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

2/8/21

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

U.S. AIR FORCE

DISTRIBUTION STATEMENT A - Approved for public release; distribution is unlimited.412TW-PA-21155

Integrity - Service - Excellence

Environment

Flight Test

- Increasing pressure to deliver test results sooner
  - Timelines are aggressive
  - Budgets are limited
- Testing is not entirely predictable
  - Exploratory in nature
    - Purpose is to uncover problems
  - Retesting is common
  - Test points are not flown as planned
    - Additional flights may be added
  - Result – Plans must frequently adapt to changes
- Workload requests can go beyond our capacity
  - At times work must be turned downed or delayed

Environment

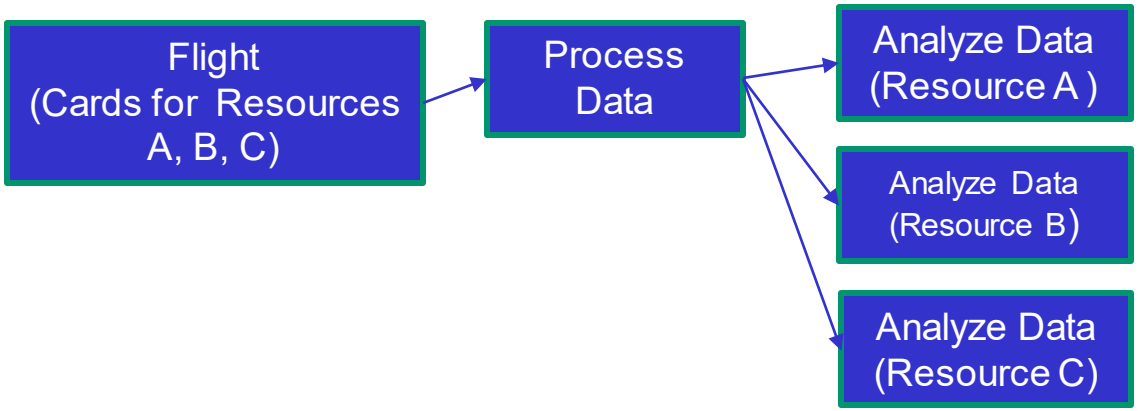
Current PM Tool Shortfalls

- Project Scheduling Tools
  - Based on network scheduling techniques
    - Time consuming to create
    - Special skills required
  - Not built to handle a highly dynamic environment
    - Built on premise that a project can be planned up front and executed to the plan with minimal changes
      - Highly dynamic changes causes plan to be updated continuously
        - » Problem mitigated by raising the level of network
        - » Lose insight in the process
- Capacity Planning
  - No simple metric to determine ability to take on more work

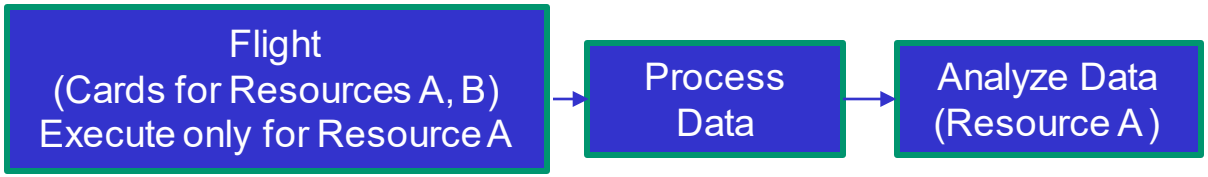
Environment

Current PM Tool Shortfalls - Example

Plan

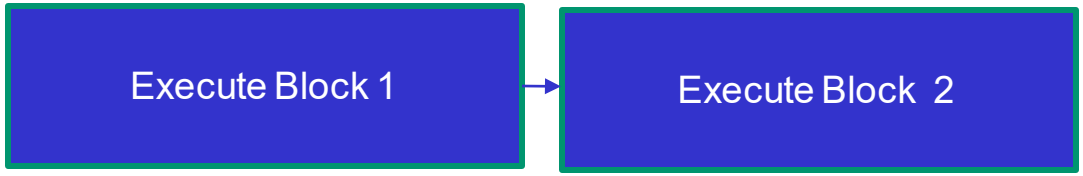


Actual



Networks schedules must be updated to reflect reality

Method to Compensate



Raise level of network to define periods of work

- Lose insight
- Minimize value of network schedule

Guiding Principles For Improvement

- Emphasis should be placed on providing test results as soon as possible
  - Periodic delivery of test results vs big bang
    - Partial completed test results vs preliminary test results
- Emphasis should be placed on producing information
  - e.g. focus should be on product throughput
- Bureaucracy must be minimized
  - Streamlined test report creation and approval
- Volatility of Testing Must be Embraced
- Workload capacity metrics should be simple and easy to collect
  - Derived from existing project data

Can Agile Techniques be Applied 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?

