

Empfehlungen für bessere Forschungssoftware

Tobias Schlauch

Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Einrichtung Simulations- und Softwaretechnik
Berlin / Braunschweig / Köln / Oberpfaffenhofen

deRSE19, 04.-06.06.2019, Potsdam



Wissen für Morgen



Software-Entwicklung im DLR

- Circa 8.200 Mitarbeiter*innen
- Geschätzte 20% davon, sind mit Software-Entwicklung befasst
- **Software-Entwicklung im DLR sehr unterschiedlich, je nach:**
 - Domäne
 - Reifegrad
 - Programmiertechnologie
 - Teamgröße



DLR Software-Engineering-Initiative

DLR Software-Engineering-Initiative

Policy-
Entwicklung

Empfehlungen
und
Werkzeuge

Schulung
und
Beratung

Wissens-
und
Erfahrungsaustausch



Software-Engineering-Empfehlungen des DLR

Checkliste

Change Management

Recommendation	Comment	Status
EÄM.2: The most important information describing how to contribute to development are stored in a central location. <i>(from application class 1)</i>	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. <i>(from application class 1)</i>		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. <i>(from application class 1)</i>		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. <i>(from application class 1)</i>		ok

Begründung und weitere Hinweise

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:



Auf den Kontext kommt es an!

Anwendungsklasse 1

- „klein“, aber andere nutzen es (auch)

Anwendungsklasse 2

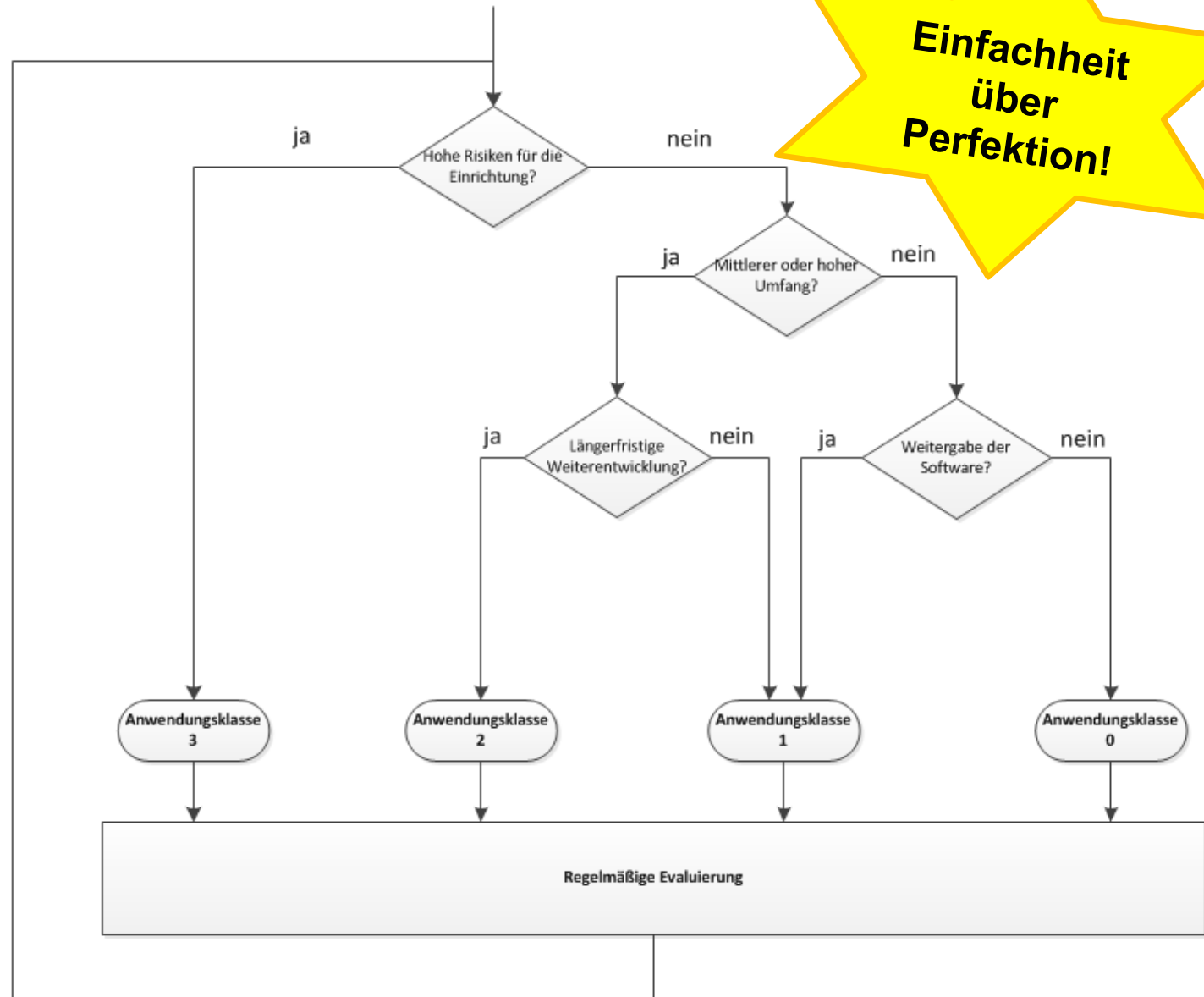
- „mittel bis groß“, andere nutzen es, längerfristige Unterstützung und Wartung

Anwendungsklasse 3

- produktartig, kritisch für den Erfolg

Anwendungsklasse 0

- „Persönliche“ Nutzung (bewusst offen gelassen)



Auf den Kontext kommt es an!

