The Software Engineering Community at DLR
How we got where we are

Carina Haupt, Tobias Schlauch

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

Workshop on Sustainable Software for Science: Practice and Experiences 5.1

German Aerospace Center (DLR)

Numbers

- More than 8000 employees
- ~20% of DLR employees involved in software development

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

Characteristics

- Variety of
  - Fields
  - Maturity
  - Software technologies
  - Team sizes
- “Developers” often do not have any training in software development

German Aerospace Center (DLR)

Numbers

- More than 8000 employees
- ~20% of DLR employees involved in software development

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

Characteristics

- Variety of
  - Fields
  - Maturity
  - Software technologies
  - Team sizes
- “Developers” often do not have any training in software development

Goal: Improve sustainability and quality of software products

German Aerospace Center (DLR)

Numbers

- More than 8000 employees
- ~20% of DLR employees involved in software development

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

Characteristics

- Variety of
  - Fields
  - Maturity
  - Software technologies
  - Team sizes
- “Developers” often do not have any training in software development

Goal: Improve sustainability and quality of software products

How to teach them software engineering?

Software Engineering Initiative of DLR

Software Engineering Initiative of DLR

- Guidelines
- Trainings
- Knowledge Provision
- Collaboration
- Experience Exchange

Software Engineering Guidelines

Guidelines support developers to self-assess their software concerning good development practices.

- Joint development with focus on **good practices**, **tools**, and **essential documentation**
- **Three maturity level** available as **checklists in different formats** to ease practical usage

Checklists for different maturity levels

Change Management

Recommendation	Comment	Status
EÄM.2: The most important information describing how to contribute to development are stored in a central location. *(from application class 1)*	Build steps are missing	todo
EÄM.5: Known bugs, important unresolved tasks and ideas are at least noted in bullet point form and stored centrally. *(from application class 1)*		ok
EÄM.7: A repository is set up in a version control system. The repository is adequately structured and ideally contains all artifacts for building a usable software version and for testing it. *(from application class 1)*		ok
EÄM.8: Every change of the repository ideally serves a specific purpose, contains an understandable description and leaves the software in a consistent, working state. *(from application class 1)*		ok

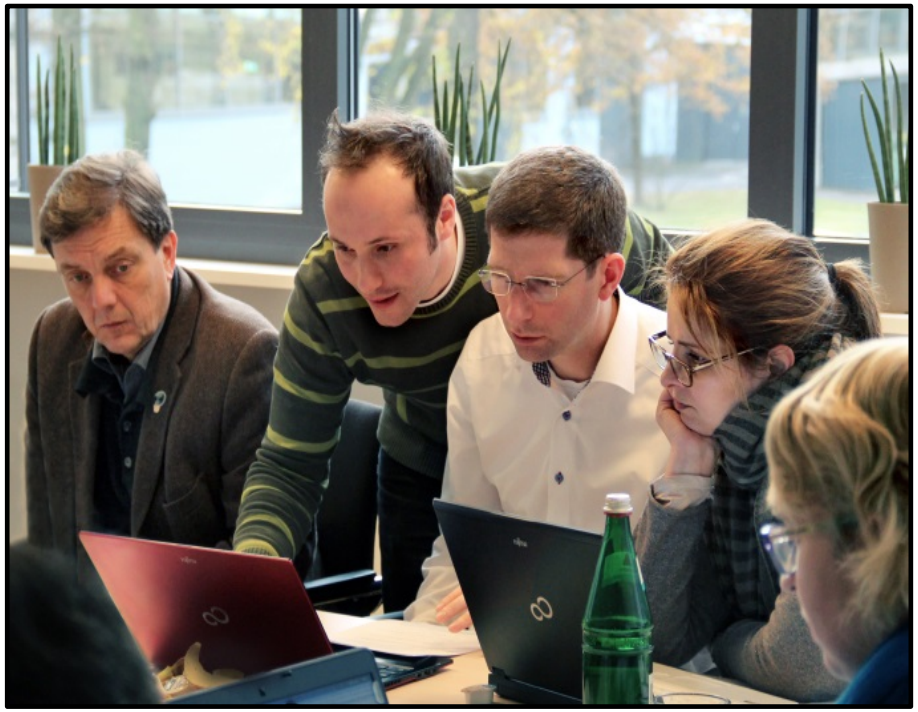
Reasoning and further advice

The repository is the central entry point for development. All main artifacts are stored in a safe way and are available at a single location. Each change is comprehensible and can be traced back to the originator. In addition, the version control system ensures the consistency of all changes.

The repository directory structure should be aligned with established conventions. References are usually the version control system, the build tool (see the Automation and Dependency Management section) or the community of the used programming language or framework. Two examples:

Trainings

Regular trainings are offered to provide hands-on experience in applying the guidelines and the DLR development tools.



