Raytheon

Customer Success Is Our Mission

SCRUM : Managing Development on Heterogeneous Systems

Eric Greene

Software Lead, IVC

11/6/2014

Agenda

- Why we use Scrum
- Domain Problem
- Software Problem
- Challenges to Agile
- Evolving Scrum to Meet Challenges
- Conclusion

Why We Use Scrum

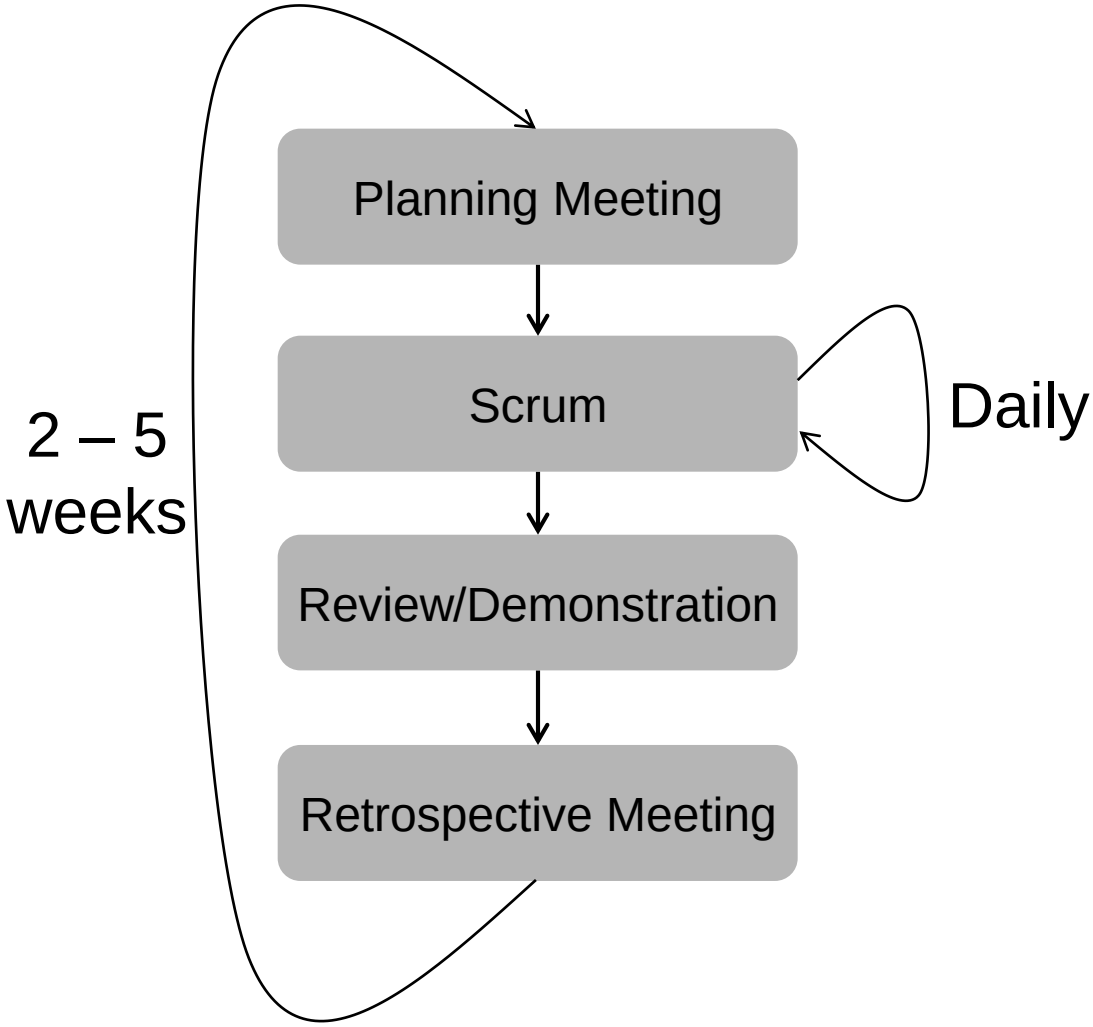
Raytheon

- To improve the speed and adaptability of software development
  - Our customers find themselves in a rapidly changing environment
  - Warfare changes daily
  - Warfighters want advanced, software-enabled features to deal with today's emerging threats
- We are in an extremely competitive environment
  - Technology is rapidly changing
  - Adaptation is a necessity
  - Lightweight processes necessary

