412th Test Wing

War-Winning Capabilities … On Time, On Cost

A Conceptual Framework for Flight Test Management and Execution Utilizing Agile Development and Project Management Concepts

5/15/19

U.S. AIR FORCE

Craig A. Hatcher
812 TSS/ENTI
(661) 277-4987

Key Objectives

- Show how Agile methodologies could be used to manage and execute flight test projects
- Show how Agile tracking metrics can be used to measure an organizations capacity to do work

Traditional Scheduling Approach

Network Scheduling

412TW

- Assumptions
  - A project can be planned up front and executed according to the plan with minimal changes
    - Tracking against a model of the project (network schedule) provides the insight needed for decision making
    - Resources can be assigned to network tasks and used in computations to determine schedule length

Flight Test Environment

412TW

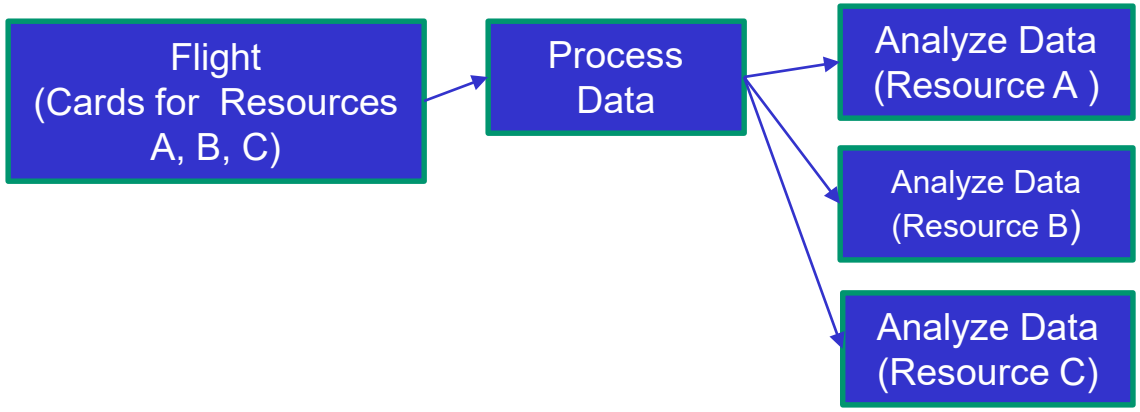
- Volatile
  - Exploratory in nature
    - Purpose is to uncover problems
  - Retesting is common
  - Test points are not flown as planned
    - Additional flights may be added
- Results
  - Schedules quickly become obsolete
    - Must be frequently updated to track with plan

Flight Test Environment

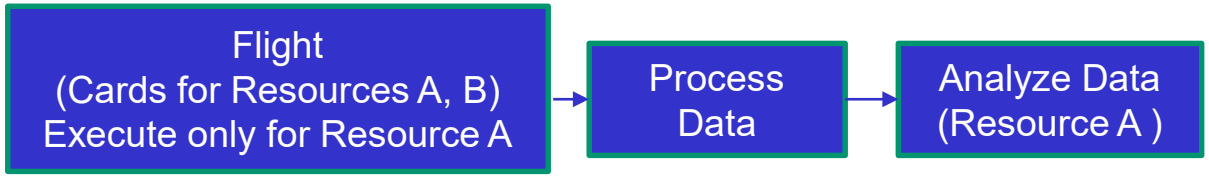
Example

412TW

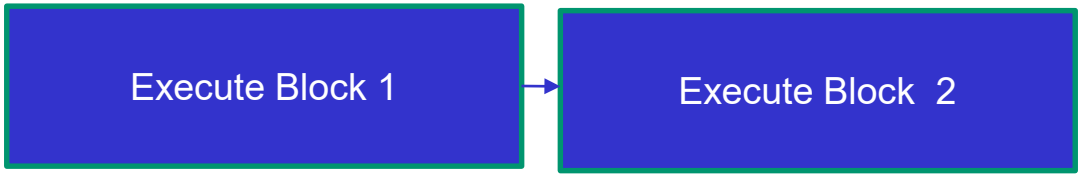
Plan



Actual



Method to Compensate



Raise level of network to define periods of work

- Lose insight
- Minimize value of network schedule

Is there a better way?

Agile?

Can Agile methods primarily used for software development be adapted to flight test?