Suggested Implementation

Planning

- The suggested product is ***Completed Test Objectives***
  - **Test Objective Scope of Work** - Detailed test planning, test execution, data processing, data analysis, and documented test results for an objective
  - In other words, the product is **analyzed test results for a test objective**
  - **Used as the mechanism for tracking progress**

The Definition of Product is Key to the Implementation of Agile

Suggested Implementation

Planning

- A prioritized list (**Backlog**) of test objectives, based on importance to the customer, will be created
- Test objectives will be broken into sub-objectives if the work to provide test results cannot be accomplished within a month

Suggested Implementation

Planning

Backlog		
	Priority	**Test Objective**
	1	Certify Weapon A
	2	Clear Flight Envelope for Region A
	3	Clear Flight Envelope for Region B
	4	Certify Weapon B
	5	Validate Performance Specification A
	6	Clear Flight Envelope for Region C
	7	Validate Performance Specification B
	8	Certify Software Version X Capability A
	9	Certify Weapon C
	10	Certify S/W Version X capability B
	11	Certify Weapon D
	12	Certify Software Version X Capability C
	13	Certify Software Version X Capability D
	14	Certify Software Version X Capability E

Suggested Implementation

Planning - Backlog Development

- Each test objective must be **sized**
  - Sizing will be accomplished via Comparative Estimating
    - Will compare work by relative size
    - 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

Suggested Implementation

Planning - Backlog Development

- Requires a **Baseline Test Objective** to compare against
  - The Baseline Test Objective must be executable during 1 month
    - Usually a common small test objective
  - **Baseline Test Objective Example**
    - 1/2 day to write test cards
    - 1 flight
    - 1 day to process data
    - 1 day to analyze data and document results
  - Baseline Test Objective is given a value of 1
    - Value called a Story Point

Suggested Implementation

Planning - Backlog Development

- 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
- Test team works together to determine test objective size
  - Ensures there is some consistency in estimation

Suggested Implementation

Planning - Backlog Development

- **The method is meant to be simple and quick**
  - Attempting to be overly precise is not necessary
    - Perfect consistency is not required
  - The method is subjective and estimates just have to be good enough

Agile Principle - Simplicity – the art of maximizing the work not done – is essential

Suggested Implementation

Planning - Backlog Development

Backlog			
	Priority	Test Objective	Story Point Value
	1	Certify Weapon A	4
	2	Clear Flight Envelope for Region A	3
	3	Clear Flight Envelope for Region B	3
	4	Certify Weapon B	4
	5	Validate Performance Specification A	2
	6	Clear Flight Envelope for Region C	3
	7	Validate Performance Specification B	3
	8	Certify Software Version X Capability A	3
	9	Certify Weapon C	4
	10	Certify S/W Version X capability B	5
	11	Certify Weapon D	4
	12	Certify Software Version X Capability C	2
	13	Certify Software Version X Capability D	3
	14	Certify Software Version X Capability E	3

Story Point is unit less measure of *size*

Test Objectives *sized* via comparison with a baseline test objective

Suggested Implementation

Planning - Schedule Construction

- Philosophy
  - **Must be simple create**
    - Will estimate by comparison vs bottom up 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
  - **Must be simple to status**
    - Track progress based on test objective/sub-objective completion
      - *Measure progress in terms of product delivered*
        - » Only when we have documented test results do we have something of value to the customer
  - **Must handle change easily**
    - Work prioritization vs scope control

Producing Increments of a Product is a True Measure of Progress or **Throughput**

