Agile Acquisition

Pete Modigliani

pmodigliani@mitre.org

(703) 983-9131

8 May 18

See more at aida.mitre.org

MITRE

Purpose / Outline

Discuss adoption of Agile development practices in federal acquisition

- Agile Overview
- Program Structure
- Requirements
- Contracting
- Cost Estimation
- Testing
- Barriers and Enablers

Agile Guiding Principles

Small, Frequent Releases

Review Working Software Not Extensive Docs

Responsive to Changes in Ops, Technologies, Budgets, Threats

Active User Involvement

Agile Acquisition: How IT Acquisition programs can leverage Agile Software Development practices

Tailored from Agile Manifesto (see backup slide)

Agile Is. . .

- **A mindset and cultural change, not simply a process to follow**
- **A journey of continuous improvement**
- **Tailored for your organization's vision, culture, and needs**

https://enterprise-knowledge.com/ek_agileorganizations/

Large Software Projects Rarely Succeed

Two Simple Rules

1. **Smaller is better**
2. **See #1**

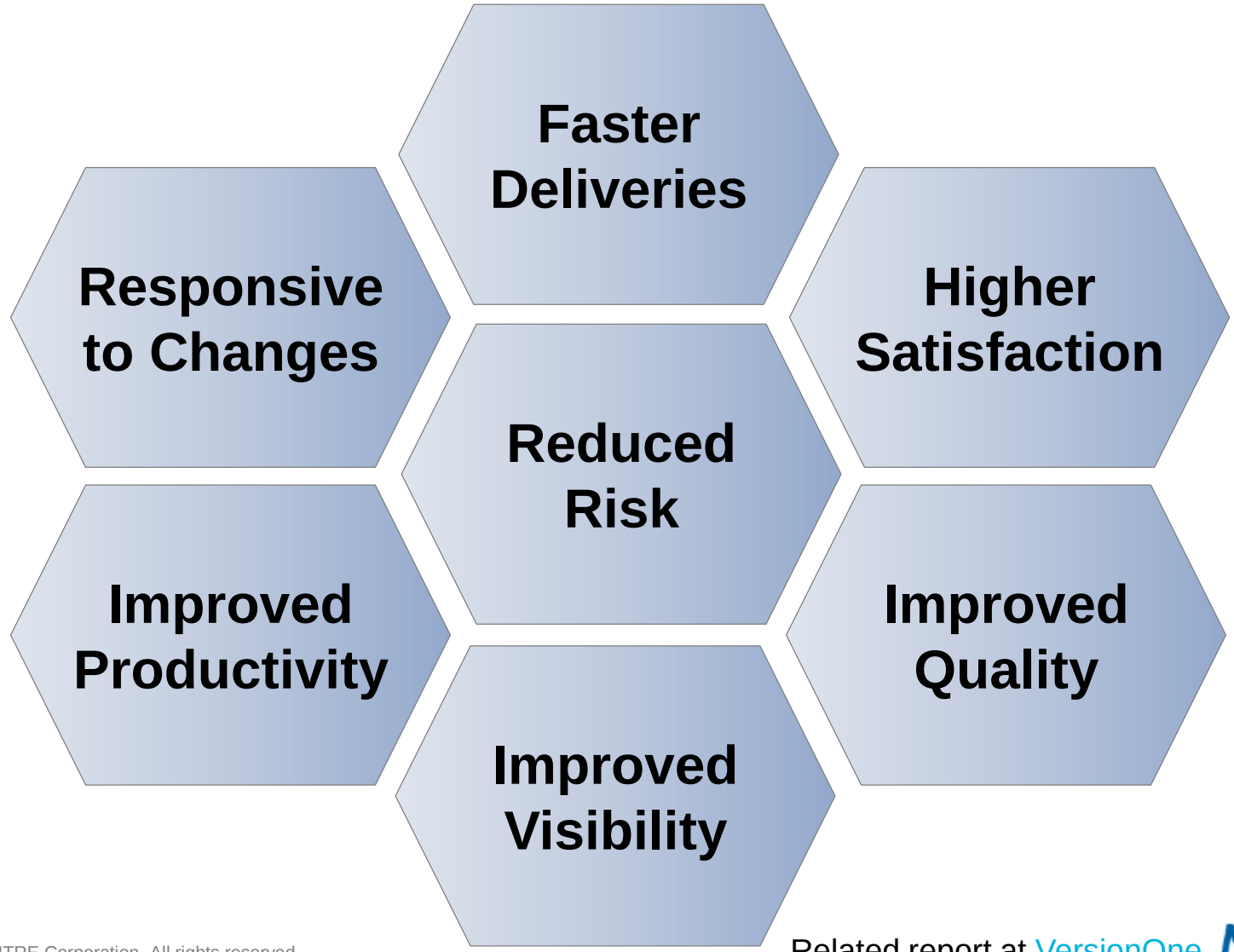
Project Size	Successful*
Grand	6%
Large	11%
Medium	12%
Moderate	24%
Small	61%

*** Success:** On Time, On Budget, Satisfactory Result

Source: Standish Group 2015 CHAOS Report

MITRE

Benefits of Agile



Related report at VersionOne

MITRE

Programs Well-Suited for Agile Adoption

Software Intensive Systems

- Incrementally deliverable
- Rapid delivery
- Short lifespan
- Loosely coupled architecture

Accessible Stakeholders

- Users, relevant representatives can engage throughout
- Able to share ops insights, feedback

Uncertain Solution Space

- Rapidly changing demand, technology
- Evolving requirements

Follow-on Increments

- First increment may be infrastructure via traditional methods
- Iteratively develop capabilities

MITRE

Acquisition via Traditional vs Agile Methods

	Traditional	Agile
Mindset	Define rigid requirements, design, develop, produce	Collaborative culture to iteratively deliver priority capabilities to users
Size/Scope	5-Year Increments	<6 month releases
Requirements	Defined upfront via large requirement documents and contracts	Iteratively defined and prioritized via dynamic backlogs
Contracts	Rigid, product based, long timelines, limited changes	SW Development-as-a-Service via iterative task orders
Cost Estimate	Exhaustive upfront analysis, rigid baselines	Iterative, integrated, collaborative
Testing	Long timelines following system development	Automated, daily, integrated throughout development

