content

- ✓ scenario
- ✓ problem
- ✓ solution
- ✓ evaluation
- ✓ future work
- ✓ summary

scenario


source
forge
planbox
FogBugz
Bugzilla
JIRA
trac
Integrated SCM & Project Management
mantis
BUG TRACKER
REDMINE

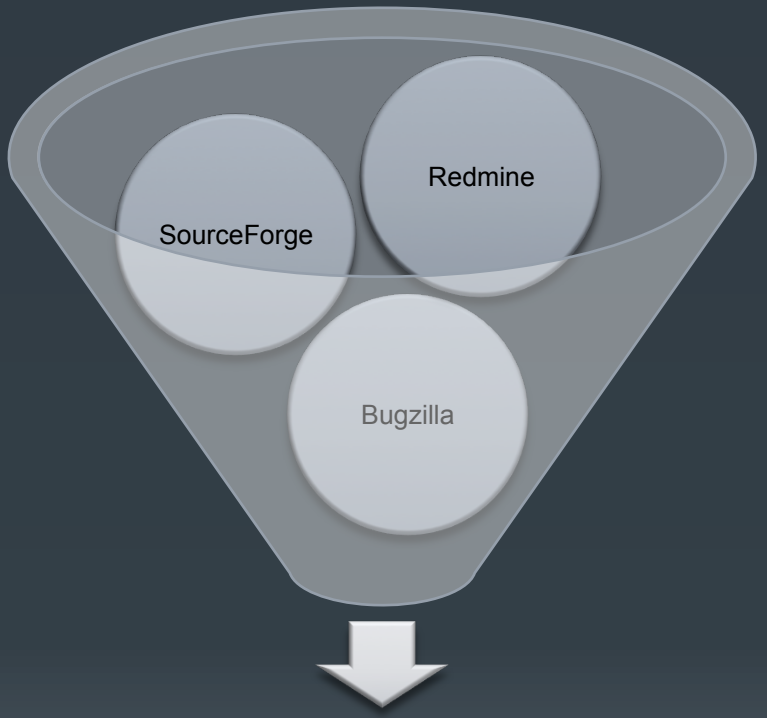
scenario

- researchers mine bug tracker to extract information
  - analyze the bugs
  - reference them to the original code
  - to predict failures in the code

problem

- different bug tracking systems
  - different structure and components
  - various scales of classification

problem



standardised representation of issues

problem

- changing API of bug trackers

problem


Bugzilla – Main Page
version 3.3.4+
Home | New | Search | Find | Reports | Requests | New Account | login password Log in
Welcome to Bugzilla
File a Bug
Search
Open a New Account
Enter a bug # or some search terms
Quick Search
Quick Search help | Install the Quick Search plugin
Bugzilla User's Guide | Release Notes
Home | New | Search | Find | Reports | Requests | New Account | login password Log in

problem


Bugzilla – Main Page
version 4.4+
Home | New | Browse | Search |
Search
[?] | Reports | New Account | Log In | Forgot Password
Welcome to Bugzilla
File a Bug
Search
Open a New Account
Enter a bug # or some search terms
Quick Search
Quick Search help | Install the Quick Search plugin
Bugzilla User's Guide | Release Notes
Home | New | Browse | Search |
Search
[?] | Reports | New Account | Log In | Forgot Password

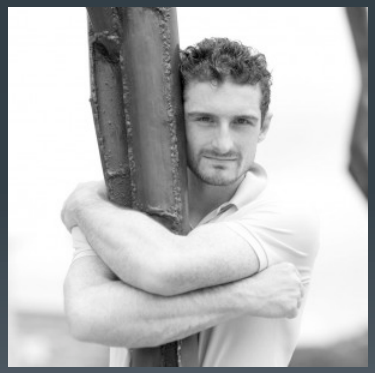
solution

- adaptive miner
  - standardized tracker
  - generates mining plan
  - independent of tracking system

standardized tracker

mozkito

Kim Herzig
Post-doc Researcher
Microsoft Research



Sascha Just
Software Engineering Chair
(Prof. zeller)

generate a mining plan

- mark standardized bug reports
- put them in a tracking system
- retrieve the markers
- create a mining plan

retrieve the information

- HTML surface of the bug tracker
- Crawler searches the markers
- have to remember the path to find

contribution

- standardized view of bug reports
- reception of a benchmark
- approach of adaptive mining

evaluation

- apply our method to the benchmark
- use three different kinds of tracking systems
  - SourceForge
  - JIRA
  - Bugzilla

evaluation

- choose 20 random bugs
- put them into three bug tracking systems
- compare the results after applying our tool with the original benchmark data

results

- perfect scenario
  - retrieve all information
  - aquivalence to the original data
- unsatisfying result

results

- What problems could appear:
  - don‘t retrieve all information of the original reports
  - only two of three bug tracking systems yield good statistics

future work

- on-the-fly production of bug tracker miners, given a way to deploy our created data set
- Analyzing Web Service Front-End Evolution
  - aspects like security and testing
- Interface Optimization and Data Traceability

summary

- problem of non-adaptive issue miners
- data set
- adaptive mining plan generator
- evaluation