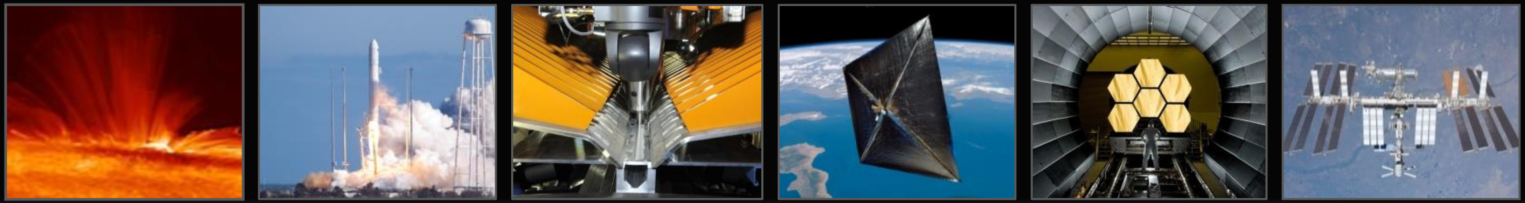
National Aeronautics and Space Administration

SLS Flight Software Testing: Using a Modified Agile Software Testing Approach

Dr. Albanie T. Bolton
Software Systems Engineering Branch



marshall

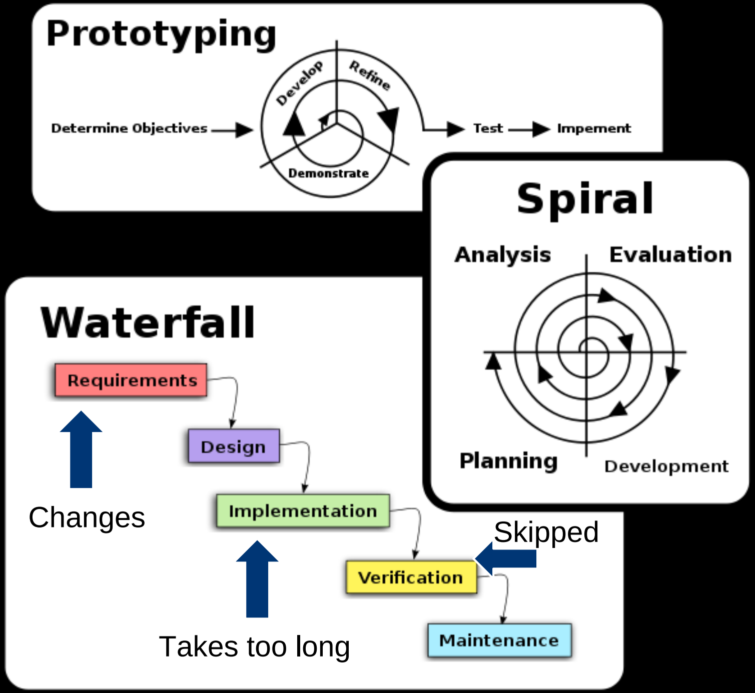
www.nasa.gov

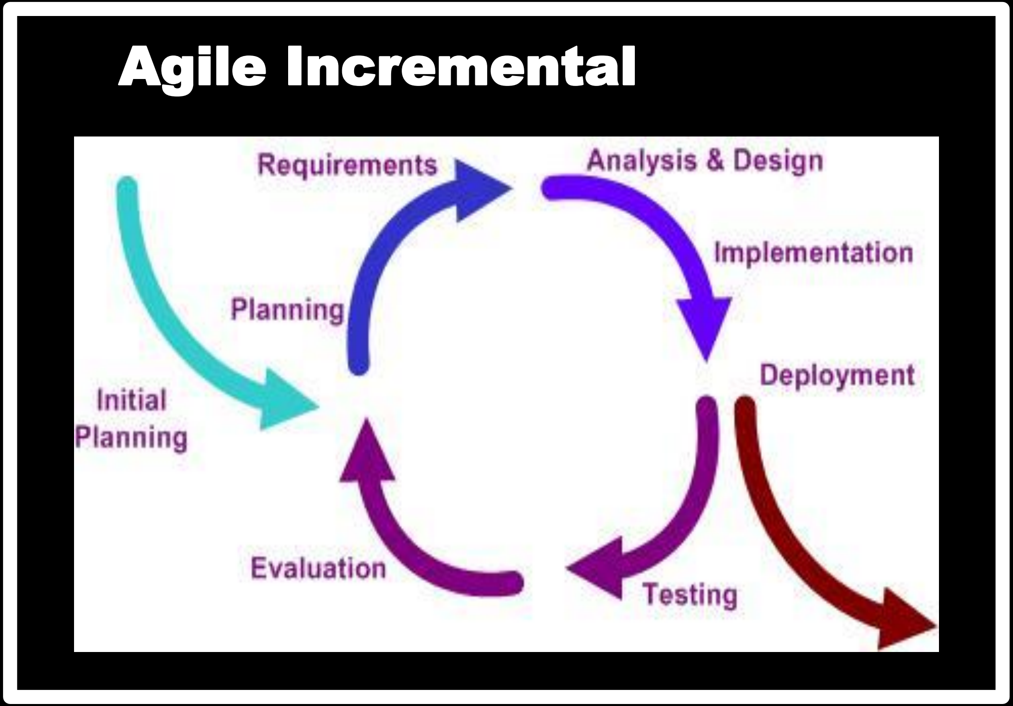
Flight Software Conference 2016

Agenda

- Software Processes vs. Agile Process
- Agile Software Development
- Benefits
- Sprint Life Cycle
- Sprint Phases
- Agile Testing
- Testing Strategies
- Value of Agile Testing

Software Processes vs. Agile Process





Waterfall development is another name for the more traditional approach to software development

Rapid Adaptable

AGILE

Quality-driven

