

Guidelines and Environment to Support Sustainable Software Development at DLR

Carina Haupt
Simulation and Softwaretechnology
German Aerospace Center (DLR)

carina.haupt@dlr.de
@caha42



Knowledge for Tomorrow



DLR

Some Facts around Software Development

Some numbers...

- About 8.200 employees
- More than 3.000 employees develop software

Variety of

- Fields
- Maturity
- Software technologies
- Team sizes
- Backgrounds

How to support scientists to develop sustainable software?



Software Engineering Initiative of DLR

Software Engineering Initiative of DLR

Network

Guidelines

Tools

Trainings

Knowledge and
Experience
Exchange



Our approach for DLR Details

- Software Engineering Network
- KnowledgeExperienceExchanges
- SoftwareEngineering.Wiki
- Trainings
- RSE Consulting
- Guidelines



ErfahrungsaustauschWorkshop

Software Engineering
Fallstrick
15. - 16.

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!

- Before you
- Any Software
- You require document Software Engineering

This Wiki space is moderated by the department [SE](#)

Blog Posts
Aus DLR Open Blog: Folge-WAW
DLR Open II - Thema und Termin steht - Anmeldung!
2017
heering
2017

Embed System Software Engineering
Call for Teachers
Mehr unter: [s.dlr.de/7xt5](#)

Get Involved!
Get Involved!
Tools
Learn about specific SE Tools!

Kontakt:
Tobias Schlauch
tobias.schlauch@dlr.de

Anmeldung über das Bildungsprogramm im Intranet

Mehr Details unter:
[s.dlr.de/7xt5](#) (Wiki)

WF: f...
PS: 21...
Se: PS-1...
PS: NA
WT: f...
5. NAV
Own
Middle
Hel...

The DLR Software Engineering Guidelines

Topics

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

- Guideline document & Checklists
- Joint development with focus on **good practices, tools, and essential documentation**
- **77 recommendations** give advice in different fields of software engineering:

Requirements
Management

Software
Architecture

Design &
Implementation

Change
Management

Software Testing

Release
Management

Automation &
Dependencies

<https://zenodo.org/record/1344612> (EN)
<https://zenodo.org/record/1344608> (DE)



The DLR Software Engineering Guidelines

Using Checklists

Check List

Änderungsmanagement		
EÄM.2: Die wichtigsten Informationen, um zur Entwicklung beitragen zu können, sind an einer zentralen Stelle abgelegt.		
EÄM.5: Bekannte Fehler, wichtige ausstehende Aufgaben und Ideen sind zumindest stichpunktartig in einer Liste festgehalten und zentral abgelegt.		
EÄM.7: Ein Repository ist in einem Versionskontrollsystem eingerichtet. Das Repository ist angemessen strukturiert und enthält möglichst alle Artefakte, die zum Erstellen einer nutzbaren Version der Software und deren Test erforderlich sind.		
EÄM.8: Jede Änderung des Repository dient möglichst einem spezifischen Zweck, enthält eine verständliche Beschreibung und hinterlässt die Software möglichst in einem konsistenten, funktionierenden Zustand.		

Concrete Guideline

EÄM.7 Ein Repository ist in einem Versionskontrollsystem eingerichtet. Das Repository ist angemessen strukturiert und enthält möglichst alle Artefakte, die zum Erstellen einer nutzbaren Version der Software und deren Test erforderlich sind.	ab 1 Das Repository ist der zentrale Einstiegspunkt in die Entwicklung. Dadurch sind alle wesentlichen Artefakte sicher gespeichert und an einer Stelle auffindbar. Einzelne Änderungen können nachvollzogen und dem jeweiligen Urheber zugeordnet werden. Darüber hinaus stellt das Versionskontrollsystem die Konsistenz aller Änderungen sicher. Die Verzeichnisstruktur des Repository sollte man anhand bestehender Konventionen ausrichten. Quellen dafür sind typischerweise das Versionskontrollsystem, das Build-Werkzeug (vgl. Abschnitt 4.8 Automatisierung und Abhängigkeitsmanagement) oder die Community der eingesetzten Programmiersprache bzw. des verwendeten Frameworks. Dazu zwei
---	---

