The Economics of Software Process Improvement

Capers Jones, Chief Scientist Emeritus

November 21, 2005
PREVIEW OF 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 enterprise 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
SOFTWARE PROCESS IMPROVEMENT ISSUES

• What are the benefits of a process improvement program?

• What are the risks of a process improvement program?

• How much can development productivity be improved?

• How much can maintenance productivity be improved?

• How much can schedules be improved?

• How much can quality be improved?

• How much will the improvements cost?

• How long will the improvements take?

• What is the ROI of a successful process improvement program?
PROCESS IMPROVEMENT BENEFITS AND RISKS

Benefits:

• Better customer satisfaction

• Better staff morale

• Reduced risk of project failures

Risks:

• Spending money without achieving positive results

• Selecting methods that don’t work for all projects

• Expecting major improvements in one year
PROCESS IMPROVEMENT RANGES

- Development productivity: 15% to 50% *
- Maintenance productivity: 25% to 60% *
- Schedule reductions: 10% to 70% *
- Quality of software: 25% to 90%
- Customer satisfaction: 15% to 30%
- Staff morale: 10% to 25%

* High levels of reuse are required for maximum benefits
SOFTWARE REUSE

• Reuse provides less than 20% of software functions in 2005.

• As of 2005 reuse problems sometimes outweigh value.

• Some reuse problems are Federal: taxes on reused materials.

• By 2015 reuse value should outweigh problems.

• Major sources of reusable components circa 2005:
  
  Purchased Packages  
  Legacy In-house applications  
  Custom applications  
  Commercial reusable components  
  Objects  
  Mergers and acquisitions
MAJOR COMPONENT REUSE PROBLEMS

• Lack of standard interfaces

• Random update schedules by vendors

• Marginal quality of software components and products

• Regression of past functions

• Steep learning curve of object-oriented techniques

• Lack of reusable test plans and test suites

• Potential tax liabilities under IRS rule 401
## HOW IMPORTANT IS SOFTWARE PROJECT MANAGEMENT?

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>SUCCESSFUL PROJECTS</th>
<th>CANCELED PROJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sizing</td>
<td>Good</td>
<td>Poor</td>
</tr>
<tr>
<td>Planning</td>
<td>Very Good</td>
<td>Very Poor</td>
</tr>
<tr>
<td>Estimating</td>
<td>Very Good</td>
<td>Very Poor</td>
</tr>
<tr>
<td>Milestone tracking</td>
<td>Good</td>
<td>Very Poor</td>
</tr>
<tr>
<td>Measurement</td>
<td>Good</td>
<td>Very Poor</td>
</tr>
<tr>
<td>Change control</td>
<td>Excellent</td>
<td>Poor</td>
</tr>
<tr>
<td>Quality</td>
<td>Excellent</td>
<td>Very Poor</td>
</tr>
<tr>
<td>Risk Analysis</td>
<td>Good</td>
<td>Very Poor</td>
</tr>
<tr>
<td>Overall</td>
<td>Very Good</td>
<td>Very Poor</td>
</tr>
</tbody>
</table>
## TECHNICAL REASONS FOR SOFTWARE FAILURES

<table>
<thead>
<tr>
<th>Unsuccessful Projects</th>
<th>Successful Projects</th>
</tr>
</thead>
<tbody>
<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>
SOCIAL REASONS FOR SOFTWARE FAILURES

Unsuccessful Projects
- Excessive schedule pressure
- Severe friction with clients
- Poor communications
- Divisive politics
- Naive senior executives
- Management malpractice
- Technical malpractice
- Untrained Generalists

Successful Projects
- Realistic schedule expectation
- Cooperation with clients
- Good communications
- Politics held in check
- Experienced senior executives
- Capable management
- Capable technical staff
- Trained Specialists

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
IMPROVING SOFTWARE PRODUCTIVITY AND QUALITY

- 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).
QUANTITATIVE AND QUALITATIVE GOALS

What It Means to be *Best In Class*

1. Software project cancellation due to major overruns = zero
2. Software cost overruns < 5% compared to formal budgets
3. Software schedule overruns < 3% compared to formal plans
4. Development productivity > 25 function points per staff month
5. Software reuse of design, code and test cases averages > 75%
6. Development cost < $500 per function point at delivery
7. Development schedules average 25% shorter than industry average
QUANTITATIVE AND QUALITATIVE GOALS (cont.)

