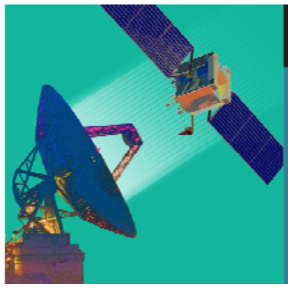
Ground System Architectures Workshop



Session 11A

Challenges and experiences of adopting agile ground software development

Supannika Mobasser and Steve Rosemergy

The Aerospace Corporation

AEROSPACE

Overview

- *To embrace the rapid rate of change in ground software system development, it is crucial to be flexible, resilient, and robust. May be we should use Agile, but …..*

- ***Would Agile be a good fit for large scale mission-critical programs?***
- ***How do we know whether we should use Agile?***
- ***Should we use Agile as it is used in the small scale commercial software-intensive development projects?***
- ***How should we tailor the process so that it is still Agile but also be compliant with government regulations?***
- ***What to do with the contract and plan-driven processes?***

Agile and Rapid Rate of Change

To embrace the rapid rate of change, Agile:

- Welcome changing requirements
- Quick turnaround time
- No big design up front
- Just in time and just enough

But

- Difficult things take a long time; impossible things, a little longer
- Too large to move fast
- But the processes and regulations require …………..
- May be we just have to be agile, but not do Agile.

Session Goal

- Provide an opportunity for agile practitioners to share their experiences and learn from others
- Working group format
  - Presentations followed by Interactive discussions
  - Open discussion
- Contributors
  - Participants with all levels of agile expertise are welcome

Introduce yourself

- Your name, affiliation
- What is your concern, issue about Agile?
- What would your expected takeaways from this working group?

Schedule

Time	Presentation and Discussion
1:00 – 1:30pm	Session Overview
1:30 – 2:15pm	“Embracing Change to Overcome Challenges with Large-Scale Agile Software Development”, Steve Rosemergy, The Aerospace Corporation
2:15 – 3:00pm	General discussions - Pains, struggles, and barriers in adopting agile ground software development - Failed Agile attempts - Going forward, how to increase agility?
3:00 – 3:30pm	Break
3:30 – 4:15pm	“Agile Fit Check”, Sue Mobasser, The Aerospace Corporation
4:15 – 5:00pm	Free Form Discussion

AEROSPACE

Manifesto for Agile Software Development

http://agilemanifesto.org/

“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 & interactions	over	**Processes** & tools
Working **software**	over	Comprehensive **documentation**
Customer **collaboration**	over	**Contract** negotiation
Responding to **change**	over	Following a **plan**

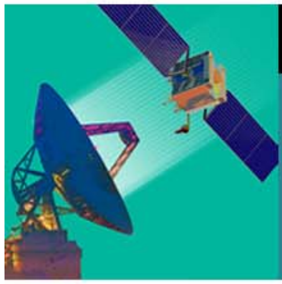
That is, while there is value in the items on the right, we value the items on the left more.”

AEROSPACE

Agile 12 principles

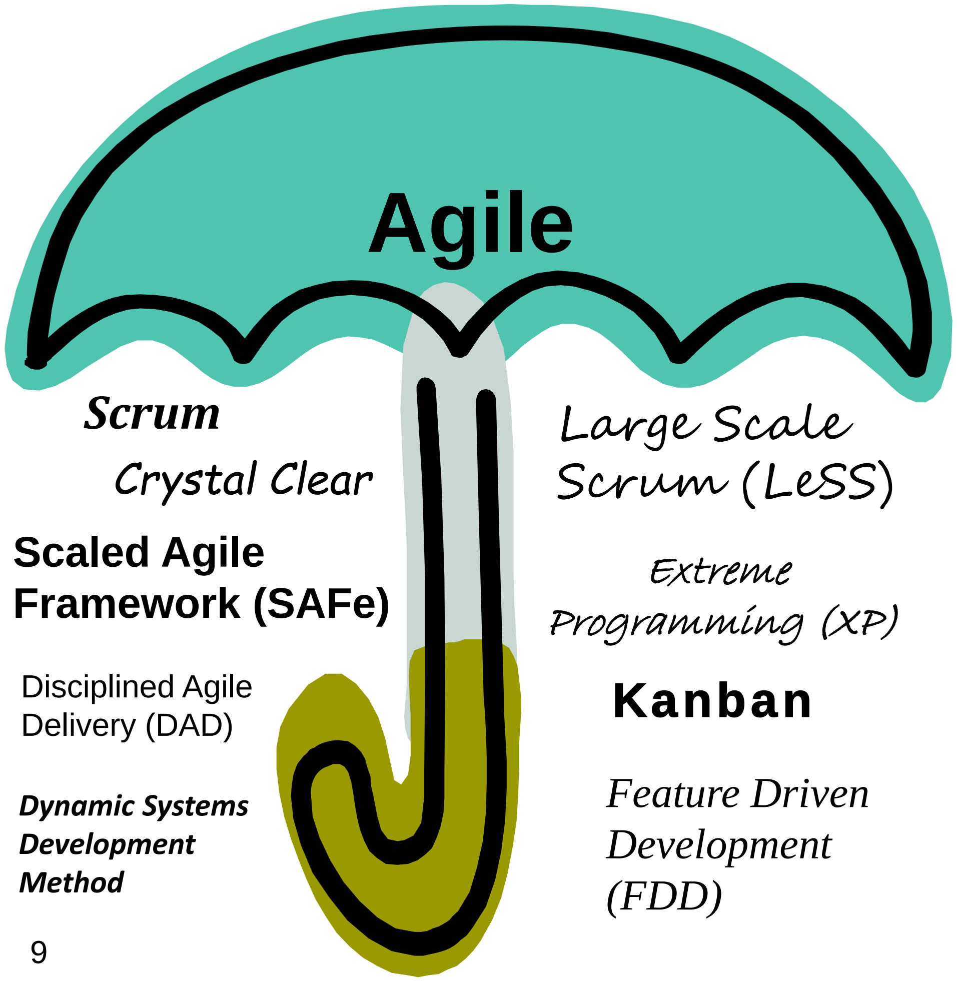
http://agilemanifesto.org/principles.html

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 to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of 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.

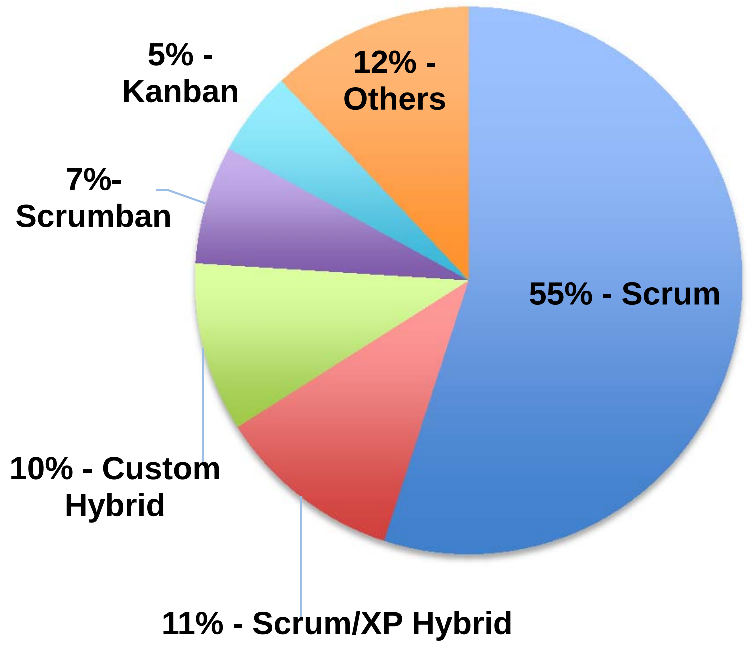


Agile Methodologies

Scrum: the most popular agile methodology in the commercial sector


Agile
Scrum
Crystal Clear
Scaled Agile Framework (SAFe)
Disciplined Agile Delivery (DAD)
Dynamic Systems Development Method
Large Scale Scrum (LeSS)
Extreme Programming (XP)
Kanban
Feature Driven Development (FDD)