Topic Overview

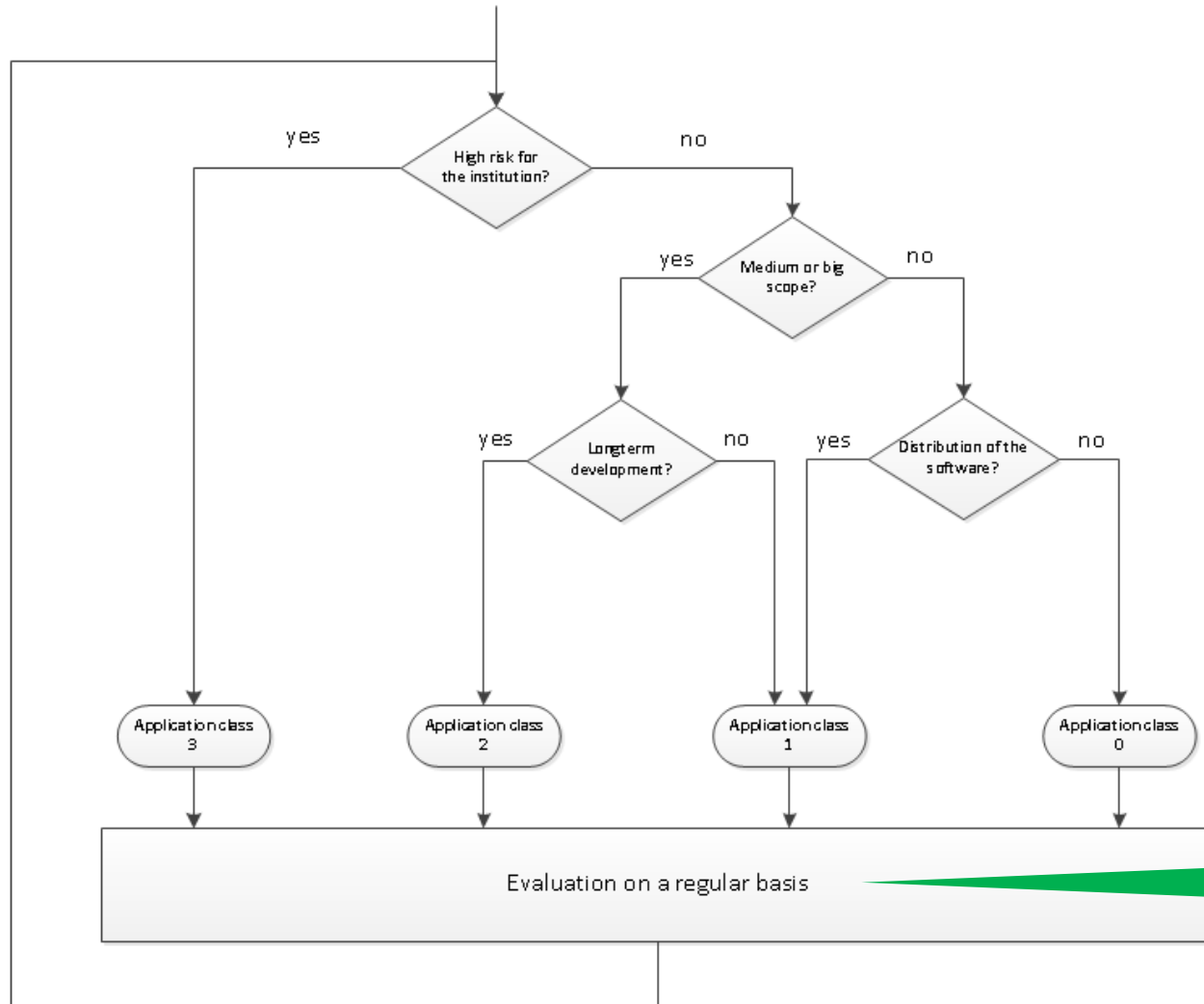
4.4 Änderungsmanagement

Gegenstand des Änderungsmanagements¹¹ ist, systematisch und nachvollziehbar Änderungen an der Software durchzuführen. Ursachen für Änderungen sind beispielsweise Anforderungen, Fehler oder Optimierungen. Das Änderungsmanagement unterstützt dabei, den Überblick über den Entwicklungsstand zu behalten und die verschiedenen Entwicklungsaufgaben zu koordinieren.

In diesem Zusammenhang beschreibt der **Änderungsprozess**, wie **Änderungswünsche** (z.B. Anforderungen, Fehler, Optimierungen) prinzipiell auf Entwicklerseite abgearbeitet werden und anschließend ggf. in Form einer neuen Software-Version zur Verfügung stehen. Dieser Prozess ist im Detail in jedem Entwicklungskontext unterschiedlich. Daher ist es wichtig, diesen im Entwicklungsteam abzustimmen und kontinuierlich zu verbessern. In der Praxis ist darauf zu achten, dass sich die Abläufe effizient umsetzen lassen. Daher ist auf angemessenen Einsatz von Werkzeugen und Automatisierung zu achten.

