



WELCOME



How Do You Use Agile Methods on Highly-Critical Systems that Require Earned Value Management?

Phyllis Marbach, INCOSE LA Chapter President; Senior Software Engineer at Boeing – Retired

June 6 | 1:00 PM ET | [REGISTER NOW](#)

- ❑ Today's session will be recorded.
- ❑ An archive of today's talk will be available at: www.sercuarc.org/serc-talks/
- ❑ Use the Q&A box to queue questions, reserving the chat box for comments, and questions will be answered during the last 5-10 minutes of the session.
- ❑ If you are connected via the dial-in information only, please email questions or comments to Ms. Mimi Marcus at SERCTalks@stevens.edu.
- ❑ Any issues? Use the chat feature for any technical difficulties or other comments, or email SERCTalks@stevens.edu.



The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense, ASD(R&E), nor the SERC.

No Warranty. This Stevens Institute of Technology Material is furnished on an “as-is” basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

This material has been approved for public release and unlimited distribution.



How Do You Use Agile Methods on Highly-Critical Systems that Require Earned Value Management?

Phyllis Marbach

Engineering Consultant

6 June 2018

Contents

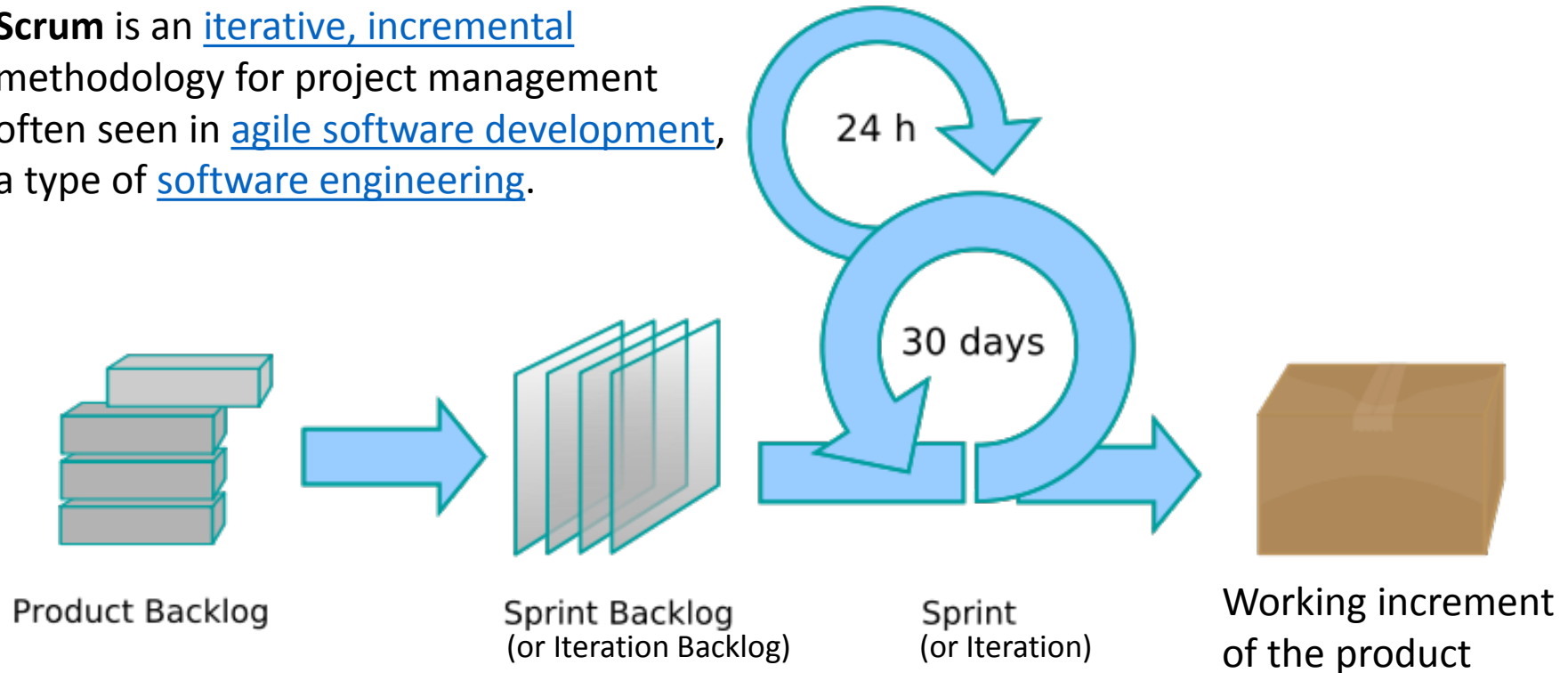


- Introduction to Agile (Scrum)
- Agile Systems Engineering (SE) Framework
- Differences in Agile and Traditional SE
- Agile Earned Value Management (EVM)
 - Planning
 - Implementing
 - Measuring
- Highly-critical systems and Agile
- The Government Accounting Office (GAO) and Agile
- References: see the NDIA white paper

Introduction to Agile (Scrum)



