U.S. ARMY

DEVCOM AVIATION & MISSILE CENTER

U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND AVIATION & MISSILE CENTER

Agile Development in the Department of Defense (DoD)

06 NOVEMBER 2024

JILLIAN IVEY | RELIABILITY ENGINEER
RAM ENGINEERING & SYSTEM ASSESSMENT DIVISION

WHY AGILE?

Mid-2000s – DoD acknowledged need to shift IT acquisition process to match the fast-paced commercial IT sector

- Reasons for shift[3] –
  - Software functionality demands of our systems are increasing
  - Keeping up with the rapid pace of technology change is needed
  - Warfighters are called upon to execute diverse & rapidly changing missions, and our SW needs to better support them

2010 NDAA – DoD proposed new acquisition approach for IT systems

- IT systems need to be designed to include[7] –
  - Early and continual involvement of the user
  - Multiple rapidly executed increments or releases of capability
  - Early, successive prototyping to support an evolutionary approach
  - A modular open-systems approach
- DoDI 5000.02 - published and supports tailoring or adoption of iterative development methods (i.e. Agile)

2018 & 2019 NDAA – Directed the use of Agile Pilots[2]

- Eight pilot programs were designated to adopt agile or iterative development methods
  - AIAMD selected as a Pilot
- DoDI 5000.87– establishes the use of the software acquisition pathway for custom SW capabilities developed for the DoD

March 2024 – Army Directive 2024-02 establishes policy & assigns responsibilities for adopting modern software practices Army wide

- Topics include – continuous integration/continuous delivery (CI/CD), agile, lean, and development, security & operations (DevSecOps)

AGILE BASICS

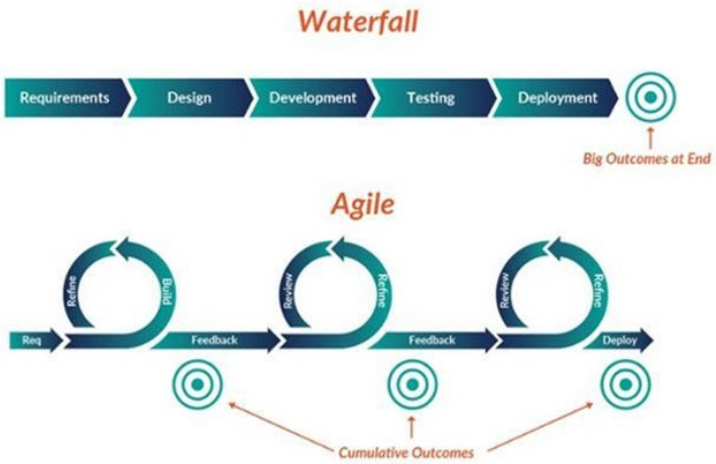
DEFINITION OF AGILE

OVERVIEW

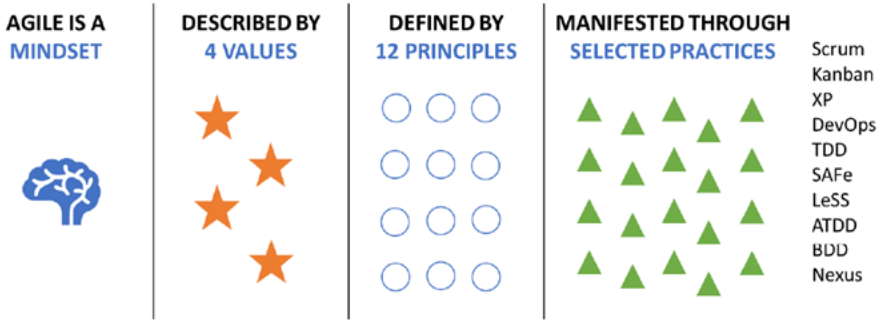
- Prioritizes flexibility, collaboration, and iterative progress **OVER** rigid planning and extensive documentation
  - True paradigm shift in the way system development is planned, executed, and structured[9]

ORIGINS & EVOLUTION

- Origins trace back to 1970s – firms began questioning effectiveness of traditional approaches
- In 2001, Agile Alliance published The Agile Manifesto – outlining the core values and principles of Agile development
  - **Agile Mindset emphasizes quickly delivering value to users while avoiding work that does not directly add user value**[10]
- Today, Agile Principles have been applied to other fields– project management, marketing, and product development



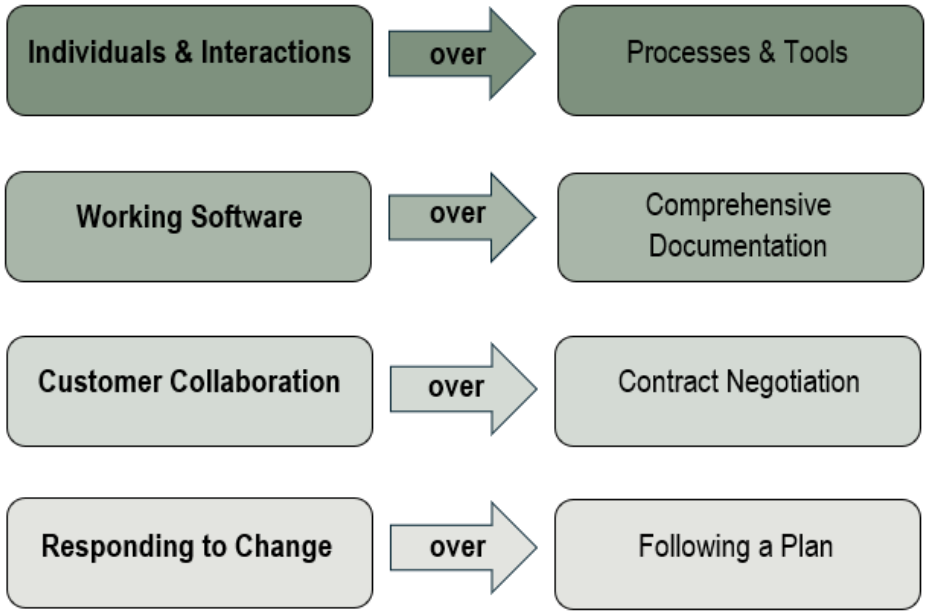
Waterfall vs. Agile Software Development [23]



Agile Mindset [21]

AGILE BASICS

THE MANIFESTO



Four Values of Agile

Twelve Principles

- Detailed guidelines for implementing & practicing the Four Values
- Agile Principles can be traced back to Lean Manufacturing Principles
  - — Lean aims to create value with fewer resources by optimizing workflow, reducing waste, and improving overall efficiency

The Agile Manifesto outlines guidance for Agile development known as the Four Values and Twelve Principles

Four Values

- Creates environment that is collaborative, flexible, and focused on delivering real value

01	Our **highest priority is to satisfy the customer** through the **early and continuous delivery** of valuable software.	02	**Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.	03	**Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
04	**Businesspeople and developers must work together daily** throughout the project.	05	Build projects around **motivated individuals**. Give them the environment and support they need and trust them to get the job done.	06	The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
07	**Working software** is the primary measure of progress.	08	Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to **maintain a constant pace indefinitely**.	09	Continuous attention to **technical excellence** and **good design** enhances agility.
10	**Simplicity** – the art of maximizing the amount of work not done–is essential.	11	The best architectures, requirements, and designs emerge from **self-organizing teams.**	12	At regular intervals, the team reflects on how to become more effective, then **tunes and adjusts** its behavior accordingly.

Twelve Principles of Agile

AGILE BASICS

COMMON ROLES

Development Team

- Small group with all the skills to do the work
- Self organized and manages own work within Sprint
- Direct Ownership of their Tasks
- Focuses on steady delivery of high-quality work

Product Owner

- "Voice of the Customer"
- Owns the Product Vision & Roadmap
- Maps the Customer Journey
- Ensures Product Backlog is prioritized, refined, and delivered
- Plans Sprints, Product Iteration content and releases

Scrum Master

- Facilitates the Scrum process
- Ensures Team is fully functional & productive
- Identifies and removes obstacles blocking Team's progress
- Looks to enhance productivity
- Organizes various Scrum Events

Tech Lead

- Bridges gap between technical architecture & detailed design
- Advocates for technical quality (code reviews, performance testing, and automated testing)
- Defends technical integrity

DoD Programs have additional roles and responsibilities than those listed above.

