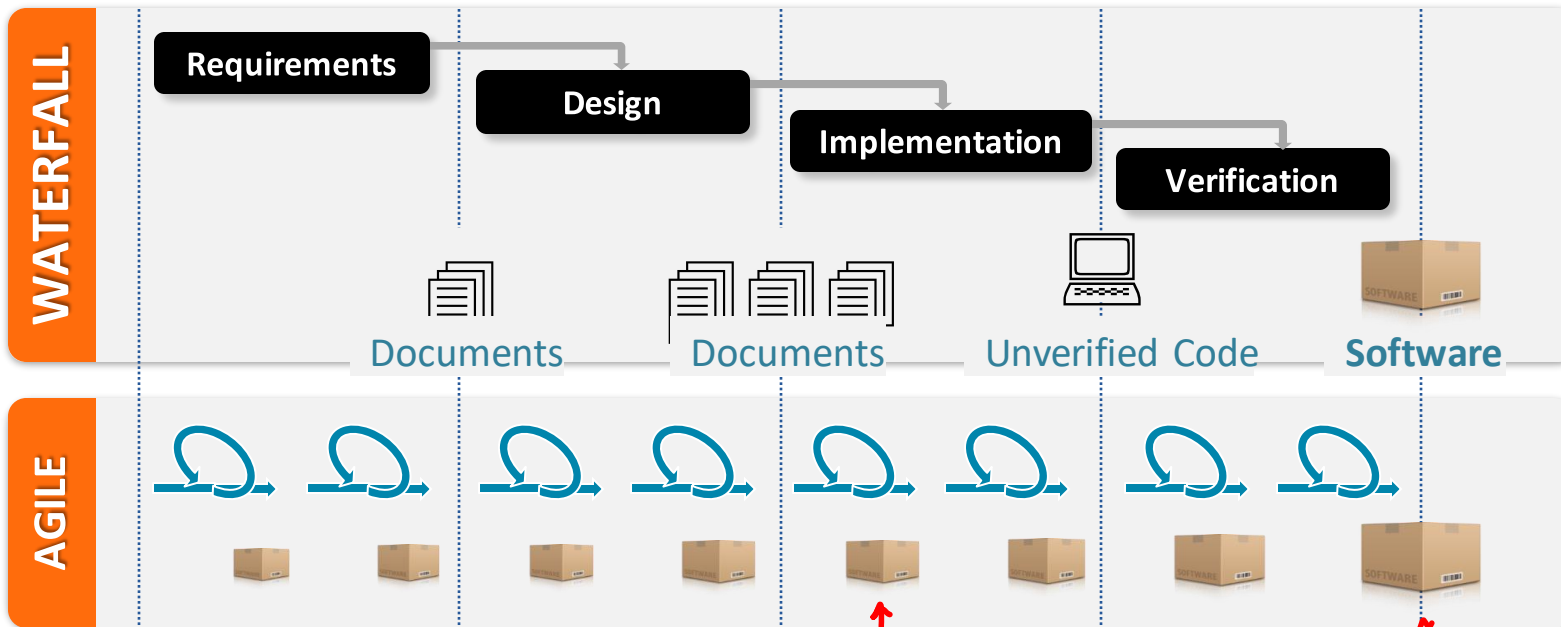
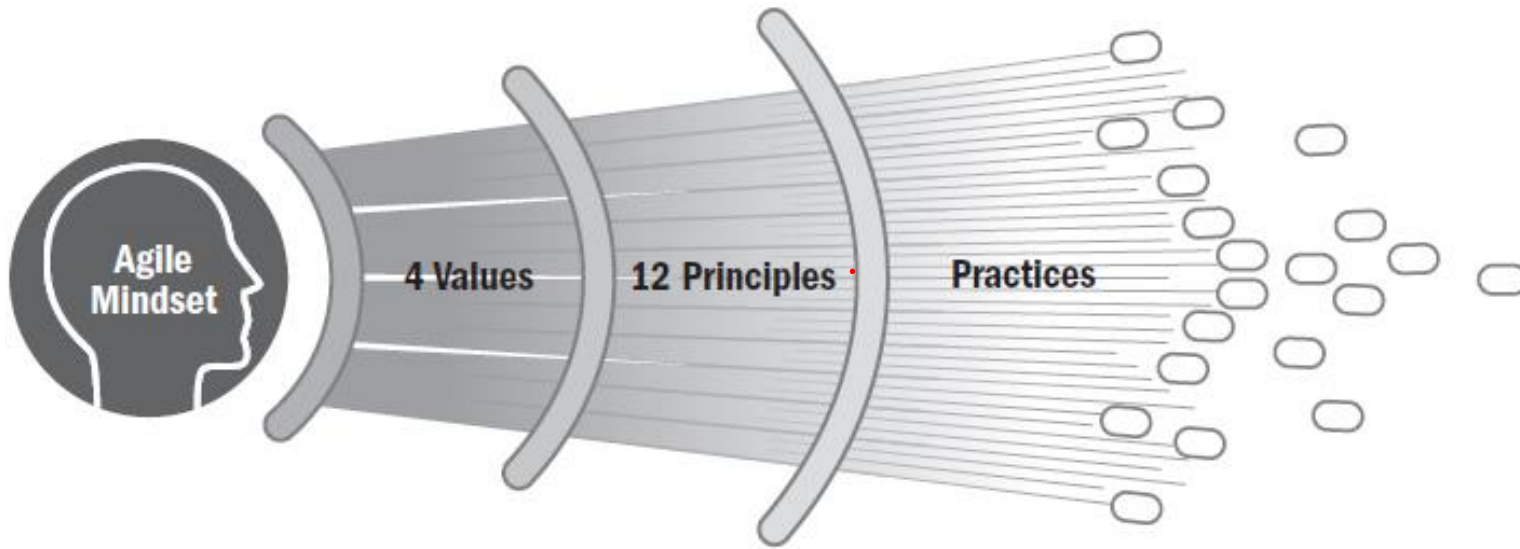
Why Agile Systems Engineering?

Agile Systems Engineering offers several advantages for managing and developing complex embedded products:

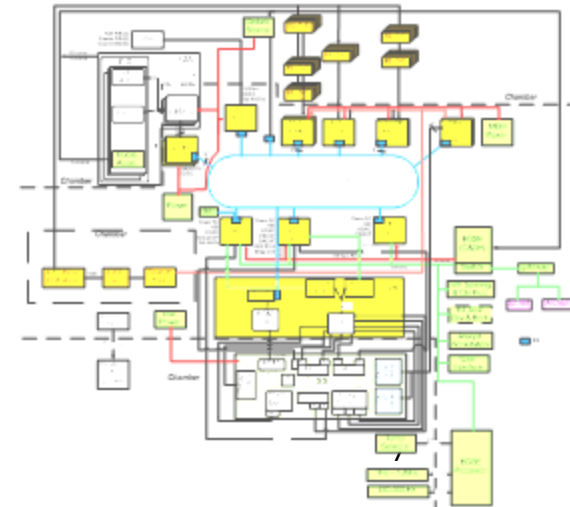
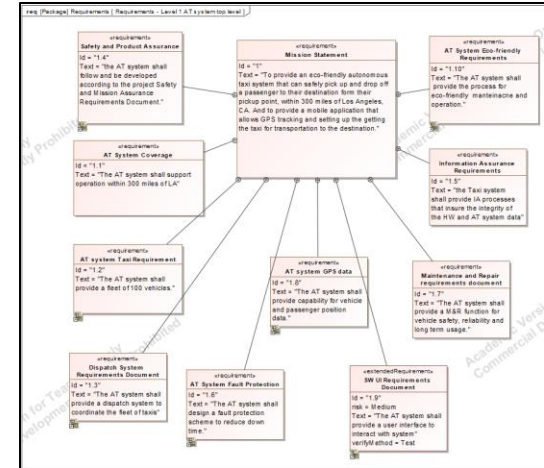
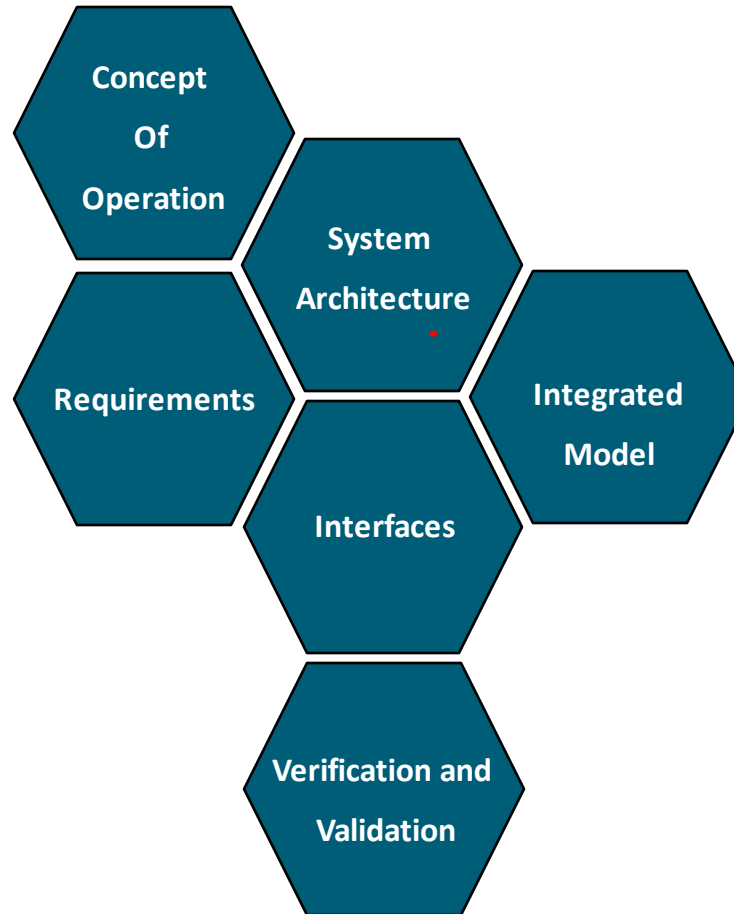
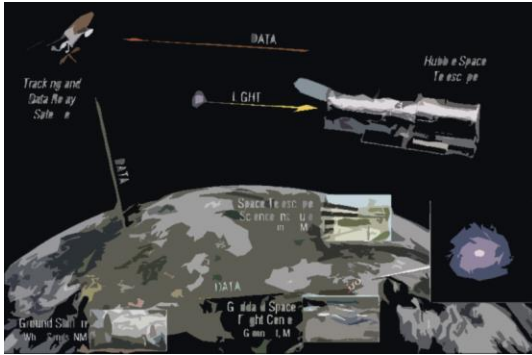
- **Enhanced Adaptability and Flexibility:** Agile Systems Engineering embraces flexibility and adaptability as core principles.
- Agile methodologies, such as Scrum or Kanban, enable teams to respond quickly to changes in project requirements or priorities
- By embracing change and iterative development, Agile Systems Engineering reduces the risk of costly late-stage modifications and ensures that the product remains relevant and competitive.
- **Faster Time-to-Market and Reduced Costs:** Agile Systems Engineering promotes incremental development and frequent releases of usable product increments. This approach accelerates the time-to-market for complex embedded products.
- **Comprehensive Risk Management:** Comprehensive risk management is a strategic approach that encompasses the identification, assessment, mitigation, and monitoring of risks across all aspects of an organization's operations. When discussing "more comprehensive risk management," it typically refers to an approach that is more thorough, inclusive, and proactive in addressing various types of risks.