8. Software defect potentials average < 2.5 per function point
9. Software defect removal efficiency averages > 96% for all projects
10. Software delivered defects average < 0.1 per function point
11. Software maintenance assignment scopes > 3,500 function points
12. Annual software maintenance < $75 per function point
13. Customer service: Best of any similar corporation
14. User satisfaction: Highest of any similar corporation
15. Staff morale: Highest of any similar corporation
16. Compensation and benefits: Best in your industry
## U.S. SOFTWARE PERFORMANCE LEVELS

<table>
<thead>
<tr>
<th></th>
<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>
<td>Fair Requirements</td>
</tr>
<tr>
<td>Estimating</td>
<td>Poor</td>
<td>Design</td>
<td>Good Schedule Demands</td>
</tr>
<tr>
<td>Planning</td>
<td>Fair</td>
<td>Coding</td>
<td>Good Reviews</td>
</tr>
<tr>
<td>Tracking</td>
<td>Poor</td>
<td>Reviews</td>
<td>Fair Acceptance Test</td>
</tr>
<tr>
<td>Measuring</td>
<td>Poor</td>
<td>Testing</td>
<td>Good Usage</td>
</tr>
</tbody>
</table>

Overall                      | Poor               | Good             | Fair           |

**Conclusion:** U. S. technical skills are better than U. S. management skills. Project management and quality are frequent problem areas.
**RESULTS ON >10,000 FUNCTION POINT PROJECTS**

## Worst-case Scenario

<table>
<thead>
<tr>
<th></th>
<th>Probability of Selected Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual estimates</td>
<td>40%</td>
</tr>
<tr>
<td>Manual plans</td>
<td>45%</td>
</tr>
<tr>
<td>Informal tracking</td>
<td>15%</td>
</tr>
<tr>
<td>Minimal quality control</td>
<td>0%</td>
</tr>
</tbody>
</table>

**Litigation probability > 50% if project is done under contract.**
RESULTS ON >10,000 FUNCTION POINT PROJECTS

<table>
<thead>
<tr>
<th>Single-factor Scenarios</th>
<th>Probability of Selected Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cancel</td>
</tr>
<tr>
<td>2) Manual estimates</td>
<td></td>
</tr>
<tr>
<td>Automated plans</td>
<td>37%</td>
</tr>
<tr>
<td>Informal tracking</td>
<td></td>
</tr>
<tr>
<td>Minimal quality control</td>
<td></td>
</tr>
<tr>
<td>3) Manual estimates</td>
<td>35%</td>
</tr>
<tr>
<td>Manual plans</td>
<td></td>
</tr>
<tr>
<td>Formal tracking</td>
<td></td>
</tr>
<tr>
<td>Minimal quality control</td>
<td></td>
</tr>
<tr>
<td>4) Automated estimates</td>
<td>33%</td>
</tr>
<tr>
<td>Manual plans</td>
<td></td>
</tr>
<tr>
<td>Informal tracking</td>
<td></td>
</tr>
<tr>
<td>Minimal quality control</td>
<td></td>
</tr>
<tr>
<td>5) Manual estimates</td>
<td>30%</td>
</tr>
<tr>
<td>Manual plans</td>
<td></td>
</tr>
<tr>
<td>Informal tracking</td>
<td></td>
</tr>
<tr>
<td>Optimal quality control</td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2005 by Capers Jones. All Rights Reserved.
## RESULTS ON >10,000 FUNCTION POINT PROJECTS

### Two-factor Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Probability of Selected Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cancel</td>
</tr>
<tr>
<td>6) Manual estimates</td>
<td>27%</td>
</tr>
<tr>
<td>Automated plans</td>
<td></td>
</tr>
<tr>
<td>Formal tracking</td>
<td></td>
</tr>
<tr>
<td>Minimal quality control</td>
<td></td>
</tr>
<tr>
<td>7) Automated estimates</td>
<td>23%</td>
</tr>
<tr>
<td>Automated plans</td>
<td></td>
</tr>
<tr>
<td>Informal tracking</td>
<td></td>
</tr>
<tr>
<td>Minimal quality control</td>
<td></td>
</tr>
<tr>
<td>8) Automated estimates</td>
<td>20%</td>
</tr>
<tr>
<td>Manual plans</td>
<td></td>
</tr>
<tr>
<td>Formal tracking</td>
<td></td>
</tr>
<tr>
<td>Minimal quality control</td>
<td></td>
</tr>
</tbody>
</table>
# RESULTS ON >10,000 FUNCTION POINT PROJECTS