Agile methodology Used



[Source: VersionOne 2013]

Agile characteristics

Short Iterations

High interactions and collaborations

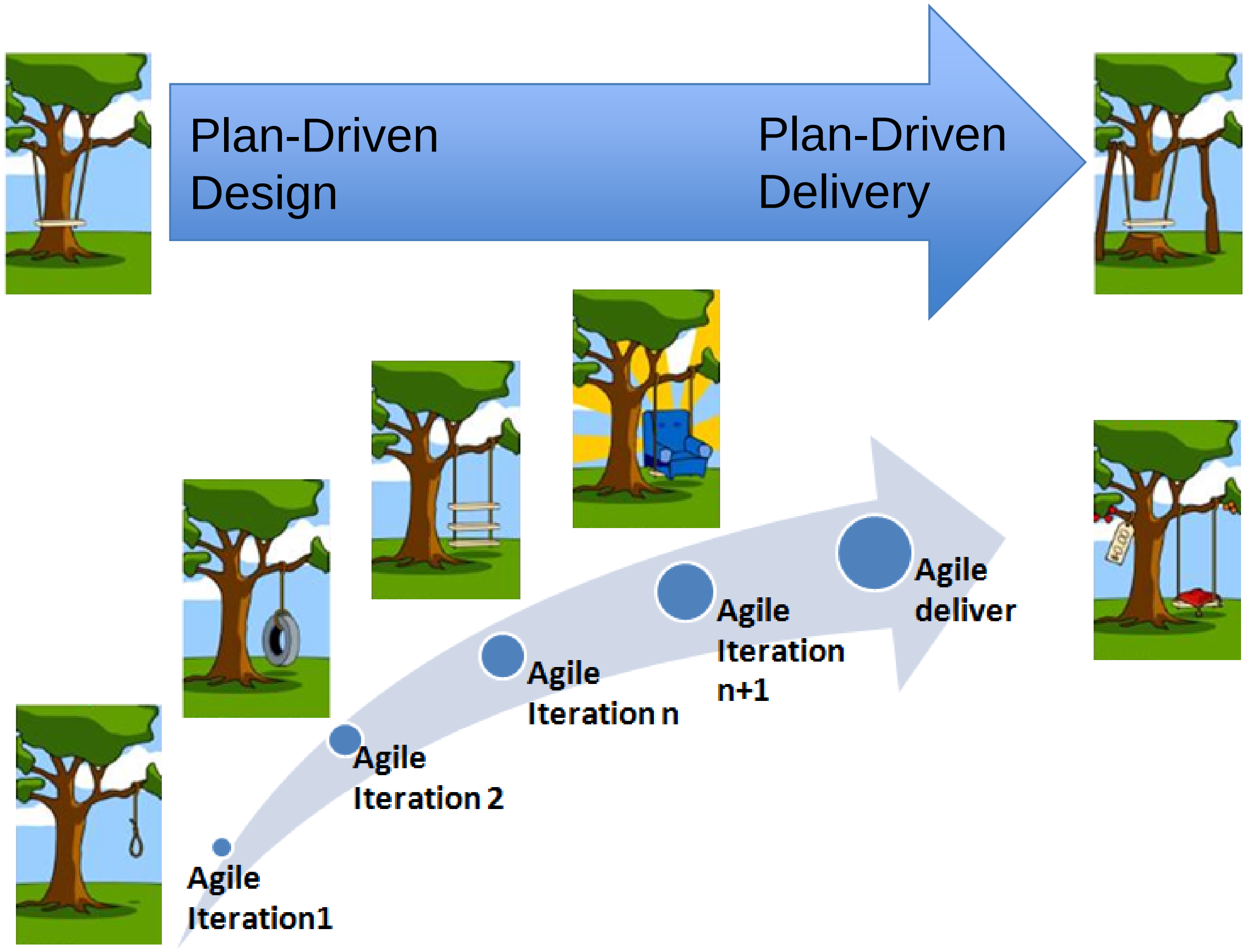
Minimum essential documentation

Measure progress from working software

A team of 3-9 people

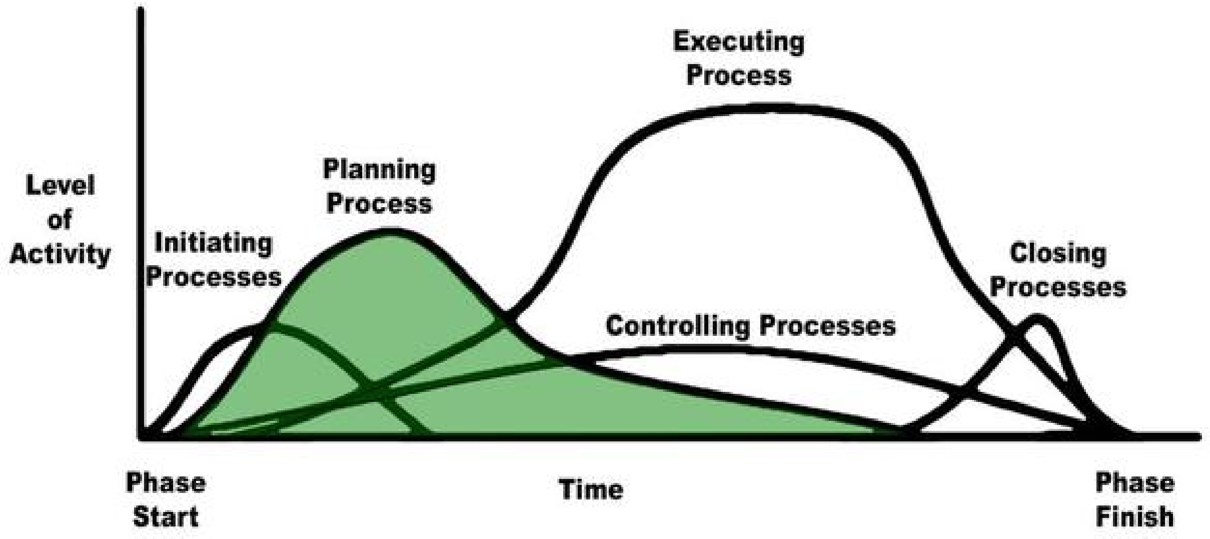
$$\text{OK with} \left\{ \begin{array}{c} \text{Reqm} \\ \text{Design} \\ \text{Plan} \\ \text{Code} \\ \text{Scope} \end{array} \right\} \text{Volatility}$$

