Six Sigma Requirement Development
Tools Assure More Reliable Software

SERE International Conference
NIST Gaithersburg
June 19, 2013

Dr. Samuel Keene, FIEEE
s.keene@ieee.org
Abstract

- Getting good product requirements and have them understood across the development team is the number 1 system challenge. These problems contribute to 90% of the today’s critical system issues. Design for Six Sigma gives us a set of tools to better develop system requirements, promote their cross functional understanding and establish system requirements traceability. This presentation introduces 5 such development aids.
Notorious Failures (assignable cause)

• Jupiter Fly by – Programmed to switch power supplies if communication not received within in 7 days (15 year mission)
• Mars Climate Orbitor (1998) *mix of metric and Imperial units*
• FP&L 2008 power outage: “SW performed exactly as it was programmed to do”
Biggest System Challenge

Systems Management – Brendan Murphy

1. Requirements Challenge
   - Needs Data
   - Context Data

2. Interface Challenge
   “Tell people not just so they understand, tell them so they cannot misunderstand” (Mark Twain)

3. Managing Changes (Keene)
   Design Understandability
   Situational awareness
   Manage unintended consequences
   Configuration control
Small Changes are Error Prone with unintended consequences

<table>
<thead>
<tr>
<th>LOC Changed</th>
<th>Likelihood of error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 line</td>
<td>50%</td>
</tr>
<tr>
<td>5 lines</td>
<td>75%</td>
</tr>
<tr>
<td>20 lines</td>
<td>35%</td>
</tr>
</tbody>
</table>

Classic Example: DSC Corp, Plano Texas, 3 bits of a MSLOC program were changed leading to municipal phone outages in major metropolitan areas.

“Accurately capturing requirements is the major factor in the failure of 90% of large software projects”

Six Sigma

- Six Sigma is a process of asking questions that lead to tangible, quantifiable answers that ultimately produce profitable (trustworthy) results

Mikel Harry

Question Requirements, Question Data, Question Process, Question Assumptions – Samuel Keene
Six Sigma focus systematically using:

1. Cognitive tools
   - Eg., Mind Map, Flow charts, Swim Lane
2. Analytical tools
   - Eg., FMEA, FTA, QFD
3. Statistical decision support tools
   - Special Cause vs Common Cause
DFSS Tools Aid Requirements Development

• **Situational awareness** – (Environmental focus)

• **Navigability** - (internal consistency, interfaces)

• **Product understandability** to reduce interoperability problems and improve the robustness to design changes. This also helps in assessing the completeness of the design and identifying improvement opportunities.

• **Compact focus** to see the overall design in a single view or a more limited view (if more frames are required)

• **Differing System Views** to get “fresh” examinations of the system under development, looking for improvement opportunities and design completeness.
holistic (h -l st k)

a. Emphasizing the importance of the whole and the interdependence of its parts.

b. Concerned with wholes rather than analysis or separation into parts: holistic medicine; holistic ecology.

Note: Safety, security, reliability, and survivability are system attributes.
Pareto Principle

- Vilfredo Pareto - Italian economist from the 1800’s
  “Very few parts cause most of the system problems”
- 80% of the problems are linked to only 20% of the causes
- Sometimes 99%, 1%
- Focus on: Important, New, Critical, and Difficult
Presentation Goals

- “DFSS tools” focus for requirements development, team building, product understandability, and robustness
  1. Mind map
  2. Kano
  3. GQM
  4. Defect Prevention Process
  5. QFD
  6. Pair Programming
Tool 2: Customer Fulfillment: Kano Diagram

- **Satisfaction**
  - Unexpected (Unspoken)
  - Specified
  - Expected (Unspoken)

- **Dissatisfaction**
  - Requirement Unfulfilled
  - Requirement Fulfilled
Metrics drive behavior: right metrics drive the right behavior
Metrics should answer questions of interest
Ratio (variable) measures are the best metrics
Plan tool usage and action plans
Tool 4: Defect Prevention Process (DPP)

- Diagnose Pareto significant failures
  - Frequency
  - Severity
- Understand underlying fault
  - Parse similar faults
  - How to preclude or mitigate
  - How to detect sooner
- Teach JIT fault avoidance by phase
- Stop/Start problems
- Six Sigma fixes product and the process
DFSS Focus

• Process *generation* over process *improvement*
• Requirements development focus
• Requirements allocation and flowdown
• Cross functional development and process/product visualization tools

• Tools map link
Valuable Links

- *RISKS-FORUM Digest* _ Peter G. Neumann
- Standard for Software Reliability Prediction
  IEEE_P_1633
- iSixSigma.com
- Wikipedia (DFSS)
- Mindjet.com (Mindmap)
- Edward Tufte (the Leonardo Da Vinci of data)