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



Copyrights specified as freely licensed media http://en.wikipedia.org/wiki/File:Scrum_process.svg

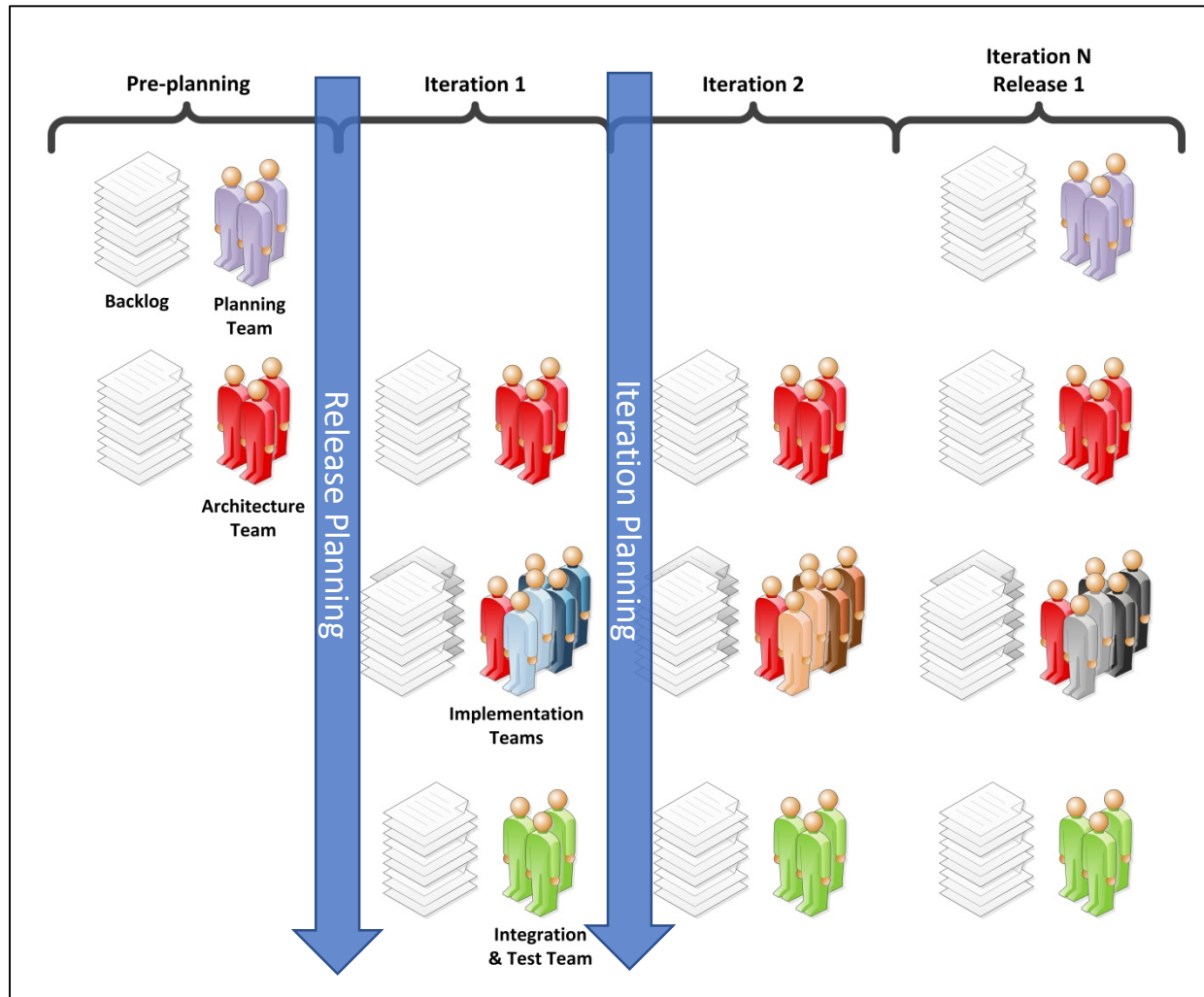
Agile Includes:



- Scrum: Iterative, incremental methodology for project management
- Lean: reduce waste, reduce Work in Process (WIP); have what you need to start; once an item is started finish it before starting the next item;
- eXtreme Programming: pair programming, continuous integration, automated testing, test first development,
- Crystal Methods: frequent delivery, reflective improvement, daily stand-up meetings, side-by-side programming, burn charts, automated tests, configuration management, frequent integration

Agile is often called Lean-Agile

Agile Systems Engineering (SE) Framework



Copyright 2013, 2014 © by Larri Rosser, Phyllis Marbach, Gundars Osvalds, David Lempia. Permission granted to INCOSE to publish and use.

Ref: Systems Engineering For Software Intensive Projects Using Agile Methods.

Agile SE Framework

Differences from Traditional SE



- Changes to the architecture – Flexible, modular architecture framework; rather than having Big Design Up Front (BDUF)
- Changes to the process – iterative, incremental rather than traditional or sequential
- 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.

Copyright 2013, 2014 © by Larri Rosser, Phyllis Marbach, Gundars Osvalds, David Lempia. Permission granted to INCOSE to publish and use.