Agile Development Overview

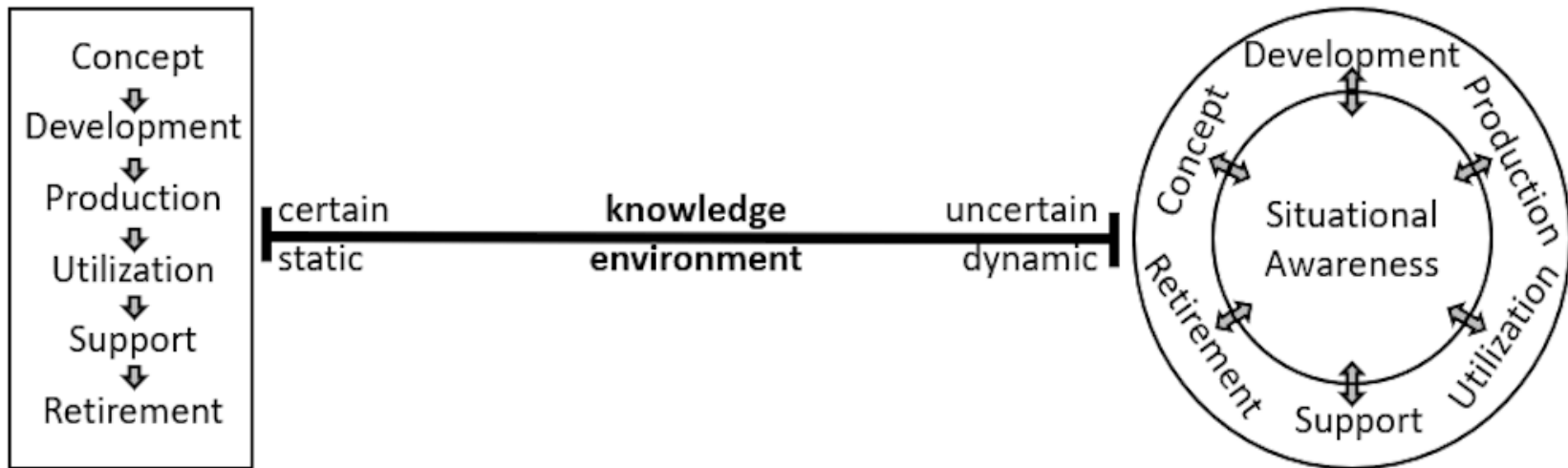


Systems Engineering



Agile System Engineering Overview

Agile Systems Engineering is a principle-based methods for designing, building, sustaining, and evolving systems when knowledge is uncertain and/or environments are dynamic.

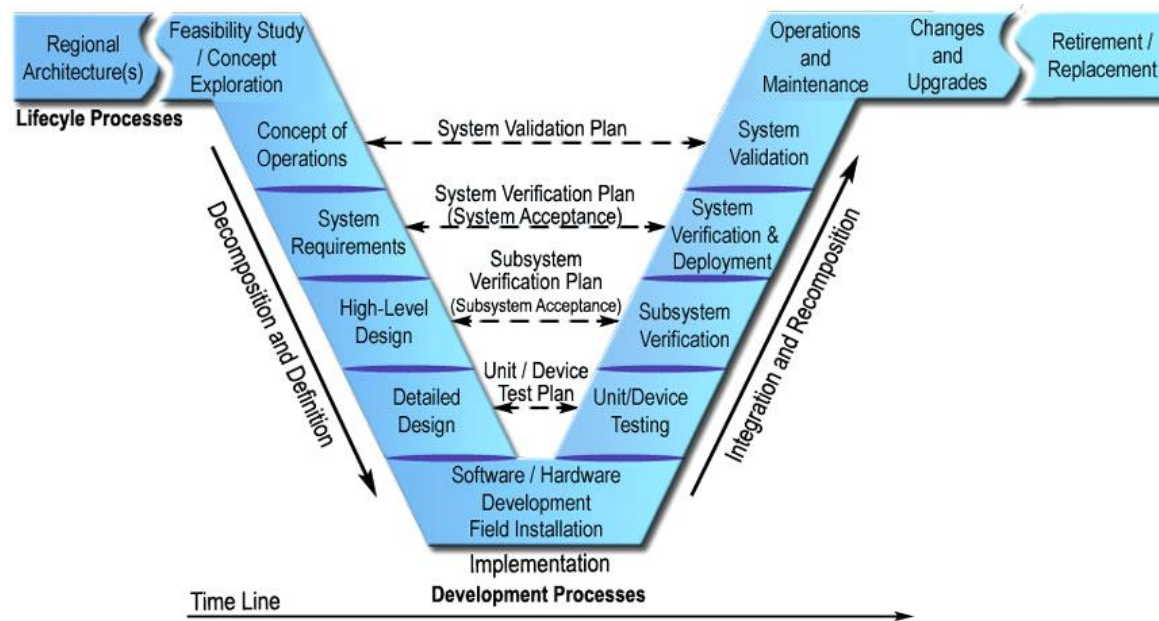


Agile Systems Engineering integrates the iterative and flexible nature of Agile methodologies with the structured and disciplined approach of Systems Engineering. It combines the benefits of both approaches to enhance system development

Vee Diagram Framework

The formalized application to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.

- INCOSE SE Vision 2020



Agile Systems Engineering Concepts

- **Agile Principles:** Agile methodologies, such as Scrum or Kanban, emphasize flexibility, iterative development, customer collaboration, and responding to change.
- **Iterative Development:** A development process, where requirements, design, development, and testing activities are performed incrementally in short cycles, known as sprints.
- **User Collaboration:** Agile Systems Engineering emphasizes close collaboration with users and stakeholders throughout the development process.
- **Cross-Functional Teams:** Agile Systems Engineering promotes the formation of cross-functional teams composed of members from different disciplines
- **Embracing Change:** Agile Systems Engineering recognizes that requirements and user needs may evolve over time.

Agile Challenges in Complex Embedded Systems (1 of 2)

- **Regulatory Compliance:** The aerospace industry is highly regulated, and projects must adhere to strict safety and quality standards
- **Large-Scale Projects:** Aerospace projects often involve large teams working on complex systems with long development cycles.
- **Hardware Development:** Unlike software development, aerospace projects involve the development of physical hardware that requires manufacturing, testing, and integration.
- **Safety Critical Systems:** Complex product often involve safety-critical components, such as aircraft control systems.

Agile Challenges in Complex Embedded Systems (2 of 2)

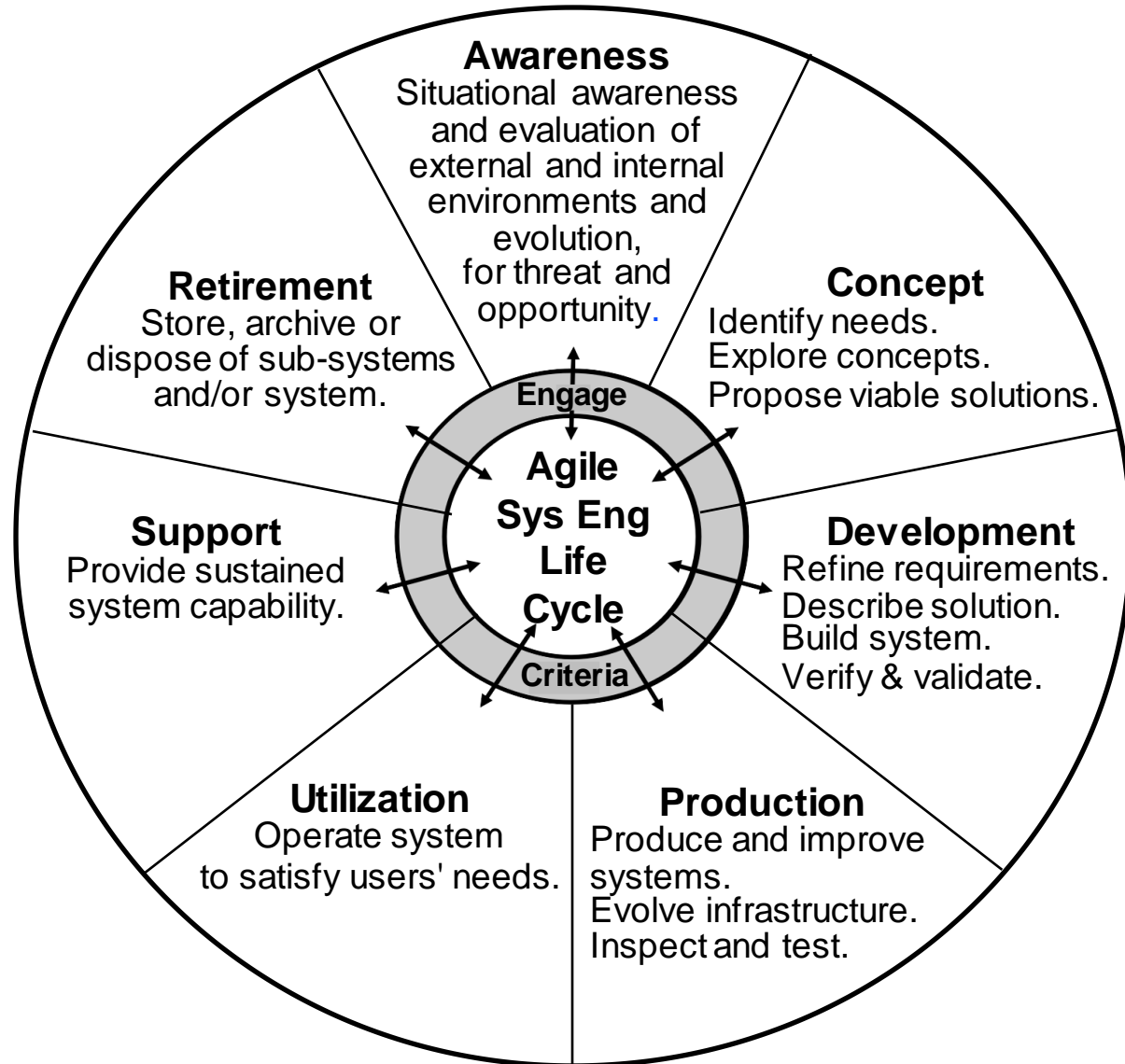
- **Legacy Systems and Processes:** The Aerospace Industry has a long history, and many organizations have well-established processes, legacy systems, and a traditional hierarchical structure.
- **Supplier and Customer Collaboration:** Aerospace projects often involve collaboration with multiple suppliers and customers, each with their own processes and priorities.
- **Security :** Most Aerospace companies work with Defense Agencies which require secured environment and protocols. This limits Agile collaboration and reuse
- **Complex Stakeholder Relationship:** Clients are represented by third parties