- Notable additions – Architecture Owner, DevSecOps Engineer/Lead, Independent Test Teams, Systems Engineer, Project & Program Managers

AGILE BASICS

COMMON FRAMEWORKS

	Scrum	Kanban	XP	Lean
Emphasis	Team collaboration, iterative development, continuous improvement	Flow, continuous delivery, minimizing work in progress	Technical excellence, customer involvement, rapid feedback	Elimination of waste, value stream mapping, continuous improvement
Best For	Projects with well-defined requirements & predictable development process	Team working on complex projects w/ changing requirements & unpredictable workflows	Projects with rapidly changing requirements & a focus on quality	Projects with a focus on efficiency & reducing waste
Structure	Highly structured, with clear roles, events, and artifacts	Visual framework with a focus on visualization and limiting work in progress	Focus on software engineering practices: paired programming, continuous integration, test-driven development	Customer-centric approach and the creation of value through the removal of non-value-added tasks

Agile Frameworks Comparison [19]

AGILE BASICS

COMMON PRACTICES & TOOLS

Roadmap

- Highly flexible & adaptable visualization of strategic objectives to be achieved over time
- Defined at a high level
- Evolve based on external conditions, feedback, and priority

Backlogs

- Dynamic list of prioritized work items
  - Most basic work item: User Stories
- Can be used at the program, product, release, and/or sprint level
- Types – Product Backlog, Increment/Release Backlog, and Sprint Backlog

Boards

- Visualization of the flow of work & tracks ownership of work items
- Shows work prioritized by PO, work in progress, and work completed
- May also show blocked work and work dependencies

Test-Driven Development (TDD)

- Automated Test are written before code
- Ensures code meets requirements & functions correctly

Pair Programming

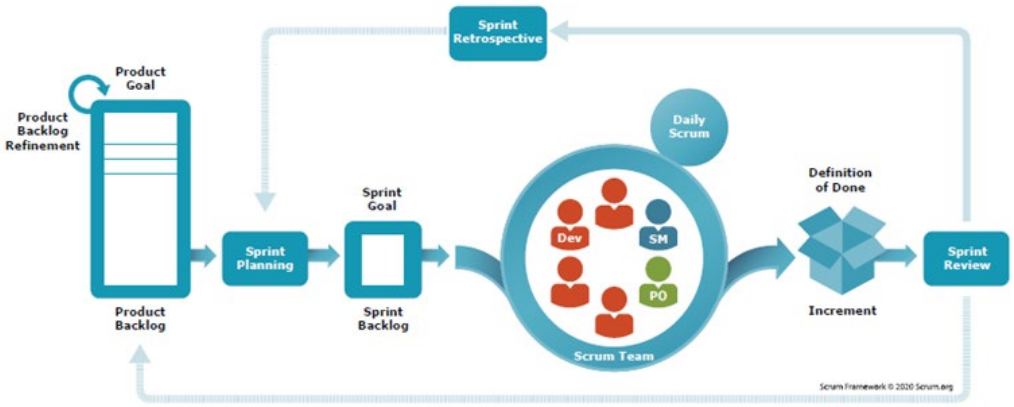
- Two programmers work together
  1. Writes code
  2. Reviews & provides feedback

AGILE BASICS

DEVELOPMENT CYCLE & SCRUM EVENTS

Iterative Development

- Sprint – time-boxed period for a team to complete a set of work items known as increments (typically 2-4 weeks)
- Release – usable increments delivered at regular intervals to the users/stakeholders (typically 3-5 Sprints)



Scrum Framework [26] - © ***Scrum.org***.

Sequence	Ceremony	Frequency	Timing	Participants	Goals
1	Backlog Grooming	Every 2 weeks (60 minutes)	Just before Kanban/Sprint Planning Meeting	PO, SM, All Team Members	Add, Edit, Delete Issues in the Product Backlog Prioritize "Backlog" Issues to 'To-Do' Estimate the complexity of prioritized items
2	Sprint Planning	Every 2 weeks (60 minutes)	Just before iteration/sprint starts	PO, SM, All Team Members	Determine work that can be completed in the upcoming 2-week work cycle and add to Sprint Backlog
3	Standups	Daily (15 minutes)	During iteration/sprint	SM, All Team Members	SM asks each team member: • What issue(s) did you complete since we last met? • What issue(s) are you currently working on? • Is there anything blocking you?
4	Sprint Demo	Every 2 Weeks (60 minutes)	At end of iteration/sprint	PO, SM, All Team Members, Stakeholder Reps	Team members demo completed work and discuss the increment PO and Stakeholders ask questions, give feedback Work is accepted/rejected by PO
5	Retrospective	Every 2 weeks (60 minutes)	After Sprint Demo	PO, SM, All Team Members	Continuous improvement to enhance team performance(what team should continue doing, stop doing/fix, and what we haven't tried that we should experiment with)

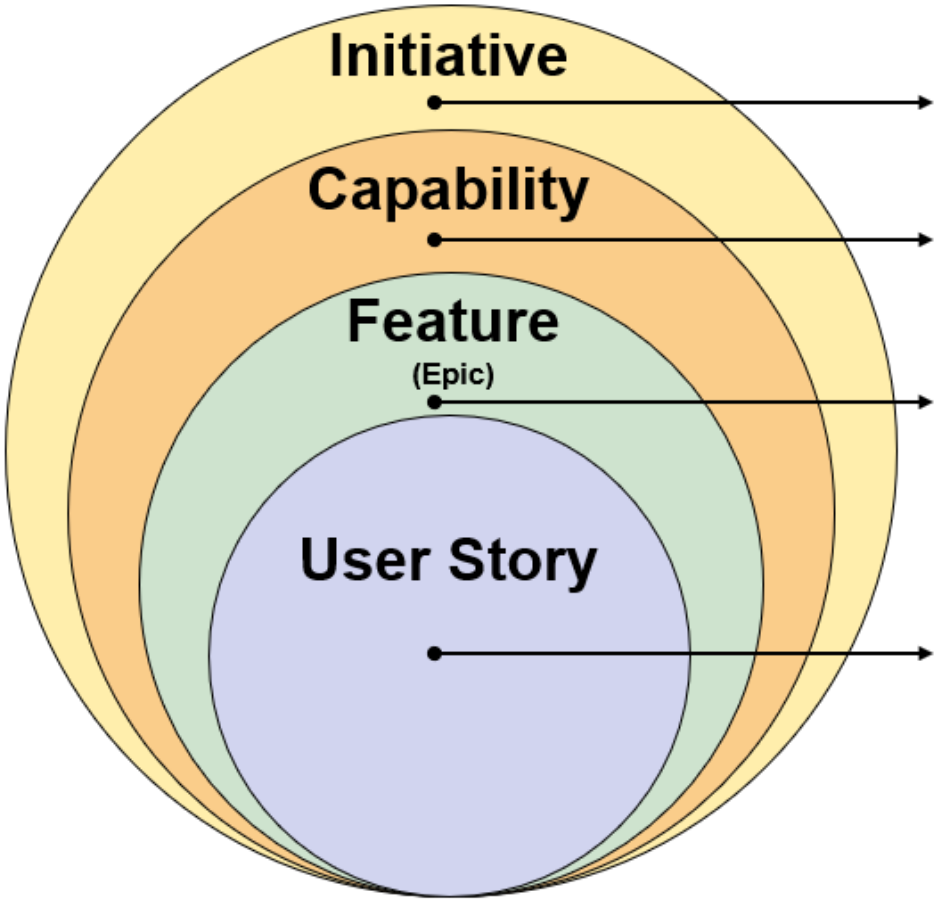
Scrum Ceremonies [22]

AGILE BASICS

PRODUCT BACKLOG

The Backlog is a prioritized list of all work items that need to be completed for a project (i.e. requirement definitions, enhancements, bug fixes, etc.)