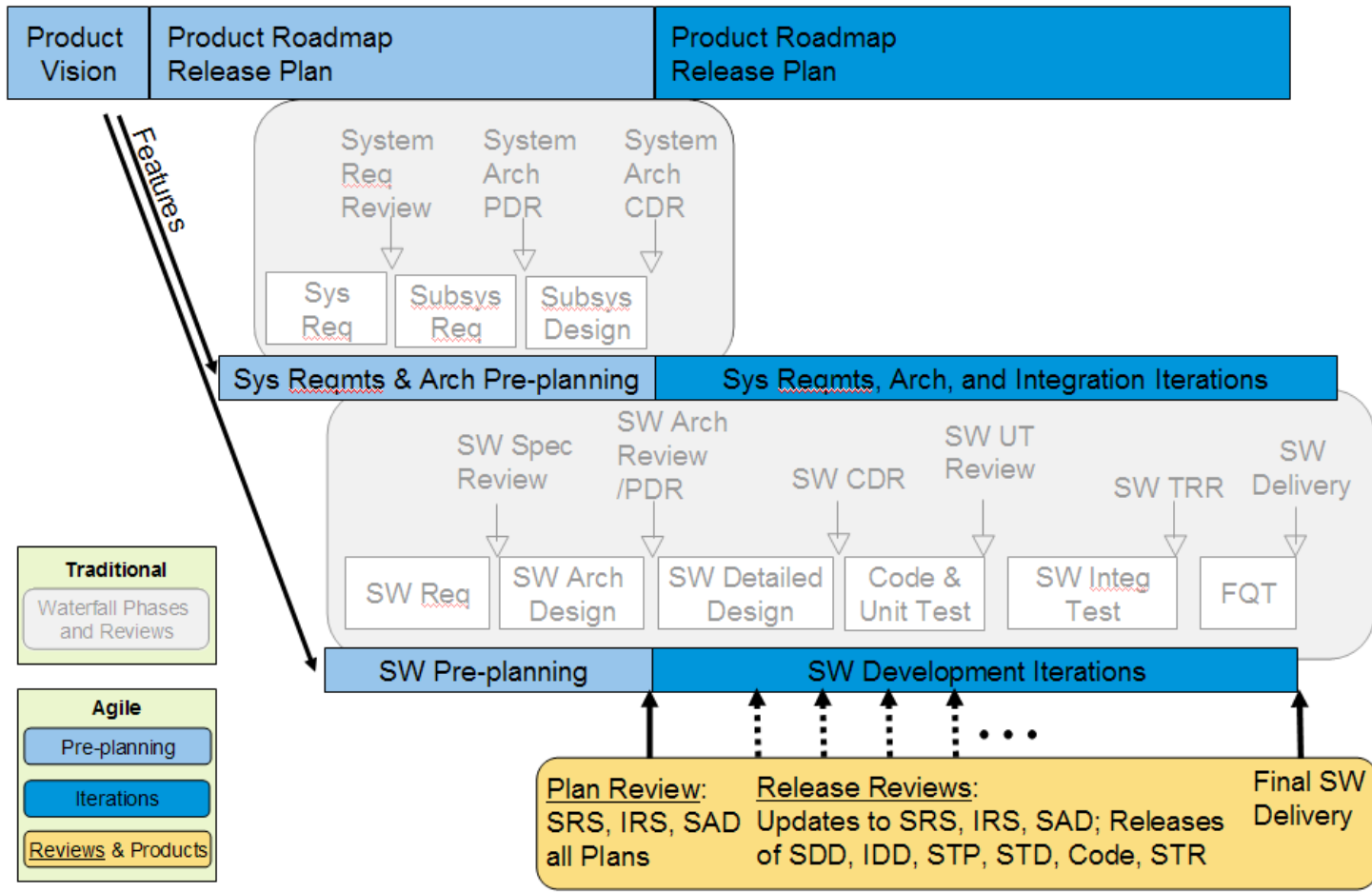
Agile SE Framework Summary



- SE and SWE 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
- Demonstrate working products frequently
- Absorb changes to mission requirements
- Artifacts such as requirements, architecture, system design and verification traceability are produced incrementally

Copyright 2013, 2014 © by Larri Rosser, Phyllis Marbach, Gundars Osvalds, David Lempia. Permission granted to INCOSE to publish and use.