Suggested Implementation

Planning - Schedule Construction Overview

- Three tiers of schedules will be constructed
  - **Program** – high level schedule showing the timeline to complete all the releases required to complete the program
  - **Release** – mid-level schedule that will track test objective/sub-objective completion for a defined release
    - 3 months long
  - **Sprint** – low level schedule that will show progress of completing test objectives/sub-objectives during a month of execution

Test Results Provided to Customer at each Release

Suggested Implementation

Planning - Schedule Creation

- To create a schedule, the amount of work a test team can accomplish each month must be determined
  - e.g. # of Story Points per month (Velocity)
  - Velocity of a team is initially determined based on experience
  - Over a period of time a team can use past performance to understand their velocity

Suggested Implementation

Planning – Release Planning

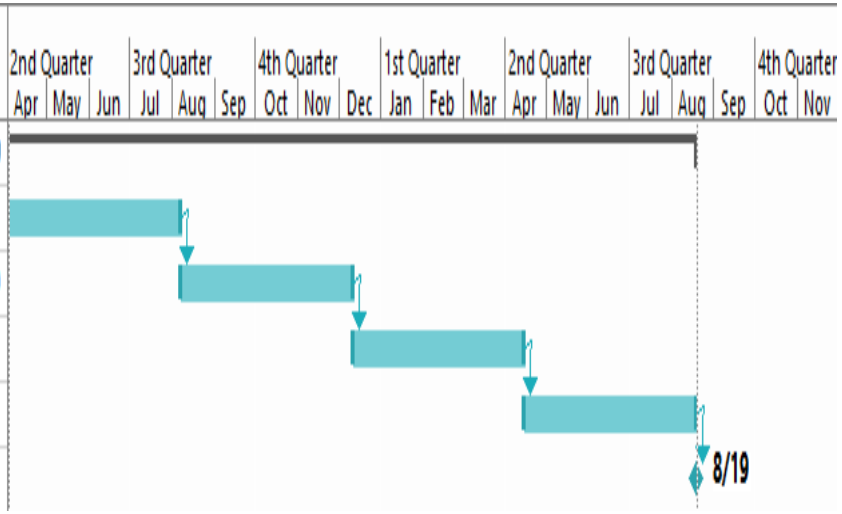
Backlog				
		Priority	Test Objective	Story Point Value
	Release 1 (3 mo)	1	Certify Weapon A	4
		2	Clear Flight Envelope for Region A	3
		3	Clear Flight Envelope for Region B	3
		4	Certify Weapon B	4
		5	Validate Performance Specification A	2
		6	Clear Flight Envelope for Region C	3
		7	Validate Performance Specification B	3
			Produce Release 1 Report	3
	Release 2	8	Certify Software Version X Capability A	3
		9	Certify Weapon C	4
		10	Certify S/W Version X capability B	5
		11	Certify Weapon D	4
		12	Certify Software Version X Capability C	2
		13	Certify Software Version X Capability D	3
		14	Certify Software Version X Capability E	3
			Produce Release 2 Report	3

Suggested Implemenation

Planning - Program Schedule

- Once the release plan is defined then a program schedule can be produced

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



- Additionally, the first release can be broken down into monthly increments (Sprints) based on the expected team velocity