Concept

- Intensive two-day course
- Small groups with up to 15 participants
- Hands-on experience on the basis of a complete example project using DLR provided tools
- Trainings are offered on a yearly basis at different DLR locations across Germany

Additional trainings are offered on request for specific topics such as unit testing, open source, and others.

Knowledge Provision and Collaboration
SoftwareEngineering.Wiki

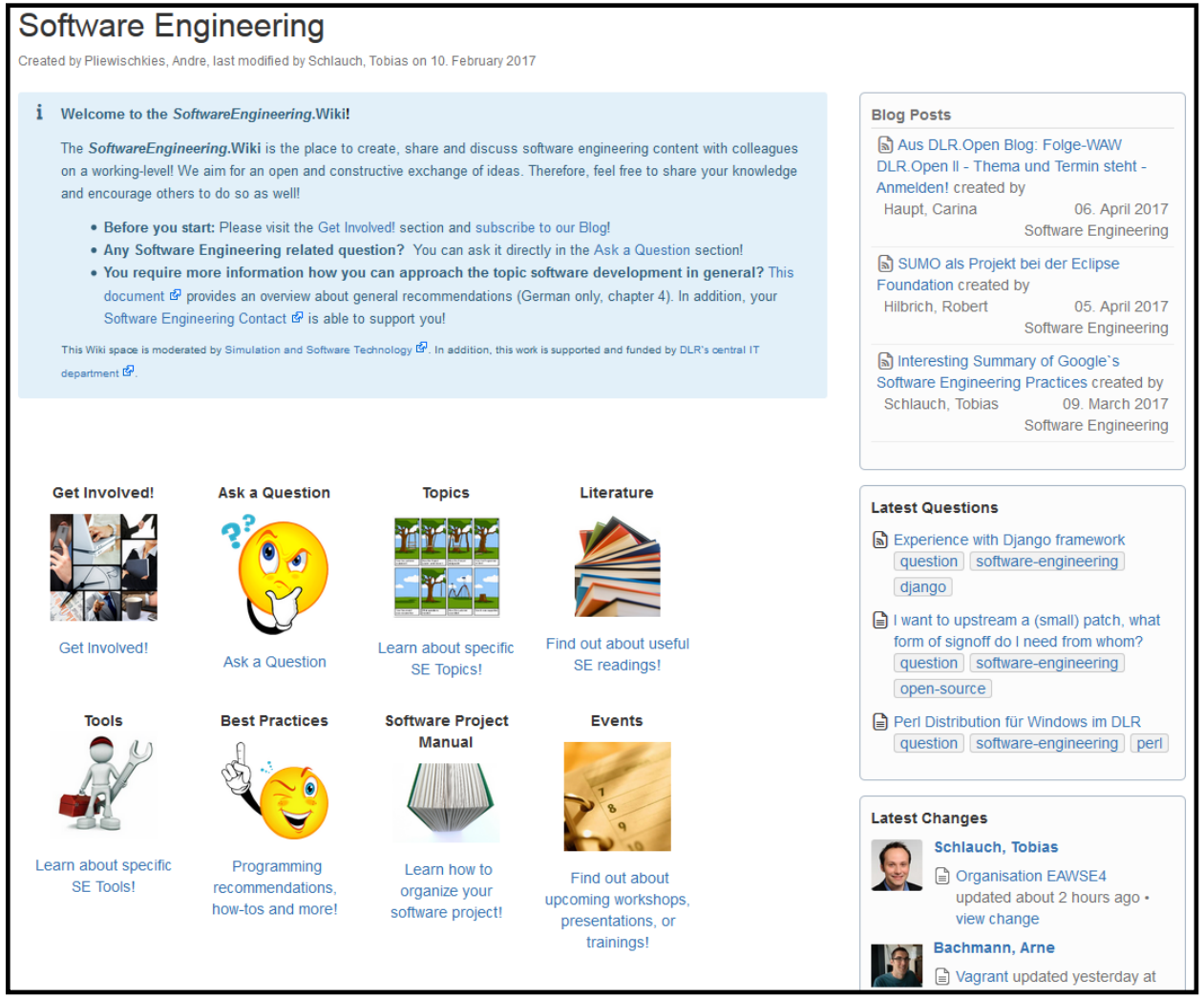
Internal Wiki space to share software engineering knowledge and experiences.