Agile and Customer Milestone Reviews

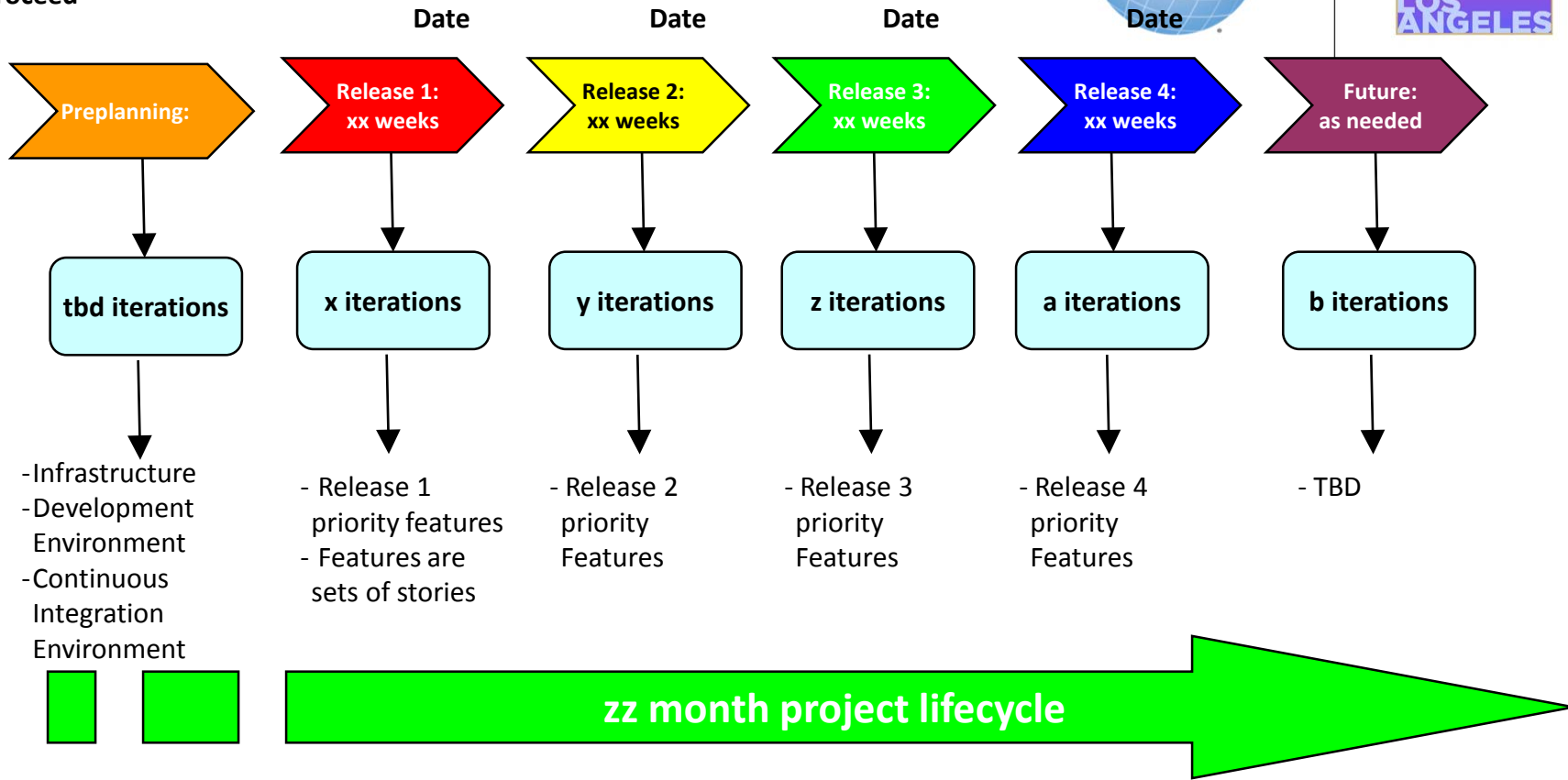


- Traditional**
 - Waterfall Phases and Reviews
- Agile**
 - Pre-planning
 - Iterations
 - Reviews & Products

Once the plans are in place, the product backlog iterations begin

Agile Roadmap Template

Authority To Proceed



Projects using agile practices should have an identified period of performance. During that period of performance releases are defined that have specific features or capabilities. This is called a Roadmap.

Estimate the Project for the Period of Performance



- Initial planning estimates Features in story points
- You may not know the specific stories to create each Feature in the beginning
- Story points are relative to each other
- Find the Feature that is the easiest or fastest to do
- Give it 1
- Estimate rest of Features relative to that 1 using
 - Powers of 2: 1, 2, 4, 8, 16, 32, 64, 128, 256, . . . or
 - Fibonacci Series: 1, 2, 3, 5, 8, 13, 20, 40, 100, . . .

Estimate Features and Prioritize



Feature	Story Points
Feature 1	1
Feature 2	40
Feature 3	100
Feature 4	20
Feature 5	256
Feature 6	13
Feature 7	8
Feature 8	5
...	...
Feature xx	256



Feature	Story Points
Feature 1	1
Feature 8	5
Feature 4	20
Feature 7	8
Feature 6	13
Feature 2	40
Feature 3	100
Feature 5	256
...	...
Feature xx	256

**Release
1**

Features are ordered into the roadmap based on milestones/releases to meet customer expectations. Use Rolling Wave Planning for later releases.

