Agile and Lean SE: Complimentary and Contradictory Relationship
09:30 – 10:30

Lean and Agile Systems Engineering
The Gordon Center For Systems Engineering
Technion, Haifa, Israel
06-Jan-2014

Rick Dove
Paradigm Shift International and Stevens Institute of Technology
Before Relating Agile to Lean...

First, we need to know what agile means ...and doesn’t mean
(probably not what you think)

- History of agile knowledge
- What makes an agile system agile?
  (agile SE must first be an agile system)
- What does agile SE look like
  ...and not look like
- So now – how does agile relate to lean?
Agile ≠ Software Methods

For many people agile is solely associated with the family of software development processes that uses that label.

An older, more general, more precise agile concept definition is the root of the software usage and employed in other domains.
Agility was identified in a 1991 study funded by US OSD* to determine the competitive operational strategy that would follow lean

Agility in domain independent systems was investigated by 250 organizations and 1000+ people in the 90’s, funded by DARPA

In 2001 the software development community adopted the word and general concept, with a variety of branded practices that bury the core concept

Fundamental understandings of agile are critical for effective SE employment to deliver agile benefits

OSD: Office of the Secretary of Defense
Working Definition of Agility

The ability to **thrive** in an environment of **continuous and unpredictable change**.

The focal point here is **change**
- the ability to initiate it, and the ability to respond to it.

**Thrive** is a key word,
- it implies long term success, as opposed to a lucky response, and it implies wielding agility as an offensive as well as a defensive capability.

**Continuous and unpredictable** underscores the new long-term picture and also distinguishes agility from mere flexibility, enabling successful change even when there is little advance notice and no prior expectation.

More Precisely…

Agility is the ability of a system to thrive in an uncertain and unpredictably evolving environment; deploying effective response to both opportunity and threat, within mission.

Effective response has four metrics:

- **timely** (fast enough to deliver value),
- **affordable** (cost can be repeated as often as necessary),
- **predictable** (can be counted on to meet the need), and
- **comprehensive** (anything/everything within mission).

In a Word

Agile is Responsive

Lean is Efficient

Useful labels that say what they mean

Agile : Lean = Forgiving : Mean
Need for Agility

The pace of technology is reducing the useful lifetime of deployed systems and increasing the risk in long development programs.

The pace of social collaboration on a global scale increases the pace of technological and social innovation.

Systems are becoming more complex, yet are needed in shorter time frames.
Quick History

1980s: Japanese lean manufacturing concepts demonstrate scary superiority

1990s: General world-wide scramble to catch up in Lean manufacturing
US OSD/DARPA asked what’s next for a head start?
Agile enterprise & systems concepts developed & socialized

Adoption

1993: Hewlett Packard was the first to initiate a program to educate its customers and bring to market IT support under the Agile Enterprise banner.

1996: DoD’s Command and Control Research Program began an exploration of agile command and control that continues today.

2001: the Agile Manifesto for Software Development adopted the agile label as appropriately descriptive and fundamentally consistent with their concepts.

Somewhere in there SOA (Service Oriented Architecture) and Web Services defined agile IT systems to enable agile enterprise strategies.

2010ish: Agile Software Development accepted as useful and mainstream.

2012ish: Systems Engineering community gets interested … and confused.
Agility is the gold standard for Information Age militaries. Facing uncertain futures and new sets of threats in a complex, dynamic, and challenging security environment, militaries around the world are transforming themselves, becoming more information-enabled and network-centric.

And quite a few more.

the Agile path started in DoD with this 1995 publication.
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

- amplifiers
- speakers
- signal tuners

Drag-and-Drop
Reuseable
Components

- playback units (tape, CD, DVD)
- video displays (TV, computer)
- content sources (TIVO, P2P)
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

- amplifiers
- speakers
- signal tuners
- playback units (tape, CD, DVD)
- video displays (TV, computer)
- content sources (TIVO, P2P)

Drag-and-Drop Reusable Components

Examples of Typical Reconfigurable/Scalable System Configurations

Audio tape

Video media

Net in/out
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play

Examples of Typical Reconfigurable/Scalable System Configurations

Plug-and-Play Evolving Passive Infrastructure Rules/Standards/Principles

Audio tape

Video media

Net in/out

Encapsulated Modules

amplifiers

speakers

signal tuners

playback units (tape, CD, DVD)

video displays (TV, computer)

content sources (TIVO,P2P)

Drag-and-Drop Reusable Components

‘40s/’50s

‘90s

‘00s
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

- amplifiers
- speakers
- signal tuners
- playback units (tape, CD, DVD)
- video displays (TV, computer)
- content sources (TIVO, P2P)