Cooperative Iterative

Not a process, it's a **philosophy** or **set of values**

Agile Software Development

- **Agile software development** is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
- The Agile Manifestion reads as follows: [1]

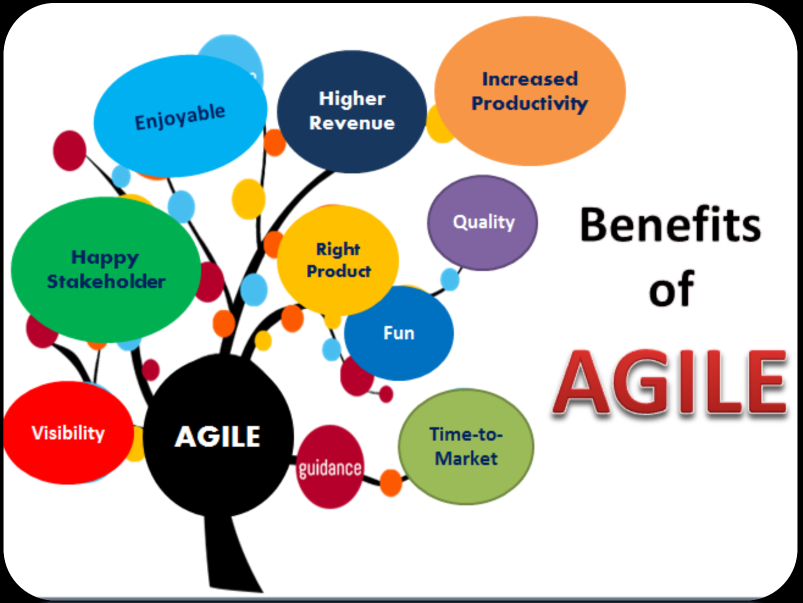
  We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

  - **Individuals and interactions** over processes and tools
  - **Working software** over comprehensive documentation
  - **Customer collaboration** over contract negotiation
  - **Responding to change** over following a plan

[1] Beck, Kent; et al. (2001). "Manifesto for Agile Software Development". Agile Alliance. http://agilemanifesto.org/. Retrieved 14 June 2010.

