Clearing the Agile Mist

Driving to Clear Communication as Agile Goes Program Wide

Michael Coughenour
Lockheed Martin IS&GS, System Engineering Technologist;
INCOSE Agile SE WG Co-Chair
mike.coughenour@lmco.com 618-910-0133

Jim Brake
Lockheed Martin IS&GS, Senior Manager
jim.brake@lmco.com 719-277-5438

Brad Newman
Lockheed Martin IS&GS, MBSE SME
bradford.j.newman@lmco.com 719-277-4118
Creating Solutions to Meet Customers’ Needs

No problem there – we know how to do it well
So Why Do We Need to Change?

- Rapid change – increasing Ops Tempo
- Increased focus on operational suitability
- Shrinking resources – increased financial pressure (Affordability)
- Increasing complexity
- Increasing unpredictability
- Increasing uncertainty
- Rate of Technology Change (Moore’s Law)

And,,,

- We want to realize the benefits of Agile Practices across the entire program
## Increasing Complexity

<table>
<thead>
<tr>
<th>Complex (Dynamic Complexity)</th>
<th>Complicated (Detailed Complexity)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Composable Design</strong></td>
<td><strong>Decompositional Design</strong></td>
</tr>
<tr>
<td><img src="chart.jpg" alt="chart" /></td>
<td><img src="chart.jpg" alt="chart" /></td>
</tr>
</tbody>
</table>

### Complex (Dynamic Complexity)
- Composable Design

### Complicated (Detailed Complexity)
- Decompositional Design
But how is value realized?— the ends of the spectrum

- Disciplined, rigorous engineering with predictable outcomes
  - Have to wait for the value
- Fly by the seat of the pants design-and-build
  - Value is delivered in quickly but with unpredictable results

How do we get the best of both worlds – Agile Systems Development

- Flexible application of dependable practices focused on value delivered in short increments

Agile = Bringing Value As Quick As Possible
Not to get derailed but …

- ‘Big A’ Agile – Repeatable application of agile principles and practices
- ‘Little a’ agile – being nimble

Why the distinction?
- People don’t know what you mean when you say “Agile”
- People don’t know what you hear when they say “Agile”
So What’s the Problem

Everyone is using the term but with a different meaning
- And often a different intent

Recognize any of these?
- “They said they were doing agile but it looks the same as always.”
- “They said they were doing agile but when I dug into it I did not recognize it as what I know of agile.”
- “That’s not what I thought agile was?”

Communication
- People who don’t really understand agile but want the benefits it promises
- Misunderstandings lead to failure of expectations

Terms pretty clear in SWD but Agile is now being applied in HW, at the system level, and across programs
Unfortunately

- People throw “agile” around like they really know what they’re talking about when they don’t
- People use the Buzz word because it gets attention
- People blowing smoke
- There is considerable motivation for hi-jacking the term
  - People making money
  - People with career ambitions
  - People who say/claim agile to get sales
  - People who speak agile to get buy-in for something else
AGILE SYSTEMS Engineering

VS.

Agile SYSTEMS ENGINEERING

Where we are
Management versus Technical Practices

At a high level it’s simple, really

- Management practices - how we manage the work at hand (e.g. project management - scrum)

- Technical practices – how engineering products get created (e.g. paired programming, rapid prototyping)

But if you don’t clarify what is meant when someone says Agile .... Two conversations ensue.

The one you are having
and
the one the person you are talking to is having
Agile Management Practices, Approaches, and Frameworks

- Short, Time-Boxed Iterations with Frequent Deliveries
- Continuous Planning
- Small, Self-Directed Teams
- Early and Frequent Stakeholder Involvement
- Daily Standup Meetings
- Frequent Feedback/Early Learning (e.g. retrospectives)
- Backlog Driven Management

- Examples of Popular Approaches
  - Scrum
  - Scaled Agile Framework (SAFe) (Leffingwell)
  - Agile Project Management (APM) Framework (Highsmith)
  - Disciplined Agile Delivery (DAD) (Ambler)
Agile Technical SE Practices - A Broad Stroke

- **‘Evolutionary’ / Incremental**
  - Early SE bounding – more SE with each ‘time box’
  - Often relies on fixed schedule and cost with flexible technical scope
  - Emergent design, emergent architecture
  - Balanced prescriptive and emergent

- **‘Expedited’ / Lean / Right-Sized**
  - SE fit to the program characterization & urgencies
  - Prescriptive architecture
  - Incremental Commit Model (ICSM) – Value focused scaling

- **Agile SWD with SE (Ad-Hoc SE)**
  - Organizations & projects driven by Agile SWE but recognizing need for SE
Systems Engineering examples (young area)
– From value focused Scaling
What do Agile Programs Look Like?

- Traditional SE Managed as Agile
- Traditional SE with Agile Development
- Traditionally managed Evolutionary Agile SE technical approaches (Plan driven)
  - Scaled SE with Traditional Development
  - Scaled SE as precursor (leading) to Agile Development
- “Agilely” Managed Evolutionary Agile technical approaches
  - Ad-hoc SE driven by Agile Development (essentially no prescriptive design)
  - Scaled SE as precursor (leading) to Agile Development (the whole enchilada)
- Expedited SE – Traditional or Agile Management
- Combinations of approaches on complex programs
- Examples of Scaled SE Technical Practices (to come)
  - Seminar series being developed

Balancing Prescriptive and Emergent
Agile Taken to The Enterprise Level

- Could be viewed from the team/development program scaling to big or complex levels

- At Scale – a hybrid approach of technical practices are applied together

- But what about technical and operational
  - Dev Ops
Scaled Solution Description (Architecture)

Back of the Envelope  Whiteboard  Diagrammed  Statically Modeled  Dynamically Modeled

Technical Maturity
Technical Rigor
Design Uncertainty
Technical Execution Risk
Problem Complexity
Understanding the difference is one thing
When someone says “agile” it can mean many things
Knowing that is half the battle
Figuring out what they mean by “agile” is the other half
The key is to know the right questions to ask
  – So here are some
How Can You Know? – Some Key Questions

- Is your effort organized for realizing value early and often?
- Big A vs. Little a
  - Is your effort a development program/project?
  - Is your program much more responsive to external change than normal?
  - Are your ‘agile’ methods documented, repeatable?
- Do you deliver (put into use) pieces of the system in short intervals?
- Do you create & review/deliver engineering products in increments?
- Are you managing the work with Scrum?
- Which formal agile SWD/SE methods are you using?
- Are you talking about the agility of the engineering or the agility of the system that comes from the engineering?
The term “Agile” can mean so many things
  – Management approach
  – Systems engineering approach
    ▪ Technical practices
  – Development approach

This mix of possible applications of Agile practices results in many variants of “agile” programs

This is one case where being informed is half the battle.
The other half is being able to know how to avoid confusion – to ask the right questions.
• Example Criteria
  – EA is concerned with the ecosystem – SA fits a solution into the ecosystem
  – SOS A focuses on collaboration of systems – SA focuses on creating a system/solution
  – SA determines the right mix of technologies for a solution – SWA creates the behaviors allocated to SW/services
  – SA handles analysis of alternatives
  – SA allocates aspects of the solution to TTPs, people, technologies (HW/SW)
  – SWA: performance, functionality …

Application to:
• Agile Sys Dev
• Agile SE
• Agile SW Dev

Other work is fleshing out the Continuum
Solution Descriptors (Requirements)

- Goals and Objectives
- Informal Descriptors
- Formal Descriptors (Shalls/User Stories)
- Engineered
- Modeled (Allocated)

Technical Rigor

Technical Execution Risk