Problems and solutions: Maintaining an integrated system in a community of volunteers

Thomas Østerlie
Overview

• Purpose: Brief introduction to the thesis
• Main message
  – Need for more research on integrated systems maintenance
  – Systems integration requires rethinking some premises of software maintenance (problem solving/setting)
• Outline
  – Why? Part I: Motivation and goals
  – How? Part II: Research
  – What? Part III: Results and conclusions
Part I: Motivation and goals
Outline

• Part I: Motivation and goals
  – Systems integration
  – Software maintenance
  – Software maintenance work
  – Research goals and questions

• Part II: Research

• Part III: Results and conclusions
# Systems integration

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Application software development</th>
<th>Systems integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software composition</td>
<td>Coherent set of software modules</td>
<td>Third-party components and application code</td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Developers and users</td>
<td>Providers, integrators, and users</td>
</tr>
<tr>
<td>Source code</td>
<td>Full source available</td>
<td>No source code for third-party components</td>
</tr>
<tr>
<td>Execution model</td>
<td>Single computer</td>
<td>Often distributed between multiple computers</td>
</tr>
<tr>
<td>Ownership and control</td>
<td>Single team or organization</td>
<td>Distributed between multiple organizations</td>
</tr>
</tbody>
</table>
Origins of maintenance term
Software maintenance

Requirement analysis

Software design

Software implementation

Testing

Maintenance burden
1970s: 40% total lifecycle cost
1980s: 55% total lifecycle cost
1990s: 90% total lifecycle cost

Increasing life expectancy
Organizations spend more time maintaining legacy systems, than developing new software

Software maintenance
Maintaining integrated systems

• Current state of software maintenance research
  – Limited emphasis on software maintenance in the software engineering literature
  – Practically everything focused on application software maintenance
  – Practically no research on maintaining integrated systems

• The challenge
  – Research-informed software engineering practice
  – How to inform practice when researchers know little about how integrated systems are maintained?
Software maintenance work

- How integrators maintain an integrated system in practice
- Software maintenance as knowledge-intensive work
- Knowledge-intensive work
  - **Practice-based**: Work and knowledge interrelated
  - **Situational**: How work unfolds over time while options and dilemmas remain unresolved for those at work
  - **Contextual**: Work takes place within the broader social and organizational context
Research goals and questions

• Goals
  – Explore maintenance of an integrated system in its context of development and use
  – Explore the intertwined social and technical factors that influence software maintenance work

• Research questions
  – RQ1: How is knowledge of software failures developed during geographically distributed software maintenance?
  – RQ2: How do software developers build knowledge of how to replace a business-critical software system?
  – RQ3: What are the characteristics of maintaining an integrated system in a distributed community of volunteers?
Part II: Research
Outline

• Part I: Motivation and goals
• Part II: Research
  – Research setting: Gentoo
  – Research process
• Part III: Results and conclusions
The Gentoo community

- Geographically distributed community
  - 320 volunteer software integrators
  - 38 countries and 17 time zones
  - An unknown number of users

- Internet-based collaboration
  - Internet Relay Chat
  - Mailing lists
  - Internet-based defect tracking system
  - Internet-based revision control system

- Gentoo and maintenance
  - Perpetual cycle of corrective and adaptive maintenance
The Gentoo technology

- Gentoo community develops and maintains
  - **Portage**: A software system for integrating third-party OSS with different Unix-like operating systems
    - Supported operating systems: 4
    - Supported hardware platforms: 8
    - Supported third-party packages: 8486
    - Total number of installation scripts: 23911
  - **Gentoo Linux**: GNU/Linux distribution based around Portage

- Gentoo community operates
  - **Infrastructure**: For distributing third-party OSS to computers running Portage
## Research process

<table>
<thead>
<tr>
<th>Study</th>
<th>Period</th>
<th>Activities</th>
<th>Materials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empirical fieldwork</td>
<td>January 2004 – December 2004</td>
<td>• Observation • Participation (reporting bugs, fixing bugs, coding)</td>
<td>• Fieldnotes • 1027 IRC logs • 70 documents • Mailing list archives</td>
</tr>
<tr>
<td>Analysis of corrective maintenance work</td>
<td>September 2005 – March 2006</td>
<td>Formalized document analysis</td>
<td>Gentoo's defect tracking system's database (20,000 problem reports)</td>
</tr>
<tr>
<td>Investigating transferability of findings</td>
<td>March 2006 – September 2006</td>
<td>3 feedback session with professional systems integrators</td>
<td>Notes from the three sessions</td>
</tr>
</tbody>
</table>
Part III: Results and conclusions
Outline

• Part I: Motivation and goals
• Part II: Research
• Part III: Results and conclusions
  – Problems and solutions revisited
  – Empirical illustration: Corrective maintenance
  – Summary
## Problems and solutions revisited

