Comparing Traditional and Agile Systems Engineering

Phyllis Marbach, Boeing

2 February 2016

INCOSE
International Council on Systems Engineering

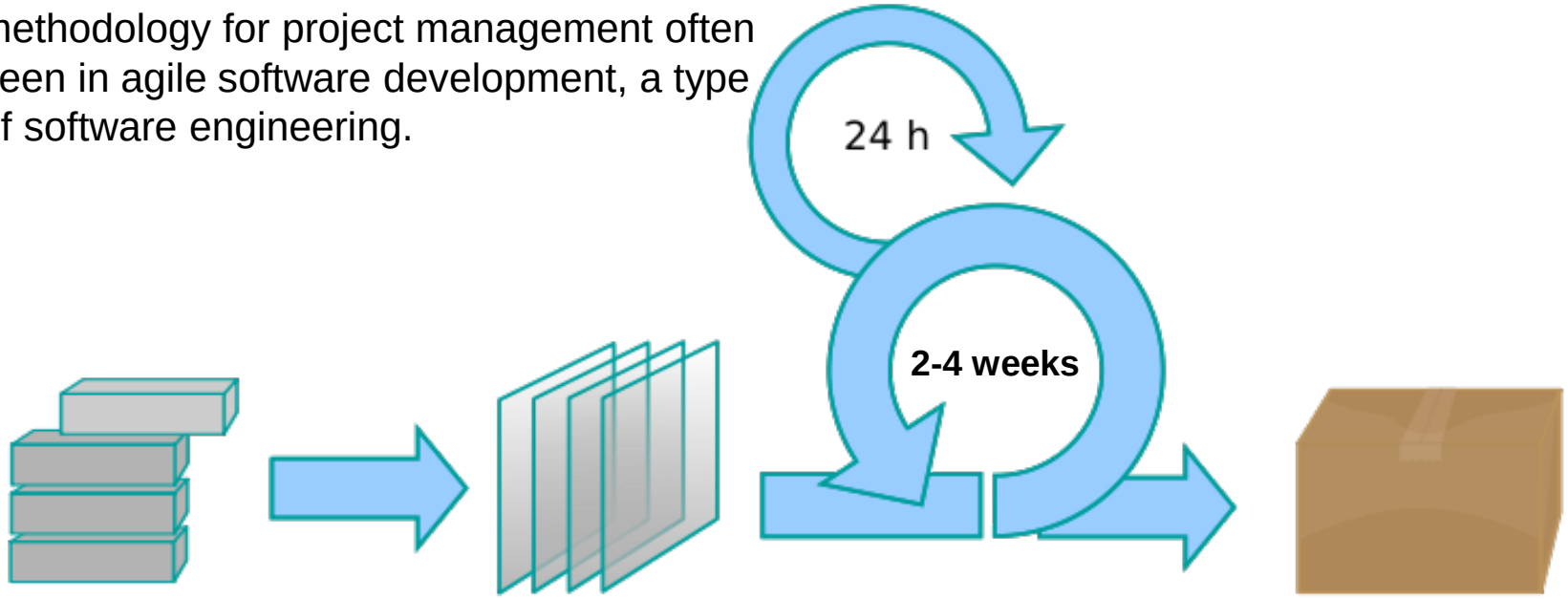
Much of this Content from:

- Systems Engineering for Software Intensive Projects Using Agile Methods
- Presented at INCOSE International Symposium 2014
- Authors:
  - Larri Rosser, Raytheon, Garland, TX
  - Phyllis Marbach, Boeing, Huntington Beach, CA
  - Gundars Osvalds, Praxis Engrg, Annapolis Junction, MD
  - David Lempia, Rockwell Collins, Cedar Rapids, IA




Introduction to Agile (Scrum)

