Team Software Process (TSP) In Context

Capers Jones, President

August 15, 2011
This presentation is dedicated to:

- Watts Humphrey, the developer of Team Software Process (TSP)
- Allan Albrecht, the developer of function point metrics

Watts was an industry leader in achieving high quality software.

Al was an industry leader in developing metrics that could quantify high quality software.
Innovations Pioneered by Watts Humphrey

- IBM process assessments 1972
- IBM inspection methods 1972
- SEI inspection methods 1987
- Personal Software Process (PSP) 1995
- Team Software Process (TSP) 2000
Innovations Pioneered by Allan Albrecht

• IBM function point metrics (co inventor) 1975
• IBM backfiring (LOC to function points) 1977
• IBM function point training course 1978
• Co-inventor of feature points 1986
• First function point certification 1987
<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall development</td>
<td>1962</td>
</tr>
<tr>
<td>Structured development</td>
<td>1975</td>
</tr>
<tr>
<td>Object-Oriented development (OO)</td>
<td>1980</td>
</tr>
<tr>
<td>Rapid Application Development (RAD)</td>
<td>1984</td>
</tr>
<tr>
<td>Iterative development</td>
<td>1985</td>
</tr>
<tr>
<td>Rational Unified Process (RUP)</td>
<td>1986</td>
</tr>
<tr>
<td>Agile development</td>
<td>1997</td>
</tr>
<tr>
<td>Personal Software Process (PSP)</td>
<td>2000</td>
</tr>
<tr>
<td>Team Software Process (TSP)</td>
<td>2000</td>
</tr>
</tbody>
</table>
SIGNIFICANT SOFTWARE INNOVATIONS

- Software defect severity scale (IBM) 1956
- Automated change management tools 1967
- High-level programming languages 1969
- Software process assessments (IBM) 1970
- Structured coding 1971
- Design and code inspections (IBM) 1972
- Automated project management tools 1973
- Automated cost and quality estimation (IBM) 1974
SIGNIFICANT SOFTWARE INNOVATIONS

• Function point metrics (IBM) 1975
• Joint application design (JAD) (IBM) 1976
• Backfiring LOC to function points (IBM) 1977
• Software reusability 1979
• Commercial software estimating tools 1980
• Object-oriented programming 1981
• Complexity analysis tools 1985
• SEI capability maturing model (CMM/CMMI) 1985
SIGNIFICANT SOFTWARE INNOVATIONS

• Software development/maintenance workbenches 1986
• Test coverage analysis tools 1990
• Use cases for requirements 1994
• IBM Orthogonal defect classification 1995
• Commercial software benchmarks 1997
• Static analysis tools 1997
• Automated testing tools 1997
• Six-Sigma for Software 2000
• Launch of Wiki-based collaboration 2001
<table>
<thead>
<tr>
<th>Issue</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements &lt; 50% complete</td>
<td>1966</td>
</tr>
<tr>
<td>Requirements change &gt; 2% per month</td>
<td>1975</td>
</tr>
<tr>
<td>Requirements defects resist testing</td>
<td>1976</td>
</tr>
<tr>
<td>Testing &lt; 60% efficient in finding bugs</td>
<td>1978</td>
</tr>
<tr>
<td>Bad fixes &gt; 7% of all defect repairs</td>
<td>1979</td>
</tr>
<tr>
<td>About 5% of modules contain &gt; 50% of defects</td>
<td>1979</td>
</tr>
<tr>
<td>About 35% of large projects are cancelled</td>
<td>1980</td>
</tr>
<tr>
<td>Most estimates are excessively optimistic</td>
<td>1980</td>
</tr>
<tr>
<td>Average defect removal &lt;85% in U.S.</td>
<td>1980</td>
</tr>
</tbody>
</table>
SOFTWARE PROBLEMS HELPED BY TSP

- Requirements < 50% complete 1966
- Requirements change > 2% per month 1975
- Requirements defects resist testing 1976 **
- Testing < 60% efficient in finding bugs 1978 **
- Bad fixes > 7% of all defect repairs 1979 **
- About 5% of modules contain > 50% of defects 1979 **
- About 35% of large projects are cancelled 1980 **
- Most estimates are excessively optimistic 1980 **
- Average defect removal < 85% in U.S. 1980 **
# U.S. AVERAGES FOR SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

