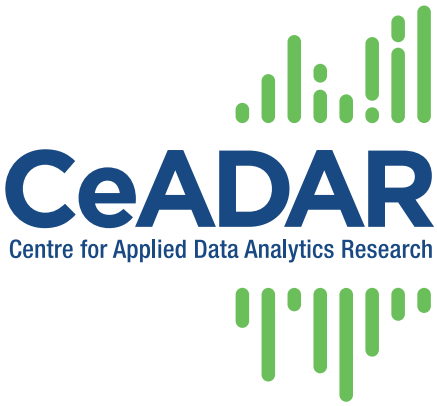

CeADAR – Centre for Applied Data Analytics Research

Enterprise Ireland Data Analytics Technology Centre

Intelligent Analytic Interfaces: Process Summarisation - Technical Specification

Document Type:	Technical Spec
Project Title:	CeADAR
DDN Theme:	1 – Intelligent Analytic Interfaces
Theme Leader:	Brian Mac Namee (DIT)
DDN Sub-Theme:	1.3 – Passive Analytics
Authors:	Ingo R. Keck, John Kelleher
Document Version:	1.0
Date of Delivery to ISG:	30th May 2014
Number of pages:	5

ABSTRACT

Process summarisation involves generating an overview of the current state of a process in a business. These overviews are useful for managers as they enable them to evaluate performance to date (of both the process and employee performance) but also to plan for the future. A very important process in software companies is bug- and issue-tracking. In the *SummIt* project, we will generate a data set from an open soure project and will then use text summarisation, keyword detection and frame filling to provide overviews of the current state of the bug-handling and its development over time.

CeADAR is a research partnerhip comprising University College Dublin, University College Cork, and Dublin Institute of Technology.

`http://www.ceadar.ie`

Contents

1 Description of Industry Needs

Process summarisation involves generating an overview of the current state of a business process. These overviews are useful for managers as they enable them to evaluate performance to date (of both the process and employees), and also to plan for the future. Bug tracking is an important process in the software industry. Managing the correct handling of issues is crucial and requires a quick oversight of the actual state of each process and its development over time. Furthermore, bug tracking data is easily available and the tracking process is a good approximation for many different types of processes in many different industries.

Imagine a bug-tracking manager summing up the previous week on a Friday afternoon or a Monday morning. His/her basic questions will be:

- What is the actual status of issues/bugs in the system?
- What are the changes since yesterday or last week?
- What is the performance of a specific user?
- Is there a problem looming up?
- Are there issues which are classified differently by employees compared to users?

While current issue trackers already provide an impressive amount of structured information on the status of issues, there is a substantial amount of free text provided by users and administrators in these systems that, up to now, has not been widely exploited for summarisation. The analysis and integration of the information in this free text into process summaries is the focus of the *SummIt* project.

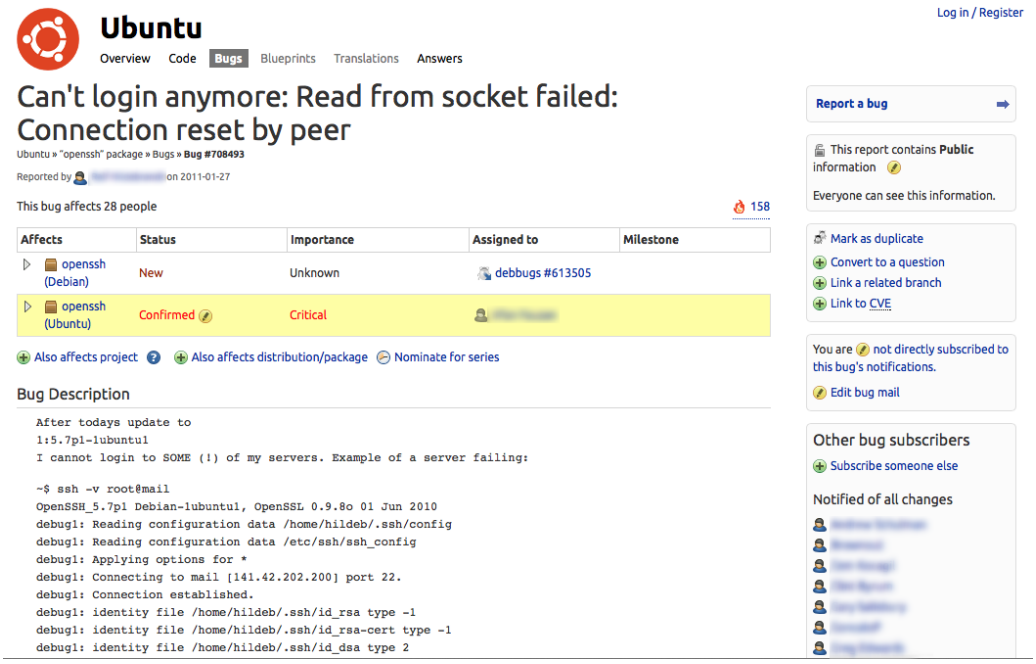

Figure 1: The launchpad bug tracker for Ubuntu. At the top, structured information is given like, for example, the current status of the bug, the affected part and the importance. “Bug Description" gives unstructured textual information about the nature of the bug and can be used for frame filling.

2 System(s) Involved

As an example dataset, we will scrape publicly available content from the launchpad bug-tracker for Ubuntu (see figure 1). This dataset provides a nice mixture of structured information from the bug tracking system and corresponding structured- and unstructured content in the comment section and log of each bug. The scraped data will be saved in a database for further evaluation.

For text summarisation and data mining, we will use the NLTK package for Python and appropriate machine learning packages.

For visualisation, we will continue to focus on providing a web interface, that allows a platform independent application and access from anywhere.

