The Forgotten “-llities”

James D. Willis, Jr.
SPEC Innovations
10440 Balls Ford Road
Suite 230
Manassas VA 20109
Topics

• What is an “-Ility”?  

• How might we organize “-Ilities”?  

• How Should Systems Engineers  View “-Ilities”?  

• Summary
Most Common Lists of -llities

**RAM-T (Eng)**
- Reliability
- Availability
- Maintainability
- Testability

**RASR (DBs)**
- Reliability
- Availability
- Scalability
- Recoverability

**RAMS (Safety)**
- Reliability
- Availability
- Maintainability
- Safety

**RASUI (SW)**
- Reliability
- Availability
- Serviceability
- Usability
- Instability

**FURPS (SW)**
- Functionality
- Usability
- Reliability
- Performance
- Supportability
Are there more –Ilities?

Accessibility  Executability  Performability  Supportability
Accountability  Extensibility  Portability  Suitability
Adaptability  Evolvability  Practibilty  Survivability
Administrability  Fidelity  Practicality  Tailorability
Affordability  Flexibility  Predictability  Testability
Agility  Functionality  Producibility  Traceability
Availability  Integratibility  Recoverability  Trainability
Capability  Interoperability  Reliability  Transportability
Composability  Interpretability  Repeatability  Trustability
Configurability  Maintainability  Responsibility  Understandability
Compatibility  Manageability  Reusability  Upgradability
Demonstrability  Mobility  Scalability  Usability
Deployability  Modifiability  Serviceability  Verifiability
Durability  Operability  Stability  Vulnerability
What is the Definition of “-Ility”

The developmental, operational, and support requirements a program must address (e.g., availability, maintainability, vulnerability, reliability, supportability, etc.).
What is an “-Iality”: Other Terms

“Feature”

“Characteristic”

“Attribute”

“Quality Goals”

“Constraints”

“Other properties”

Most Common: Non-functional requirement
## Functional vs Nonfunctional Requirements (SW)

<table>
<thead>
<tr>
<th>Functional</th>
<th>Nonfunctional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product features</td>
<td>Product properties</td>
</tr>
<tr>
<td>Describe the work that is done</td>
<td>Describe the character of the work</td>
</tr>
<tr>
<td>Describe the actions with which the work is concerned</td>
<td>Describe the experience of the user while doing the work</td>
</tr>
<tr>
<td>Characterized by verbs</td>
<td>Characterized by adjectives</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Functional</th>
<th>Nonfunctional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Functions and behaviors</td>
<td>Criteria that can be used to judge the operation of a system</td>
</tr>
<tr>
<td>System Design</td>
<td>System Architecture</td>
</tr>
<tr>
<td>What a system is supposed to DO</td>
<td>What a system is supposed to BE</td>
</tr>
<tr>
<td></td>
<td>Characteristic of a system that applies across a set of functional or system requirements.</td>
</tr>
</tbody>
</table>

Software Architecture Notes: [making the ilities come true](http://www.softwarearchitecturenotes.com/architectureRequirements.html)
MBSE: LML Top-level Schema

• Show that for SEs the it is necessary for us to work Functional with nonfunctional
<table>
<thead>
<tr>
<th>Accessibility</th>
<th>Executability</th>
<th>Performability</th>
<th>Supportability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accountability</td>
<td>Extensibility</td>
<td>Portability</td>
<td>Suitability</td>
</tr>
<tr>
<td>Adaptability</td>
<td>Evolvability</td>
<td>Practibility</td>
<td>Survivability</td>
</tr>
<tr>
<td>Administrability</td>
<td>Fidelity</td>
<td>Practicality</td>
<td>Tailorability</td>
</tr>
<tr>
<td>Affordability</td>
<td>Flexibility</td>
<td>Predictability</td>
<td>Testability</td>
</tr>
<tr>
<td>Agility</td>
<td>Functionality</td>
<td>Producibility</td>
<td>Traceability</td>
</tr>
<tr>
<td>Availability</td>
<td>Integratibility</td>
<td>Recoverability</td>
<td>Trainability</td>
</tr>
<tr>
<td>Capability</td>
<td>Interoperability</td>
<td>Reliability</td>
<td>Transportability</td>
</tr>
<tr>
<td>Composability</td>
<td>Interpretability</td>
<td>Repeatability</td>
<td>Trustability</td>
</tr>
<tr>
<td>Configurability</td>
<td>Maintainability</td>
<td>Responsibility</td>
<td>Understandability</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Manageability</td>
<td>Reusability</td>
<td>Upgradability</td>
</tr>
<tr>
<td>Demonstrability</td>
<td>Mobility</td>
<td>Scalability</td>
<td>Usability</td>
</tr>
<tr>
<td>Deployability</td>
<td>Modifiability</td>
<td>Serviceability</td>
<td>Verifiability</td>
</tr>
<tr>
<td>Durability</td>
<td>Operability</td>
<td>Stability</td>
<td>Vulnerability</td>
</tr>
</tbody>
</table>

© 2011 Systems and Proposal Engineering Company. All rights reserved.
How Can We Organize “-Ilities”?
How can we organize this disparate List?