Agile Considerations in Complex Embedded Systems

- To overcome these challenges, implementing Agile requires a tailored approach that combines Agile principles with industry-specific adaptations.
- This involves careful planning, stakeholder education, cultural transformation, and a willingness to experiment and iterate on Agile practices to find the best fit for the specific needs of the aerospace projects.

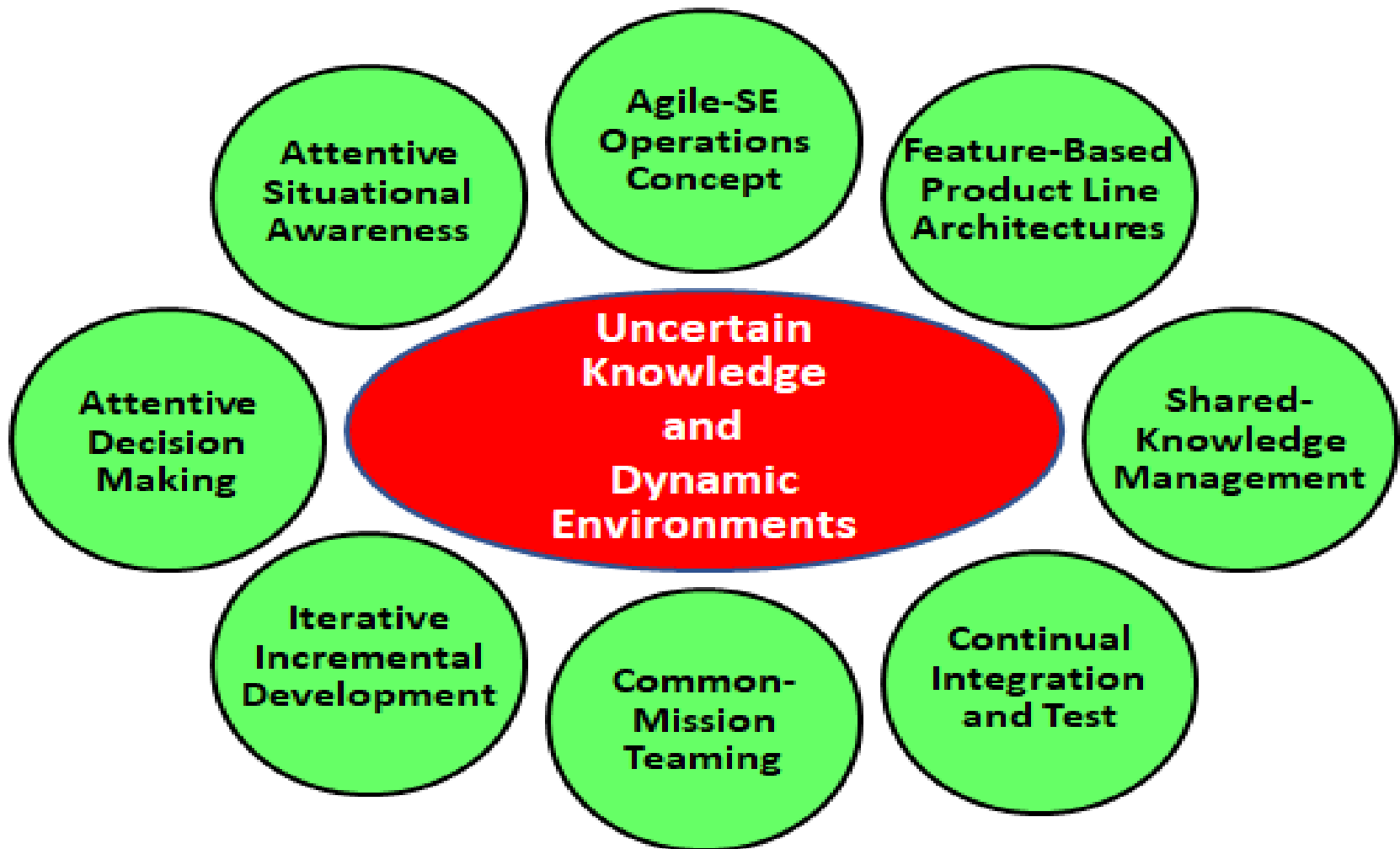


Asynchronous/ Simultaneous Agile Life-Cycle Framework



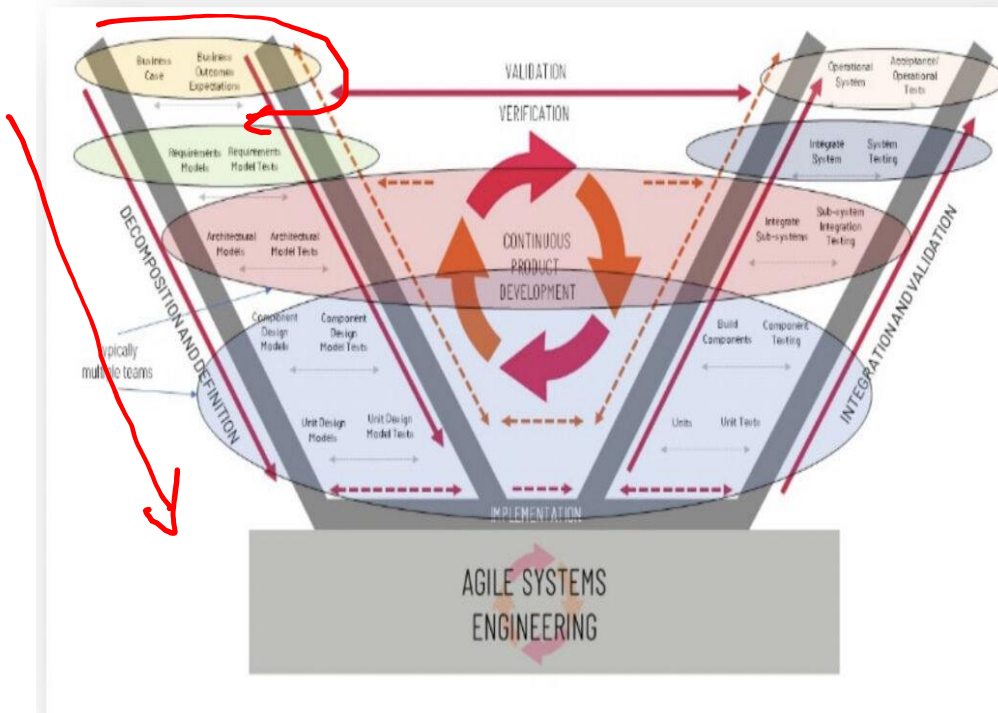
**Awareness Stage
is Critical Driver
of Agility**

Agile Systems Engineering Concepts



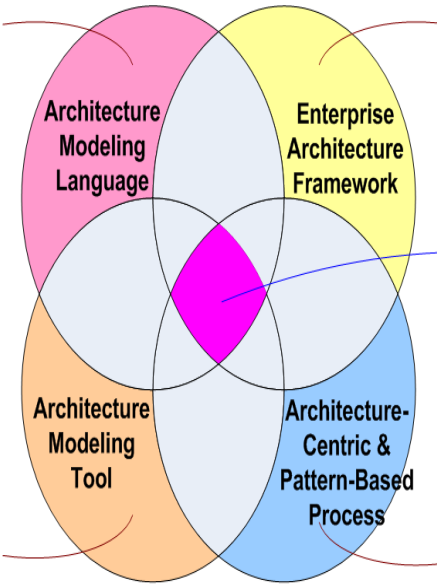
Individually each of these aspects can offer improved ability to deal with uncertain knowledge and dynamic environments

Agile Hardware and Agile Systems Engineering Techniques



Most mature, typically mix of SysML + UML 2 syntax & semantics w/ Tool *ModSim* & *Team-Modeling* support (see below)

Still relatively immature, Tool must enforce SysML/UML2 syntax & semantics (see above) + *ModSim* & *Team-Modeling* (e.g., *MagicDraw/Cameo*, *Sparx EA*, etc.)



Still relatively immature, EAF must support recursive *Patterns-of-Patterns* for *Systems-of-Systems* fractal problems (e.g., DoDAF, TOGAF, UAF)

maximum **System-of-Systems Architecture Model (SoSAM) effectiveness**

Least mature, most MBSE processes emphasize *Big Design Up Front* (BDUF) and are *Ad Hoc/Chaotic* (cf. CMMI Level 0) (contrast *Agile MBSE™* = {*Robust MBSE techniques*} ∩ {*Lean Engineering techniques*})

Agile MBSE™ System-of-Systems Architecture Tetrad

© 2003-2021 PivotPoint Technology Corp.

