Software Design-for-Six Sigma (SDFSS) is Emerging!

By Robert W. Stoddard
Motorola Six Sigma Master Black Belt
Senior Member of Technical Staff
Software Engineering Institute

This material is approved for public release.

Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University
CMU Service Marks

The following are service marks of Carnegie Mellon University:

• Personal Software Process℠
• PSP℠
• Team Software Process℠
• TSP℠

The following are registered in the U.S. Patent & Trademark Office by Carnegie Mellon University:

• Capability Maturity Model®
• CMM®
• Capability Maturity Model Integration®
• CMMI®
• SEI’s Architecture Tradeoff Analysis Method®
• ATAM®
SDFSS Opportunities

Impacts without SDFSS

• Multi-year development projects failed to deliver a working product (min cost=$7M)
  • Failure 1: Did not model performance of new chipset, processor or language
  • Failure 2: Did not adequately characterize the market and business case
  • Failure 3: Did not adequately test the product

Benefits with SDFSS

• Business cases modeling all reasonable uncertainties in market and customer segments
  • Schedules with uncertainties modeled
  • Req’ts identified, with KJ analysis, to delight customers
  • Design of Experiments used to: optimize and patent fuel efficiency; test object-oriented software; test robustness with fault insertion testing; reduce flight test by 10x
Purpose of this Talk

• To proclaim that software DFSS, within a holistic DFSS approach to product development, is coming of age,

• To demonstrate that many gaps, in translating traditional DFSS concepts to software engineering, may be solved by the adoption of a number of Software Engineering Institute (SEI) technologies.

Thus, DFSS does not have to be re-invented for Software Engineering!
Target Audience

• Executives and Directors contemplating investing in Software DFSS

• Deployment champions who may be tasked with the training and roll-out of Software DFSS

• DFSS and Software Engineering Leaders who need to understand both disciplines, and who can lead in translating and interpreting key concepts and tools between the two disciplines.
Holistic View of Software DFSS

- Portfolio for Six Sigma
- Marketing for Six Sigma
- Product Commercialization for Six Sigma
- Technology Platform R&D for Six Sigma
- Sales & Distribution for Six Sigma
- Supply Chain for Six Sigma
SEI Technologies Boost DFSS!

The SEI Capability Maturity Model Integrated (CMMI) has a product development perspective that overlaps significantly with all of the DFSS methodologies!
The CMMI, TSP and PSP

CMMI - for organizational capability

TSP - for quality products on cost and schedule

PSP - for individual skill and discipline

CMMI Models

Bodies of Knowledge

- Systems Engineering
- Software Engineering
- Integrated Product and Process Development
- Supplier Sourcing

Model Framework

- CMMI-SE/SW/IPPD/SS (Continuous)
- CMMI-SE/SW/IPPD/SS (Staged)
- CMMI-SE/SW/IPPD (Continuous)
- CMMI-SE/SW/IPPD (Staged)
- CMMI-SE/SW (Continuous)
- CMMI-SE/SW (Staged)
- CMMI-SW (Continuous)
- CMMI-SW (Staged)
The Maturity Levels

1. Process unpredictable, poorly controlled, and reactive
2. Process characterized for projects and is often reactive
3. Process characterized for the organization and is proactive
4. Process measured and controlled
5. Focus on continuous process improvement

Optimizing

- Quantitatively Managed
- Defined
- Managed
- Initial
## PAs by Both Representations

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Process Management</th>
<th>Project Management</th>
<th>Engineering</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Organizational Innovation and Deployment</td>
<td></td>
<td></td>
<td>Causal Analysis and Resolution</td>
</tr>
<tr>
<td>4</td>
<td>Organizational Process Performance</td>
<td>Quantitative Project Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Organizational Process Focus</td>
<td>Integrated Project Management for IPPD Risk Management Integrated Teaming Integrated Supplier Management</td>
<td>Requirements Development Technical Solution Product Integration Verification Validation</td>
<td>Decision Analysis and Resolution Organizational Environment for Integration</td>
</tr>
</tbody>
</table>
## Performance Results Summary


<table>
<thead>
<tr>
<th>Improvements</th>
<th>Median</th>
<th># of data points</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>34%</td>
<td>29</td>
<td>3%</td>
<td>87%</td>
</tr>
<tr>
<td>Schedule</td>
<td>50%</td>
<td>22</td>
<td>2%</td>
<td>95%</td>
</tr>
<tr>
<td>Productivity</td>
<td>61%</td>
<td>20</td>
<td>11%</td>
<td>329%</td>
</tr>
<tr>
<td>Quality</td>
<td>48%</td>
<td>34</td>
<td>2%</td>
<td>132%</td>
</tr>
<tr>
<td>Customer Satisfaction</td>
<td>14%</td>
<td>8</td>
<td>-4%</td>
<td>55%</td>
</tr>
<tr>
<td>Return on Investment</td>
<td>4.0 : 1</td>
<td>22</td>
<td>1.7 : 1</td>
<td>27.7 : 1</td>
</tr>
</tbody>
</table>