Backlog Hierarchy:



Very high-level, large-scaled issues – mission needs

→ Typical Duration: Multiple Releases (2-5 Years)

Ex. Improve Customer Engagement

Better understood, large-scaled issues – express business functionality or identify constraints

→ Typical Duration: Multiple Releases (w/in 1 Year)

Ex. Implement Personalized Recommendations

Issues scoped down to workable deliverables – major functional areas that the product needs to deliver

→ Typical Duration: Single Release (3-6 months)

Ex. Personalization Settings for Recommendations

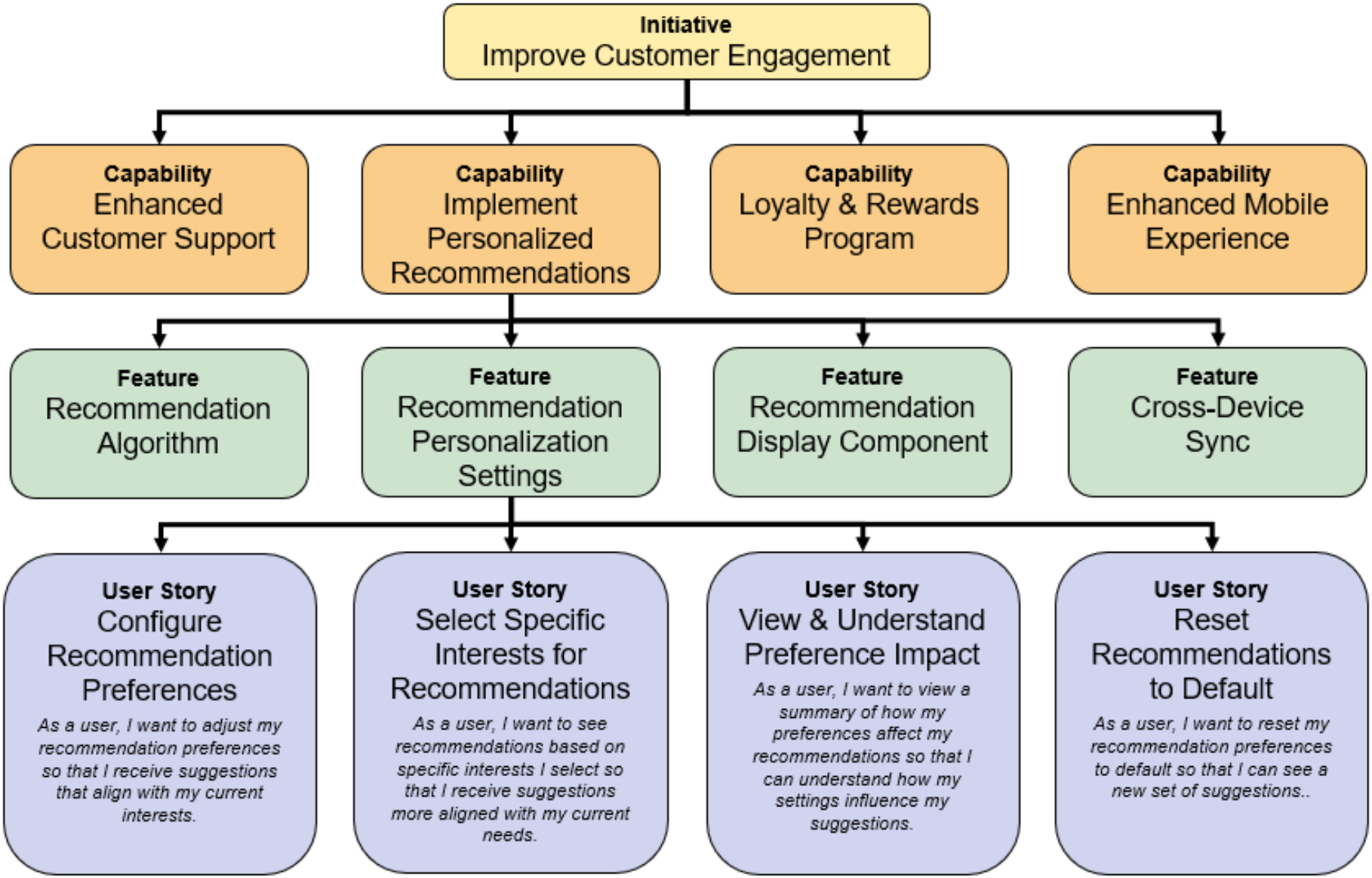
Definition of a requirement or functionality from an end-user perspective – captures the 'what' & 'why' but not the "how"

→ Typical Duration: Single Sprint (2-4 weeks)

Ex. As a user, I want to adjust my recommendation preferences, so I receive suggestions that align with my interests

AGILE BASICS

WORK BREAKDOWN



AGILE BASICS
ROADMAP

Q1	Q2	Q3	Q4
Improve Customer Engagement			
Implement Personalized Recommendations		**Enhanced Customer Support**	
Release 1	**Release 2**	**Release 3**	**Release 4**
○ **Cross Device Sync** Synch Rec. Across Devices Consistency Switching Devices	○ **Recommendation Display Component** Design Display to Visual Guidelines Ensure Usable Interactive Elements Optimize Display for Mobile Devices	○ **Live Chat Integration** Provide Chat Support Information Integrate Chat Support Real-Time Live Chat Support	○ **AI-Powered Chatbots** Chatbot for Common Queries Option from Chatbot to Human Support
○ **Personalization Settings** Configure Rec.Preferences Select Interests for Rec. View Preference Impact Reset Rec. to Default	○ **Recommendation Algorithm** Suggest Products from Purchase History Algorithm to Prioritize Categories Include New Arrivals in Rec.	○ **Automated Ticketing System** Convert Suport Request into Tickets Priority Levels for Support Tickets	○ **Comprehensive Help Center** Search Function to Help Center Expand Help Center with Articles Interactive Troubleshooting Guides
		○ **Multi-Channel Support Integration** Single Ticketing System Consistent Responses	

AGILE BASICS

ACCEPTANCE CRITERIA & DEFINITION OF DONE

Acceptance Criteria

- Detailed specifications, unique to each work item, that must be adhered for the client(s) to accept as complete
  - User's perspective, relatively stable throughout development

Objectives

- Ensures everyone has a common understanding of problem
- Used to clarify what should be built and when work is complete
- Help verify work via automated tests

Definition of Done (DoD)

- Global list of requirements, applicable to all work items, that must be adhered for the team to accept as complete
  - Developer's perspective, evolve throughout development

Objectives

- Builds common understanding w/in Team about Quality & Completeness
- Used as a checklist that PBI's are checked against
- Sets the standard for what "done" means

User Story: *As a user, I want to adjust my recommendation preferences, so I receive suggestions that align with my interests.*

Acceptance Criteria:

- ❑ User can access a "Preferences" section within their account settings.
- ❑ Users can select/deselect specific categories to customize their recommendations.
- ❑ There is a "Save" button that allows users to save their adjustments.
- ❑ Users receive updated recommendations based on their selections.

Definition of Done Checklist:

- ❑ Code peer review?
- ❑ Code completed w/out error?
- ❑ Unit Testing?
- ❑ Code Review?
- ❑ Localization & Translation?
- ❑ Localization Testing passed?
- ❑ Regression Testing?
- ❑ Automation Tests written and passed?
- ❑ Acceptance Criteria met?
- ❑ Signed off by Product Owner?

AGILE BASICS
METRICS

- Core Agile metrics can be broken down into the following types:
  - **Process Metrics**: relate efficiency of processes
  - **Quality Metrics**: relate quality of work preformed
  - **Product Metrics**: relate the delivery value to customers
  - **Cost Metrics**: relate Agile cost measures
  - **Value Assessment Metrics**: evaluates the impact of work
  - **Flow Metrics:** relates efficiency and effectiveness of the *flow of work* to users
- Common metrics are listed below, more in-depth information can be found in the DoD Agile Metrics Guide

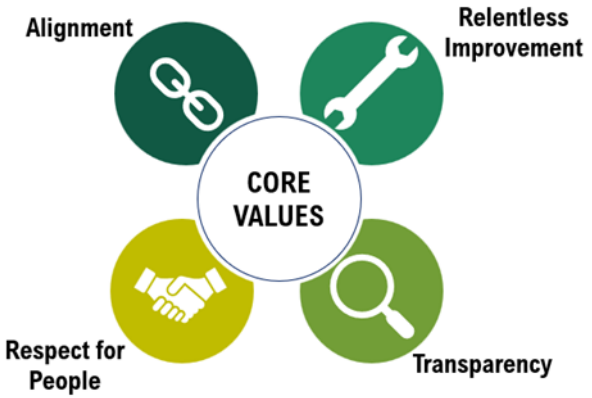
Story Points	→ Measures the relative effort or complexity of a user story within a sprint → Determined through team discussion rather than absolute time estimates
Velocity	→ Measures the amount of work (usually in story points) completed by a team in a sprint → ***Capacity:*** (velocity variation) is the total number of story points that can be completed per sprint, based on previous team velocity metrics
Burndown Chart	→ Visualization of the remaining work (usually in story points or tasks) vs. time within sprint
Burnup Chart	→ Tracks the cumulative completed work vs. time ▪ Helps visualize progress towards completing the planned scope
Test Coverage	→ Measures the proportion of code or requirements covered by automated tests ▪ Provides insight into the level of testing integrated w/in the end-to-end development process
Lead Time	→ Measures the total time taken from when a solution is initiated until it's completion & delivery

SCALED AGILE FRAMEWORK (SAFe)
OVERVIEW

Designed to implement lean-agile thinking & patterns of improved flow at an enterprise level to deliver value to customers at scale

