

Agile Systems Engineering

INCOSE Chesapeake Chapter
Laurel, MD
September 2011

Dr. Suzette S. Johnson
Agile Engineering
Northrop Grumman Corp
Annapolis Junction, MD



Discussion Outline

- Defining an Agile Environment
- Requirements, Use Cases, User Stories
- Levels Planning
- User Story Verification and Validation
- Summary
- References

2.2 Definition of Systems Engineering

Systems engineering is a perspective, a process, and a profession, as illustrated by these three representative definitions.

Systems engineering is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect. (Ramo¹)

Concentrates on the design and application of the whole system

Systems engineering is an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near optimal manner, the full range of requirements for the system. (Eisner²)

It focuses on defining customer needs and required functionality early in the development cycle....

Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE³)

Reference: (2011). INCOSE SE Handbook V3.2.1, p. 7.

An Agile Environment

- Adaptive, Responsive, Evolving, Continuous Improvement
- Improved transparency of progress
- End-to-end accountability and ownership
- Reduces time-to-deploy operational capability
- Ability to adapt to changing requirements and new technological advancements



Agile is an empirical process

Agile Principles



Early and Continuous Delivery of Value	A Working System is the Primary Measure of Progress	Welcome Changing Requirements
Deliver a Working System Frequently	Business People and Developers Must Work Together Daily	Motivated and Empowered Individuals
Face-to-face Conversation	Promote Sustainable Development	Continuous Attention to Technical Excellence
Simplicity	The Best Architectures, Requirements and Designs Emerge From Self-Organizing Teams	Regular Team Reflection on How to Become More Effective

<http://agilemanifesto.org/>

3.4.4 Agile Development³⁶

The preceding discussions have emphasized the benefits of orderly, hierarchical baseline progression followed by a corresponding verification sequence. Recognizing that the development process may require more flexibility in some circumstances, the Agile Development approach provides a tailoring framework, based on opportunity to simplify control methods and to assess the risks in so doing. The extent of tailoring is determined by whether the opportunity to shorten the project cycle is worth the risk of doing development steps out of sequence or in parallel.

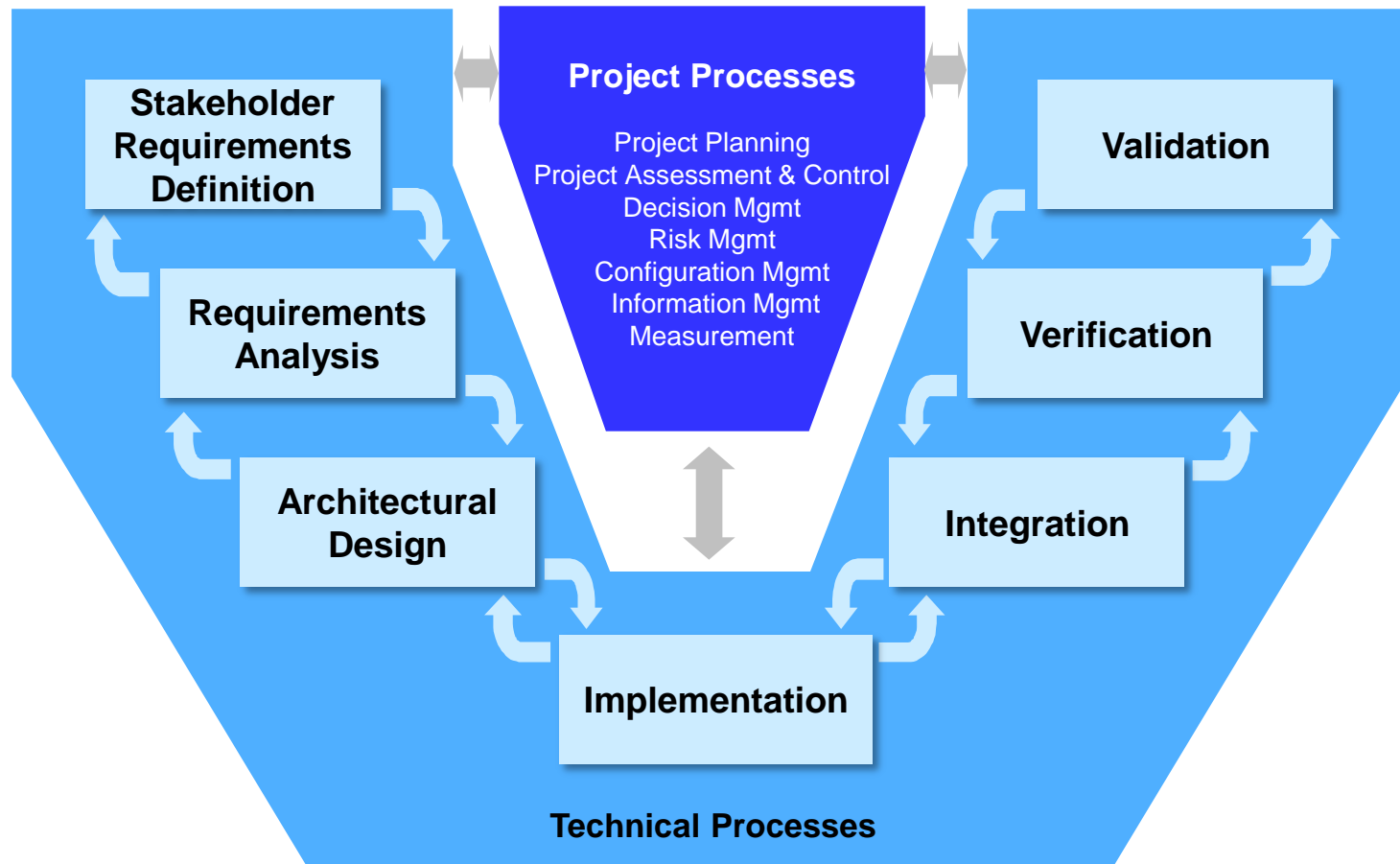
The Agile Alliance (www.agilealliance.com) is dedicated to: developing iterative and agile methods, seeking a faster and better approach to software and system development, and challenging more traditional models. There are many articles describing the agile concepts.³⁷ The key objective is flexibility, and allowing selected events to be taken out of sequence (see Figure 3-10 and Figure 3-11) when the risk is acceptable.