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Problem solving</th>
<th>Problem setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting point</td>
<td>More or less clearly defined problems</td>
<td>Problem situations that are troubling, puzzling, unclear</td>
</tr>
<tr>
<td>Process</td>
<td>Rational decision-making</td>
<td>Sensemaking</td>
</tr>
<tr>
<td></td>
<td>• Analytic</td>
<td>• Action-oriented</td>
</tr>
<tr>
<td></td>
<td>• Choose among available tools to reach solution</td>
<td>• Collective process of trial and error</td>
</tr>
<tr>
<td></td>
<td>• Problem remains more or less stable</td>
<td>• Emergent understanding of the problem situation</td>
</tr>
<tr>
<td>End point</td>
<td>Problem solved</td>
<td>Problem situation reaches closure</td>
</tr>
</tbody>
</table>
Problem solving

• "Change the position of just one line to leave the giraffe in exactly the same position as before."
• How would you go about solving it?
Conventional corrective maintenance process
Gentoo's formal corrective maintenance process

1. Report bug (User)
2. Assign problem to herd (Bug Wranglers)
3. Assign problem to debuggers (Herd)
4. Resolve bug (Herd/user)
Fri 10 22:45:52 <nmjc> foser: just install standard xorg version, you'll lose all
of the Gnome benefits though
Fri 10 22:45:55 <nmjc> foser or you can just set up a chroot
Fri 10 22:46:14 <gerry/np> back in an hour or so
Fri 10 22:48:08 <nmjc> njck have a look at https://bugs.gentoo.org/show_bug.cgi?id=66836
Fri 10 22:49:06 <genBot> foser: Bug 66836: mjck, P2, x86, gnome@gentoo.org, NEW:
emerge gnome-lib.so fails with several 'undefined reference's
Fri 10 22:54:28 <nmjc> foser: USE="-msound" might solve it, also try telling him
to remerge sound

----- Comment #4 From foser 2004-10-11 04:02 PST -----
what is the output of 'glib-config --cflags --libs gthread' ?
I think your glib is for some reason compiled incorrectly ..
what is the output of 'gpkg -l glib-1' (needs gentoolkit ) ?

----- Comment #5 From Stephen Anderson 2004-10-11 06:30 PST -----
-L/usr/include/glib-1.2 -L/usr/lib/glib/include -D_REENTRAYNT
-L/usr/lib -Lgthread -lglib -lpthread
Starting point: Problem situation

• Unclear what the problem really is
  – Local/global: User error or problem for many users?
  – Location: Problem with Gentoo or third-party software?
  – Responsibility: Location determines responsibility for correction

• Indirect access to failures
  – Variability → Difficult to reproduce failures → Indirect data →
    Multiple valid interpretations of available data
  – Problem of relevance: If the problem is unclear, it is unclear what
    the constitutes relevant data
  – Distribution and limited knowledge of third-party software makes it
difficult to trace infection chains
Process: Sensemaking
## Technical and non-technical

<table>
<thead>
<tr>
<th>Statements</th>
<th>Researchers' commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reporting user:</strong> I have installed my system from scratch</td>
<td>The problem is related to the way Gentoo integrates software, and therefore the Gentoo developers' responsibility.</td>
</tr>
<tr>
<td><strong>Developer A:</strong> [talking on IRC, making reference to the systems information provided with the problem report] Is using an x86 profile for an amd64 machine troublesome?</td>
<td>The reported problem is related to the way the user's Gentoo systems configuration; therefore the user's responsibility.</td>
</tr>
<tr>
<td><strong>Developer B:</strong> [making reference to the installation script] Turning off the optional esound support might solve the problem</td>
<td>The problem may be related to how the package integrates with the esound package, and the third-party provider's responsibility.</td>
</tr>
<tr>
<td><strong>Developer A:</strong> [making reference to the compiler error provided with the problem report] Why is it that the thing can't find pthread? is that because of a missing -pthread</td>
<td>The problem is related to the use of the pthreads library, and therefore the responsibility of another herd.</td>
</tr>
<tr>
<td><strong>Developer B:</strong> sounds like the glibc library was upgraded</td>
<td>Related to the user's system configuration, and his responsibility.</td>
</tr>
</tbody>
</table>
End point: Closure

workaround rather than solution to the problem at hand
Contributions to software engineering

• Contributions to theory
  – A theoretical framework for software maintenance work
  – Sensitizing concepts for further study of integrated systems maintenance

• Contributions to method
  – Techniques for practice-based Internet research on software engineering practice
  – Method bridge the gap between practice and artifact studies

• Contributions to practice
  – Recommendations for a lenient approach to coping with variability during corrective maintenance
  – Recommendations for an opportunity-driven approach to systems replacement
## Dimensions of distribution

<table>
<thead>
<tr>
<th>Dimension (organization)</th>
<th>Application software development</th>
<th>Open Source Software Development</th>
<th>Global Software Development</th>
<th>Systems Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical/geographical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizational</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stakeholders</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Theory development

Data
- E-mail
- IRC logs
- Documents

Theory
- Communities of Practice
  - Orr 1992
  - Brown and Duguid 1999
  - Lave and Wenger 1997
- Actor-Network Theory
  - Latour 1987
  - Callon 1987
- Critical Knowledge Theory
  - Alveson 1997
- Sensemaking
  - Weick 1997

Informative concepts
- problem situation
- inscription
- translation
- interests
- ambiguity
- sensemaking