<table>
<thead>
<tr>
<th>Defect Origins</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>1.00</td>
<td>77%</td>
<td>0.23</td>
</tr>
<tr>
<td>Design</td>
<td>1.25</td>
<td>85%</td>
<td>0.19</td>
</tr>
<tr>
<td>Coding</td>
<td>1.75</td>
<td>95%</td>
<td>0.09</td>
</tr>
<tr>
<td>Documents</td>
<td>0.60</td>
<td>80%</td>
<td>0.12</td>
</tr>
<tr>
<td>Bad Fixes</td>
<td>0.40</td>
<td>70%</td>
<td>0.12</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>5.00</strong></td>
<td><strong>85%</strong></td>
<td><strong>0.75</strong></td>
</tr>
</tbody>
</table>

(function points show all defect sources - not just coding defects)

(code defects = 35% of total defects)
BEST IN CLASS SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

<table>
<thead>
<tr>
<th>Defect Origins</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>0.40</td>
<td>85%</td>
<td>0.08</td>
</tr>
<tr>
<td>Design</td>
<td>0.60</td>
<td>97%</td>
<td>0.02</td>
</tr>
<tr>
<td>Coding</td>
<td>1.00</td>
<td>99%</td>
<td>0.01</td>
</tr>
<tr>
<td>Documents</td>
<td>0.40</td>
<td>98%</td>
<td>0.01</td>
</tr>
<tr>
<td>Bad Fixes</td>
<td>0.10</td>
<td>95%</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>2.50</strong></td>
<td><strong>96%</strong></td>
<td><strong>0.13</strong></td>
</tr>
</tbody>
</table>

OBSERVATIONS

(Most often found in systems software > SEI CMM Level 3 or in TSP projects)
## POOR SOFTWARE QUALITY - MALPRACTICE

(Data expressed in terms of defects per function point)

<table>
<thead>
<tr>
<th>Defect Origins</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>1.50</td>
<td>50%</td>
<td>0.75</td>
</tr>
<tr>
<td>Design</td>
<td>2.20</td>
<td>50%</td>
<td>1.10</td>
</tr>
<tr>
<td>Coding</td>
<td>2.50</td>
<td>80%</td>
<td>0.50</td>
</tr>
<tr>
<td>Documents</td>
<td>1.00</td>
<td>70%</td>
<td>0.30</td>
</tr>
<tr>
<td>Bad Fixes</td>
<td>0.80</td>
<td>50%</td>
<td>0.40</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>8.00</strong></td>
<td><strong>62%</strong></td>
<td><strong>3.05</strong></td>
</tr>
</tbody>
</table>

**OBSERVATIONS**

(Most often found in large water fall projects > 10,000 Function Points).
## DEFECT POTENTIALS AND REMOVAL EFFICIENCY FOR EACH LEVEL OF SEI CMM

(Data Expressed in Terms of Defects per Function Point
For projects nominally 1000 function points in size)

<table>
<thead>
<tr>
<th>SEI CMM Levels</th>
<th>Defect Potentials</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI CMMI 1</td>
<td>5.25</td>
<td>80%</td>
<td>1.05</td>
</tr>
<tr>
<td>SEI CMMI 2</td>
<td>5.00</td>
<td>85%</td>
<td>0.75</td>
</tr>
<tr>
<td>SEI CMMI 3</td>
<td>4.75</td>
<td>90%</td>
<td>0.48</td>
</tr>
<tr>
<td>SEI CMMI 4</td>
<td>4.50</td>
<td>93%</td>
<td>0.32</td>
</tr>
<tr>
<td>SEI CMMI 5</td>
<td>4.25</td>
<td>96%</td>
<td>0.17</td>
</tr>
</tbody>
</table>
## DEFECT POTENTIALS AND REMOVAL EFFICIENCY FOR EACH LEVEL OF SEI CMM

(Data Expressed in Terms of Defects per Function Point
For projects 10,000 function points in size)

<table>
<thead>
<tr>
<th>SEI CMM Levels</th>
<th>Defect Potentials</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI CMMI 1</td>
<td>6.50</td>
<td>75%</td>
<td>1.63</td>
</tr>
<tr>
<td>SEI CMMI 2</td>
<td>6.25</td>
<td>82%</td>
<td>1.13</td>
</tr>
<tr>
<td>SEI CMMI 3</td>
<td>5.50</td>
<td>87%</td>
<td>0.71</td>
</tr>
<tr>
<td>SEI CMMI 4</td>
<td>5.25</td>
<td>90%</td>
<td>0.53</td>
</tr>
<tr>
<td>SEI CMMI 5</td>
<td>4.75</td>
<td>94%</td>
<td>0.29</td>
</tr>
</tbody>
</table>
**DEFECTS AND SOFTWARE METHODOLOGIES**