- Accounts for different parts of an organization and the interdependencies within

	Traditional	SAFe
Scale	Small/medium-sized teams, focusing on individual workflow	Large enterprises with multiple teams working on interconnected projects
Structure	Individual teams, focused on self-organization, iterative development & continuous delivery	Structure added with configuration levels with specific roles, activities & artifacts ensuring alignment
Planning & Cadence	Short Term Planning: Sprints [1-4 weeks]	Longer Planning Cycle: Program Increments (PI) [8-12 weeks]
Metrics	Captured at Team Level (velocity, burndown & story points, etc.)	Captured at All Levels (business value metrics, lean management metrics, solution metrics, etc.)
Value Delivery	Delivered incrementally through user stories & sprint goals	Delivered via Value Streams – aligning objectives with execution across multiple teams & PIs



Key Concepts

Agile Release Train (ART)

- Long-lived, cross-functional, team of Agile teams that have all the capabilities – software, hardware, firmware, useability & others – needed to define, implement, test, evaluate, deploy, release, and if appropriate operate solutions

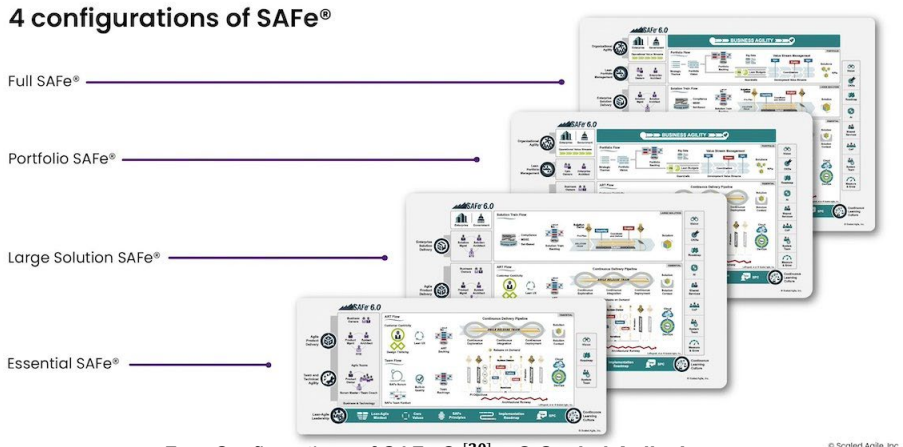
Value Streams

- Represent the series of steps that an organization uses to build products & solutions that provide a continuous flow of value to a customer
  1. Operational Value Streams (OVS) – sequence of activities needed to deliver a product or service to users/stakeholders
  2. Development Value Streams (DVS) – sequence of activities to develop & support solutions used by OVS

Approved for Public Release

SAFe
CONFIGURATION LEVELS

Specifically designed to accommodate different levels of agile development, acknowledging differing level of organizational complexities



*Four Configurations of SAFe ® [20] - © **Scaled Agile, Inc**.*

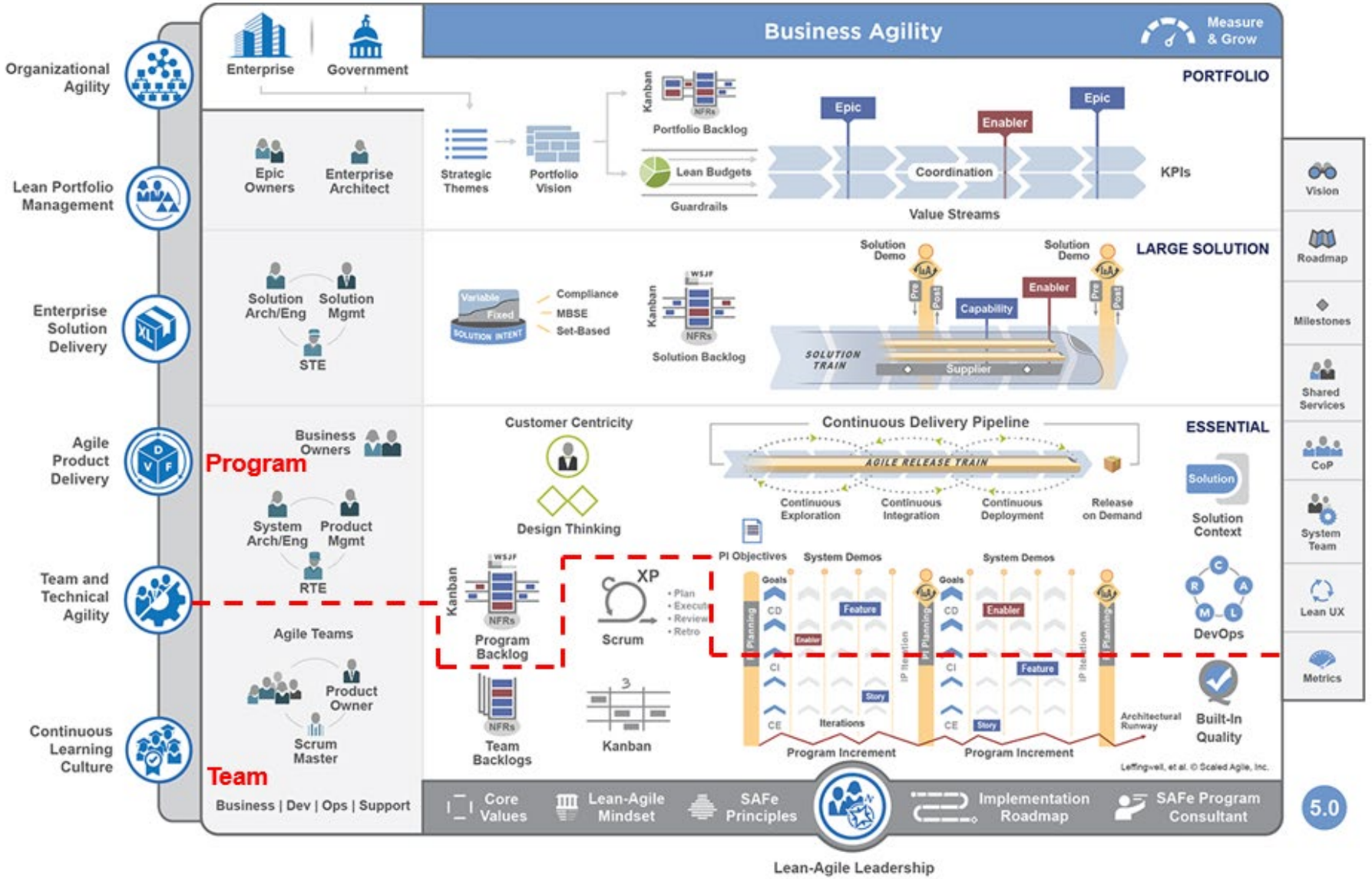
Essential SAFe *(Team & Program Levels)*
- Most basic, offers foundational elements of SAFe
- Establishes the fundamentals of the Lean-Agile Leadership, Team & Technical Agility, and Agile Product Delivery competencies
- Introduces ART constructs w.r.t. continuous flow of value

Large Solution SAFe *(Team, Program, & Large Solution Levels)*
- Extension of Essential, introduces Enterprise Solution Delivery competency
- For building large/complex systems requiring coordination between multiple ARTs, Suppliers, and any future ARTS

Portfolio SAFe *(Team, Program, Large Solution & Portfolio Levels)*
- Extension of Large Solution SAFe, used when managing portfolios of independent products and systems from a resourcing, investment and business value perspective
- Lean Portfolio Management competency aligns execution to strategy & organizes development and organizes development through multiple value streams

Full SAFe *(Team, Program, Large Solution & Portfolio Levels)*
- Combines all configurations, and includes all seven core competencies
- Providing a complete framework addressing the needs of all levels of the organization (teams to portfolio)

SAFe
LEVEL BREAKDOWN



SAFe version 5.0 – Full SAFe Configuration [25] *- © **Scaled Agile, Inc**.*

SAFe
TEAM LEVEL KEY ASPECTS

ROLES

Product Owner (PO)
Responsible for defining User Stories & prioritizing the *Team Backlog*

Maintains integrity of the team's work while ensuring execution of program priorities

Scrum Master (SM)
Educates team on Agile – ensures agreed Agile process is implemented

Removes impediments & fosters environment for high-performing dynamics, continuous flow & relentless improvement

Team Members
Developers, testers, designers, etc., who collaborate to deliver increments of working software

EVENTS

Daily Standup
Daily meetings where team members synchronize activities and plan work for the day