Speed and Adaptation are our Greatest Drivers

The Standard Sprint

Raytheon



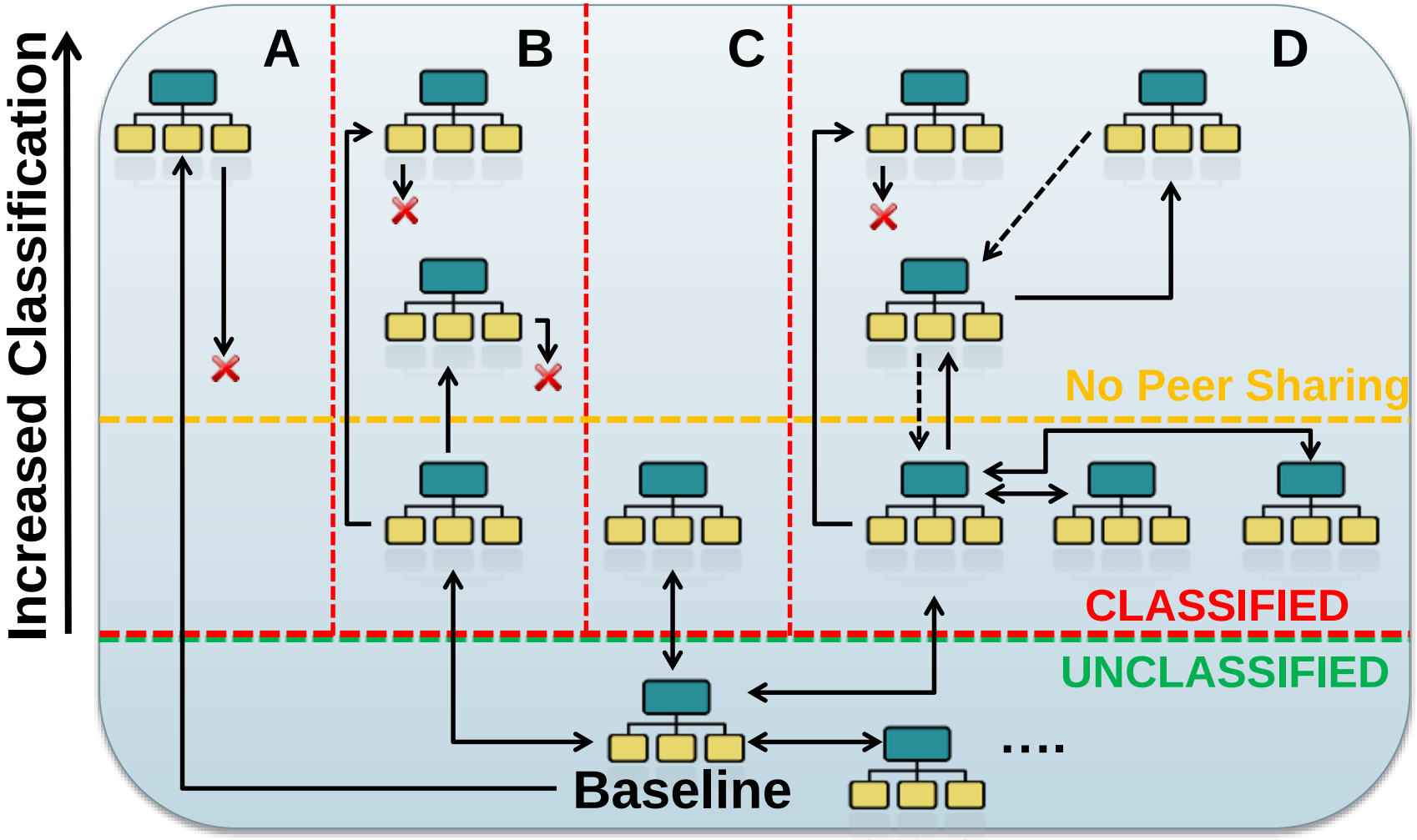
A Normal Sprint Within Scrum

Domain Problem

- Defense contractors face an unusually large challenge implementing heterogeneous solutions
  - Security is *the* key driver for this problem
  - Access can be highly restrictive and can take many months to a year to obtain
- Silo effect - departmental segregation damages communication and solution sharing
- Data/file transfers and downgrades can be difficult or not allowed

