FastFix Open-Source Formative Evaluation Process in Remote Software Maintenance

Alessandra Bagnato, TXT e-solutions – Corporate Research
Javier Cano and Christophe Joubert, Prodevelop

May 18th 2012

Industrial Track at RCIS 2012, Sixth International Conference on Research Challenges in Information Science 2012

“Be Fit, Fast and Alert when testing, maintaining or solving bugs of your Future Internet applications” FITTEST & FastFIX & ALERT & Io.Test Joint Workshop
1. Project Overview
2. Open Source Formative Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
OVERVIEW

1. Project Overview
2. Open Source Formative Evaluation Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
European project FASTFIX (ICT- 258109, June 2010 - February 2013) develops a monitoring control platform for remote software maintenance.

Within this project, we have been working on a formative evaluation process that evaluates methods, concepts and prototypes as the project is being developed.
WP 1 Project Management and Quality Assurance
WP 2 FastFix conceptual framework
WP 3 Context observation and User modeling
WP 4 Event correlation
WP 5 Fault replication
WP 6 Patch generation and self-healing
WP 7 Performance, security and privacy

**WP 8 Trials and validation**
WP 9 Dissemination and exploitation

Formative Evaluation is part of WP8
PURPOSE to ensure that the goals of the under-development software are being achieved and to improve the software, if necessary, by means of identification and subsequent remediation of problematic aspects.
OVERVIEW

1. Project Overview

2. Open Source Formative Evaluation Phases

3. Agile Development Process (Scrum)

4. Open Source Supporting Tools

5. Performance Requirements

6. FastFix Formative evaluation at work

7. Current metrics

8. Next steps
2.- FORMATIVE EVALUATION PROCESS

1. Analysis
2. Metrics
3. Tools
4. Testing
5. Monitoring
6. Reporting
FORMATIVE EVALUATION PROCESS

Analyze the project and development process.

Select tools to perform metrics collection.

Define tests to execute. Define which tool will collect what.

Select metrics to collect.

Define who is responsible for monitoring.

Select the reports to create from the data.
Analyze the project

Find the basic natures that define the project. In the case of FastFix:

<table>
<thead>
<tr>
<th>Strong focus in research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed teams</td>
</tr>
<tr>
<td>Development of new Software Artifacts</td>
</tr>
<tr>
<td>Integration into common platform</td>
</tr>
</tbody>
</table>
Analyze the development process to use. Try and find points that are worth monitoring.

In the case of FastFix:

- **Iterative process:**
  Requirements change as development advances.

- Requirements and issues appear at each iteration.

- Testing is to start early and happen often.
Find interesting metrics to be gathered.

In the case of FastFix:

tickets opened, solved, closed, overdue, etc.

committers activity, evolution of code base, regular activity.

builds and tests stability.

reports on ranked issues found in source code quality.
There exist metrics that indicate a good development of the project but are not directly visible:

- Issues discussed among partners.
- Detection of issues before they promote to a problem.
- Responses and actuations on comments from others.
- Honoring deadlines.
Select tools that integrate into the development process. In the case of FastFix:

<table>
<thead>
<tr>
<th>Monitorization target</th>
<th>Monitoring tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source code quality.</td>
<td>Static code analysis.</td>
</tr>
<tr>
<td>Components integration to form the FastFix platform.</td>
<td>Continuous integration.</td>
</tr>
<tr>
<td>Tests execution.</td>
<td>Automated builds and tests execution.</td>
</tr>
<tr>
<td>Platform integration in target applications.</td>
<td>Integration trials after internal releases.</td>
</tr>
<tr>
<td>Open and resolved issues.</td>
<td>Issues statistics reports.</td>
</tr>
</tbody>
</table>
Non-invasive tools to gather metrics

Automate the process of metrics gathering as much as possible:

- Integration into existing tools.
- Periodic jobs in build servers.
- Periodic scripts that collect information.
- Periodic scripts that generate reports.

Tools invisible until notifications are necessary
Automated tools collect metrics silently
Start testing early, test often

The best formative evaluation is the one that includes all stakeholders in the development:

- Include end users as soon as possible.
- Iterative testing with small increments.
- Make release cycles short.

Project execution

Testing and Trials

M0. Project starts.


M33. Project ends.

Internal Releases
Monitor evolution, watch for deviations

Monitor metrics gathered periodically (daily):

- Look for deviations
  - I.e.: continuous integration failing last week.

- Take corrective actions
  - I.e.: review build system logs to correct errors.

- Use automated notifications
  - I.e.: set up automated notifications to responsibilities of latest changes when build fails.
Create reports

- Collect all metrics in a single report
  - LOC
  - # of tests
  - Code quality

- Highlight areas to improve
  - Build stability
  - Tests

- Highlight good areas
  - Code quality
  - Dependencies management
The overall **benefit** of the process is to **correct problems** and to **include new requirements** before the production phase of the software project and **during** the whole life of the project.
1. Project Overview
2. Open Source Formative Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
Development: sprints from backlog. Produces an increment of the platform.
Agile development process

Development: sprints from backlog. Produces an increment of the platform.