Anwendungsklasse 1

- „klein“, aber andere nutzen es (auch)

Anwendungsklasse 2

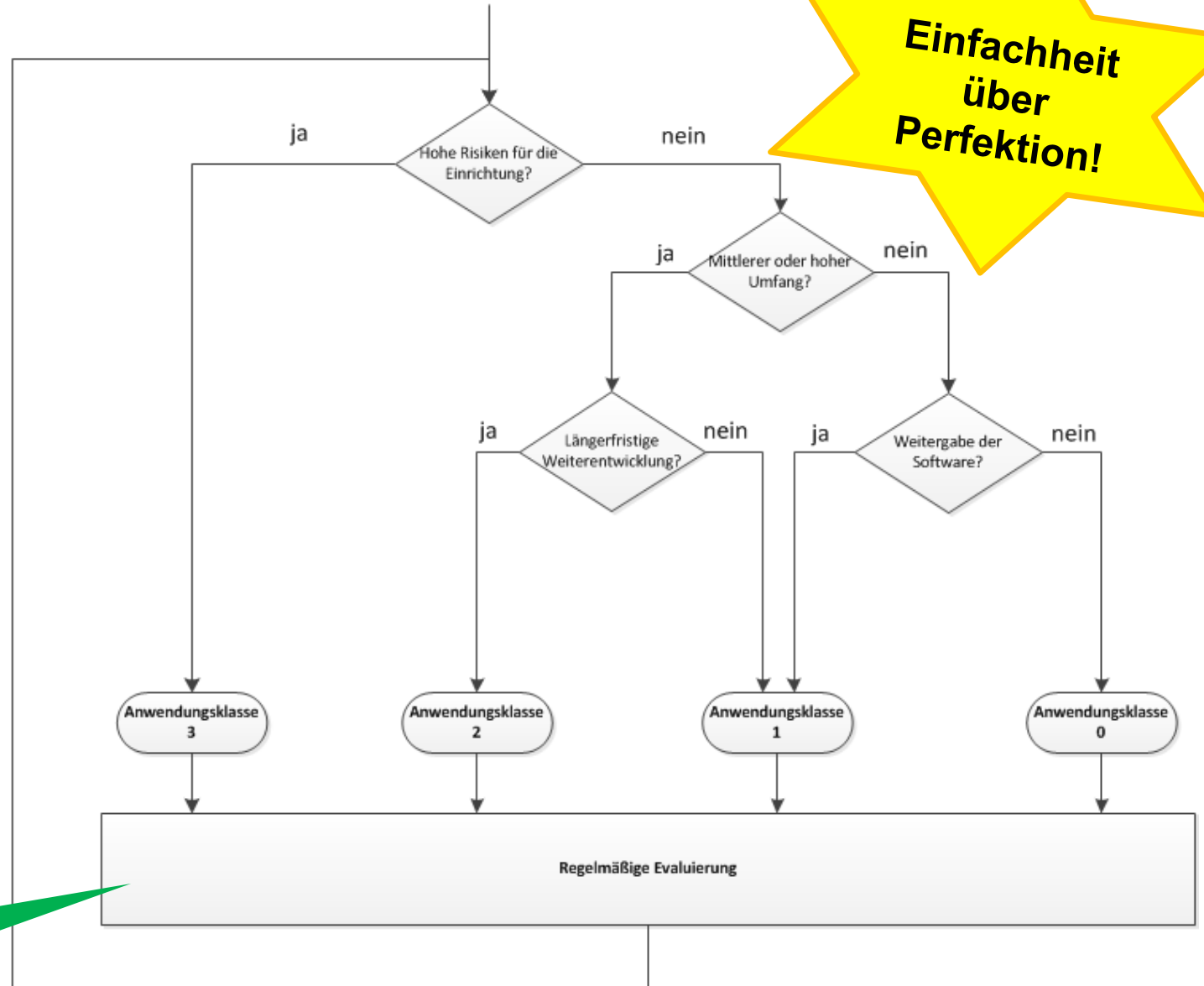
- „mittel bis groß“, andere nutzen es, längerfristige Unterstützung und Wartung

Anwendungsklasse 3

- produktartig, kritisch für den Erfolg

Anwendungsklasse 0

- „Persönliche“ Nutzung (bewusst offen gelassen)



Zuordnung kann sich ändern!