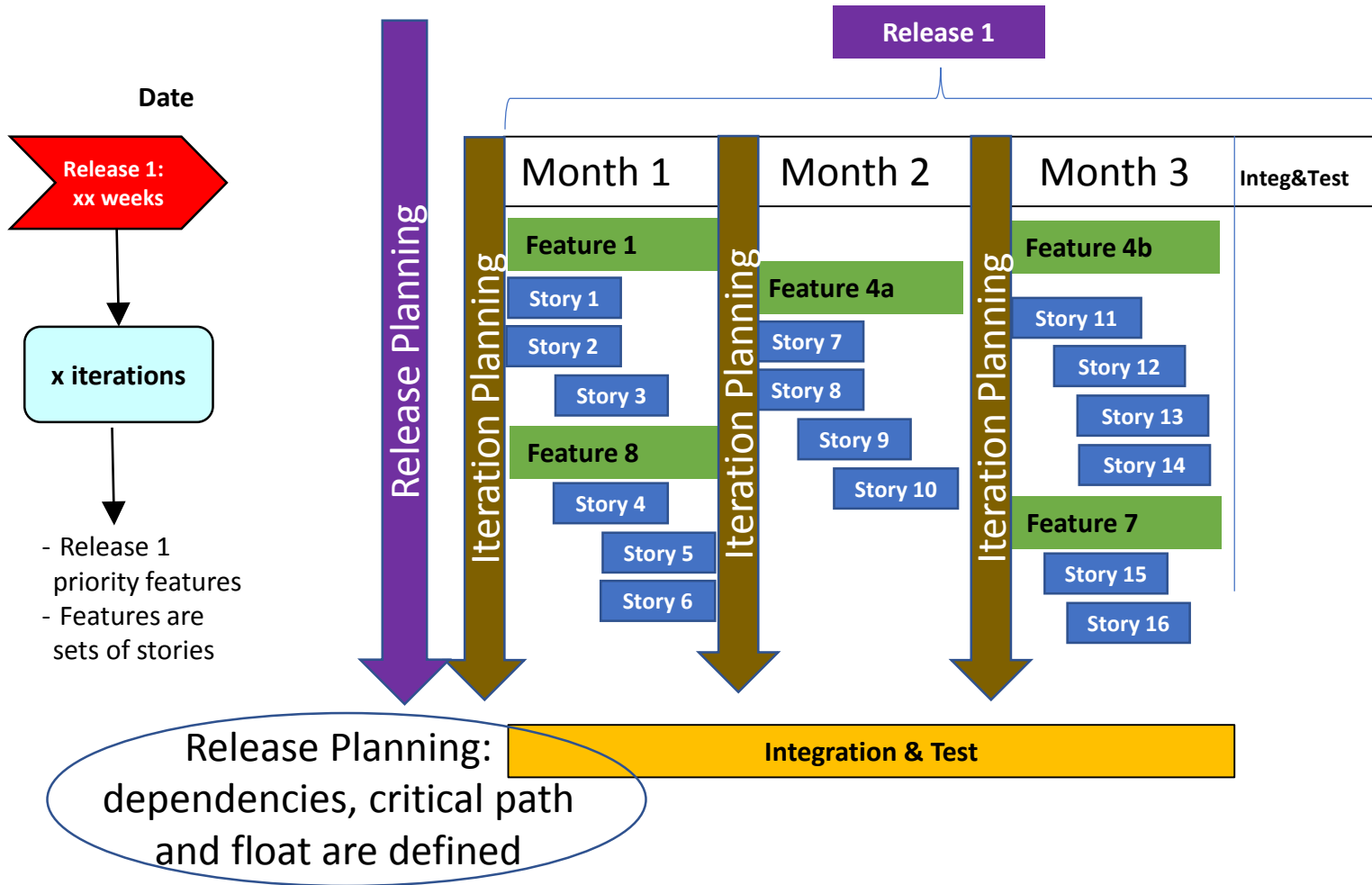
Agile and Earned Value Management (EVM)



- Each software release is a milestone deliverable
- Progress reported weekly on the % complete story points for each feature's planned total story points
- An EVM work package (WP) is one or more Features
- Story points are the estimated effort to complete a backlog item, or user story
 - Example, some features might have a total of 120 story points and 10 user stories
 - Others might have 20 story points and 2 user stories
- Business value is in the completion of the Features (WP)
- A release might have a partial feature delivered that will be finished in the next release (Feature 4a and 4b)

Features are measured in story points, a measure of effort or complexity

Roadmap provides the big picture - Details come during Release and Iteration Planning



Earned Value Management (EVM)



- Earned Value Management (EVM) is a management system that integrates scope, schedule, cost and performance policies, procedures and processes by which programs manage work.
- Helps project managers measure project performance and progress in an objective manner.
- Project Plan – identifies work to be accomplished
- Planned Value (PV) – Valuation of planned work, budgeted cost for work scheduled (BCWS)
- Earned Value (EV) – Metrics that quantify work accomplished, budgeted cost for work performed (BCWP)
- Actual Cost of Work Performed (ACWP) is cumulative hours and/or cost to time now for work.

Budgeted Cost of Work Performed (BCWP)



The value of work completed at a given point in time, synonymous with “Earned Value”. Expressed in dollars representing the portion of the budgeted value of the baseline Work Package that is complete.