- Lifecycle phase
- Dependency and Priority
- Cost and value
- Criticality

Group these by
- Relationship
- Timeline on Lifecycle
- Dependencies
- Aggregation
- Priority
- Value

Questions:
- Do -ilities describe the product
- Are they more associated with SE functions leading to design?
- Do they drive product design
- Are they key to ensuring the product parts can be integrated?
- How do they relate to SE processes?
Dynamic Relationship

Seeking to establish and maintain balance between two attributes in a dynamic environment.
Hierarchical Relationships: Example 1

- Agility
  - Debuggability
  - Extensibility
  - Scalability
  - Securability
  - Testability
  - Understandability

Hierarchical Relationships: Example 2

General Utility
- Portability
  - Reliability
  - Human Engineering
  - Efficiency
  - Testability
  - Understandability
  - Modifiability
- Usability
- Maintainability
  - Device-Independence
  - Self-Containedness
  - Accuracy
  - Completeness
  - Robustness
  - Consistency
  - Accountability
  - Device Efficiency
  - Accessibility
  - Communications
  - Self-descriptiveness
  - Structuredness
  - Conciseness
  - Legibility
  - Augmentability
Boehm+Brown+Lipow1976 fig.1, corrected to match qualities named in text. "General Utility" and "Device Efficiency" are not defined there.
How Should Systems Engineers View “-illities”?
What is a System?

...combination of interacting elements organized to achieve one or more stated purposes.

...an integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements.
What is a System?

People

Things

Processes
What is a System?

**People**
- Operability
- Suitability
- Survivability
- Trainability
- Understandability

**Things**
- Affordability
- Adaptability
- Agility
- Usability
- Verifiability
- Vulnerability
- Integratibility
- Performability
- Repeatability

**Processes**
Systems Engineering Lifecycle: Traceability

Current Operations and Support
- Architecture Development
- System Design
- Hardware/Software Acquisition

Future Operations and Support
- Operational T&E and Transition
- Integration and Test

Future Operations and Support
- Integration

Future Operations and Support
- Decomposition

Future Operations and Support
- Traceability

Future Operations and Support
- Program Management

Demolition and Disposal

INCOSE Systems Engineering Handbook v. 3.2.1
INCOSE-TP-2003-002-03.2.1 January 2011

© 2011 Systems and Proposal Engineering Company. All rights reserved.
Systems Engineering Lifecycle: Integratability

Current Operations and Support
- Architecture Development
- System Design
- Hardware/Software Acquisition
- Operational T&E and Transition
- Integration and Test

Future Operations and Support
- Future Operations and Support
- Demolition and Disposal

Integratability
- Decomposition
- Integration

Program Management

© 2011 Systems and Proposal Engineering Company. All rights reserved.
Systems Engineering Lifecycle: Verifiability

Current Operations and Support

Future Operations and Support

Demolition and Disposal

Architecture Development

System Design

Operational T&E and Transition

Hardware/Software Acquisition

Future Operations and Support

Verifiability

Decomposition

Integration

Program Management

© 2011 Systems and Proposal Engineering Company. All rights reserved.
Measurement of -llities

• Standard measurements – not always agreed to

• Some accepted measurements
  – Availability - \( P_A = 1 - \frac{MTTR}{MTBF} \)
  – Maintainability - MTTR *mean time to repair (or restore)*
  – Reliability – MTBF *mean time between failure*
  – SW Maintainability - Lines-of-code measures, McCabe Measures, Halstead Complexity Measures
  – Security – Malware statistics, Firewall statistics, Vulnerability
-Ility Related Research

• 2006-2007 John W. Dahlgren MITRE
  – “System Complexity, the “ilities” and Robustness” Project

• Current - SEArI Systems Engineering Advancement Research Initiative - MIT
  – “Ingenuity, Innovation, and the ilities: Creating Capabilities for the Long Run”
Increasing Emphasis and Demand

In DoD

• Interoperability
  – CJCSI 6212.01E Interoperability And Supportability Of Information Technology And National Security Systems

• Producibility
  – DoDI 5000.02 Operation of the Defense Acquisition System

• DOTMLPF (Doctrine, Organization, Training, Materiel, Leadership and education, Personnel, and Facilities)
  – Embedded throughout Joint and Service Standards

Commercial World...
Even Commercial Interest is Increasing
Summary

• Little recent SE discussion and writing on ‘-ilities’
• “-Ilities” are key system attributes
• Many useful and/or necessary “-ilities” are
  – Not understood well
  – Often forgotten…or ignored
• Systems Engineers should work to integrate more –ilities into systems development

• Recommendation: Increase discussion and interchange among SEs on the topic of “-ilities” and h