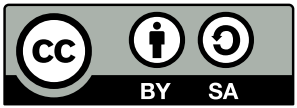
L01 Issue Tracking

Markus Raab

Institute of Information Systems Engineering, TU Wien



Issue Tracking

1. Issue Tracking
2. Bug Reporting
3. Bug Triage
4. Meeting
   - Preview

Learning Outcomes

After successful completion of L01 and H1
students will be able to

- remember terms and characteristics of issue tracking systems,
- report bugs,
- triage bugs.

Term: Issue

"Issue" is a very general term, it means nearly anything:

- a problem or bug
- a proposal
- a question
- a feature request
- a task
- a TODO list entry
- ...

Issue Tracking

Allows to:

- track the latest status of issues: open, resolved, ...
- discuss the issue (`@mention`)
- add semantics (metadata): tags, project, milestone, relationships, priorities, ...

Queries

Based on full text and/or semantic queries, issues on `https://issues.libelektra.org`:

- I created:
  `is:open is:issue author:@me`
- I am assigned to:
  `is:open is:issue assignee:@me`
- without assignee:
  `is:open is:issue no:assignee`
- not updated this year:
  `is:open is:issue updated:<2022`
- that have a label:
  `is:open is:issue label:floss2022W`
- to be fixed before 1.0:
  `is:open is:issue milestone:0.9.*`

Issue Tracking Systems

- text files with metadata in git
- conversationally-rich: Debbugs, GitLab or GitHub issues
- semantically-rich: e.g. Bugzilla
- broader scope: Redmine, Trac
- specialization: misconfiguration tracker

Unsuitable “Issue Tracking Systems”

- EMail
- Forums
- chats like IRC
- text files
- TODO markers
- ...

if without metadata or not in version control.

Interfaces

- Web
- EMail
- REST
- CLI tools, e.g. reportbug
- ...

Elektra has

- TODO files in doc/todo (12 files)
- TODO markers in source code (263 markers)
- `https://issues.libelektra.org` (217 open issues)

Automatic Closing of Issues

Ideally only fixed bugs would be closed but:

- issues become irrelevant
- maintainers disappear
- systems depreciate
- focus shifts
- ...

In Elektra issues+PR close after 365+14 days automatically, see `.github/stale.yml`.

The 249 issues still can be found via `is:closed is:issue label:stale`

Netiquette

- Never forget that you are talking to human beings.
- Be as careful, respectful and gentle as possible.
- Expect as little as possible.
- Only judge on technical issues, never on persons.
- There are no golden rules, cultures can disagree on everything.

Task

Do you agree with that list? Discuss your experiences.

Quoting

- If you want to reply to several points:

```
1 @ghost wrote:
2 > I wrote something
3 The answer
```

- If you reply to a statement given somewhere else:

```
1 @ghost wrote in [link to comment]:
2 > I wrote something
3 The answer
```

- Often no reply necessary if you only want to (dis)agree.

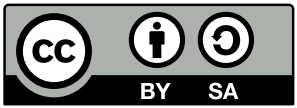
Best Practices

- First read attentively, then write.
- If in doubt: start a new issue.
- Split up issues that discuss unrelated problems.
- Prefer methods of automatic closing of issues.
- Fix issues you are assigned, ask for help or unassign if you give up.

L01 Issue Tracking

Markus Raab

Institute of Information Systems Engineering, TU Wien



Bug Reporting

1 Issue Tracking

2 Bug Reporting

3 Bug Triage

4 Meeting
- Preview

First Steps in Bug Reports

Make sure that:

- You use the correct issue tracker.
- You read about how to use that issue tracker.
- Use specialized helper programs, if available, like reportbug.

Steps to Reproduce the Problem

- be precise
- be clear
- be complete
- ideally syntax of tests

$\rightarrow$ verify yourself

Actual Result

- describe the symptoms
- avoid opinions or conclusions here
- describe what you see

Expected Result

- how you would like the software to behave
- suggestions how to solve the problem

System Information

- version or sha of commit
- include errors, logs, etc.
- operating system or docker container
- versions of other relevant software

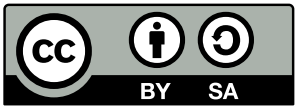
Best Practices

- learn about the community in guidelines
- always include symptoms, separate diagnosis
- reproduce using your own report
- sometimes an incomplete report can be better than no report
- reply to further questions

L01 Issue Tracking

Markus Raab

Institute of Information Systems Engineering, TU Wien



Bug Triage

1. Issue Tracking
2. Bug Reporting
3. Bug Triage
4. Meeting
   - Preview

Reproduce the Problem

Try to do what is described in the issue, possible problems:

- There is an error in the description, e.g. a wrong command.
- The description is missing essential steps to do.
- The issue is already fixed or otherwise outdated.

$\rightarrow$ Fix such problems in the issue description!

Locate Problem in Source Code

- Via error messages: `kdb -vd`
- Via debugger or backtrace:
  - Additionally install `-dbgsym` packages.
  - Even better compile with: `ENABLE_DEBUG`.
- Via logger:
  - Compile with `ENABLE_LOGGER`.
  - Modify `src/libs/elektra/log.c` as needed.

System Information

Hints about further affected systems, e.g.,

- information about your system
- version information
- programming language

might further triage the bug, i.e., help the person working on it.

Best Practices

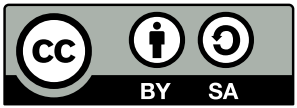
Bug triage

- makes fixing bugs easier.
- can help to find the right person to fix a task.
- is needed for imprecise unclear or incomplete issues
  $\rightarrow$ avoid this in the first place

L01 Issue Tracking

Markus Raab

Institute of Information Systems Engineering, TU Wien



Meeting

1 Issue Tracking

2 Bug Reporting

3 Bug Triage

4 Meeting
- Preview

L02 Source Code Management

[1] Markus Raab and Gergö Barany. Introducing context awareness in unmodified, context-unaware software. In *Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE,*, pages 218–225. INSTICC, ScitePress, 2017. ISBN 978-989-758-250-9. doi: 10.5220/0006326602180225.