Example Formula: $BCWP = \% \text{ Complete} * BAC$, where $\% \text{ Complete} = \text{completed Work Package Story Points} / \text{current total Work Package Story Points}$.

Note: the Work Package baseline budget (BAC) of the calculation does not fluctuate if the team adds or removes Stories / Story Points provided that the baseline Work Package scope does not change. Changing the total Story Points in a Work Package would impact the $\% \text{ Complete}$ recognition but does not require a baseline change.

Example: BAC WP for WP xyz has a baselined estimated cost of \$100,000. The team targeted to deliver 33 of 100 Story Points by the end of Sprint n for WP xyz. At time now the team delivered 28 Story Points. The BCWP is: $100,000 * (28/100) = \$28,000$

The $\% \text{ complete}$ is: $(28/100) * 100 = 28\%$

Cost Variance



A metric for cost performance on a program. It is the difference between budgeted cost of work performed and actual cost ($CV = BCWP - ACWP$.) A positive value indicates a favorable position and a negative value indicates an unfavorable condition, and is expressed in dollars or as a percentage. The calculation for CV dollars is the difference of $BCWP - ACWP$. The calculation for CV percentage is the $((CV \text{ divided by } BCWP) * 100)$.

Example: The team targeted to deliver 33 of 100 Story Points weighted Story Value by the end of Sprint n for WP xyz. At time now the team delivered 28 Story Points of weighted Story Value. ACWP cum to date for WP xyz is \$25,000 at end of Sprint n.

The Cost Variance, CV: $(100,000 * (28/100) - \$25,000 = \3000

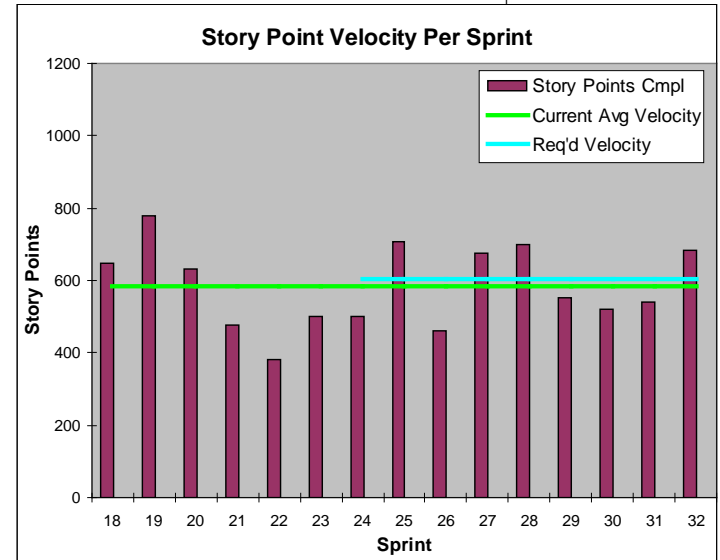
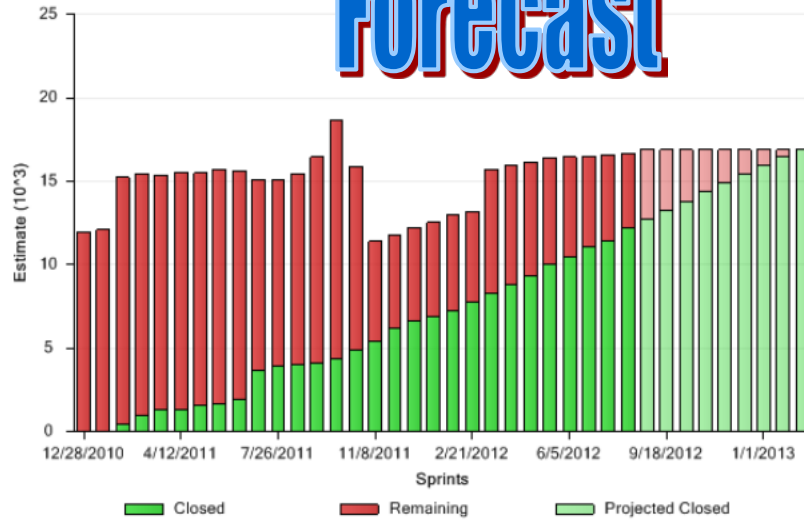
$CV \% = (\$3,000 / \$28,000) * 100 = 11\%$

From: An Industry Practice Guide for Agile on Earned Value Management Programs, Version 1.1, March 31, 2017, NDIA IPMD page 31.