- N = 30 organizations, as of June 2006
- Results expressed as change over varying periods of time
The CMMI Performance Results Web Site

Results by: [http://www.sei.cmu.edu/cmmi/results.html](http://www.sei.cmu.edu/cmmi/results.html)

- Performance category & organization
- Brief statements & graphical examples
- Full source documents

Objective and Scope

There is a widespread demand for credible, quantitative evidence about the results of process improvement based on CMMI models. The results presented here are from publicly available conference presentations, published papers, and individual collaborations with the SEI.

Together, these results provide proof of concept about the potential of CMMI-based process improvement. The results show that CMMI often leads to very impressive improvements in product quality, project performance, and organizational performance; however, the individual results presented here may not be repeatable in every organization.
The SEI Team Software Process (TSP) and Personal Software Process (PSP) significantly enhance the software development teaming within Product Commercialization and Technology Platform R&D for Six Sigma!
The Personal Software Process

Reference: “The TSP and PSP Tutorial” by the SEI (Burton, Davis, McHale), 2000
PSP vs non-PSP Results

Project A:
- Disciplined team design process used to create sound developer work packets
- Personal Software Process used consistently by developers
  - SEI developed course for software developers which provides process at an individual level for producing software components and documentation (user & technical)
- Peer Reviews conducted on the most critical 20% of the software

Project B:
- We decided that this project was “too far along” to benefit from process insertion

Reference: 3rd Annual CMMI Technology Conference and User Group, Kent Schneider, 2003
## PSP vs non-PSP Test Time & Cost

<table>
<thead>
<tr>
<th></th>
<th>Project A</th>
<th>Project B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration / Acceptance Test Cost</td>
<td>$78.K</td>
<td>$612K</td>
</tr>
<tr>
<td>Normalized (per KLOC)</td>
<td>$0.95K</td>
<td>$4.05K</td>
</tr>
<tr>
<td>Time to Accept (months)</td>
<td>3.7</td>
<td>14.6</td>
</tr>
<tr>
<td>Normalized (months per 100 KLOC)</td>
<td>4.5</td>
<td>9.7</td>
</tr>
</tbody>
</table>

Reference: 3rd Annual CMMI Technology Conference and User Group, Kent Schneider, 2003
## PSP vs non-PSP Quality

<table>
<thead>
<tr>
<th></th>
<th>Project A</th>
<th>Project B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (months)</td>
<td>31.8</td>
<td>43.0</td>
</tr>
<tr>
<td>Size (KLOC)</td>
<td>82</td>
<td>151</td>
</tr>
<tr>
<td>Developer Defect Density</td>
<td>9.4</td>
<td>17.3</td>
</tr>
<tr>
<td>Peer Review Exit Density</td>
<td>4.78</td>
<td>17.3</td>
</tr>
<tr>
<td>Delivered Defect Density</td>
<td>1.55</td>
<td>5.27</td>
</tr>
</tbody>
</table>

Reference: 3rd Annual CMMI Technology Conference and User Group, Kent Schneider, 2003
TSP Builds Software Teams

Skill-building
- Personal plans
- Practice in planning
- Planning methods
- Quality measures
- Defined processes

Team-building
- Commitment
- Aggressive plans
- Quality ownership
- Project goals
- Plan ownership
- Plan detail
- Team roles
- Team resources

Team-working
- Quality priority
- Cost of quality
- Follow the process
- Review status
- Review quality
- Communication
- Change management

Engineering Disciplines
Team Disciplines
Management Disciplines

Integrated Product Teams

Reference: “The TSP and PSP Tutorial” by the SEI (Burton, Davis, McHale), 2000
The TSP Launch Process

**Day 1**
1. Establish product and business goals
2. Assign roles and define team goals
3. Produce development strategy

**Day 2**
4. Build top-down and next-phase plans
5. Develop the quality plan
6. Build bottom-up and balanced plans

**Day 3**
7. Conduct risk assessment
8. Prepare management briefing and launch report
9. Hold management review

10. Launch postmortem

Reference: “The TSP and PSP Tutorial” by the SEI (Burton, Davis, McHale), 2000
Benefits Seen from TSP

Average time to move to the next maturity level

- Level 2: 34 months
- Level 3: 25 months
- Level 4: 30 months
- Level 4 + PSP/TSP: 20 months
- Level 5: 10 months

Based on 216 Boeing organization assessments conducted between 1991-2000

Reference: Keynote at European SEPG, John Vu, 2001
Additional Benefits from TSP

Source: CMU/SEI-2003-TR-014

Defects/KLOC

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
<th>TSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.5</td>
<td>6.24</td>
<td>4.73</td>
<td>2.28</td>
<td>1.05</td>
<td>0.06</td>
</tr>
</tbody>
</table>

© 2005 by Carnegie Mellon University
SEI Technologies Boost DFSS!