Agile Fundamental Principles

Agile Manifesto

- We are uncovering better ways of developing software by doing 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

- That is, while there is value in the items on the right, we value the items on the left more

Agile Fundamental Principles

12 Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale
4. Business people and developers must work daily throughout the project
5. Build projects around motivated individuals. Give them an environment that support their needs and trust them to get the job done
6. The most efficient and effective method of conveying information to and with a development team is face-to-face conversation
7. Working software is the primary measure of progress
8. Agile processes promote development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely

Agile Fundamental Principles

12 Principles

9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity – the art of maximizing the work not done – is essential
11. The best architectures, requirements, and designs emerge from self-organizing teams
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Translated Agile Principles

Agile Manifesto

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

Individuals and Interactions	Over	Processes and Tools
Delivered Test Results	Over	Comprehensive Documentation
Customer Collaboration	Over	Formal Scope Negotiation
Responding to Change	Over	Following a Plan

- That is, while there is value in the items on the right, we value the items on the left more

Do These Statements Reflect Our Beliefs in T&E?

Translated Agile Principles

12 Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of test information
2. Welcome changing requirements, even late in the test program. Agile processes harness change for the benefit of the weapon system/warfighter
3. Deliver test results frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale
4. System Program Offices (SPOs) and testers must work daily throughout the project
5. Build projects around motivated individuals. Give them an environment that support their needs and trust them to get the job done
6. The most efficient and effective method of conveying information to and with a test team is face-to-face conversation
7. Delivered test information is the primary measure of progress
8. Agile processes promote sustainable test execution. The sponsors, testers, and users should be able to maintain a constant pace indefinitely

Translated Agile Principles

12 Principles

9. Continuous attention to technical excellence, good test discipline, and good safety discipline enhances agility
10. Simplicity – the art of maximizing the work not done – is essential
11. The best test planning emerges from self-organizing teams
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Do These Statements Make Sense in T&E?

Handling Volatility
Key Agile Principles - Structure

- The key to handling volatility is in how the project is structured
  - Need a project structure that is flexible but also minimizes multi-tasking
  - Solution
    - Break down a project into a set of Releases
      - For flight test, a release contains a set of test results useful to the SPO
      - *Agile Principle - Deliver test results frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale*

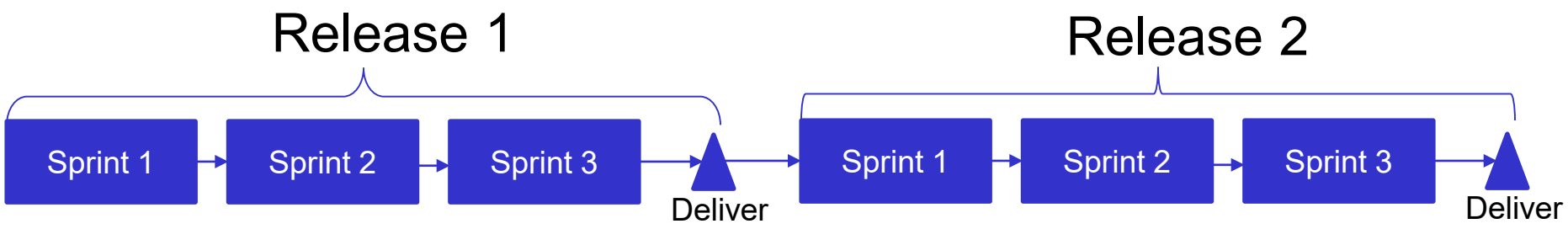
Handling Volatility

Key Agile Principles - Structure

- The key to handling volatility is in how the project is structured (cont.)
  - Solution (cont.)
    - Break down each release into multiple cycles (Sprints) of duration between (2wks and a month)
      - Fix Scope of work for each Sprint – minimizes multi-tasking
      - Allow changes to be introduced at the end of each Sprint – enables flexibility
        - » Changes must support release definition
      - Establish an emergency process that breaks these rules on an exception basis
    - Changes outside the scope of the release can be added at the next release