A snippet....

Reference: (2011). INCOSE SE Handbook V3.2.1 . p.40.

NG ISO/IEC/IEEE 15288-Based SE Process V-Model



Our User Story for this Presentation

- User Story
 - As a Systems Engineer I want to have a better understanding as to how Systems Engineering practices align with the Agile framework so I have insight as to how this might be implemented in an Agile environment.
- Acceptance Criteria
 - Communicate how requirements analysis and design occur when using an Agile approach
 - Discuss considerations regarding architecture
 - Understanding of the relationship of requirements to user stories and why user stories
 - Insight into the Agile framework and how it helps manage changing requirements and priorities
 - Identification of verification and validation practices within the context of the Agile framework

The Agile Framework

Capabilities and User Stories

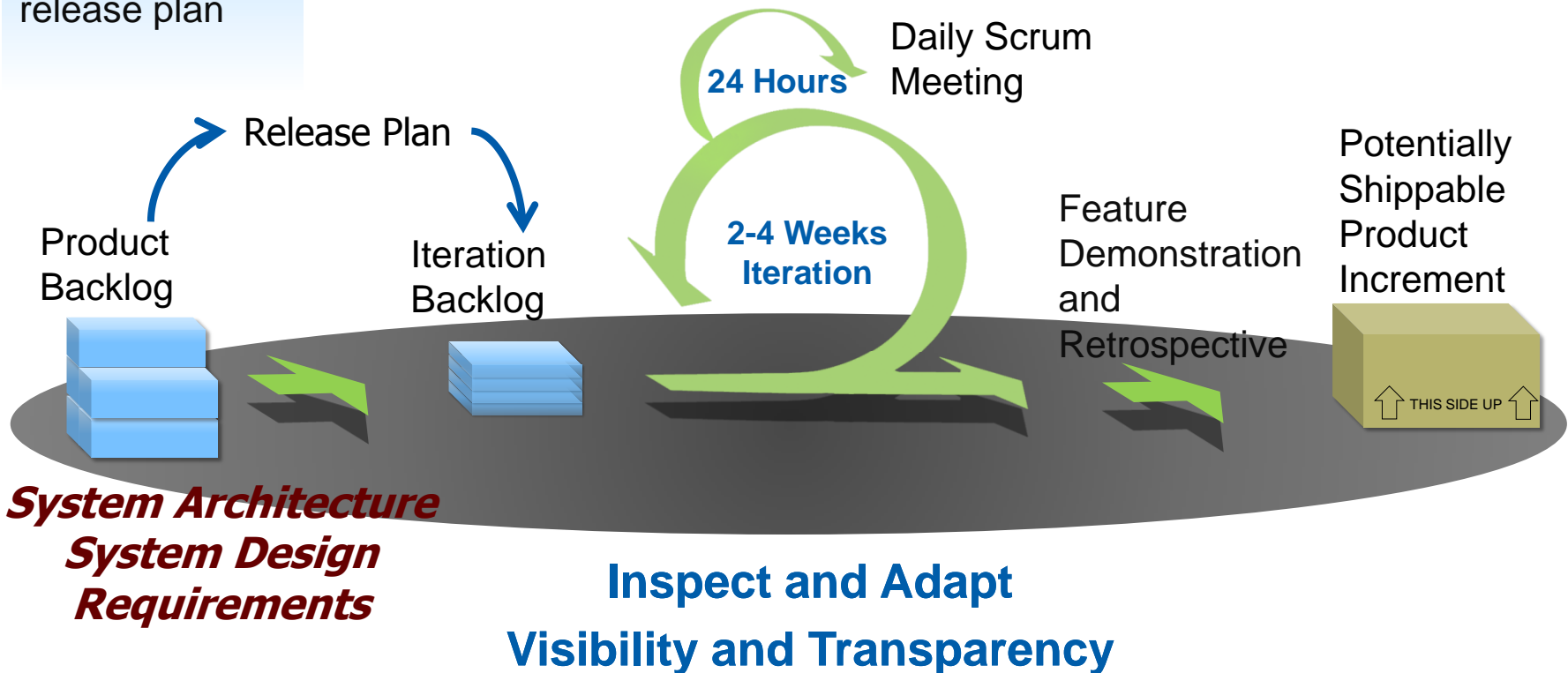
Prioritized by Product Owners

Creates the release plan

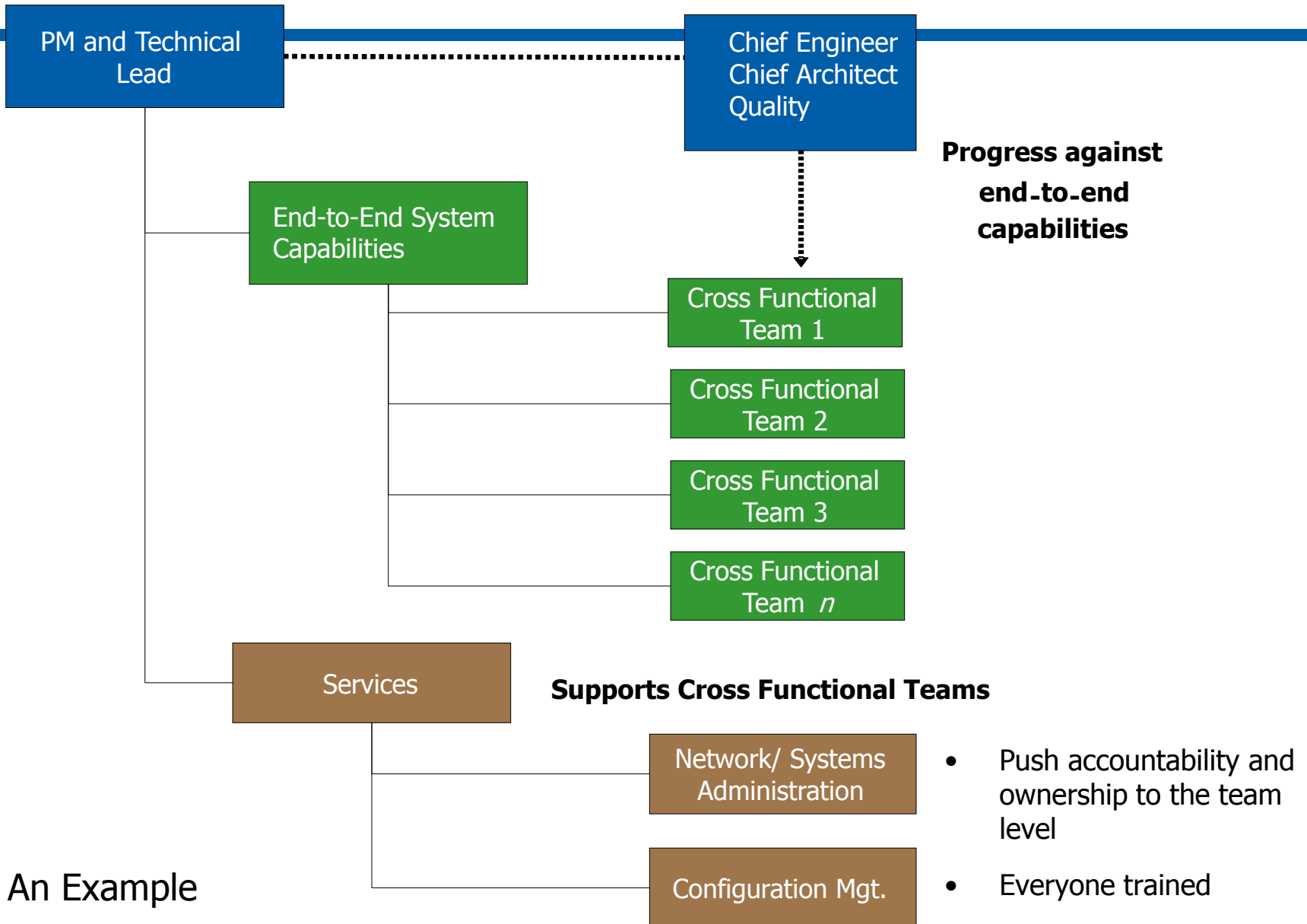
The Daily Tasks managed by the Cross Functional Team