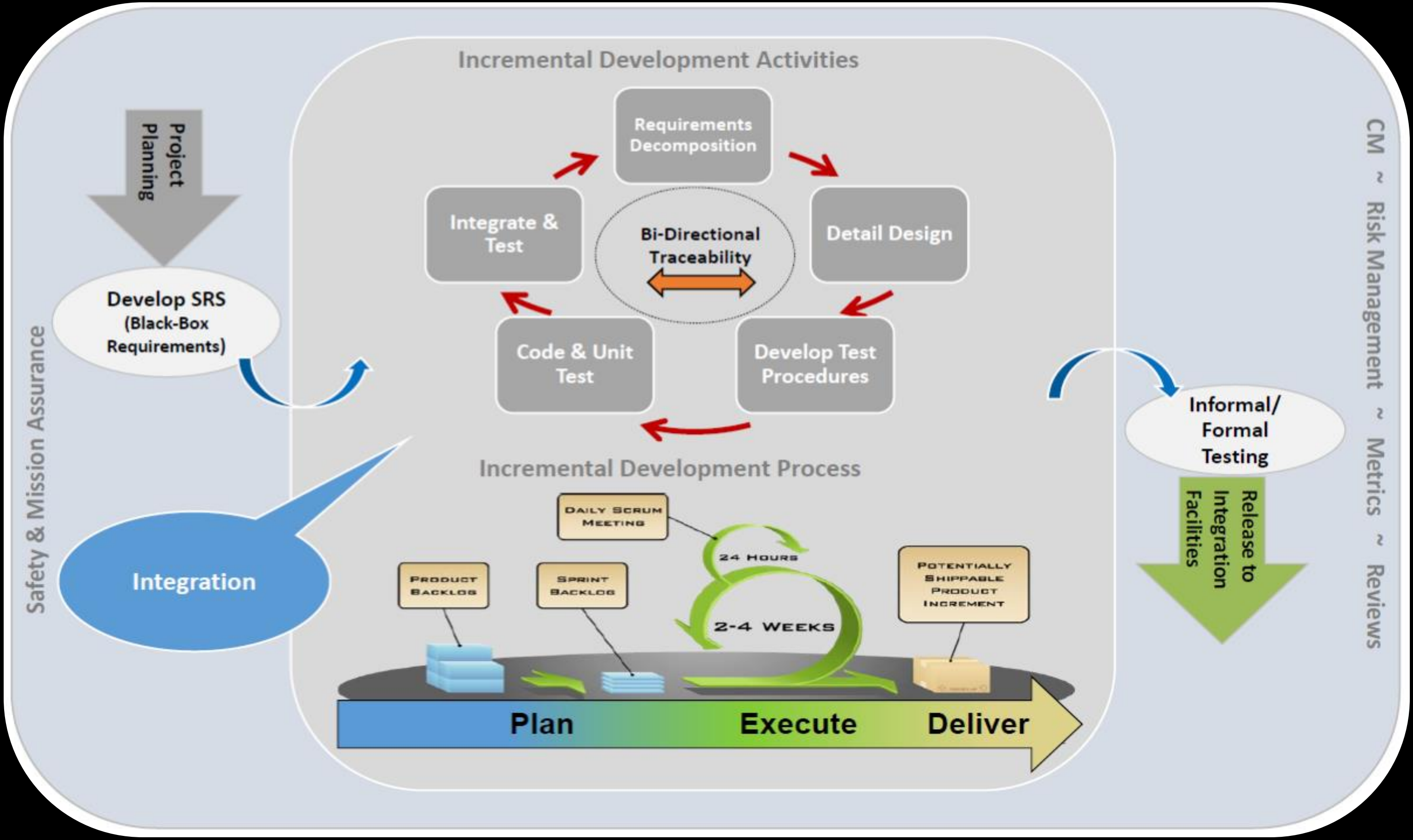
SLS Flight Software Project Benefits of Agile

- Don't need all dependencies up front
- Can develop the known functions without having the details of every requirement
- Better team communication
- Accelerated delivery of product functionality (working solutions) as a means to demonstrate progress to our customers
- Cross-discipline communication, ownership, and accountability of for our products – maximize stakeholder involvement
- Capture and institutionalization of our processes
- Change and complexity management
- Effective project monitoring and control