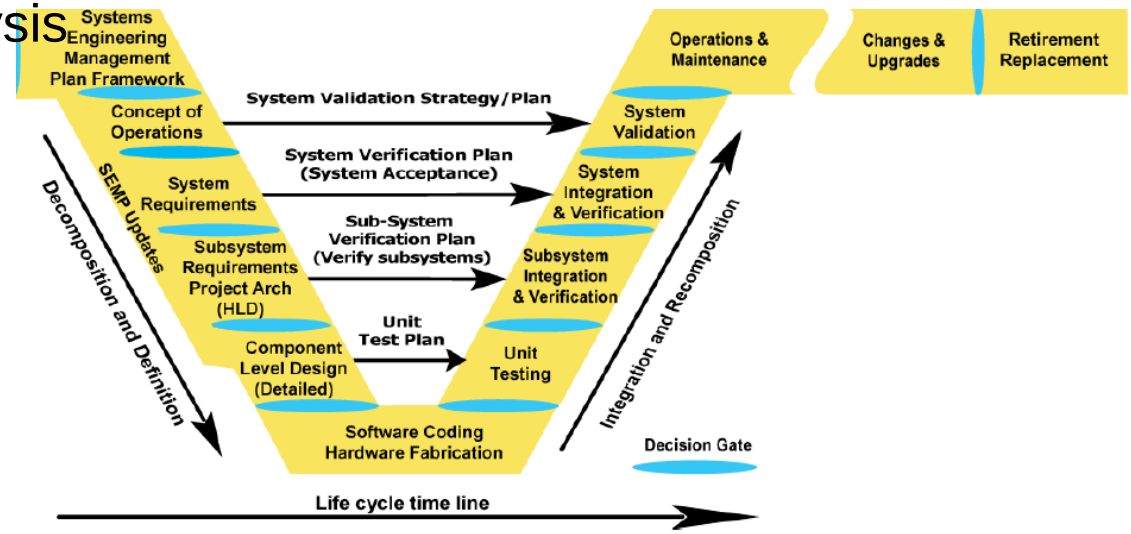
Scrum is an iterative, incremental methodology for project management often seen in agile software development, a type of software engineering.



Product Backlog **Iteration Backlog** **Iteration** **Working increment of a capability**