Design, Code, Integrate, Test

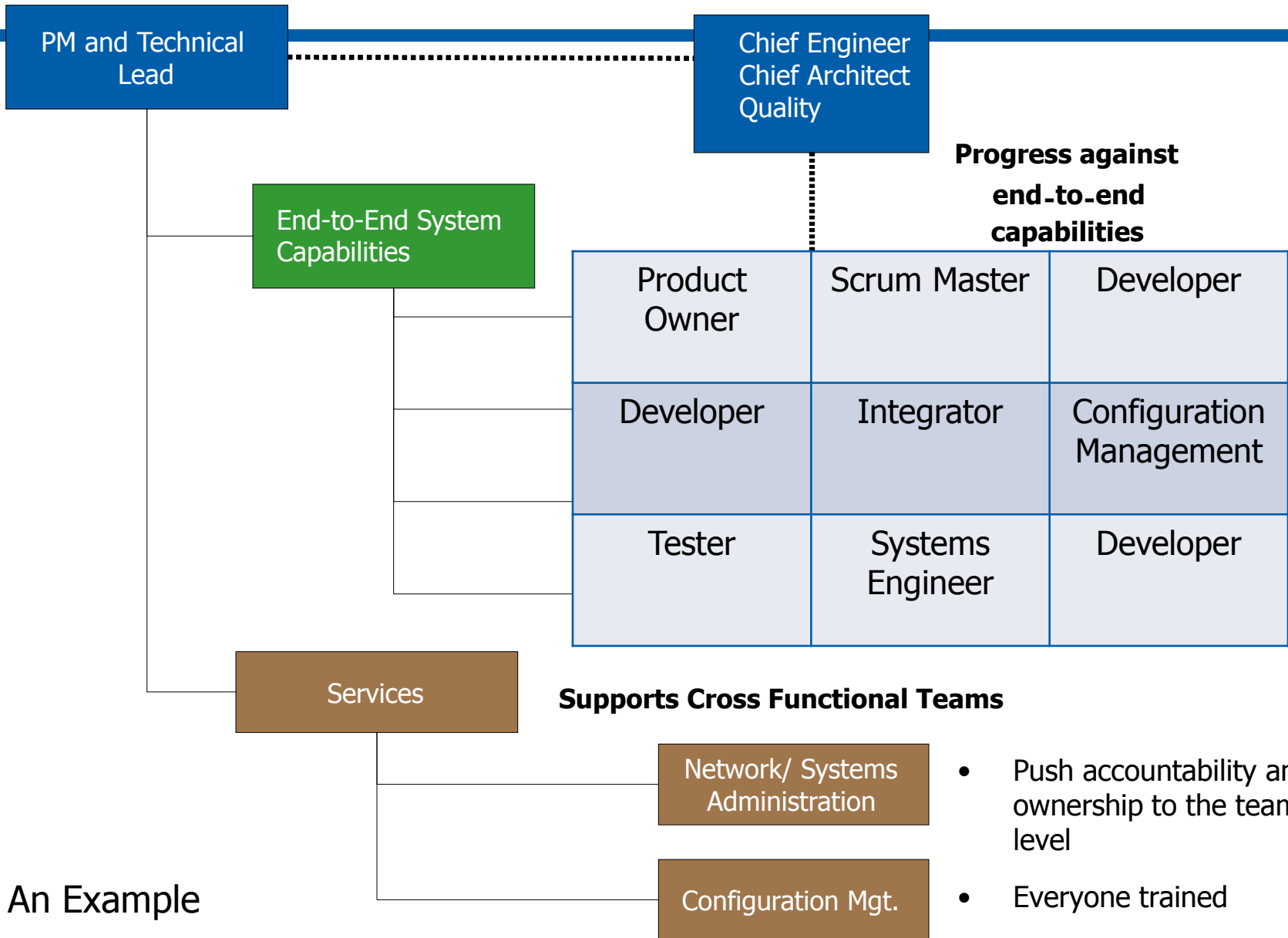
Commitment
Identification of Impediments
Communication



Project Team Structure



Project Team Structure



An Example

Project Team Structure

Product Owner	Scrum Master	Developer
Developer	Integrator	Configuration Management
Tester	Systems Engineer	Developer

Product Owner	Scrum Master	Developer
Developer	Integrator	Configuration Management
Tester	Systems Engineer	Developer

Product Owner	Scrum Master	Developer
Developer	Integrator	Configuration Management
Tester	Systems Engineer	Developer



Systems Engineer
Systems Engineer
Systems Engineer

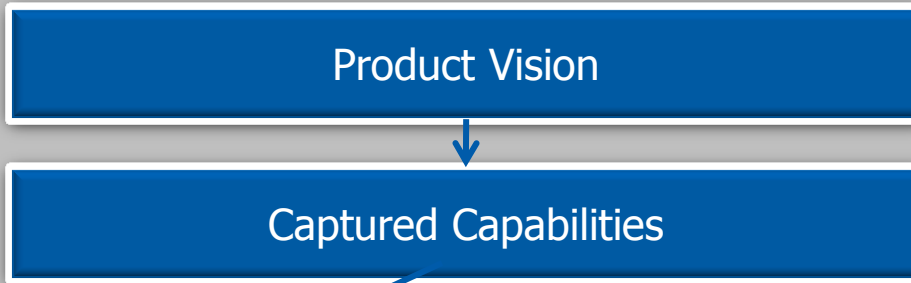


*Cross Team Collaboration
Understanding the bigger picture*

A Cross-functional team is responsible for capability/thread from planning to final acceptance. Systems Engineers from the teams collaborate to maintain an end-to-end systems view.

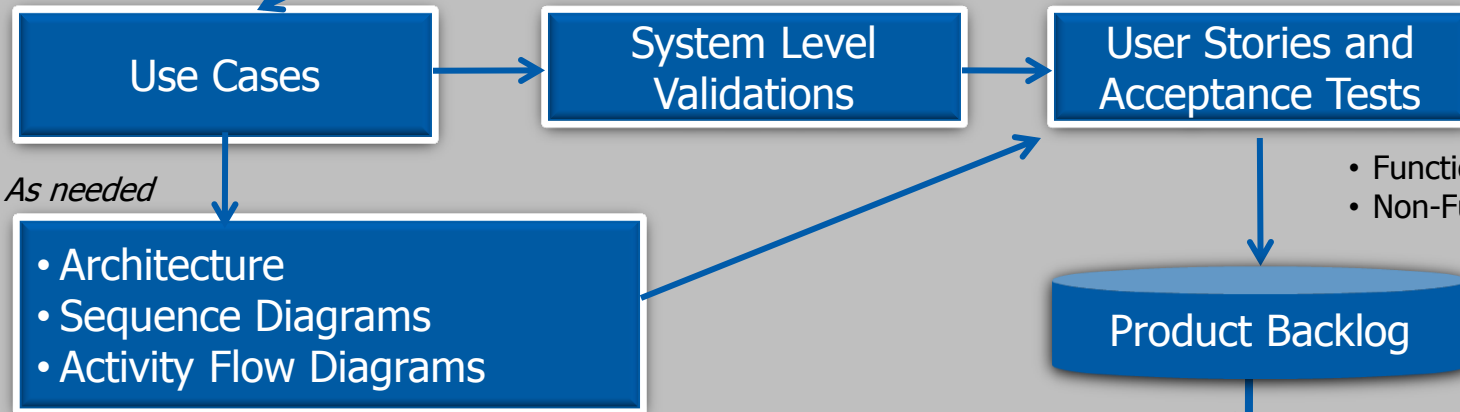
Delivering Value

Creating the Vision



*System Architecture
Requirements mapped to stories
Revisit architecture and design each release and iteration*