Suggested Implementation

Planning - Release Planning

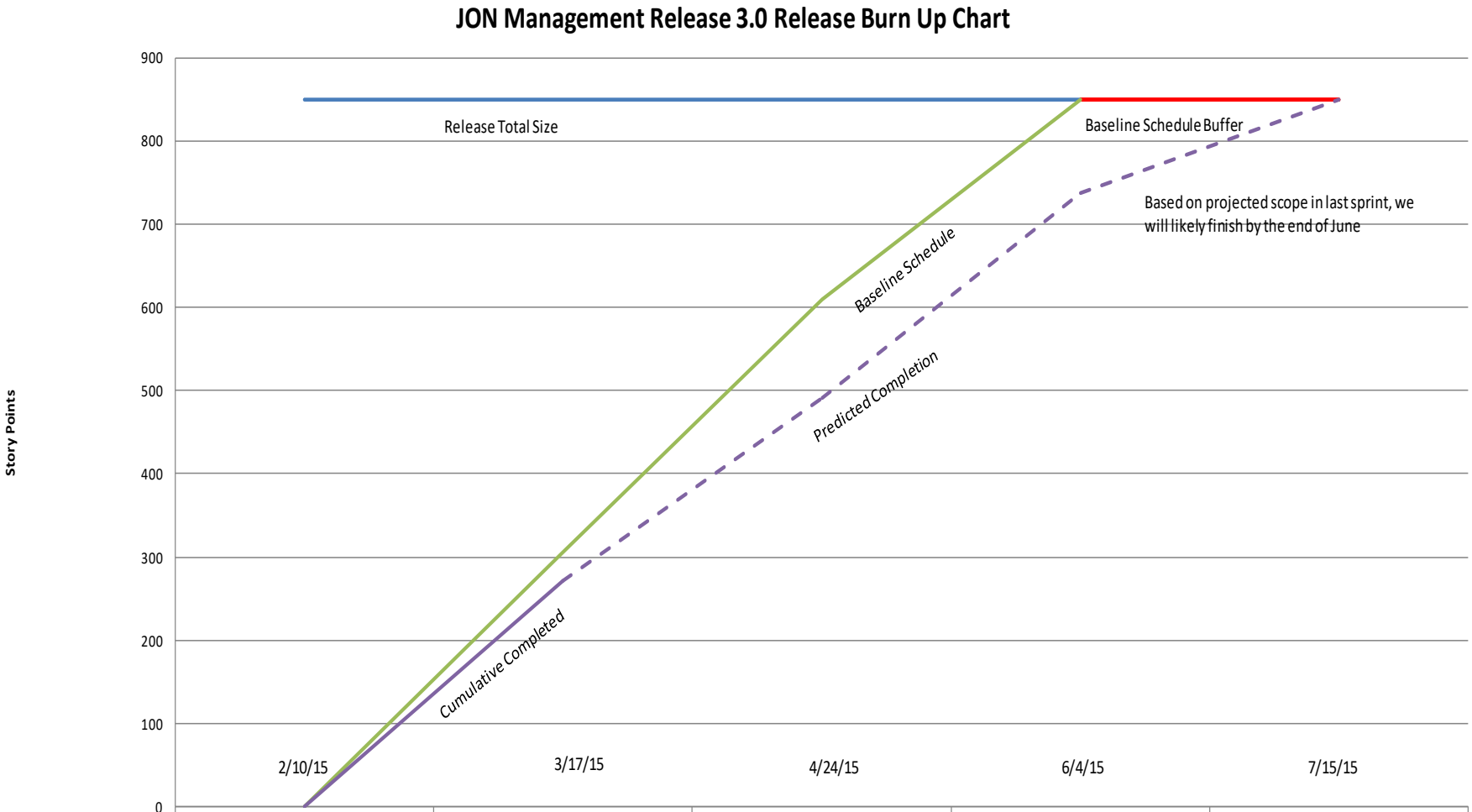
Release 1				
	Sprint 1			
		Priority	**Test Objective**	**Story Point Value**
		1	Certify Weapon A	4
		2	Clear Flight Envelope for Region A	3
		3	Clear Flight Envelope for Region B	3
	Sprint 2			
		4	Certify Weapon B	4
		5	Validate Performance Specification A	2
		6	Clear Flight Envelope for Region C	3
	Sprint 3			
		7	Validate Performance Specification B	3
			Produce Release 1 Report	3

Sprint – is a fixed time increment for work

- Proposing using a month as the Sprint length

Suggested Implementation

Planning - 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

Suggested Implementation

Planning – Sprint Schedule

- A Sprint schedule must be developed
  - Provides insight into progress during month of execution

Kanban Chart						
Sprint 1						
	Detail Plan	Schedule	Test	Produce Data	Analyze Data	Document Results
Certify Weapon A	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	
Clear Envelope for Region A	X	X				
Test Point 1	X	X				
Test Point 2	X	X				
Test Point 3	X	X				
Clear Envelope for Region B	X	X				
Test Point 1	X	X				
Test Point 2	X	X				

Test Report	Write	Tech Edit	CTF Review	Tech Expert Review	Senior Management Review
Release 1 Report					

Report tracking in 3rd Sprint only

Suggested Implementation

Execution - Schedule Status

- Program schedule updated after each release
- Release schedule updated at the end of the Sprint – once a month
  - **Only completed Test Objectives/Sub-objectives are counted**
    - **No partial credit**
- Kanban Chart updated at the completion of each activity