Agile vs Plan-Driven Methods

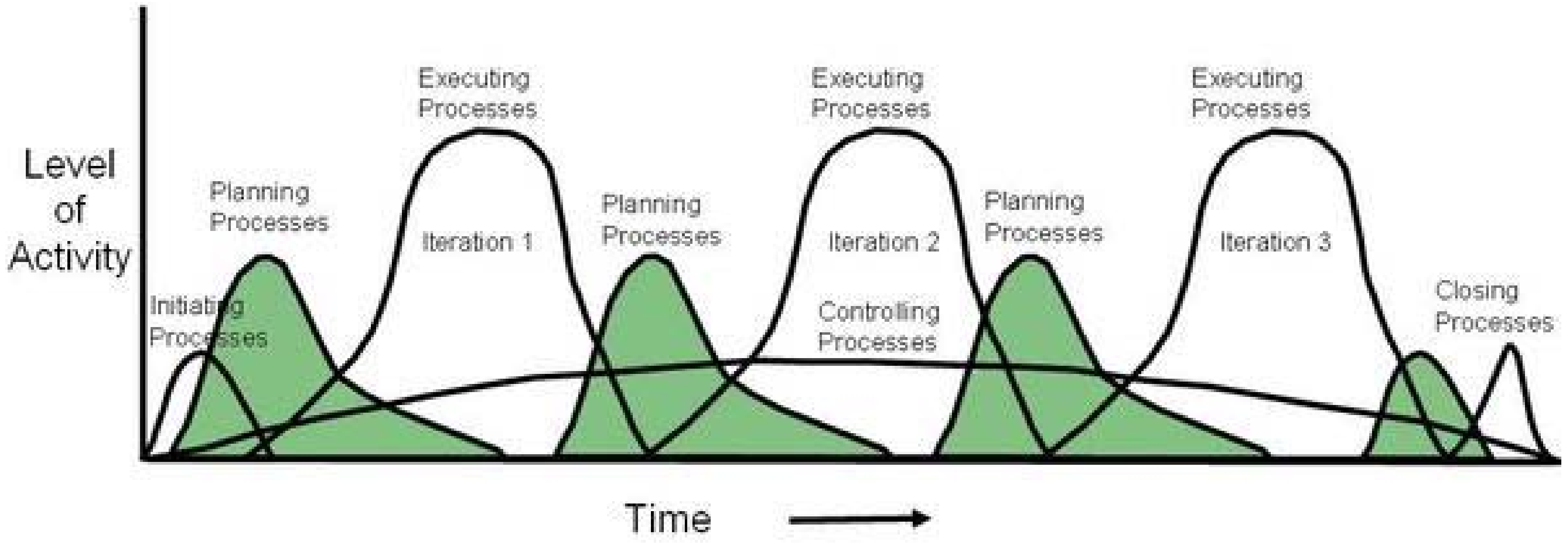


Ref: adapted from https://umangsoftware.wordpress.com/tag/waterfall/

Plan-Driven Project Management Lifecycle



Agile Project Management Lifecycle



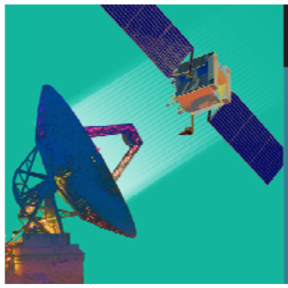
Ref: http://rgalen.com/agile-training-news/2012/5/30/the-agile-project-managerno-upfront-estimates.html

Schedule

Time	Presentation and Discussion
1:00 – 1:30pm	Session Overview
1:30 – 2:15pm	“Embracing Change to Overcome Challenges with Large-Scale Agile Software Development”, Steve Rosemergy, The Aerospace Corporation
2:15 – 3:00pm	General discussions - Pains, struggles, and barriers in adopting agile ground software development - Failed Agile attempts - Going forward, how to increase agility?
3:00 – 3:30pm	Break
3:30 – 4:15pm	“Agile Fit Check”, Sue Mobasser, The Aerospace Corporation
4:15 – 5:00pm	Free Form Discussion

AEROSPACE

Ground System Architectures Workshop



Session 11A

Embracing Change to Overcome Challenges with Large-Scale Agile Software Development

Steve Rosemergy

The Aerospace Corporation

AEROSPACE

Agenda

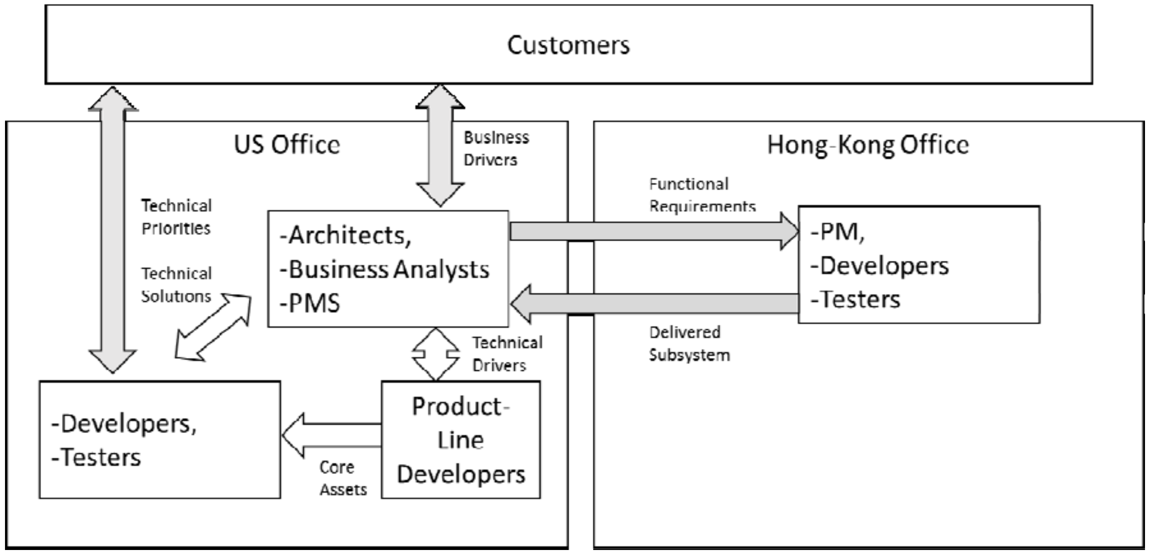
- Facing Agile Software Development Failure: A Case Study
- Lessons on Scaling Agile Software Development
- Adopting Agile: Recommendations for Government Acquisition

Case Study in Agile Adoption & Scaling

- Company X: software company, with 8000 employees distributed across 4 continents
- Product: software product line: Financial Services Software, many derivative products from a core product
- Code Size: 5M+ SLOC
- Platform: Multi-platform Fat-Client & cloud-based services
- Agile Method: Tailored Scrum
- Context: Initial success adopting agile method locally; success using off-shore teams to extend software

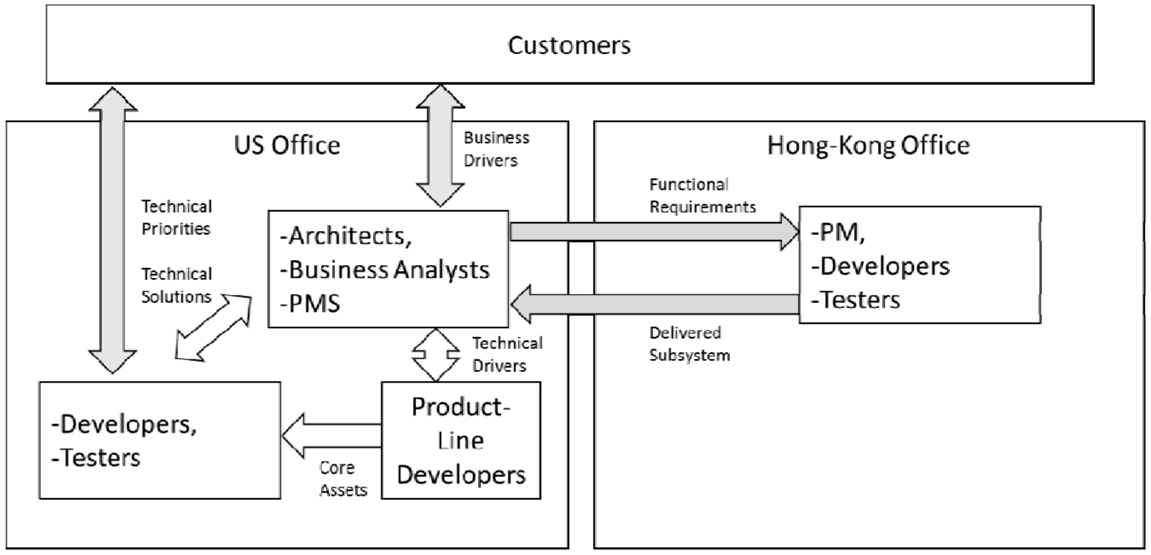
Organizational Context and Customers

- Architects and business analysts spanned multiple agile teams
- Customers interfaced with Business Analysts, Architecture, and Development Team
- Requirements and customer interaction was managed by US office
- Hong-Kong team interfaced with architects, business analysis, and project managers to build extensions to products