Notional Agile Terms and Timelines

Release
≤6 Months

Sprint	Sprint	Sprint	Sprint	Sprint

Capability delivered

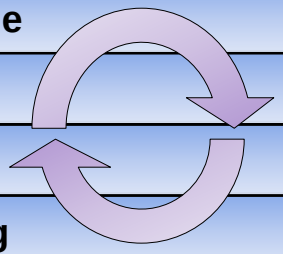
Comprised of multiple sprints

Sprint
≤1 Month

Prioritized capabilities developed, integrated, tested

Demonstrated to users – Potential to deliver capability

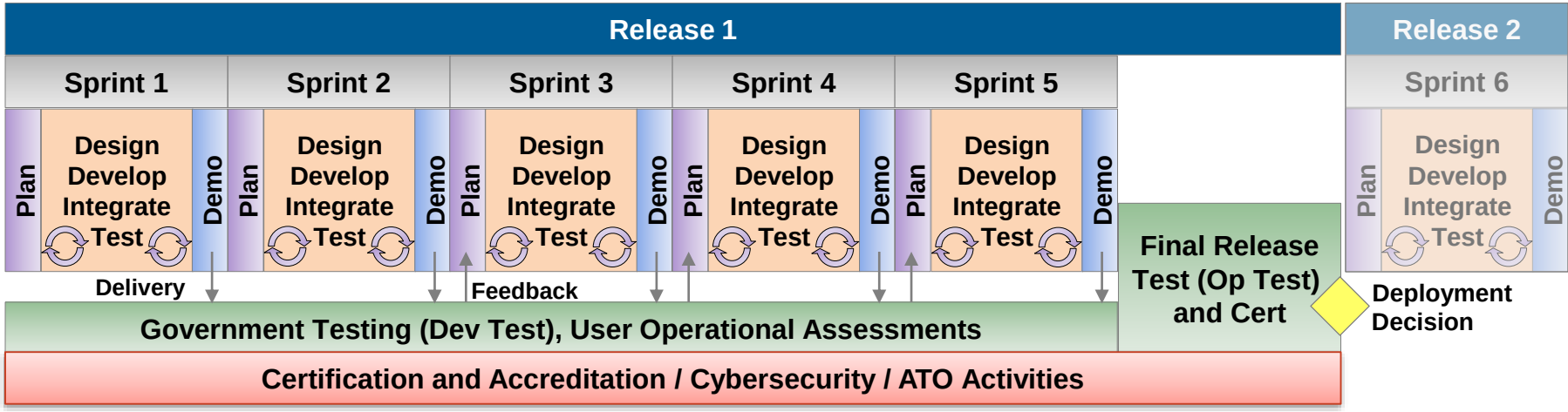
Planning			
	Design	How to go from user story to code	**Demo**
	Develop	Develop code and track tasks	Demonstrate functionality to users and stakeholders
	Integrate	Continuously, at least daily	
	Test	Automated and integrated testing	



Agile terms and timelines will vary and often shorten over time

MITRE

Structuring an Agile Release



- **Develop structured time-box and tailor processes to support**
  - Hold schedule, while flexing scope – Continual improvement
- **Gov't testers, certifiers, and users involved EARLY and often**
  - Minimizes rework and surprises at the end of the release
  - Maximize use of automation – integration, build, regression testing

Lengths Based on Operational, Acquirer, Contractor Agreement

12-month alternative in backup

MITRE

Tailored Agile Model on AiDA

Acquisition in the Digital Age

Agile Fundamentals Overview

MATERIEL SOLUTION ANALYSIS (MSA) PHASE

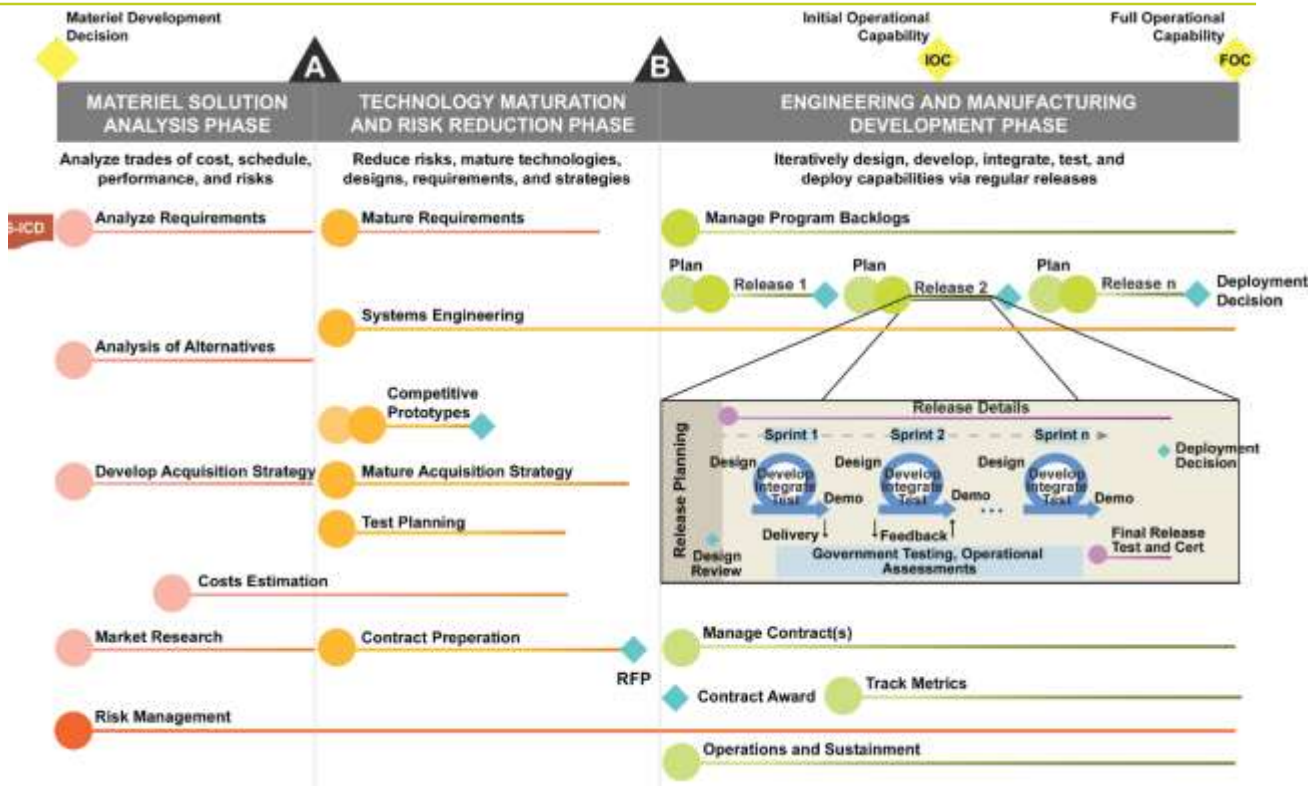
Materiel Development Decision (MDD)
Analyze Requirements
Analysis of Alternatives (AoA)
Develop Acquisition Strategy
Market Research
Cost Estimation
Risk Management