Agile Systems Engineering Techniques

Agile Development vs. Agile Systems

An agile *development process* is:

- Nimble, dexterous and swift
- Adaptive and response to new, sometimes unexpected, information that becomes available during product/system development
- Opposite the traditional belief that requirements and design solutions should be frozen as early as possible

Many agile project management techniques could be applied to hardware development

Many agile PM techniques could be applied to the hardware development process

An agile *system* is:

- Flexible, reconfigurable, extensible
- Scalable in the sense of capacity
- Flexible in terms of functions and performance levels (such systems can be modified after initial deployment by addition of modules or modification of performance levels)

Agile hardware systems require specific architectures and design principles

Agile systems require specific architectures and design principles

Agile Approach to Design and Iteration

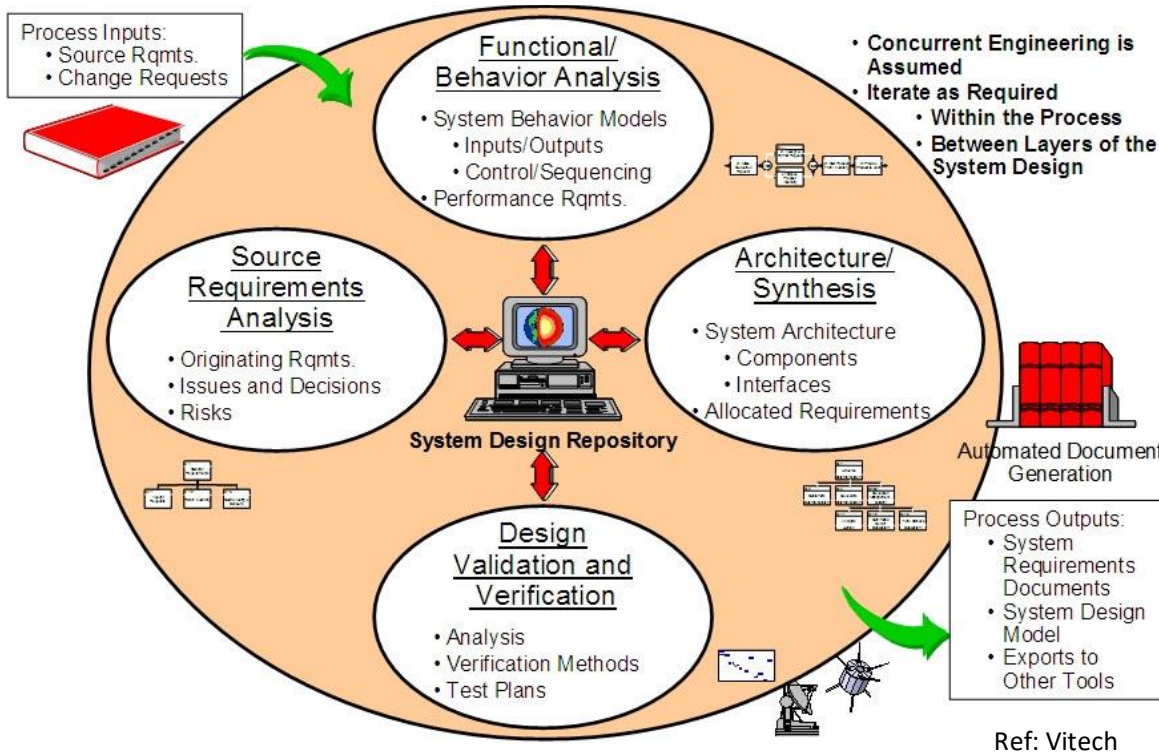
- Devote needed time up front on behavioral decomposition
- Truly understand the affect of key stakeholders influence on the system operation
- Refine the top-level use cases in system requirements
- It will be the basis for system validation that will demonstrate users intended use of the system



Cross-Cutting Relationships Between Model Elements

- ^{AS}ALLOCATE Behavior to Blocks (Structure)
- ^ESATISFY Requirement with Blocks
- REFINE use cases with requirements
- TRACE requirements with customer requirements
- VERIFY requirements with test cases

