

# Maturing the Economics of Agile Development

---

**Jennifer Manning**  
**June 2016**



# Purpose / Outline

**Provide a comprehensive understanding of Agile development cost and economic analytics**

- **Background**
- **Research Goal**
- **Agile Overview**
- **Agile Cost and Economics Toolkit**
- **Summary**

## ■ Large software programs rarely succeed as planned

“Program failures and cost overruns plague three-quarters of large federal IT programs” Rep Darrell Issa, House Oversight & Government Reform Hearing on IT Investment Reforms

Project Size	Successful
Grand	6%
Large	11%
Medium	12%
Moderate	24%
Small	61%

Source: Standish Group 2015 CHAOS Report

- GAO recently added IT Acquisition to its high risk list<sup>1</sup>. *Many of GAO's recommendations explicitly or implicitly promote Agile acquisition*
- FY10 NDAA and FY15 NDAA directed DoD to develop new Acquisition models for IT that enable small, frequent releases with active user involvement
- Agile development has emerged as the leading software development methodology commercially, with growing adoption across federal agencies

<sup>1</sup>GAO 15-290 High Risk Series Report (Feb 2015)

# Research Goal

- Agile development challenges many of the traditional acquisition areas including cost estimation, economic analysis, and system engineering
- **Research Goal: Develop a comprehensive cost and economic toolkit that helps improve acquisition decision making and program success by:**

Characterizing the technical baseline

Developing a Work Breakdown Structure (WBS)

Identifying cost increases, decreases, and key cost drivers

Identifying benefit areas and impacts

Understanding economic indicators

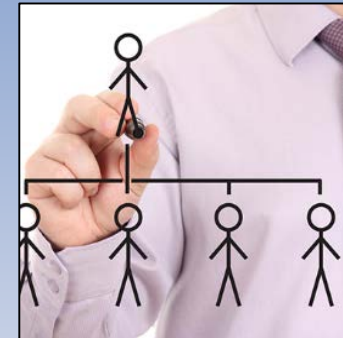
Quantifying schedule impacts

Identifying and assessing risks

Understanding the unique business case considerations faced by Government



## Improve Organizational Agility



**New designs, processes, and incentives to improve how organizations sense and respond to an uncertain environment.**

# Guiding Agile Principles

## Small, Frequent Releases



Iteratively Developed

## Review Working Software



Vice Extensive Docs

## Responsive to Changes



Operations, Tech, Budgets

## Active User Involvement



To Ensure High Ops Value

# Agile Terms and Timelines



**Capability delivered to users**  
**Comprised of multiple sprints**



**Priority capabilities developed, integrated, and tested**  
**Demonstrated to users – Potential to deliver capability**

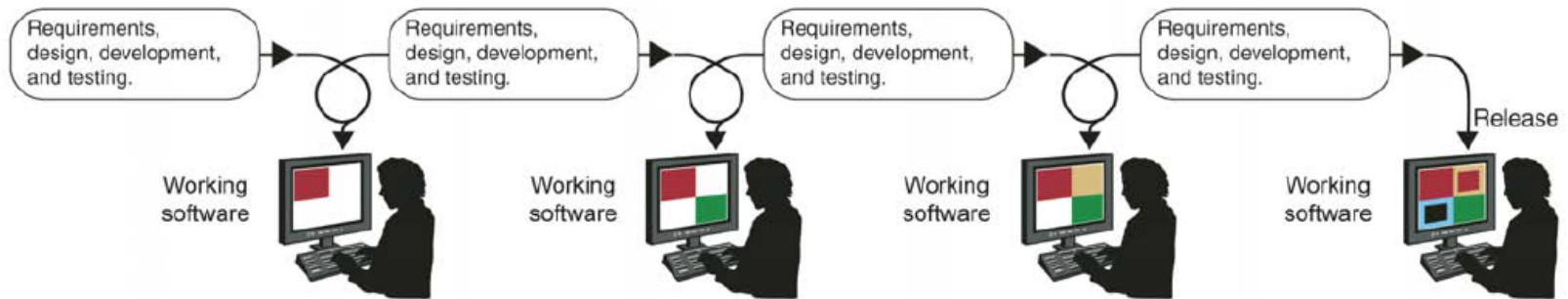
	Design	Develop	Integrate	Test	Demo
Planning	How to go from user story to code	Develop code and track tasks	Continuously, at least daily	Automated and integrated testing	Demo functionality to users and stakeholders