The SEI Product Line Practice work overlaps significantly with both Portfolio for Six Sigma, and Technology Platform R&D for Six Sigma!

- Portfolio for Six Sigma
- Marketing for Six Sigma
- Product Commercialization for Six Sigma
- Technology Platform R&D for Six Sigma
- Sales & Distribution for Six Sigma
- Supply Chain for Six Sigma
Software Product Lines

- Products pertain to Market strategy/Application domain
- Products share an Architecture
- Products are built from Components

Product lines
- take economic advantage of commonality
- bound variability

# Product Line Practice Framework


## Essential Activities

<table>
<thead>
<tr>
<th>Software Engineering</th>
<th>Technical Management</th>
<th>Organizational Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Definition</td>
<td>Configuration Management</td>
<td>Building a Business Case</td>
</tr>
<tr>
<td>Architecture Evaluation</td>
<td>Data Collection, Metrics, and Tracking</td>
<td>Customer Interface Management</td>
</tr>
<tr>
<td>Component Development</td>
<td>Make/Buy/Mine/Commission Analysis</td>
<td>Implementing an Acquisition Strategy</td>
</tr>
<tr>
<td>COTS Utilization</td>
<td>Process Definition</td>
<td>Funding</td>
</tr>
<tr>
<td>Mining Existing Assets</td>
<td>Scoping</td>
<td>Launching and Institutionalizing</td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td>Technical Planning</td>
<td>Market Analysis</td>
</tr>
<tr>
<td>Software System Integration</td>
<td>Technical Risk Management</td>
<td>Operations</td>
</tr>
<tr>
<td>Testing</td>
<td>Tool Support</td>
<td>Organizational Planning</td>
</tr>
<tr>
<td>Understanding Relevant Domains</td>
<td></td>
<td>Organizational Risk Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structuring the Organization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Technology Forecasting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Training</td>
</tr>
</tbody>
</table>

March 2, 2007

© 2005 by Carnegie Mellon University
CelsiusTech: Ship System 2000

A family of 55 ship systems

Integration test of 1-1.5 million SLOC requires 1-2 people
Rehosting to a new platform/OS takes 3 months
Cost and schedule targets are predictably met
Performance/distribution behavior are known in advance
Customer satisfaction is high
Hardware-to-software cost ratio changed from 35:65 to 80:20

Nokia Mobile Phones

Product lines with 25-30 new products per year

Across products there are
• varying number of keys
• varying display sizes
• varying sets of features
• 58 languages supported
• 130 countries served
• multiple protocols
• needs for backwards compatibility
• configurable features
• needs for product behavior change after release

SEI Technologies Boost DFSS!

The SEI Software Architecture work overlaps significantly with both Portfolio for Six Sigma, Product Commercialization for Six Sigma, and Technology Platform R&D for Six Sigma!
Software Architecture Technology (SAT) Initiative’s Focus

Ensure that business and mission goals are predictably achieved by using effective software architecture practices throughout the development lifecycle.

Axioms Guiding Our Work

- Software architecture is the bridge between business and mission goals and a software-intensive system.
- Quality attribute requirements drive software architecture design.
- Software architecture drives software development throughout the life cycle.

SEI’s Architecture Tradeoff Analysis Method® (ATAM®)

ATAM is an architecture evaluation method that

- focuses on multiple quality attributes
- illuminates points in the architecture where quality attribute *tradeoffs* occur
- generates a context for ongoing quantitative analysis
- utilizes an architecture’s vested stakeholders as authorities on the quality attribute goals

Conceptual Flow of the ATAMSM

Business Drivers → Quality Attributes → Scenarios
Software Architecture → Architectural Approaches → Architectural Decisions

Analysis

imparts

distilled into

Risk Themes

Tradeoffs
Sensitivity Points
Non-Risks
Risks

March 2, 2007
ATAM Led to the Development of Other Methods and Techniques

Quality Attribute Workshop
- What if the quality requirements are not well-understood?

Attribute Driven Design
- What if there’s no architecture?

Architecture Reconstruction using ARMIN
- What if I don’t know my system’s architecture?

Quality Attribute General Scenarios
- Our scenarios tend to be incomplete or ambiguous.

Quality Attribute Tactics
- What are some of the most important questions to ask?

Cost Benefit Analysis Method
- Which risks should I work on first?

Views and Beyond Approach
- What information should be included in my architecture documentation?