TECHNOLOGY MATURITY AND RISK REDUCTION (TMRR) PHASE

Milestone A
Mature Requirements
Competitive Prototyping
Systems Engineering
Mature Acquisition Strategy
Contract Preparation
Risk Management
Request for Proposal

ENGINEERING & MANUFACTURING DEVELOPMENT (EMD) PHASE

Milestone B
Manage Program Backlogs
Release Execution
Manage Contracts
Metrics
Scaling Agile

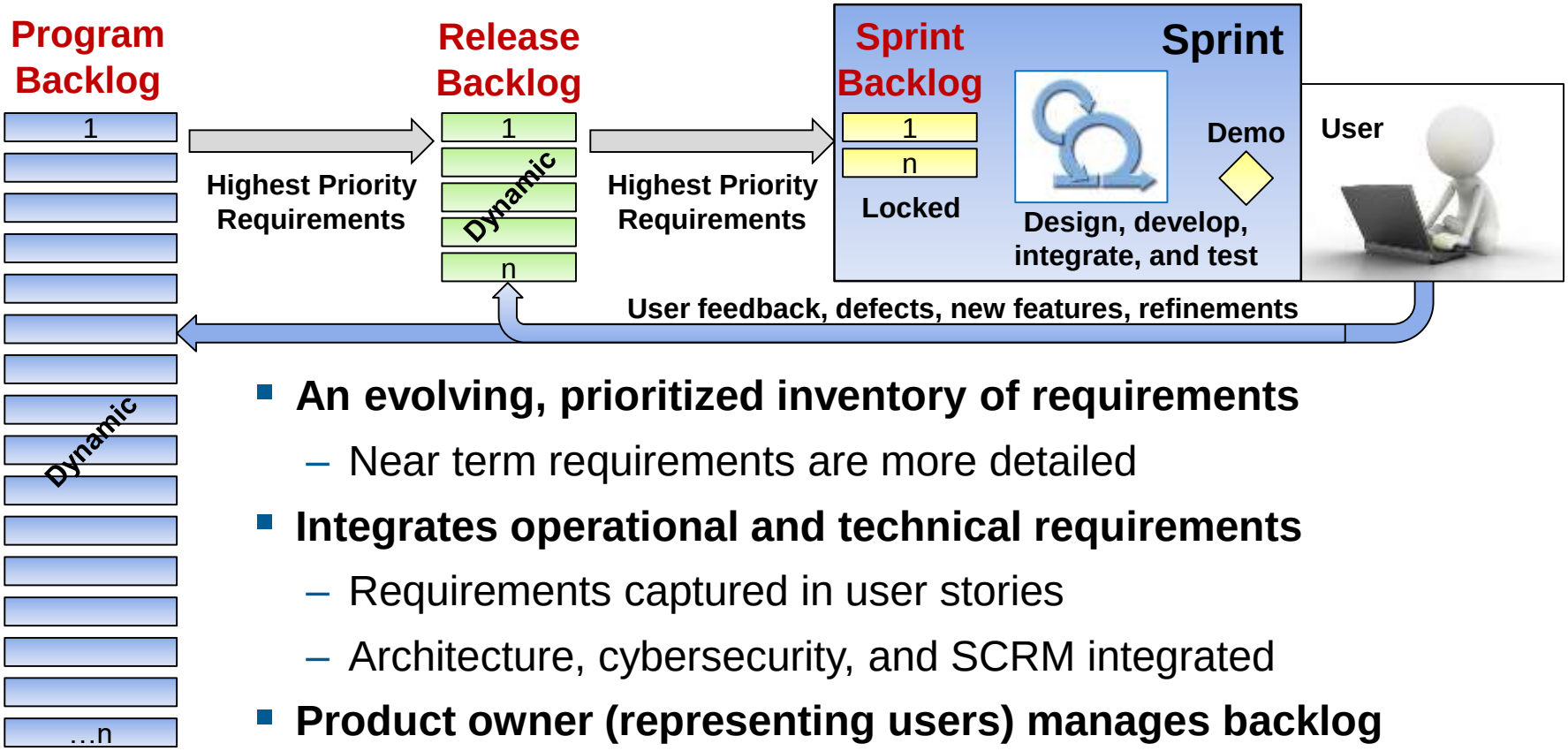


Detailed how-to guidance for adopting Agile by functional area and across the lifecycle

See more at aida.mitre.org

MITRE

Agile Requirements Backlog



- **An evolving, prioritized inventory of requirements**
  - Near term requirements are more detailed
- **Integrates operational and technical requirements**
  - Requirements captured in user stories
  - Architecture, cybersecurity, and SCRM integrated
- **Product owner (representing users) manages backlog**
  - Collaborates with end users on priorities, issues, operations