**Agile terms and timelines will vary based on Agile approach used**

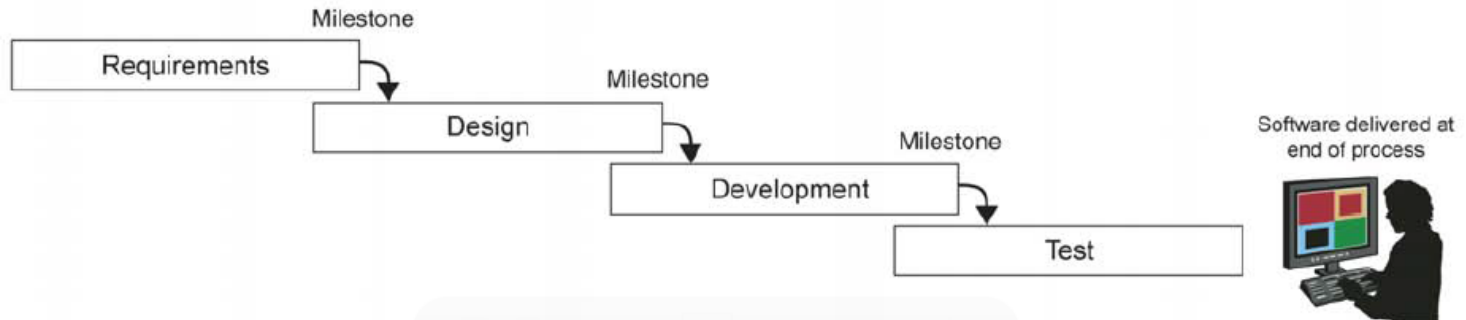
# Agile versus Waterfall

**In Agile, cost and schedule are fixed; scope is variable**

## Agile iterations



## Waterfall phases



**In Waterfall, scope is fixed; cost and schedule are variable**

Graphic Source: GAO Effective Practices and Federal Challenges in Applying Agile Methods (GAO-12-681 July 2012)  
<http://www.gao.gov/assets/600/593091.pdf>

# Work Breakdown Structure (WBS)

- **Purpose:** Develop a comprehensive Agile WBS framework that describes the Agile activities needed to develop, integrate, test, implement, and sustain software across the life cycle
- **Approach:**
  - Identify and review existing related Agile activity breakdowns and decompositions
  - Analyze, compile, modify, and augment with detailed activities to develop a comprehensive life cycle Agile WBS Model and Dictionary
  - Augment and extend WBS Model with extensions relevant to cost analysis community

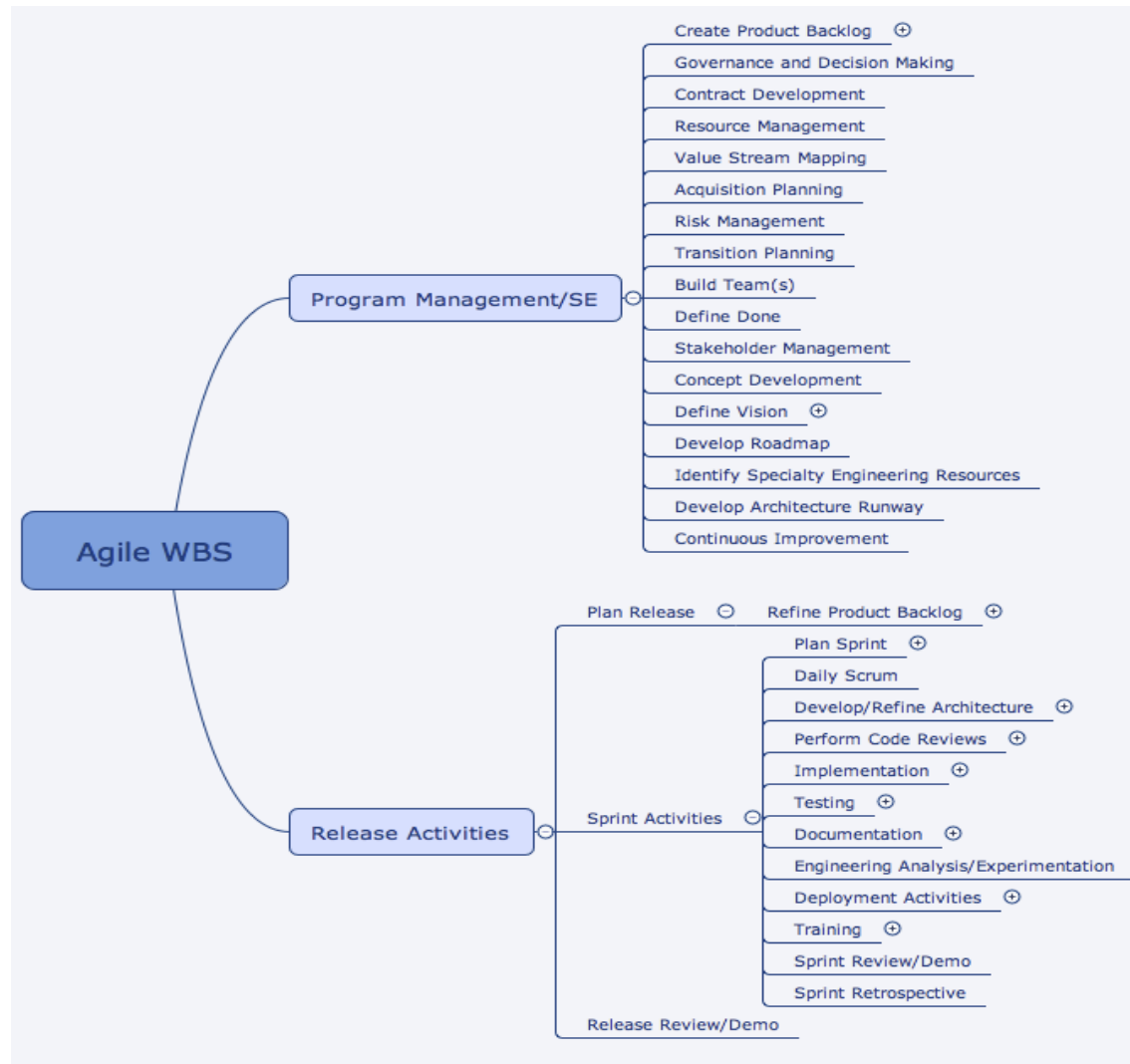
**WBS provides insights on key activities to consider and evaluate**