Embrace Change as Normal

Example

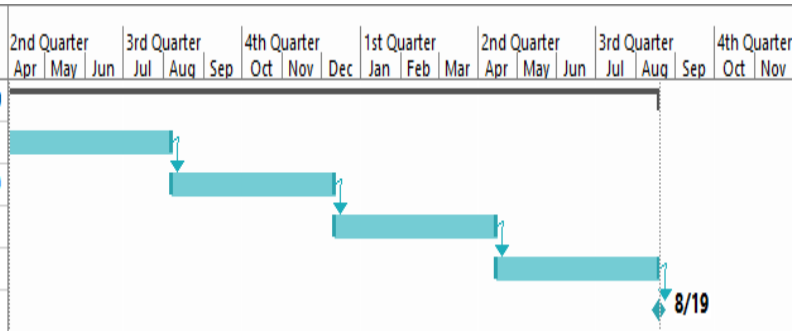

Release 1
Release 2
Sprint 1
Sprint 2
Sprint 3
Deliver
Sprint 1
Sprint 2
Sprint 3
Deliver

Schedule Estimating

Key Agile Principles

- Estimating should be simple and done at high level
  - Does not make sense to spend much time developing a detailed plan only to have it evolve significantly
    - *Agile Principle - Simplicity – the art of maximizing the work not done – is essential*
  - Agile scheduling is done in 3 tiers
    - Tier 1 – High Level Project Schedule
      - Shows expected # of releases and overall timeline

ID	Task Mode	Task Name	Duration	Start	Finish
1		**Project**	**360 days**	**Wed 4/3/19**	**Wed 8/19/20**
2		Release 1	90 days	Wed 4/3/19	Tue 8/6/19
3		Release 2	90 days	Wed 8/7/19	Tue 12/10/19
4		Release 3	90 days	Wed 12/11/	Tue 4/14/20
5		Release 4	90 days	Wed 4/15/2	Tue 8/18/20
6		Project End	0 days	Wed 8/19/2	Wed 8/19/20



Schedule Estimating
Key Agile Principles

- Estimating should be simple and done at high level (cont.)
  - Tier 2 – Release Schedule
    - Need a simple method for estimating and tracking instead of a network schedule
      - Key Principle - *To estimate or measure true progress we need a metric that can accurately show the technical progress of the project in terms of product delivered*
        - » Reflects progress from the customer's point of view

Schedule Estimating
Flight Test Application

- Suggest that the metric for tracking progress in flight test is Completed Test Objectives
  - Definition of completed test objectives is dependent on customers
    - Using our traditional model of flight test, the scope of a completed test objective would include following
      - Detailed test planning, test execution, data processing, data analysis, and documented test results for that objective
      - Definition would change if reporting is not required
  - Only when we have documented test results do we have something of value to the customer

Producing Increments of a Product that has Value to the Customer is a True Measure of Progress or **Throughput**

Schedule Estimating

- To build a schedule, the size of test objectives and the velocity of the test team must be calculated
- Size – a measure of complexity of a test objective
  - Utilizes a mechanism called a Story Point
    - A dimensionless unit of measure
    - To measure size, a baseline test objective must be established
      - *Must be totally executed within a Sprint*
      - Typically, a small test objective will be used and given a story point value of 1
      - All other test objectives are given a story point value based on their relative size to the baseline test objective
        - » i.e. Test objective 2 is twice the size as the baseline test objective so it gets a value of 2
        - » If test objectives are too large to fit in a Sprint they are broken down into sub-objectives

Schedule Estimating

- Technique uses comparative estimating
  - A 1991 study by Vicinanza et al., "Software-Effort Estimation: Exploratory Study of Expert Performance" showed that experts are more accurate in comparative estimation, vs. bottoms up.
  - *Agile Principle - Simplicity – the art of maximizing the work not done – is essential*

Story Points Result in a Unit of Outcome - Throughput

Schedule Estimating

- Velocity – measure of a team's rate of progress
  - A measure of throughput expressed in increments of product developed and completed during one iteration
    - # of Story Points per Sprint
  - Velocity of a team is determined based on experience
  - Over a period of time a team can understand their throughput
- Size * Velocity = Time to complete
  - # of Story Points * Story Points per Sprint * Cycle Length = time to complete