Developing Understanding

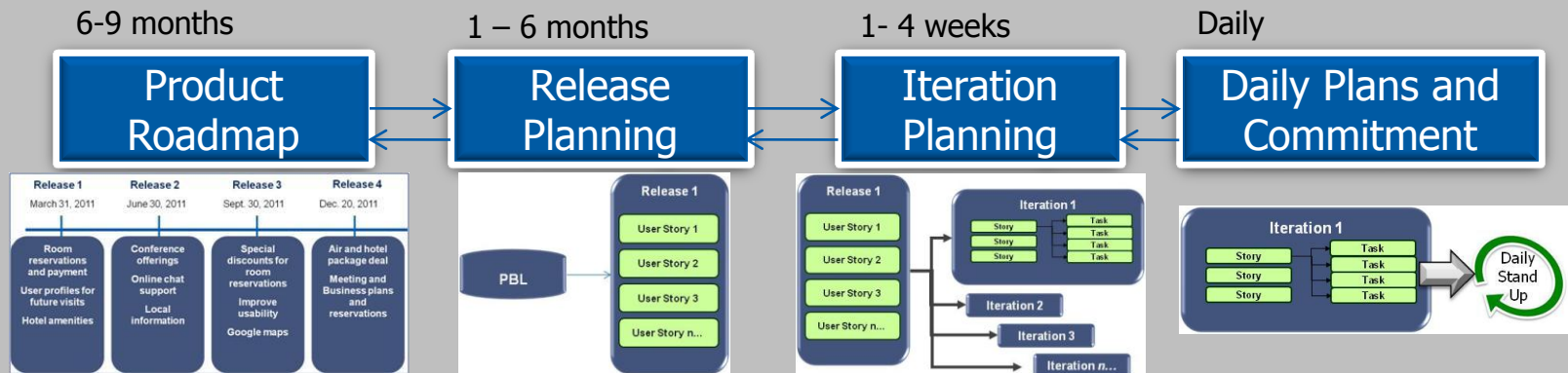


As needed

- Architecture
- Sequence Diagrams
- Activity Flow Diagrams

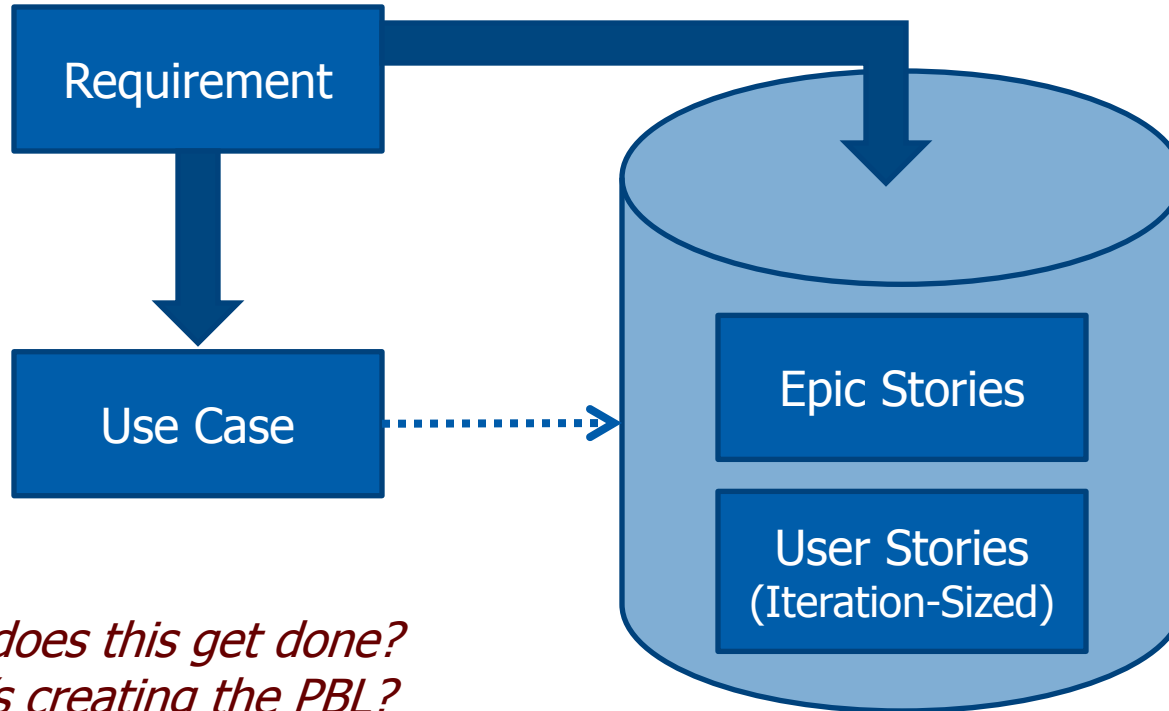
- Functional
- Non-Functional

Planning and Estimating the Work



Agile Systems Engineering Ontology

Simplified view for creating the Product Backlog



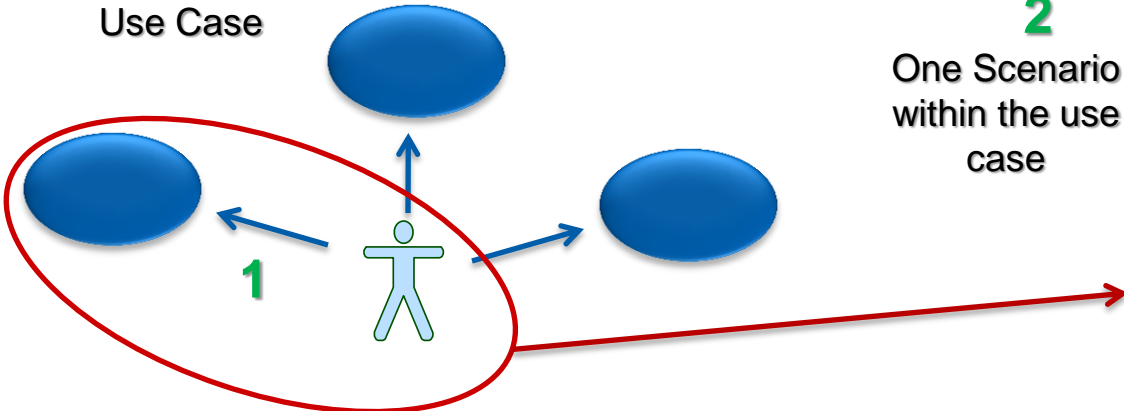