# WBS (cont.)

Background  
 Research Goal  
 Agile Overview  
 Agile Cost and Economics Toolkit  
 Summary

- **WBS developed in using native Agile paradigm**
  - PM/SE Activities
  - Release Activities
  - Sprint Activities
  
- **WBS includes both development and sustainment activities**



PM = Program Management  
 SE = System Engineering

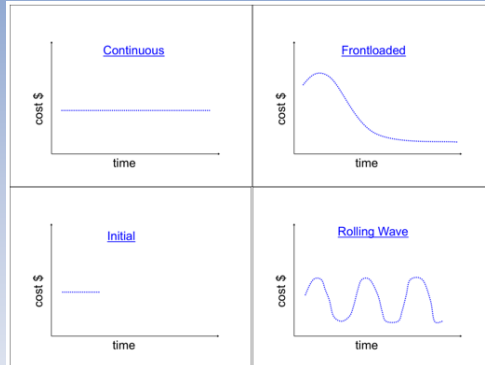
# WBS (cont.)

## WBS Cost Extensions provide additional insights

### Major Cost Element

- PM/SE
- Software Development and Integration
- System Test and Evaluation
- Training
- Fielding/Deployment

### Cost Profile



### Cost Drivers

Cost Driver
# Epics and Complexity
# Work Items in Backlog
Program Size and Scope
# of Stakeholders
Level of Mission Criticality
Program Size and Complexity
# Story Points Committed to Release
# Story Points Committed to Sprint

# Technical Baseline Considerations

Background  
 Research Goal  
 Agile Overview  
 Agile Cost and Economics Toolkit  
 Summary

## Software Development

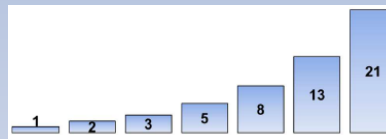
### Sizing

- Story points are commonly used and are native to Agile paradigm
- Story points are a unit of measurement to estimate relative complexity of user stories
- Story points are unique to each team
- Size of entire effort likely uses relative sizing technique

### Complexity

- Story point complexity is relative and can be assigned using:

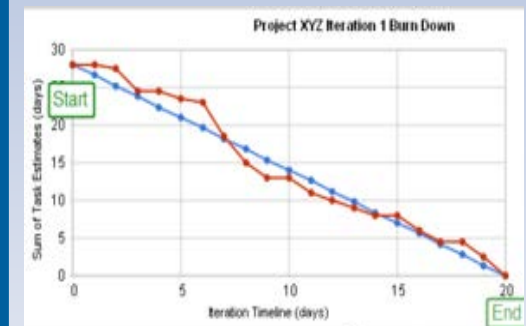
- **Fibonacci Series**



- **T-Shirt Sizing (S/M/L/XL)**
- **Planning Poker**

### Productivity

- Velocity is the amount of work the team estimates it can deliver in a sprint
- Burn Down charts track progress