Drag-and-Drop
Reusable Components

Plug-and-Play Evolving
Active Infrastructure
Responsible-Parties

Examples of Typical
Reconfigurable/Scalable
System Configurations

Plug-and-Play Evolving
Passive Infrastructure
Rules/Standards/Principles

Assembly → User/Owner

Audio tape

Video media

Net in/out

Video/Surround

Digital/Internet

'40s/'50s

'90s

'00s
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

amplifiers  speakers  signal tuners
playback units (tape, CD, DVD)
video displays (TV, computer)
content sources (TIVO, P2P)

Drag-and-Drop
Reusable Components

Plug-and-Play Evolving
Active Infrastructure
Responsible-Parties

Examples of Typical
Reconfigurable/Scalable
System Configurations

Plug-and-Play Evolving
Passive Infrastructure
Rules/Standards/Principles

Assembly
User/Owner
Readiness
Stores

Audio tape

Video media

Net in/out

Video/Surround

Digital/Internet

'40s/'50s  '90s  '00s

Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

amplifiers  speakers  signal tuners
playback units (tape, CD, DVD)
video displays (TV, computer)
content sources (TIVO, P2P)

Drag-and-Drop
Reusable Components

Plug-and-Play Evolving
Active Infrastructure
Responsible-Parties

Examples of Typical
Reconfigurable/Scalable
System Configurations

Plug-and-Play Evolving
Passive Infrastructure
Rules/Standards/Principles

Assembly
User/Owner
Readiness
Stores

Audio tape

Video media

Net in/out

Video/Surround

Digital/Internet

'40s/'50s  '90s  '00s
Case: Home Entertainment Technology Migration
agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

- amplifiers
- speakers
- signal tuners
- playback units (tape, CD, DVD)
- video displays (TV, computer)
- content sources (TIVO, P2P)

Drag-and-Drop
Reusable Components

Plug-and-Play Evolving
Active Infrastructure
Responsible-Parties

Examples of Typical
Reconfigurable/Scalable
System Configurations

Plug-and-Play Evolving
Passive Infrastructure
Rules/Standards/Principles

Audio tape

Video media

Net in/out

Video/Surround

Digital/Internet

'40s/'50s

'90s

'00s
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

- amplifiers
- speakers
- signal tuners
- playback units (tape, CD, DVD)
- video displays (TV, computer)
- content sources (TIVO, P2P)

Drag-and-Drop Reusable Components

Plug-and-Play Evolving Active Infrastructure
Responsible-Parties

Examples of Typical Reconfigurable/Scalable System Configurations

Plug-and-Play Evolving Passive Infrastructure
Rules/Standards/Principles

Audio tape
Video media
Net in/out
Video/Surround
Digital/Internet

'M40s/'50s

'M90s

'00s
Fundamental Concept
Reusable modules Reconfigurable in a Scalable architecture (RRS)
agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

Drag-and-Drop Reusable Components

Plug-and-Play Evolving Active Infrastructure Responsible-Parties

Examples of Typical Reconfigurable/Scalable System Configurations

Plug-and-Play Evolving Passive Infrastructure Rules/Standards/Principles

Variety/Time/Maturity/Range/Increments/Migrations/Evolutions/etc
System X-Ray Vision
(revealing the bones)

What i’m about to tell you is gonna change your life forever. Are you really sure you want to know it?

http://awespendo.us/animemangacomics/kermit-at-the-doctor/
Here's a Box of Bones

ERECTOR=MECCANO
Here is a System-Construction-Kit System
this agile architecture pattern provides adaptable structure (Agile 101)

**Integrity Management**
- Module mix evolution
- Module readiness
- System assembly
- Infrastructure evolution

**Infrastructure**
- Active
- Passive

**Rules/Standards**
- Interconnect Standards
- Safety Standards
- Product ConOps
- User ConOps

**Modules/Components**
- Gears/Pulleys
- Motors
- Wheels
- Tools
- Joiners, Axels, Small Parts
- Structural Material

**System Assembly**
- Product System Eng.
- Retail Distributors
- Owner/Builder
- Product Manager

**Product ConOps**
- User ConOps

**Safety Standards**
- Safety Standards

**Structural Material**
- Structural Material

**Rules/Standards**
- Radio Control Standards

**Active**
- Plane
- Helicopter
- Mobile Radar

**Passive**
- Gears/Pulleys
- Motors
- Wheels
- Tools
- Joiners, Axels, Small Parts
- Structural Material
Developing the System-Construction-Kit System Architecture

...how do we answer the questions? (Agile 102)

Modules/Components

- Integrity Management
  - Module mix evolution
  - Module readiness
  - System assembly
  - Infrastructure evolution

- Infrastructure
  - Active
  - Passive

- Rules/Standards
  - Sockets?
  - Signals?
  - Security?
  - Safety?
  - Service?