Security Greatly Impacts Heterogeneous Systems Solutions

The Classified Programs Scenario



Implementing Enterprise Solutions Can Be Challenging

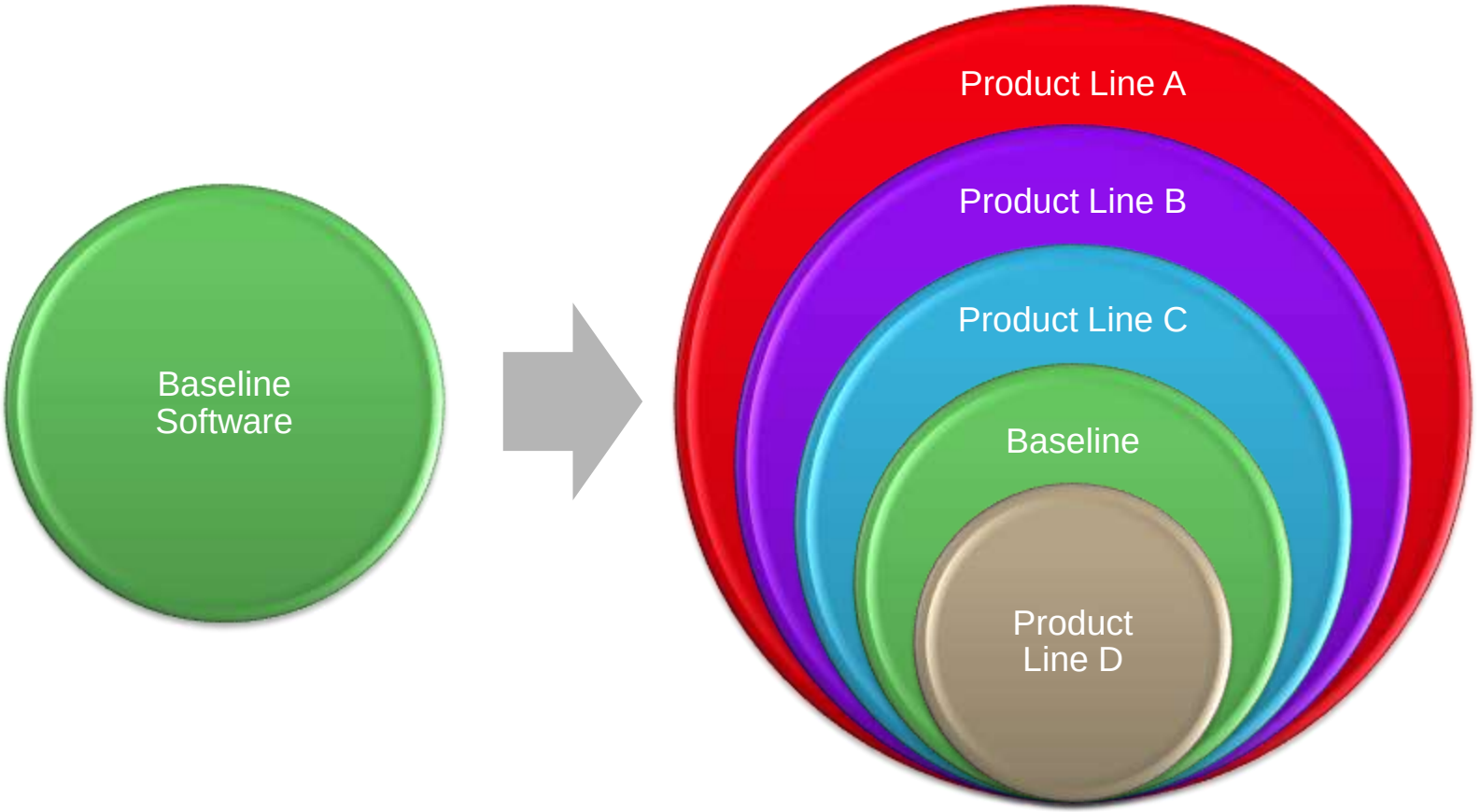
Software Problem

Raytheon

- The domain complexity historically has led to:
  - Multiple solutions to the same problem; single use “homegrown” software, usually with large maintenance risks
  - Some variation of waterfall used to deploy enterprise solutions
  - Common solutions frequently becoming modified to the extent they can be considered distinct, separate software from the predecessor
  - Very expensive maintenance costs or unsupported software

Domain Complexity Impacts Software Methodology Choices

The Software Scenario



Common Software or 5 Different Applications?

Challenges to Agile

Raytheon

- Product ownership
  - Rarely can any single customer fully represent the gamut or understand all requirements
  - Many tasks cannot be mentioned or broken down at a common level
  - Previous implementation attempts have led to multiple planning, retrospective meetings each sprint
- Team cross-functionality
  - Frequently, the development team cannot access all areas
  - There can be a long lead time to gain access
  - Typically, a greater risk of accruing technical debt exists

Truly Heterogeneous Teams are Rare in this Domain

Challenges to Agile (cont.)

Raytheon

- Light Documentation
  - Documentation complexity can frequently grow at a greater than linear rate when many product lines are involved
- Maintenance
  - As seen in slide 8 (*The Software Scenario*), each product line may contain a significantly different final product
  - Each product line may also use different configuration management software, release processes, etc.
  - Tailored releases can be very costly when in frequent increments

Challenges to Agile (cont.)

Raytheon

- Other Considerations
  - Product lines may have vastly different software/hardware platform availability
    - Linux or Windows only
    - Java restricted domains
  - Not all product lines are equal - new technology can require years for approval, hardware can be aged