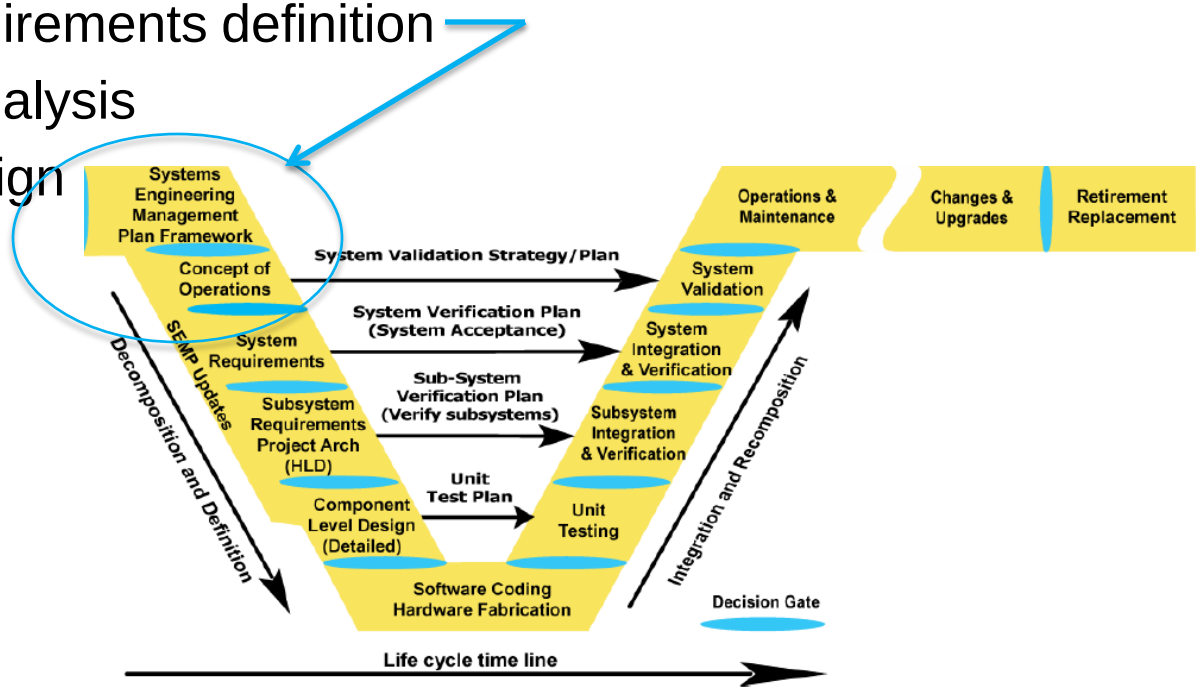
Systems Engineering

- An interdisciplinary approach and means to enable the realization of successful systems (INCOSE handbook)
- Focus is on the role of Systems Engineering (SE) in support of implementation
- Technical processes addressed:
  - Stakeholder requirements definition
  - Requirements analysis
  - Architectural design
  - Implementation
  - Integration
  - Verification



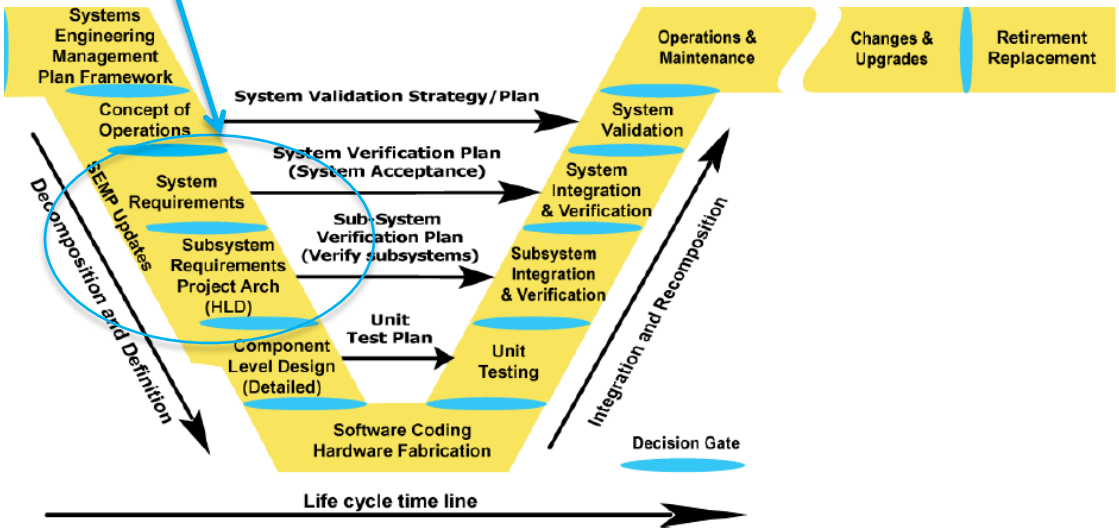
Systems Engineering

- An interdisciplinary approach and means to enable the realization of successful systems (INCOSE handbook)
- Focus is on the role of SE in support of implementation
- Technical processes addressed:
  - Stakeholder requirements definition
  - Requirements analysis
  - Architectural design
  - Implementation
  - Integration
  - Verification