*How does this get done?
Who's creating the PBL?*

The Product Backlog

The list of ranked or prioritized stories to be worked

Use Case to Scenario to User Story

Use Case

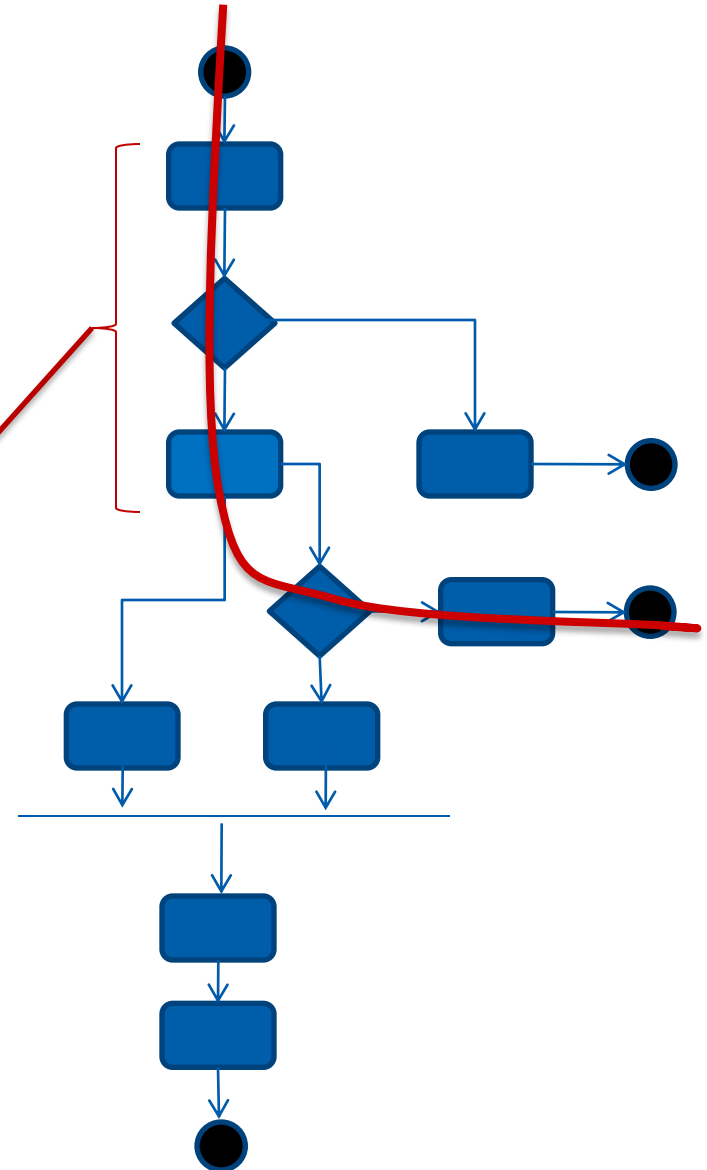


2

One Scenario within the use case

3

A user story is a segment of a scenario

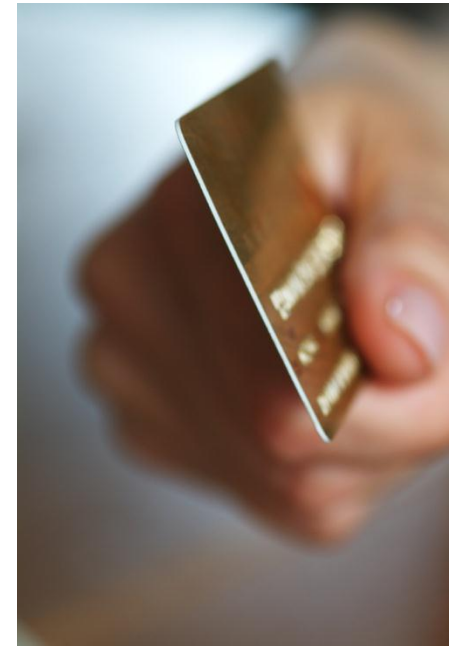


As a [user/system] I want [what] so that [why]....