## Two-factor Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Probability of Selected Outcomes</th>
<th>Cancel</th>
<th>Delays</th>
<th>On time</th>
<th>Early</th>
</tr>
</thead>
<tbody>
<tr>
<td>9) Manual estimates Automated plans Informal tracking Optimal quality control</td>
<td>18%</td>
<td>20%</td>
<td>54%</td>
<td>8%</td>
<td></td>
</tr>
<tr>
<td>10) Manual estimates Manual plans Formal tracking Optimal quality control</td>
<td>16%</td>
<td>17%</td>
<td>58%</td>
<td>9%</td>
<td></td>
</tr>
<tr>
<td>11) Automated estimates Manual plans Informal tracking Optimal quality control</td>
<td>13%</td>
<td>15%</td>
<td>62%</td>
<td>10%</td>
<td></td>
</tr>
</tbody>
</table>
### RESULTS ON >10,000 FUNCTION POINT PROJECTS

#### Three-factor Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Automated estimates</th>
<th>Automated plans</th>
<th>Formal tracking</th>
<th>Minimal quality control</th>
<th>Probability of Selected Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>12)</td>
<td>10%</td>
<td>12%</td>
<td>67%</td>
<td>11%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Automated estimates</td>
<td>Automated plans</td>
<td>Formal tracking</td>
<td>Minimal quality control</td>
<td>Probability of Selected Outcomes</td>
</tr>
<tr>
<td>13)</td>
<td>8%</td>
<td>10%</td>
<td>69%</td>
<td>13%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manual estimates</td>
<td>Automated plans</td>
<td>Formal tracking</td>
<td>Optimal quality control</td>
<td>Probability of Selected Outcomes</td>
</tr>
<tr>
<td>14)</td>
<td>5%</td>
<td>8%</td>
<td>72%</td>
<td>15%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Automated estimates</td>
<td>Manual plans</td>
<td>Formal tracking</td>
<td>Optimal quality control</td>
<td>Probability of Selected Outcomes</td>
</tr>
<tr>
<td>15)</td>
<td>3%</td>
<td>6%</td>
<td>74%</td>
<td>17%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Automated estimates</td>
<td>Automated plans</td>
<td>Manual tracking</td>
<td>Optimal quality control</td>
<td>Probability of Selected Outcomes</td>
</tr>
</tbody>
</table>
### RESULTS ON >10,000 FUNCTION POINT PROJECTS

#### Best-case Scenario

<table>
<thead>
<tr>
<th>Probability of Selected Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel</td>
</tr>
<tr>
<td>1%</td>
</tr>
</tbody>
</table>

**16) Automated estimates**
- Automated plans
- Formal tracking
- Optimal quality control
BODY OF KNOWLEDGE FOR SOFTWARE PROJECTS

- Problem 1: Requirements grow at > 2% per month
- Problem 2: Defects and errors often > 5 per function point
- Problem 3: Cumulative defect removal often < 85%
- Problem 4: Testing defect removal often < 70%
- Problem 5: Average schedules => function points ^ 0.4 power
- Problem 6: Clients expect schedules of <= function points ^ 0.3
- Problem 7: Above 10,000 function points project failures > 50%
- Problem 8: Most delays due to poor quality and requirements creep
- Problem 9: Project Managers not equipped to defend estimates
Information systems and outsourcers have smaller staffs than average.

Military and systems projects have larger staffs than average. Projects with tight schedules have larger staffs than average.
GROWTH OF NEW FUNCTIONS AFTER REQUIREMENTS

Range is 1% to 3% per month growth in new requirements

Maximum growth > 250%
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
SOFTWARE PAPERWORK

Application Size in Function Points

Web applets and client-server < 250 words per function point in plans, specifications
Less Paperwork Than Expected

More Paperwork Than Expected

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

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

Application Size in Function Points

Probability of Cancellation

SEI CMM 1  > 40% failures
SEI CMM 3  < 15% failures
SEI CMM 5  < 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 CMM Level 1</td>
<td>35%</td>
<td>40%</td>
</tr>
<tr>
<td>SEI CMM Level 2</td>
<td>30%</td>
<td>30%</td>
</tr>
<tr>
<td>SEI CMM Level 3</td>
<td>20%</td>
<td>15%</td>
</tr>
<tr>
<td>SEI CMM Level 4</td>
<td>12%</td>
<td>05%</td>
</tr>
<tr>
<td>SEI CMM Level 5</td>
<td>10%</td>
<td>03%</td>
</tr>
</tbody>
</table>
SOFTWARE LIFE EXPECTANCY