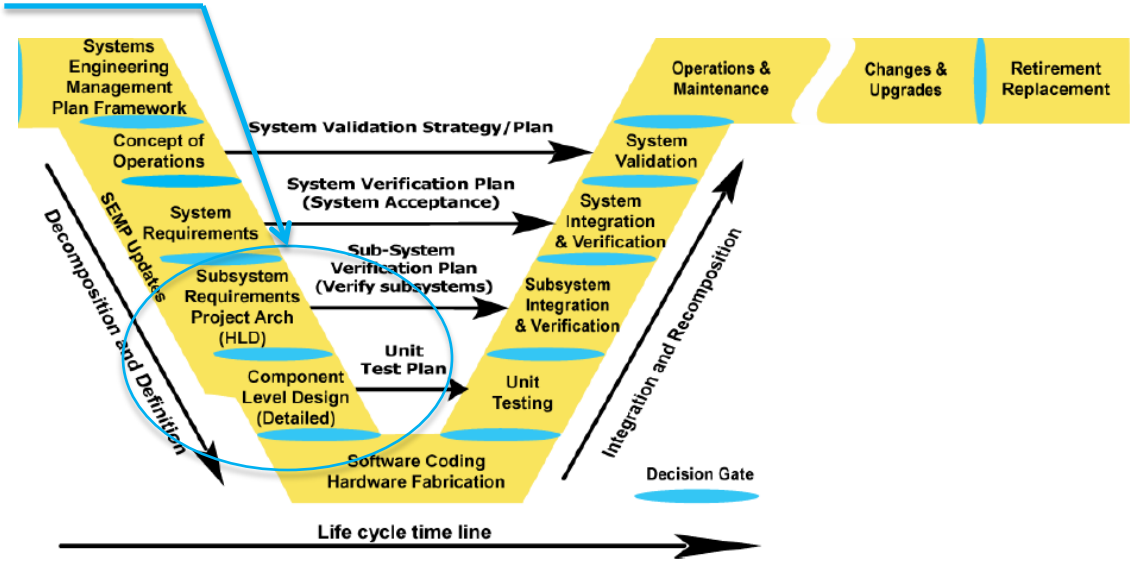
Systems Engineering

- An interdisciplinary approach and means to enable the realization of successful systems (INCOSE handbook)
- Focus is on the role of SE in support of implementation
- Technical processes addressed:
  - Stakeholder requirements definition
  - Requirements analysis
  - Architectural design
  - Implementation
  - Integration
  - Verification



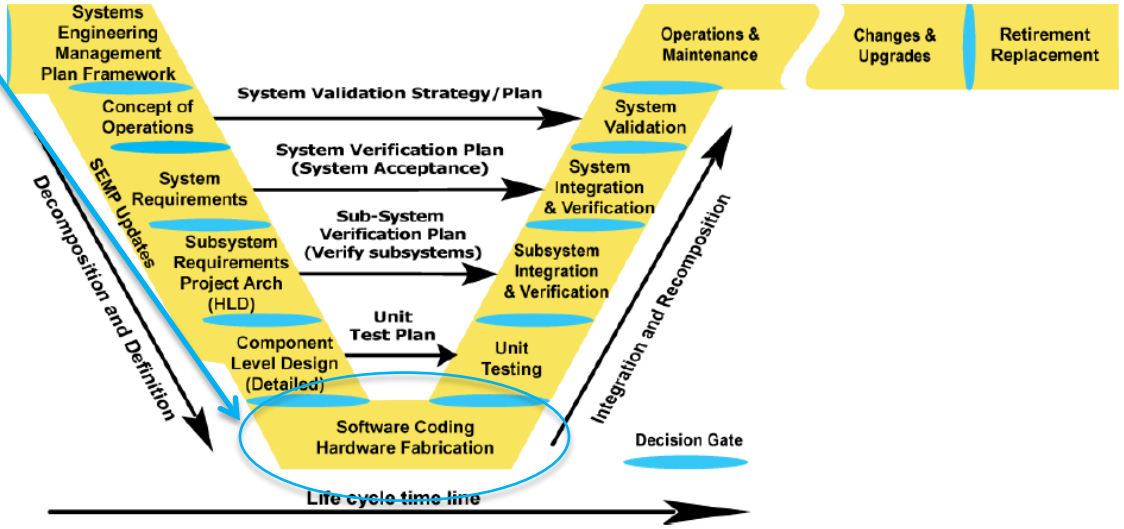
Systems Engineering

- An interdisciplinary approach and means to enable the realization of successful systems (INCOSE handbook)
- Focus is on the role of SE in support of implementation
- Technical processes addressed:
  - Stakeholder requirements definition
  - Requirements analysis
  - Architectural design
  - Implementation
  - Integration
  - Verification