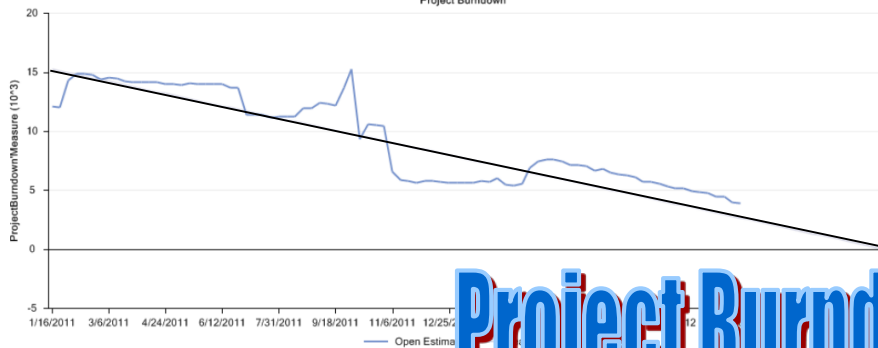
Metrics used to track progress



Forecast



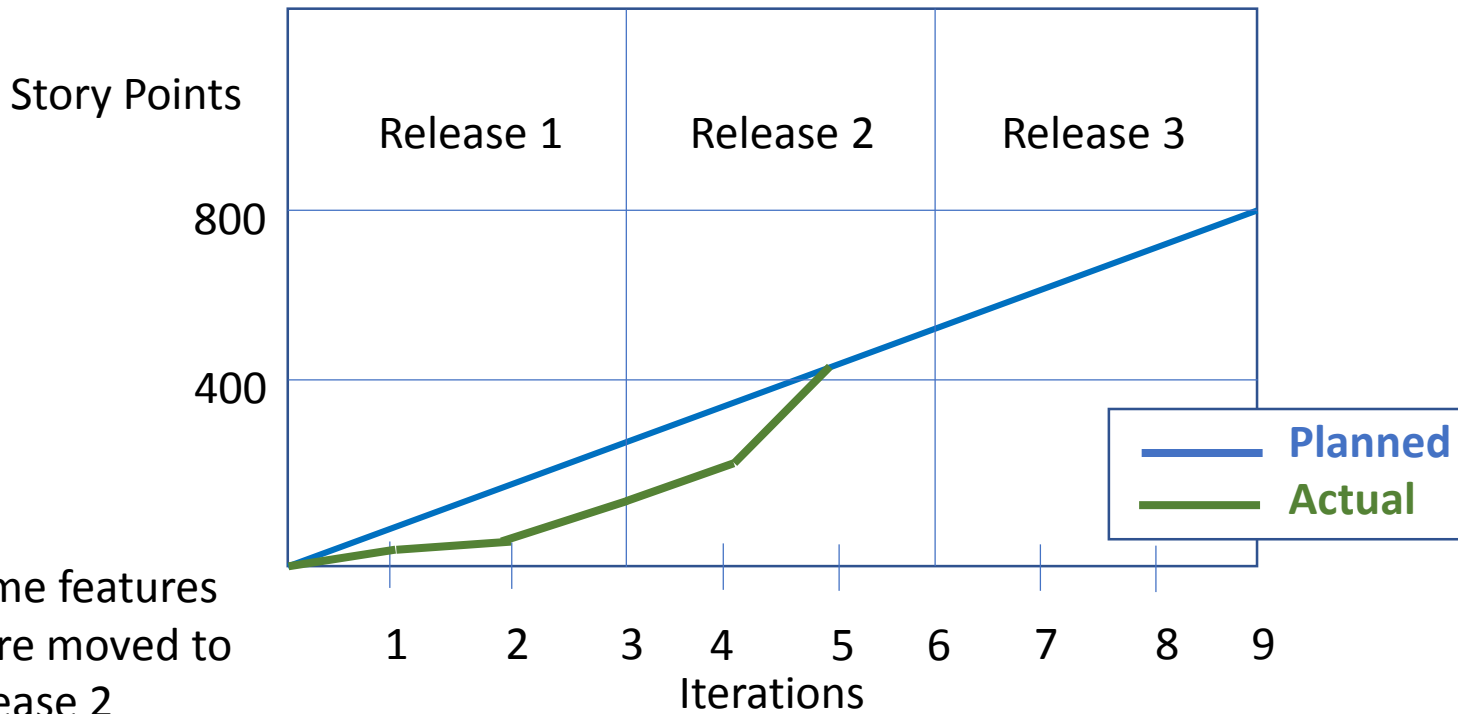
Velocity



Project Burndown

Customer attends the demonstrations every 3 weeks and has visibility of the detailed progress

Reporting story points complete each iteration versus planned



Some features were moved to release 2

Percent complete reporting of that feature, the high value work product in development, can be used for the EVM measures.

Highly-Critical Systems



- A system that must be highly reliable, that retains this reliability as it evolves without incurring prohibitive costs
- Types of critical systems
 - Safety critical: avoid loss of life, serious injury, or damage to environment
 - Mission critical: avoid loss of ability to complete the mission objectives
 - Business critical: avoid economic costs
 - Security critical: avoid loss of sensitive data
- Examples of highly critical systems:
 - Vehicles: air, ground, sea, space, unmanned and manned
 - Defense systems: communications, missiles,
 - Manufacturing systems: chemical, industrial,
 - Infrastructure, Transportation, Health Care

Using Agile practices as highly-critical systems are developed will enable the system to evolve

Government Accounting Office (GAO)