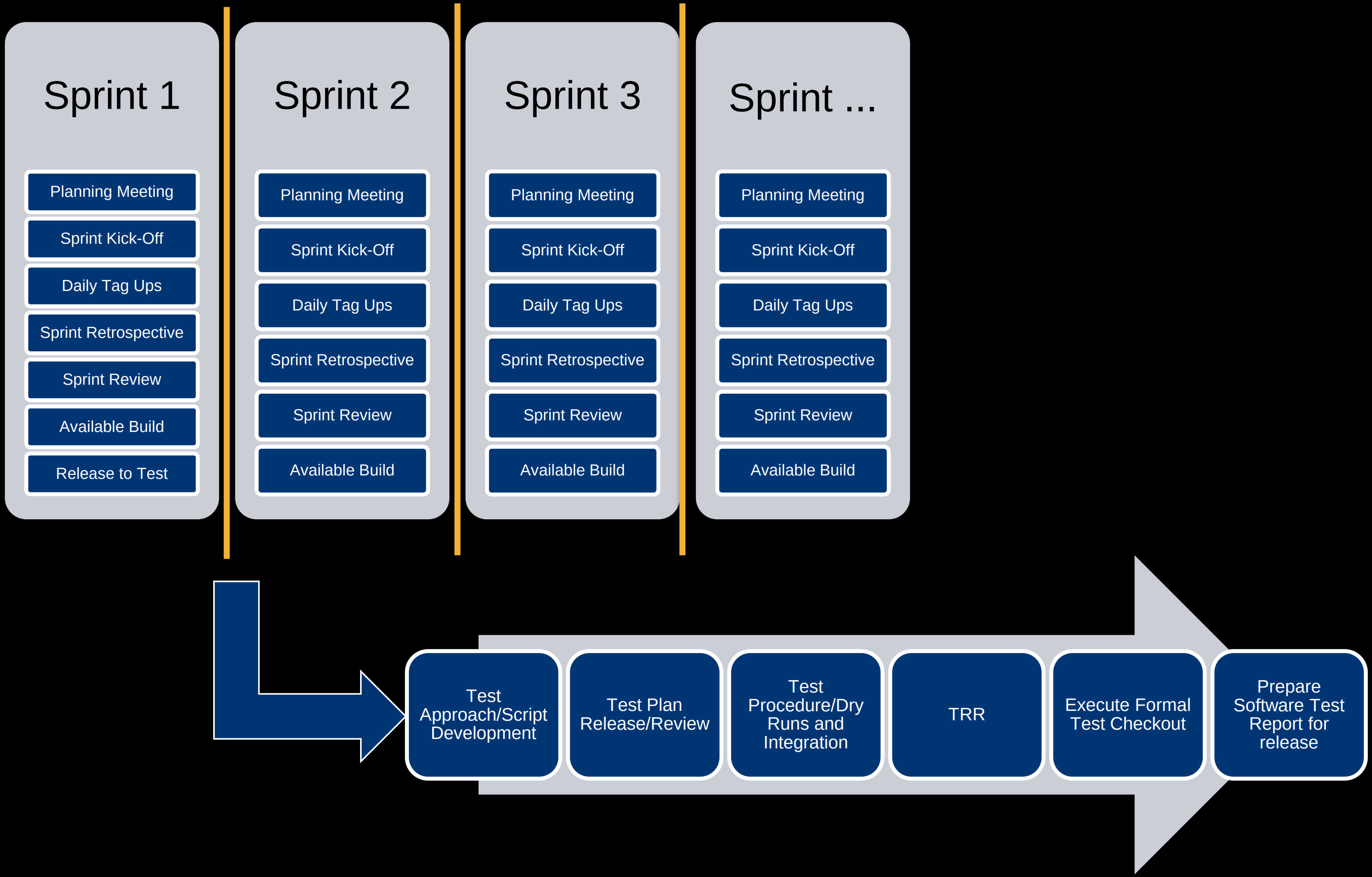


Software Test & Verification Twist: Agile Development

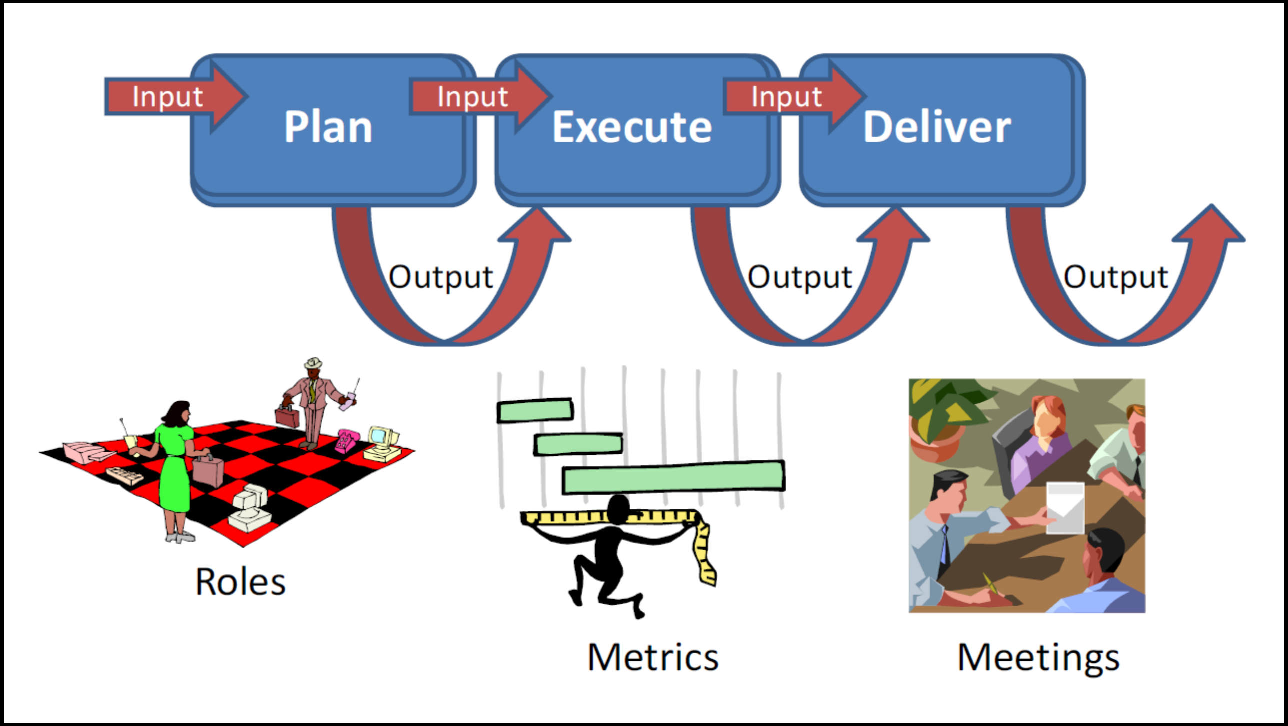
- Leverages from Scrum
- Merge the plan driven nature of the NASA project with a flexible incremental development framework