- Variety/Time/Maturity/Range/Increments/Migrations/Evolutions/etc
Sorting Out the Architectures

Ferris wheel has a functional architecture.

Erector/Meccano has an adaptable architecture.

The adaptable architecture enables the building and changing of the functional architecture.

One could argue that the adaptable architecture is also a functional architecture.

(but why bother?)
### RRS Principles – two are necessary
### the other eight are amplifiers

**Encapsulated Modules**
- 1:1 physical/functional packaging
- Black box to other modules
- Functional methods can change, but interface protocols cannot

**Evolving Minimal Standards (Infrastructure)**
- Defines module-interface protocols/standards (and operating rules)
- Enables and constrains agility
- Delicate balance of requisite variety and parsimony

<table>
<thead>
<tr>
<th>Encapsulated Modules</th>
<th>Evolving Minimal Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plug Compatibility</td>
<td>Redundancy and Diversity</td>
</tr>
<tr>
<td>(Facilitated Interfacing)</td>
<td></td>
</tr>
<tr>
<td>Reusable</td>
<td>Scalable</td>
</tr>
<tr>
<td>Facilitated Reuse</td>
<td>Elastic Capacity</td>
</tr>
<tr>
<td>Reconfigurable</td>
<td></td>
</tr>
<tr>
<td>Flat Interaction</td>
<td>Distributed Control and Information</td>
</tr>
<tr>
<td>Deferred Commitment</td>
<td>Self-Organization</td>
</tr>
</tbody>
</table>
Design the Architecture of Your Construction Set

Construction (response) architecture different from system functional architecture. Response architecture is a domain-focused engineering architecture.
The UURV Environment Drives the Need

Agile systems are defined in counterpoint to their operating environments.

To design and develop a system that can deal effectively with changing environments it is useful to articulate the nature of changes that should be considered.

Agile systems have effective situational response options, within mission, under:

- **Unpredictability**: randomness among unknowable possibilities.
- **Uncertainty**: randomness among known possibilities with unknowable probabilities.
- **Risk**: randomness among known possibilities with knowable probabilities.
- **Variation**: randomness among knowable variables and knowable variance ranges.
## Response Situation Analysis (RSA)

<table>
<thead>
<tr>
<th>Change Domains</th>
<th>General Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proactive</strong></td>
<td></td>
</tr>
<tr>
<td>Creation (and Elimination)</td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td>Proactive</td>
</tr>
<tr>
<td>Migration</td>
<td>Innovative/Composable</td>
</tr>
<tr>
<td>Modification (of Capability)</td>
<td>Takes Preemptive Initiative</td>
</tr>
<tr>
<td><strong>Reactive</strong></td>
<td></td>
</tr>
<tr>
<td>Correction</td>
<td>Reactive Proficiency</td>
</tr>
<tr>
<td>Variation</td>
<td></td>
</tr>
<tr>
<td>Expansion (of Capacity)</td>
<td>Reactive Proficiency</td>
</tr>
<tr>
<td>Reconfiguration</td>
<td></td>
</tr>
</tbody>
</table>

- **Innovative (Composable)**
- **Agile**
- **Fragile**
- **Resilient**

**General Characteristics**

- **Proactive**
  - Takes Preemptive Initiative
  - Creates Opportunity

- **Reactive**
  - Seizes Opportunity
  - Copes with Adverse Events
# Pro Forma RSA for Agile SE Project Management