Beispiel für Anwendungsklasse 1

Generische Empfehlungen

1. Nutzung eines Versionskontrollsystem
2. Einhaltung grundlegender Entwicklungspraktiken
3. Bereitstellung essentieller Dokumentation
4. Erstellung einer nutzbaren Version ist möglichst automatisiert
5. Interner Release:
 - a) Test des Softwarestands
 - b) Kennzeichnung durch Release-Nummer
6. Öffentlicher Release: Einhaltung der DLR Open Source Guidelines

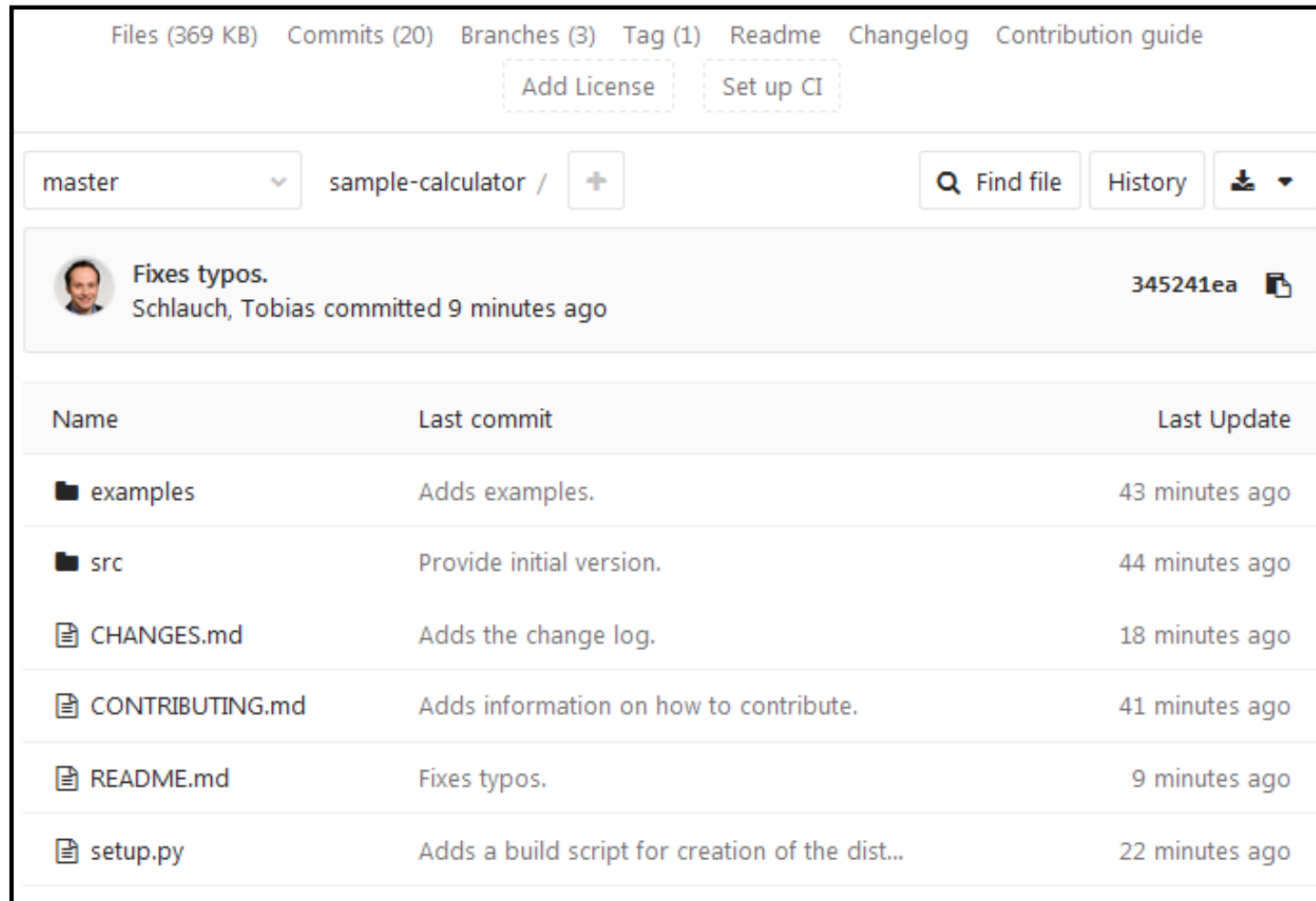
Anpassung an den
konkreten
Entwicklungskontext
erforderlich!



Beispiel für Anwendungsklasse 1

Mögliche Umsetzung am Beispiel (1/2)

Git Repository für
Code, Beispiele,
Build-Skript, und
Dokumentation



The screenshot shows a GitHub repository page for 'sample-calculator'. At the top, there are navigation links for 'Files (369 KB)', 'Commits (20)', 'Branches (3)', 'Tag (1)', 'Readme', 'Changelog', and 'Contribution guide'. Below these are buttons for 'Add License' and 'Set up CI'. The repository name 'sample-calculator' is shown with a dropdown menu set to 'master'. A search bar 'Find file' and a 'History' link are also visible. A recent commit by Tobias Schlauch is highlighted, with the message 'Fixes typos.' and the commit hash '345241ea'. Below this is a table listing the repository's files and folders, including 'examples', 'src', 'CHANGES.md', 'CONTRIBUTING.md', 'README.md', and 'setup.py', along with their last commit messages and update times.