New Task: Build Product for Asian Markets

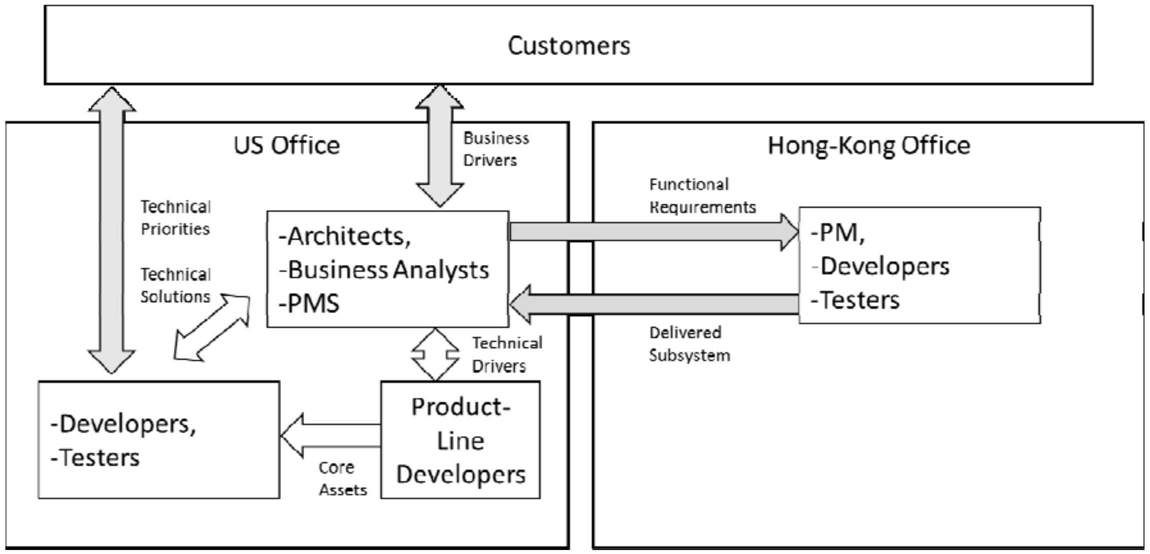
- US and Hong Kong teams to distribute work
- Work Breakdown: Functional decomposition to ease integration
- Home office coordinates customer interaction and partitioning of responsibility
- Hong-Kong team builds functional services



Implementation Issues and Challenges

Challenges & Issues

- Remote and local teams had significant trouble integrating subsystems together (many semantic interoperability issues)
- Interoperability issues created tension resulting in transactional meetings
- Initial conclusion: Agile methods don't scale



Identifying the Issues

Meeting of the Minds and Findings

- The minimum lead time to answer developer questions from Hong-Kong office was 1.5 days (due to 13 hour time difference), questions raised past mid-week resulted in further delays
- Semantic interoperability was a major area of difficulty: Hong Kong team had difficultly understanding the end-use of the functionality they implemented; US team had difficulty maintaining continuous integration environment
- Dividing responsibility functionally created a hierarchy that restricted information flow between the customer and the developer. The Hong-Kong team relied on second-hand information to complete tasks
- Communication mechanisms were highly transactional (overly status oriented); The conveyance of information between the customer and the Hong-Kong development team was inefficient at best and ineffective at worst.
- Global team could not collaborate daily, time differences proved exceedingly difficult

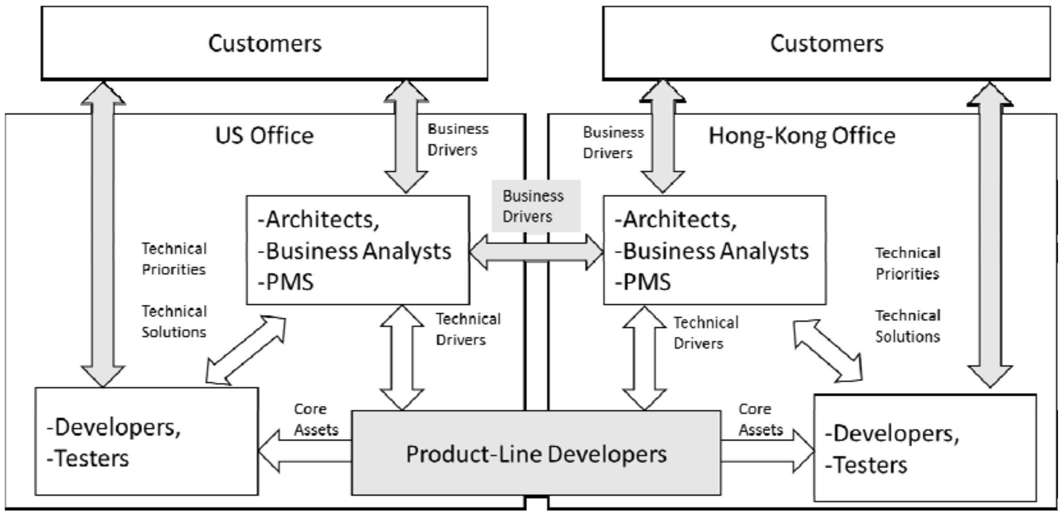
Analyzing the Issues

Together, the US and Hong Kong teams re-examined their use of agile methods in light of the agile principles and determined:

- Delegation of responsibility cannot be done for expediency sake (by the US team): this inhibited frequent delivery, collaboration, and undermined trust
- The Hong-Kong team was not an equal player: this undermined the team's ability to self-organize, maintain a constant pace, respond to change, and deliver working software
- Organizational constructs were designed to address US markets, not international markets

Addressing the Challenges

- Company X reorganized its development organization to:
  - Create teams in each target locale, with a single product-line development team as an anchor across teams
  - Responsibility was divided by locale – allowing developers, architects, and business analysts to interface with customers in each locale



Agenda

- Facing Agile Software Development Failure: A Case Study
- ➢ Lessons on Scaling Agile Software Development
- Adopting Agile: Recommendations for Program Offices

AEROSPACE

Scaling Agile: Lessons Learned

- Periodically evaluating organizational structures, project constraints, engineering practices, and project outcomes in light of the agile principles and values to identify impediments is value-added
- Expanding development efforts across locales may require more than process adjustments to ensure success
- Scaling agile methods requires willingness to change how teams structure and organize the work as project size, complexity, and context vary
- Signs that agile methods may require a closer look:
  - Transactional Meetings
  - Functional or organizational hierarchies where single points of communication breakdown can occur
  - Semantic breakdowns and finger pointing
  - Many “internal” deliveries (not validated by customers)
  - Significant number of defect escapes

Agenda

- Facing Agile Software Development Failure: A Case Study
- Lessons on Scaling Agile Software Development
- ➢ Adopting Agile: Recommendations for Program Offices

Adopting Agile: Recommendations for Program Offices

Many contractors are ready to employ agile methods. It is recommended that program offices cultivate relationships with contractors to better understand the purpose and scope of its use.

To this end, it is also recommend that program offices start by:

- Seek out foundational knowledge on proposed methods from independent sources when presented with proposals by contractors
- Although counter to the values and principles of the Agile Manifesto, ensure that contracts establish integrated roles and access to practice data by government representatives.
- Establish guidance to contractors for managing the incremental deliveries consistent with program lifecycle needs and ensure that software development plans are consistent with this guidance
- Collect experience data and conduct periodic independent quality checks
- Hardware and System of Systems Dependencies: Hardware and Systems may need to be decoupled in order to establish and maintain short-delivery cycles
- Monitor and Adjust your Approach: Singular/Immutable approaches to managing decentralized software development methods may lead to undesirable outcomes. Make allowances for change that both enhances program office insight and promotes collaboration with contractor

Publications

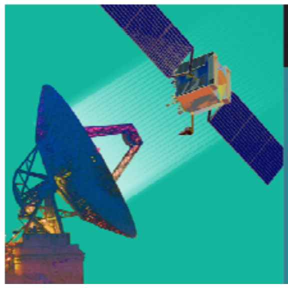
- Houston, Dan X., and Steven W. Rosemergy. Assessing Product Development Agility. In *Managing Software Process Evolution: How to handle process change?* Marco Kuhrmann, et al. editors. To be published by Springer early 2016.