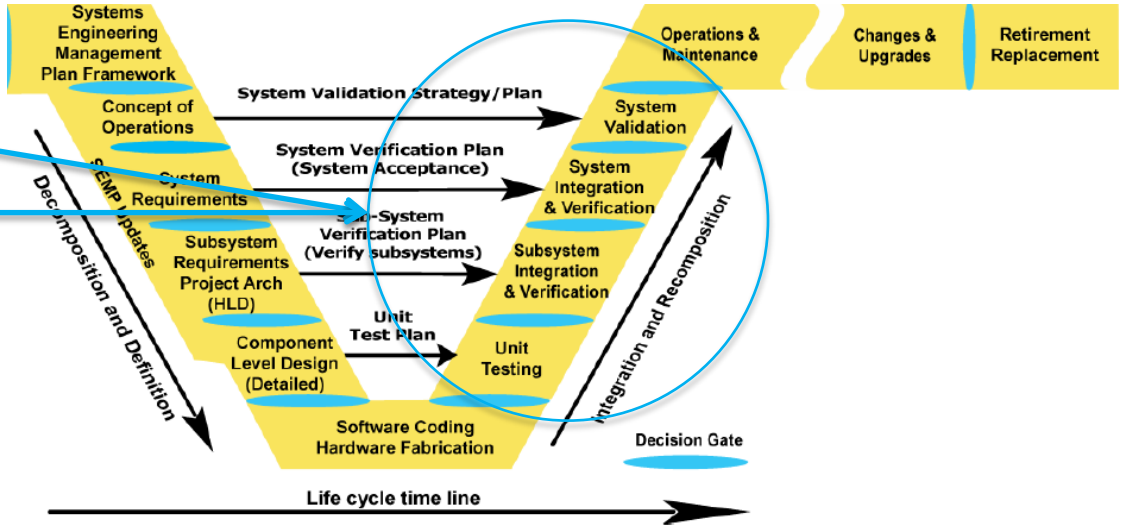
Systems Engineering

- An interdisciplinary approach and means to enable the realization of successful systems (INCOSE handbook)
- Focus is on the role of SE in support of implementation
- Technical processes addressed:
  - Stakeholder requirements definition
  - Requirements analysis
  - Architectural design
  - Implementation
  - Integration
  - Verification



Systems Engineering

- An interdisciplinary approach and means to enable the realization of successful systems (INCOSE handbook)
- Focus is on the role of SE in support of implementation
- Technical processes addressed:
  - Stakeholder requirements definition
  - Requirements analysis
  - Architectural design
  - Implementation
  - Integration
  - Verification



Agile SE Framework

- Changes to the architecture – modular and evolving
- Changes to the process – iterative, incremental
- Changes to the roles
  - SE become members of the implementation teams;
  - SE staffing remains more level throughout the development to support and maintain the architecture, requirements, testing, verification, artifact development, etc.