"Out-of-the-box" Scrum Can Actually Lead to Higher Dev Costs

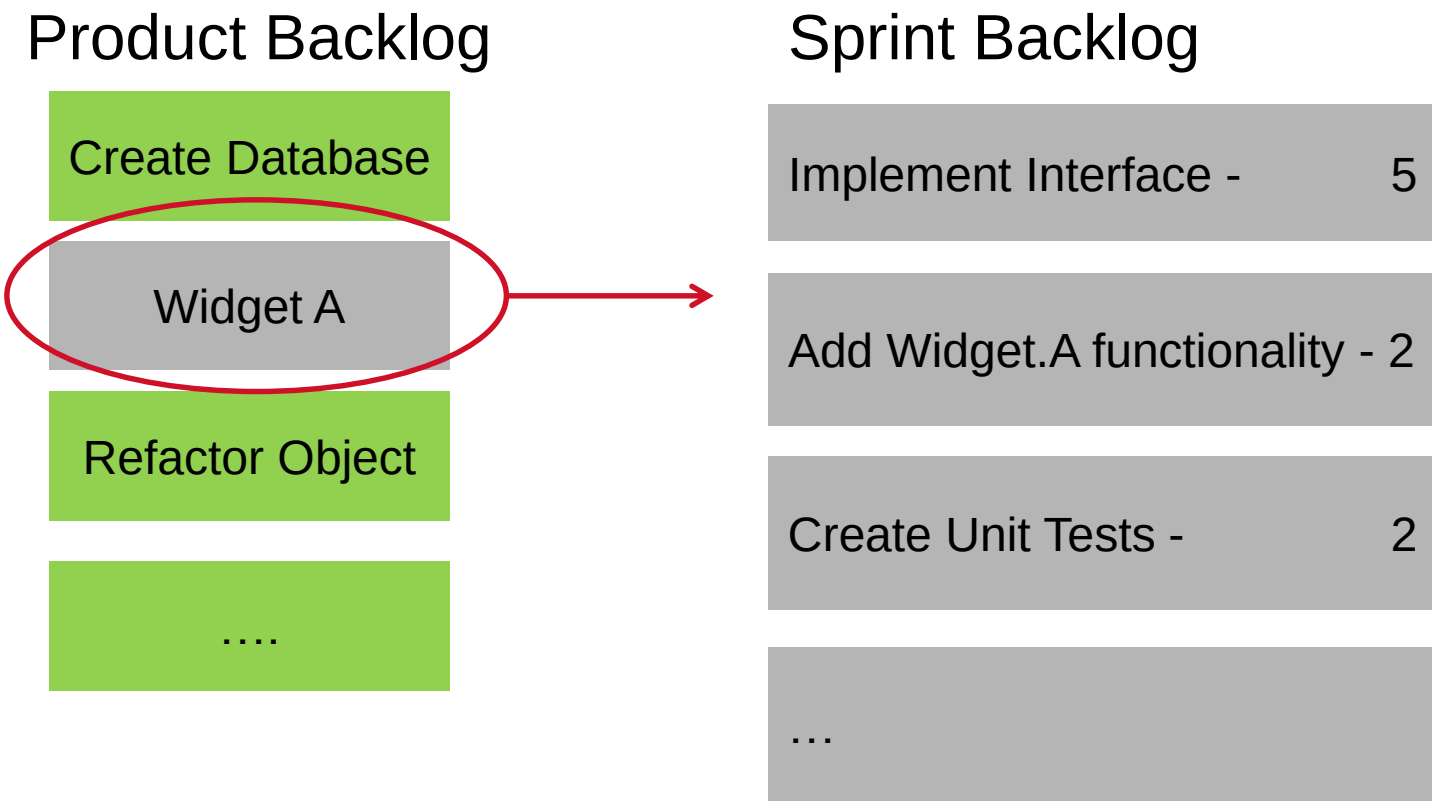
Evolving Scrum

Raytheon

- Product ownership
  - Product ownership is the most important element of Scrum - How can Scrum fidelity be upheld without clear ownership?
  - *We use a power group of vested stakeholders to represent the product line gamut*
- User stories
  - Previously, planning meetings and retrospection was held at each area due to this – for a small team this is unnecessarily burdensome and should be avoided
  - *Use of generic user stories is required, with a defined breakdown pattern approved by the customer*

Power Stakeholders, Generic User Stories

Evolving Scrum (cont.)



