Advances in Measuring Software Size and Productivity

Dr. Bill Curtis
Executive Director, CISQ

Productivity Analysis Objectives

- Improvement
- Estimation
- Productivity Analysis
- Benchmarking
- Managing Vendors
Productivity Analysis Measures

Size
- Instructions
- Functions
- Requirements
- Functional
- Structural
- Behavioral

Primary Measures

Effort
- Hours
- Roles
- Phases
- Application
- Project
- Organization

Quality

Adjustment Measures

Productivity Analysis

Software Size Measures

Instructions | Lines of Code
---|---
Most frequently used. Different definitions of a line can cause counts to vary by 10x. Smaller programs often accomplish the same functionality with higher quality coding.

Requirements-based | Use Case Points, Story Points
---|---
Use Case Points have not become widely used and need more development. Story points are subjective to each team and are susceptible to several forms of bias.

Functions | Function Points
---|---
Popular in IT. Several counting schemes (IFPUG, NESMA, Mark II, COSMIC, etc.). Manual counting is expensive and subjective—certified counters can differ by 10%. Automated FPs taking root.
**Function Point Estimation**

![Graph with the equation $y = 7.79x + 43.50$ and $R^2 = .95$.](image)

- Functional size can be estimated from external inputs and outputs
- Upfront functional analysis provides basis for good estimates
- Repository of FP data provides basis for calibrating estimates

**Automated Function Points**

- Mirrors IFPUG counting guidelines, but automatable
- Specification developed by international team led by David Herron of David Consulting Group
- Submitted thru OMG’s fasttrack as ISO 19515, currently under review

---

Ebert & Dumke (2007). Software Measurement, p.188.
Automated Enhancement Points

• IT shops found that both automated and manual Function Points had severe limitations in productivity analysis → they did not include the size of non-functional code

• The Automated Enhancement Points specification measures both functional and non-functional code and integrates them into one size measure

Effort — Weakest Measure

After the fact estimates
- Memory lapses
- Time-splicing
- Inconsistency

Under-reporting
- Contract issues
- HR issues
- Impressions

Lack of normalization
- Roles included
- Phases included
- Hours in P-Year

Unreliable, Inconsistent
**How Quality Affects Productivity**

**Assumption:** Productivity is a stable number  
**Reality:** Productivity is unstable, tending to decline

Unless you take action !!!

**Carry-forward Rework**

- **Release N**
  - Develop N
  - Rework N
  - Unfixed defects release N

- **Release N+1**
  - Develop N+1
  - Rework N+1
  - Rework N
  - Unfixed defects release N
  - Unfixed defects release N+1

- **Release N+2**
  - Develop N+2
  - Rework N+2
  - Rework N+1
  - Rework N
  - Unfixed defects release N+1
### Example of Quality Impact

<table>
<thead>
<tr>
<th>Project A (Plodders)</th>
<th>Project B (Better, Faster, Cheaper)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 developers, 3 months</td>
<td>20 developers, 3 months</td>
</tr>
<tr>
<td>$120k per FTE</td>
<td>$120k per FTE</td>
</tr>
<tr>
<td>3 FPs per staff month</td>
<td>4 FPs per staff month</td>
</tr>
<tr>
<td>180 FPs delivered</td>
<td>240 FPs delivered</td>
</tr>
<tr>
<td>$3,333/FP cost</td>
<td>$2,500/FP cost</td>
</tr>
</tbody>
</table>

*Project B is 25% more productive*

### However !!!

<table>
<thead>
<tr>
<th>Project A (Plodgers)</th>
<th>Project B (Better, Faster, Cheaper)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 critical violations per FP</td>
<td>5 critical violations per FP</td>
</tr>
<tr>
<td>$500 per fix</td>
<td>$500 per fix</td>
</tr>
<tr>
<td>Cost for 360 fixes = $180k</td>
<td>Cost for 1200 fixes = $600k</td>
</tr>
<tr>
<td>Total Cost to Own = $780k</td>
<td>Total Cost to Own = $1,200k</td>
</tr>
<tr>
<td>$4,333/FP of TCO</td>
<td>$5,000/FP of TCO</td>
</tr>
</tbody>
</table>

*Project A is 13.4% more productive*

---

### Quality-Adjusted Productivity

- Size of both functional and non-functional code segments
- Automated Enhancement Points
- Quality-Adjusted Productivity
- Effort & Cost
- Automated Technical Debt
- Productivity
- Estimation
- Benchmarks
- Value & ROI
- Etc.
Best Practices in Productivity Analysis

1) Segment baselines
2) Beware sampling effects
3) Understand variation
4) Evaluate demographics
5) Investigate distributions
6) Account for maturity effects
7) Beware external data sets

2 — Segment Baselines

Multiple baselines are usually the most valid

<table>
<thead>
<tr>
<th>Year</th>
<th>Projects</th>
<th>Productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Corporate</td>
<td>1981</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>1980</td>
<td>21</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>1981</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>1980</td>
<td>12</td>
</tr>
<tr>
<td>Engineering &amp; Defense</td>
<td>1981</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1980</td>
<td>6</td>
</tr>
<tr>
<td>Business Applications</td>
<td>1981</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1980</td>
<td>3</td>
</tr>
</tbody>
</table>
3 — Beware Sampling Effects

![Graph showing lines of code/person-years from 1980 to 1982 with two peaks representing lots of small projects and lots of large projects.]

CISQ Membership Is Free — www.it-cisq.org

Over 2000 individual members from large software-intensive organizations:

- Accenture
- Amazon
- Infosys
- IBM
- Bosch
- Aetna
- John Deere
- Booz Allen Hamilton
- Credit Suisse
- Gartner
- IAOP
- Allianz
- Capital One
- CA Technologies
- Dell
- FedEx
- Discovery Health
- CenturyLink
- MITRE
- Barclays

© 2018 Consortium for IT Software Quality (CISQ) www.it-cisq.org