Discussions

Schedule

Time	Presentation and Discussion
1:00 – 1:30pm	Session Overview
1:30 – 2:15pm	“Embracing Change to Overcome Challenges with Large-Scale Agile Software Development”, Steve Rosemergy, The Aerospace Corporation
2:15 – 3:00pm	General discussions - Pains, struggles, and barriers in adopting agile ground software development - Failed Agile attempts - Going forward, how to increase agility?
3:00 – 3:30pm	Break
3:30 – 4:15pm	“Agile Fit Check”, Sue Mobasser, The Aerospace Corporation
4:15 – 5:00pm	Discussion - Failed Agile attempts - Going forward, how to increase agility? - Agile Guidance

Ground System Architectures Workshop



Agile Fit Check Tool

Supannika Mobasser, The Aerospace Corporation

Supannika.k.mobasser@aero.org

AEROSPACE

Outline

- Introduction and Literature review
  - –Balancing Agility and Discipline Model
- Agile Fit Check
  - –Agile Fit Check Criteria
  - –Agile Fit Check Tool
  - –Agile Fit Check Case Studies
- Discussion

Motivation

- How do we know that this project / this proposal would be a good fit for an agile development?
- What should be the evaluation criteria?

Background models

Current models identifying fitness for agile software development

- Balancing Agility and Discipline [Boehm and Turner, 2004]
  - Use five evaluation criteria and process-related risks to identify the appropriate process model
- MITRE Defense Agile Acquisition Guide [MITRE 2014]
  - Provide sixteen assessment criteria to distinguish between traditional and agile practices

Balancing Agility and Discipline Model

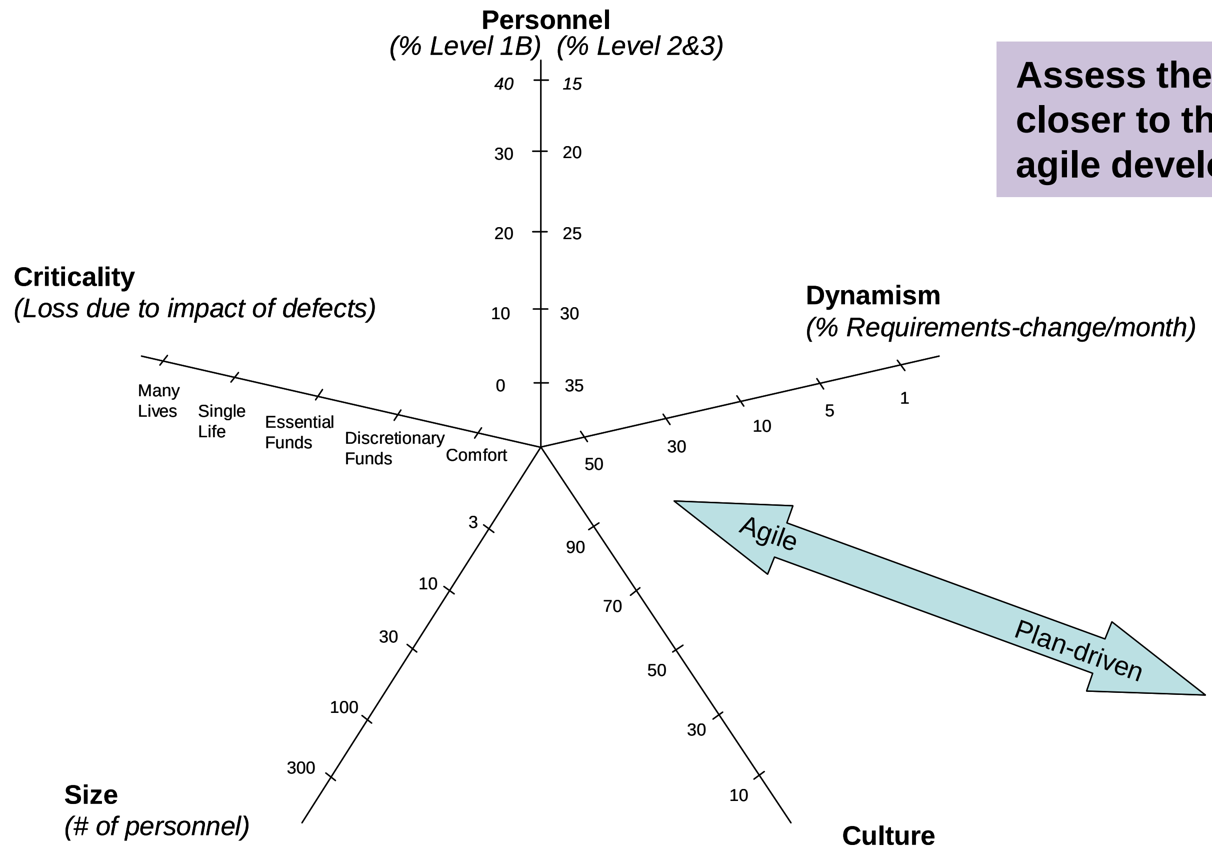
Based on “Balancing Agility and Discipline” book [Boehm and Turner 2004]

- A good starting model to check whether a project should follow an agile or plan-driven development
  - Assess the project characteristics with a spider diagram
  - Assess agile-driven risks and plan-driven risks
    - To tailor the process to mitigate the risks
- However, this model can not be used as is
  - More appropriate for an in-house development
  - Missing some system-level / government-specific evaluation criteria

Balancing Agility and Discipline Model

Five Critical Decision Factors

- Size, Criticality, Dynamism, Personnel, Culture



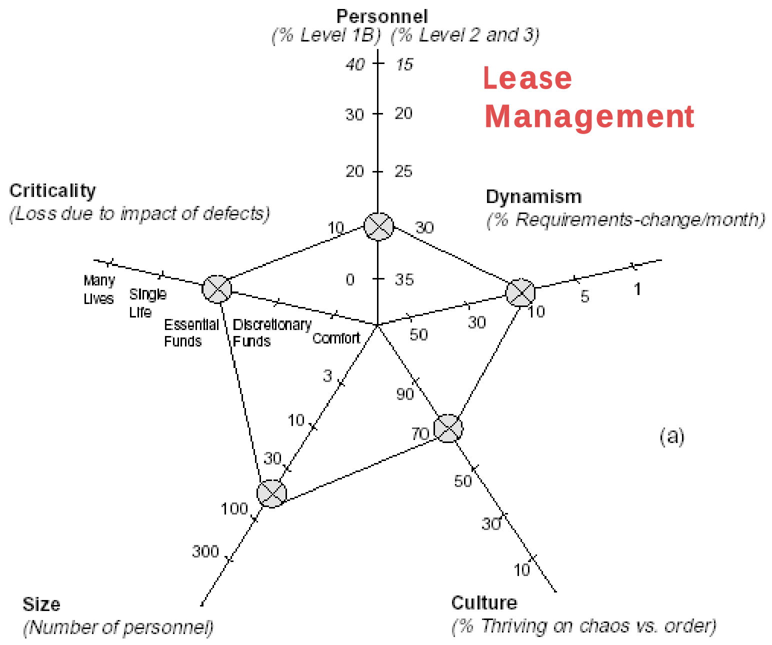
Assess the project characteristics, the closer to the center, the more suitable an agile development would be.

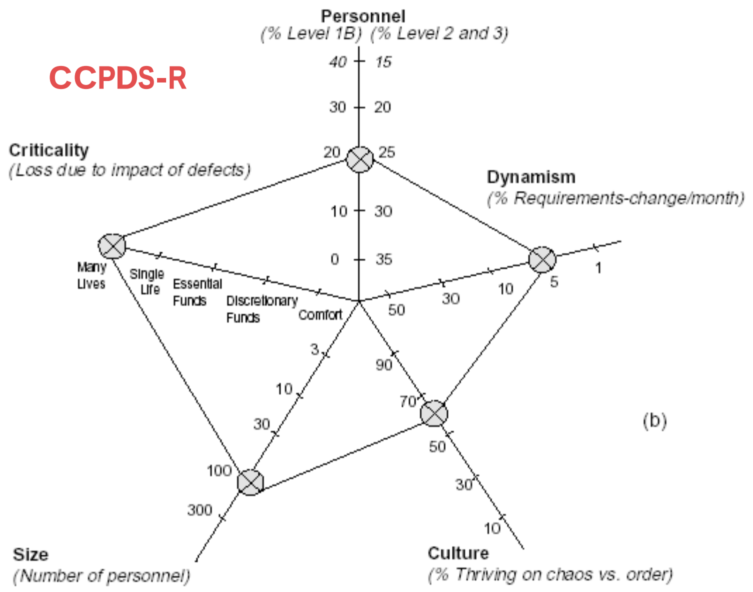
Ref: Balancing Agility and Discipline: A Guide for the Perplexed, Barry Boehm and Richard Turner, Addison Wesley, 2004

Balancing Agility and Discipline Model

Comparing the Case Study Projects

The closer to the center, the more suitable an agile development would be.