Sprint Planning
Team plans the work for the upcoming Sprint aligning with PO prioritization items

Sprint Review
Team demonstrates the work completed during the sprint & update *Team Backlog* for subsequent development

PO reviews key objectives completed & inspects the software for product for usability

Sprint Retrospective
Team reflects on their process and identifies improvements

ARTIFACTS

Team Backlog
Prioritized list of User Stories/Enabler Tasks team will work on

Sprint Backlog
Subset of work items from the Team Backlog, committed for a sprint

Increment
The product or deliverable created by a team during a sprint

Approved for Public Release

SAFe
PROGRAM LEVEL KEY ASPECTS

ROLES

Product Management
Responsible for identifying customer needs by developing the *Program Vision* & *Program Roadmap*

Prioritizes features in *Program Backlog* and guides work through *Program Kanban*

Release Train Engineer (RTE)
Servant Leader & Coach for the ART, organizes ART events & processes and assists the teams in delivering value

Communicates with stakeholders, resolves impediments manages risk & drives improvement

Systems Architect/Engineer
Defines and communicates the shared technical & architectural vision for an ART

Business Owners
Represents business stakeholders and ensures that the large solution meets their needs & expectations

EVENTS

Scrum of Scrums
RTE & Scrum Masters coordinate the dependencies of the ART and provide visibility into progress & impediments of PI objectives

PI Planning
Cadence-based event, Teams w/in ART plan & commit to delivering a set of prioritized capabilities (from the *Program Roadmap*) over a PI period

System Demo
End of Sprint demonstration and review of software changes & deliverable products to elicit feedback from users

Inspect and Adapt (I&A)*
End of PI workshop to review the ART's performance, identify improvement opportunities, and plan adjustments for the next PI

ARTIFACTS

Program Vision
High-level description that articulates the overarching goal or objective that the ART seeks to achieve

Program Backlog
Prioritized list of features, enablers, defects and other work items that that need to be delivered across multiple PIs to achieve the *Program Vision* and goals

Program Roadmap
Forecast of new capabilities & features (from *Program Backlog*), as planned objectives for the next few PIs (3) – vision of Solution-Level objectives

Program Kanban
Visualizes the flow of features & dependencies across Agile teams w/in an ART

Approved for Public Release

SAFe

LARGE SOLUTION LEVEL KEY ASPECTS

ROLES

Solution Management

Works with users/stakeholders understanding needs and define requirements to create the *Solution Vision* & *Solution Roadmap*

Prioritizes capabilities in the Solution Backlog and guides work in *Solution Kanban*

Solution Train Engineer (STE)

Servant Leader & Coach for the Solution Train, facilitates work across all ARTs & Suppliers in the Value Stream

Solution Architect/Engineer

Defines and communicates shared technical & architectural vision across a Solution Train

Supplier Manager

responsible for managing the relationship and collaboration with external suppliers/vendors who provide components, subsystems, services, or other inputs to the large-scale solution development

EVENTS

Solution Increment Planning

Cadence-based event, coordination of multiple ARTs & Suppliers towards common solution objectives

Solution Sync

Weekly event, STE provides visibility into Solution Train progress towards the Solution Roadmap, discuss epic/feature development, and scope adjustments

Solution Demo

Presentation of key product improvements & demonstration of working product (i.e.software) to users/stakeholders for feedback at the end of the PI

ARTIFACTS

Solution Vision

Describes overarching goals & objectives of the large solution; providing a guiding framework for all participating ARTs

Solution Backlog

Prioritized list of initiatives/needs & capabilities for multiple ARTs, required to deliver the large solution - prioritized based on business value, time criticality, and other dependencies

Solution Roadmap

Details timeline & major milestones for the larger solution, showing the execution of Initiatives & Capabilities (from *Solution Backlog*) for the next few PIs (3-5)

Solution Kanban

Visualizes the flow of initiatives, capabilities, & dependencies across multiple ARTs & Suppliers

Approved for Public Release

SAFe
PORTFOLIO LEVEL KEY ASPECTS

ROLES

Portfolio Management
Sets strategic direction, allocates budgets, and prioritizes investments across various Value Streams & solutions in the Portfolio Roadmap

Enterprise Architect
Defines the architectural principles/guidelines that guide solution development across the organization

Epic Owners
Represents the business or customer needs at the portfolio level and prioritizes initiatives/epics in the *Portfolio Backlog*

Lean Portfolio Management (LPM) Team
Coordinates and facilitates portfolio-level activities – investment decisions, budgeting & monitors portfolio performance

EVENTS

Lean Portfolio Sync
Regular meetings to review Portfolio performance, address risks & dependencies, and adjust Portfolio priorities & investments

Strategic Portfolio Planning
Sets strategic themes & objectives, aligns Portfolio investments with business strategy, and identifies the major initiatives/epics to be done in the *Portfolio Backlog*

ARTIFACTS

Strategic Themes & Objective
High-level business goals & priorities
Guides the Portfolio investments & solution development

Portfolio Backlog
Prioritized list of epics and initiatives based on investment & development w/in Portfolio
Provides visibility into upcoming work & helps make investment decisions

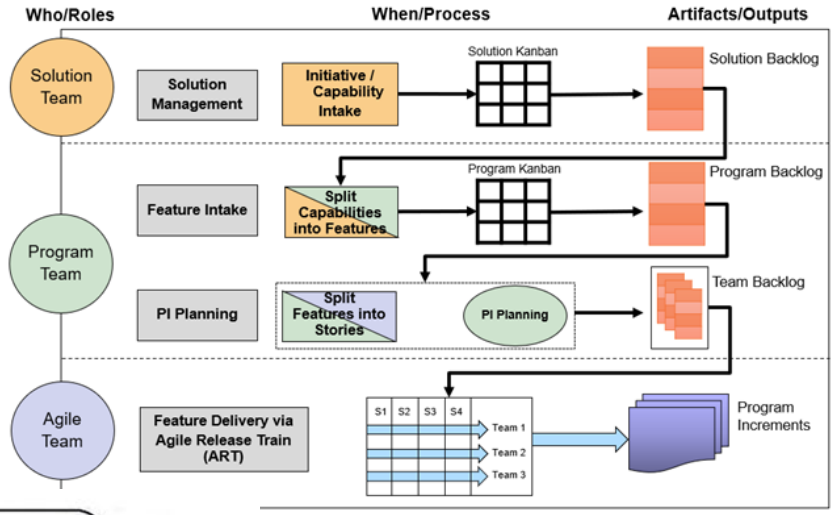
Portfolio Kanban Board
Visualizes the flow of epics and initiatives through various stages w/in Portfolio - ensures work aligns with strategic objectives & capacity constraints

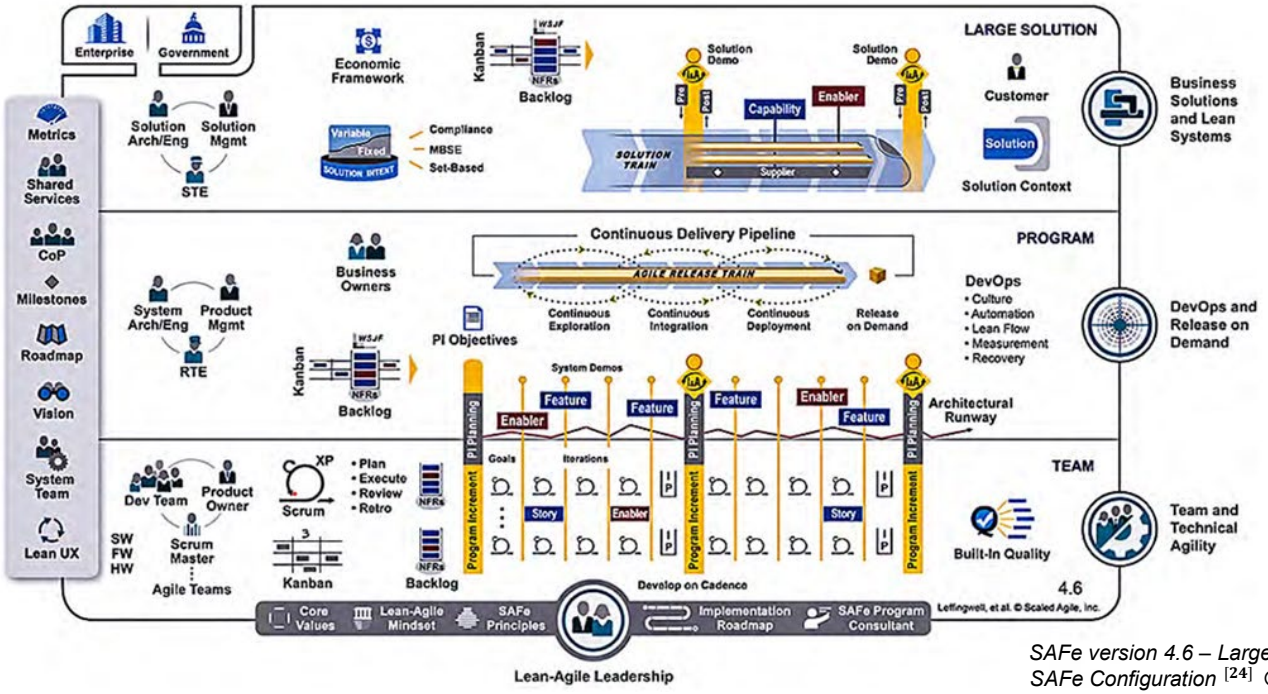
Approved for Public Release

LARGE SOLUTION SAFe
OVERVIEW

Typical development used in aerospace, defense, and government industries

- **Solution Train** – a system of systems – aligns multiple ARTs and Suppliers for the purpose of addressing greater requirements and delivering more complex solutions