Simple to Status

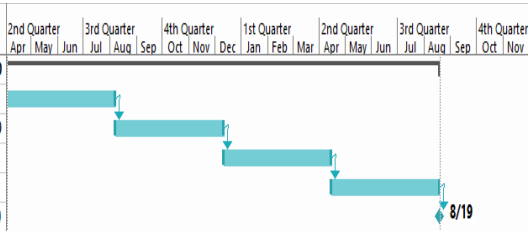
Suggested Implementation

Execution Tracking - Levels of Insight

412TW

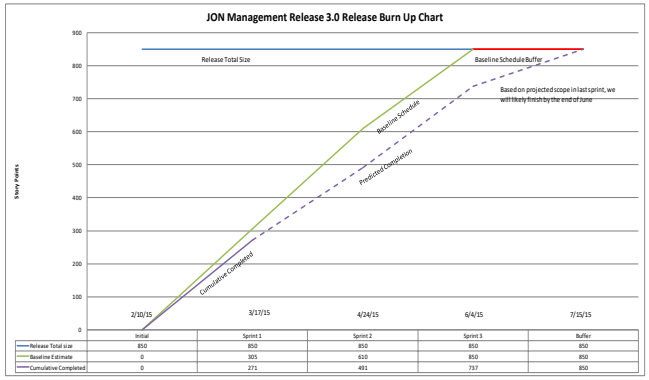
Tier 1 - Project Level

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



Backlog			
	Priority	Test Objective	Story Point Value
	1	Certify Weapon A	4
	2	Clear Flight Envelope for Region A	3
	3	Clear Flight Envelope for Region B	3
	4	Certify Weapon B	4
	5	Validate Performance Specification A	2
	6	Clear Flight Envelope for Region C	3
	7	Validate Performance Specification B	3
	8	Certify Software Version X Capability A	3
	9	Certify Weapon C	4
	10	Certify S/W Version X capability B	5
	11	Certify Weapon D	4
	12	Certify Software Version X Capability C	2
	13	Certify Software Version X Capability D	3
	14	Certify Software Version X Capability E	3

Tier 2 - Release Level



- Progress tracked based on Story Point value of test objectives completed
  - No partial credit

Release 1				
	Sprint 1			
		Priority	Test Objective	Story Point Value
		1	Certify Weapon A	4
		2	Clear Flight Envelope for Region A	3
		3	Clear Flight Envelope for Region B	3
	Sprint 2			
		4	Certify Weapon B	4
		5	Validate Performance Specification A	2
		6	Clear Flight Envelope for Region C	3
	Sprint 3			
		7	Validate Performance Specification B	3
			Produce Release 1 Report	3

Tier 3 - Sprint Level

Kanban Chart						
Sprint 1						
	Detail Plan	Schedule	Test	Produce Data	Analyze Data	Document Results
Certify Weapon A	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	
Clear Envelope for Region A	X	X				
Test Point 1	X	X				
Test Point 2	X	X				
Test Point 3	X	X				
Clear Envelope for Region B	X	X				
Test Point 1	X	X				
Test Point 2	X	X				
Test Report	Write	Tech Edit	CTF Review	Tech Expert Review	Senior Management Review	
Release 1 Report						