CCPDS-R: USAF Command Center Processing and Display System Replacement for early missile warning

Lease Management project fits for Agile process more than CCPDS-R

Ref: Balancing Agility and Discipline: A Guide for the Perplexed, Barry Boehm and Richard Turner, Addison Wesley, 2004

Outline

- Introduction and Literature review
  - Basic Agile Concepts
  - Balancing Agility and Discipline Model
- ➡ Agile Fit Check
  - Agile Fit Check Criteria
  - Agile Fit Check Tool
  - Agile Fit Check Case Studies
- Conclusion

AEROSPACE

Agile Fit Check Criteria

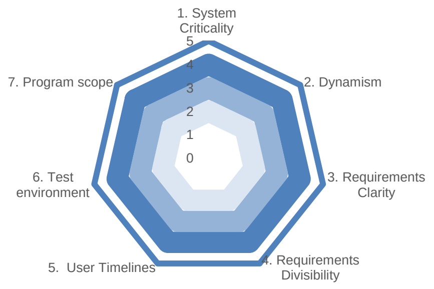
- A combination of Balancing Agility and Discipline model, MITRE's agile assessment list, and other sources
- To check for agile fitness, need to consider program's characteristics and commitments from both Government and Contractor(s)

1. **System's characteristics**
   - Q: Is the nature of the system applicable to agile development?
     – Project size, criticality, volatility, clarity
2. **Government's level of commitment**
   - Q: Is the government ready to support agile development?
     – Leadership support, contract type, stakeholders' representatives
3. **Contractor's level of commitment**
   - Q: Is the offeror or contractor ready to support agile development?
     – Collaboration, team organization

Agile Fit Check Criteria
1. System’s Characteristics

- Agile methods allow “fail fast, fail often”, respond very well to changes, and provides frequent feedback.
- The projects that benefit most from an agile development have high requirements or scope volatility, low requirement clarity, the ability to decompose requirements into small increments, the ability to accept frequent upgrades, and the ability to test throughout development cycles.
- Although it has been proven that Agile can be used in mission/life-critical projects, it is more effective to apply Agile to low criticality projects. It is also more efficient to use Agile on projects that are built on a mature infrastructure or architecture.

Agile Fit Check Criteria
1. System's Characteristics



Q: Are the system's characteristics compatible with agile development?

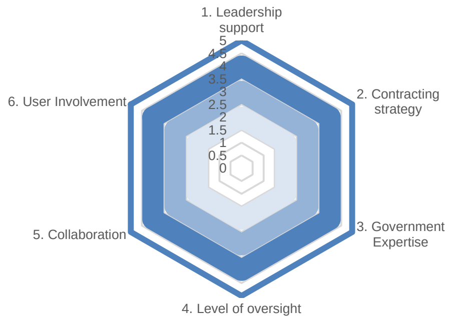
Criteria	Agile-driven	Plan-driven
1. System Criticality (loss due to impact of defect)	Loss of comfort	Life-critical issues
2. Requirements Volatility (Sprint-level requirements)	50 requirements change per month	1 requirement change per month
3. Requirements Clarity	Unclear; proof of concept; unprecedented	Well understood; constitutional
4. Requirements Divisibility	Decomposable into small tasks to fit with short iterations	Tightly integrated; tightly coupled; difficult to decompose
5. User Timelines	OK with iterative development or frequent upgrades (1 year)	Operational environment does not allow iterative development
6. Test environment	OK with testing throughout development, automated testing	Unable to do parallel development testing; no resources, tools, or not operable
7. Program scope	Limited to the application layer with existing / mature infrastructure	Program spans core capabilities and underlying platform or infrastructure

AEROSPACE

Agile Fit Check Criteria
2. Government's Level of Commitment

- Agile developments require constant customer involvement and preferably co-located collaboration from all of its success-critical stakeholders.
- It is crucial for agile projects to have leadership support, a flexible contracting strategy, frequent and effective collaboration, and good oversight tools.
- An effective agile development also requires availability of target users for quick feedback.
- Most importantly, the government team needs to have sufficient knowledge of the agile process in order to set the right expectations and contribute effectively to the agile project.

Agile Fit Check Criteria
2. Government's Level of Commitment



Q: Is the government ready to support agile development?

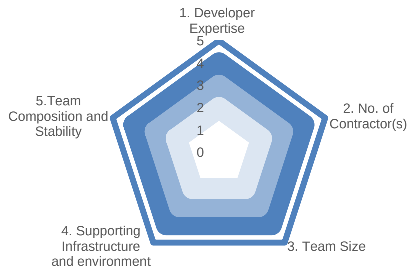
Criteria	Agile-driven	Plan-driven
1. Leadership support	Leadership supports non-traditional processes and methods	Leadership prefer a traditional development approach
2. Contracting strategy	Contract strategy support agile timelines and approach / process (steps, milestones, sequence)	Contract strategy does not support agile timelines
3. Government Expertise	Government has knowledge about expectations in agile development	Government is not ready / no knowledge about agile development
4. Level of oversight	Program office has authority for program decision; has right tools and metrics	Require high level authority to make decision
5. Collaboration	Government and developers can collaborate frequently and effectively	Geographically dispersed; limited collaboration; no budget to travel
6. User Involvement	User representatives or end users available for frequent interaction	No target user rep or not available/ accessible

AEROSPACE

Agile Fit Check Criteria
3. Contractor’s Level of Commitment

- Agile developments require a dedicated and experienced development team.
- A good agile team requires an extremely high level of synchronization and quick turnaround process with a sustainable pace.
- Hence, an agile team should be a small team with an agile-ready mindset, knowledge, and skills.
- Due to the high degree of collaboration and frequent baselining, agile developments work best with one or a few contractors. It is also best to have a co-located and low to zero turn-over team with supporting collaboration and development infrastructure to support the one-voice, yet dynamic and rapid-fielding objective.

Agile Fit Check Criteria Contractor’s Level of Commitment



Q: Is the contractor ready for agile development?

Criteria	Agile-driven	Plan-driven
1. Developer Expertise (Familiarity to agile approach)	Agile-ready; trained and experienced scrum master and developer	Lack of agile development experience
2. No. of Contractor(s)	One or a few contractors	Many contractors
3. Team Size	Small team (3 people)	300 people
4. Supporting Infrastructure and Environment	Co-located team; good collaboration tools	Distributed teams among several time zones; lack of collaboration infrastructure
5. Team Composition and Sustainable Pace	All team members are stable and have worked on previous projects together. Work on one project at a time with 40-hour work week.	Personnel turn-over across entire team. Work on multiple-project at a time or unable to commit to the whole project life cycle.

AEROSPACE

Agile Fit Check Criteria Summary

Three major components to determine the fitness of a certain project

System's Characteristics	Government's Level Commitment	Contractor's Level Commitment
1. System Criticality (loss due to impact of defect)	1. Leadership Support	1. Developer Expertise (Familiarity to agile approach)
2. Dynamism (requirements change / month)	2. Contracting Strategy	2. No. of Contractor(s)
3. Requirements Clarity	3. Government Expertise	3. Team Size
4. Requirements Divisibility	4. Level of Oversight	4. Supporting Infrastructure and environment
5. User Timelines	5. Collaboration	5. Team Composition and Sustainable Pace
6. Test environment	6. User Involvement	
7. Program scope		

AEROSPACE

Agile Fit Check Tool

Background

- Agile Fit Check Tool version 1.0 is a Microsoft Excel-based tool that determines the agile readiness level using three Agile Fit Check criteria
- The Agile Fit Check tool incorporates several state-of-the-art research studies.
- It plots the results in a radar chart to show how good a fit an agile development method would be for a particular system.