Incremental Development Activities
Project Planning
Develop SRS (Black-Box Requirements)
Requirements Decomposition
Detail Design
Develop Test Procedures
Code & Unit Test
Integrate & Test
Bi-Directional Traceability
Safety & Mission Assurance
Integration
Incremental Development Process
Daily Scrum Meeting
24 Hours
Product Backlog
Sprint Backlog
2-4 Weeks
Potentially Shippable Product Increment
Plan
Execute
Deliver
Informal/ Formal Testing
Release to Integration Facilities
CM ~ Risk Management ~ Metrics ~ Reviews

Flight Software Sprint Cycle



Sprint Life Cycle



- During a Sprint, the team performs tasks that lead to a potential deliverable product
- A series of Sprints are executed leading up to a Formal Release

Sprint Teams

- **Multi-disciplinary teams**
    - Requirements
    - Design
    - Test
    - Subject Matter Experts
    - Support roles (CM, SA, SEPG)

- Assigned by Product Owner and FSW Leads taking availability into account
- Plan to evolve into fairly standard teams, but will flex memberships over time based on need/focus of planned sprint/expertise
- Train and rotate scrum masters so that the process is well understood by many

Sprint Increments

- Time-boxed set of tasks/activities
  - 2 – 4 Weeks
- Activities captured in Sprint Backlog derived from Product Backlog
- Supporting artifacts are identified during planning along with “done-done” criteria
  - A checklist is being developed that will capture the default set of tasking/artifacts for stories
- Goal of every sprint to have some subset of functionality implemented; preferably in a way that can be demonstrated **WITH all** the supporting artifacts completed

Sprint Phases

	PLAN	EXECUTE	DELIVER
Inputs	Product Owner proposed functionality with Sprint Backlog tasks identified from the (release) prioritized Product Backlog	Team agreed to Sprint Backlog with task sprint priorities, estimates, and accepted responsibilities	Completed work products that meet task completion criteria/Deliverable functionality/confirmation of completed tasks and status, minutes, issues, risks
Output(s)	Team agreed to Sprint Backlog with task sprint priorities, estimates, completion criteria, and accepted responsibilities	Completed work products that meet task completion criteria/Deliverable functionality/confirmation of completed tasks and status, minutes, issues, risks	Buy-in from Product Owner/Possible redirection; designated work products (including sprint review and retro meeting work products and CM/SA process/product audits) under appropriate level of control.
Roles	Product Owner; ScrumMaster; Team; Moderator (for estimating)	ScrumMaster; Team; Management resolve issues, as needed	Product Owner; ScrumMaster; Team; Management; Other Stakeholders
Meetings	Scrum of Scrums (before Sprint); Sprint Planning Meeting	Daily Stand-up Meeting	Sprint Retrospective and Sprint Review Meeting

Flight Software Planning Phase

- The Team works with the Product Owner to:
  - Estimate the assigned tasks using Task Based Estimating
  - Review the total estimate for the Sprint and based on known constraints/team velocity, finalizes task assignments for the Sprint
  - Accept responsibility for each task by the team members
  - Agree to the Sprint Schedule including daily standup time/place, official start/end of Sprint, and close out meeting dates.
  - Identify “done-done” criteria and artifacts/location for each task
  - FSW Leads participate

Input	Product Owner proposed functionality – Sprint backlog tasks identified and prioritized
Output	Team agreed to Sprint backlog, estimates, and accepted responsibility
Roles	Product Owner, ScrumMaster, Team, Moderator (for estimating)
Metrics	# of Sprint Tasks accepted by team, Total # of Story Points
Meetings	Sprint Planning Meeting

Flight Software Execution Phase

- The Team performs the agreed to tasks, meeting daily
  - Scrum Master required to provide status at FSW Weekly (Scrum of Scrums)
- Issues captured by the Scrum Master
  - Handled within the team where possible
  - Escalate to Team Leads and/or Management where required

Input	Team agreed to Sprint backlog, estimates, and accepted responsibility
Output	Completed work products that meet task closure criteria
Roles	ScrumMaster, Team, Management (issue resolution)
Metrics	Daily Status
Meetings	Daily Stand-up Meeting

Flight Software Delivery Phase

- Completed artifacts are stored appropriately
- The team holds 2 meetings to close out the Sprint activities:
  - Sprint Retrospective – internal reflection on how team performed
  - Sprint Review – out-brief to product owner and management on completed functionality
- Integration of the code base back into the trunk occurs between sprint cycles
- Ready for next sprint cycle when an integrated build is available

Input	Completed work products that meet task closure criteria
Output	Buy-in from Product Owner/possible re-direction, designated work products
Roles	Product Owner, ScrumMaster, Team, Management (authorization)
Metrics	# of Tasks Completed, # of Story Points completed
Meetings	Sprint Review Meeting, Sprint Retrospective (Team only)

Sprint Team Responsibilities

- Team members participate in all phases of the Sprint (Sprint Planning meeting, Sprint backlog selection, and estimating process)
- During the planning meeting, team members accept/select tasks from the Sprint backlog to work
  - Team member availability is to be taken into account during planning
  - Some team members may be required to participate in more than one Sprint at any given time
- Team members are <u>required</u> to attend the daily stand-up meetings unless pre-coordinated with the ScrumMaster
- Each team member is <u>accountable</u> for the tasks they accept
  - Team members collaborate within the team and ask for help from other team members or the ScrumMaster (especially if it is an external impediment) before they are in jeopardy of not completing accepted tasks

Sprint Team Activities con't…..