Example

Requirement: The hotel reservation system shall enable the user to make reservations.

- Use Case: Vacation Planner makes hotel room reservation
- Actor: Vacation Planner
- Success Guarantee: Room reserved; credit card charged
- Main Scenario:
 - Search for discounted rooms.
 - Room identified.
 - Save selection.
 - Vacation Planner selects payment type
 - Vacation Planner submits credit card number, expiration date, name.
 - System validates card
 - System charges credit card
 - Confirmation number provided



Requirement to User Stories

The hotel reservation system shall enable the user to make reservations.

As a premiere member, I want to search for available discounted rooms.

As a vacationer, I want to search for available rooms.

As vacationer, I want to save my selections.

As a vacation planner I want to pay for my reservation with a credit card.

Requirement to User Stories

The hotel reservation system shall enable the user to make reservations.

User Story

As a vacation planner I want to pay for my reservation with a credit card so I can confirm my reservation.

Acceptance Criteria and Verification

Verify:

- Demonstrate with American Express
- Demonstrate with Master Card
- Demonstrate with Visa

What About Other Types of Requirements?

As a vacationer and user of the hotel website, I want the system to be available 99.99% of the time...

As a vacationer, I want web pages to download in <4 seconds...

Non-functional requirements

Performance requirements

Constraints

As the hotel website owner, I want 10,000 concurrent users to be able to access the site at the same time with no impact to performance...

Describes system behavior or characteristics

Product Vision and Roadmap



*Longer term planning
High level capabilities (sometimes written as goals)*

Release Plan, Iteration Plan, Daily Plan

Example: Hotel Website

Requirement 1: The hotel reservation system shall allow the user to make reservations.

Release Plan (Set of User Stories)

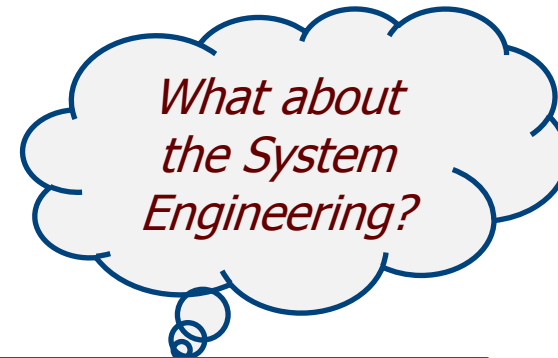
User Stories		Points
As a vacationer, I want to search room availability...	Demonstrate with search on 1 room Demonstrate with search on executive suite....	12
As a vacationer, I want to save my request...	Demonstrate....	8
As a vacationer, I want to pay with a credit card...	Demonstrate...	21

Iteration Plan (User Stories with Tasks)

	Hours
Design Review	4
Install Baseline	4
Documentation	8
Acquire Test Data	8
Code	24
Develop Tests	8
Run Tests	8

The Daily Plan

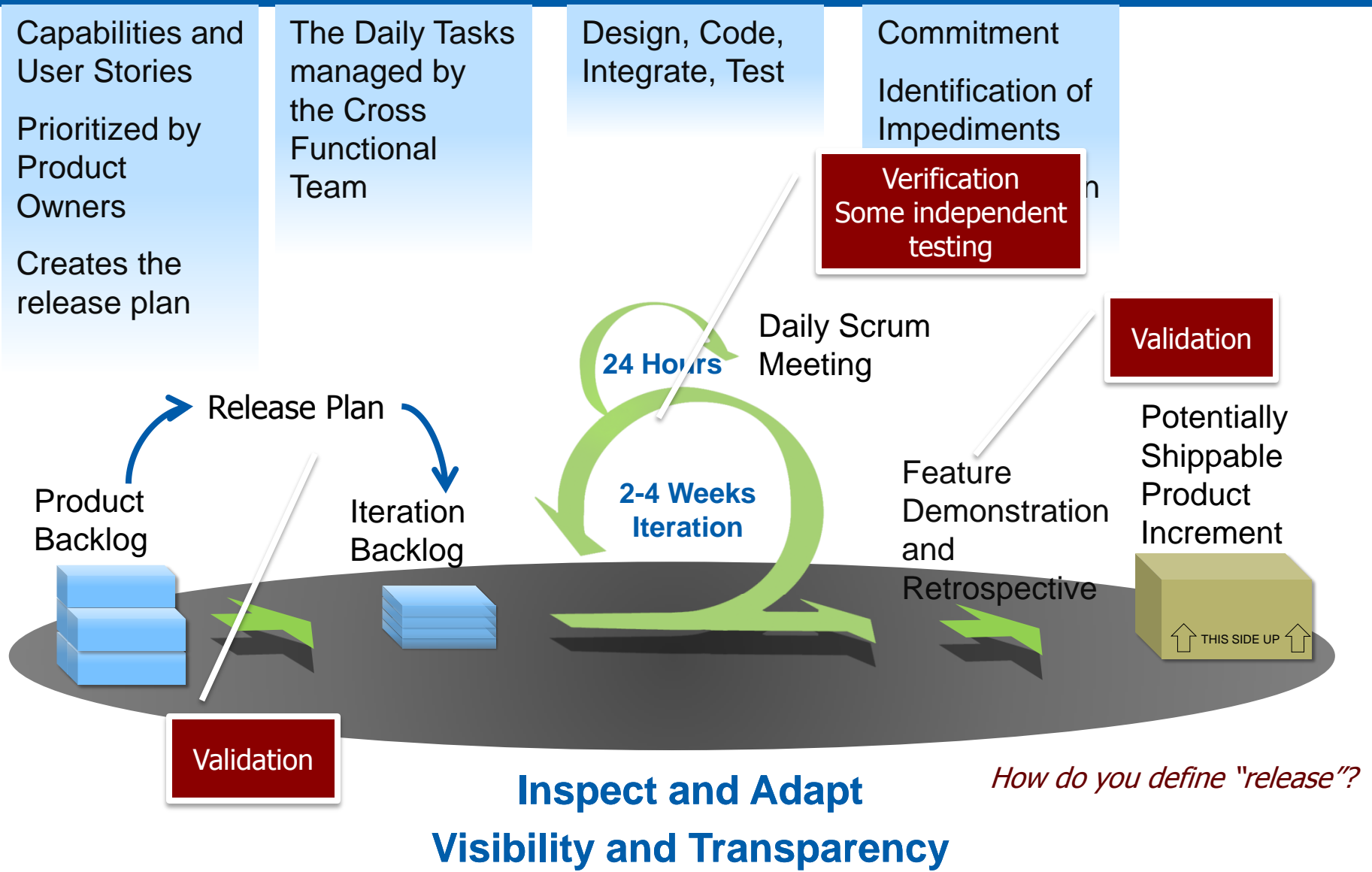
Yesterday I started on the interface....
Today I plan to...
The one thing standing in my way...



