Proposed Unified “ility” Definition Framework

Andrew Long
October 2012
Motivation

• Increased interest in system “ilities”

• Lack of common understanding among practitioners
  • Definitions
  • Relationships between one ility and another
  • Metrics

• Objective: Provide a framework to promote shared discussion of system “ilities”
What are system “ilities”?

• -ility
  • Latin: a suffix; meaning, ability, ability to [do something]
  • In systems engineering “ilities” are desired system properties

• “ilities” describe the system (non-functional) rather than specific system behaviors (functional)
  • Functional requirements define what a system is supposed to do; e.g. Performance
  • Non-functional requirements define how a system is supposed to be.
Some examples?

- Accessibility
- Accountability
- Adaptability
- Administrability
- Affordability
- Auditability
- Availability
- Credibility
- Compatibility
- Configurability
- Correctness
- Customizability
- Debugability
- Degradability
- Determinability
- Demonstrability
- Dependability
- Deployability
- Distributability
- Durability
- Effectiveness
- Evolvability

- Extensibility
- Fidelity
- Flexibility
- Installability
- Integrity
- Interchangeability
- Interoperability
- Learnability
- Maintainability
- Manageability
- Mobility
- Modifiability
- Modularity
- Operability
- Portability
- Predictability
- Provability
- Recoverability
- Reliability
- Repeatability
- Reproducibility
- Resilience

- Responsiveness
- Reusability
- Robustness
- Safety
- Scalability
- Sustainability
- Serviceability
  (a.k.a. supportability)
- Securability
- Simplicity
- Stability
- Survivability
- Sustainability
- Tailorability
- Testability
- Timeliness
- Traceability
- Ubiquity
- Understandability
- Upgradability
- Usability
- Versatility

“List of system quality attributes” - Wikipedia, May 2012
Examples of “ilities” within the DoD

- Safety, Reliability, Availability, Maintainability, Testability (Traditional Systems Engineering)
- Interoperability (Net-Centric Warfare)
- Scalability / Extensibility (Software Design)
- Survivability (Military Platforms: Aircraft, Maritime, Ground)
- Resiliency (Space Systems, Cyber)
  - Engineering Resilient Systems OSD(SE) initiative
- Flexibility (DARPA F6)
- Adaptability (Defense Science Board 2010 Report, DARPA AVM/ META)
- Affordability (Defense Acquisitions)
  - USD/AT&L 2010 Better Buying Power memo “Affordability as a requirement”
- Sustainability (Supply Chain, Industrial Base, Work Force)
Common element: Uncertainty

In system engineering, uncertainties occur in performance & value expectations

- Performance: Variance between actual and desired system performance resulting from uncertainty within contexts (e.g. design, production, operations, etc.)
- Value: Variance in “expected value” resulting from feedback of delivered value, resulting from changing context, stakeholders, needs, etc.

“ilities” account for a system’s ability to change / react to uncertainty

Designing for Uncertainty

• System “changeability” taxonomy (Ross et al.) provides start for defining system ilities

  • Change Agent
  • Change Effect
  • Change Objective
  • Change Enablers
  • Change Considerations

Change Agents

• Instigator, or force, which employs a given change mechanism in order to achieve a desired change effect

• Two approaches to choosing a change agent
  • **Internal**: System self-imposed change (Adaptability)
    • Associated with upfront / current decisions to respond to change (i.e. Pre-planned / Baked-in)
    • Commonly used to address known - knowns
  
  • **External**: Decision-maker imposed change (Flexibility)
    • Associated with future / delayed decisions to respond to change (i.e. Real options)
    • Used to address known - knowns & known - unknowns
Change Effect

- The difference in system states (performance or value) before and after a change has taken place

- Three choices of change effects
  - Expandability (Do More / Less)
    - Ross et al. references scalability, but expandability used to de-conflict w/ more common scalability definitions
  - Modifiability (Add, Remove, Alter)
  - Robustness (No Change)
The specific approach / plan / goal / strategies employed to achieve a desired change effect

- Objectives enabled by change enablers and must account for change considerations
- Think of these as common “ility” families.