- Team to work **collaboratively**….not all inclusive list!
- Cross-Team collaboration is encouraged… consider a regular sharing meeting
- Requirements
  - Evaluate SRS requirements for task (and other source info) and update if needed (following process)
  - Support WB requirements development and capture
- Design
  - Perform design, code, and unit test aspects of the task
- **Test**
  - **Perform evaluation of test needs for white box requirements**
  - **If needed, develop WB level test case/procdures & execute test**
  - **Provide support**
- SMEs
  - Provide expertise related to Vehicle on requirements and design development and capture
- Support
  - Provides expertise related to process definition/implementation/execution

Views of Agile Testing

eXtreme Testing

- Automated unit testing
  - Developers write tests
  - Test first development
  - Daily builds with unit tests always 100% pass
- Functional testing
  - Customer-owned
  - Comprehensive
  - Repeatable
  - Automatic
  - Timely
  - Public

Focus on automated verification – **enabling agile software development**

Exploratory Testing

- Manual testing by professional skilled testers
- Freedom, flexibility and fun for testers
- Controllability, reliability and high quality for managers
- Optimized to find bugs
- Continually adjusting plans, re-focusing on the most promising risk areas
- Following hunches
- Minimizing time spent on documentation

Focus on manual validation **– making testing activities agile**

Agile Testing Strategies

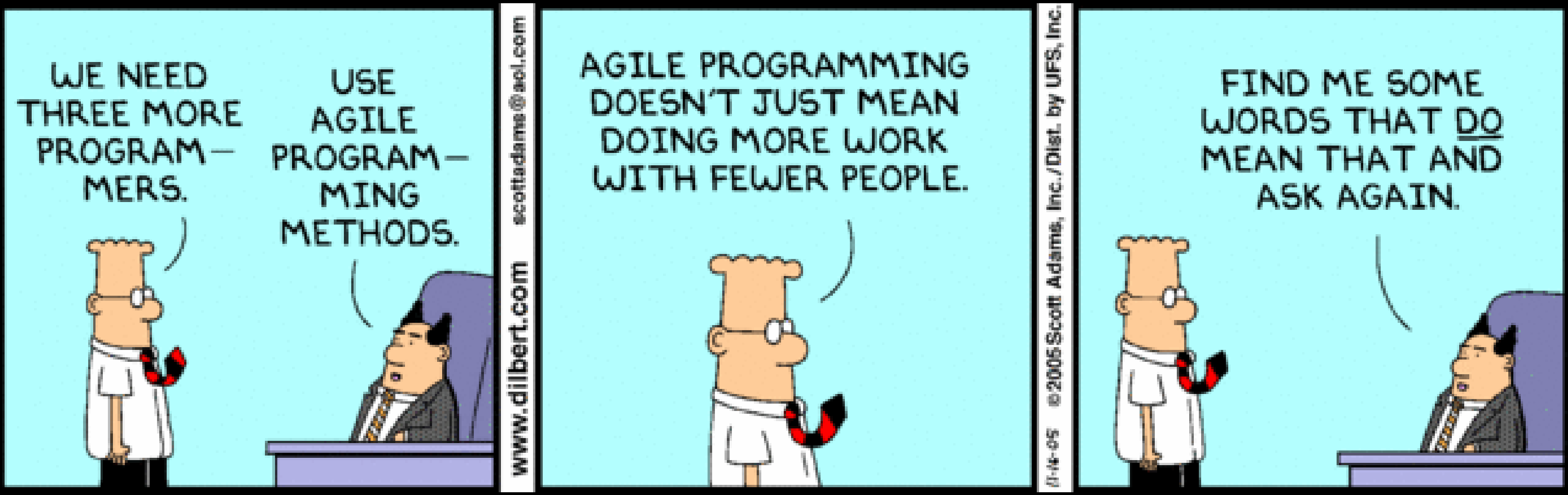
- Get your developers involved (Test Driven Development(TDD), unit testing)
- Automate regression tests
- Scenario based testing when appropriate
- Generate test case documentation whenever possible (from exploratory tests or acceptance criteria)
- Adopt a good toolset to assist with collaboration and automation

Agile Testing Value

- Testing in agile software development should provide the information that stakeholders need to make decisions and steer the development into the right direction
- This information must be provided promptly
- Testing provides data on the status of the deliverables and generates *project intelligence*
- Project intelligence is knowledge of risks and benefits
- Knowledge of risks, benefits, test records and results are more valuable than test documentation and infrastructure
- We can increase the value of testing most by
  - Improved intelligence
  - Providing intelligence earlier

Review of Agile Testing

If you think agile means team members doing whatever they want with no requirements, rules, documentation, and testing then you may read too much Dilbert …….