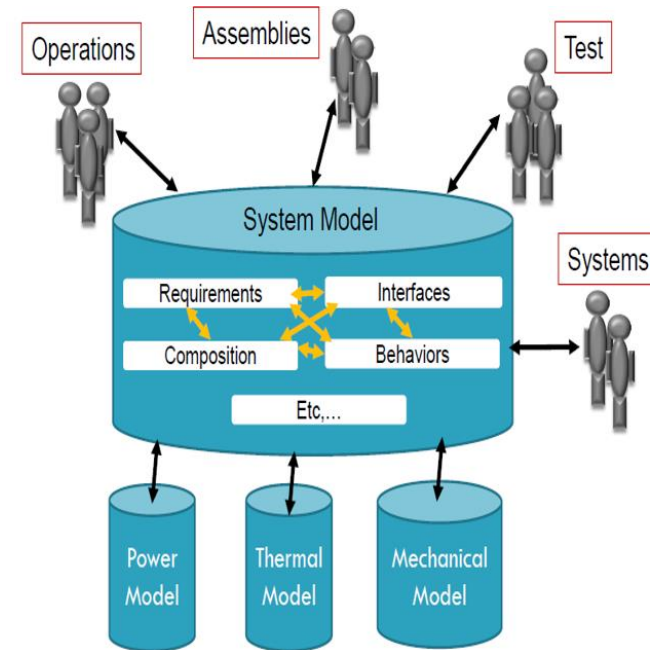
Model-Based Approach



Repository inputs/outputs

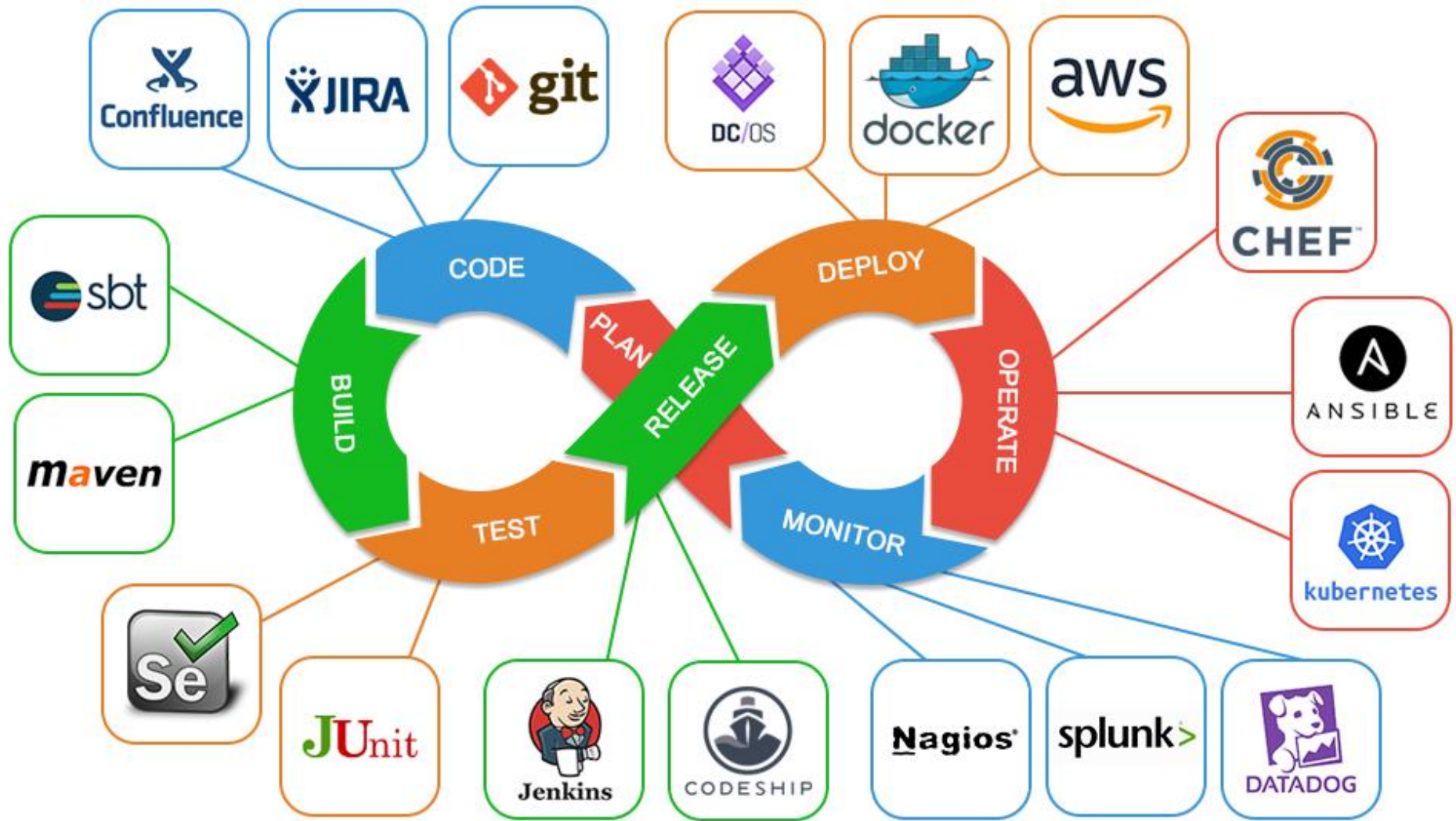
Single Design repository

Collaborative Design Environment



The right tool for the right job

DevOps Establish A Continuous Pipeline of Delivery



Lean Thinking

A. Quality:

Work is passed directly to next Process with no defects

B. Productivity:

Minimize wasted movement, warehouses, and double handling

C. Productivity:

Problems are identified and solved real time

D. Lead Time:

Shortest supply chain, highest flexibility to satisfy customer demand

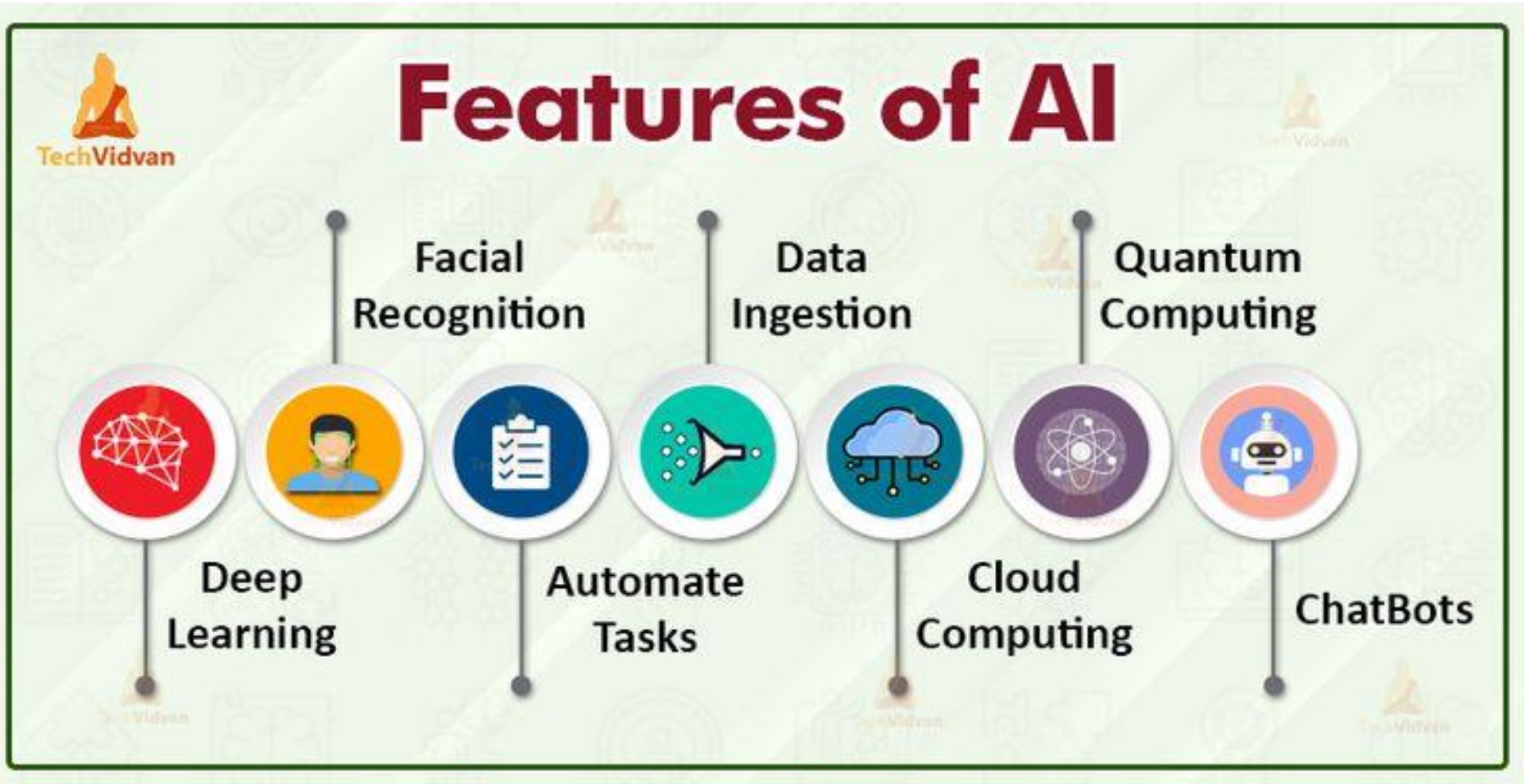
E. Team Member Morale:

Value of work is more visible, recognized

F. Cost:

Reduced Inventory Levels

Artificial Intelligence Enables Continuous Learning



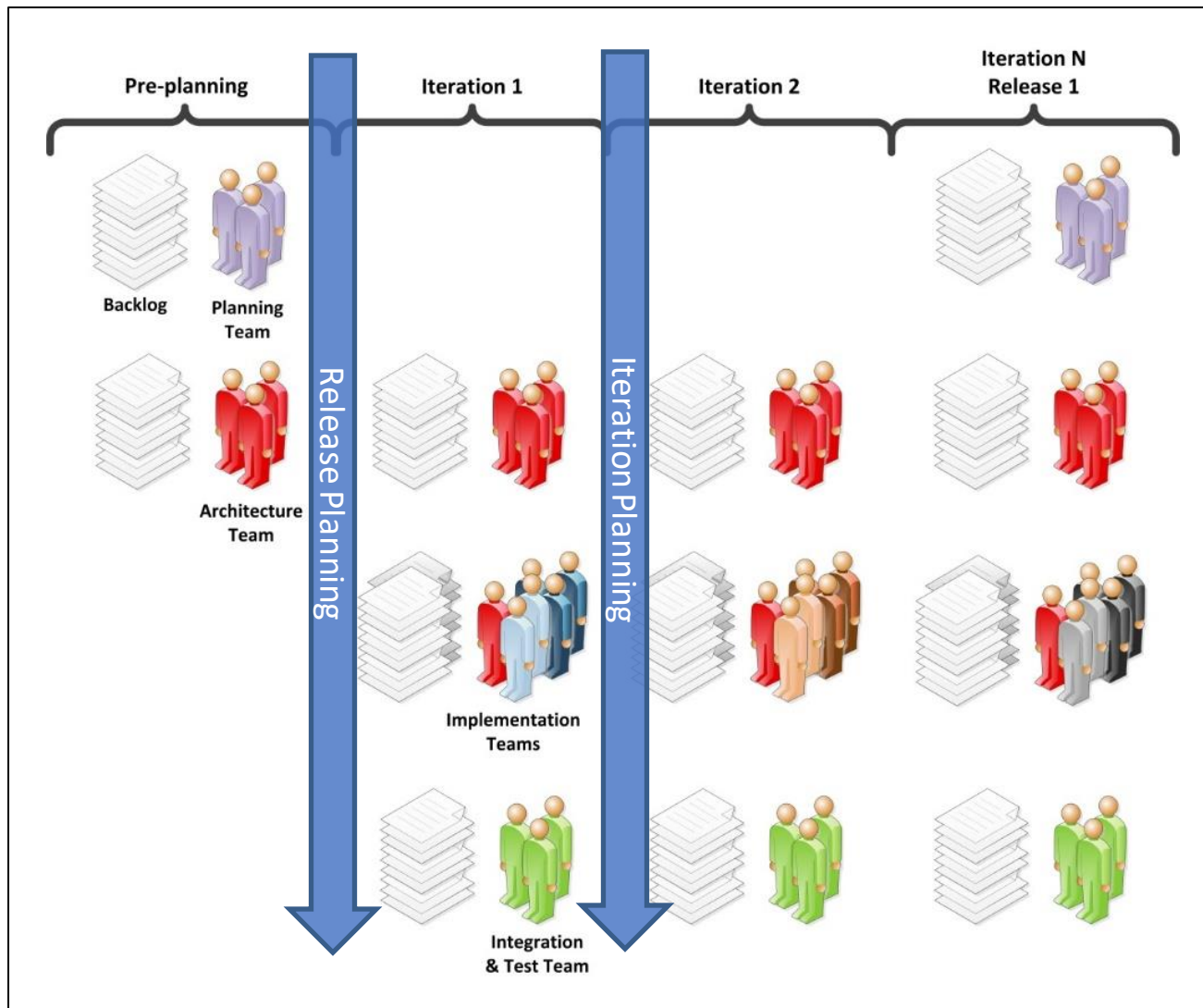
Big Data and Data Analytics

DATA
ANALYTICS



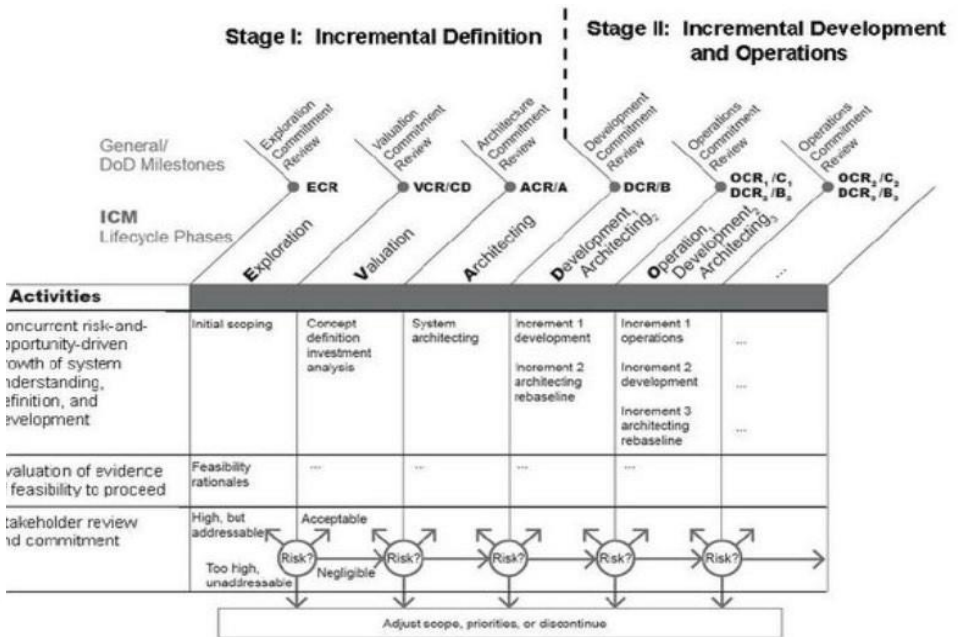
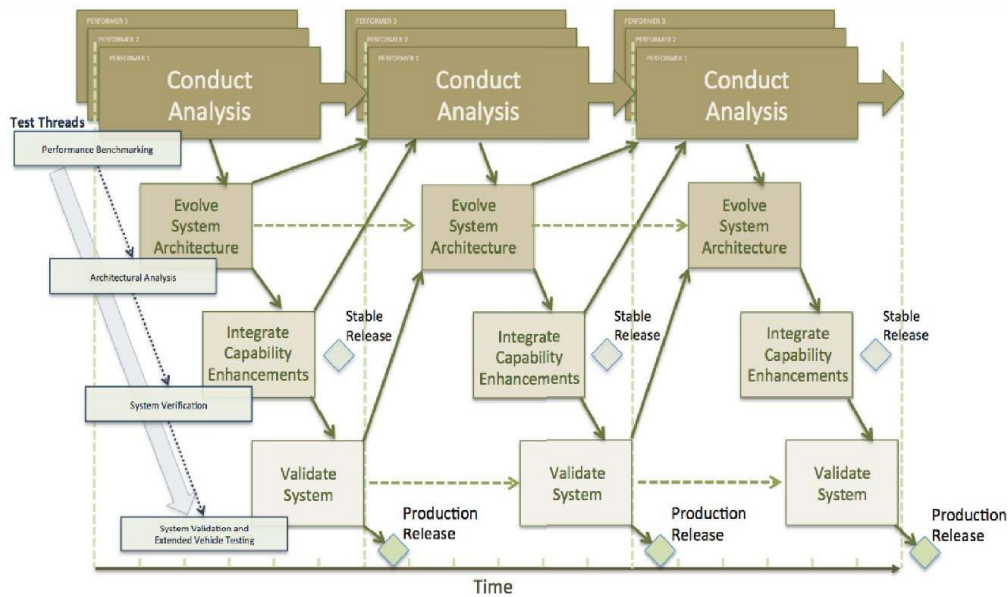
Agile Systems Engineering Aligned Development Frameworks

Agile Systems Engineering (SE) Framework



Copyright 2013, 2014 © by Larri Rosser, Phyllis Marbach, Gundars Osvalds, David Lempia.