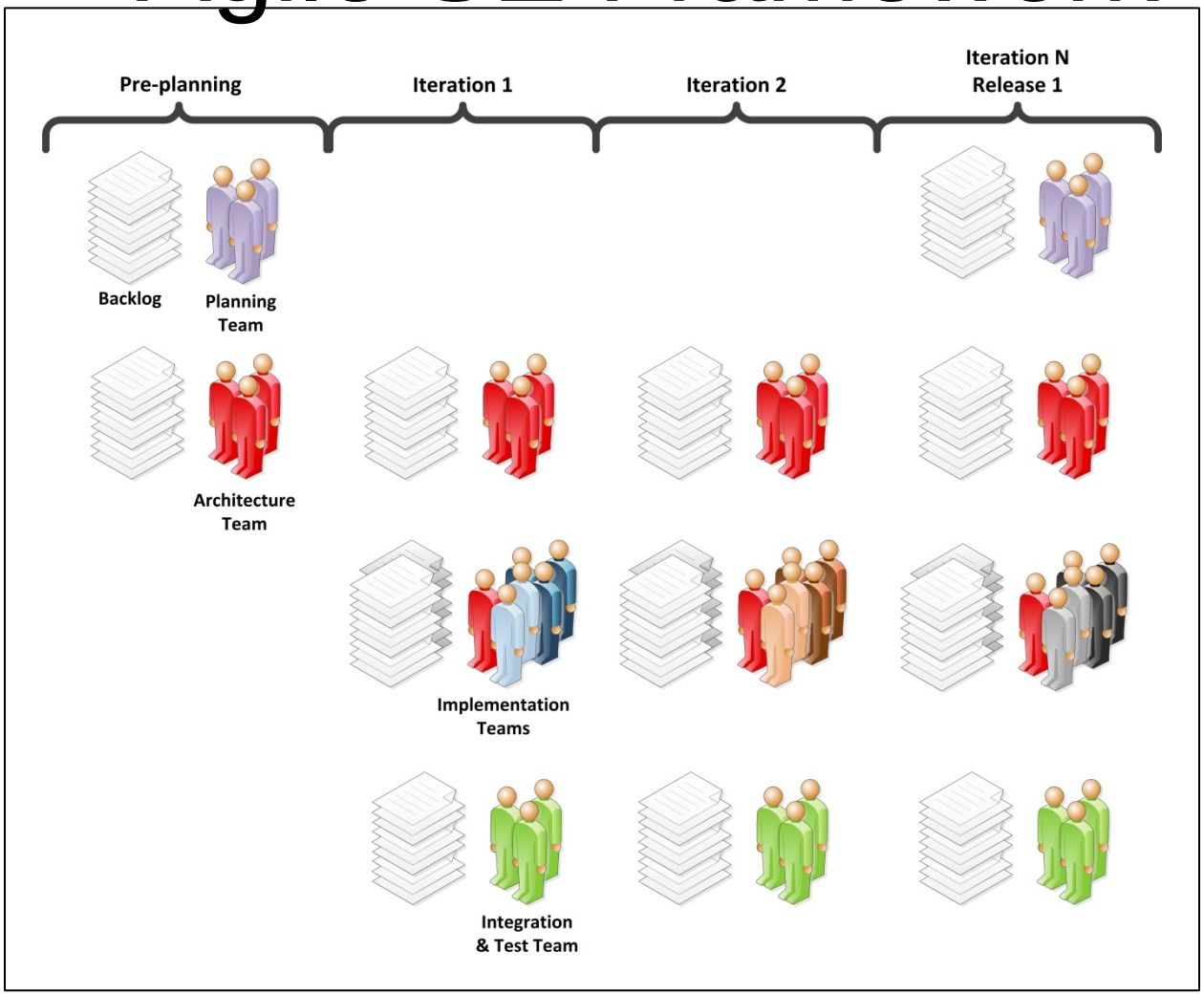
SE Architect Role

- SE identify and analyze architecture dependencies
- Create and continuously update an architecture description
- Participate with the SE Team
- Participate on one or more Implementation Teams
- Work one iteration ahead of the developers