Iteratively Define, Prioritize, and Tackle Requirements

MITRE

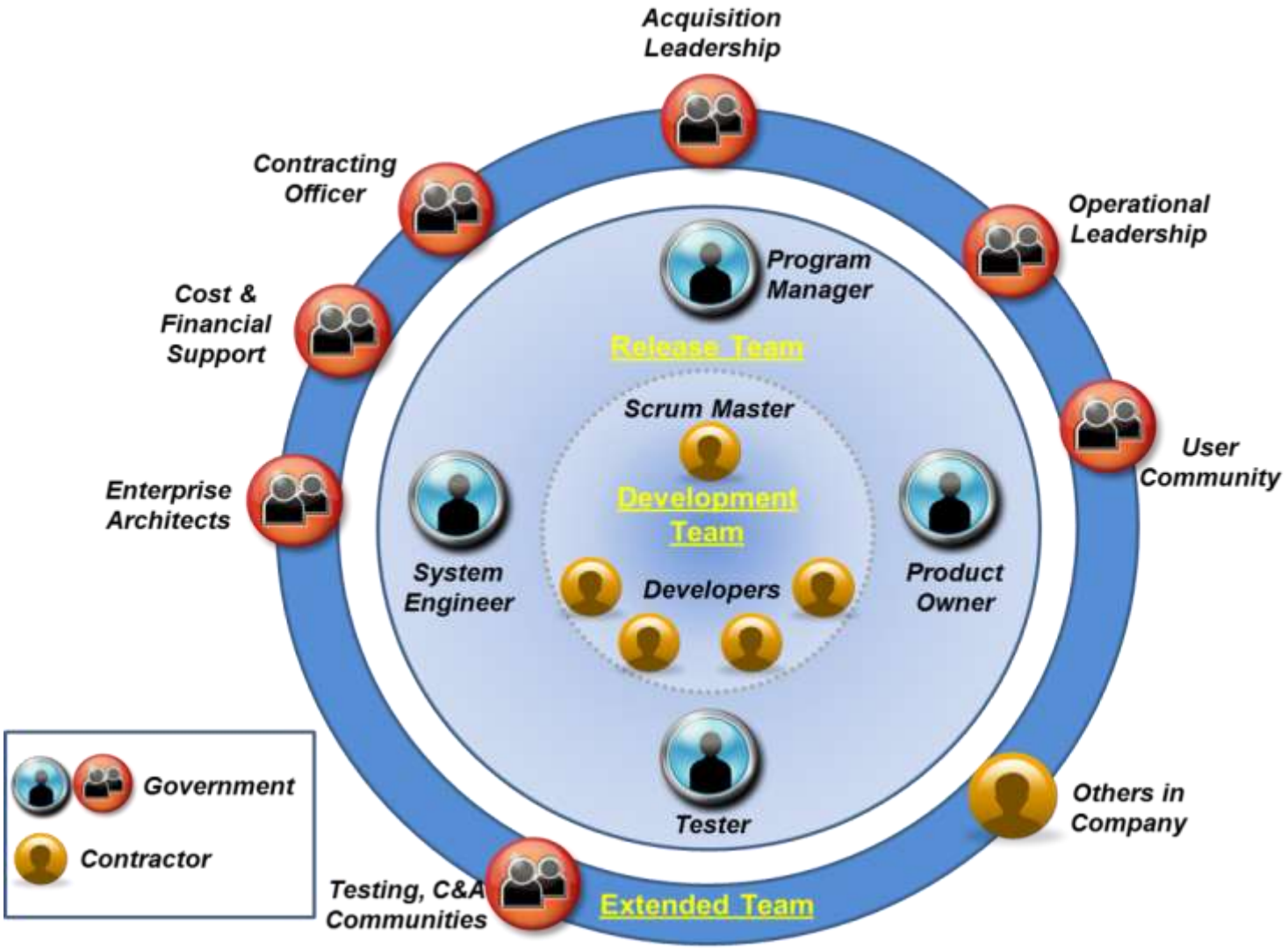
Agile Requirements Can Be in User Stories

As a [user role], I want to [goal] so I can [reason]

As a registered user, I want to log in so I can access subscriber-only content

- **Concise, written descriptions of a capability valuable to a user**
- **High-level description of features**
- **Written in user language, not technical jargon**
- **Provides information to estimate level of effort**
- **Small and succinct**
- **Worded to provide a testable result**
- **Traceable to overarching mission threads**

Potential Agile Team Construct



Multi-Team Alternatives in Backup

MITRE

Traditional vs. Agile Contracts

Contracting Area	Traditional	Agile
Timelines	Rigid contracting processes with long contracting timelines	**Streamlined** contracting processes
Requirements	Defined in detail up front	Ability to **reprioritize requirements**
Scope	**Locked in** at contract award	**Flexible contracts** with ability to "learn" from development processes
Contractors	Traditional contractors with no or limited Agile experience	**Qualified contractors** with experience in Agile
Contract Management	Inconsistent contract management	Close management of **Government contractor relationships**
Incentives	Incentivized to deliver against fixed requirements	Incentivized to be **efficient and collaborative**
Technical Evaluation	Award made based on strength of the technical solution	Award based on **strength of the team** and experience with Agile

Product vs Services Based Contracts

	Product	Services
Core Strategy	Acquire a defined software product	Acquire time of an Agile developer
Selection	Strongest technical solution	Strongest development team
Requirements Changes	Costly and Timely	Flexibility
Gov't / Contractor Relationship	Separate – Less visibility	Enables close teaming
System Integrator	Contractor	Government
Development Strategy	Contractor Driven	Government Driven
Contract Types	FFP, Cost Reimbursement Completion	FFP, T&M, Cost Reimbursement Term
Summary	More difficult for Agile	Best option for Agile

Elements of a Potential Agile Contract Strategy

- **Portfolio-level (capability-based) agile development contract**
  - Manage development via quick execution of orders for each release (e.g., 6 months) within each phase
  - T&M for max flexibility (transition to FFP or CR after initial period)
  - Scope and requirements can adjust over time

- **Services-based contract: SW Development As A Service (SDAAS)**
  - Contract for the services of the development team
  - Cost-boxed and time-boxed releases and sprints
  - Requirements in product backlog are flexible, prioritized by Gov't
  - Structure releases (e.g. 6 months) via separate task orders