Reference: "Future Directions of the Software Architecture Technology Initiative", Second Annual SATURN Workshop, Mark Klein, 2006
## ATAM Benefits Observed by Bosch

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
</tr>
</thead>
<tbody>
<tr>
<td># attending stakeholders</td>
<td>16</td>
<td>20</td>
<td>21</td>
<td>18</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td># evaluators</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Duration of ATAM workshop</td>
<td>2 days</td>
<td>2 days</td>
<td>2 days</td>
<td>2 days</td>
<td>3 days</td>
<td></td>
</tr>
<tr>
<td># generated scenarios</td>
<td>44</td>
<td>71</td>
<td>47</td>
<td>66</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td># analyzed scenarios</td>
<td>12</td>
<td>11</td>
<td>9</td>
<td>12</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td># identified decisions</td>
<td>64</td>
<td>62</td>
<td>51</td>
<td>49</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td># identified tradeoffs</td>
<td>8</td>
<td>15</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td># identified sensitivity points</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td># identified risks</td>
<td>44</td>
<td>57</td>
<td>17</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td># identified non-risks</td>
<td>17</td>
<td>26</td>
<td>9</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td># identified risk themes</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td># business drivers</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Reference: “Architecture Reviews @ Bosch”, 1st SEI SATURN, Ferber, 2005
ATAM Benefits Observed by Bosch

On Architecture
- Risk avoidance
- Better Documentation
- Identification of non-risks

On People
- Disappointment
- Consciousness and Influence of Architecture and of Business
- More and documented insight

Management
- Long term adjustment of strategy (ATAM one piece in the puzzle)

Reference: “Architecture Reviews @ Bosch”, 1st SEI SATURN, Ferber, 2005
<table>
<thead>
<tr>
<th>DFSS</th>
<th>Week One Topics</th>
<th>Week Two Topics</th>
<th>Week Three Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• DFSS Overview</td>
<td>• Critical Parameter management</td>
<td>• Linear and Multiple Regression</td>
</tr>
<tr>
<td></td>
<td>• CDOV Process</td>
<td>• DFMEA</td>
<td>• RSM</td>
</tr>
<tr>
<td></td>
<td>• DFSS tools and PM</td>
<td>• Basic Stats/Minitab</td>
<td>• Monte Carlo</td>
</tr>
<tr>
<td></td>
<td>• VOC/KJ</td>
<td>• Hypothesis testing</td>
<td>• Robust Design</td>
</tr>
<tr>
<td></td>
<td>• QFD</td>
<td>• Confidence Intervals</td>
<td>• Analytical/Empirical Tolerance Optimization</td>
</tr>
<tr>
<td></td>
<td>• 1st Prin. Modeling (Monte Carlo)</td>
<td>• ANOVA</td>
<td>• CPM</td>
</tr>
<tr>
<td></td>
<td>• Pugh</td>
<td>• MSA</td>
<td>• Architecture and Design-based Software Reliability Modeling</td>
</tr>
<tr>
<td></td>
<td>• DFSS Scorecards</td>
<td>• SPC</td>
<td>• Software Reliability Growth Testing and Modeling</td>
</tr>
<tr>
<td></td>
<td>• Six Sigma and CMMI synergies</td>
<td>• Design and Process</td>
<td>• Motorola Lab’s TRAMS</td>
</tr>
<tr>
<td></td>
<td>• Parametric SW Project Forecasting</td>
<td>• Capability</td>
<td>• Taguchi Noise Testing</td>
</tr>
<tr>
<td></td>
<td>• Reqts Management Processes</td>
<td>• DOE</td>
<td>• Small memory management</td>
</tr>
<tr>
<td></td>
<td>• Developing SW OpnI Profiles</td>
<td>• Full Factorial Designs</td>
<td>• Throughput and timing analysis</td>
</tr>
<tr>
<td></td>
<td>• SW Quality Attribute Workshops</td>
<td>• Fractional Factorial Designs</td>
<td>• Orthogonal Defect Classification</td>
</tr>
<tr>
<td></td>
<td>• Attribute-Driven SW Architecture</td>
<td>• Modeling</td>
<td>• Advanced SW Inspection</td>
</tr>
<tr>
<td></td>
<td>• Active Reviews for Intermediate Designs</td>
<td>• Advanced DOE</td>
<td>• Human Error Analysis</td>
</tr>
<tr>
<td></td>
<td>• SW Architecture Tradeoff Analysis Method (ATAM)</td>
<td></td>
<td>• Cleanroom Software Engineering</td>
</tr>
<tr>
<td></td>
<td>• Cost Benefit Analysis of Architecture Decisions</td>
<td></td>
<td>• Agile/Extreme Programming</td>
</tr>
<tr>
<td></td>
<td>• Software Product Line Planning and Execution</td>
<td></td>
<td>• SEI Personal and Team Software Process and relationships to DFSS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Usability Engineering</td>
</tr>
</tbody>
</table>