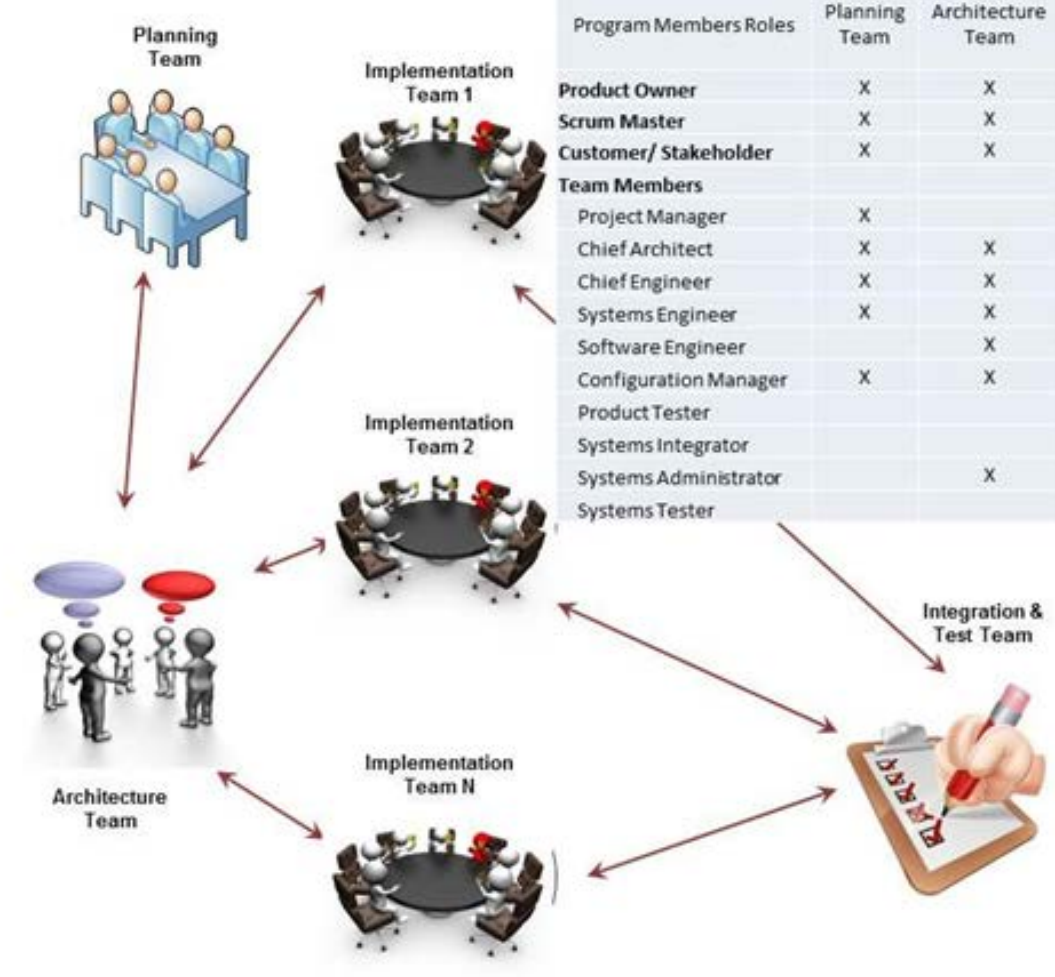
SE Process

- SE and software development work together to:
  - Define capabilities
  - Implement capabilities
  - Test capabilities
  - Inspect the results
  - Adapt capabilities as needed
  - Maintain system integrity
- Larger programs with several teams working in parallel need SE engaged
- Each aspect of development (requirements, design, implementation, test, verification) is continually revisited throughout the development lifecycle

Agile SE Framework



Agile Teams



Program Members Roles	Planning Team	Architecture Team	Implementation Team(s)	Integration & Test Team
Product Owner	X	X	X	X
Scrum Master	X	X	X	X
Customer/ Stakeholder	X	X	X	X
Team Members				
Project Manager	X			
Chief Architect	X	X		
Chief Engineer	X	X	X	X
Systems Engineer	X	X	X	X
Software Engineer		X	X	
Configuration Manager	X	X	X	X
Product Tester			X	
Systems Integrator				X
Systems Administrator		X	X	X
Systems Tester			X	X

Example Planning Team RACI

PLANNING TEAM				
ROLES	Product Owner	Scrum Master	Team Member	Customer/ Stakeholder
Scope	C	C	R	A
Define Deliverables (Product Level)	C	C	R	A
Technical Management Mission	A	C	R	C
Needs Analysis	A	C	R	C
Requirement Articulation (Product Capability Backlog)	R	C	C	A
Requirements Management	A	C	R	C
Meeting Facilitator/ Impediment Remover	A	R	C	C

RACI Matrix Legend	
Responsible (one)	Leads the task completion with tangible deliverables
Accountable (one)	Delagated the responsibility for task, approves completion
Consulted (many)	Multiple contributors provide special knowledge or expertise
Informed (many)	Members that will be informed of the task status and deliverables

Example Arch Team RACI

ARCHITECTURE TEAM				
ROLES	Product Owner	Scrum Master	Team Member	Customer/ Stakeholder
Vision	A	C	R	C
Roadmap	A	C	R	C
Architecture Framework/ System Design	C	C	R	A
Define and Maintain Interfaces	A	C	R	C
Architecture Product Backlog	A	C	R	C
Concept of Operations (CONOP)	C	C	R	A
Perform Trade Studies	A	C	R	C
Meeting Facilitator/ Impediment Remover	A	R	C	C

RACI Matrix Legend	
Responsible (one)	Leads the task completion with tangible deliverables
Accountable (one)	Delagated the responsibility for task, approves completion
Consulted (many)	Multiple contributors provide special knowledge or expertise
Informed (many)	Members that will be informed of the task status and deliverables

Example Implementation RACI

IMPLEMENTATION TEAM				
ROLES	Product Owner	Scrum Master	Team Member	Customer/ Stakeholder
Develop/ Maintain Software Design (Detailed Design	A	C	R	C
Software Implementation	A	C	R	C
Integration - unit test, SW integration as possible	A	C	R	C
Verification	A	C	R	C
Maintain/ Verify System Capabilities as possible	A	C	R	C
Maintain Interface Definitions of SW/ Component	A	C	R	C
Perform Trade Studies	A	C	R	C
Develop/ Maintain Test Procedures	A	C	R	C
Meeting Facilitator/ Impediment Remover	A	R	C	C