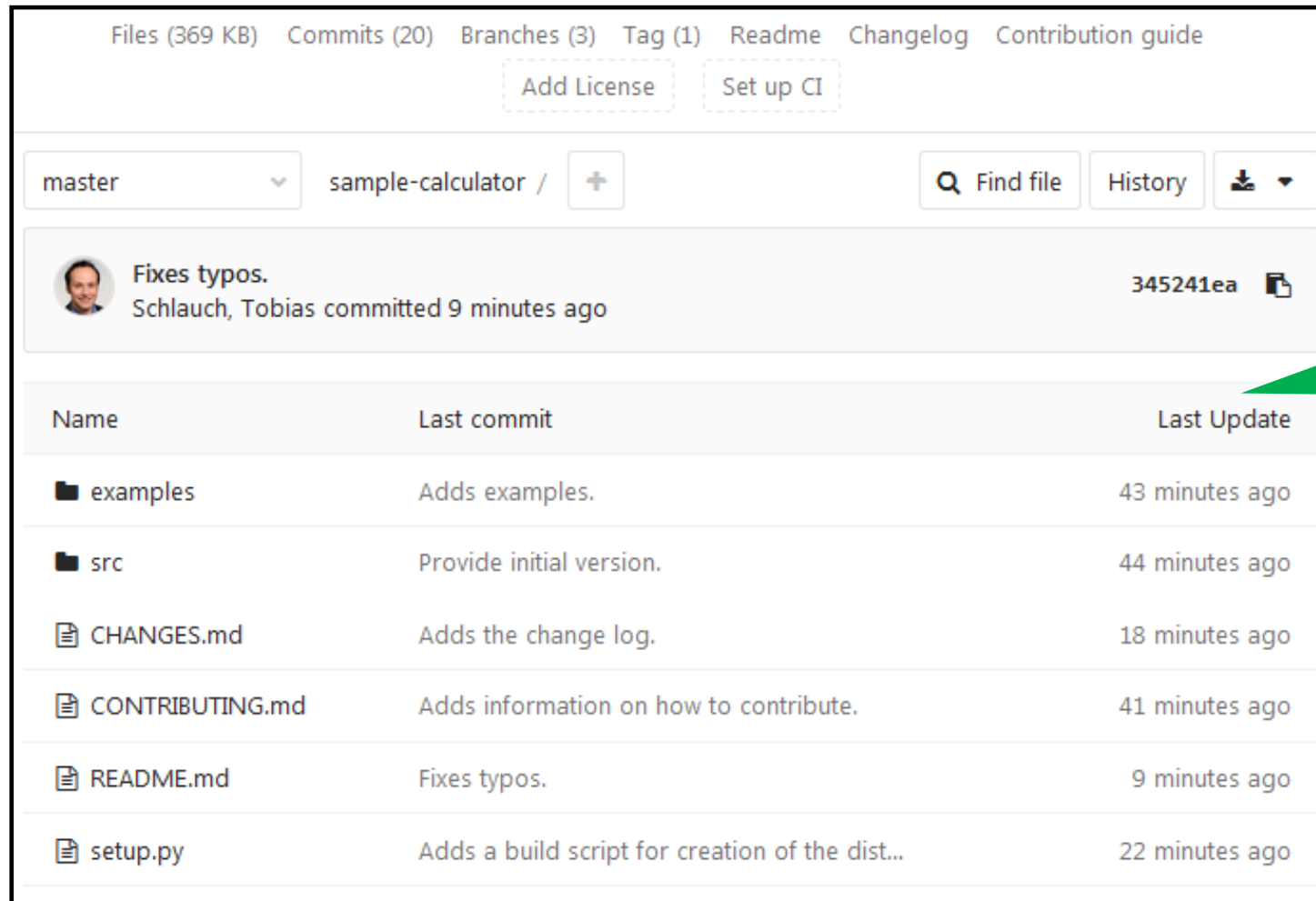
Name	Last commit	Last Update
examples	Adds examples.	43 minutes ago
src	Provide initial version.	44 minutes ago
CHANGES.md	Adds the change log.	18 minutes ago
CONTRIBUTING.md	Adds information on how to contribute.	41 minutes ago
README.md	Fixes typos.	9 minutes ago
setup.py	Adds a build script for creation of the dist...	22 minutes ago



Beispiel für Anwendungsklasse 1

Mögliche Umsetzung am Beispiel (1/2)

Git Repository für
Code, Beispiele,
Build-Skript, und
Dokumentation



Files (369 KB) Commits (20) Branches (3) Tag (1) Readme Changelog Contribution guide

Add License Set up CI

master sample-calculator / Find file History

Fixes typos.
Schlauch, Tobias committed 9 minutes ago 345241ea

Name	Last commit	Last Update
examples	Adds examples.	43 minutes ago
src	Provide initial version.	44 minutes ago
CHANGES.md	Adds the change log.	18 minutes ago
CONTRIBUTING.md	Adds information on how to contribute.	41 minutes ago
README.md	Fixes typos.	9 minutes ago
setup.py	Adds a build script for creation of the dist...	22 minutes ago

- Code in kleine Funktionen unterteilt
- PEP8 Kodierempfehlungen angewendet
- Beispiele zeigen übliche Eingaben und Ausgaben

Beispiel für Anwendungsklasse 1

Mögliche Umsetzung am Beispiel (1/2)

Files (369 KB) Commits (20) Branches (3) Tag (1) Readme Changelog Contribution guide