Black=as is; Blue=modify for Software Engineering; Green=SW Creation
DFSS

**Week One Topics**

- DFSS Overview
- CDOV Process
- DFSS tools and PM
- VOC/KJ
- QFD
- 1st Prin. Modeling (Monte Carlo)
- Pugh
- DFSS Scorecards
- Six Sigma and CMMI synergies
- Parametric SW Project Forecasting
- Reqs Management processes
- Developing SW Objectives
- SW Quality Attribute Pipelines
- Attribute-Driven SW Attributes
- Active Reviews for SW Designs
- SW Architecture and Analysis Method (AAM)
- Cost Benefit Analysis of Architecture Decisions
- Software Product Line Planning and Execution
- DFSS Overview
- CDOV Process
- DFSS tools and PM
- VOC/KJ
- QFD
- 1st Prin. Modeling (Monte Carlo)
- Pugh
- DFSS Scorecards
- Six Sigma and CMMI synergies
- Parametric SW Project Forecasting
- Reqs Management processes
- Developing SW Objectives
- SW Quality Attribute Pipelines
- Attribute-Driven SW Attributes
- Active Reviews for SW Designs
- SW Architecture and Analysis Method (AAM)
- Cost Benefit Analysis of Architecture Decisions
- Software Product Line Planning and Execution

**Week Three Topics**

- Capture and model schedule uncertainty with Monte Carlo simulation
- Describing customer experiences, behaviors and defining non-stated customer req’ts
- Describe how QFD may be used to prioritize SW features based on product req’ts
- Quantifying SW performance and quality attributes in terms of the software system decomposition (physical or functional)
- Demonstrate use of the Pugh concept selection matrix during architecture decision-making

- Analytical/Empirical Tolerance
- Reliability and Multiple Regression
- FMEA
- Basic Statistical Tools: Hypotheses Testing
- Confidence Intervals
- MSA
- Software Reliability Growth Modeling
- Software Reliability Modeling
- Software Quality Analysis
- Software and Hardware Quality Assurance
- TRAMS
- Software Testing
- Software Quality Management
- Throughput and timing analysis
- Orthogonal Defect Classification
- Advanced SW Inspection
- Human Error Analysis
- Cleanroom Software Engineering
- Agile/Extreme Programming
- SEI Personal and Team Software Process and relationships to DFSS
- Usability Engineering

March 2, 2007

© 2005 by Carnegie Mellon University

page 37
## DFSS

<table>
<thead>
<tr>
<th>Week One Topics</th>
<th>Week Three Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• DFSS Overview</td>
<td>• Linear and Multiple Regression</td>
</tr>
<tr>
<td>• CDOV Process</td>
<td>• RSM</td>
</tr>
<tr>
<td>• DFSS tools and PM</td>
<td>• Analytical/empirical Tolerance Optimization</td>
</tr>
<tr>
<td>• VOI/KJ</td>
<td>• CPM</td>
</tr>
<tr>
<td>• QFD</td>
<td>• Scrum</td>
</tr>
<tr>
<td>• 1st Prin. Modeling (Monte Carlo)</td>
<td>• Six Sigma and CMMI synergies</td>
</tr>
<tr>
<td>• Pugh</td>
<td>• Six Sigma and CMMI synergies</td>
</tr>
<tr>
<td>• DFSS Scorecards</td>
<td>• Parametric SW Project Forecasting</td>
</tr>
<tr>
<td>• Six Sigma and CMMI synergies</td>
<td>• Reqs Management Processes</td>
</tr>
<tr>
<td>• Parametric SW Project Forecasting</td>
<td>• Developing SW Opnl Profiles</td>
</tr>
<tr>
<td>• Reqs Management Processes</td>
<td>• SW Quality Attribute Workshops</td>
</tr>
<tr>
<td>• Developing SW Opnl Profiles</td>
<td>• Attribute-Driven SW Architecture</td>
</tr>
<tr>
<td>• SW Quality Attribute Workshops</td>
<td>• Active Reviews for Intermediate Designs</td>
</tr>
<tr>
<td>• Attribute-Driven SW Architecture</td>
<td>• SW Architecture Tradeoff Analysis Method (ATAM)</td>
</tr>
<tr>
<td>• Active Reviews for Intermediate Designs</td>
<td>• Cost Benefit Analysis of Architecture Decisions</td>
</tr>
<tr>
<td>• SW Architecture Tradeoff Analysis Method (ATAM)</td>
<td>• Software Product Line Planning and Execution</td>
</tr>
<tr>
<td>• Cost Benefit Analysis of Architecture Decisions</td>
<td>• SEI Personal and Team Software Process and relationships to DFSS</td>
</tr>
<tr>
<td>• Software Product Line Planning and Execution</td>
<td>• Usability Engineering</td>
</tr>
<tr>
<td>• Discuss SDFSS scorecard metrics by CDOV phase</td>
<td>• Understand the synergies between SDFSS and the SEI CMMI process areas</td>
</tr>
<tr>
<td>• Understand and define software quality attributes as defined by standard</td>
<td>• Learn what parametric forecasting is and how it can reduce variation in SW</td>
</tr>
<tr>
<td>scenarios format</td>
<td>forecasts</td>
</tr>
<tr>
<td>• Revisit the SW Requirements best practices and metrics defined by the</td>
<td>• Learn what Software Operational Usage Profiles are and why they are important</td>
</tr>
<tr>
<td>Mot Engr Synergy Team</td>
<td>to development and test</td>
</tr>
</tbody>
</table>