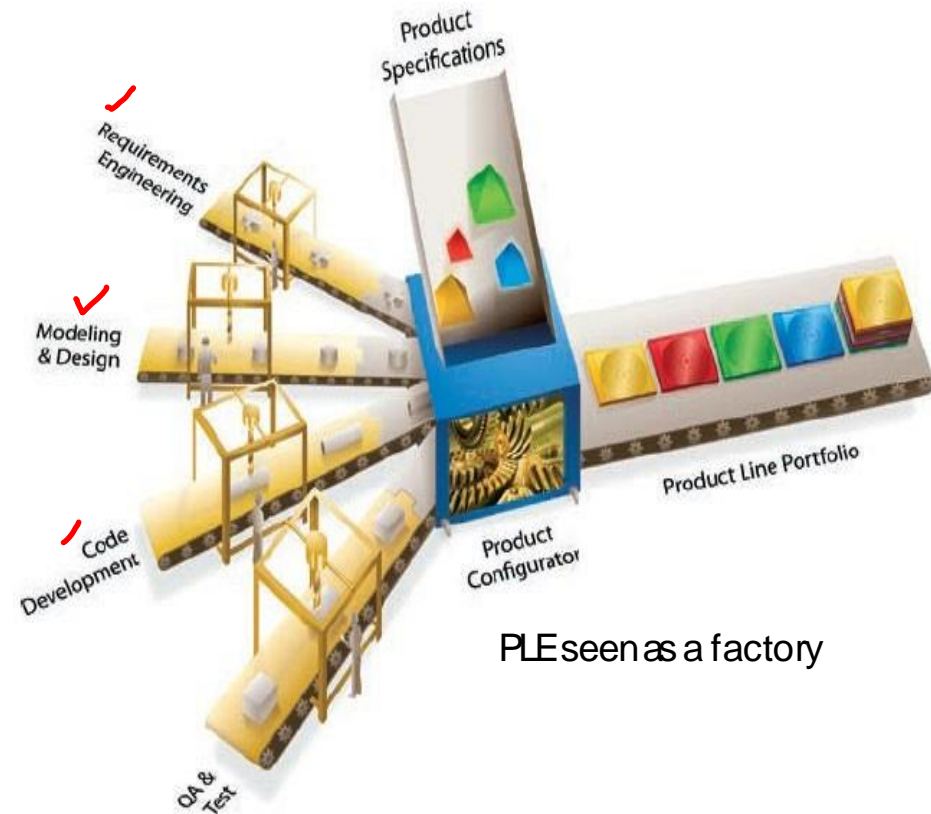
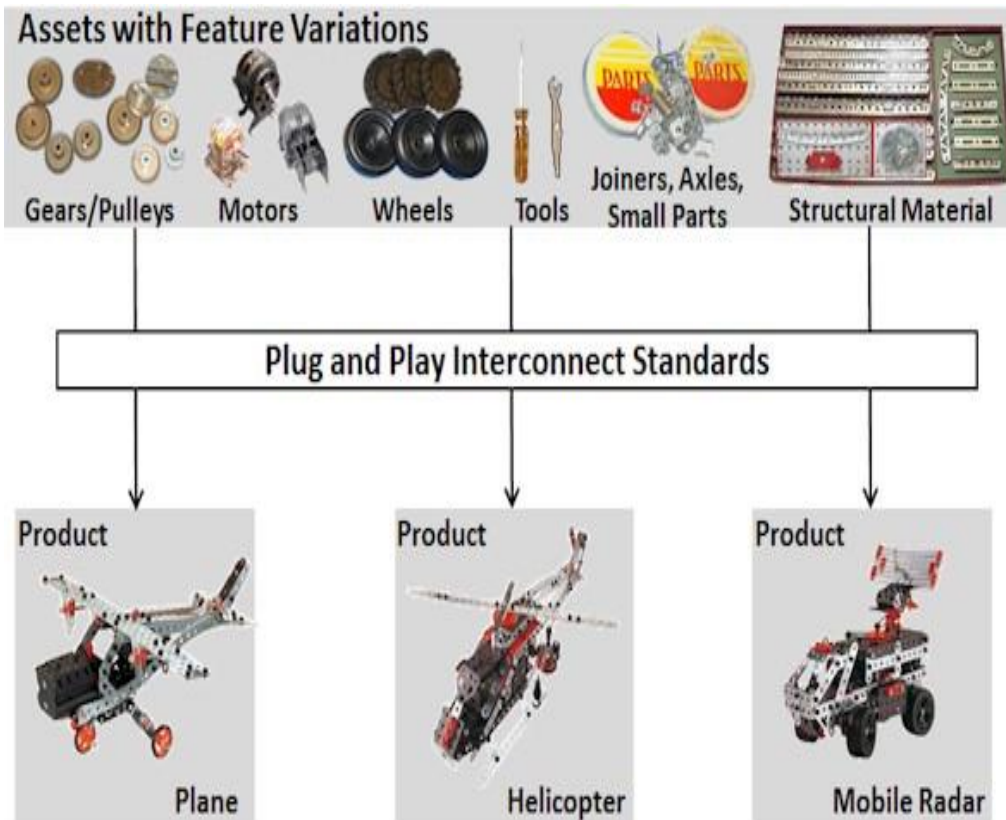
Incremental Commitment Spiral Model



'An implementer's view of the evolutionary systems engineering for autonomous unmanned systems' Chris Scrapper, Ryan Halterman, Judith Dahmann. IEEE Systems Conference, 2016

[Using the Incremental Commitment Model to Integrate System Acquisition, Systems Engineering, and Software Engineering.](#) Barry Boehm, Jo Ann Lane, CrossTalk, October 2007.

Feature-Based Product Line Architectures



Fundamentals of Agile Systems Engineering—Part 1. Rick Dove, Ralph LaBarge. NOOSE, IS14, 2014.

Product Line Engineering Comes to the Industrial Mainstream. Paul C. Clements. NOOSE, INSIGHT, August 2019, 22:2

Enterprises Level SAFe 6.0

SAFe 6.0

Select SAFe configuration

OVERVIEW ESSENTIAL LARGE SOLUTION PORTFOLIO FULL

BUSINESS AGILITY

Organizational Agility



Operational Value Streams

Lean Portfolio Management



Portfolio Flow



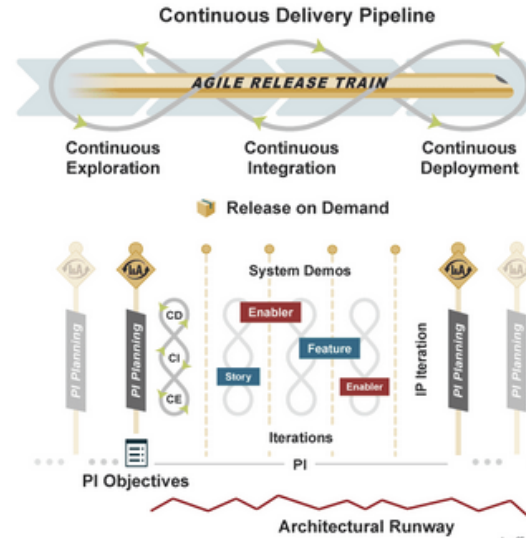
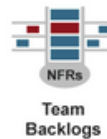
PORTFOLIO