Devise Strategy to Embrace Change and Rapid Deliveries

Cost Considerations

- **Cost estimating techniques for an Agile program are not very different, but do require an iterative, integrated, and collaborative approach**
- **Early-on, cost estimates will be required for the entire project**
- **Detailed cost estimates will be developed prior to each release**
- **Agile promises some real cost savings however, many key benefits of Agile may not be realized as cost savings**

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

MITRE

Scaling Agile for Large Programs

- **Requires sound engineering discipline**
  - Integration of multiple smaller efforts increases complexity
- **Robust Enterprise Architecture required**
  - Provides framework for individual efforts to map to
- **System Performance**
  - Designs and testing on system-of-systems and individual component

- **Strategies, Backlogs, Roadmaps**
  - Define clear program structure that defines mission and business environments
- **Cross-Team Integration**
  - Frequent collaboration across development teams to track progress, issues, and solutions

See the multiple scaling techniques at https://aida.mitre.org/agile/scaling-agile/

Testing in an Agile Environment

- **Testing is continuous and heavily automated**
  - As developers check-in code
  - Nightly integration testing
  - Testers are embedded in development team
- **Test documentation key to a minimum**
- **Test Driven Development**
  - Define test cases upfront
  - Develop software until passes tests
- **Active collaboration with product owner**

Agile Shortens Testing Timelines – Early ID of SW Issues

Barriers and Enablers for Agile Adoption

	Barriers	Enablers
Culture	Change = Risk Heavy Oversight Resistance to Tailoring	Small empowered teams Delegated decisions Review SW, not docs
Processes/Policies	Long Timelines Define requirements upfront Costly contract mods	Capstone contracts Requirements backlogs SW Dev as a Service
User Involvement	Few end-users available	High bandwidth comms Demo interim sprints
Program Structure	Fixed Scope/Requirements APB/EVM Management Limited ent arch, standards	Small iterative releases Tailored processes Ent arch guides dev
Experience	Limited experience in government and industry	Train gov't/ctr teams Use Agile coaches

Enable Agile Adoption for Enterprise/Program

- **Clearly define WHY go to Agile**
  - What operational, acquisition, development factors are driving Agile?
- **Provide visible leadership for Agile**
  - Shape culture, champion new methods, risk tolerance to adopt Agile
- **Tailor environment to enable Agile success**
  - Redesign core processes, reviews, docs for small frequent releases
  - Get stakeholders from each functional area to shape new environment
  - Identify key metrics to track, reward early success and failures
- **Identify target software intensive programs/areas to pilot Agile**
  - Provide pilots a quality team and maximum flexibility
- **Gov't and Contractors go through Agile training together**
  - Bring in Agile SMEs as coaches to guide new roles and processes

BACKUP SLIDES

Agile Manifesto – Values and Principles

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. **Continuous delivery of valuable software**
2. **Welcome changing requirements**
3. **Deliver working software in weeks/months**
4. **Work together daily**
5. **Build projects around motivated individuals**
6. **Face-to-face conversation**
7. **Working software is the measure of progress**
8. **Promote sustainable development**
9. **Good design enhances agility**
10. **Simplicity is essential**
11. **Self-organizing teams**
12. **Reflect on how to become more effective**

http://agilemanifesto.org

MITRE

GAO: 10 Best Practices for Agile Adoption

GAO Report 12-681 Effective Practices and Federal Challenges in Applying Agile Methods

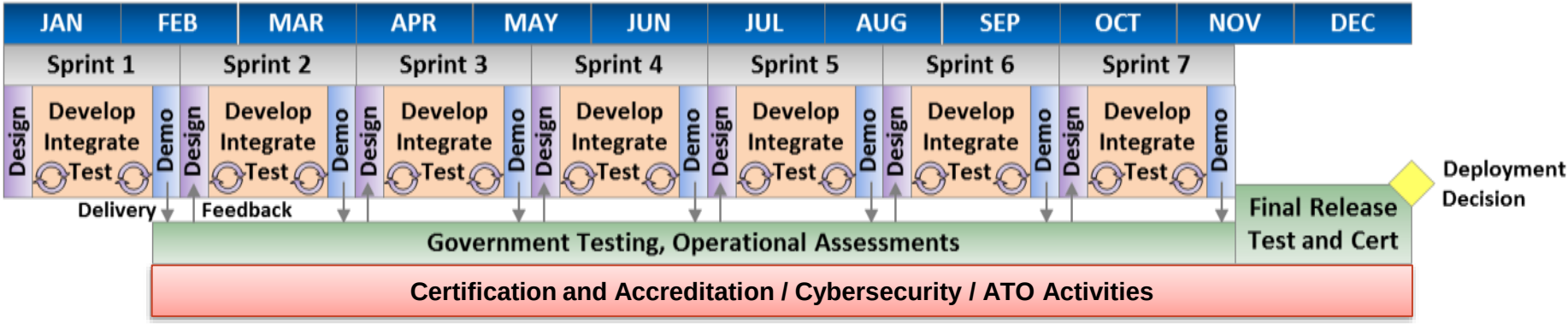
1. **Start with Agile guidance and an Agile adoption strategy**
2. **Enhance migration to Agile concepts using Agile terms, such as user stories (used to convey requirements), and Agile examples, such as demonstrating how to write a user story**
3. **Continuously improve Agile adoption at both the project level and organization level**
4. **Identify, address impediments at the organization and project levels**
5. **Obtain stakeholder/customer feedback frequently**
6. **Empower small, cross-functional teams**
7. **Include requirements related to security and progress monitoring in your queue of unfinished work (the backlog)**
8. **Gain trust by demonstrating value at the end of each iteration**
9. **Track progress using tools and metrics**
10. **Track progress daily and visibly**

GAO: 14 Challenges to Agile Adoption

GAO Report 12-681 Effective Practices and Federal Challenges in Applying Agile Methods

1. Teams had difficulty collaborating closely
2. Procurement practices may not support Agile projects
3. Teams had difficulty transitioning to self-directed work
4. Customers did not trust iterative solutions
5. Staff had difficulty committing to more timely and frequent input
6. Teams had difficulty managing iterative requirements
7. Agencies had trouble committing staff
8. Compliance reviews difficult to execute within an iteration timeframe
9. Timely adoption of new tools was difficult
10. Federal reporting practices do not align with Agile
11. Technical environments were difficult to establish and maintain
12. Traditional artifact reviews do not align with Agile
13. Agile guidance was not clear
14. Traditional status tracking does not align with Agile

