One tool to rule them all?

Integration or survival of the fittest

Carina Haupt

German Aerospace Center (DLR)
Department Intelligent and Distributed Systems
Berlin / Braunschweig / Cologne

Research Software Engineering
Conference 2017

Software Development @ DLR
Some numbers

Some numbers

- More than 8000 employees
- More than 1500 employees develop software
- More than 150 Mio EUR personnel costs per year for software development

→ DLR is one of the biggest „software houses“ in Germany

Characteristics

- „Developer“ often do not have any training in software development
- Variety of used software technologies
    - i.E. over 30 programming languages

Where we started

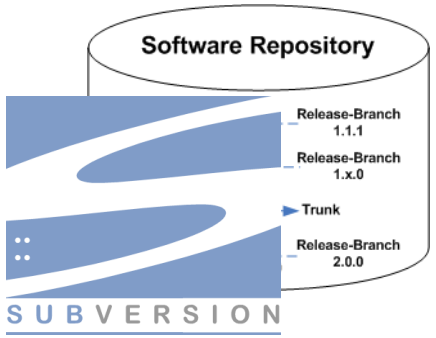
Software Engineering Tools @ DLR

Team

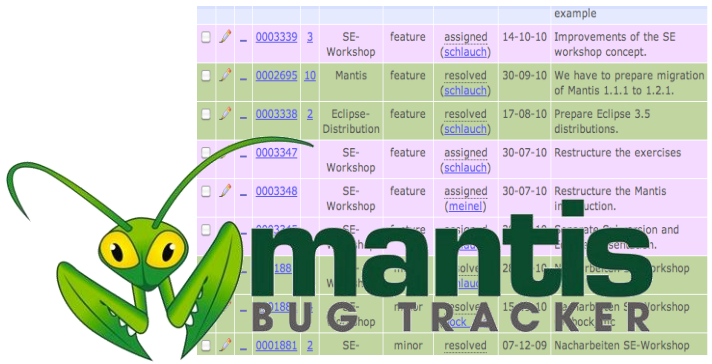
planning

changes

Version Control System

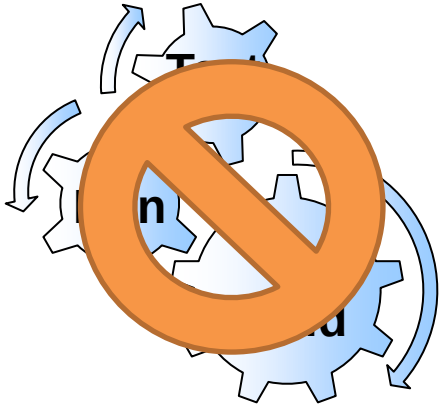


quality checks



Issue Tracker

feedback



Integration-Server

The Naive Approach
Survival of the fittest

cvs

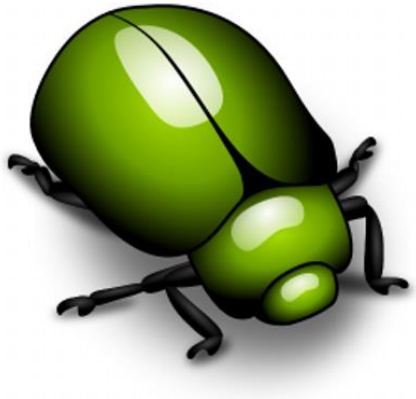
Bazaar

mantis BUG TRACKER

FogBugz

SUBVERSION

Find the best tool for each task



git

mercurial

JIRA

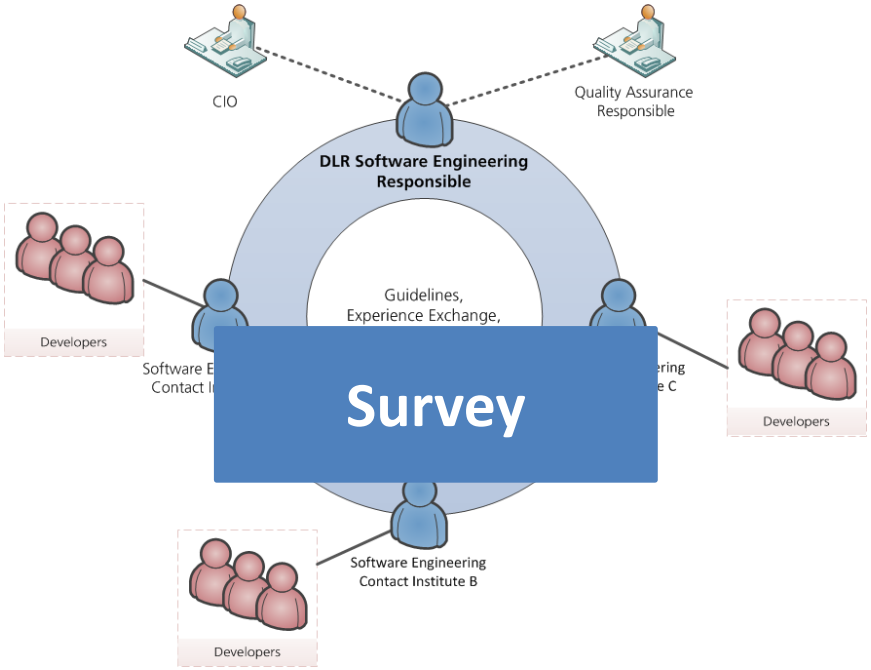
GEMINI

But wait…perhaps lets ask the users first!

The study

Which ressources do we have to access „users“?

DLR SE-Workshops: Yearly meeting of SE interested employees



DLR SE-Network: Each institute has one point of contact for all SE needs

Personal network

What the users said
Characterisation of development

