Process Mining for Process Conformance Checking in an OSS Project: An Empirical Research

Kouzari Elia*, Sotiriadis Lazaros, Stamelos Ioannis

*Corresponding author (ekouzari@csd.auth.gr)
OUTLINE

- Process Mining
- The lifecycle of a bug in Bugzilla
- Koha Integrated Library System
- Methodology
- Findings
- Process Conformance Check
- Discussion
- Future steps
PROCESS MINING

- fields & sectors
- tools
- algorithms
- platforms
PROCESS MINING

- process discovery
- process conformance
- process enhancement

THE LIFECYCLE OF A BUG IN BUGZILLA

- Bugzilla OSS bug tracking system
- customized workflow to meet the needs of every community/project

KOHA INTEGRATED LIBRARY SYSTEM

- web-based open source ILS
- written in Pearl
- distributed under GNU general public license
- very active community of developers around the world
- uses Bugzilla for bug tracking
Koha suggests that users should review the Bugzilla Bug Writing Guidelines prior to submitting a bug to Koha.

The large and active community of developers and users makes Koha an ideal candidate for process conformance checking.
METHODOLOGY

Research Question

“Does the community of Koha Open Source ILS conform to the bug resolution process described in the project’s pages? If not, what is the actual process followed?”
FINDINGS - 1\textsuperscript{ST} STAGE

- **Input:** event log with 97,372 events for 19,311 bugs
- **Output:** a “spaghetti model”

Partially enlarged to show the most common activities.
FINDINGS - 2\textsuperscript{ND} STAGE

- **Input**: the filtered 47.9\% of the most common activities of the diagram

- **Output**: a clearer process model

Model affected by the status of the **first 283 bugs** in the event log (No history for the resolution process of these bugs was available rather than their status set to “Closed”)
**FINDINGS – 3RD STAGE**

- **Input**: an event log ignoring the first 283 bugs

- **Output**: an improved process model

Most common process path followed:

1. Needs Sign Off
2. Signed Off
3. Passed QA
4. Pushed to Master
5. Pushed to Stable
6. Resolved
7. Resolution/Fixed
8. Closed
**FINDINGS – 4TH STAGE**

- **Input**: an event log ignoring the first 283 bugs keeping only dimensions “Status” and “Resolution”
- **Filter**: endpoint filter (value FIXED) (10% of cases, 17% of events)
- **Output**: the bug resolution process in Koha
Process Conformance Check

- The discovered process is very similar to the one described in the community’s blogs and wikis.

**Most common process path followed:**

1. Needs Sign Off
2. Signed Off
3. Passed QA
4. Pushed to Master
5. Pushed to Stable
6. Resolved
7. Resolution/Fixed
8. Closed

*Process extracted by 3rd process mining attempt using an event log*

*Process described in project’s documentation*
PROCESS CONFORMANCE CHECK

- some obvious process variations

<table>
<thead>
<tr>
<th>Process Variance</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOLVED -&gt; FIXED -&gt; CLOSED</td>
<td>4.14%</td>
</tr>
<tr>
<td>PATCH-Sent -&gt; RESOLVED -&gt; FIXED -&gt; CLOSED</td>
<td>1.64%</td>
</tr>
<tr>
<td>ASSIGNED -&gt; PATCH-Sent -&gt; RESOLVED -&gt; FIXED -&gt; CLOSED</td>
<td>1.42%</td>
</tr>
<tr>
<td>Needs SignOff -&gt; Failed QA</td>
<td>0.37%</td>
</tr>
<tr>
<td>ASSIGNED -&gt; RESOLVED -&gt; FIXED -&gt; CLOSED</td>
<td>0.32%</td>
</tr>
</tbody>
</table>
DISCUSSION

- bug resolution process followed can change through time - depending on project’s maturity
- process variances do not conform to the proposed guidelines - freedom to act and modify procedures in OSS communities
- different bug resolution process than that proposed by Bugzilla - OSS tools can be freely customised to address the needs of a community
- Koha maintains a very active and organised community and contains excellent documentation - less structured and active OSS communities might use Koha’s processes as benchmarks for process enhancement
FUTURE STEPS

- process conformance with Bugzilla’s latest process for bug management (released on 2017)
- further empirical research on process conformance with other OSS projects
- investigation towards a “state-of-the-art” process for managing bugs in OSS projects
THANK YOU FOR YOUR ATTENTION

QUESTIONS???