Alternate Agile Structure: 12 Month Release



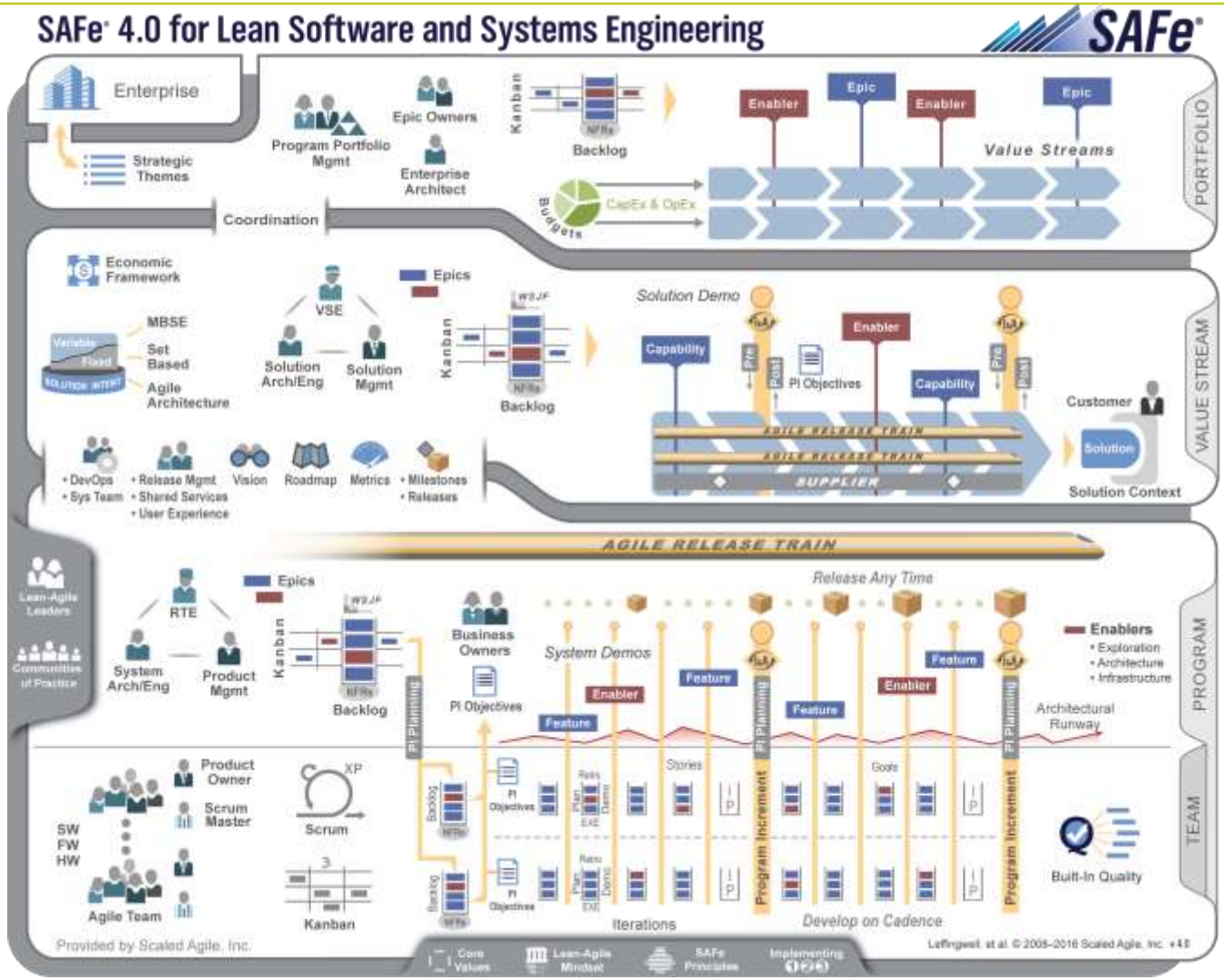
- 6-Week Sprints
  - Continual development, integration, and testing
  - Regular demonstration of capabilities to users
- Gov't Dev Test, Op Test, C&A, Users involved early and often
  - Minimizes work at the end of the release

Release Length Based on Program, Ops, and Technical Risk

MITRE

Scaled Agile Framework (SAFe)

http://www.scaledagileframework.com/



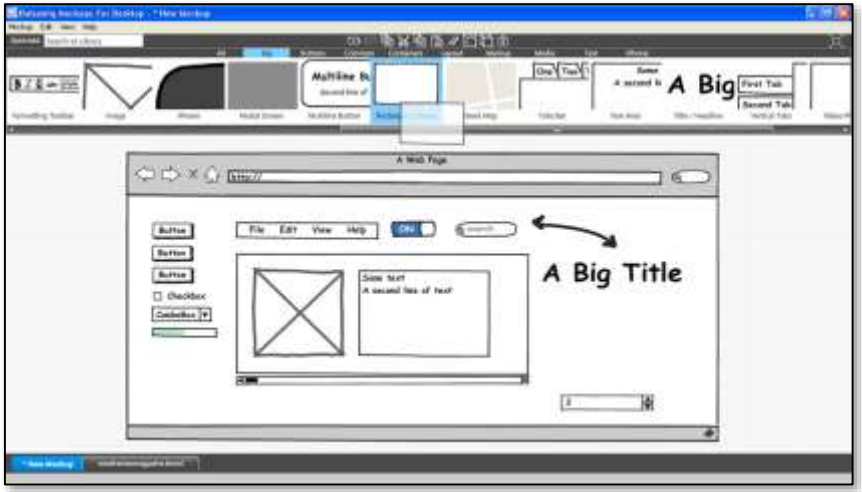
MITRE

Storyboards and Mockups Help Team Visualize the System and Features

Storyboards

Narrative visual depictions set in time to describe system use

Mockups



Visual depictions of the feature of the system

MITRE

Contracting For Agile

- **Commercial firms use in-house developers**
  - Government requires contracted support
- **Contracting Challenges**
  - Complex laws and regulations
  - Long contracting timelines
  - Costly change requests
  - Defined requirements to select contractor

Design Contract Strategies to Support Short Delivery Timelines

How is Contracting For Agile Different?