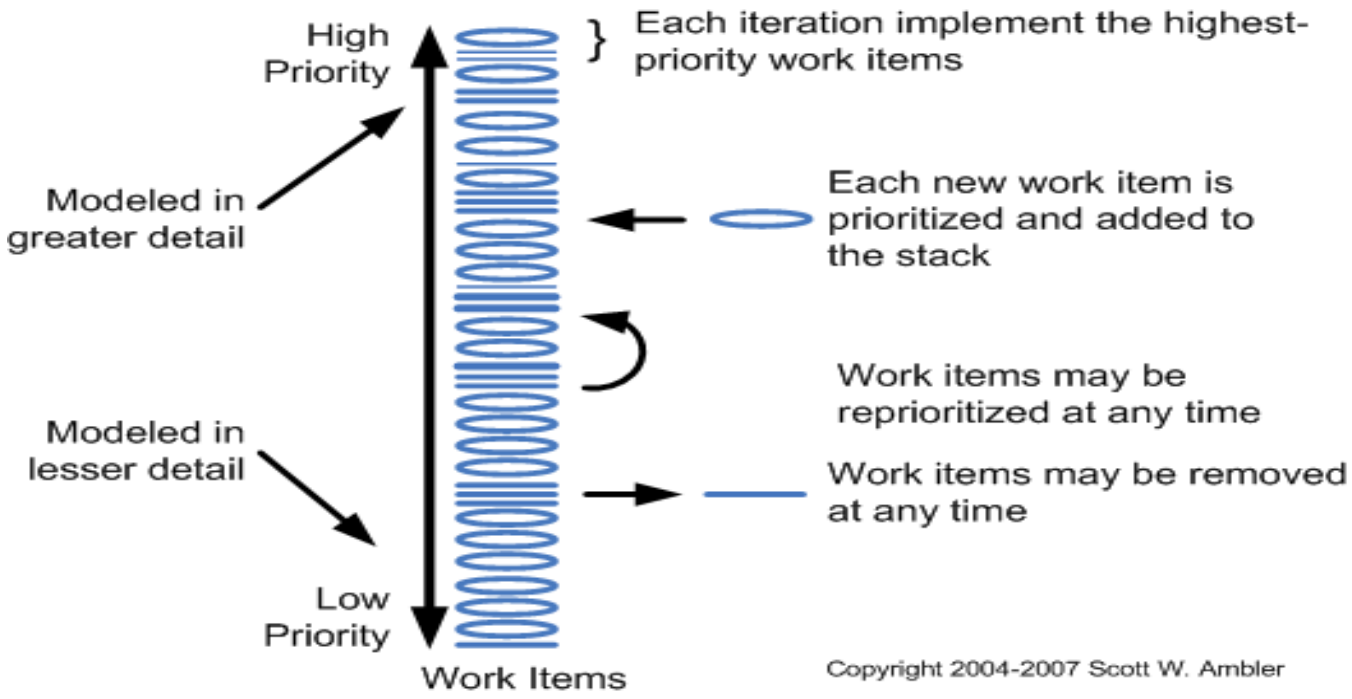
Knowing Size and Velocity Enables Test Objectives to be Assigned to Sprints

Workload Prioritization

- A prioritized list of test objectives for a project (Backlog) must be created
  - If test objectives are too large to fit in the pre-defined Sprint time then they are broken down into smaller sub-objectives
    - Methods to handle objectives that cannot be broken down
  - Test Objectives/sub-objectives should be prioritized based on best value to customer
- Sprint plans are defined based on test objective priority
- Multiple Sprints should be defined to work down test objectives/sub-objectives to create a release