Add License Set up CI

master sample-calculator / Find file History

Fixes typos.
Schlauch, Tobias committed 9 minutes ago 345241ea

Name	Last commit	Last Update
examples	Adds examples.	43 minutes ago
src	Provide initial version.	44 minutes ago
CHANGES.md	Adds the change log.	18 minutes ago
CONTRIBUTING.md	Adds information on how to contribute.	41 minutes ago
README.md	Fixes typos.	9 minutes ago
setup.py	Adds a build script for creation of the dist...	22 minutes ago

Git Repository für
Code, Beispiele,
Build-Skript, und
Dokumentation

- Code in kleine Funktionen unterteilt
- PEP8 Kodierempfehlungen angewendet
- Beispiele zeigen übliche Eingaben und Ausgaben

- Build-Skript zur Paketierung und Installation
- Release-Nummern orientieren sich an Semantic Versioning
- CHANGES.md beschreibt wesentliche Änderungen aus Nutzersicht

Beispiel für Anwendungsklasse 1

Mögliche Umsetzung am Beispiel (2/2)

What is SampleCalculator?

SampleCalculator is a command line tool to calculate characteristic values of a sample.

It provides the following features:

- Reading sample values from command line and CSV (Colon Separated Values) files.
- Calculation of average, variance, and standard deviation.
- Configurable logging of results and interim results.
- Easy integration of new input sources
- Extensible by easily adding new calculations

SampleCalculator targets **scientists** who want to easily perform such calculations as part of their workflow and **Python developers** who want to integrate the functionalities into their software. We implemented as we have not found a suitable, zero-dependency alternative.

The current version is only an initial alpha version which is **NOT** suited for production use. Particularly, it is not sufficiently tested with large data sets. It requires **Python >= 3.4** and has been only tested on **Windows 7** so far. However, it should basically work on operating system.

How can I install it?

- Make sure that you use Python >= 3.4
- Download the [latest package](#)
- Extract it to a directory

- [README.md](#) enthält die Hauptdokumentation
- [CONTRIBUTING.md](#) stellt Informationen für Beitragende bereit



Beispiel für Anwendungsklasse 1

Mögliche Umsetzung am Beispiel (2/2)

What is SampleCalculator?

SampleCalculator is a command line tool to calculate characteristic values of a sample.

It provides the following features:

- Reading sample values from command line and CSV (Colon Separated Values) files.
- Calculation of average, variance, and standard deviation.
- Configurable logging of results and interim results.
- Easy integration of new input sources
- Extensible by easily adding new calculations

SampleCalculator targets **scientists** who want to easily perform such calculations as part of their workflow and **Python developers** who want to integrate the functionalities into their software. We implemented as we have not found a suitable, zero-dependency alternative.

The current version is only an initial alpha version which is **NOT** suited for production use. Particularly, it is not sufficiently tested with large data sets. It requires **Python** ≥ 3.4 and has been only tested on **Windows 7** so far. However, it should basically work on operating system.

How can I install it?

- Make sure that you use Python ≥ 3.4
- Download the [latest package](#)
- Extract it to a directory

- README.md enthält die Hauptdokumentation
- CONTRIBUTING.md stellt Informationen für Beitragende bereit

- Beschreibung des Zwecks der Software (Was? Für wen? Warum?)
- Überblick der Hauptfunktionen
- Wichtige Nutzungsrandbedingungen



Beispiel für Anwendungsklasse 1

Mögliche Umsetzung am Beispiel (2/2)

What is SampleCalculator?

SampleCalculator is a command line tool to calculate characteristic values of a sample.

It provides the following features:

- Reading sample values from command line and CSV (Colon Separated Values) files.
- Calculation of average, variance, and standard deviation.
- Configurable logging of results and interim results.
- Easy integration of new input sources
- Extensible by easily adding new calculations

SampleCalculator targets **scientists** who want to easily perform such calculations as part of their workflow and **Python developers** who want to integrate the functionalities into their software. We implemented as we have not found a suitable, zero-dependency alternative.

The current version is only an initial alpha version which is **NOT** suited for production use. Particularly, it is not sufficiently tested with large data sets. It requires **Python >= 3.4** and has been only tested on **Windows 7** so far. However, it should basically work on operating system.

How can I install it?

- Make sure that you use Python >= 3.4
- Download the [latest package](#)
- Extract it to a directory

- README.md enthält die Hauptdokumentation
- CONTRIBUTING.md stellt Informationen für Beitragende bereit

- Beschreibung des Zwecks der Software (Was? Für wen? Warum?)
- Überblick der Hauptfunktionen
- Wichtige Nutzungsrandbedingungen

- Grundlegende Nutzungs- und Installationsinformationen
- Zukünftige Pläne und Ideen



Lessons Learned

SE-Empfehlungen helfen in Bezug auf eine Software, den aktuellen Zustand zu bestimmen, Verbesserungen zu identifizieren sowie Aktivitäten und Diskussionen zu fokussieren, aber

... ist kein Werkzeug für Unerfahrene und einige Details müssen noch verbessert werden:

- Bessere Kennzeichnung von Prioritäten
- Transparente Darstellung von Abhängigkeiten
- Mehr praktische Umsetzungsbeispiele



["A Good Service Just Turned Into Better"](#) by [Abu Zafor](#) is licensed under [CC BY-SA 2.0](#)



Lessons Learned

SE-Empfehlungen helfen in Bezug auf eine Software, den aktuellen Zustand zu bestimmen, Verbesserungen zu identifizieren sowie Aktivitäten und Diskussionen zu fokussieren, aber

... ist kein Werkzeug für Unerfahrene und einige Details müssen noch verbessert werden:

- Bessere Kennzeichnung von Prioritäten
- Transparente Darstellung von Abhängigkeiten
- Mehr praktische Umsetzungsbeispiele

Unterstützende Umgebung ist entscheidend!