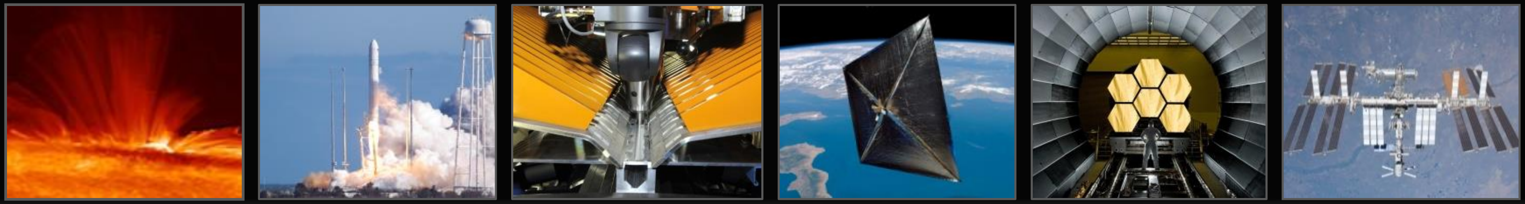

WE NEED THREE MORE PROGRAM-MERS.
USE AGILE PROGRAM-MING METHODS.
www.dilbert.com scottadams@aol.com
AGILE PROGRAMMING DOESN'T JUST MEAN DOING MORE WORK WITH FEWER PEOPLE.
11-16-05 ©2005 Scott Adams, Inc./Dist. by UFS, Inc.
FIND ME SOME WORDS THAT DO MEAN THAT AND ASK AGAIN.

