Agile Systems Architecting
~
Erik Philippus

looking at systems engineering through the agile lens

Agenda

• Introduction
  Double-sided view on Agile System Engineering,
  Agility in System Engineering, Engineering Agile Systems
  Process versus Product optimisation.

• Agile SYSTEMS ENGINEERING
  The Agile Manifesto, Agile Principles & Attributes,
  Scrum Basics, Multidisciplinary Scrum,
  Dealing with Uncertainty: Delayed Decisions
  Piecemeal Engineering: Staged Development,
  Set-Based Design: Narrowing the Design Space,
  7 Rules of Thumb for Agility in System Engineering.

• AGILE SYSTEMS Engineering
  The Price of Agility, When do we need Agility in Systems?
  Agile System Attributes, Response Ability,
  10 Design Principles for Agile Systems
Agenda

- **Introduction**

- **Agile SYSTEMS ENGINEERING**

- **AGILE SYSTEMS Engineering**
  The Price of Agility, When do we need Agility in Systems? Agile System Attributes, Response Ability, 10 Design Principles for Agile Systems

---

Agile Systems Engineering: A Double-sided View

**Agility in the system engineering process:**

Delay the freeze point as long as possible as new information becomes available during product development.

**Agility in the resulting system itself:**

Realization of systems that can respond rapidly to changed requirements after initial fielding of the system.
Engineering Agile Systems

• The ability to predict the future demand or requirements of a system is often severely compromised.

  Systems must be both flexible and must have the ability to change from one state or operating condition to another rapidly, without large switching costs or increases in system complexity.

  Agile systems are flexible, reconfigurable, extensible

  1. scalable in the sense of capacity (output/unit time) manufacturing systems that can change the capacity rapidly to adapt to actual market demand

  2. flexible in terms of functions and performance levels modification after initial deployment by addition of modules or modification of performance levels

Agility in Systems Engineering

• Increasing speed at which new products and systems are designed and introduced into the market place

• Uncertainty in future user needs and operating conditions and ambiguity in the "true" requirements

  focus on flexibility and speed in the upstream process of conceiving, designing and implementing systems

  1. adaptive and response to new (unexpected) information that becomes available during system development;

  2. opposite the traditional belief in system engineering that requirements and design solutions should be frozen as early as possible.
Agility in Systems Engineering

Dealing with uncertainties during development and manufacturing

- ambiguities in customer requirements,
- the viability of new technologies,
- appropriateness of one manufacturing process over another,
- ...

usually there is the expectation that these uncertainties can be resolved before the system is shipped

focus on process innovation rather than product innovation

Agenda

- Introduction
  Double-sided view on Agile System Engineering, Agility in System Engineering, Engineering Agile Systems
  Process versus Product optimisation.

- Agile SYSTEMS ENGINEERING
  The Agile Manifesto, Agile Principles & Attributes, Scrum Basics, Multidisciplinary Scrum,
  Dealing with Uncertainty: Delayed Decisions
  Piecemeal Engineering: Staged Development,
  Set-Based Design: Narrowing the Design Space,
  7 Rules of Thumb for Agility in System Engineering.

- AGILE SYSTEMS Engineering
  The Price of Agility, When do we need Agility in Systems?
  Agile System Attributes, Response Ability,
  10 Design Principles for Agile Systems
Agile Manifesto

Utah, February 2001

The Agile Principle
Agile Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning attitude</td>
<td>• Take advantage of lessons learned and adapt both processes and systems to meet customer needs.</td>
</tr>
<tr>
<td>Focus on value to customer</td>
<td>• Customer prioritizes requirements and progress is measured by operational features.</td>
</tr>
<tr>
<td>Short iterations delivering value</td>
<td>• Goal of each release is a working system.</td>
</tr>
<tr>
<td></td>
<td>• Rolling planning horizon.</td>
</tr>
<tr>
<td></td>
<td>• Risk-driven, reality-based iteration planning.</td>
</tr>
<tr>
<td>Neatness to change (design</td>
<td>• Change is seen as inevitable; ergo embrace change applies.</td>
</tr>
<tr>
<td>processes and system for change)</td>
<td></td>
</tr>
<tr>
<td>Continuous integration</td>
<td>• Integration is an ongoing activity.</td>
</tr>
<tr>
<td></td>
<td>• Integration and testing are as automated as possible.</td>
</tr>
<tr>
<td>Test-driven (decomposable</td>
<td>• Tests are written before any other artifacts (design, code).</td>
</tr>
<tr>
<td>progress)</td>
<td>• Capabilities (requirements) are defined by the tests (empirical evidence) that validate them.</td>
</tr>
<tr>
<td>Lean attitude (remove non-</td>
<td>• An idle ceremony as necessary; just enough (or just too little) process.</td>
</tr>
<tr>
<td>value-added activities)</td>
<td>• Decisions delayed until latest feasible time.</td>
</tr>
<tr>
<td>Team ownership</td>
<td>• Team has primary responsibility and authority over its own plans and processes.</td>
</tr>
<tr>
<td></td>
<td>• Quality-performance is everyone’s responsibility.</td>
</tr>
</tbody>
</table>