The DLR Software Engineering Guidelines Tailoring Checklists

An application class provides an initial starting point. Recommendations can be added and removed to fit the context.



Application class 1

- „small“, but other use it

Application class 2

- „medium – large“, other use it, long-term support

Application class 3

- „products“, critical for success of department or institute

Application class 0

- Personal „use“ (intentionally left blank)

Classification may change over time!



The DLR Software Engineering Guidelines

Motivation

For whom is it suited?

- New developers: Finding out about suitable SE practices
- Senior developers: Making sure to not forget something

In which situations could they support you?

- Finding out about the current status of a software
- Reasoning about / finding improvements concerning SE practices
- Convince your sponsor, boss, colleagues etc. to introduce a suitable level of SE practices



Trainings Structure

Sustainable Software Development for Scientists of DLR

- Based on Application Class 1
- Implementation example

Software Development for Teams

- Based on Application Class 2+3
- Example team



Training: Sustainable Software Development for Scientists of DLR

Recommendations of Application Class 1

1. Manage your code using a version control system

Part I

2. Apply a basic coding style, strive for a modular design, avoid code duplication and over-engineering

3. Automate creation of a usable version of your software

Part II

4. Provide essential documentation

5. Test your software and assign a proper release number before you release it

6. Apply a license in case of a public release



Training: Sustainable Software Development for Scientists of DLR Structure

Part I: Working with Git and GitLab - *based on SW Carpentries*

- Basics
- Process alone
- Process with students or in a small team

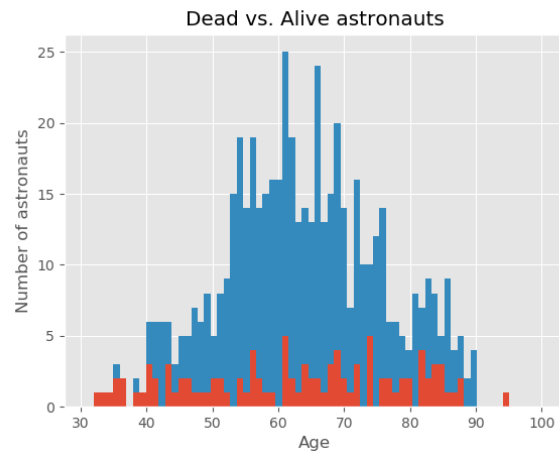
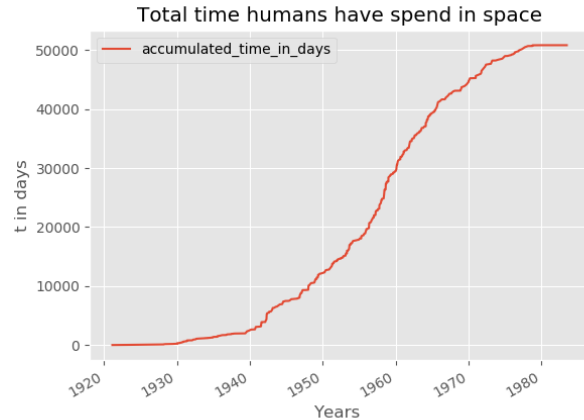
Part II: Make your script sustainable

- Tips and tricks for understandable code
- Essential software documentation
- Introducing open source licenses
- Release management
- Software citation



Implementation Example

Astronaut Analysis



```

C:\Users\schlauch\git\sustainable-sw-dev-for-scientists\03_sharing-script\part_0\main.py - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Werkzeuge Makro Ausführen Erweiterungen Fenster ?
main.py
15 def died_with_age(row):
16     if pd.isnull(row["date_of_death"]):
17         return None
18     born = row["birthdate"]
19     today = row["date_of_death"]
20     return today.year - born.year - (today.month < born.month or
    (today.month == born.month and today.day < born.day))
21
22 plt.style.use("ggplot")
23 df = pd.read_json("astronauts.json")
24 df = df.rename(index=str, columns={"astronaut": "astronaut_id",
    "astronautLabel": "name", "birthplaceLabel": "birthplace",
    "sex_or_genderLabel": "sex_or_gender"})
25
26 df = df.set_index("astronaut_id")
27 df = df.dropna(subset=["time_in_space"])
28 df["time_in_space"] = df["time_in_space"].astype(int)
29 df["time_in_space"] = pd.to_timedelta(df["time_in_space"], unit="m")
30 df["time_in_space_D"] = df["time_in_space"].astype("timedelta64[D]")
31 df["birthdate"] = pd.to_datetime(df["birthdate"])
32 df["date_of_death"] = pd.to_datetime(df["date_of_death"])
33 df.sort_values("birthdate", inplace=True)
34 df["alive"] = df["date_of_death"].apply(is_alive)
35 df["age"] = df["birthdate"].apply(calculate_age)
36 df["died_with_age"] = df.apply(died_with_age, axis=1)
37
Python file length: 3.849 lines: 97 Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