Aimed project quality
- ~ 50 % basic
- ~ 30 % medium
- ~ 20 % high

Team sizes
- Mainly one person projects
- Similar amount of small (2-6 person) and medium sized teams (6-20 person)
- Few big teams (> 20 person)

- **Currently used tools**
  - Mainly SVN & Mantis
  - Self hosted alternatives (Git, Jira)
  - Additional tooling (mainly Jenkins for CI)

What the users said
Requirements (extract)

- High level **overview** of project
  - Different views for different roles
- Planning possibilities (roadmap, backlog, sprints)
- Version control system
  - Code collaboration
  - Merge support
- Issue tracking
- Documentation
- Automatization (Continuous Integration, Deployment)
- **Integration** of issues, documentation, code
- IDE integration
- Access control (external users)
- **Easy and fast setup**
- **Easy to use**

What the users said
Our results

Requirements

- Easy to setup
- Easy to use
- Easy to learn
- Highly integrated tools

To be covered functionality

- Source code repository
- Issue tracking
- Documentation
- Continous Integration
- Reports (for project head)

Target group

- Projects aiming for basic or medium quality
  → High quality projects set up their own tools
- One person and small sized teams
  → Have no resources for overhead

What the users said
Our results

What we learned

- „Easy“ and integration beats missing functionality
- If it is not easy to set up, people will not use it
- If it is not easy to use, projects will not keep using it