(Data Expressed in Terms of Defects per Function Point  
For projects nominally 1000 function points in size)

<table>
<thead>
<tr>
<th>Software methods</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall</td>
<td>5.50</td>
<td>80%</td>
<td>1.10</td>
</tr>
<tr>
<td>Iterative</td>
<td>4.75</td>
<td>87%</td>
<td>0.62</td>
</tr>
<tr>
<td>Object-Oriented</td>
<td>4.50</td>
<td>88%</td>
<td>0.54</td>
</tr>
<tr>
<td>Agile</td>
<td>4.00</td>
<td>90%</td>
<td>0.40</td>
</tr>
<tr>
<td>Rational Unified Process (RUP)</td>
<td>4.25</td>
<td>92%</td>
<td>0.34</td>
</tr>
<tr>
<td>PSP and TSP</td>
<td>3.50</td>
<td>96%</td>
<td>0.14</td>
</tr>
<tr>
<td>Hybrid with 85% certified reuse</td>
<td>1.75</td>
<td>99%</td>
<td>0.02</td>
</tr>
</tbody>
</table>
(Data Expressed in Terms of Defects per Function Point
For projects nominally 10,000 function points in size)

<table>
<thead>
<tr>
<th>Software methods</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall</td>
<td>7.00</td>
<td>75%</td>
<td>1.75</td>
</tr>
<tr>
<td>Iterative</td>
<td>6.25</td>
<td>82%</td>
<td>1.13</td>
</tr>
<tr>
<td>Object-Oriented</td>
<td>5.75</td>
<td>85%</td>
<td>0.86</td>
</tr>
<tr>
<td>Agile</td>
<td>5.50</td>
<td>87%</td>
<td>0.72</td>
</tr>
<tr>
<td>Rational Unified Process (RUP)</td>
<td>5.50</td>
<td>90%</td>
<td>0.55</td>
</tr>
<tr>
<td>PSP and TSP</td>
<td>5.00</td>
<td>95%</td>
<td>0.25</td>
</tr>
<tr>
<td>Hybrid with 85% certified reuse</td>
<td>2.25</td>
<td>96%</td>
<td>0.09</td>
</tr>
</tbody>
</table>
QUALITY PREDICTION RULES OF THUMB

RAISE APPLICATION SIZE IN FUNCTION POINTS TO THESE POWERS TO PREDICT SOFTWARE DEFECT POTENTIALS

(Requirements, design, code, documents, and bad fixes)

<table>
<thead>
<tr>
<th>Method</th>
<th>Power</th>
<th>Size</th>
<th>Defects</th>
<th>Defects per Function Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall</td>
<td>1.24</td>
<td>1000</td>
<td>5,248</td>
<td>5.25</td>
</tr>
<tr>
<td>Agile</td>
<td>1.20</td>
<td>1000</td>
<td>3,981</td>
<td>3.98</td>
</tr>
<tr>
<td>RUP</td>
<td>1.21</td>
<td>1000</td>
<td>4,266</td>
<td>4.27</td>
</tr>
<tr>
<td>TSP</td>
<td>1.18</td>
<td>1000</td>
<td>3,467</td>
<td>3.47</td>
</tr>
</tbody>
</table>
MOVING TO EXCELLENCE IN SOFTWARE ENGINEERING

• Start with an assessment and baseline to find out what is right and wrong with current practices.

• Commission a benchmark study to compare your performance with best practices in your industry.

• Stop doing what is wrong.

• Do more of what is right.

• Set targets: *Best in Class* *****, Better than Average****, Better than Today***.

• Develop a three-year technology plan.