RACI Matrix Legend	
Responsible (one)	Leads the task completion with tangible deliverables
Accountable (one)	Delagated the responsibility for task, approves completion
Consulted (many)	Multiple contributors provide special knowledge or expertise
Informed (many)	Members that will be informed of the task status and deliverables

Example I&T Team RACI

INTEGRATION AND TEST TEAM				
ROLES	Product Owner	Scrum Master	Team Member	Customer/ Stakeholder
Software Backup/ SW Baseline Test Baseline	A	C	R	I
System Integration	A	C	R	I
Validation	A	C	R	C
Meeting Facilitator/ Impediment Remover	A	R	C	C

RACI Matrix Legend	
Responsible (one)	Leads the task completion with tangible deliverables
Accountable (one)	Delagated the responsibility for task, approves completion
Consulted (many)	Multiple contributors provide special knowledge or expertise
Informed (many)	Members that will be informed of the task status and deliverables

Challenges with Traditional SE and Agile Software Dev

- Lack of Rapid Response
- Big Design Up Front
- Architecture Interpretation
- Non-Functional Requirements (NFR)
- Responding to Change at Scale
- Verification, Validation and Test

Agile SE Framework Conclusion

- SE and software development work together to develop and evolve the work products iteratively
- Define “just enough” architecture and requirements prior to the beginning of implementation
- Release Planning and Iteration Planning are essential to detail the work and coordinate the teams
- Release products frequently
- Absorb changes to mission requirements
- Include requirements, architecture, system design and validation by SE on large scale agile projects

Will Iterative and Incremental Development Work for you?

- Types of questions to ask:
  - Is the system complex?
  - Do you expect to have discoveries during the development that will cause changes?
  - Is there an involved and committed customer or product owner (PO) that can help prioritize the product backlog and define the requirements?
  - Can the PO be present at the planning and demonstrations to verify the team(s) are on track?

References

ADAPT 2013. *Achieving Better Buying Power 2.0 For Software Acquisition: Agile Methods.* The Agile Defense Adaption Proponents Group of The Association for Enterprise Information. http://www.afei.org/WorkingGroups/ADAPT/Pages/default.aspx

Brown et al. 2010. Brown Nanette, Nord Robert, Ozkaya Ipek. *Enabling Agility Through Architecture.* CrossTalk. http://www.sei.cmu.edu/library/assets/whitepapers/brown-nord-ozkaya-crosstalk-Nov10.pdf

DoD. 2010. *Better Buying Power.* Department of Defense. http://www.acq.osd.mil/docs/USD_ATL_Guidance_Memo_September_14_2010_FINAL.PDF

DoD. 2012. *Defense Acquisition Guidebook.* Department of Defense. https://acc.dau.mil/CommunityBrowser.aspx?id=289207&lang=en-US

Frank M. 2000. *Cognitive and Personality Characteristics of Successful Systems Engineering.* INCOSE International Symposium Proceedings. https://www.incose.org/ipub/00/contents/s_1_6/163_101.pdf

Honour, Eric. C. 2004. *Understanding the Value of Systems Engineering.* INCOSE International Symposium. http://www.incose.org/secoe/0103/ValueSE-INCOSE04.pdf

INCOSE. 2011. *INCOSE Systems Engineering Handbook.* International Council on Systems Engineering, v3.2.2. http://www.incose.org/ProductsPubs/products/sehandbook.aspx

ISO/IEC. 2008. *15288 Systems and software engineering — System life cycle processes*. ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission). http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43564

Leffingwell, Dean. 2011. *Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise.* Pearson Education, Inc., Boston, MA. http://deanleffingwell.com/book-agile-software-requirements/

Standish Group. 1994. *The Chaos Report.* The Standish Group International. http://www.csus.edu/indiv/v/velianitis/161/ChaosReport.pdf

Standish Group. 2013. *The Chaos Manifesto.* The Standish Group International. http://versionone.com/assets/img/files/ChaosManifesto2013.pdf