- Community / Teamkultur / Policy
- Werkzeuge / Trainings



["A Good Service Just Turned Into Better" by Abu Zafor](#) is licensed under [CC BY-SA 2.0](#)



Umgebung im DLR

SE-Netzwerk / WAWs / Wiki

Software Engineering

Created by Pliewischkies, Andre, last modified by Schlauch, Tobias on 10. February 2017

Welcome to the *SoftwareEngineering.Wiki*!

The *SoftwareEngineering.Wiki* is the place to create, share and discuss software engineering content with colleagues on a working-level! We aim for an open and constructive exchange of ideas. Therefore, feel free to share your knowledge and encourage others to do so as well!

- **Before you start:** Please visit the *Get Involved!* section and subscribe to our Blog!
- **Any Software Engineering related question?** You can ask it directly in the *Ask a Question* section!
- **You require more information how you can approach the topic software development in general?** This document provides an overview about general recommendations (German only, chapter 4). In addition, your *Software Engineering Contact* is able to support you!

This Wiki space is moderated by *Simulation and Software Technology*. In addition, this work is supported and funded by DLR's central IT department.

Blog Posts

- Aus DLR Open Blog: Folge-WAW DLR Open II - Thema und Termin steht - Anmelden! created by Haupt, Carina 06. April 2017 Software Engineering
- SUMO als Projekt bei der Eclipse Foundation created by Hilbrich, Robert 05. April 2017 Software Engineering
- Interesting Summary of Google's Software Engineering Practices created by Schlauch, Tobias 09. March 2017 Software Engineering

Get Involved!



Get Involved!

Ask a Question



Ask a Question

Topics



Learn about specific SE Topics!

Literature



Find out about useful SE readings!

Tools



Learn about specific SE Tools!

Best Practices



Programming recommendations, how-tos and more!