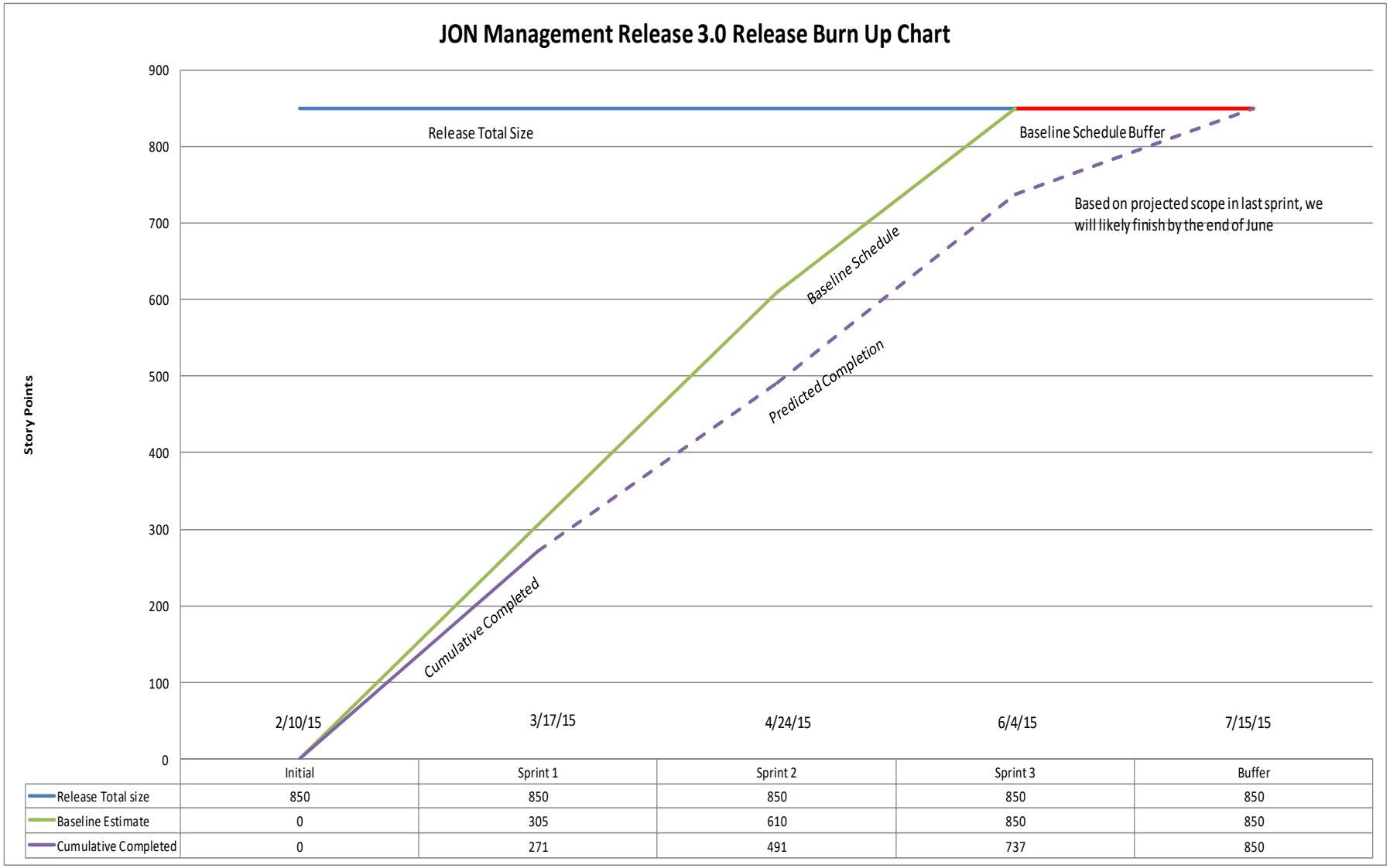
Workload Prioritization

Project Backlog



Scope Managed via Prioritization

Tier 2 Release Schedule



	Initial	Sprint 1	Sprint 2	Sprint 3	Buffer
Release Total size	850	850	850	850	850
Baseline Estimate	0	305	610	850	850
Cumulative Completed	0	271	491	737	850

Tier 3 - Sprint Tracking

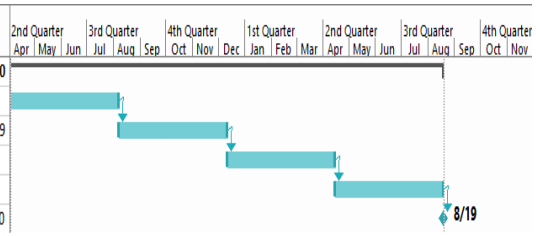
Kanban Chart						
Sprint 2						
	Detail Plan	Schedule	Test	Produce Data	Analyze Data	Document Results
Test Objective 1	**X**	**X**	**X**	**X**	**X**	**X**
Test Point 1	X	X	X	X	X	
Test Point 2	X	X	X	X	X	
Test Point 3	X	X	X	X	X	
Test Objective 2	**X**	**X**				
Test Point 1	X	X				
Test Point 2	X	X				
Test Point 3	X	X				

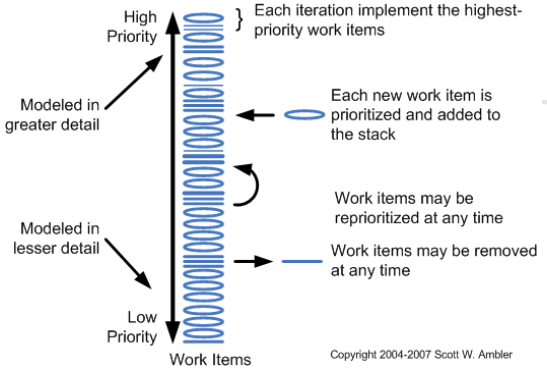
- Kanban Chart provides detailed status tracking within a Sprint
- Cannot take credit for completion of objective (count Story Points) until entire process is complete
  - Once an objective is complete it is marked as such in the backlog