# Cost Considerations

- **Agile projects do long-term planning**
  - Early-on, upfront analysis will be required that demonstrates a high-level understanding of the program and its associated costs
  - Initial program-level cost estimates will have low fidelity for high-level increments
  - Detailed cost estimates will be developed prior to the development of each release
- **Cost estimating techniques for an Agile development are not necessarily different**
- **Estimating costs in an Agile environment requires an iterative, integrated, and collaborative approach**

# Cost Increases / Cost Decreases

Life Cycle Cost Element	Cost Increases	Cost Decreases
<b>PM/SE</b>	<ul style="list-style-type: none"> <li>* Start-up learning inefficiencies</li> <li>* Some new roles, such as Product Owner</li> <li>* Higher level of engagement for the Government</li> </ul>	None
<b>Software Development</b>	<ul style="list-style-type: none"> <li>* Higher labor rates</li> <li>* Additional unplanned rework during development</li> </ul>	<ul style="list-style-type: none"> <li>* Less unplanned work after development</li> <li>* Higher productivity</li> <li>* Less software growth due to requirements uncertainty</li> </ul>
<b>Integration and Test</b>	<ul style="list-style-type: none"> <li>* Initial costs to set-up automated testing</li> <li>* Continuous integration and testing performed</li> <li>* Regression testing on each release</li> </ul>	<ul style="list-style-type: none"> <li>* Less complex, smaller batch testing</li> <li>* High use of automated testing</li> <li>* Earlier detection of software defects</li> </ul>
<b>Deployment</b>	<ul style="list-style-type: none"> <li>* Initial costs to develop deployment scripts</li> </ul>	<ul style="list-style-type: none"> <li>* High use of automated deployment and DevOps</li> </ul>
<b>Training</b>	<ul style="list-style-type: none"> <li>* More frequent training deliveries</li> </ul>	<ul style="list-style-type: none"> <li>* Greater degree of built-in usability</li> </ul>
<b>Sustainment</b>	<ul style="list-style-type: none"> <li>* Some increases from above elements continue</li> </ul>	<ul style="list-style-type: none"> <li>* Reduction in # of software defects post-deployment</li> <li>* Reduction in enhancement post-deployment</li> </ul>

# Life Cycle Cost Impact

Life Cycle Cost Element	Cost Impact Range	
	Best Case	Worst Case
Program Management/ System Engineering	=	+
Software Development	-	=
Integration and Test	=	+
Fielding/Deployment	=	++
Training	+	++
Sustainment	--	-

++ significant increase, + increase, = no impact, - decrease, -- significant decrease

**Largest area of potential decreased cost lies in Sustainment**

# Benefit Considerations

## ■ Key benefit areas

- Responsiveness to uncertainty and change
- Improved productivity
- Faster deployment of capability to the field
- Improved quality
- Increased user satisfaction
- Program stability



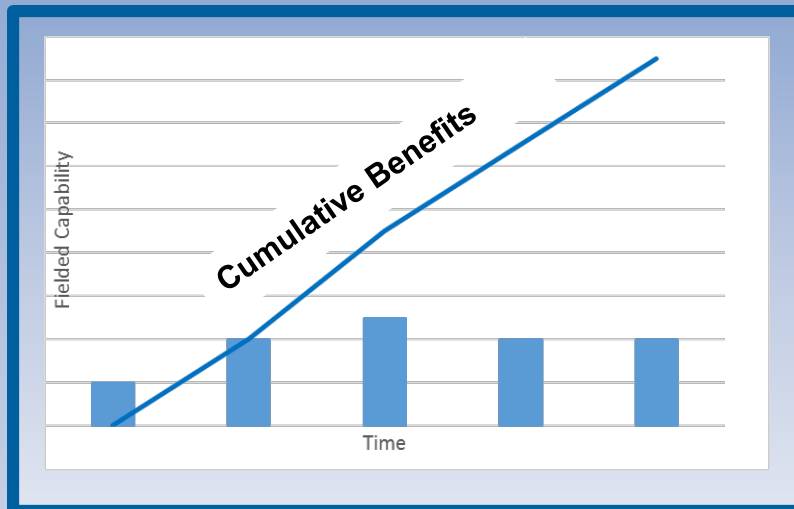
## ■ Data from several studies has shown benefits can be realized through:

- Schedule reduction
- Productivity improvement
- Quality improvement
- Customer Satisfaction improvement

## ■ Data has wide ranges across the studies and ranged from marginal improvement to significant improvement

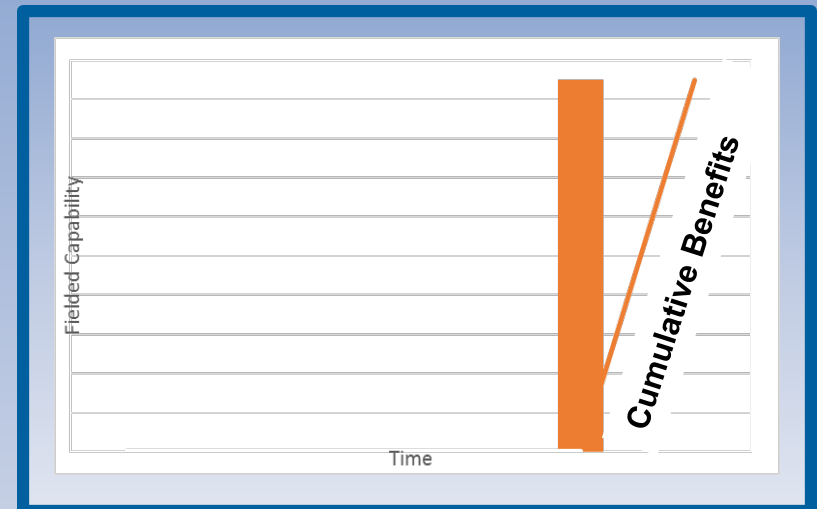
# Benefits Considerations (cont.)

## Agile



- The iterative nature of Agile means earlier realization of benefits. Benefits will start to accrue as software is fielded

## Waterfall



- Waterfall does not field capability until the end of development. Benefits do not begin until this point



# Risk Considerations

- **Technical Debt**
  - Poor management of technical debt can lead to complex and brittle systems
- **Bow Wave of Deferred Functionality**
  - Continually deferring functionality can cause failure to deliver on established cadence
- **Organization Culture**
  - Culture must be conducive to Agile adoption or program implementation will suffer
- **Operational Ability to Accept Releases**
  - If development of new software significantly outpaces the system's ability to accept it, then the operational value of the software decreases as it remains on the shelf
- **Never Ending Optimization**
  - Definition of done must be defined and 'must have' and "nice to have" features identified or the program may expend efforts on improvements with diminishing return
- **Correct Implementation of the Agile Methodology**
  - Programs not correctly implementing Agile methodology will experience management and development issues
- **Provide for Team Coordination**
  - Program must operate as a cohesive social group or Agile implementation will be suboptimal

# Business Case Qualifiers

## Qualifier 1

Quality and experience of team used by both the government and developer

## Qualifier 3

Fit of Agile development to program needs and constraints

## Qualifier 2

Degree of Agile development methods used by both the government and developer

## Qualifier 4

Degree of Agile enablement by external environment (stakeholders and processes)

**Overarching qualifiers impact all business case areas**

# Business Case Considerations

Background  
Research Goal  
Agile Overview  
Agile Cost and Economics Toolkit  
Summary

## ■ Developing and assessing the business case for Agile development presents new challenges...

- How can ROI be determined when requirements aren't fixed?
- How does a business case ensure the benefits of Agile are adequately considered?
- At what levels of Agile process should a business case be conducted?
- What happens life cycle phases are not clearly separable?
- How does a business case assess projects that use hybrid development methods?
- What are the unique challenges of Agile development in the Federal Government?

# Summary

- **Agile is not a panacea and is not always the best fit for every program**
- **Long term planning *is* done in Agile development and cost estimation is a critical activity**
- **Agile development can have some increased cost areas upfront, but may lead to a net decrease in total life cycle costs**
- **Agile promises some real cost savings however, many key benefits of Agile may not be realized as cost savings**
- **Assessing the cost and economics of Agile development is not a ‘one size fits all’ formula**
- **Business case qualifiers affect all cost and benefit areas and need to be considered, particularly for the Federal Government**

Understanding the unique cost and economic aspects of Agile helps improve an organization’s agility and enables them to make timely, data-driven decisions

# Thank You

---

[jmanring@mitre.org](mailto:jmanring@mitre.org)

804-741-3823

**MITRE**

---