• Include: capital equipment, offices, tools, methods, education, culture, languages and return on investment (ROI).
## TECHNICAL REASONS FOR SOFTWARE FAILURES

<table>
<thead>
<tr>
<th>Unsuccessful Projects</th>
<th>Successful Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inappropriate methodologies</td>
<td>Optimal methodologies</td>
</tr>
<tr>
<td>No automated sizing tools</td>
<td>Automated sizing tools</td>
</tr>
<tr>
<td>No automated estimation tools</td>
<td>Automated estimation tools</td>
</tr>
<tr>
<td>No automated planning tools</td>
<td>Automated planning tools</td>
</tr>
<tr>
<td>No progress reporting</td>
<td>Accurate progress reporting</td>
</tr>
<tr>
<td>Inaccurate cost collection</td>
<td>Accurate cost collection</td>
</tr>
<tr>
<td>No measurement data</td>
<td>Substantial measurement data</td>
</tr>
<tr>
<td>Inaccurate metrics</td>
<td>Accurate metrics</td>
</tr>
<tr>
<td>No design reviews</td>
<td>Formal design reviews</td>
</tr>
<tr>
<td>No code inspections</td>
<td>Formal code inspections</td>
</tr>
<tr>
<td>No defect tracking</td>
<td>Formal defect tracking</td>
</tr>
<tr>
<td>Informal change control</td>
<td>Formal change control</td>
</tr>
<tr>
<td>Unstable requirements (&gt;30%)</td>
<td>Stable requirements (&lt; 10%)</td>
</tr>
</tbody>
</table>

Copyright © 2011 by Capers Jones. All Rights Reserved.
### SOCIAL REASONS FOR SOFTWARE FAILURES

<table>
<thead>
<tr>
<th>Unsuccessful Projects</th>
<th>Successful Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excessive schedule pressure</td>
<td>Realistic schedule expectation</td>
</tr>
<tr>
<td>Severe friction with clients</td>
<td>Cooperation with clients</td>
</tr>
<tr>
<td>Poor communications</td>
<td>Good communications</td>
</tr>
<tr>
<td>Divisive politics</td>
<td>Politics held in check</td>
</tr>
<tr>
<td>Naive senior executives</td>
<td>Experienced senior executives</td>
</tr>
<tr>
<td>Management malpractice</td>
<td>Capable management</td>
</tr>
<tr>
<td>Technical malpractice</td>
<td>Capable technical staff</td>
</tr>
<tr>
<td>Untrained Generalists</td>
<td>Trained Specialists</td>
</tr>
</tbody>
</table>

- Quality Assurance
- Testing
- Planning and Estimating
OTHER CORRELATIONS WITH SOFTWARE FAILURES

**Intermittent Failure Factors**

- Geographic separation of team with inadequate communication
- Multiple sub-contractors involved with inadequate communication
- Extraordinary storage or timing constraints
- Projects using “low bid” as sole contract criterion
- Staffing build up > 15% per month
- Staff attrition > 40% of project team
- Abrupt introduction of new technologies
- Projects by companies that are downsizing
- New executives replace proven methods with latest fads
- Trained personnel retire or change jobs
## U.S. SOFTWARE PERFORMANCE LEVELS

<table>
<thead>
<tr>
<th>PROJECT MANAGEMENT</th>
<th>TECHNICAL STAFFS</th>
<th>SOFTWARE USERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sizing</td>
<td>Fair</td>
<td>Requirements</td>
</tr>
<tr>
<td>Estimating</td>
<td>Poor</td>
<td>Design</td>
</tr>
<tr>
<td>Planning</td>
<td>Fair</td>
<td>Coding</td>
</tr>
<tr>
<td>Tracking</td>
<td>Poor</td>
<td>Reviews</td>
</tr>
<tr>
<td>Measuring</td>
<td>Poor</td>
<td>Testing</td>
</tr>
<tr>
<td>Overall</td>
<td>Poor</td>
<td>Good</td>
</tr>
</tbody>
</table>

**Conclusion:** U. S. technical skills are better than U. S. management skills. Project management and quality are frequent problem areas.
**Function points raised to the 0.3 power is the average schedule plan**

**Function points raised to the 0.4 power is the average schedule result**

*(TSP = 0.38 power)*
Web applets and Agile projects < 250 words per function point in plans, specifications

Less Paperwork Than Expected

Military software > 4000 words per function point in plans, specs, and manuals

More Paperwork Than Expected

Total Volume of Pages Produced (Requirements, Design, Documentation)
RISK OF PROJECT FAILURE

- **SEI CMM 1** > 40% failures
- **SEI CMM 3** < 15% failures
- **SEI CMM 5** < 3% failures
- **Agile** < 15% failures
- **TSP/PSP** < 3% failures

*for 10,000 function point projects*
# RISKS OF FAILURE OR DELAY BY CMM LEVEL

(Complex projects of 10,000 function points in size)