Concept

- Open to contributions of all DLR employees
- Moderation by a small central group

Main content categories

- News
- Information about topics like architecture, testing, etc.
- Official programming guides
- Experiences concerning development tools
- Questions & answers



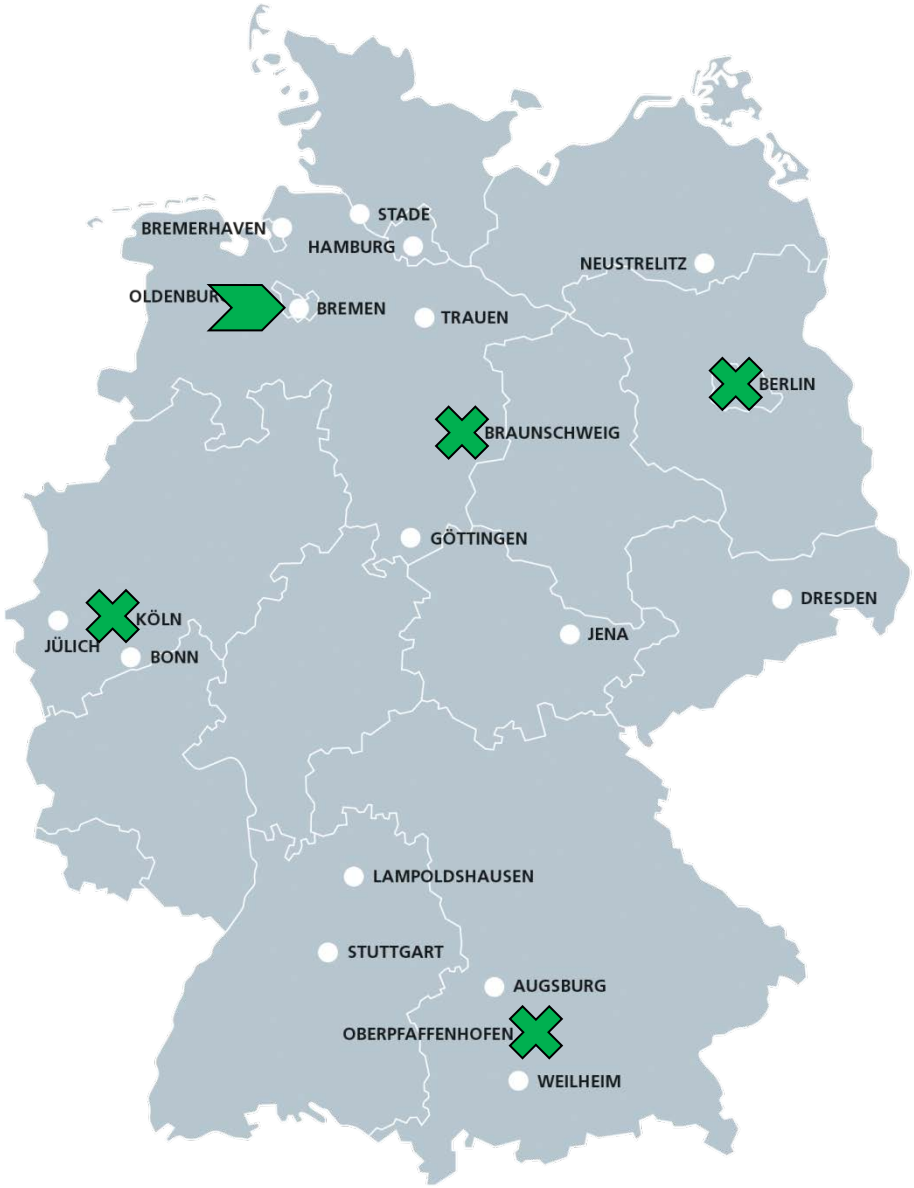
Experience Exchange
Workshops

Regular knowledge exchange workshops are held to actively involve DLR scientists and to foster exchange.

Concept

- Intensive 1.5-day workshop to provide *knowledge*, *experience exchange* and *networking opportunities*
- Dedicated main topic supported by keynotes of invited experts
- Active involvement of the participants through group work, experience reports, technical presentations, and lightning talks
- Results are shared via the *SoftwareEngineering.Wiki*

Since 2014, four knowledge exchange workshops have been organized at different locations across Germany. About 50 scientists participated in every workshop.



Summary and Outlook

First steps have been taken to build a self-reliant software engineering community at DLR.

Key success factors

- Establishment of a vital software engineering core community
- Joint development of practical software development guidelines
- Raising management awareness and achieving management support
- Wholesome support of domain scientist and DLR institutes

Next steps

- Strengthen community (exchange, “inner source”)
- Provide further, community-driven solutions to ease implementation of guidelines

Questions?

If you have any more questions, just let us know.

More Information can be found in our paper.

Or just ask us here, or via mail:

carina.haupt@dlr.de

tobias.schlauch@dlr.de