Agility is about:

1. acknowledging the importance of people,
2. forming cross-functional teams,
3. obtaining high-bandwidth communication,
4. constantly reflecting and improving,
5. delivering value,
6. changing plans to take advantage of opportunities.
Scrum Basics

Scrum facilitates an incremental, feature-driven & time-boxed product realisation process.

SCRUM of Scrums

'meta' scrum team

system architect
product owner
integration & test engineer
debate team A
debate team B
debate team C

scrump team A
scrump team B
scrump team C
Agility in Systems Engineering

Production decisions will be delayed as far as possible to understand what competitors will be offering and what the latest emerging trend might be.

An early design freeze may increase the speed of development, but it is obviously difficult to modify or change a frozen concept.

There is a basic difference between an Agile and a fast running system engineering process.

Agility in Systems Engineering
Piecemeal Engineering

Karl Popper rejected revolutionary reform because "we can neither easily monitor the society-wide ramifications nor reverse our leaps".

monitoring and elimination of mistakes is more effective in the the small than in the large

Staged Product Development
Regular checks of defined and monitored internal or external influences
Set based design is based on the philosophy that one should consider working on a set of design solutions in parallel until one is forced to reduce the set of options to a smaller set.

**Instead of making an upfront overall design, a sequence of decision steps is defined and followed**

Narrowing the design space:
- new information become available
  - e.g. feasibility constraints
- long-lead items have to be made or ordered
- lack of time and/or resources

Set-based design is based on the philosophy that one should consider working on a set of design solutions in parallel until one is forced to reduce the set of options to a smaller set.

**Instead of making an upfront overall design, a sequence of decision steps is defined and followed**

Narrowing the design space:
- new information become available
  - e.g. feasibility constraints
- long-lead items have to be made or ordered
- lack of time and/or resources

Toyota implements set-based design extensively.

They make widespread use of a set of clay models before selecting a final vehicle for production.
Rule #1:

**Systems Engineering is a Learning-Based Process**

For the most part, engineers do not know how to build the systems they are trying to build; it is their job to find out how to build such systems.
Rule #2:

**Systems Engineering has Customer Value as prime focus**

Apart from being clear and concise, requirements must be also ranked in terms of importance. To add this dimension, system engineers should have more complete and multi-faceted interfaces with the customer.

Rule #3:

**Systems Engineering must exploit Short Iterations**

Instead of a 1-pass process according to a strict V-model, prototyping, modeling, demonstrating, and testing can all be iterative within an integrated systems engineering and development cycle.
Rule #4:

Systems Engineering must be neutral toward change

Unless systems engineering performs its activities and processes with an eye toward supporting change rather than avoiding or denying it, change will become an enemy (rather than an annoying but faithful family member).

Rule #5:

Systems Engineering is a Lean process

Lean is about the feasible delay of decisions that can have significant impact on operational acceptance or high-priority functionality. Delayed decisions can retain design flexibility longer, enabling more rapid reaction to internal or external changes.
Rule #6:

Systems Engineering is based on Team Ownership

Providing the systems engineering team with the authority and flexibility of owning their own process could radically improve their effectiveness.

Rule #7:

Systems Engineering adopts Just Enough Process Improvement

Not all uncertainties during system development and manufacturing can be resolved by incorporation of Agility in the process. Agility in systems is also needed to provide system adaptability and responsiveness after the system is shipped.
Agenda

• Introduction
  Double-sided view on Agile System Engineering,
  Agility in System Engineering, Engineering Agile Systems
  Process versus Product optimisation.

• Agile SYSTEMS ENGINEERING
  The Agile Manifesto, Agile Principles & Attributes,
  Scrum Basics, Multidisciplinary Scrum,
  Dealing with Uncertainty: Delayed Decisions
  Piecemeal Engineering: Staged Development,
  Set-Based Design: Narrowing the Design Space,
  7 Rules of Thumb for Agility in System Engineering.

• AGILE SYSTEMS Engineering
  The Price of Agility, When do we need Agility in Systems?
  Agile System Attributes, Response Ability,
  10 Design Principles for Agile Systems

Agility doesn't come for free ...