<table>
<thead>
<tr>
<th>SEI CMM LEVEL</th>
<th>Delay &gt; 1 year</th>
<th>Termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI CMMI Level 1</td>
<td>35%</td>
<td>40%</td>
</tr>
<tr>
<td>SEI CMMI Level 2</td>
<td>30%</td>
<td>30%</td>
</tr>
<tr>
<td>SEI CMMI Level 3</td>
<td>20%</td>
<td>12%</td>
</tr>
<tr>
<td>SEI CMMI Level 4</td>
<td>12%</td>
<td>04%</td>
</tr>
<tr>
<td>SEI CMMI Level 5</td>
<td>08%</td>
<td>02%</td>
</tr>
<tr>
<td>SEI CMMI Level 5 + TSP</td>
<td>05%</td>
<td>02%</td>
</tr>
</tbody>
</table>
SOFTWARE LIFE EXPECTANCY

After 10 years compilers may not be available

After 5 years restructuring and complexity analysis are needed
Expect about 7% per year new and changed features after the first release.
AVERAGE PRODUCTIVITY RATES (NEW PROJECTS)

Maximum productivity
> 100 function points per staff month

Average productivity
is 8 -12 function points per staff month

Minimum productivity
< 0.5 function points per staff month

Function Points per Staff Month

Application Size in Function Points
PRODUCTIVITY RATES FOR ENHANCEMENT SOFTWARE PROJECTS

New features for existing applications

Overhead of base application

Major structural changes

Application Size in Function Points

Function Points per Staff Month

Copyright © 2011 by Capers Jones. All Rights Reserved.
**PRODUCTIVITY RATES (OVERALL AVERAGE)**

- **Month**: 7, 8, 9, 10
- **Ints per Staff**: 15, 16, 15, 11, 12, 13, 14

---

**Application Size in Function Points**

- Small Enhancements
- Mid-sized projects
- Massive new projects

---

Copyright © 2011 by Capers Jones. All Rights Reserved.
SOFTWARE QUALITY IMPROVEMENT

Defects per FP

SEI CMM 1

SEI CMM 3

PSP/TSP & SEI CMM 5

Defect Removal Efficiency

U.S. Average

Best in Class

Malpractice

Copyright © 2011 by Capers Jones. All Rights Reserved.
SEVEN STAGES OF SOFTWARE EXCELLENCE

Stage 0: Assessment, Baseline, Benchmark analysis

Stage 1: Focus on Project Management

Stage 2: Focus on Development and Maintenance Methods

Stage 3: Focus on New Tools and Approaches

Stage 4: Focus on Infrastructure

Stage 5: Focus on Reusability

Stage 6: Focus on Industry Leadership

Stage 7: Focus on continuous improvement forever!
## TIME REQUIRED TO ADVANCE FROM STAGE TO STAGE

(Duration in Calendar Months)

<table>
<thead>
<tr>
<th>Enterprise Software Population</th>
<th>&lt;10</th>
<th>11-100</th>
<th>101-1000</th>
<th>&gt;1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 0\ Assessment/Baseline</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Stage 1\ Management</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Stage 2\ Methods</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Stage 3\ Tools</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Stage 4\ Infrastructure</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Stage 5\ Reusability</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>Stage 6\ Leadership</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
<td>36</td>
<td>48</td>
<td>60</td>
</tr>
</tbody>
</table>
## THE QUALITY AND PRODUCTIVITY BENEFITS FROM COMPLETING EACH STAGE

<table>
<thead>
<tr>
<th>Stage</th>
<th>Defect Reduction</th>
<th>Productivity Increase</th>
<th>Schedule Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 0 Assessment</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stage 1 Management</td>
<td>-10%</td>
<td>0</td>
<td>-10%</td>
</tr>
<tr>
<td>Stage 2 Methods</td>
<td>-50%</td>
<td>25%</td>
<td>-15%</td>
</tr>
<tr>
<td>Stage 3 Tools</td>
<td>-10%</td>
<td>35%</td>
<td>-15%</td>
</tr>
<tr>
<td>Stage 4 Infrastructure</td>
<td>-5%</td>
<td>10%</td>
<td>-5%</td>
</tr>
<tr>
<td>Stage 5 Reusability</td>
<td>-85%</td>
<td>65%</td>
<td>-50%</td>
</tr>
<tr>
<td>Stage 6 Leadership</td>
<td>-5%</td>
<td>5%</td>
<td>-5%</td>
</tr>
<tr>
<td>Overall Results</td>
<td>-90%</td>
<td>350%</td>
<td>-70%</td>
</tr>
</tbody>
</table>
## PROCESS IMPROVEMENT EXPENSES PER CAPITA

