What Is the Value of Your Software?

Jelle de Groot, Ariadi Nugroho, Thomas Back and Joost Visser

June 5, 2012

Rembrandt Tower, 14th floor
Amstelplein 1
1096 HA Amsterdam

info@sig.eu
www.sig.eu

Outline

Background
Approach
Exploratory Study
Conclusion
Framing the notion of software value

Software value = Software production value

**Produce**
- Cost to produce or replace
  - Based on actual software development expenses in the past
  - Based on market average prices for software components and development resources

**Exchange**
- Transaction Price
  - Custom software assets are not commodities traded on a competitive market

**Use**
- Future Benefits
  - **Innovation**: new products or services
  - **Optimization**: more efficient production
  - **Conformance**: to laws and regulations
  - **Reliability**: produce with stable quality
Background

Issues related to software value

- Firms invest heavily on software, but often unsure of its value
- Accounting practice tends to over estimate software value

How can we determine software value more objectively?
Technical debt as a component of software value

- Technical quality affects software value
- Poorly written code reduces value

**Impairment based on technical debt**

Value = Area

Value = Area - Debt
Approach

Measuring technical debt

- Based on SIG quality model of software maintainability
- Quality gap to the “ideal” level determines debt
- Debt estimate is based on more than 900 system snapshots

SIG Quality Model

<table>
<thead>
<tr>
<th>Rework Fraction (RF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-star</td>
</tr>
<tr>
<td>2-star</td>
</tr>
<tr>
<td>3-star</td>
</tr>
<tr>
<td>4-star</td>
</tr>
<tr>
<td>5-star</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1-star</th>
<th>2-star</th>
<th>3-star</th>
<th>4-star</th>
<th>5-star</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-star</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-star</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-star</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-star</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-star</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Approach

Proposed variants in measuring software value

- V1: Subtracts repair cost (technical debt) from rebuild value
- V2: Exclude parts that require rework from valuation
- V3: Subtracts extra maintenance cost (interest) from rebuild value

V1: Fix the problem
V2: Replace problematic parts
V3: Operate with higher cost
Applying the proposed valuation methods to 367 systems

Descriptive Statistics

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size (KLOC)</td>
<td>77.0</td>
</tr>
<tr>
<td>Quality (star rating)</td>
<td>3.0</td>
</tr>
<tr>
<td>Rebuild value (MY)</td>
<td>7.8</td>
</tr>
<tr>
<td>Rework Fraction (%)</td>
<td>35.0</td>
</tr>
<tr>
<td>Repair effort (MY)</td>
<td>1.9</td>
</tr>
</tbody>
</table>
Explorative Study

Comparison of three valuation models

The three valuation models give similar values around €7 per LOC.
Explorative Study

Software value across different technologies

C# and Java systems have the highest value. ABAP systems have the lowest value.
Explorative Study

Technical debt across different technologies

Java systems have the lowest technical debt averaging €1.6 per LOC
Conclusion

• The notion of technical debt can be used to estimate the value of software that takes into account technical quality
• No significant difference in the results given by the proposed valuation models (RE-based, RF-based, Interest-based)
• C# systems have the highest value averaging €10 per LOC

Future Work

• Compare results with traditional valuation approaches
• Devise a method to estimate business value of software
Estimating Rework Fraction

Rework fraction is determined based on more than 900 system snapshots.

For each quality leap:

\[ RF = \text{MAX}(\text{UI}, \text{US}, \text{UC}, \text{MC}, \text{Dup}) \]