Detailed planning saved to the last responsible moment

1. User Stories
2. Tasks under a User Story
3. Preparing for the next iteration

The Agile Framework



Iteration Demonstration and Acceptance

- Transparency and information sharing
- Team presents what it accomplished during the iteration
- Typically takes the form of a demo of new features or underlying architecture
- Time-boxed
- Whole team participates
- Feedback from stakeholders and users
- User Stories validated and accepted
- Metrics updated



*User Story
Validation*

Requirements Mapping

	A	B	C	D	E	F	G	N	O	Z	AA	
1	Project Requirements			Stories								
2	SS ID	Text		Verification Summary	Iteration 1	Iteration 2	FD 0032 (U) As a user I want to browse online help	FD 0077 (U) As a user I want to display Metadata Result on 2D Map	Iteration 3	FD 0005 (U) As an administrator I want to monitor system health and status and derive Availability	Iteration 4	(U) The site for location information
3	SS-19713	(U) The SYSTEM shall provide a detailed online help capability that is context sensitive.	Verified			X						
4	SS-19714	(U) The SYSTEM shall provide an environment to support the online help capability from content providers.	Verified			X						

- Requirement to story mapping

- Requirement to Story to Test to Verification
- Updated each iteration

Requirement ID	Text	Spec Paragraph	Test	Verification Method
SS-19707	(U) The SYSTEM shall collect and maintain metrics on the number of users logged in on the system: Total number, average daily, max/min simultaneously logged on, daily totals of users logged in by organization.	3.1.1.6, (U) Infrastructure	Test Objectives	Inspection
SS-19709	(U) The SYSTEM shall collect and maintain metrics on the number, size and type of queries successfully and non-successfully executed by the system: Number, size and type of queries successfully and non-successfully executed; totals by day/month; average daily numbers; max/min number, max number simultaneously run.	3.1.1.6, (U) Infrastructure	Test Objectives	Inspection

Final Notes

- Requirements Analysis and Design
 - Upfront requirement analysis is done during release planning for that release
 - Uses cases are a tool and provide the greater context for user stories
 - Architecture and initial design are first developed during iteration/release 0
 - Every iteration the team determines if a design review is needed
- Requirements and User Stories
 - High level requirements can be mapped to user stories
 - User stories convey understanding (user, need, why)
 - User stories create the Product Backlog
- Validation and Verification
 - High level requirements have tests and each user story has tests.
 - Validation begins with the release planning phase and occurs again during the iteration boundaries
 - Each story has acceptance criteria that is defined before the release/iteration begins
 - Requirements Traceability/Verification Matrix is updated each iteration

Checkpoint: Our User Story

- User Story
 - As a Systems Engineer I want to have a better understanding as to how Systems Engineering practices align with the Agile framework so I have insight as to how this might be implemented in an Agile environment.
- Acceptance Criteria
 - Communicate how requirements analysis and design occur when using an Agile approach
 - Discuss considerations regarding architecture
 - Understanding of the relationship of requirements to user stories and why user stories
 - Insight into the Agile framework and how it helps manage changing requirements and priorities
 - Identification of verification and validation practices within the context of the Agile framework

References and Recommended Readings

Agile Requirements and Collaboration

Requirements by Collaboration	Ellen Gottesdiener, EBG Consulting
Collaboration Explained	Jean Tabaka, Rally Software
User Stories Applied	Mike Cohn

Agile Development Practices

Agile Software Requirements	Dean Leffingwell
Agile Software Development with Scrum	Ken Schwaber and Mike Beedle
Agile Testing	Lisa Crispin and Janet Gregory
Agile Estimating and Planning	Mike Cohn

The INCOSE SE Handbook V3.2.1 states: “INCOSE use: Permission to reproduce and use this document or parts thereof by members of INCOSE and to prepare derivative works from this document *for INCOSE use is granted*, with attribution to INCOSE and the original author(s) where practical, provided this copyright notice is included with all reproductions and derivative works. Content from ISO/IEC 15288:2002 and 15288:2008 is used by permission, and is not to be reproduced other than as part of this total document.”

NORTHROP GRUMMAN