<table>
<thead>
<tr>
<th>Stage</th>
<th>Small &lt; 100 staff</th>
<th>Medium 100-1000</th>
<th>Large &gt; 1000 staff</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$125</td>
<td>$150</td>
<td>$250</td>
</tr>
<tr>
<td>Assessment</td>
<td></td>
<td></td>
<td><strong>SEI CMM 1</strong></td>
</tr>
<tr>
<td>1</td>
<td>$1000</td>
<td>$2500</td>
<td>$3000</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td><strong>SEI CMM 2</strong></td>
</tr>
<tr>
<td>2</td>
<td>$1500</td>
<td>$2500</td>
<td>$3500</td>
</tr>
<tr>
<td>Methods</td>
<td></td>
<td></td>
<td><strong>PSP/TSP</strong></td>
</tr>
<tr>
<td>3</td>
<td>$2500</td>
<td>$3500</td>
<td>$5000</td>
</tr>
<tr>
<td>Tools</td>
<td></td>
<td></td>
<td><strong>SEI CMM 3</strong></td>
</tr>
<tr>
<td>4</td>
<td>$1500</td>
<td>$2000</td>
<td>$3000</td>
</tr>
<tr>
<td>Infrastructure</td>
<td></td>
<td></td>
<td><strong>SEI CMM 4</strong></td>
</tr>
<tr>
<td>5</td>
<td>$2000</td>
<td>$2500</td>
<td>$3500</td>
</tr>
<tr>
<td>Reusability</td>
<td></td>
<td></td>
<td><strong>SEI CMM 5</strong></td>
</tr>
<tr>
<td>6</td>
<td>$1000</td>
<td>$1000</td>
<td>$2000</td>
</tr>
<tr>
<td>Leadership</td>
<td></td>
<td></td>
<td><strong>SEI CMM 5</strong></td>
</tr>
<tr>
<td>Overall Results</td>
<td>$9625</td>
<td>$14150</td>
<td>$20250</td>
</tr>
</tbody>
</table>

Copyright © 2011 by Capers Jones. All Rights Reserved.
RATES OF PROCESS IMPROVEMENT CORRELATED TO INITIAL RANKING

Excellent 1
Good 2
Average 3
Mediocre 4
Poor 5

Year 1 Year 2 Year 3 Year 4 Year 5
BEST CASE RETURN ON INVESTMENT (ROI)

- Assume improvement costs of about $1,000,000
- Value of better quality $5,000,000
- Value of shorter schedules $4,000,000
- Value of higher productivity $3,000,000
- Value of reduced maintenance $2,000,000
- Value of better customer satisfaction $5,000,000
- TOTAL VALUE $20,000,000 *
- RETURN ON INVESTMENT $20 to $1

* Assumes 2 years of improvements and 3 years of results
UNSUCCESSFUL PROCESS IMPROVEMENT

- Assume improvement costs of about $1,000,000
- Value of better quality $100,000
- Value of shorter schedules $100,000
- Value of higher productivity $100,000
- Value of reduced maintenance $100,000
- Value of better customer satisfaction $100,000
- TOTAL VALUE $500,000 *
- RETURN ON INVESTMENT $0.5 to $1

* Assumes 2 years of improvements and 3 years of results
STAGE 0: ASSESSMENT, BASELINE, BENCHMARKS

Key Technologies

- SEI Assessment (Levels 1 through 5)
- Six-Sigma Baseline, Benchmark
- SPR Assessment, Baseline, Benchmark
- ISO 9001 - 9004 Audit
- TickIT assessment
- Putnam Baseline, Benchmark
- Gartner Baseline, Benchmark
- David’s Baseline, Benchmark
- IFPUG Baseline, Benchmark
- ISBSG Benchmarks (commercially available)
STAGE 1: FOCUS ON PROJECT MANAGEMENT

Key Technologies

> Project Sizing
> Project Schedule Planning
> Project Cost Estimating
> Project Quality Estimating
> Functional Metrics
> Project Measurement
> Project Milestone Tracking
> Package Acquisition
> Risk Analysis
> Value Analysis
STAGE 2: FOCUS ON DEVELOPMENT PROCESSES

