Improving Bug Tracking Systems

Alice submits new bug report

Alice reads comment, sends response

Developer has to wait for a response. Most questions are answered within 1 day, but some are open for years [1].


First results

Traversing derived decision tree: trained with 2875 bug reports from ECLIPSE JDK, bug reports related to 20 most frequently fixed files, considering only 20 most active reporters

Comment #2: I’m not sure. MacBook Pro from 2006? No, not debugging.

THE PROBLEM

Bugzilla has many question fields to fill in for a bug report. Not all need to be answered for each bug, but different ones for different bugs. Which?

THE SOLUTION

Formulate a set of general questions; each question asks for information that developers use to find the location of a bug.

What is the affected component? Which version is affected? Which version is other?

What platforms are affected? Can you provide a screenshot? How important is the bug? Which version is affected? ...

Derive a decision tree from question/answer-pairs and bug locations for a large number of fixed bugs.

What is the affected component? What platform?

Gather information from the user guided by the decision tree. As we traverse the decision tree, the bug location’s search space is narrowed down.

THE FUTURE?

Alice ... A Real Person

Bob ... A Bug-Tracking System

NEXT STEPS

1. Identify information needs in large sample of bug reports to compile a catalogue of questions to use in an expert system.

2. Using the catalogue, collect answers and defect locations for another large sample of bug reports; use it to train a prediction model.

3. Evaluate predictions and conduct usability studies.

THANKS FOR STICKING WITH!