March 2, 2007

© 2005 by Carnegie Mellon University
### DFSS

<table>
<thead>
<tr>
<th>Week One Topics</th>
<th>Week Two Topics</th>
<th>Week Three Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• DFSS Overview</td>
<td>• DFSS Overview</td>
<td>• DFSS Overview</td>
</tr>
<tr>
<td>• CDOV Process</td>
<td>• CDOV Process</td>
<td>• CDOV Process</td>
</tr>
<tr>
<td>• DFSS tools and PM</td>
<td>• DFSS tools and PM</td>
<td>• DFSS tools and PM</td>
</tr>
<tr>
<td>• VOC/KJ</td>
<td>• VOC/KJ</td>
<td>• VOC/KJ</td>
</tr>
<tr>
<td>• QFD</td>
<td>• QFD</td>
<td>• QFD</td>
</tr>
<tr>
<td>• 1st Prin. Modeling (Monte Carlo)</td>
<td>• 1st Prin. Modeling (Monte Carlo)</td>
<td>• 1st Prin. Modeling (Monte Carlo)</td>
</tr>
<tr>
<td>• Pugh</td>
<td>• Pugh</td>
<td>• Pugh</td>
</tr>
<tr>
<td>• DFSS Scorecards</td>
<td>• DFSS Scorecards</td>
<td>• DFSS Scorecards</td>
</tr>
<tr>
<td>• Six Sigma and CMMI synergies</td>
<td>• Six Sigma and CMMI synergies</td>
<td>• Six Sigma and CMMI synergies</td>
</tr>
<tr>
<td>• Parametric SW Project Forecasting</td>
<td>• Parametric SW Project Forecasting</td>
<td>• Parametric SW Project Forecasting</td>
</tr>
<tr>
<td>• Reqts Management Processes</td>
<td>• Reqts Management Processes</td>
<td>• Reqts Management Processes</td>
</tr>
<tr>
<td>• Developing SW Opnl Profiles</td>
<td>• Developing SW Opnl Profiles</td>
<td>• Developing SW Opnl Profiles</td>
</tr>
<tr>
<td>• SW Quality Attribute Workshops</td>
<td>• SW Quality Attribute Workshops</td>
<td>• SW Quality Attribute Workshops</td>
</tr>
<tr>
<td>• Attribute-Driven SW Architecture</td>
<td>• Attribute-Driven SW Architecture</td>
<td>• Attribute-Driven SW Architecture</td>
</tr>
<tr>
<td>• Active Reviews for Intermediate Designs</td>
<td>• Active Reviews for Intermediate Designs</td>
<td>• Active Reviews for Intermediate Designs</td>
</tr>
<tr>
<td>• SW Architecture Tradeoff Analysis Method (ATAM)</td>
<td>• SW Architecture Tradeoff Analysis Method (ATAM)</td>
<td>• SW Architecture Tradeoff Analysis Method (ATAM)</td>
</tr>
<tr>
<td>• Cost Benefit Analysis of Architecture Decisions</td>
<td>• Cost Benefit Analysis of Architecture Decisions</td>
<td>• Cost Benefit Analysis of Architecture Decisions</td>
</tr>
<tr>
<td>• Software Product Line Planning and Execution</td>
<td>• Software Product Line Planning and Execution</td>
<td>• Software Product Line Planning and Execution</td>
</tr>
</tbody>
</table>

- **Discuss the nature of Quality Attributes driving most architecture decisions, thereby highlighting the significance of Quality Attribute workshops**
- **Learn a new SEI proposed method for architecture team collaboration on new architecture designs**
- **Learn the SEI approach to conducting Architecture Tradeoff Analysis using stakeholder inputs**
- **Understand and apply the SEI Cost Benefit Analysis methods for architecture decisions; awareness of Motlabb's Decision-Driven Architecture methods and tools**
- **Understand and improve organizational planning to execute proper software product line management using late-breaking SEI methods**
### DFSS