Key Technologies

- Early sizing and risk assessment
- Reviews and Inspections
- Automated static analysis
- Joint Application Design (JAD)
- Quality Function Deployment (QFD)
- Six-Sigma methodology
- Team Software Process (TSP)
- Personal Software Process (PSP)
- Rational Unified Process (RUP)
- Agile or XP methodologies
- ISO 9001 - 9004 Certification with caution
- SEI maturity levels (CMM and CMMI)
- Geriatric technologies for legacy systems
STAGE 3: FOCUS ON NEW TOOLS & APPROACHES

- Key Technologies -- New Tools
  - Integrated tool suites
  - Web and Internet Tools
  - Requirements analysis tools
  - Requirements validation tools
  - Static analysis; inspections and automated tools
  - Automated testing tools
  - Reverse Engineering and maintenance tools
STAGE 4: FOCUS ON INFRASTRUCTURE

Key Technologies

- Staff Specialization
- Formal Measurement Organization
- Formal Maintenance Organization
- Formal Quality Assurance Organization
- Formal Testing Organization
- Formal Process Improvement Organization
- Improved Hiring Practices
- Improved Compensation Plans
- Competitive Analysis
- Outsource Analysis
## STAGE 5: FOCUS ON REUSABILITY

### Key Technologies

- **Reusable Architectures**
- **Reusable Requirements**
- **Reusable Designs**
- **Reusable Interfaces**
- **Reusable Source Code**
- **Reusable Plans**
- **Reusable Estimates**
- **Reusable Data**
- **Reusable Human Interfaces**
- **Reusable Test Plans**
- **Reusable Test Cases**
- **Reusable Documentation**

High quality reuse has best ROI of any technology:

> $40 per $1 expended.

Low quality reuse has worst ROI of any technology:

> - $15 for every $1 expended.
STAGE 6: FOCUS ON INDUSTRY LEADERSHIP

Key Technologies

- Baldrige Award
- Deming Prize
- SEI CMMI Level 5 for major software sites
- Best 100 Companies to Work For
- Market share grows > 20% from baseline
- Time to market better than competitors by > 30%
- Acquisition of Competitors
- Become a Software Outsourcer
STAGE 7: Keeping Excellence After Achieving Excellence

Key Technologies

> Measure results of every project
> Produce monthly reports for managers and teams
> Produce annual reports for top executives
> Publicize results to clients and media
> Train new hires in best practices
> Inform new executives of best practices!!
> Insist on best practices with contractors
> Set targets for annual improvements every year
> Do not abandon success once it achieved!!
ATTRIBUTES OF BEST IN CLASS COMPANIES

1. Good project management
2. Good technical staffs
3. Good support staffs
4. Good measurements
5. Good organization structures
6. Good methodologies
7. Good tool suites
8. Good environments
GOOD PROJECT MANAGEMENT

• Without good project management the rest is unachievable

• Attributes of project good management:
  – Fairness to staff
  – Desire to be excellent
  – Strong customer orientation
  – Strong people orientation
  – Strong technology orientation
  – Understands planning and estimating tools
  – Can defend accurate estimates to clients and executives
  – Can justify investments in tools and processes
GOOD SOFTWARE ENGINEERING TECHNICAL STAFFS

• Without good engineering technical staffs tools are not effective

• Attributes of good technical staffs:
  – Desire to be excellent
  – Good knowledge of applications
  – Good knowledge of development processes
  – Good knowledge of quality and defect removal methods
  – Good knowledge of maintenance methods
  – Good knowledge of programming languages
  – Good knowledge of software engineering tools
  – Like to stay at the leading edge of software engineering
GOOD SUPPORT STAFFS

- Without good support technical staffs and managers are handicapped
- Support staffs > 30% of software personnel in leading companies
- Attributes of good support staffs:
  - Planning and estimating skills
  - Measurement and metric skills
  - Writing/communication skills
  - Quality assurance skills
  - Data base skills
  - Network, internet, and web skills
  - Graphics and web-design skills
  - Testing and integration skills
  - Configuration control and change management skills
GOOD SOFTWARE MEASUREMENTS