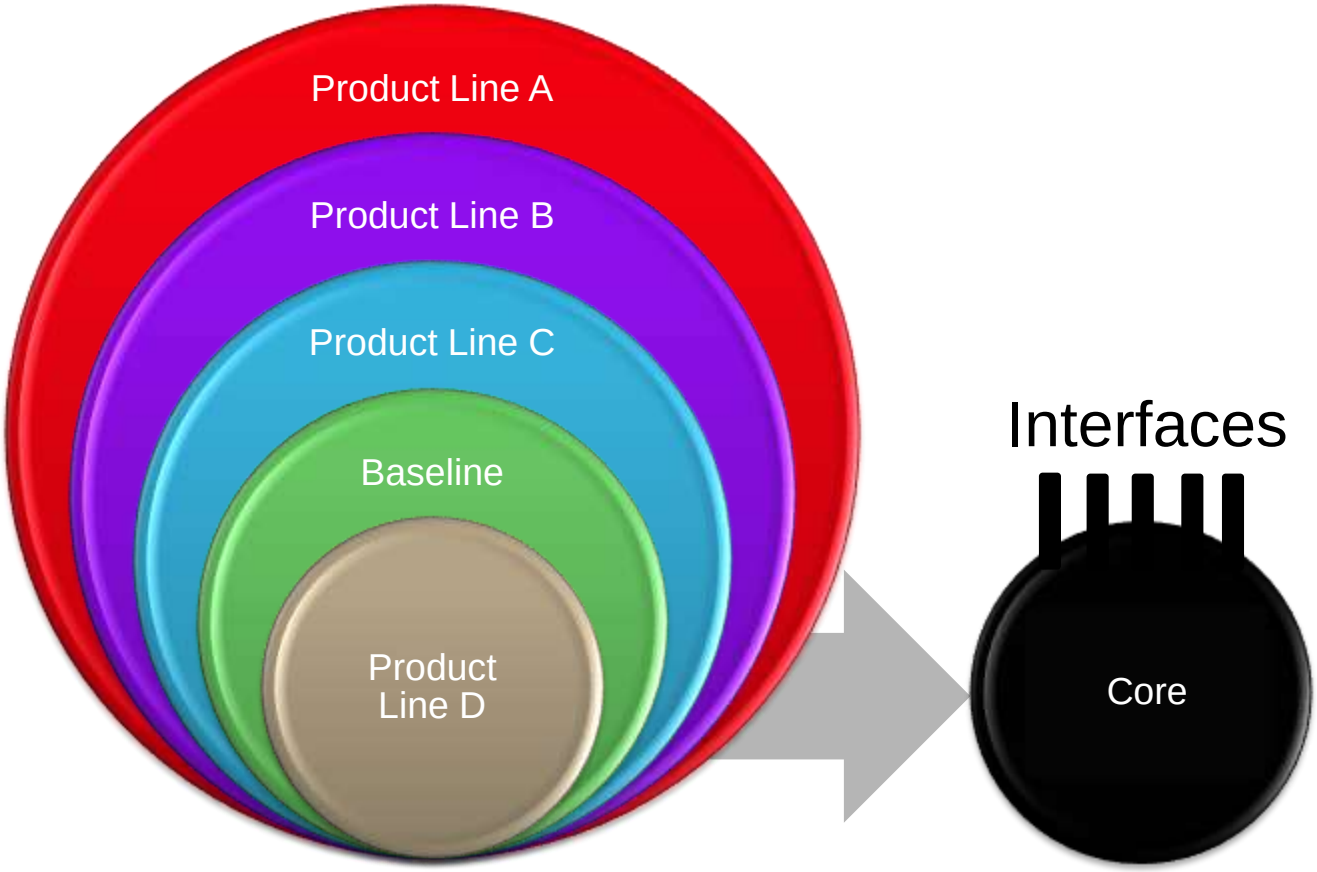
Using Generic User Stories

Evolving Scrum (cont.)

Raytheon

- Preserving Heterogeneous Teams
  - Baseline area should be used to host as much development as possible – use good object oriented principles and interfacing
  - Final architecture should be understood across team
  - Pair up for special tasks when necessary
- Maintenance
  - Enterprise dedication to common toolsets is necessary, especially configuration management tools
  - If not adequately planned for at the beginning of the project, a significant maintenance cost will be incurred
    - Creation of core architecture is key

Core Architecture



Key Point: Identification of Core Product

Evolving Scrum (cont.)

Raytheon

- Core documentation
  - Created at baseline level before sprint 1 - major revisions only when new product lines are added to project
  - This should be reviewed and updated at the end of sprint before the retrospective meeting
  - Each area should contain a description document for interface to the core
- Other artifacts
  - Detailed explanation of generic user stories usually is required at the applicable area
  - Other documentation should be light and process specific

Conclusion

Raytheon

- Development for heterogeneous systems in the defense domain requires more rigor than allowed under agile models
- A greater than normal focus on architecture at the beginning and throughout the lifecycle provides key information for our development teams
- Modification of elements in scrum allow us to implement a light-weight methodology in a very constrained environment
- We believe our system's complexity can be mirrored in other technological domains and our considerations can be extended past the defense industry