National Aeronautics and Space Administration

Dr. Albanie T. Bolton

Email: albanie.t.bolton@nasa.gov

Software Systems Engineering Branch



marshall

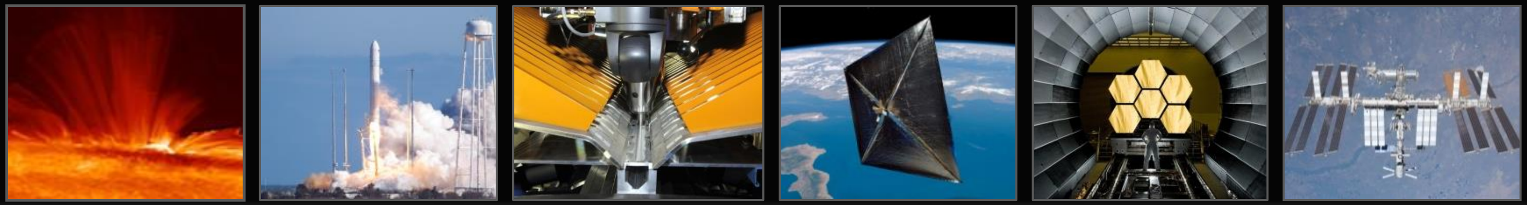
www.nasa.gov

Flight Software Conference 2016

www.nasa.gov

National Aeronautics and Space Administration

INFORMATIONAL CHARTS



marshall

www.nasa.gov

Definitions

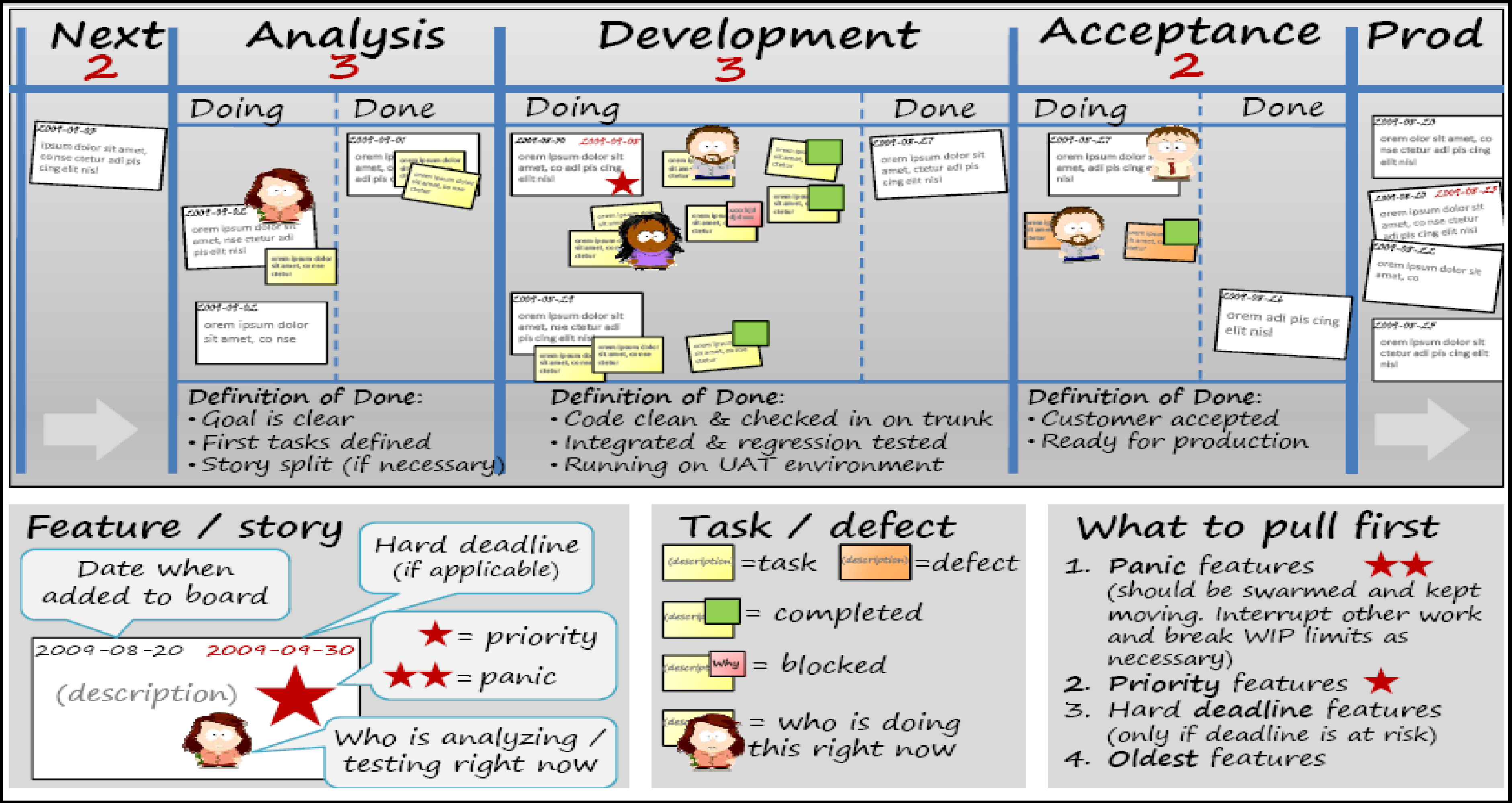
- **Scrum** –an iterative, incremental framework for software engineering.
- **Sprint** –a “time-boxed” software engineering iteration that plans, executes, and delivers specific project-defined tasks.
- **Product Backlog** –a high-level list of capabilities or functions (stories) that is maintained throughout the entire project. The list emerges over the life of the project.
- **Sprint Backlog** –a management prioritized list of tasks the team accepts as ‘work’ during a sprint. These tasks are broken down from the stories. Each task has a responsible person and estimate.

Planning and Execution Meetings

- **Sprint Planning Meeting** – held prior to the start of a Sprint. The following steps are performed with the initial Sprint Backlog:
  - Sprint tasks broken down from the assigned, estimated, and assigned/accepted
  - task completion criteria are defined (aka “done-done”)
  - duration (“time-box”) of the Sprint is determined
  - Sprint meeting dates are established.
- **Daily Standup Meetings** – held daily at the same time (goal 15 minutes or less). The following is reported by each Team member:
    - What they accomplished the previous day
    - What they plan to accomplish today
    - Any impediments to accomplishing tasks.
  - The ScrumMaster is responsible for capturing the minutes, attendees, actions, issues, and/or risks
  - All team members expected to participate (exceptions are to be documented).

Kanban

- Lean approach to agile development
- Similar to Scrum


Next 2
Analysis 3
Development 3
Acceptance 2
Prod
Doing
Done
Doing
Done
Doing
Done
Definition of Done:
• Goal is clear
• First tasks defined
• Story split (if necessary)
Definition of Done:
• Code clean & checked in on trunk
• Integrated & regression tested
• Running on UAT environment
Definition of Done:
• Customer accepted
• Ready for production
Feature / story
Date when added to board
Hard deadline (if applicable)
2009-08-20
2009-09-30
(description)
★ = priority
★★ = panic
Who is analyzing / testing right now
Task / defect
(description) =task
(description) =defect
= completed
Why
= blocked
= who is doing this right now
What to pull first
1. Panic features ★★
(should be swarmed and kept moving. Interrupt other work and break WIP limits as necessary)
2. Priority features ★
3. Hard deadline features
(only if deadline is at risk)
4. Oldest features