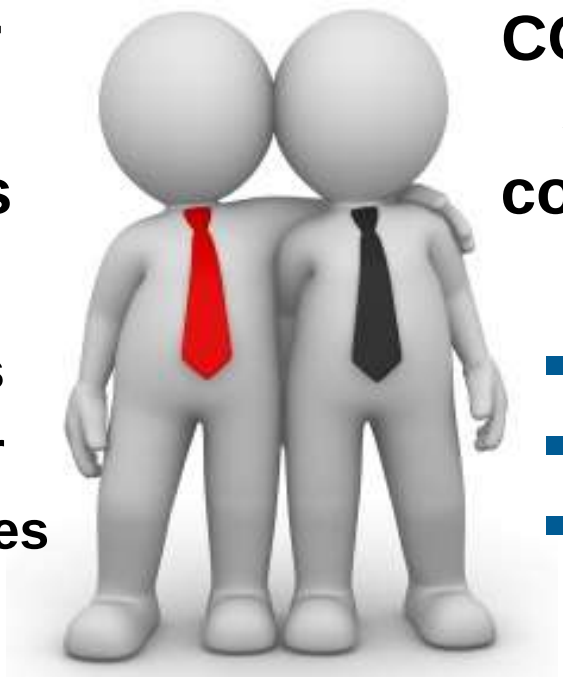
- **Traditional IT acquisition programs contract for completion-based end-product capability based on defined requirements**
  - Requirements locked; changes handled by contract modifications
- **Agile development achieved by acquiring a development team delivering labor hours vs. a defined end-product**
  - Services contract provides flexibility to change release requirements continuously
  - Designed to support short development and delivery timelines and changes
  - Gov can issue orders for each release based on requirements captured in product backlog

Design Contract Strategies to Support Short Delivery Timelines

Contracting Officer = Agile Business Partner

PMs must partner with COs early to develop strategies

- **Streamline processes**
- **Incentivize contractor**
- **Responsive to changes**



Program Manager **Contracting Officer**

CO is key linchpin to a successful Gov't – contractor partnership

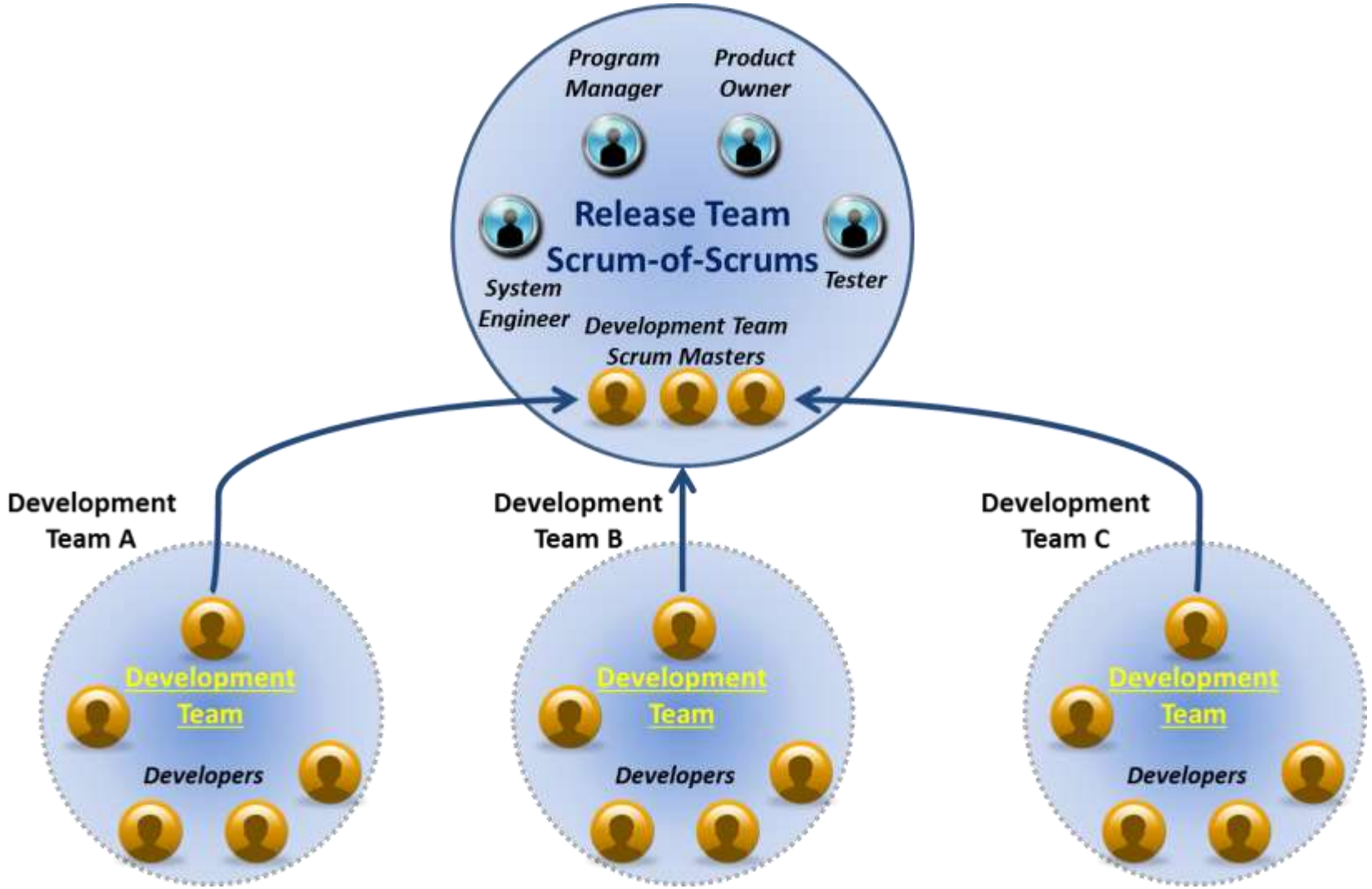
- **Active collaboration**
- **Dedicated support ideal**
- **Manage performance**

"This is all about asking the program manager and the contracting officer to take a higher workload and more risk in exchange for the greater good."