- The tool was developed with the proposal phase in mind
  - But can be used at any time to help mitigate agile development risks

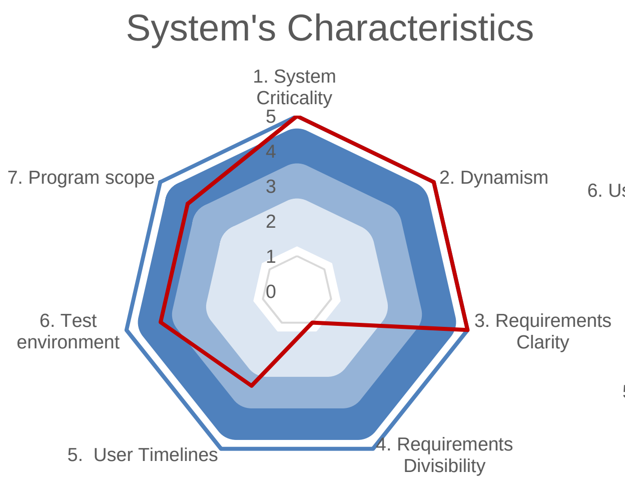
How to use Fit Check tool?

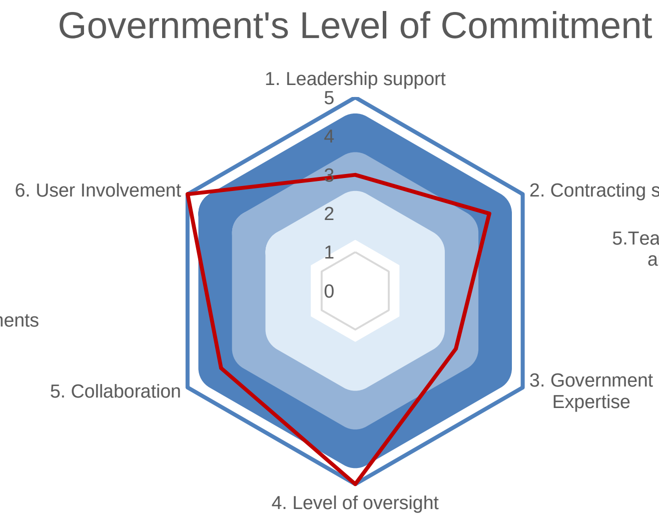
1. Identify the following profiles based on the given criteria
   - System's Characteristics
   - Government's Level of Commitment
   - Contractor's Level of Commitment
2. Click run to plot the radar graphs
3. Identify non-compliance criteria
   - The tool does not tell you whether you should or should not follow an Agile approach, but it will tell you the risks or challenges of following an Agile approach
     - E.g. Not having "user involvement" is a challenge for Agile
   - However :
     - Having stable requirements is necessary for a traditional process, but that does not mean that having unstable requirements is required for Agile
4. Discuss with your team on possible process tailoring option
   - Pick and choose appropriate practice(s) to fill the gap(s)

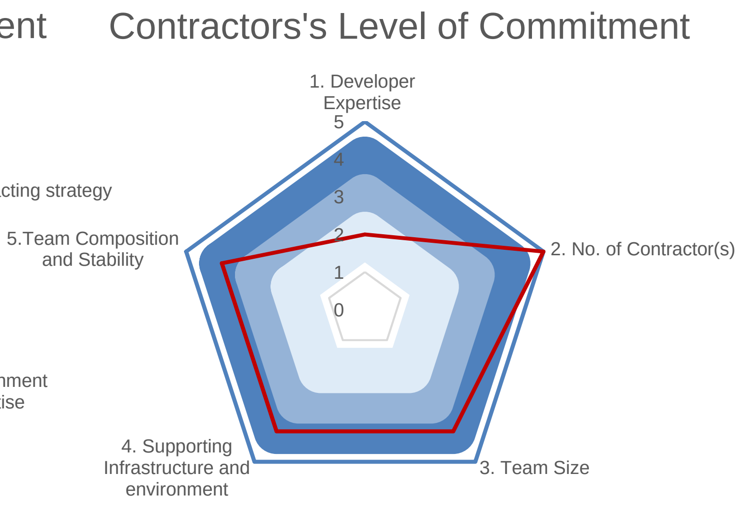
Ground System Architectures Workshop

Case Study 1

- Challenges in adopting agile in several aspects





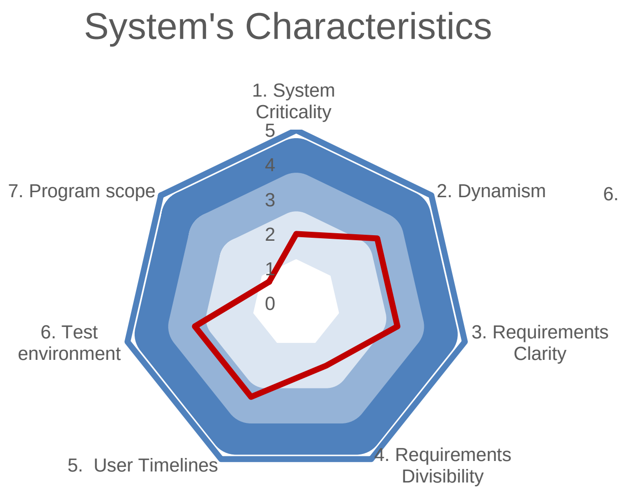


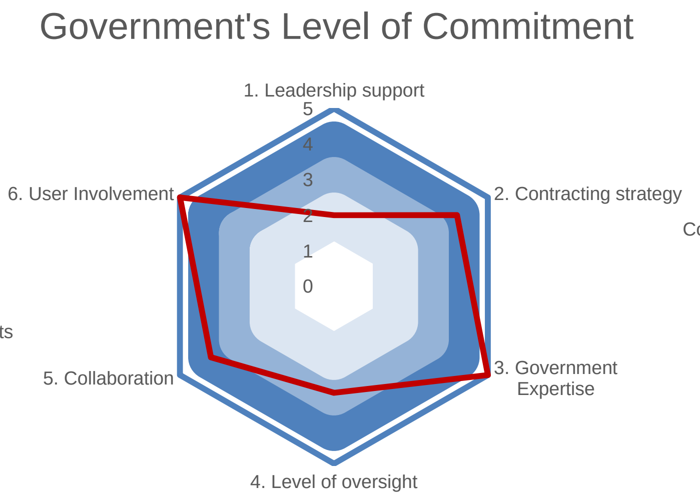
The closer to the center, the more suitable an agile development would be

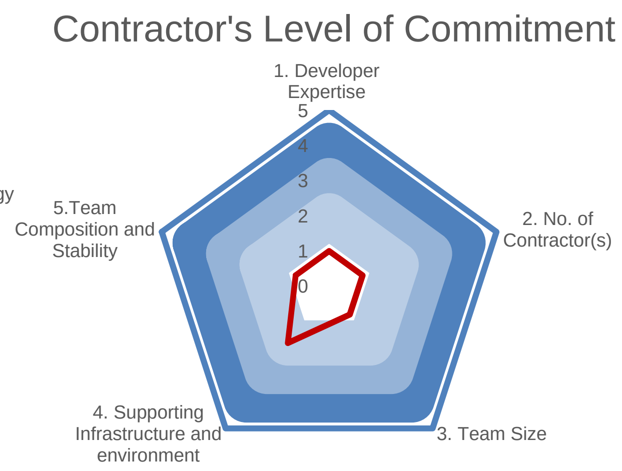
AEROSPACE

Case Study 2

- Challenges in agile adoption on the government side







The closer to the center, the more suitable an agile development would be

AEROSPACE

Discussions

Open discussions

- Time to share your experiences about agile adoption
- 3 topics
  - Pains, struggles, and barriers in adopting agile ground software development
  - Failed Agile attempts
  - Going forward, how to increase agility?