• Without good measurements progress is unlikely

• Attributes of good measurements:
  – Function point analysis of entire portfolio
  – Annual function point benchmarks
  – Life-cycle quality measures
  – User satisfaction measures
  – Development and maintenance productivity measures
  – Soft factor assessment measures
  – Hard factor measures of costs, staffing, effort, schedules
  – Measurements used as management tools
GOOD ORGANIZATION STRUCTURES

• Without good organization structures progress is unlikely

• Attributes of good organization structures:
  – Balance of line and staff functions
  – Balance of centralized and decentralized functions
  – Organizations are planned
  – Organizations are dynamic
  – Effective use of specialists for key functions
  – Able to integrate “virtual teams” at remote locations
  – Able to integrate telecommuting
GOOD PROCESSES AND METHODOLOGIES

• Without good processes and methodologies tools are ineffective

• Attributes of good methodologies:
  – Flexible and useful for both new projects and updates
  – Scalable from small projects up to major systems
  – Versatile and able to handle multiple kinds of software
  – Efficient and cost effective
  – Evolutionary and able to handle new kinds of projects
  – Unobtrusive and not viewed as bureaucratic
  – Transferable to new hires, contractors, consultants
GOOD TOOL SUITES

• Without good tool suites, management and staffs are handicapped

• Attributes of good tool suites:
  – Both project management and technical tools
  – Quality tools (static analysis; testing, etc. are critical)
  – Functionally complete
  – Mutually compatible
  – Easy to learn
  – Easy to use
  – Tolerant of user errors
  – Secure
GOOD ENVIRONMENTS AND ERGONOMICS

• Without good office environments productivity is difficult

• Attributes of good environments and ergonomics:
  – Private office space for knowledge workers
    (> 90 square feet; > 6 square meters)
  – Avoid small or crowded cubicles with 3 or more staff
  – Adequate conference and classroom facilities
  – Excellent internet and intranet communications
  – Excellent communication with users and clients
MOST EFFECTIVE PROCESS IMPROVEMENT METHODS

1. Defect removal efficiency measurements
2. Function point productivity and quality measurements
3. Automated static analysis (C, Java, COBOL, SQL etc.)
4. Formal design and code inspections
5. Early sizing and early risk assessments
6. Joint Application Design (JAD) for requirements
7. Automated project management tools
8. Automated cost estimating tools
9. Automated complexity analysis and reduction tools
10. Automated change control tools
11. CMMI, TSP and PSP, RUP
12. Six-Sigma for software
# BEST METHODS BY SIZE PLATEAU

<table>
<thead>
<tr>
<th>Function Points</th>
<th>Best Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>PSP, Agile, XP</td>
</tr>
<tr>
<td>100</td>
<td>PSP, Agile, XP</td>
</tr>
<tr>
<td>1,000</td>
<td>RUP, TSP, XP, Agile</td>
</tr>
<tr>
<td>10,000</td>
<td>TSP, RUP</td>
</tr>
<tr>
<td>100,000</td>
<td>TSP, RUP</td>
</tr>
<tr>
<td>Function Points</td>
<td>CMMI LEVEL</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>10</td>
<td>1 through 5</td>
</tr>
<tr>
<td>100</td>
<td>1 through 5</td>
</tr>
<tr>
<td>1,000</td>
<td>3 through 5</td>
</tr>
<tr>
<td>10,000</td>
<td>3 through 5</td>
</tr>
<tr>
<td>100,000</td>
<td>5 only</td>
</tr>
</tbody>
</table>
SOFTWARE IMPROVEMENT GUIDELINES

**DO**
- Think long range: 3 to 5 years
- Consider all factors:
  - Management
  - Process
  - Tools
  - Organization
  - Skills and training
  - Programming Languages
  - Environment
- Plan expenses of up to $15,000 per staff member
- Consider your corporate culture

**DON’T**
- Expect immediate results
- Concentrate only on Agile methods or any other “silver bullet”
- Expect major improvements for minor expenses
- Ignore resistance to change
REFERENCES TO PROCESS IMPROVEMENT


Humphrey, Watts; Managing the Software Process; Addison Wesley, 1989.

Jones, Capers; Assessments, Benchmarks, and Best Practices; Addison Wesley, 2000.


Jones, Capers & Bonsignour, Olivier; The Economics of Software Quality; Addison Wesley, 2011.


Wiegers, Karl; Creating a Software Engineering Culture; Dorset House, 1996.

Web sites:

IFPUG.ORG ITMPI.ORG ISBSG.ORG ASQ.ORG SEI.CMU.EDU, IEEE.ORG