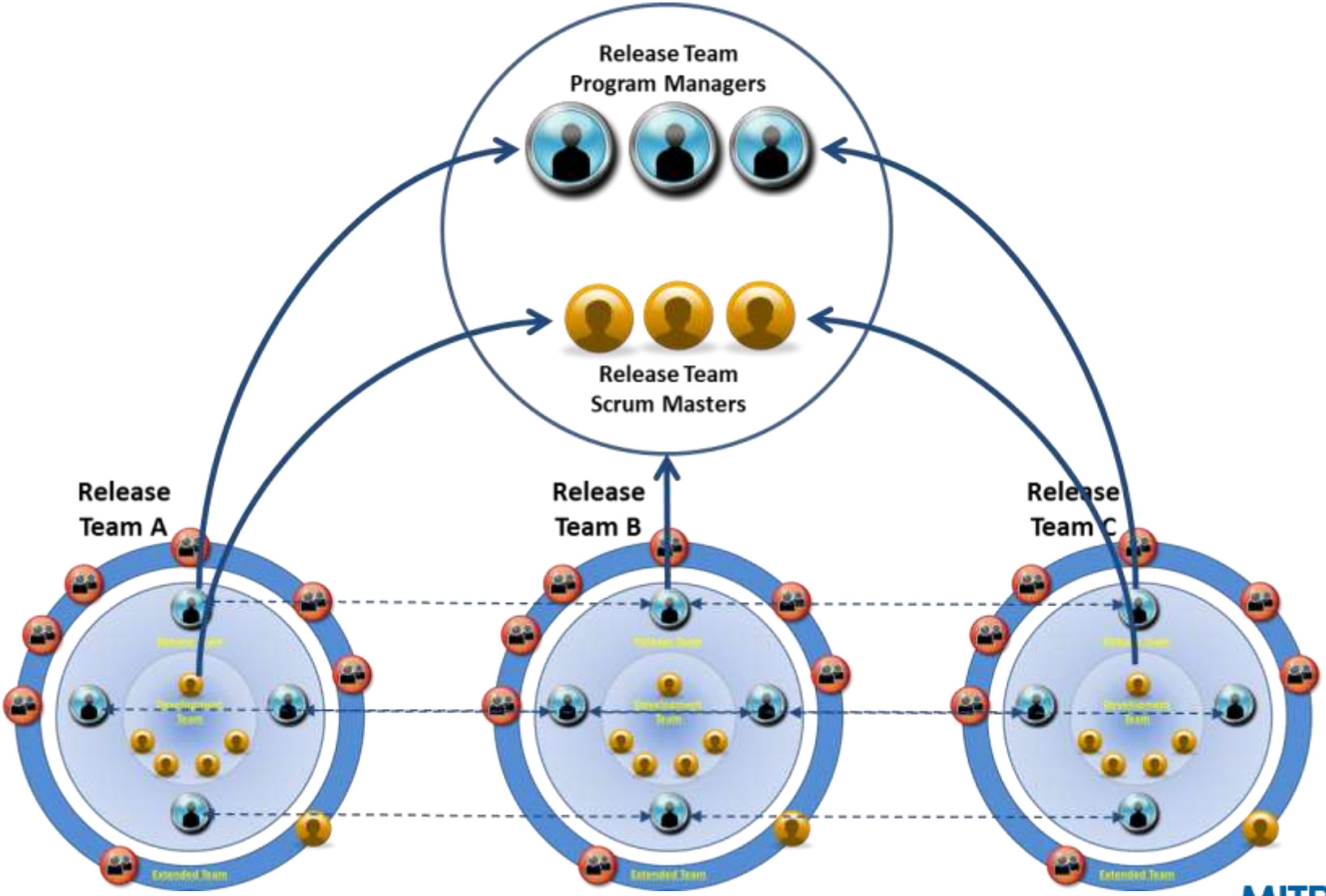
- John Inman, Contracting Officer for US CIS

MITRE

Potential Agile Team Construct 2



Potential Team Construct 3



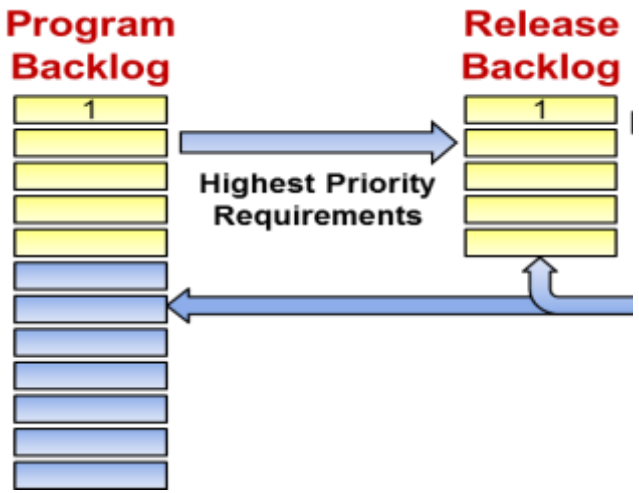
MITRE

Requirements in Context

Contract Requirements

Tasks and activities that describe what the contractor needs to provide under a contract

(“6 FTE software development support”)



Agile Requirements

Functionality or capabilities often expressed in user stories and managed in a product backlog

(“Content search capability”)

Understand the Difference To Avoid Miscommunication

MITRE

Contract Vehicles

Multiple Award Contract	Single IDIQ Contract	GSA BPA
		GSA
IDIQ contract awarded to multiple contractors who compete for work via task orders	IDIQ contract awarded to single contractor with task orders to develop releases	Existing GSA Schedule contract (eg. Sched 70) w/releases developed via call orders

- **Consider a PEO, portfolio, or enterprise-level contract vehicle**
  - Streamlined contracting processes result in faster awards, deliveries
  - Standardized, effective, and efficient contract management