Intentional Agility means more effort in thinking, planning, rethinking, modifying

• where and to what extend do we need flexibility?
• what assumptions are questionable, unstable or incorrect?
• which influencing variables may change?
• which components in the system may be affected?
• how do we explain the need for flexibility to our contracting body?
• ...

increased complexity, cost,
additional interfaces,
technical penalties, ...
Engineering Agile Systems

Agility in systems is especially needed, when the systems are:

- Expensive, involving significant upfront investment cost.
- Long-lived, e.g. >10 years.
  User requirements may change significantly during the lifecycle.
- Significant switching costs exist,
  the expense might be too large for building an
  entirely new system each time the requirements change.

Systems Attributes

| **Agility** | Property of a system that can be modified from outside the system by an agent |
| **Flexibility** | Property of a system that can be changed easily. |
| **Robustness** | ability to perform under a variety of circumstances; system itself is capable of self-modification |
| **Adaptability** | ability of a system to change internally and autonomously to follow changes in its environment. |
Effective Response

Agility is the effective response to opportunity and problem

An effective response is one that is:
- Timely - fast enough to deliver value
- Affordable - at a cost that leaves room for another response
- Reliable - can be counted on to meet all expectations
- Comprehensive - can satisfy everything within mission boundary

Agility ≠ As fast as possible

Response Proficiency Space

Opportunistic  Agile

Fragile  Innovative

Proactive

Reactive
### Response Ability
#### Trade Off Analysis

<table>
<thead>
<tr>
<th>Response Competences</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>response time</td>
</tr>
<tr>
<td><strong>Proactive Dynamics</strong></td>
<td></td>
</tr>
<tr>
<td>1. creation/elimination</td>
<td></td>
</tr>
<tr>
<td>2. improvement</td>
<td></td>
</tr>
<tr>
<td>3. migration</td>
<td></td>
</tr>
<tr>
<td>4. modification</td>
<td></td>
</tr>
<tr>
<td><strong>Reactive Dynamics</strong></td>
<td></td>
</tr>
<tr>
<td>1. correction</td>
<td></td>
</tr>
<tr>
<td>2. variation</td>
<td></td>
</tr>
<tr>
<td>3. expansion/contraction</td>
<td></td>
</tr>
<tr>
<td>4. reconfiguration</td>
<td></td>
</tr>
</tbody>
</table>

### Architecting Agile Systems

**Reusable modules, reconfigurable in a scalable framework**

**10 Agile-System Design Principles**
Design Principles
For Agile Systems

1 Reusability

Encapsulated Modularity (Self-Contained Units)
Modules are encapsulated, distinct, separable, self-sufficient units;
Modules are cooperating toward a shared common purpose.

Plug Compatibility
Modules share defined interaction and interface standards;
Modules are easily inserted or removed.

Facilitated Reuse
Modules are reusable/replicable;
Responsibilities for ready re-use/replication and for management,
maintenance, and upgrade of component inventory are specifically designated.

2 Reconfigurability

Flat Interaction
Modules communicate directly on a peer-to-peer relationship;
Parallel rather than sequential relationships are favored.

Deferred Commitment
Module relationships are transient when possible;
Decisions and fixed bindings are postponed until immediately necessary;
Relationships are scheduled and bound in real-time.

Distributed Control and Information
Modules are directed by objective rather than method;
Decisions are made at point of maximum knowledge;
Information is associated locally, accessible globally, and freely disseminated.

Self-Organization
Module relationships are self-determined;
Component interaction is self-adjusting or negotiated
3 Scalability

Evolving Standards (Framework)
Frameworks standardize inter-module communication and interaction; define module compatibility, and have responsibilities designated for evolution and compatibility.

Redundancy and Diversity
Duplicate modules are employed to provide capacity right-sizing options and fail-soft tolerance; Diversity among similar modules employing different methods is exploited.

Elastic Capacity
Module populations may be increased and decreased within the framework.

References

• Toward Agile Systems Engineering Processes
  Richard Turner, Systems and Software Consortium

• Engineering Agile Systems: Creative-Guidance Frameworks
  for Requirements and Design
  Rick Dove, Stevens Institute of Technology

• Fundamental Principles for Agile Systems Engineering
  Rick Dove, Stevens Institute of Technology

• Agile SYSTEMS ENGINEERING versus AGILE SYSTEMS engineering
  Reinhard Haberfellner, TU Graz & Olivier de Weck, MIT
  strategic.mit.edu/PDF_archive/3%20Refereed%20Conference/  _3_59_INCOSE-2005-AGSEvsEAGS.pdf