Levels of Insight

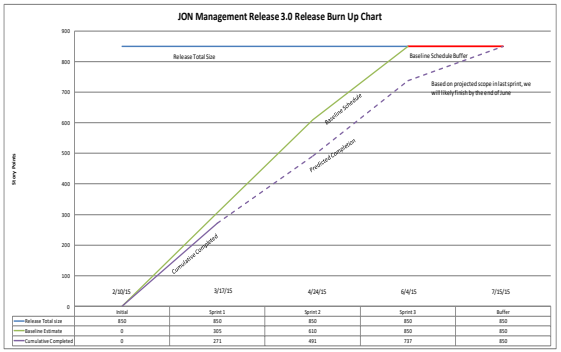
Tier 1 - Project Level

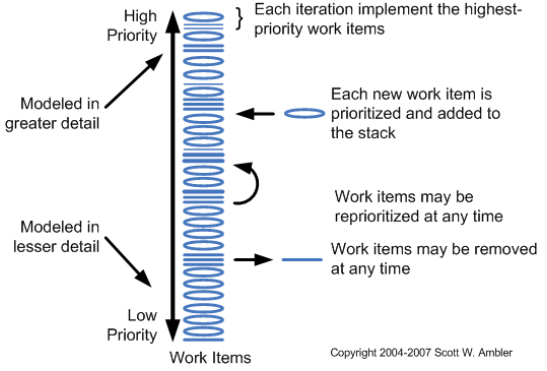
ID	Task Mode	Task Name	Duration	Start	Finish
1		**Project**	**360 days**	**Wed 4/3/19**	**Wed 8/19/20**
2		Release 1	90 days	Wed 4/3/19	Tue 8/6/19
3		Release 2	90 days	Wed 8/7/19	Tue 12/10/19
4		Release 3	90 days	Wed 12/11/	Tue 4/14/20
5		Release 4	90 days	Wed 4/15/2	Tue 8/18/20
6		Project End	0 days	Wed 8/19/2	Wed 8/19/20