Our decision

→ We go with a simple integrated software engineering tool suite

ASSEMBLA

GitHub

PHABRICATOR

RhodeCode®

GitLab

trac
Integrated SCM & Project Management

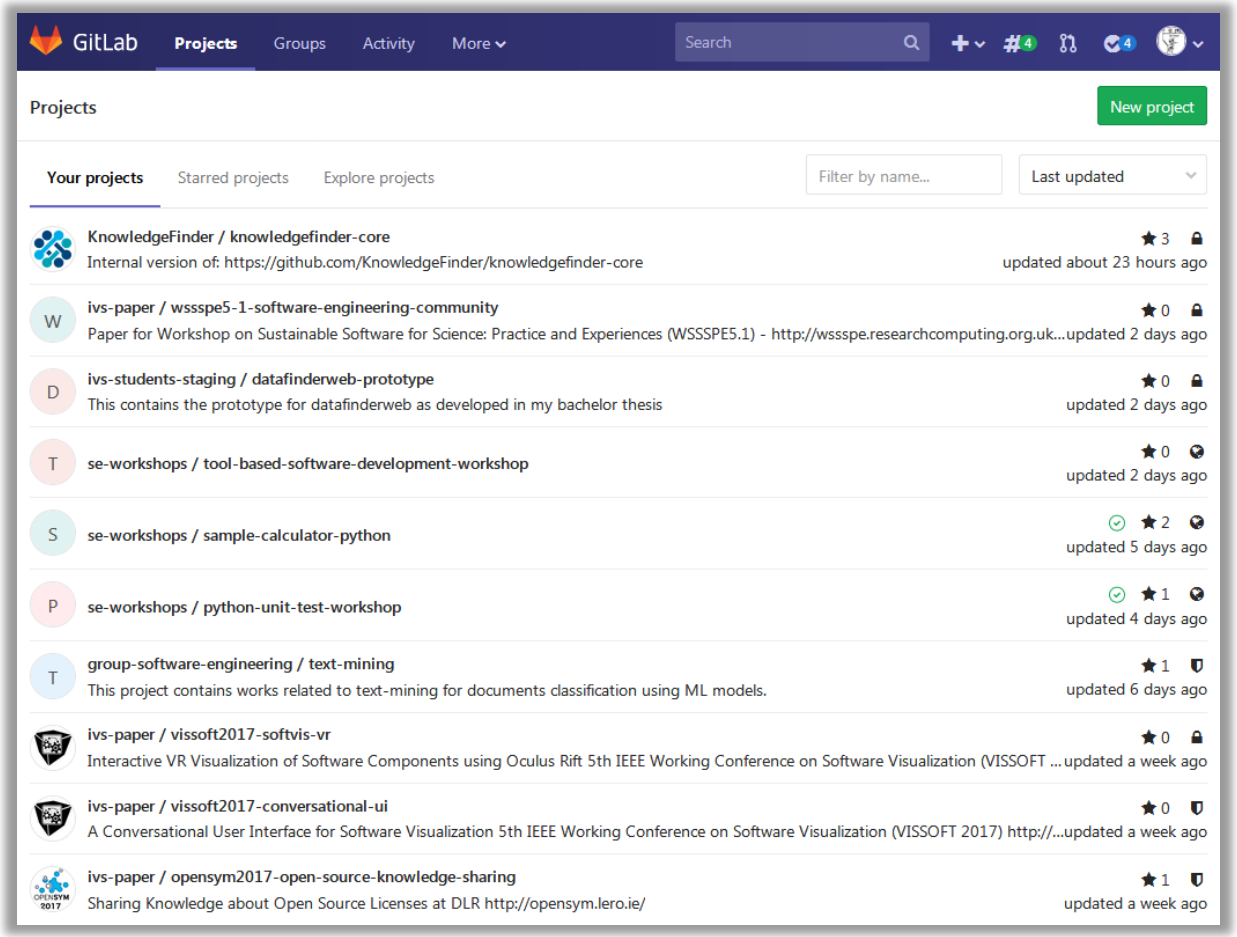
REDMINE
flexible project management

The New Approach
One tool to rule them all

Source Code Repository
Issue Tracker
Documentation
CI

GitLab

