Software Economics

Gain financial, managerial, and technical control of complex software portfolios

Dan Sturtevant, PhD
CEO, Silverthread Inc.
dan@silverthreadinc.com
M: 781.223.8200
Finite key resources are spent developing software systems. Some projects go well while others face significant risk.

- 50% of software dollars are wasted
- 90% of software projects are delayed or worse
Which means many more capabilities could be developed with the same resources
Based on 15 years of MIT research, we help enterprises assess and improve technical health & business outcomes

**Unhealthy Codebase**
- Module Too Big
- API Circumvented
- Layer Compromised
- Cyclical

**Healthy Codebase**
- Modular
- Layered
- Hierarchical dependence

<table>
<thead>
<tr>
<th>Developer productivity</th>
<th>Developer time fixing bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Features/Year</td>
<td>69%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Developer productivity</th>
<th>Developer time fixing bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Features/Year</td>
<td>20%</td>
</tr>
</tbody>
</table>
We helped one Air Force customer achieve a >20x ROI by making wiser financial choices and driving refactoring.

100 major systems

**HEALTHY EXAMPLE:** System X
Team had built and grown great system → Team rewarded with future work

**REPAIR EXAMPLE:** System Y
System challenged, and fixed → Team reports 3X productivity, very rapid ROI

**REBUILD EXAMPLE:** System Z
Major issues, Total rewrite best option → $Billions allocated to recapitalize

Copyright © Silverthread 2014-2019
How to get started: Proof of Concept Ideas

Step 1:
- Fully automated diagnostic deep dive into a few strategic systems

Step 2:
- Portfolio analysis across your domain for full visibility

CodeMRI® Diagnostics

CodeMRI® Care

CodeMRI® Portfolio
Silverthread provides benefits at multiple levels

<table>
<thead>
<tr>
<th>Executive Leadership</th>
<th>Project Leadership</th>
<th>Technology Leadership</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Decide what to continue with maintenance, refactor, rewrite, accept, or reject</td>
<td>• Estimate development cost and schedule more effectively</td>
<td>• Define design rules and audit the code’s adherence in real-time</td>
</tr>
<tr>
<td>• Know where to focus finite effort and resources</td>
<td>• Construct ROI-based business cases for improvement</td>
<td>• Manage refactoring, cloud transformation, migration to an open platform, etc.</td>
</tr>
<tr>
<td>• Increase agility, security, and quality across the organization</td>
<td>• Control system architecture to ensure it does not degrade</td>
<td>• Build a codebase that developers understand and can confidently modify</td>
</tr>
</tbody>
</table>

### CodeMRI® Portfolio

### CodeMRI® Diagnostics

### CodeMRI® Care

#### Indigo 17.11.29 Analysis

- **Primary language:** C++
- **Number of files:** 33,769
- **Lines of code (LOC):** 9,935,938
- **Average file size (LOC):** 294
- **Diagnostic confidence:** Moderate

#### Economic Outcomes

- **Cost to produce 1000 LOC:** $30,597
- **Days to develop 1000 LOC:** 45
- **Money wasted per additional $1M invested:** $766,886
- **Bug to feature ratio (labor hours):** 42%
- **Number of bug LOC added / exposed per year:** 1,155,388
- **Number of bug LOC released per year:** 784,641

#### Technical Health

- **Number of critical Cores:** 1
- **Number of emerging Cores:** 7
- **Files with high complexity:** 1722
- **Files with problematic complexity:** 5478
- **Files potentially affected by single change:** 1976
- **Complex files vs. scanned codebase:** 16%

#### Design Quality

- **The “technical health” of a codebase predicts both the Cost of Ownership and Quality and Risk associated with a particular program.**

#### Legend

- **Unlikely to have negative effects on business outcomes; should be closely monitored.**
- **Significantly impacted outcomes as a result of technical health.**