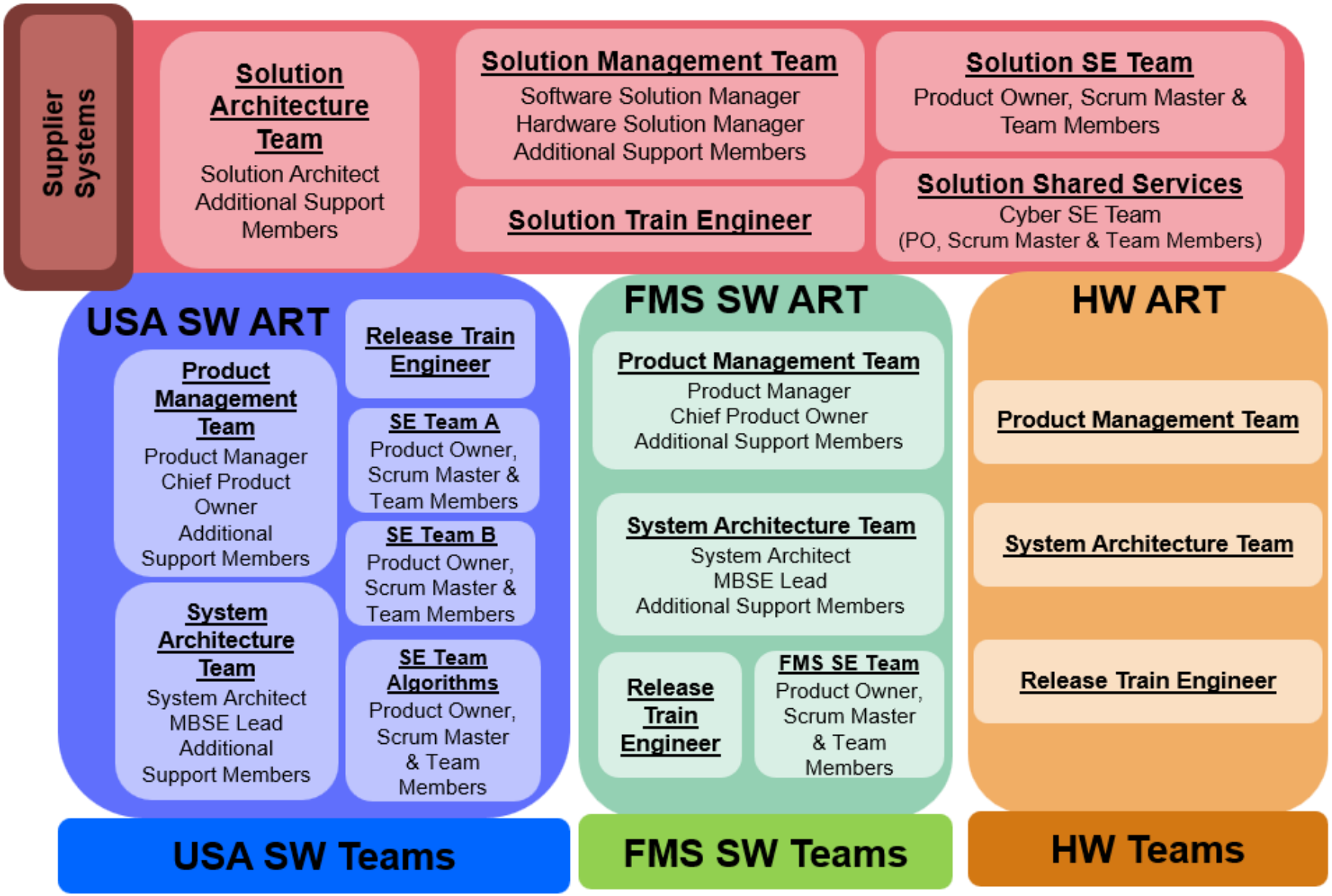
SAFe version 4.6 – Large Solution SAFe Configuration [24] **© *Scaled Agile, Inc.***

Systems Engineering (SE) activities occur at Solution & Program Levels

- SE work is planned & executed on same Agile cadences as Team Level

LARGE SOLUTION SAFe

ORGANIZATIONAL STRUCTURE



LARGE SOLUTION SAFe

PROGRAM INCREMENTS (PI)

PI incorporates the new capabilities, defect corrections and other improvements based on coordinated priorities

- 13-week phase, comprised of 4 Sprints
  - Fourth Sprint – Allocates time for training, innovation, finalization of current PI release, and prep for next PI

YEARLY CADENCE															
PI 1 = 13 Weeks				PI 2 = 13 Weeks				PI 3 = 13 Weeks				PI 4 = 13 Weeks			
Sprint 1 = 3 Weeks	Sprint 2 = 3 Weeks	Sprint 3 = 3 Weeks	Sprint 4 = 4 Weeks	Sprint 1 = 3 Weeks	Sprint 2 = 3 Weeks	Sprint 3 = 3 Weeks	Sprint 4 = 4 Weeks	Sprint 1 = 3 Weeks	Sprint 2 = 3 Weeks	Sprint 3 = 3 Weeks	Sprint 4 = 4 Weeks	Sprint 1 = 3 Weeks	Sprint 2 = 3 Weeks	Sprint 3 = 3 Weeks	Sprint 4 = 4 Weeks

Common PI Planning Events –

1. Supplier Prioritization:

— Supplier Managers meet with Solution Management, STE, and Users – *Supplier Backlog* refinement & prioritize tasks for upcoming PI

2. Prioritization Meeting

— Solution Management meets with users & stakeholders to:
  i. Refine Solution Roadmap
  ii. Develop a prioritized set of Features

3. Pre-PI Planning

— Solution Train meets to refine Prioritization Meeting Lists w.r.t. *Solution Vision* & *Intent* for upcoming PI
  - Representatives made aware of Test/Certification Events, Integration Activities, Infrastructure Shifts, Supplier Impacts, etc.

— Ends with a prioritized list of Features for upcoming PI

4. PI Planning

— Teams meet to plan and load their Sprints w.r.t. the Pre-PI Planning objectives & integration of Supplier deliverables
  - Teams coordinate any dependencies with other teams or shared services

BREAKDOWN OF PI & SPRINT EVENTS

Week	Monday	Tuesday	Wednesday	Thursday	Friday
1		PI Planning	PI Planning	PI Planning Sprint 1 Start	
2					
3					
4			Sprint 1 End	Sprint 2 Start	
5		Sprint 1 System Demo			
6					
7			Sprint 2 End	Sprint 3 Start	
8		Sprint 2 System Demo			
9					
10			Sprint 3 End	Sprint 4 Start Supplier Prioritization	
11		Sprint 3 System Demo	Prioritization Meeting	Prioritization Meeting	
12			Pre-PI Planning		
13					
14	Sprint 4 End Solution Demo	PI Planning	PI Planning	PI Planning	

Approved for Public Release

THANK YOU.

“U.S. Army Combat Capabilities Development Command Aviation & Missile Center provides increased responsiveness to the nation's Warfighters through aviation and missile capabilities and life cycle engineering solutions.”