ART Flow

Customer Centricity



Team Flow



ESSENTIAL



Vision

OKRs

Roadmap

AI

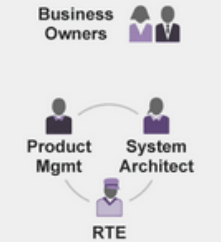
Shared Services

CoP

System Team

Measure & Grow

Agile Product Delivery



Team and Technical Agility



Business & Technology

Lean-Agile Leadership



Lean-Agile Mindset



Core Values



SAFe Principles



Implementation Roadmap

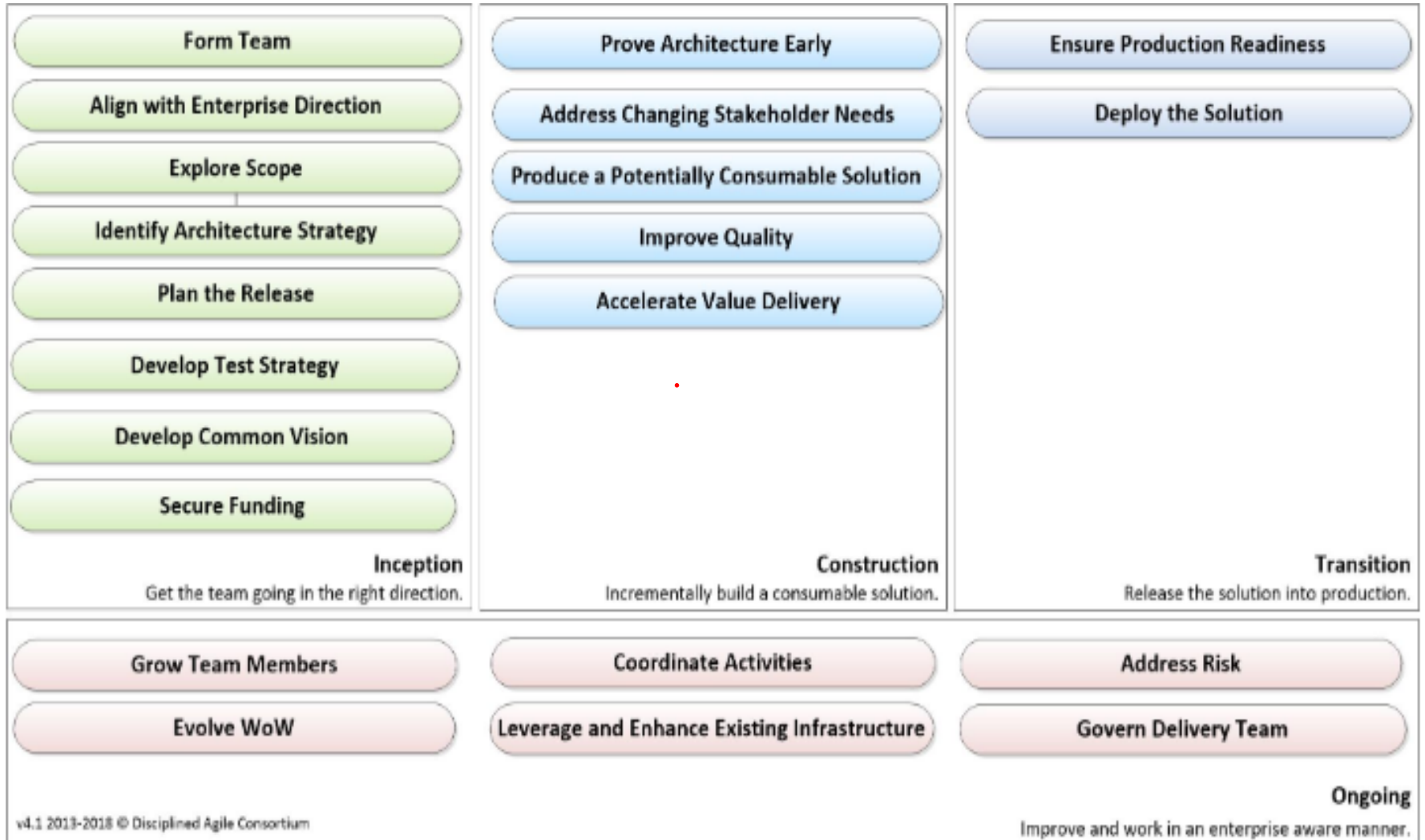


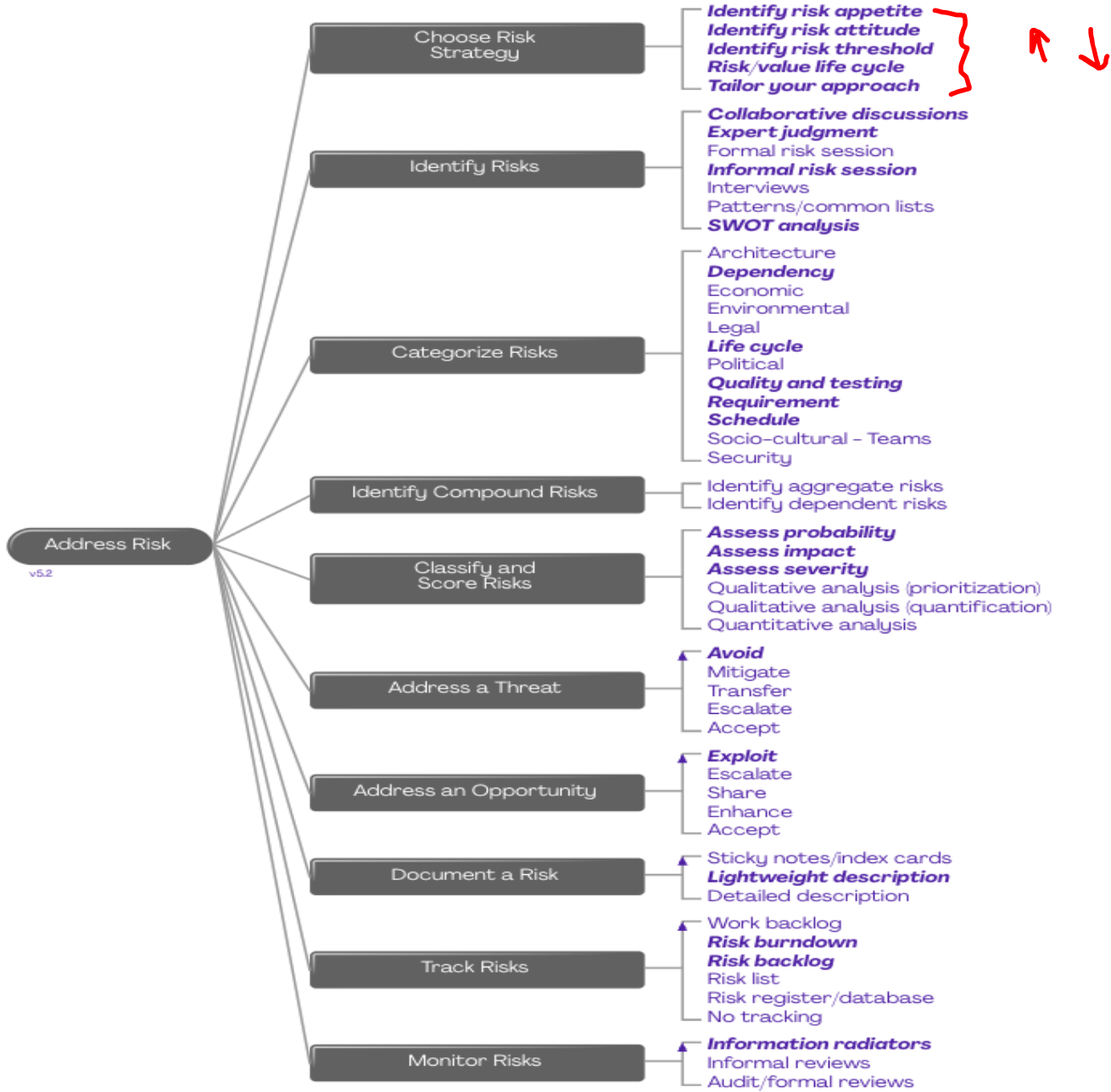
SPC



Continuous Learning Culture

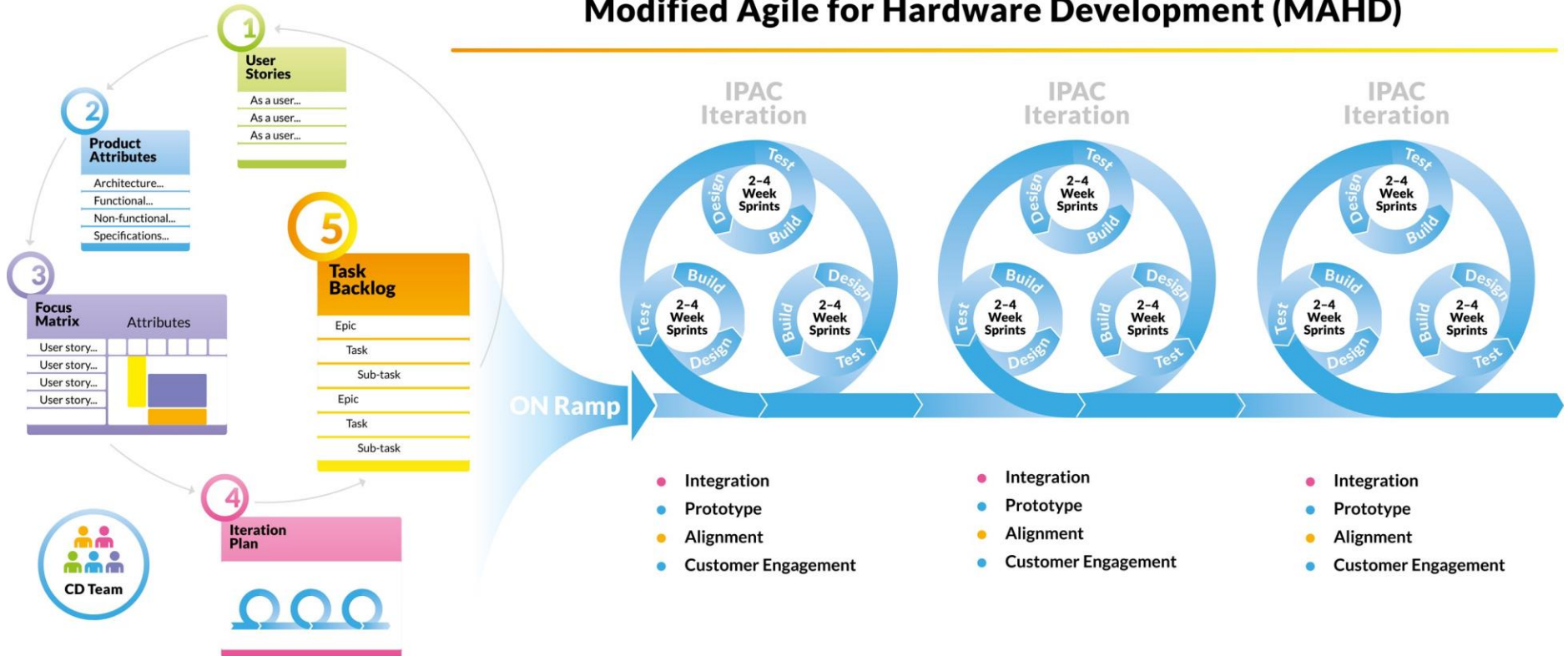
Disciplined Agile Capabilities





Agile Hardware and Agile Systems Engineering Techniques

Modified Agile for Hardware Development (MAHD)