Response Metrics: **t=time, s=scope, p=predictability**

<table>
<thead>
<tr>
<th>Change Domain</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Creation (and Elimination)</strong></td>
<td>- project management strategy (t); project team (t, c); system requirements (t, p); system architecture (t, s); system design (t, c, p); development activity plans (t); V&amp;V/test plans (t); team collective understanding and learning (t, p); product development [software code, hardware build documentation] (t, c, p).</td>
</tr>
<tr>
<td><strong>Improvement</strong></td>
<td>- activity effort estimating (p); activity completion to plan (t, c, p); reducing uncertainty and risk (t, p, s).</td>
</tr>
<tr>
<td><strong>Migration</strong></td>
<td>- compelling new technology availability (t, c, s); project scope change (s); lean process principles (p).</td>
</tr>
<tr>
<td><strong>Modification (of Capability)</strong></td>
<td>- new added team member unfamiliar/uncomfortable with management strategy (t); new environmental dynamics (t, c, p).</td>
</tr>
<tr>
<td><strong>Correction</strong></td>
<td>- wrong requirement (t); inadequate developer (t); failed V&amp;V/test (t, c); non-compliant supplier (t, c).</td>
</tr>
<tr>
<td><strong>Variation</strong></td>
<td>- expertise and skill levels among team members (p); grace period on schedule (t, c); deliverable performance range (p); availability, interaction, and expertise of customer involvement (s).</td>
</tr>
<tr>
<td><strong>Expansion (of Capacity)</strong></td>
<td>- 2x project scope change (t, c, p, s); team-size changes of x-y engineers distributed across n-m locations (t, s).</td>
</tr>
<tr>
<td><strong>Reconfiguration</strong></td>
<td>- unanticipated expertise requirement (t); development activity-sequence priority change (t).</td>
</tr>
</tbody>
</table>
Enterprise IT-Infrastructure Design

- MyFab = Web-accessible operations transparency
- MyProjects = Web-accessible strategic-project portfolio manager
- MyFab = Web-accessible operations transparency
- Fab = Foundry Plant
- A&T = Assembly & Test Plant

- = Bus Interface Module (BIM)
- = Extract/Transfer/Load (ETL) Interface Modules

www.parshift.com/Files/PsiDocs/Rkd050324CserPaper.pdf
Encapsulated Development Process

- Designed to Accommodate Requirements Evolution -

www.parshift.com/Files/PsiDocs/Rkd050324CserPaper.pdf
Agile Process and Agile Product

![Diagram showing the relationship between Agile Process and Agile Product](image-url)
Agile SE Life Cycle Framework
“Scrum’s roles, artifacts, events, and rules are immutable, and although implementing only parts of Scrum is possible, the result is not Scrum.

Scrum exists only in its entirety, and functions well as a container for other techniques, methodologies, and practices.” (Schwaber and Sutherland 2013)
Scrum has an Agile Architecture Pattern (AAP) **Structure** suitable for agile SW development, but not for agile systems-engineering …

… because the RSA is different for an agile systems-engineering process, and the Scrum AAP **strategy** is inadequate for systems engineering.
Everything falls into 3 categories:
Core – Barnacles – Rumor/Hearsay

There’s rules of the game and there’s strategies for playing the game.

Burndown charts, for instance, might be a strategy, but are not rules, not core.

Planning poker is not part of scrum at any level – it’s a cute idea. It’s got zip-zero-nothing to do with Scrum.

Scrum Master is chief impediment remover. That’s it.

Scrum helps you identify interesting questions but gives you no answers.

Video and text at: www.frequency.com/video/alistair-cockburn-core-scrum/129758869/-/5-7017272
Agile Software Development 2.0?
Lean and Agile Confusion and Complementarity

We see the phrase "Lean and Agile" appearing together as what might be Agile Software 2.0 complimenting emergence.

Joshua Kerievsky has a thought-starting presentation on "Lean Startup" at www.infoq.com/presentations/Lean-Startup.

Kerievsky's presentation reveals the emergence of an Agile 2.0 understanding, but his choice of rhetoric polarizes this emergence as a lean dominated lead.

Fundamentally he is showing how a more mature understanding of agile software development is removing the wasted resource commitments in the first generation of agile software concepts - a timey rethinking and a good start.

But...The roots of agile and lean should be kept in mind.
Both Seek Customer Satisfaction

Each adopts and adapts practices that help achieve their definition of satisfaction. They are appropriate philosophies in relation to a context.

Lean: deals effectively with cost and quality improvement in a stable environment.

Agile: deals effectively with uncertainty and opportunity in a changing environment.

Common Words: Best Practices

- Treat people with respect
- Overtime is wasteful
- Defer commitment
- Hire good people
- Continuous improvement

Met the plan

Met the need

Darwinian vertical evolution. Refinement. Robust but fragile.


(et. al)
Lean & Agile: Orthogonal Focus

Agility deals with “design-for-transformation”.

In a very general interpretation,

Lean values efficiency of operation and achieves this mainly through operational principles;

Agile values effective response ability and achieves this mainly through architectural principles.