REFERENCES (1)

[1]	Cunningham, Ward. (2001). *Manifesto for Agile Software Development*. Website. Retrieved from https://agilemanifesto.org/
[2]	Defense Acquisition University (DAU). (n.d.). *Software in NDAAs | Adaptive Acquisition Framework*. https://aaf.dau.edu/aaf/software/ndaa/
[3]	Defense Acquisition University (DAU). *CLE076 Introduction to Agile Software Acquisition*. DAU. Retrieved from https://dau.csod.com/ui/lms-learning-details/app/curriculum/941f6c35-7848-4f81-85fd-78f921c31ead
[4]	Department of Defense. (2020). *Operation of the Adaptive Acquisition Framework (DoDI 5000.02).* Retrieved from https://www.dau.edu/cop/e3/documents/dod-instruction-500002-operation-adaptive-acquisition-framework
[5]	Department of Defense. (2020). *Operation of the Software Acquisition Pathway (DoDI 5000.87).* Retrieved from https://www.dau.edu/cop/pm/documents/dod-instruction-500087-operation-software-acquisition-pathway
[6]	Inflectra Corporation. (2024). *Different SAFe Configurations Explained*. Inflectra Website. Retrieved from https://www.inflectra.com/Ideas/Topic/SAFe-Configurations.aspx
[7]	Office of the Secretary of Defense. (2010). *A New Approach for Delivering Information Technology Capabilities in the Department of Defense*. Retrieved from https://apps.dtic.mil/sti/tr/pdf/ADA534200.pdf
[8]	Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). (2023). *Agile 101: An Agile Primer v2.0.* Retrieved from https://aaf.dau.edu/storage/2023/05/Agile-101-Primer-v2.0_16May2023.pdf

REFERENCES (2)

[9]	Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). (2019). *Agile 101: An Agile Primer v1.0.* Retrieved from https://www.dau.edu/sites/default/files/Migrated/CopDocuments/Agile%20101%20v1.0.pdf
[10]	Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). (2020). *Agile Software Acquisition Guidebook: Best practices & lessons learned from the FY18 NDAA Section 873/874 Agile Pilot Program v1.0*. Retrieved from https://www.dau.edu/sites/default/files/Migrated/CopDocuments/Guidebook%20-%20OSD%2C%20Agile%20Software%2C%20V1.0.pdf
[11]	Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). (2020). *Agile Metrics Guide: Strategy Considerations and Sample Metrics for Agile Development Solutions v1.2.* Retrieved from https://aaf.dau.edu/wp-content/uploads/2022/08/Agile-Metrics-Guide.pdf
[12]	Project Management Institute, Inc. (2024). *SAFe From a Value Stream Perspective: Looking at SAFe from a Value Stream Perspective*. Website. Retrieved from https://www.pmi.org/disciplined-agile/da-flex-toc/safe-from-a-value-stream-perspective#:~:text=The%20Value%20Stream%20In%20SAFe%2C%20Partly%20Correct&text=In%20other%20words%2C%20value%20streams,the%20delays%20in%20delivering%20value
[13]	Rahate, Piyush (Lean-Agile Coach). (2021). *Exploring the Difference: Definition of Done vs Acceptance Criteria.* Agilemania. Retrieved from https://agilemania.com/definition-of-done-vs-acceptance-criteria
[14]	Scaled Agile, Inc. (2024). *Learn Scaled Agile Framework (SAFe).* Website. Retrieved from https://scaledagileframework.com/

REFERENCES (3)

[15]	Tervonen, David F. (Deputy Director, Integrated Program Management Acquisition, Analytics and Policy). (2020). *AAP Agile and Earned Value Management: A Program Manager's Desk Guide (Public Release 21-S-0413).* Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). Retrieved from https://www.acq.osd.mil/asda/ae/ada/ipm/docs/AAP%20Agile%20and%20EVM%20PM%20Desk%20Guide%20Update%20Approved%20for%20Nov%202020_FINAL.pdf
[16]	U.S. Department of the Army. (2024). *Enabling Modern Software Development and Acquisition Practices (Army Directive 2024-02)*. Retrieved from https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN40433-ARMY_DIR_2024-02-000-WEB-1.pdf
[17]	Urooj, Sidra (Team Coach at Handelsbanken). (2022). *Definition of Done vs Acceptance Criteria. LinkedIn.* Retrieved from https://www.linkedin.com/pulse/definition-done-vs-acceptance-criteria-sidra-urooj

REFERENCES (4)
IMAGE REFERENCES

[18]	Agile Cheetah. (2022). *Agile Project Roadmap* [Image]. Agile Cheetah Website. Retrieved from https://agilecheetah.com/agile-project-roadmap/
[19]	Bryson, By. (2023). *Agile Frameworks Comparison* [Image]. In *Agile Software Development: A Key To Innovative Digital Solutions In Tanzania*. IPF Softwares Website. Retrieved from https://ipfsoftwares.com/blogs/agile-software-development-a-key-to-innovative-digital-solutions-in-tanzania
[20]	Dodsworth, Quinn. (2024). *4 configurations of SAFe®* [Image]. In "The 4 levels of the Scaled Agile Framework ® explained". Published in *Agile and Scaled Agile, Projects and Programmes*. Pm-Partners Website. Retrieved from https://www.pm-partners.com.au/insights/the-4-levels-of-the-scaled-agile-framework-explained/
[21]	Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). (2023). *Figure 4: Agile Mindset, Value, Principles, and Practices* [Image]. In *Agile 101: An Agile Primer v2.0* [page 7]. Retrieved from https://aaf.dau.edu/storage/2023/05/Agile-101-Primer-v2.0_16May2023.pdf
[22]	Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). (2023). *Table 3 – Scrum Ceremonies* [Image]. In *Agile 101: An Agile Primer v2.0* [page 27]. Retrieved from https://aaf.dau.edu/storage/2023/05/Agile-101-Primer-v2.0_16May2023.pdf
[23]	Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)). (2019). *Figure 4 - Waterfall vs. Agile software development* [Image]. In *Agile 101: An Agile Primer v1.0* [page 9]. Retrieved from https://www.dau.edu/sites/default/files/Migrated/CopDocuments/Agile%20101%20v1.0.pdf

REFERENCES (5)
IMAGE REFERENCES CONT.

[24]	Scaled Agile Inc. (2020). *SAFe 4.6 – Large-Solution SAFe Configuration* [Image]. In *Using Large-Solution SAFe in a High-Profile U.S. Civilian Public Sector Agency* by Rico, David F.. Retrieved from https://davidfrico.com/y-safe-case-study-iv.pdf
[25]	Scaled Agile Inc. (2024). *SAFe 5.0 – Full SAFe Configuration* [Image]. Inflectra Website. Retrieved from https://www.inflectra.com/Ideas/Topic/SAFe-Configurations.aspx
[26]	Scrum.org^{TM}: The Home of Scrum. (2020). *Scrum Framework* [Image]. In *What is Scrum?.* Scrum.org^{TM}: The Home of Scrum Website. Retrieved from https://www.scrum.org/resources/what-scrum-module

Backup Slides

AGILE BASICS

COMMON ROLES

Users	→ Work closely with the Agile team to convey operational concepts, requirements/needs and to provide feedback on developed capabilities. → Participate in continuous testing activities, acceptance testing, and post-development assessments
Product Owner	→ Bridge between the users, stakeholders and development team (dev. team) → Responsible for conveying product's requirements to the dev. team, provides comments on interim developments, and coordinates demo(s). to gather users' feedback → Ensures Product Backlog is prioritized, refined, and delivered accordingly
Scrum Master	→ Ensures the execution of the Scrum framework within a dev. team → Organizes and facilitates various Scrum events → Identifies and removes obstacles blocking the team's progress.
Development Teams	→ Cross-functional team responsible for the development & delivery of intended product to the user ▪ Composed of 5-12 individuals representing various functional areas (i.e. requirements, engineering/design, development, security, safety, testing, operations)

AGILE BASICS

COMMON FRAMEWORKS

Scrum

→ Focuses on delivering as much quality software as possible within a series of short timeboxes
- Emphasizes collaboration, functioning software, team self-management, and the flexibility to adapt
- Work is divided into iterative cycles (sprints), ~2–4 weeks

Pros
- Clear Structure
- Focus on Deliverables
- Transparency
- Adaptability

Cons
- Strict Framework
- Role Clarity
- Requires Commitment

** **Scrum of Scrums**** - scaled version where multiple teams work together on a larger project

Kanban

→ Visual-based framework – focuses on continuous delivery without fixed iterations
- Enables teams to balance the *demand for work to be done* with the *available capacity for new work*

Pros
- Flexibility
- Visual Management
- Work in Progress (WIP) Limits

Cons
- Less Structure
- Scaling Challenges
- Requires Strict Diligence

Extreme Programming (XP)

→ Focuses on technical practices to produce higher quality software, that brings value to the customer by producing higher quality software
- Emphasizes practices like pair programming, test-driven development, and continuous integration