Simple to Create

Suggested Implementation

Execution - Scope Management

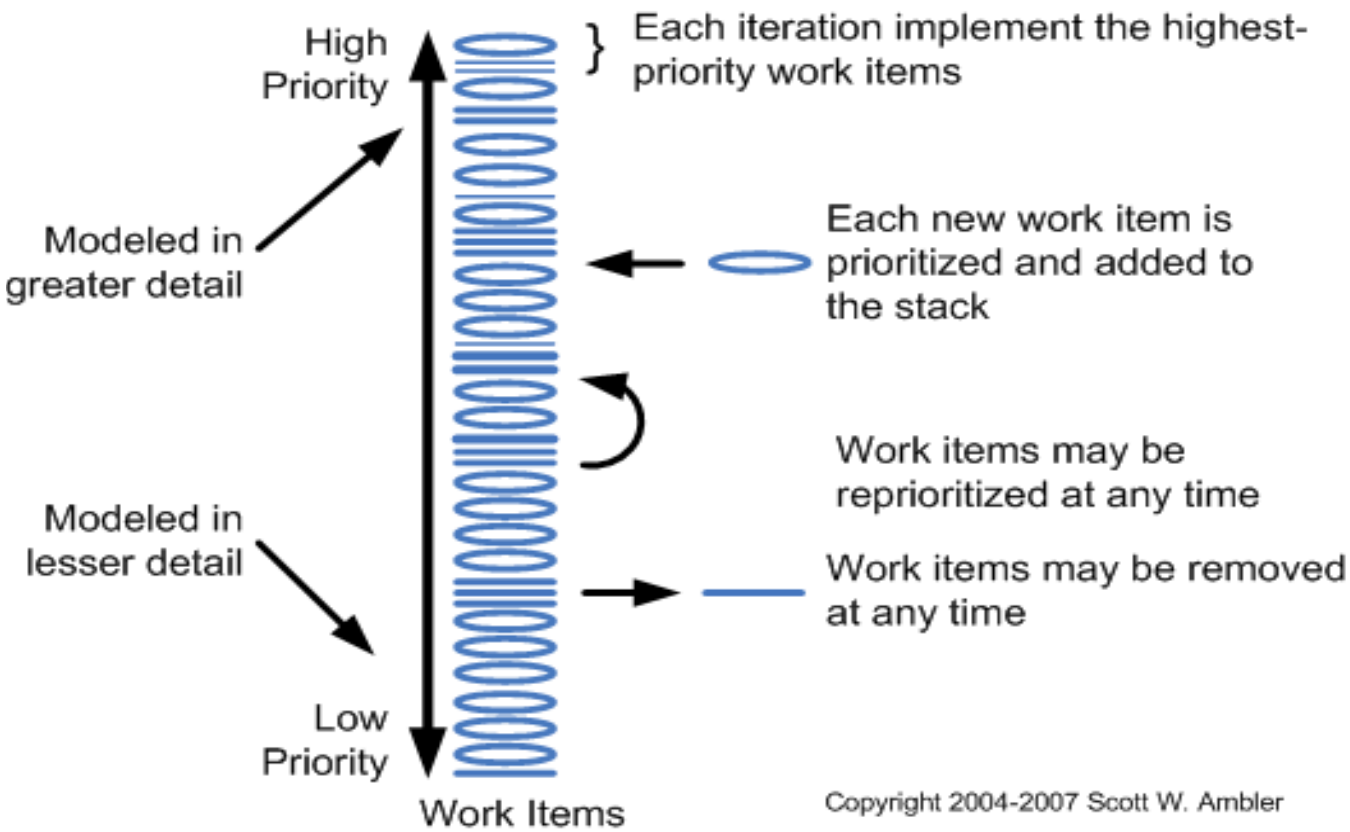
- Scope cannot be changed within a Sprint
  - Exception – Emergency
- Scope can be added at the end of a Sprint
  - Must be within scope of Release
- Scope can be added at the end of a Release
  - Any change