#### Week Two Topics

- Critical Parameter management
- DFMEA
- Basic Stats/Minitab
- Hypothesis testing
- Confidence Intervals
- ANOVA
- MSA
- SPC
- Design and Process
- Capability
- DOE
- Full Factorial Designs
- Fractional Factorial Designs
- Modeling
- Advanced DOE

- Analytical/Empirical Tolerance
- Failure Mode and Effects Analysis
- Software Reliability Growth Modeling and Testing
- Throughput and timing analysis
- Orthogonal Defect Classification

#### Using the flow-down of CTQ’s, identify the software critical parameters measured within the software structure (physically and/or functionally)

- DFSS Scorecards
- Six Sigma and CMMI synopses

#### Learn how to control software processes and critical parameters. Can provide an early indicator of quality issues

- Attribute-Driven SW Architecture
- Active Reviews for Intermediate Designs
- SW Architecture Tradeoff Analysis Method (ATAM)
- Cost Benefit Analysis of Architecture Decisions
- Software Product Line Planning and Execution

#### Use SW Design FMEA to brainstorm what can go wrong and why (apply at Req’ts, Architecture, Design and Code levels)

- Basic statistical tools to enable comparisons of before-after situations or multiple items

#### Understand how much measurement error exists in a software metric before depending on its result

- Learn how to various forms of designed experimentation to quickly decide on algorithm solutions, performance settings, optimal situations. Can be used to structure internal software testing as well as field testing
Learn how to use regression analysis to identify quantitative relationships from historical data. Useful for predictions.

Learn how to use Response Surface Methodology to derive optimal settings amongst multiple variables.

Learn how to use Monte Carlo simulation to ascertain the achievement of CTQs based on expected variation within critical parameters; also to assess risk of shipping known SW faults.

Learn how to use Robust Design to make software behavior more robust of unplanned variation in inputs or environment.
<table>
<thead>
<tr>
<th>Week One Topics</th>
<th>Week Three Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• DFSS Overview</td>
<td>• Linear and Multiple Regression</td>
</tr>
<tr>
<td>• CDOV Process</td>
<td>• RSM</td>
</tr>
<tr>
<td>• DFSS tools and PM</td>
<td>• Monte Carlo</td>
</tr>
<tr>
<td>• VOC/KJ</td>
<td>• Robust Design</td>
</tr>
<tr>
<td>QFD</td>
<td>• Analytical/Empirical Tolerance Optimization</td>
</tr>
<tr>
<td>1st</td>
<td>CPM</td>
</tr>
<tr>
<td>Pugh</td>
<td></td>
</tr>
<tr>
<td>&amp;</td>
<td></td>
</tr>
<tr>
<td>DFSS Scorecards</td>
<td>SPC</td>
</tr>
<tr>
<td>• Sim</td>
<td></td>
</tr>
<tr>
<td>• Pair</td>
<td></td>
</tr>
<tr>
<td>• For</td>
<td></td>
</tr>
<tr>
<td>• Rank</td>
<td></td>
</tr>
<tr>
<td>Developing SW Optm Profiles</td>
<td></td>
</tr>
<tr>
<td>• SW Quality Attribute Workshops</td>
<td></td>
</tr>
<tr>
<td>• Attribute-Driven SW Architecture</td>
<td></td>
</tr>
<tr>
<td>• Active Reviews for Internal Designs</td>
<td></td>
</tr>
<tr>
<td>• SW Architecture Trade-Off Analysis Method (ATA)</td>
<td></td>
</tr>
<tr>
<td>• Cost Benefit Analysis</td>
<td></td>
</tr>
<tr>
<td>• Architecture Decisions</td>
<td></td>
</tr>
<tr>
<td>• Software Product Line Planning and Execution</td>
<td></td>
</tr>
<tr>
<td>• Fractional Factorial Design</td>
<td></td>
</tr>
<tr>
<td>• Modeling</td>
<td></td>
</tr>
<tr>
<td>• Advanced DOE</td>
<td></td>
</tr>
</tbody>
</table>

**Discuss the software critical parameters in which tolerance analysis should be applied**

**Continuing to use critical parameter management throughout the development cycle and thru changes made in SW testing**

For well documented and high reliability systems, implement software reliability modeling based on software architecture and design models. Enables informed predictions early in lifecycle.