Application Size in Function Points

Years in Production Before Replacement

After 5 years restructuring and complexity analysis are needed

After 10 years compilers may not be available
ANNUAL SOFTWARE ENHANCEMENTS

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

Function Points per Staff Month

Application Size in Function Points

- **Overhead of base application**
- **New features for existing applications**
- **Major structural changes**
SOFTWARE QUALITY IMPROVEMENT

Defects per FP

Defect Removal Efficiency

SEI CMM 1

U.S. Average

SEI CMM 3

SIX SIGMA & SEI CMM 5

Best in Class

0 1 2 3 4 5 6 7 8 9 10
50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%

Malpractice
CURRENT 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>TOTAL</td>
<td>5.00</td>
<td>85%</td>
<td>0.75</td>
</tr>
</tbody>
</table>
### ‘BEST IN CLASS’ RESULTS 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>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>
SIX STAGES ON THE PATH TO SOFTWARE EXCELLENCE

Stage 0: Assessment, Baseline, Benchmark analysis
Stage 1: Focus on Project Management
Stage 2: Focus on Structured 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
### 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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Assessment</td>
<td>- 10%</td>
<td>0</td>
<td>- 10%</td>
</tr>
<tr>
<td>1</td>
<td>- 50%</td>
<td>25%</td>
<td>- 15%</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>- 10%</td>
<td>35%</td>
<td>- 15%</td>
</tr>
<tr>
<td>Methods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>- 5%</td>
<td>10%</td>
<td>- 5%</td>
</tr>
<tr>
<td>Tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>- 85%</td>
<td>65%</td>
<td>- 50%</td>
</tr>
<tr>
<td>Infrastructure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>- 5%</td>
<td>5%</td>
<td>- 5%</td>
</tr>
<tr>
<td>Reusability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>- 90%</td>
<td>350%</td>
<td>- 70%</td>
</tr>
<tr>
<td>Leadership</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Results</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2005 by Capers Jones. All Rights Reserved.
## 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>Assessment</td>
<td>$125</td>
<td>$150</td>
<td>$250 SEI CMM 1</td>
</tr>
<tr>
<td>Management</td>
<td>$1000</td>
<td>$2500</td>
<td>$3000</td>
</tr>
<tr>
<td>Methods</td>
<td>$1500</td>
<td>$2500</td>
<td>$3500 SEI CMM 2</td>
</tr>
<tr>
<td>Tools</td>
<td>$2500</td>
<td>$3500</td>
<td>$5000</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>$1500</td>
<td>$2000</td>
<td>$3000 SEI CMM 3</td>
</tr>
<tr>
<td>Reusability</td>
<td>$2000</td>
<td>$2500</td>
<td>$3500 SEI CMM 4</td>
</tr>
<tr>
<td>Leadership</td>
<td>$1000</td>
<td>$1000</td>
<td>$2000 SEI CMM 5</td>
</tr>
<tr>
<td>Overall Results</td>
<td>$9625</td>
<td>$14150</td>
<td>$20250</td>
</tr>
</tbody>
</table>
PROCESS IMPROVEMENT 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 $1,000,000

- TOTAL VALUE $15,000,000 *
- RETURN ON INVESTMENT $15 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)
> 6 Sigma Baseline, Benchmark
> SPR Assessment, Baseline, Benchmark
> ISO 9001 - 9004 Audit
> Putnam Baseline, Benchmark
> Gartner Baseline, Benchmark
> David’s Baseline, Benchmark
> IFPUG Baseline, Benchmark
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

- Reviews and Inspections
- Joint Application Design (JAD)
- 6 Sigma methodology
- Agile methodologies with caution
- ISO 9001 - 9004 Certification with caution
- SEI maturity levels
- Geriatric Technologies for legacy systems
STAGE 3: FOCUS ON NEW TOOLS & APPROACHES

- Key Technologies -- New Tools
  - Integrated Tool suites
  - Multi-media Tools
  - Repositories
  - Web and Internet Tools

- Key Technologies -- New Approaches
  - Requirements analysis tools
  - Design tools
  - Object-oriented Methods
  - 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 CMM 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
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
GOOD TOOL SUITES

• Without good tool suites, management and staffs are handicapped

• Attributes of good tool suites:
  – Both project management and technical tools
  – 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
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