Tier 2 - Release Level

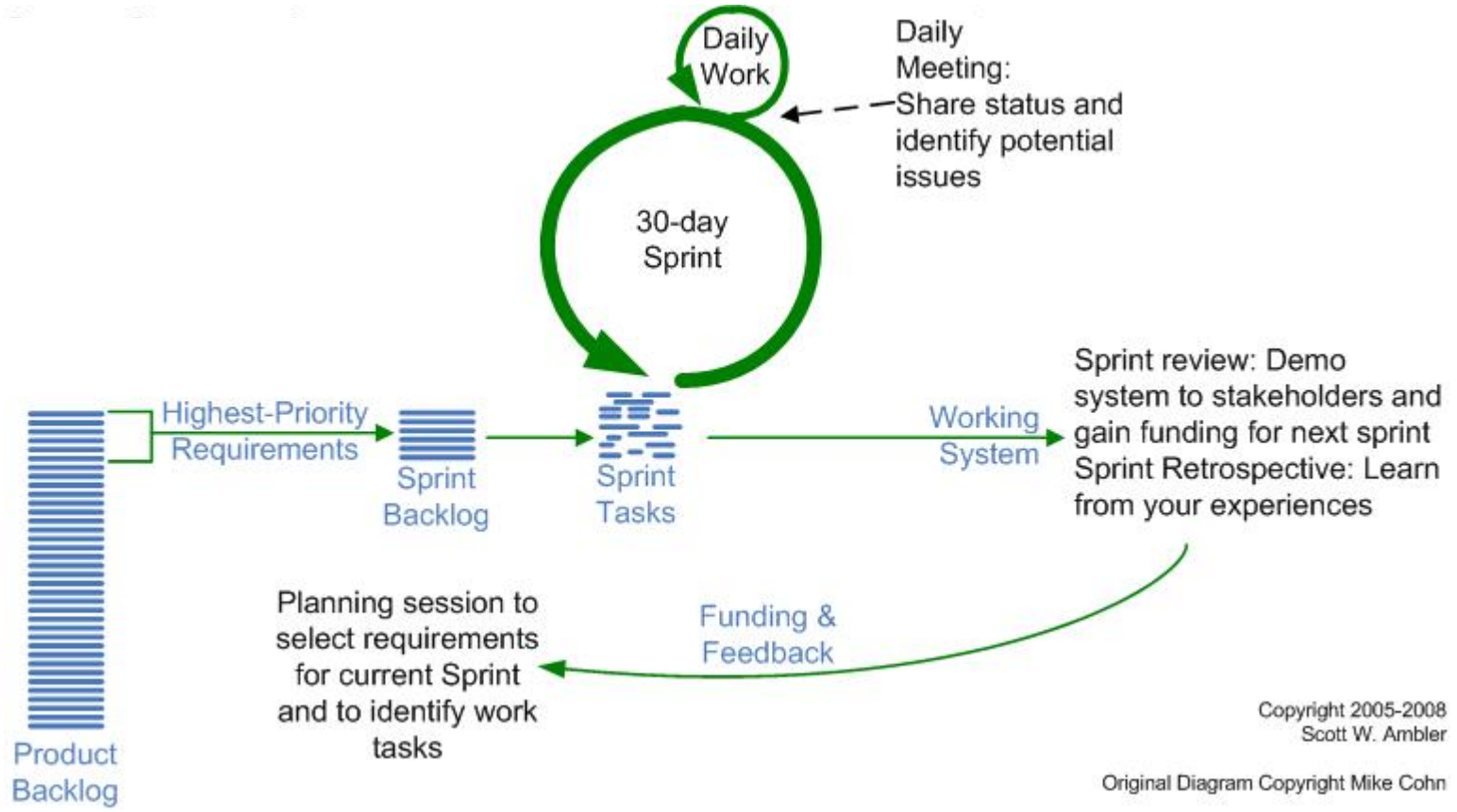




Tier 3 - Sprint Level

Kanban Chart						
Sprint 2						
	Detail Plan	**Schedule**	**Test**	**Produce Data**	**Analyze Data**	**Document Results**
Test Objective 1	**X**	**X**	**X**	**X**	**X**	**X**
Test Point 1	X	X	X	X	X	
Test Point 2	X	X	X	X	X	
Test Point 3	X	X	X	X	X	
Test Objective 2	**X**	**X**				
Test Point 1	X	X				
Test Point 2	X	X				
Test Point 3	X	X				

Execution


Daily Work
Daily Meeting: Share status and identify potential issues
30-day Sprint
Highest-Priority Requirements
Sprint Backlog
Sprint Tasks
Working System
Sprint review: Demo system to stakeholders and gain funding for next sprint
Sprint Retrospective: Learn from your experiences
Planning session to select requirements for current Sprint and to identify work tasks
Funding & Feedback
Product Backlog
Copyright 2005-2008 Scott W. Ambler
Original Diagram Copyright Mike Cohn

Organizational Impacts

- To implement Agile techniques the organization must work in line with Agile principles
  - Must structure processes with incremental delivery in mind
    - Approval and review cycles must be shortened
      - Level of approvals and reviews may have to be delegated to lower levels
  - Must execute to produce throughput
    - Deviating will cause schedule metrics to look bad
      - i.e. Fly all test points and wait to the end to process and analyze data will cause schedule metrics to show no progress till the end

Throughput/Capacity Metric

- Completed Test objective/sub-objectives can serve as a capacity metric
  - # of Story Points completed per month measures the work a test team can accomplish
  - If more work is added and the # of Story Points does not increase then the team is at capacity
    - The team is at maximum throughput

Throughput/Capacity Metric

- Completed Test objective/sub-objectives can serve as a capacity metric (cont.)
  - If a baseline test objective is defined for the organization then Story Points completed can be aggregated
    - Provides throughput/capacity metric across the organization
      - All test teams must understand the definition
- Kanban charts can be used to determine where bottlenecks to throughput exist

Can Determine if Organization Can Take on Additional Work

Benefits

- Methodology designed to handle volatility
- Methodology is simple
  - Seems to match our intuition
- Enables product (information) to be delivered to the customer incrementally
- Provides real status in terms of product (information) produced
- Provides a simple metric for measuring throughput/capacity
- Does not cost much to implement

References

- Goodpasture, J. 2010. *Project Management the Agile Way.* Fort Lauderdale, FL: J. Ross Publishing, Inc..
- Cohn, M. 2006. *Agile Estimating and Planning.* Upper Saddle River, NJ: Prentice Hall