Learn how to model software test results during the Test-Analyze-And-Fix cycles to determine test sufficiency and remaining test time to reach the software quality goal.
### DFSS

#### Week One Topics
- DFSS Overview
- CDOV Process
- DFSS tools and PM
- VOC/KJ
- QFD
- 1st Principles
- Pugh Matrix
- DFSS
- Six Sigma and CMMI synergies
- Parametric SW Project Forecasting
- Attribute-Driven SW Architecture
- Active Review
- SW Architecture Analysis
- Cost Benefit Analysis of Architecture Decisions
- Software Product Line and Execution

#### Week Three Topics
- Linear and Multiple Regression
- RSM
- Monte Carlo
- Robust Design
- Analytical/Empirical Tolerance Optimization
- CPM
- Architecture and Design-based Software Reliability Modeling
- Software Reliability Growth Testing and Modeling
- Motorola Lab’s TRAMS
- Taguchi Noise Test
- Small memory management
- Throughput and timing analysis
- Orthogonal Defect Classification
- Advanced SW Inspection
- Human Error Analysis
- Cleanroom Software Engineering
- Agile/Extreme Programming
- SEI Personal and Team Software Process and relationships to DFSS
- Usability Engineering

---

**Learn about the capability and results that the Motorola Labs TRAMS tool offers to system test groups**

**Learn how Taguchi methods may be employed to test software in the presence of uncontrollable “noise” variables**

**Learn about the concepts of small memory patterns and begin an internal sharing of other Motorola-developed patterns**

**Become aware of the tools and techniques to model timing analysis to assess software performance**

**Learn about Orthogonal Defect Classification with statistical footprints to enable early detection of issues**
# DFSS

## Week One Topics

- DFSS Overview
- CDOV Process
- DFSS tools and PM
- VOC/KJ
- QFD
- 1st Prin. Modeling
- Pugh
- DFSS Scorecard
- Six Sigma and Change Management Process
- Parametric SW Process
- Forecasting
- Requirement Engineering
- Factorial Designs
- Developmental Designs
- SW Quality Assurance
- Attributes
- Active
- Defects
- SW Inspection
- Analytical/Empirical Tolerance

## Week Three Topics

- Linear and Multiple Regression
- RSM
- Monte Carlo
- Robust Design
- Analytical/Empirical Tolerance Optimization
- CPM
- Architecture and Design-based Software Reliability Modeling
- Software Reliability Growth Testing and Modeling
- Motorola Lab’s TRAMS
- Taguchi Noise Testing
- Small memory management
- Throughput and timing analysis
- Orthogonal Defect Classification
- Advanced SW Inspection
- Human Error Analysis
- Cleanroom Software Engineering
- Agile/Extreme Programming
- SEI Personal and Team Software Process and relationships to DFSS
- Usability Engineering

---

**Discuss practical aspects of cleanroom software engineering to prevent injection of software faults**

**Learn about the shortcomings of the software inspection process and ideas for sampling to achieve the same result**

**Learn more about the process and conditions for human error and how to anticipate human error occurrences**

**Investigate and discuss the leading-edge practice of Agile methods for a more lean software process**

**Discuss the SEI methodologies for the Team and Personal Software Process and show connections to SDFSS principles**

**Discuss the measurement of software usability**
In Summary

Software DFSS, within a holistic DFSS approach to product development, is coming of age,

Many gaps, in translating traditional DFSS concepts to software engineering, may be solved by the adoption of a number of Software Engineering Institute (SEI) technologies.

Thus, DFSS does not have to be re-invented for Software Engineering!
References

http://www.sei.cmu.edu/tsp/
The Team Software Process (TSP) in Practice: A Summary of Recent Results
CMU/SEI-2003-TR-014 (see also CMU/SEI-2000-TR-015)
Noopur Davis, Julia Mullaney
Relating the Team Software Process (TSP) to the Capability Maturity Model for Software (SW-CMM)
CMU/SEI-2002-TR-008
Noopur Davis, Jim McHale, with Strategy & Overview by Watts Humphrey
The Personal Software Process (PSP)
CMU/SEI-2000-TR-022
The Team Software Process (TSP)
CMU/SEI-2000-TR-023
Watts Humphrey
Information Sources

There are several sources of information and many types of information available to you about CMMI. Sources include

- **The CMMI Web site** (www.sei.cmu.edu/cmmi/)
- **SEI Partner organizations** (www.sei.cmu.edu/collaborating/partners/partners-tech.html)
- **Online forums of CMMI users** (www.sei.cmu.edu/cmmi/adoption/knowledge-exchange.html)
- **Conferences and other events** (www.sei.cmu.edu/cmmi/adoption/cmmi-events.html)
- **Written publications** (www.sei.cmu.edu/cmmi/adoption/adoptions.html)
- **CMMI-related courses** (www.sei.cmu.edu/cmmi/training/training.html)
Contact Information

Robert Stoddard
Software Engineering Institute (Office 3110)
4500 Fifth Avenue
Pittsburgh, PA  15213
412.268.1121
rws@sei.cmu.edu

Or, contact SEI Customer Relations:
Customer Relations
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
FAX: (412) 268-5800
customer-relations@sei.cmu.edu
Questions?