- GAO is the watchdog for congress to make sure projects are using best practices when spending government funding
- The customer for these projects is really the American taxpayer
- GAO Schedule Assessment Guide: Best Practices for Project Schedules, GAO-16-89G, Feb 16, 2018
- GAO Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Capital Program Costs, GAO-09-3SP, Mar 2, 2009

<https://www.gao.gov>

Government Accounting Office (GAO)



- Armed forces: Army, Air Force, Navy, Marines, US Coast Guard; Veterans
- Bureau of Land Management
- Congress Legislative
- Dept of Agriculture
- Dept of Education
- Dept of Energy
- Dept of Health and Human Services
- Dept of Homeland Security
- Dept of Housing and Urban Dev.
- Dept of Interior
- Dept of Justice
- Dept of Labor
- Dept of Treasury
- Dept of Transportation
- Environmental Protection
- FBI, FAA, FEMA,
- NASA
- National Science Foundation
- US Customs and Border Protection
- US Postal Service

<https://www.gao.gov>

GAO Expert Meeting

17 March 2015

GAO Schedule Guide Agile Appendix



Best Practice	Agile Artifacts & Documentation
1. Capture All Activities	<ul style="list-style-type: none">• Roadmap with Prioritized Must Have Features developed with input from stakeholders and SMEs• Roadmap linkage to SOW• Prioritized product backlog consisting of epics, features, and stories• Product Backlog
2. Sequence Activities	<ul style="list-style-type: none">• Kanban Board (or similar)• Dependencies for sprints can be captured in the schedule
3. Assign Resources	<ul style="list-style-type: none">• Kanban Board (or similar)• Small teams of 5-9 people• Scrum master coaching• Training in Agile method
4. Establish Durations	<ul style="list-style-type: none">• Sprint durations consistent and between 1-3 weeks
5. Ensure Horizontal and Vertical Traceability	<ul style="list-style-type: none">• Vertical<ul style="list-style-type: none">○ Roadmap○ Product Backlog (Themes, epics, features, stories)○ Burn down / burn up charts• Horizontal<ul style="list-style-type: none">○ Kanban Board (or similar)○ Releases included in Program Schedule

GAO Expert Meeting

17 March 2015

GAO Schedule Guide Agile Appendix



Best Practice	Agile Artifacts & Documentation
6. Ensure the Critical Path is Valid	<ul style="list-style-type: none"> Releases included in Program Schedule
7. Ensure that Total Float is Valid	<ul style="list-style-type: none"> May also have Hardening sprints or catch up sprints for Unfinished work
8. Conducting a Schedule Risk Analysis	<ul style="list-style-type: none"> Iteration 0 planning Sprint Planning Sessions Uncertainty regarding number of sprints, releases and velocity Retrospectives
9. Updating the Schedule Using Actual Progress and Logic	<ul style="list-style-type: none"> Releases included in Program Schedule Daily stand up meetings Velocity metrics Burn up/Burn down charts quantifiable back up data for schedule progress Kanban board
10. Maintaining a Baseline Schedule	<ul style="list-style-type: none"> Roadmap becomes baseline from which to measure variances Demonstration to stakeholders Retrospective

References



- Measuring Integrated Progress on Agile Software Development Projects by Tamara Sulaiman, PMP, CST, SolutionsIQ and Hubert Smits, CST, Rally Software
- AgileEVM – Earned Value Management in Scrum Projects by Tamara Sulaiman, PMP, CSM and Brent Barton, CSM, CST and Thomas Blackburn, PMP, CSM SolutionsIQ SolutionsIQ InfoTech, Inc.
- Systems Engineering for Software Intensive Projects Using Agile Methods, Rosser, Larri; Marbach, Phyllis; Osvalds, Gundars; Lempia, David; International Council on Systems Engineering IS14 Conference, Las Vegas NV July 2014.
- Principles for Agile Development, Marbach, Phyllis; Rosser, Larri; Osvalds, Gundars; Lempia, David; International Council on Systems Engineering IS15 Conference, Seattle WA, July 2015.
- www.gao.org
- An Industry Practice Guide for Agile on Earned Value Management Programs, National Defense Industrial Association (NDIA), Integrated Program Management Division (IPDM), March 31, 2017



Upcoming Events



UPCOMING TALKS:

“Systems and Software Qualities Tradespace Analysis” Series



Barry Boehm, Chief Scientist, SERC; TRW Professor of Software Engineering and Director, Center for Software Engineering, University of Southern California

August 1 | 1:00 PM ET

Bill Curtis, Senior VP & Chief Scientist, CAST Software; Head of CAST Research Labs, Executive Director, Consortium for IT Software Quality (CISQ)

October 3 | 1:00 PM ET



Xavier Franch, Full Professor, Polytechnic University of Catalonia (BarcelonaTech)

December 11 | 1:00 PM ET

Please visit the [SERC Talks page](#) for more information and updates.



Thank you for joining us!

Please check back on the [SERC website](#) for today's recording and future SERC Talks information!

CONTACT

Editor-in-Chief: Dr. Barry Boehm, University of Southern California – boehm@usc.edu

Webinar Coordinator: Ms. Mimi Marcus, Stevens Institute of Technology – SERCtalks@stevens.edu