

Agile Surveillance Points

**2012 NDIA
Systems Engineering Conference
San Diego, CA**

Dick Carlson

Richard.Carlson2@Boeing.com

Top 5 Engineering Issues as of 2010

1. Increasingly, urgent demands of the warfighter are requiring effective capabilities to be fielded more rapidly than the conventional acquisition processes and development methodologies allow.
2. The quantity and quality of Systems Engineering expertise is insufficient to meet the demands of the government and defense industry
3. Systems engineering practices known to be effective are not consistently applied or properly resourced to enable early system definition
4. Technical decision makers do not have the right information & insight at the right time to support informed & proactive decision making to ensure effective & efficient program planning, management & execution.
5. The development of systems with a full level of integrity (all technical aspects considered) is longer and more expensive over the entire lifecycle as the technical solution is iterated and reworked in later stages of the development.

Source: NDIA Systems Engineering Division Task Group Report, May 2011

Agile Systems Engineering: An Oxymoron?

- **SE teams implement *Scrum** as their method for project management (as most software teams do)**
- **There are no *Agile Systems Engineering* methods**
 - *Agile*** methods are used by teams to develop software
- **Systems Engineering processes can be adapted to accommodate Scrum project management practices**
 - SE teams using the Scrum framework must tailor their engineering / organizational standard processes
 - Potential improvements to these processes occurs at the completion of each iteration

*Source: *Scrum Alliance*, scrumalliance.org/learn_about_scrum

**Source: *Agile Manifesto for Software Development*, agilemanifesto.org/

Principles and Practices of Agility

Principles:

Customer Satisfaction	Embracing Change
Frequent Delivery	Collaboration
Motivated Team	High Bandwidth
Working Software	Sustainable Pace
Technical Excellence	Simplicity
Emergent Design	Continuous Improvement

Practices:

Close customer collaboration	Short iterations
Daily stand-up meetings	Prioritized requirements
Planning	Product demos/reviews
Estimating	Self-organized teams

Source: *Dick Carlson*

SE Iteration Activities

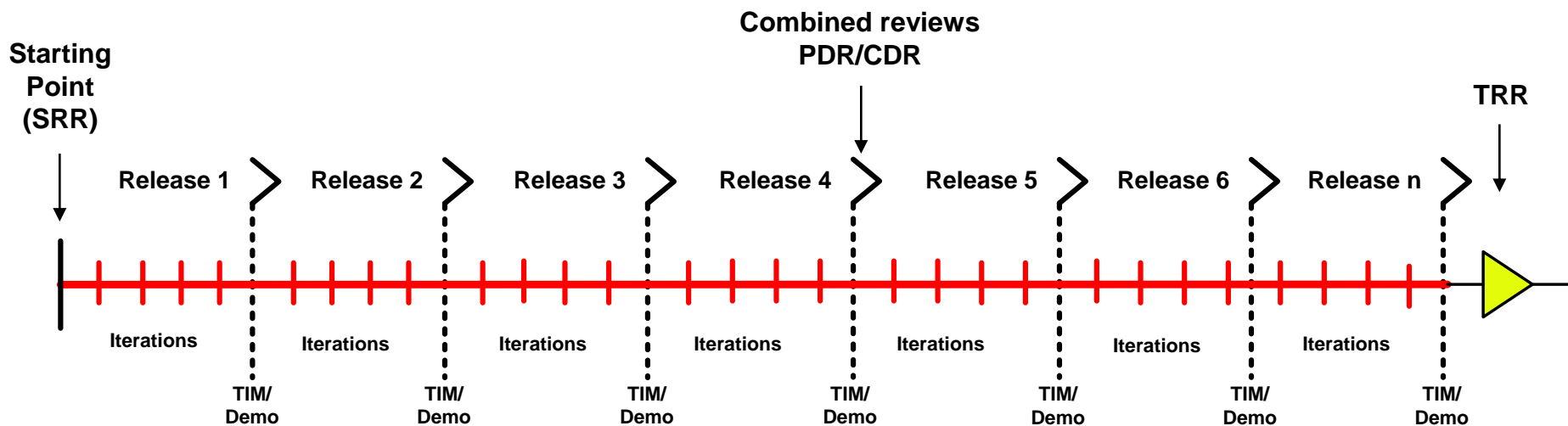
- Every iteration starts with a planning session
 - Goals are established and commitments are made
 - Determines all necessary work, and break the work down in small chunks
 - Daily Scrum meetings are planned
- Work throughout the iteration is focused and intense
 - Multi-tasking (WIP) is limited and discouraged
 - Issues and risks are identified and solutions implemented
- Each iteration concludes with an informal review
 - Iteration reviews provide an opportunity for stakeholders to see the incremental product development
- Iteration concludes with a retrospective
 - To identify lessons learned, establish best practices, and opportunities for improvement