+ JIRA



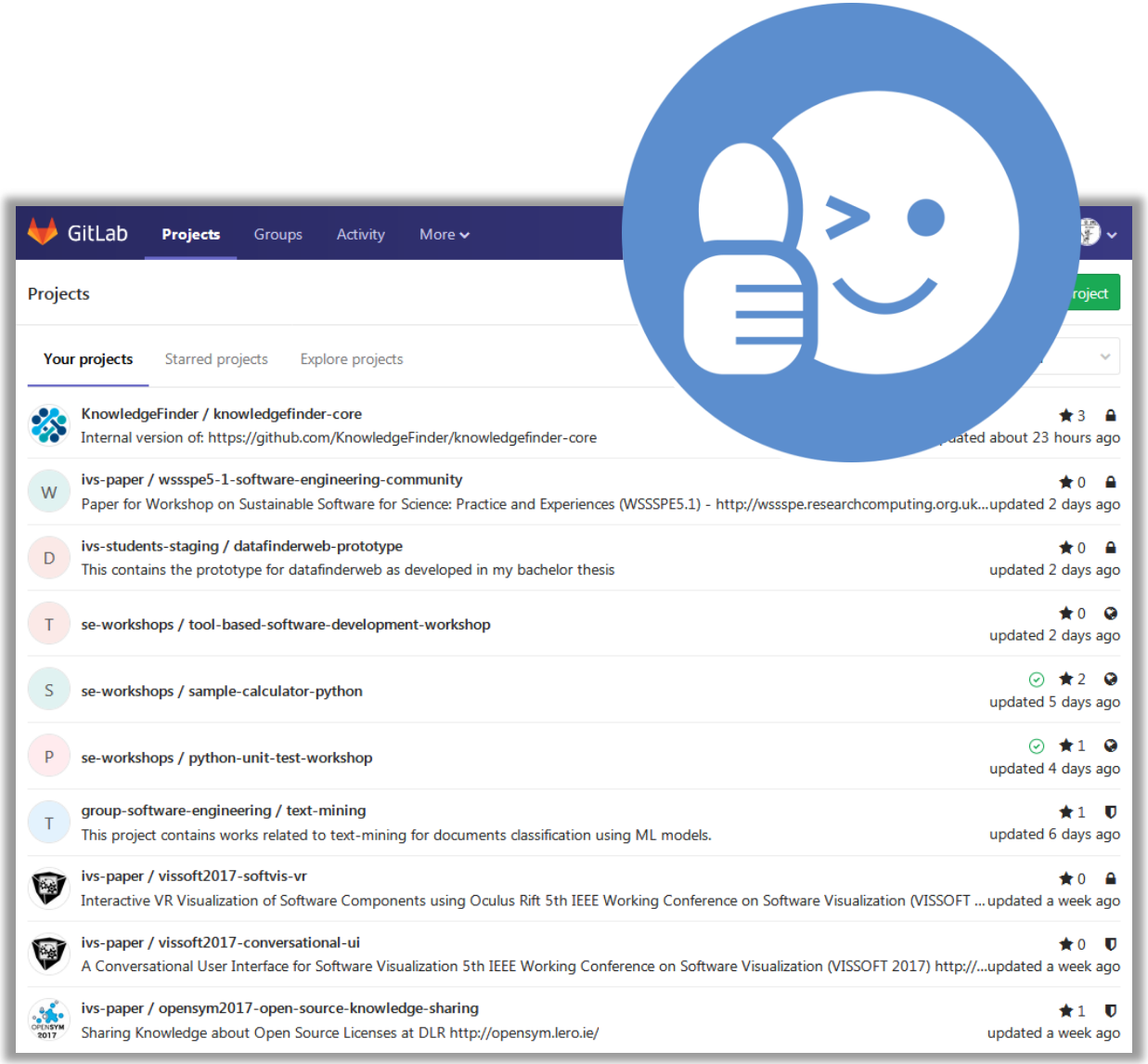
DLR

The Feedback
What our test group said

- Testversion at facility „Simulation and Software Technology“

- Stats (04.09.2017)
  - 117 active users
  - 48 GitLab groups
  - 388 GitLab projects
  - 266 mile stones
  - 2438 issues
  - 19892 comments
  - 3354 builds

- Users
  - Everybody for everything



Questions?