Flexibility without Chaos

Handles Change Easily

Suggested Implementation

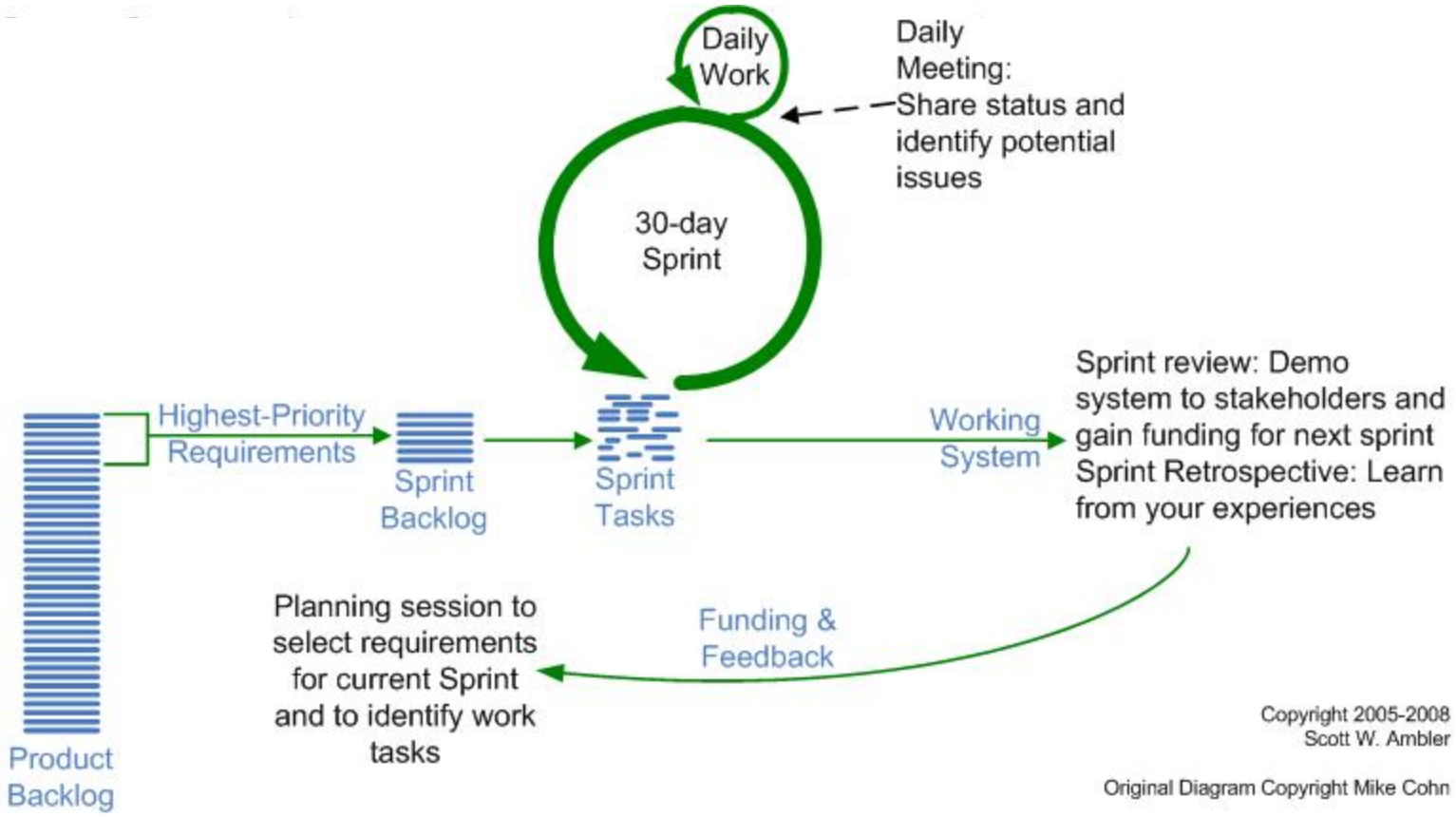
Execution - Scope Management



Handles Change Easily

Suggested Implementation

Execution - Process



Throughput/Capacity Metric

Throughput/Capacity Metric

- # of Story Points completed each month can be used as a throughput/capacity metric
  - If a baseline test objective is defined for the organization then the Story Points completed each month can be aggregated from all the projects within the organization
  - **Average # of Story Points completed per month is an estimate of organizational throughput/capacity**
- Kanban charts can be used to determine where bottlenecks to throughput exist

Can Determine if Organization Can Take on Additional Work

T&E Process Impacts

T&E Process Impacts

- Engineering processes must support Agile principles
  - Agile principles were derived from an understanding of a better way to work
  - To be effective, Agile principles must improve how T&E is executed

T&E Process Impacts

- **Must execute to produce throughput**
  - Focus must be on producing product
  - In model presented, flights will take a back seat to producing analyzed test results
  - Deviating will cause schedule metrics to look bad
    - i.e. Flying all test points early and waiting to the end to process and analyze data will cause schedule metrics to show no progress until the end

T&E Process Impacts

- **Must structure processes with incremental delivery in mind**
  - How do you report completed test objective results every release (3 months)?
    - The form in which results are reported may have to be re-thought
      - Formal Report vs Briefing
      - Structured report template where sections are added easily
        - » Grow report over time
      - Replace with Data Warehouse structure
    - Approval and review cycles must be short
      - Level of approvals and reviews may have to be delegated to lower levels

Benefits

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