Software Project Manual

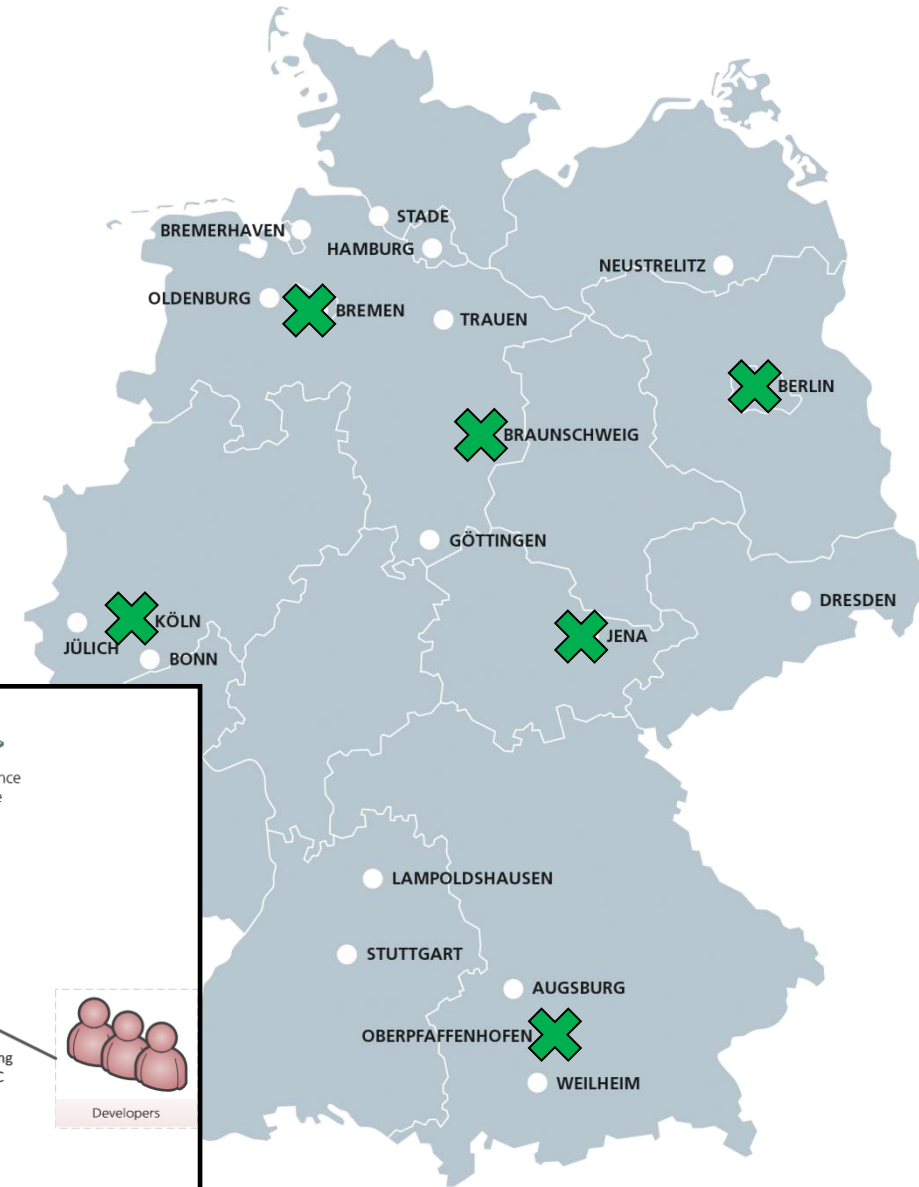
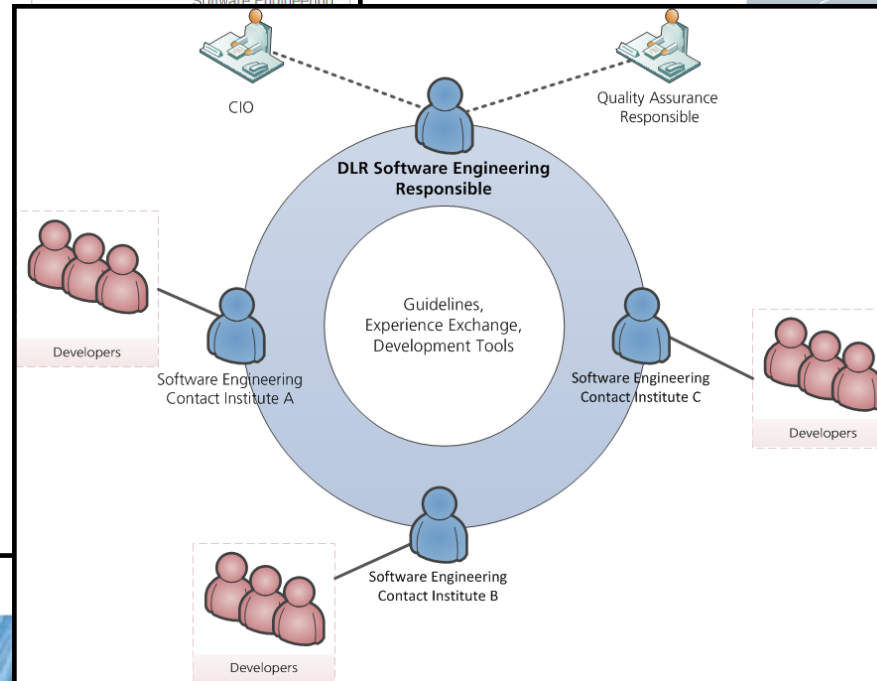


Learn how to organize your software project!