Eight Tips on How to Master Agile Systems Engineering

- 1. Understand the Foundations of Systems Engineering:** Start by gaining a solid understanding of systems engineering principles and concepts. A strong foundation in traditional systems engineering will help you appreciate the nuances and improvements that Agile brings to the field.
- 2. Learn Agile Methodologies:** Familiarize yourself with Agile methodologies like Scrum, Kanban, and Lean. Each of these has its own set of practices and principles that can be applied to systems engineering.
- 3. Combine Agile and Systems Engineering Knowledge:** Recognize the synergy between Agile and systems engineering. Understand how Agile principles can complement and enhance traditional systems engineering processes.
- 4. Practice Incremental Development:** Embrace the idea of incremental development. Break down your project into smaller, manageable increments that can be developed, tested, and delivered iteratively. This approach allows for faster feedback and adaptability.
- 5. Focus on User-Centric Design:** Place a strong emphasis on understanding and meeting the needs of end-users. Engage stakeholders regularly to gather feedback and ensure that the product aligns with their expectations and requirements.
- 6. Implement Continuous Integration and Delivery (CI/CD):** Automate testing, integration, and deployment processes to achieve a continuous flow of work. This helps in reducing bottlenecks and ensuring that code is always in a deployable state.
- 7. Seek Mentorship and Training:** Consider seeking mentorship from experienced Agile Systems Engineering practitioners. Additionally, take advantage of training courses and certifications related to Agile and systems engineering
- 8. Just Do It !!**



Caltech CTME

We customize unique learning experiences for organizations and their people, working one-on-one with leadership to design and deliver practical programs and certificate courses for teams and individuals alike

Together with our clients, we help the enterprise cultivate high-performance teams, create innovative solutions, and commercialize technologies and ideas

Programs for Individuals

- Agile Project Management Certificate
- Agile Techniques for Project Managers
- Business Agility Foundation Certificate (New)

Programs for Organizations

- Customizable Agile-centric certificate programs and workshops
- Enterprise Business Agility Certificate (New)
- Enterprise Agility Workshops

Programs Tailored for Industries

- Aerospace & Defense
- Electronics
- Media
- Communications
- Energy/Utility
- Life Sciences and Medical Devices
- Government

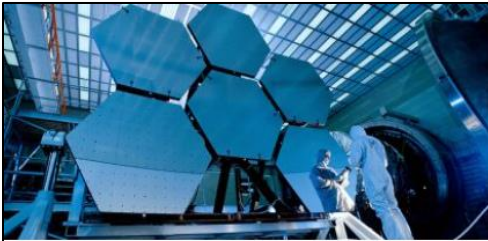
Caltech-Simplilearn Post Graduate Program in DevOps

- Comprehensive bootcamp-style program
 - Live, remote lectures, also available for on-demand viewing
 - Hands-on labs with tools
 - Optional 90-minute masterclasses once a month
- Refresher courses are optional; if a participant is aware of the basics of Agile and Linux they can start with the core learning path
- Registration at <https://ctme.caltech.edu/programs-for-individuals/software-engineering-open/devops-pgp>

	Course Name	Online (hours)	Live (hours)
Refresher courses	Linux Training	4	0
	Programming Refresher	1.5	0
	Agile Scrum Foundation	1.4	0
	Agile Scrum Master	2.9	0
Core Learning Path	DevOps Certification Training	8.5	40
	Git & Github Training	5	24
	CI/CD Pipelines with Jenkins	3.5	20
	Configuration Management with Chef, Puppet and Ansible	0	48
	Docker Certified Associate Training	6	24
	PG DO - Certified Kubernetes Administrator Training	10	40
	DevOps on AWS	8	0
	PG DO - DevOps Capstone Project	0	8
	Totals		50.8

See brochure for course details

Caltech MBSE Classes



2023 Courses



Model-Based Systems Engineering Fundamentals (MBSE) 22.0
Enroll in our live-online MBSE Fundamentals Certificate Program: Boost systems engineering...

NOV 04

LIVE-ONLINE CERTIFICATE 5 DAYS

LEARN MORE



Model-Based Systems Engineering Fundamentals (MBSE) 7
As you face the challenges of optimizing design and simulation activities, you need the toolbox of...

OCT 09

SELF-PACED ONLINE CERTIFICATE 6 WEEKS

LEARN MORE

Model-Based Systems Engineering Fundamentals (MBSE) 7

As you face the challenges of optimizing design and simulation activities, you need th...

6 WEEKS
SELF-PACED ONLINE
CERTIFICATE

OCT 09

LEARN MORE

	FMBSE	FMBSE	FMBSE	AMBSE	AMBSE	AMBSE
2024	LIVE	ON DEMAND	NIGHT	LIVE	ON DEMAND	NIGHT



What is new for 2024: Advanced Model Based Systems Engineering

michael.p.brenner@jpl.nasa.gov

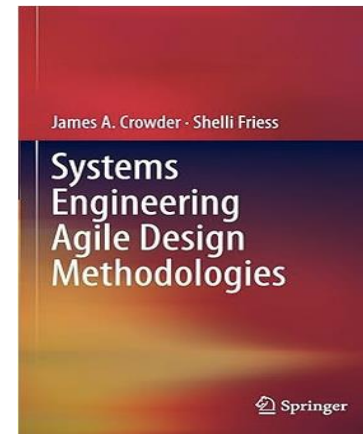
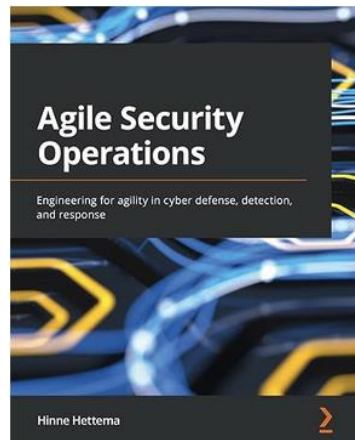
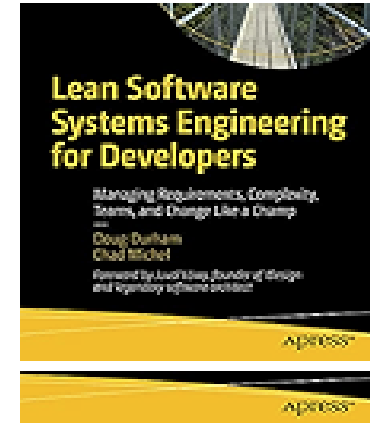
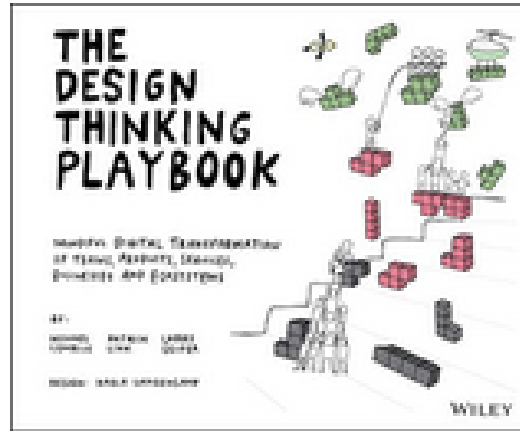
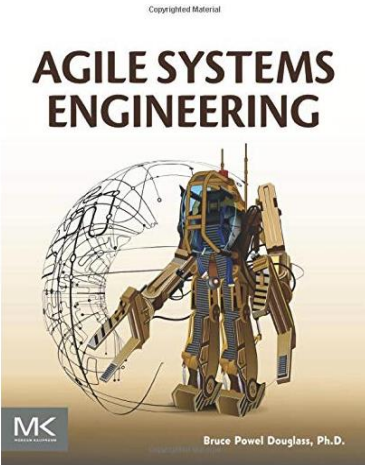
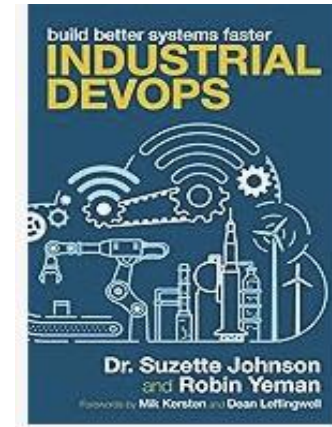
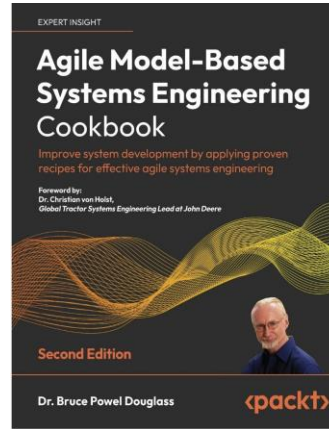
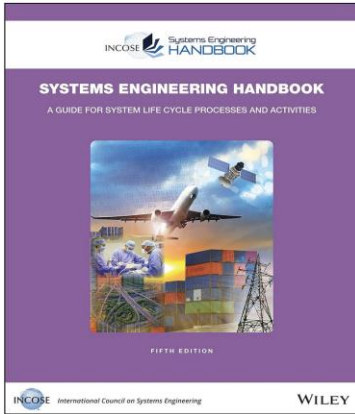
[CTME | Home \(caltech.edu\)](http://ctme.caltech.edu)

Class structure: Hands on with Magic System of System

Custom on-site courses are available

Brenner | MBSE

Agile Systems Engineering Books



Tips on How to Master Agile Systems Engineering

- **Regulatory Compliance:** The aerospace industry is highly regulated, and projects must adhere to strict safety and quality standards
- **Large-Scale Projects:** Aerospace projects often involve large teams working on complex systems with long development cycles.
- **Hardware Development:** Unlike software development, aerospace projects involve the development of physical hardware that requires manufacturing, testing, and integration.
- **Safety Critical Systems:** Complex product often involve safety-critical components, such as aircraft control systems.

Agile Systems Engineering Recap

- Agile Systems Engineering is an approach that combines Agile methodologies with Systems Engineering principles to enhance the development of complex systems in various industries, including aerospace.
- By combining the strengths of Agile methodologies with Systems Engineering practices, Agile Systems Engineering offers a flexible and iterative approach to developing complex systems in industries like aerospace. It enables teams to deliver high-quality systems that meet evolving user needs while maintaining a disciplined approach to system development and integration.
- Tailoring Agile Practices: Agile Systems Engineering adapts Agile practices to suit the needs of the systems engineering domain. For example, the use of scaled Agile frameworks like SAFe (Scaled Agile Framework) can help address the challenges of large-scale system development.