<table>
<thead>
<tr>
<th>Change Effect</th>
<th>Performance Objectives (Static)</th>
<th>Value Objectives (Dynamic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness</td>
<td>Quality (Assurance and control)</td>
<td>Resiliency</td>
</tr>
<tr>
<td></td>
<td>Safety</td>
<td>• Awareness</td>
</tr>
<tr>
<td></td>
<td>Reliability</td>
<td>• Survivability</td>
</tr>
<tr>
<td></td>
<td>Availability</td>
<td>• Susceptibility</td>
</tr>
<tr>
<td></td>
<td>Maintainability</td>
<td>• Vulnerability</td>
</tr>
<tr>
<td></td>
<td>Testability</td>
<td>• Recoverability</td>
</tr>
<tr>
<td>Expandability</td>
<td>Scalability</td>
<td>Extensibility</td>
</tr>
<tr>
<td>Modifiability</td>
<td>Configurability</td>
<td>Evolvability</td>
</tr>
</tbody>
</table>
Change Enablers

- Change enablers (e.g. design elements) enable desired objective
- “How” is the system is designed to change
- Any one, or combination of, enablers can satisfy an objective
- Any enabler can satisfy one or more objectives

<table>
<thead>
<tr>
<th>Change Enablers</th>
<th>Accessibility</th>
<th>Compatibility</th>
<th>Commonality</th>
<th>Distributability</th>
<th>Durability</th>
<th>Homogeneity</th>
<th>Heterogeneity</th>
<th>Interchangeability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mobility</td>
<td>Modularity</td>
<td>Portability</td>
<td>Repairability</td>
<td>Reusability</td>
<td>Serviceability (a.k.a. supportability)</td>
<td>Understandability</td>
<td>Usability / operability</td>
<td></td>
</tr>
</tbody>
</table>
Change Considerations

- Design considerations (e.g. conditions, resources, constraints, etc.) applied to design / operational approaches

- **Affordability** (Budget)
- **Sustainability** (Resources)
- **Agility / Responsiveness** (Schedule / Response time)
- **Manufacturability** (Technology)
- **Manageability** (Organizational)
```
Innovate - Engineer - Execute

Proposed “ility” Framework

<table>
<thead>
<tr>
<th>Change Agents</th>
<th>Flexibility (External)</th>
<th>Adaptability (Internal)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Performance Change Objectives</td>
<td>Value Change Objectives</td>
</tr>
<tr>
<td>Robustness</td>
<td>Quality (Assurance and control)</td>
<td>Resiliency</td>
</tr>
<tr>
<td></td>
<td>• Safety</td>
<td>• Awareness</td>
</tr>
<tr>
<td></td>
<td>• Reliability</td>
<td>• Susceptibility</td>
</tr>
<tr>
<td></td>
<td>• Availability</td>
<td>• Vulnerability</td>
</tr>
<tr>
<td></td>
<td>• Maintainability</td>
<td>• Recoverability</td>
</tr>
<tr>
<td></td>
<td>• Testability</td>
<td></td>
</tr>
<tr>
<td>Expandability</td>
<td>Scalability</td>
<td>Extensibility</td>
</tr>
<tr>
<td>Modifiability</td>
<td>Configurability</td>
<td>Evolvability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change Enablers</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accessibility</td>
<td>Mobility</td>
</tr>
<tr>
<td></td>
<td>Compatibility</td>
<td>Modularity</td>
</tr>
<tr>
<td></td>
<td>Commonality</td>
<td>Portability</td>
</tr>
<tr>
<td></td>
<td>Distributability</td>
<td>Repairability</td>
</tr>
<tr>
<td></td>
<td>Durability</td>
<td>Reusability</td>
</tr>
<tr>
<td></td>
<td>Homogeneity</td>
<td>Serviceability (a.k.a. supportability)</td>
</tr>
<tr>
<td></td>
<td>Heterogeneity</td>
<td>Understandability</td>
</tr>
<tr>
<td></td>
<td>Interchangeability</td>
<td>Usability / operability</td>
</tr>
<tr>
<td></td>
<td>Interoperability</td>
<td>(a.k.a. supportability)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change Considerations</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Affordability</td>
<td>Agility / Responsiveness</td>
</tr>
<tr>
<td></td>
<td>Sustainability</td>
<td>Manufacturability</td>
</tr>
</tbody>
</table>
```
Questions?