Both are concerned with operational effectiveness. But the two have a different means for achieving different ends.
lean compromises its principles to incorporate agile principles

Agile can incorporate lean without compromise

agile improves with “selected” incorporation of lean principles
References and Supportive Readings


Epilog – System Security?
The Adversary is Very Agile

Architecture:  
- Multi-agent
- Loosely coupled
- Composable
- Self organizing
- Systems of systems

Behavior:  
- Swarm intelligence
- Tight learning loops
- Fast evolution
- Innovative
- Disposable resources

Attacking a Lean security strategy
Planned vs Agile Security

Automated defenses can work against automated attacks. But the sophisticated attack is a manual operation, requiring mano-a-mano defense. An 80-20 distribution by Lockheed’s thoughts, and the 20% are focused on high value results that justify high talent, large investment, and plenty of time.

“They’re going to keep innovating, I need to out innovate them.”

…Eric M. Hutchins, Lockheed
Lead author: Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaign and Intrusion Kill Chains
Agile system security, as a minimum, must mirror the agile characteristics exhibited by the system attack community:

[S] Self-organizing – with humans embedded in the loop, or with systemic mechanisms.

[A] Adapting to unpredictable situations – with reconfigurable, readily employed resources.

[R] Reactively resilient – able to continue, perhaps with reduced functionality, while recovering.


[P] Proactively innovative – acting preemptively, perhaps unpredictably, to gain advantage.

[H] Harmonious with system purpose – aiding rather than degrading system and user productivity.
Start at the Concept of Operations

Continuous evolution of system security is necessary to maintain parity with a continuously evolving threat environment: effective response under unpredictable and uncertain circumstances, as often as necessary. The system ConOps should call out the ability to reconfigure and augment system security throughout the development and operational lifecycle of the system, and it should call out the need for rapid reconfiguration of security at the system level.

An agile-system concept of operations recognizes the need for effective asynchronous system-security change.

Effective response has four metrics: timely (fast enough to deliver value), affordable (at a cost that can be repeated as often as necessary), predictable (can be counted on to meet the need), and comprehensive (anything and everything within the mission boundary).

Value Proposition—Risk management in an evolving unpredictable environment is the value proposition for agile systems. An agile system is constructed to enable and facilitate augmentation, reconfiguration and scalability of reusable assets in response to unpredictable situations, and agility is sustained with active management of responsibilities that constantly evolve the agility-enabling capabilities.
Common security passive infrastructure enables rapid evolution, augmentation, reconfiguration. Other security modules included as desired, e.g., functional-module behavior monitoring, to name one only.
Agile Architecture for Security Reconfiguration/Augmentation/Evolution

Security does not come in a separate box, e.g., all personnel are security trained, equipment is self-secure.

Four active responsibilities, with embedded security personnel as integrated team members.

System assembly enabled and constrained by configuration standards, informed by security policy.

Security policy informs all other passive infrastructure standards, and evolves simultaneously with each.

But You Already Have Lean Security. Why Bother?
Panel: Agile and Lean SE
Are these two complementary or are they different and should be applied in specific environment
15:00 – 15:30

Lean and Agile Systems Engineering
The Gordon Center For Systems Engineering
Haifa, 06-Jan-2014

Position Statement
Rick Dove, Stevens Institute of Technology
Lean will straighten it out
Agile will roll with it.
Lean & Agile: Orthogonal Concerns

Lean: Process Operation

Agile: Process Transformation
Brainwash & Blame Loss

Acquisition and contracting want to believe there are stable and predictable project execution paths. No surprises.
If things go wrong it is clear who’s to blame.
If you want a cheap thinking result, pick either one and turn the crank.
  • I did what they said to do (blame loss).
  • I kept it simple and followed the rules (brainwash).

Lean was developed for stable environments:
  manufacturing in a closed system, stable success metrics

Agile was developed for unstable environments:
  enterprise in an open system, dynamic success metrics

Manufacturing knows what has to be done for every product. Systems engineering has to figure that out differently for every product.
  • Can you standardize the SE process life cycle to eradicate surprises?
  • Can you hire laborers or use robots to operate the process?

When you have a stable or repetitive Systems Engineering environment, Lean is a good guiding approach to take.
These two approaches, at core, are in conflict.

Q: Are these two approaches complementary or are they different and should be applied in specific environment?

Only two choices? Complimentary or exclusive?

Dave Snowden has a handle on it. Every project has its own operational personality at every point in time, and that personality is unlikely to be static. You have to be vigilantly aware, understand what is happening at every point in time, and re-fit your approach as the situation changes.

Project management cannot win with lean thinking.

But within agile project management, use lean as a tool of preference when applicable (things are under control & predictable).