SE Release Activities

- A release review is conducted at the end of every iteration series for the customer, users, and other relevant stakeholders to see
 - Best opportunity for customers to observe completed activities and work products
 - May include a brief technical interchange meeting to review what was completed
 - A demonstration of working prototypes completed
 - Summarizes activities completed by the team
 - Identifies strategic metrics (e.g. productivity, effort, defects, schedule, etc.)
 - Identifies areas of improvement

Surveillance Points

- The completion of each iteration serves as an observation point (e.g. demo/review) for key project stakeholders to observe incremental product development and provide feedback and guidance to the team
- Releases are scheduled surveillance points in a project lifecycle where the customer and users can monitor completed system functionality and provide valuable feedback to the team



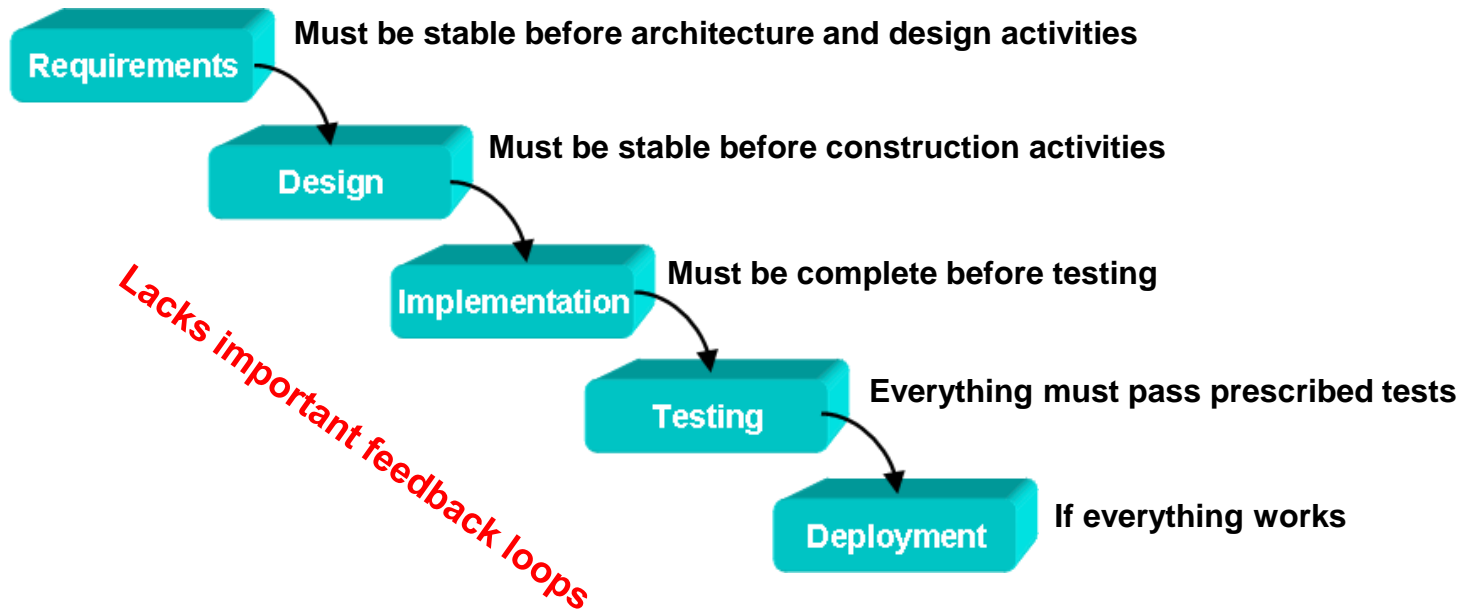
Source: Dick Carlson

Tyranny of the Waterfall Model

- Development is done in isolation from systems engineering, integration, or test
- Heavy overhead for technical reviews and programmatics
- Aggravates complexity overload and induces analysis paralysis
 - Large steps with overwhelming degrees of complexity are attempted
- Pushes many high-risk and difficult elements toward the end of a project
- Does not accommodate uncertainty or changing requirements well
- Does not yield a working version of the product until late in the process
 - Problems are discovered late
- No intrinsic feedback loops

Source: *Agile and Iterative Development: A Manager's Guide*, Larman [2004]

Classic Waterfall Model



Source: *Dick Carlson*

Classic Waterfall Milestones

- Assumes stable requirements and design early on
- Focuses heavily on documents and formal reviews
 - Docs: SEMP, SSS, OCD, SSDD, SRS, SDD, STP, etc.
 - Reviews: SRR, PDR, CDR, TRR, FQT
- Scope change results in significant impacts to design and construction
- Not required by DoD 5000



Application of Surveillance Points

- Supports incremental development
 - Focuses on functional product
- Each iteration finishes with additional functionality
 - Allows stakeholders to see incremental development progress
- Each release includes demonstrable functionality
 - Elicits customer/stakeholder feedback
 - Accommodates well to scope growth
- Not prohibited by DoD 5000

Where to start?

- Create a storyboard that includes:
 - A product vision
 - A product roadmap
 - A comprehensive release plan
 - The details of the new SE approach
- Negotiate terms of the contract with your customer
 - Results in their buy-in
- Teach your customer the benefits of the new approach and how it will be deployed