Events

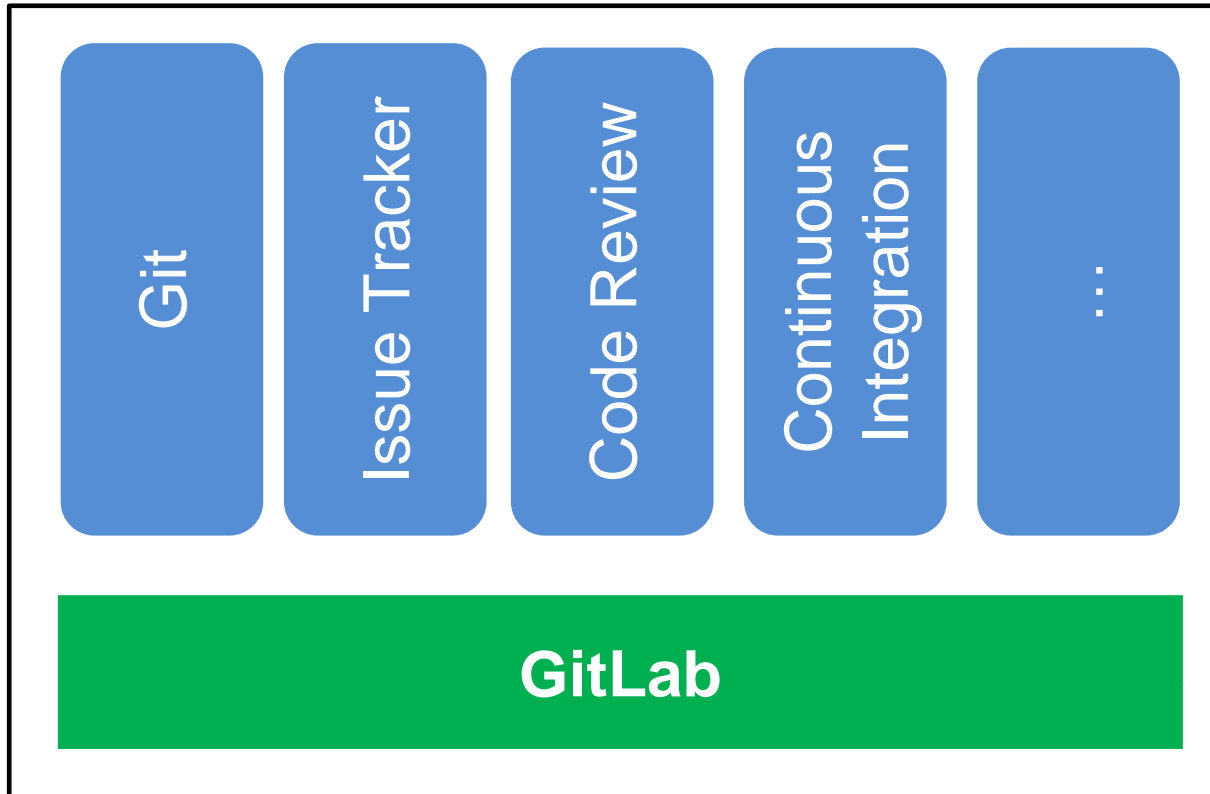


Find out about upcoming workshops, presentations, or trainings!



Umgebung im DLR

Werkzeuge / Training



Zusammenfassung und Ausblick

DLR Software-Engineering-Empfehlungen:

- Werkzeug für bessere Forschungssoftware
- **Aber:** Unterstützende Umgebung ist entscheidend!

(Ständige) nächste Schritte:

- Verbesserung der Empfehlungen
- Ausbau der Unterstützungsangebote



["strand_20110417_0077"](#) by [nederhoed](#) is licensed under [CC BY-SA 2.0](#)



Weitere Informationen



DLR Software Engineering Initiative Guidelines References About

DLR Software Engineering Initiative

Welcome to the site of the Software Engineering Initiative of the [German Aerospace Center \(DLR\)](#).

We founded the DLR Software Engineering Initiative to support researchers to develop sustainable software. A task, which is not easy, especially if you are not a trained software developer.

On this site, we share the activities which are part of the initiative to support the general discussion about good software development practice in research. An overview of the initiative in total as well as more detailed information about some activities can be found in the [references](#) section. In the [guidelines](#) section, you can find the list of recommendations that we have developed to support DLR researchers. If you have any feedback for us, please [let us know!](#)

Related Initiatives

The DLR Software Engineering Initiative is related to the Research Software Engineers (RSE) movement. The RSE movement campaigns for the adoption of software engineering practices in research and the recognition of the RSE role in academia.

The RSE movement started in [UK](#). In Germany, the [de-RSE](#) community is supporting this movement.

- [UK RSE](#) (United Kingdom)

RSE
RESEARCH SOFTWARE ENGINEERS ASSOCIATION
RSE Association, CC BY 4.0

<https://rse.dlr.de>

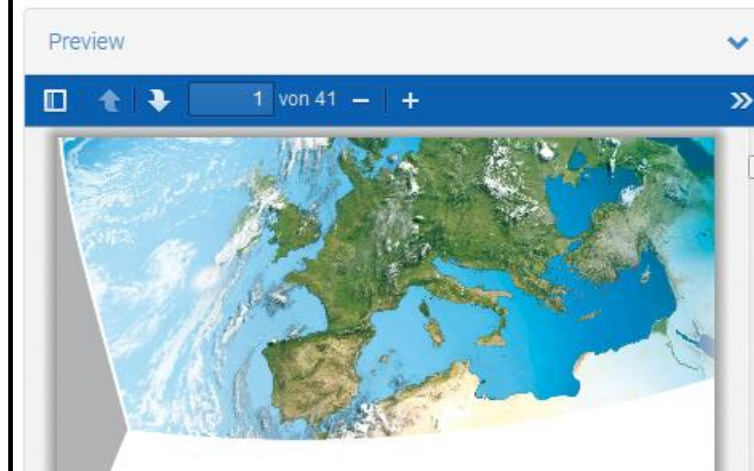
August 17, 2018

[Report](#) [Open Access](#)

DLR Software Engineering Guidelines

[Schlauch, Tobias](#); [Meinel, Michael](#); [Haupt, Carina](#)

This document describes the software engineering guidelines of the German Aerospace Center (DLR). The target group of the guidelines are DLR scientists. The guidelines shall support them to find out the status of their developed software and to improve it with regard to good software development and documentation practice. The focus of the guidelines is on retaining knowledge and supporting sustainable software development in research. We publish these guidelines to support the general discussion about good software development practice in research.



German: <https://doi.org/10.5281/zenodo.1344608>

English: <https://doi.org/10.5281/zenodo.1344612>

837

views


561

downloads

[See more details...](#)

23

[See more details](#)
 Tweeted by 31

 2 readers on Mendeley

Indexed in


Publication date:

August 17, 2018

DOI:DOI [10.5281/zenodo.1344612](https://doi.org/10.5281/zenodo.1344612)**Keyword(s):**

research-software-engineering

guidelines

Vielen Dank!

Twitter: @TobiasSchlauch

E-Mail: Tobias.Schlauch@DLR.de

Lizenzhinweis:

Alle Texte und Bilder, falls nicht anderweitig angegeben, sind unter den Bedingungen der Creative Commons Attribution 4.0 International (CC BY 4.0) lizenziert: <https://creativecommons.org/licenses/by/4.0/>

