Results of eServices Product Sizing Metrics Correlations

Yue Chen, Barry Boehm, Ray Madachy, Winsor Brown
Center for Software Engineering
University of Southern California
941 W. 37th Place, SAL Room 328
Los Angeles, CA 90089-0781
{yuec, boehm, madachy}@sunset.usc.edu
October 21, 2003

Presentation Outline

- Motivation of UML sizing
- Selection of projects
- Counting methodology
- Experiment results and analysis
- Conclusion
- Open Issues
Why UML Sizing?

- Solid information source about software size and complexity
- Countable design element
- UML is the most popular system modeling language
- Programming language independent

UML Diagrams

<table>
<thead>
<tr>
<th>MAJOR</th>
<th>VIEW</th>
<th>DIAGRAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural</td>
<td>Static view</td>
<td>Class diagram</td>
</tr>
<tr>
<td></td>
<td>Use Case view</td>
<td>Use case diagram</td>
</tr>
<tr>
<td></td>
<td>Implementation view</td>
<td>Component diagram</td>
</tr>
<tr>
<td></td>
<td>Deployment view</td>
<td>Deployment diagram</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Interaction view</td>
<td>Sequence diagram</td>
</tr>
<tr>
<td></td>
<td>Activity view</td>
<td>Activity diagram</td>
</tr>
<tr>
<td></td>
<td>State machine view</td>
<td>State transition diagram</td>
</tr>
<tr>
<td>Management</td>
<td>Model management view</td>
<td>Class diagram, package, subsystem, etc.</td>
</tr>
</tbody>
</table>
How projects are selected?

All selected projects are

- from 2001 – 2003 USC CSCI577(b) class projects
- targeted in building eService Applications
- development intensive projects (none COTS intensive)

List of 14 Projects

<table>
<thead>
<tr>
<th>CODE</th>
<th>TITLE</th>
<th>LANGUAGE</th>
<th>DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sp03t03</td>
<td>UML2Web</td>
<td>Python</td>
<td>Web</td>
</tr>
<tr>
<td>Sp03t06</td>
<td>Pilot Web Based Geotechnical Virtual Data Center</td>
<td>JSP</td>
<td>Web</td>
</tr>
<tr>
<td>Sp03t09</td>
<td>Conference Trip Planning System</td>
<td>HTML, PHP, SQL</td>
<td>Web</td>
</tr>
<tr>
<td>Sp03t13</td>
<td>Quality Information Management System for 577 Course</td>
<td>JSP</td>
<td>Web</td>
</tr>
<tr>
<td>Sp03t14</td>
<td>Caroline’s Closet</td>
<td>ASP</td>
<td>Web</td>
</tr>
<tr>
<td>Sp02t01</td>
<td>Dental Library New Booklist</td>
<td>Perl</td>
<td>Web</td>
</tr>
<tr>
<td>Sp02t06</td>
<td>ISO Interactive Web Based Contract Management System</td>
<td>PHP, HTML</td>
<td>Web</td>
</tr>
<tr>
<td>Sp02t15</td>
<td>Strategic Risk-Value Assessment Tool</td>
<td>Java, JSP</td>
<td>Web</td>
</tr>
<tr>
<td>Sp02t19</td>
<td>Opportunity Tree Framework</td>
<td>JavaScript</td>
<td>Web</td>
</tr>
<tr>
<td>Sp01t01</td>
<td>Station Data Project for the Web</td>
<td>HTML/Java</td>
<td>Web</td>
</tr>
<tr>
<td>Sp01t03</td>
<td>Pathology Image Search Engine</td>
<td>JSP/Java</td>
<td>Web</td>
</tr>
<tr>
<td>Sp01t08</td>
<td>Full-Text Titles Database</td>
<td>HTML/Java</td>
<td>Web</td>
</tr>
<tr>
<td>Sp01t14</td>
<td>Access &amp; Display Archive Image Composer</td>
<td>JSP/Java/HTML/ SQL/CDML</td>
<td>Web</td>
</tr>
<tr>
<td>Sp01t17</td>
<td>Web Mail</td>
<td>JAVA/HTML/JavaScript/JSP</td>
<td>Web</td>
</tr>
</tbody>
</table>
Counting Methodology

- The code counter tool used for counting is the Code Counter Pro by Geronesoft. The result is justified into Logical SLOC by comparing LSLOC/PSLOC ratio.

- The number of function point is backfired from SLOC based on the “Function Point Language Gearing Factors” provided by QSM.

- The project UML statistics are counted from the USC CSCI 577 course archives (2001 – 2003, 14 projects)

Level of Design Detail Matters

[Note] The numbers in the above diagram are calculated from the counting average from 14 USC CSCI 577 software engineering course projects.
Use Cases vs. Function Points

- The number of Function Point is backfired from the counting result of SLOC.
- The number of use cases is counted from the OCD document in the project “Final Deliverables” package.
- The code counter tool used for counting is the Code Counter Pro by Geronesoft.

Why there is an outlier?
The use case diagram of this project is over detailed. There is an inconsistence between the diagram and the documentation: there are 16 use cases but only 9 of them are documented with implementation detail.
Trend Line without the Outlier

\[ y = 20.158x - 13.823 \]

\[ R^2 = 0.4356 \]

Effect of Project Complexity

- Group the projects by their number of use cases.
- : the projects with greater value of the number of sequence-diagram-steps per use case comparing with its other group members.
**Comparison: FP vs. High-level Design Classes**

\[ y = 3.3481x + 34.218 \]
\[ R^2 = 0.2049 \]

**Comparison: FP vs. Number of Steps in Sequence Diagram**

\[ y = 0.2114x + 79.914 \]

[Note] The number of capability requirement is counted from the SSAD document in the project “Final Deliverables” package.
Comparison: FP vs. Number of Capability Requirements

\[ y = 2.2653x + 72.14 \]
\[ R^2 = 0.0191 \]

[Note] The number of capability requirement is counted from the SSRD document in the project “Final Deliverables” package.

\[
\begin{array}{|c|c|c|c|}
\hline
R^2 & y=a^x+b \quad \text{Linear} & y = \log(x) \quad \text{Logarithmic} & y = a^x \quad \text{Exponential} & y = x^a \quad \text{Polynomial} \\
\hline
\text{FP v.s. # of Use Case} & 0.436 & 0.442 & 0.476 & 0.548 \\
\text{FP v.s. # of classes} & 0.205 & 0.277 & 0.313 & 0.440 \\
\text{FP v.s. # of capability requirements} & 0.019 & 0.071 & 0.057 & 0.103 \\
\hline
\end{array}
\]
Conclusions

• Data shows the strongest correlation between the number of use cases and the number of backfired FPs.

• The polynomial model fits best

\[(\text{Number of FP}) \sim 4.7363(\text{Number of Use Cases})^{1.6938}, R^2 = 0.548\]

• The number of sequence diagram steps per use case seems to be a project complexity indicator

Open Issues

• The team’s documentation style
  – A complex project documentation cannot conclude a complex project implementation.

• The accuracy of backfiring SLOC to Function Point

• More project data points
Thanks!