Pros
- High Quality Software
- Customer Involvement
- Frequent Releases

Cons
- Strict & Intensive Practices
- Customer Availability

AGILE BASICS

PRACTICES & TOOLS

Scrum	→ **Sprint Planning:** Collaborative sessions to plan work for upcoming sprint → **Daily Scrum:** Short daily meetings to synchronize the team's activities, discuss progress, and identify obstacles → **Sprint Review:** *(end of sprint event)* Team demonstrates completed work and gathers feedback from users/stakeholders → **Sprint Retrospective:** *(end of sprint event)* Team reflects on process, discusses what went well, areas of improvement, and identifies action items for next sprint → **Product Backlog:** Prioritized list of work items (features, enhancements, bug fixes, etc.) for the product
Kanban	→ **Visual Kanban Board:** Representations of work items and their status – typically divided into columns for different stages of workflow → **Work in Progress (WIP) Limits:** Limits on the number of allowable work items in a workflow stage → **Pull System:** Work done is based on available capacity rather than predefined schedules or quotas → **Continuous Delivery (CD):** Focuses on delivering small increments of work more frequently – minimizes lead time & improves predictability
XP	→ **Pair Programming:** Two programmers work together – one writes code and the other reviews & provides feedback in real-time → **Test-Driven Development (TDD):** Automated tests are written before the code – ensures code meets requirements & functions correctly → **Small Releases:** Frequently releasing small functional pieces of the product – ensures the ability to gather feedback & adapt to changing requirements → **Continuous Integration (CI):** Integrating code changes into a shared repository – ensures new changes are tested & validated ASAP

AGILE BASICS

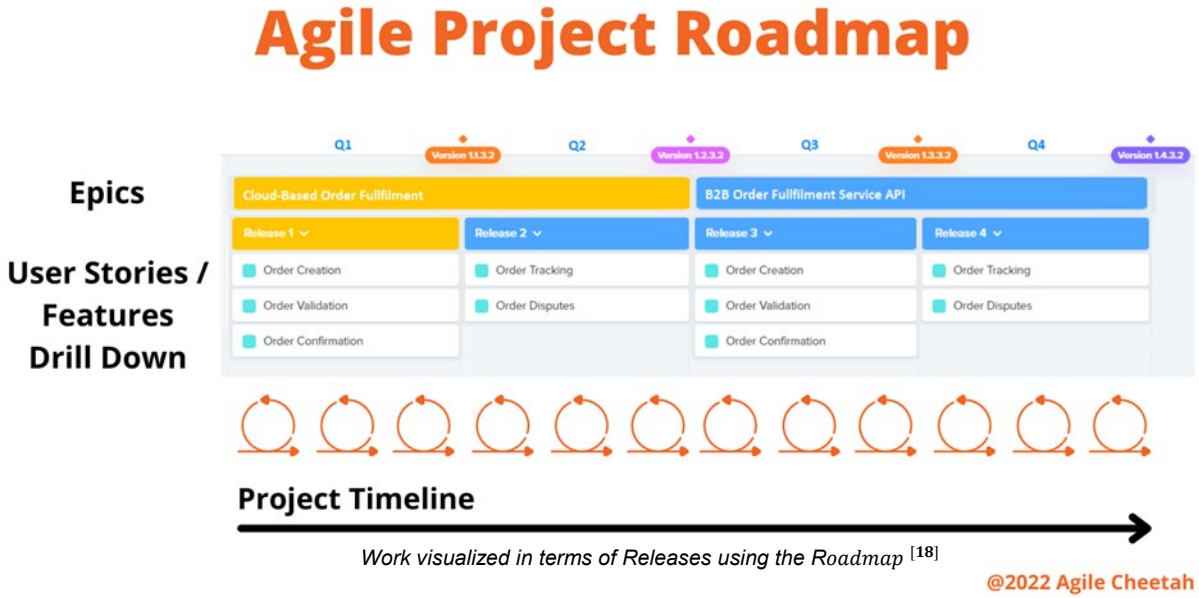
ROADMAPS & RELEASES

Roadmaps

→ Outlines product's goals and initiatives at a high-level over a span of time (typically 12 - 18 months)
- Evolve based on external conditions/dependencies, feedback, and priority

→ Align users/stakeholders on the project's vision & progress
- Provides transparency on upcoming priorities & milestones

→ Backlog items (epics & features) are derived from roadmap's strategic objectives



Work visualized in terms of Releases using the Roadmap [18]

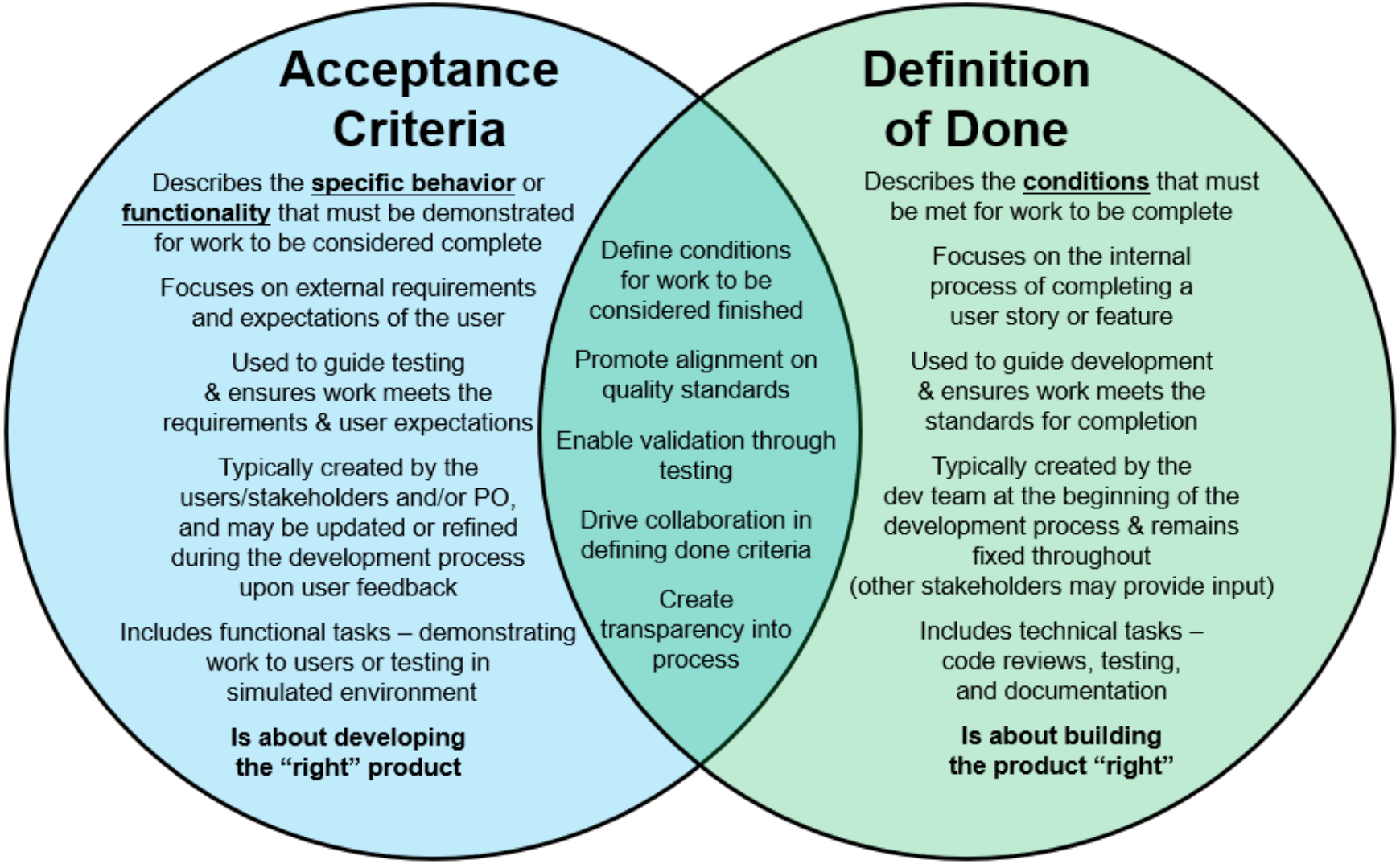
Releases

→ Usable increments of the product delivered at regular intervals (3-5 sprints)
- Prioritize delivering value early & often

→ Stakeholders and users provide feedback after every release to validate assumptions & inform future development

→ Releases are composed of multiple user stories and features – pulled from the Backlog based on priority and business value

AGILE BASICS
ACCEPTANCE CRITERIA & DEFINITION OF DONE




U.S. ARMY