```

Python 3 based data analysis
and plotting script



Making your Script sustainable

Sharing the Astronauts Analysis Script (1/2)

1. Put your code under version control
 - ▶ Usage of Git and GitLab
2. Make sure that your code is in a sharable state
 - ▶ PEP8, directory structure, functions
3. Automate creation of a usable version of your software
 - ▶ requirements.txt + pip
4. Add essential documentation
 - ▶ Purpose, installation instructions, usage examples, citation hint



Making your Script sustainable

Sharing the Astronauts Analysis Script (2/2)

5. Create a release

- ▶ Tested manually, semantic versioning, release tag

6. Add a license file

- ▶ MIT License, copyright/license header

7. *Make your code citable*

- ▶ *CFF, Zenodo*



Workshop Focus

Recommendations of Application Class 1

1. Manage your code using a version control system ✓
2. Apply a basic coding style, strive for a modular design, avoid code duplication and over-engineering ✓
3. Automate creation of a usable version of your software ✓
4. Provide essential documentation ✓
5. Test your software and assign a proper release number before you release it ✓
6. Apply a license in case of a public release ✓



Training: Sustainable Software Development for Scientists of DLR

Feedback

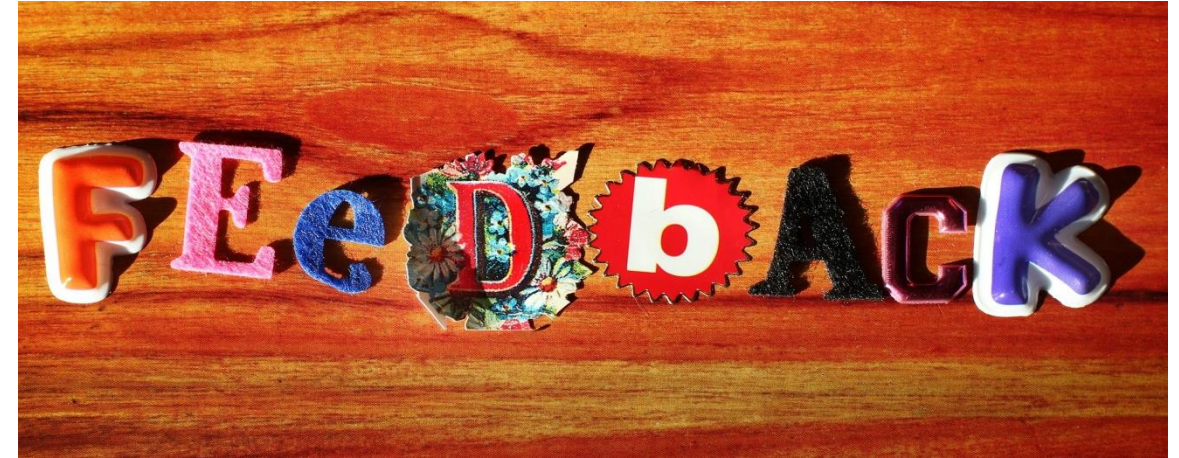
Over 100 participants in 5 trainings

Positive

- Good overview
- Live coding
- Example

Negative

- Too much content
- More depth



There is a whole movement...

de-RSE (<http://www.de-rse.org/>)

- Foundation: 26. November 2018, Berlin
- #deRSE19 Conference: 4.-6. June 2019, Potsdam



RSE UK (<http://rse.ac.uk/>)

Helmholtz Task Group „Wissenschaftliche Software“

(<https://os.helmholtz.de>)



HELMHOLTZ
Open Science



Questions?

carina.haupt@dlr.de

@caha42



#WIRROCKENSOFTWARE