To allow for simulated evaluations of *SummIt*, a small server will be developed that can simulate the status changes of an issue based on the given date.

3 Approach

For this project we will concentrate on the following parts:

- The use of structured data from the system (status of the issue, criticality, dates) to define the status of the bug handling process over time (see table 1 for possible status labels and figure 2 for a typical issue-handling workflow).
- The use of structured data from the comments part (i.e. status change comments) to track the status of the process.
- The use of the original bug report (top comment) for frame filling, i.e. the detection of whether the bug report is complete, and contains the following three parts: a description; information on how to reproduce it; and a system log.
- The use of unstructured data from the comments for summarisation.
- The extraction of information from the unstructured comments by keyword search or topic detection.
- The detection and tracking of activity by users, for example the quality of their bug reports, speed of verification, speed of fixing.

Label	*Explanation*
New	Not looked at yet.
Incomplete	Cannot be verified, the reporter needs to give more info.
Opinion	Doesn't fit with the project, but can be discussed.
Invalid	Not a bug. May be a support request or spam.
Won't Fix	Doesn't fit with the project plans, sorry.
Confirmed	Verified by someone other than the reporter.
Triaged	Verified by the bug supervisor.
In Progress	The assigned person is working on it.
Fix Committed	Fixed, but not available until next release.
Fix Released	The fix was released.

Table 1: The possible labels for a bug report on launchpad.net

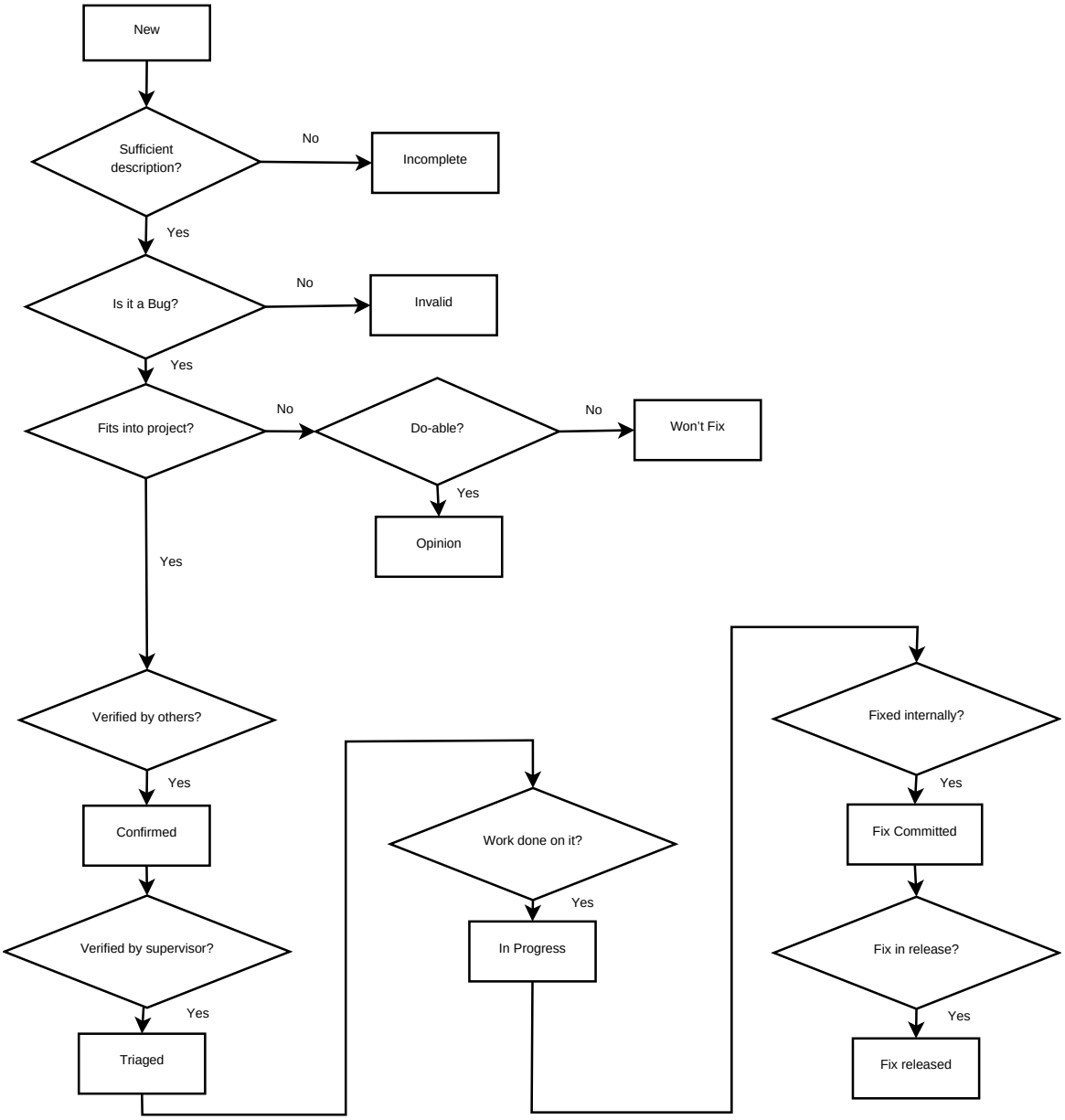

Figure 2: The workflow of a bug on launchpad.net

- The detection of repeated changes of the status in a short time (more than just an accidental change → undo)
- The comparison of labels from the system (for example criticality of a bug) with community opinion (indicated by the number of comments, number of affected and subscribed users)

We will use this information to extract meaningful summaries from the data for the last day / week and present it in an easy-to-understand visualisation in *SummIt*.