Agile development at a glance

A simplified view of the development process followed in FastFix:

- Sprint planning
- Evaluation
- Development
- Testing
- Product increment
The formative evaluation hooks into the development process using standard tools:

- Issues management
- Sprint planning
- Development
- Evaluation
- Testing
- Product increment
- Continuous integration server
- Code repository analysis
- Code quality analysis
- Sonar
- Subversion
- Jmeter
- Metrics collection
- Hudson
**Development**: sprints from backlog. Produces an increment of the platform.

**Testing** of produced platform. Gathering of metrics.

**Evaluation**: Bug and enhancement issues stored in backlog. Metrics analyzed.
1. Project Overview
2. Open Source Formative Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
4.- OPEN SOURCE SUPPORT TOOLS

- FOCUS ON OPEN SOURCE
- CRITERIA
- SELECTED TOOLS
Focus on Open Source

WHY OPEN SOURCE TOOLS?

As a research project with focus on Open Source:

- Can deliver **industrial level performance**.
- **Free** as in beer and speech.
- Minimum **investment**.
- Great **customization**.
- Huge **integration**.
Consortium selects among all alternatives according to:
- Capabilities of each tool
- Maturity of each tool
- Integration with other tools
- Familiarity among partners
Subversion as source code and Maven as project management

Mature, well known source control management. Integrates with lots of other tools.
http://subversion.tigris.org/

Mature, well known project management, specially in Java. Integrates with lots of other tools. Simplifies dependency management and build process.
http://maven.apache.org/
Hudson as continuous integration

- Off-the-shelf Integration with Subversion and Maven.
- Notifications for failed builds.

http://hudson-ci.org/
Sonar as quality analyzer

- Quality analysis with FindBugs.
- Integration with Hudson.
- Reports generation.

http://www.sonarsource.org/
Trac as issue manager

- Integration with SVN.
- Notifications.
- Milestone planning.
- Reports generation.

http://trac.edgewall.org/
1. Project Overview
2. Open Source Formative Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
5.- PERFORMANCE REQUIREMENTS

GOAL
• Usability and performance of FastFixed systems not noticeably affected.

INVESTMENT
• FastFix invests part of the effort in tuning the performance of the platform.

PROCESS
• Each iteration release testing collects metrics to evaluate performance.
1. Project Overview
2. Open Source Formative Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
6. - FASTFIX EVALUATION AT WORK

Samples from the deployed tools:

• Hudson
• Sonar
• Trac
Hudson continuous integration system schedules **builds every night of the whole platform.** Notifications in place for interested parties.
FastFix Formative evaluation at work: SONAR

- **SONAR code analysis** system analyses code just after every correct build.
TRAC issues management system is used as a planning tool, to define milestones and their tasks, to manage responsibly for tasks and internal releases.
OVERVIEW

1. Project Overview
2. Open Source Formative Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
7.- METRICS UNTIL 15TH MAY 2012

- Lines Of Code
- Developers
- Continuous integration
- Issues management
Metrics until 15th May 2012
Lines Of Code: 879742

Lines Of Code: 879742

/Software: Lines of Code

Developers: 18
Continuous integration:

- **Build stability:**

  - **Common**
    - Stable: 80%
    - Failed: 20%
    - Unstable: 0%

  - **Client**
    - Stable: 85%
    - Failed: 15%
    - Unstable: 0%

  - **Server**
    - Stable: 75%
    - Failed: 20%
    - Unstable: 5%

- **Test stability:**

  - Correct
  - Failed
  - Skipped
Issues management:
- 282 tickets managed.
- 170 closed tickets.
- 112 open tickets.
  - 29 tickets in the backlog
  - 6 tickets from internal testing in industrial partners
Code quality

- 34.4% *Sonar Rules Compliance Index*
- 2547 violations [1 blocker, 147 critical, 1610 major, 732 minor, 57 info]
- 23.4% comments, 8.4% duplications, 100% test success
1. Project Overview
2. Open Source Formative Evaluation Phases
3. Agile Development Process (Scrum)
4. Open Source Supporting Tools
5. Performance Requirements
6. FastFix Formative evaluation at work
7. Current metrics
8. Next steps
8.- NEXT STEPS

Execution of trials in industrial partners with live production applications.

Collection of all metrics data, both qualitative and quantitative.

Reports that summarize the evaluation of the project.
Thank you for your attention!

Alessandra Bagnato  
Corporate Research Division - TXT e-solutions S.p.A  
alessandra.bagnato@txtgroup.com

Francisco Javier Cano and Christophe Joubert  
Prodevelop  
fjcano@prodevelop.es, cjoubert@prodevelop.es

FastFix Project Web Site:  
https://www.fastfixproject.eu