Pains, struggles, and barriers in adopting agile ground software development

- Disconnect between Government processes (acquisition, contracts, security A&A) vs. the tempo of Agile
- Agile doesn't scale well using current best practices, “A Bridge Too Far”
- The government business model doesn’t always align with the principles and values of the agile manifesto (predefined scope and delivery)
- Understanding how to measure “done” and the metrics to determine an equivalent to earned value
- High personnel turnover can impede effectiveness of agile teams

Failed Agile attempts

- Mix Agile development cycles with traditional review process
- Learn Agile on the fly
- Solely rely on individuals and tacit/tribal knowledge
- Lack of integration infrastructure to support agile teams
- Not enough system and software engineering to begin development
- Developing overlapping capability threads
- Avoid selection criteria such as lowest-cost technically acceptable
- Tacking on IA at the end
- User stories not ready when sprint planning begins
- Not having a definition of done

AEROSPACE

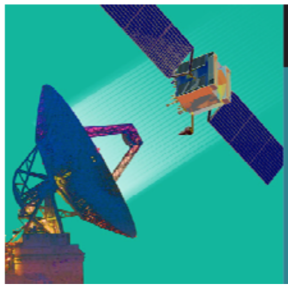
Going forward, how to increase agility?

- Have a proper agile training for both acquirer and developer
- Need a supportive collaboration infrastructure
- Plan for evolving requirements
- All about people: people have to work together and trust each other
- Teach this in ACQ 101
- Contractors tell the Government what you want
- Increase face time between Government and Contractor
- Government: Understand what questions to ask to facilitate shared understanding
- CDRLs: consider tailoring up instead of tailoring down
- Better understand what we want to measure and how to measure them

Agile Guidance

AEROSPACE

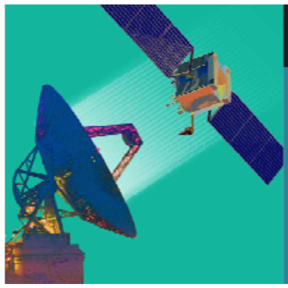
Ground System Architectures Workshop



OPEN DISCUSSION

AEROSPACE

Ground System Architectures Workshop



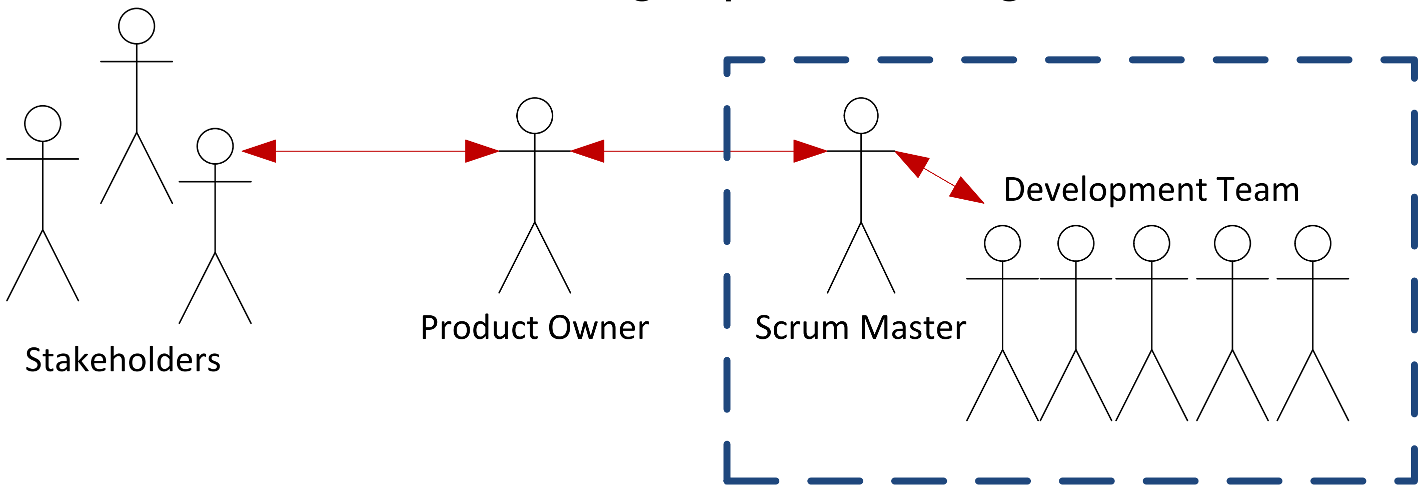
BACK UP

Agile principles / techniques that do not work for you. How did you tailor them?

- Face-to-face conversation
  - weekly tag up,

Scrum in a nutshell

- **3 roles, 3 ceremonies, 3 artifacts**
  - **Roles :** Scrum Master, Product Owner, Development Team
  - **Ceremonies:** Sprint Planning, Daily Scrum, Sprint Review & Retrospective
  - **Artifacts**: Product backlog, Sprint backlog, Burndown chart



Extreme Programming (XP) in a nutshell

XP 12 Practices

- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour Week
- On-site Customer
- Coding Standards

Scrum teams often pick and choose the preferred practices to apply

Raytheon

Customer Success Is Our Mission

Acquisition for Agile

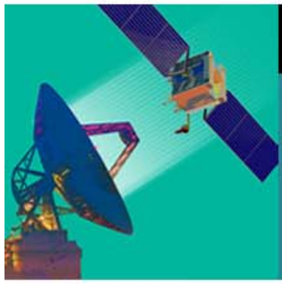
Larri Ann Rosser

Engineering Fellow

Feb 3 2016

DoD Goes Agile – What They Said

- ***SEC. 804. IMPLEMENTATION OF NEW ACQUISITION PROCESS FOR INFORMATION TECHNOLOGY SYSTEMS.***
- *(a) New Acquisition Process Required- The Secretary of Defense shall develop and implement a new acquisition process for information technology systems. The acquisition process developed and implemented pursuant to this subsection shall, to the extent determined appropriate by the Secretary–*
  - *(1) be based on the recommendations in chapter 6 of the March 2009 report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology; and*
  - *(2) be designed to include–*
    - *(A) early and continual involvement of the user;*
    - *(B) multiple, rapidly executed increments or releases of capability;*
    - *(C) early, successive prototyping to support an evolutionary approach; and*
    - *(D) a modular, open-systems approach.*



What They Did*

- Five year, single award IDIQ 🟦🟦🟦🟩
- SOW Vision to bound scope of the IDIQ 🟦🟦🟩
- Requirements expressed as features, open ended 🟦🟦▌
- Initial Task Order with fixed content to set velocity 🟦🟦🟦🟩
- Bid first TO in story points 🟦🟦
- Bid follow on work by capacity – x story points per sprint at a fixed cost per sprint 🟦🟦🟦🟩
- Mandated SAFe compliance 🟦🟦🟦

* Responses in which I participated

Challenges

- Bidding a new program in story points
- Establishing capacity before work begins
- Reconciling the desire for flexibility in function with sell-off requirements
- Architecture approach – early cost savings vs. long term agility
- Accounting for startup
- Customer aspirations vs. capabilities
- Price to Win
- Lack of alignment between what agencies want to do and what is mandated

Current Improvement Activities

- PARCA Agile EV work
- INCOSE ASSEWG Project: Agile Systems Engineering Life Cycle Model (ASELCM)
- INCOSE ASSEWG Project: Agile System Engineering and Acquisition Dynamics Model
- NDIA Agile and SE Working Group Paper In Progress: Acquisition for Agile

References

- Digital Services Playbook

https://playbook.cio.gov/#plays

- TechFAR Handbook

https://github.com/WhiteHouse/playbook/blob/gh-pages/_includes/techfar-online.md

- Scaled Agile Framework

http://www.scaledagileframework.com/

Discussions