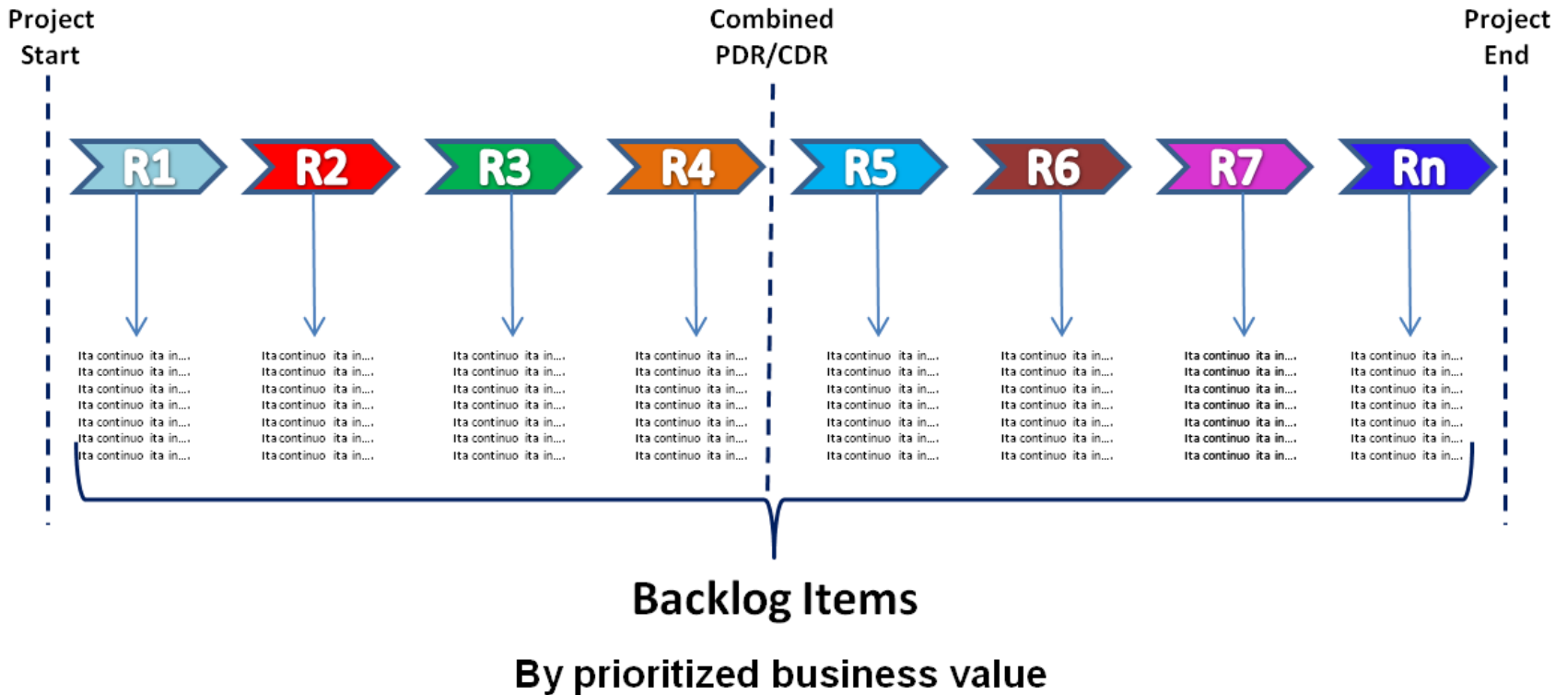
Establish a Product Vision

One that encourages and supports:

- Customer Satisfaction
- Frequent Delivery
- Motivated Team
- Working Software
- Technical Excellence
- Emergent Design
- Embracing Change
- Collaboration
- High Bandwidth
- Sustainable Pace
- Simplicity
- Continuous Improvement



Project Roadmap



Source: *Dick Carlson*

Release Planning

- Includes a release plan that speaks like a storyboard
 - Identifies details of the approach and functionality to be implemented
- Emphasizes importance of customer involvement and feedback
- Requires customer approval
- Helps transition away from milestone reviews



Negotiate With Your Customer

- Discuss contractual reporting requirements with customer and program office
 - Technical Progress
 - Earned value
- Show how surveillance points can satisfy those reporting requirements with their active participation
- Work with program office and contracts organization to formalize agreements



Teach Your Customer

- Let your customer know there is a viable alternative to the Waterfall model (3 ways to do this)
 - Respond to a proposal need for leaner methods
 - Pre-proposal – recommend an “agilistic” approach
 - Post-proposal – provide examples and show them how
- Train/coach your customer on how agility can be deployed
- Encourage your customer to actively participate in development validation—their feedback is vital!



Propose the Approach

- Plan a release strategy where each release lifecycle includes a series of short iterations, where completed system functionality can be observed by the customer, users, and other key stakeholders
- Use *Agile Modeling** techniques and functional prototypes
- Conduct daily standup meetings to optimize communication
- Conduct iteration reviews with stakeholders to demonstrate incremental product development
- Conduct release reviews to verify all completed work
- Conduct retrospectives to identify lessons learned and opportunities for improvement

*Source: <http://www.agilemodeling.com/>

Keep Everything Transparent

- Announce all activities to the greater community
- Measure performance, productivity and quality
- Keep all work visible to everyone
- Improve continuously
- Celebrate successes!



Don't do it alone

- Find an experienced coach to assist during project planning and initial implementation
- Train team and the organization on Scrum implementation
- Determine if using Scrum will add or reduce risk
 - Is the customer willing to actively participate, i.e. requirements clarification, product demos, incremental development?
 - Will there be regular and frequent customer communication?
 - Would the customer be willing to drive and prioritize requirements?
 - Is the customer willing to provide frequent feedback on items completed?



Thank You!

Dick Carlson
Richard.Carlson2@Boeing.com