

# Good practices for sustainable software development

## The DLR Software Engineering Guidelines

Carina Haupt  
Institute for Softwaretechnology  
German Aerospace Center (DLR)  
@caha42



Knowledge for Tomorrow



# DLR

## Some Facts around Software Development

### Some numbers...

- More than 1500 employees develop software

### Characteristics

- „Developer“ often do not have any training in software development
- Huge amount of software projects
- Variety of used software technologies



How to support scientists to develop sustainable software?



# Aspects of modern software development

- Distributed development processes via git, subversion,...
- Community software (github, bitbucket,...)
- Issue tracking systems
- Open source licensing (BSD, MIT, (L)GPL,...)
- Software architecture
- Build systems (CMake, Autotools,...)
- Meta build systems (Spack, EasyBuild, Conda)
- Test frameworks (GoogleTest, PyTest, junit, ...)
- Continuous integration testing (Jenkins, gitlab-ci,...)
- Integrated development environments (IDEs, e.g. Eclipse, QtCreator, MS Visual Studio)
- .etc.



It depends...



# The DLR Software Engineering Guidelines

## Topics

<https://rse.dlr.de>

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		
<b>EÄM.2:</b> Die wichtigsten Informationen, um zur Entwicklung beitragen zu können, sind an einer zentralen Stelle abgelegt.		
<b>EÄM.5:</b> Bekannte Fehler, wichtige ausstehende Aufgaben und Ideen sind zumindest stichpunktartig in einer Liste festgehalten und zentral abgelegt.		
<b>EÄM.7:</b> 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.		
<b>EÄM.8:</b> 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

<b>EÄM.7</b> 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.	<b>ab 1</b> 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

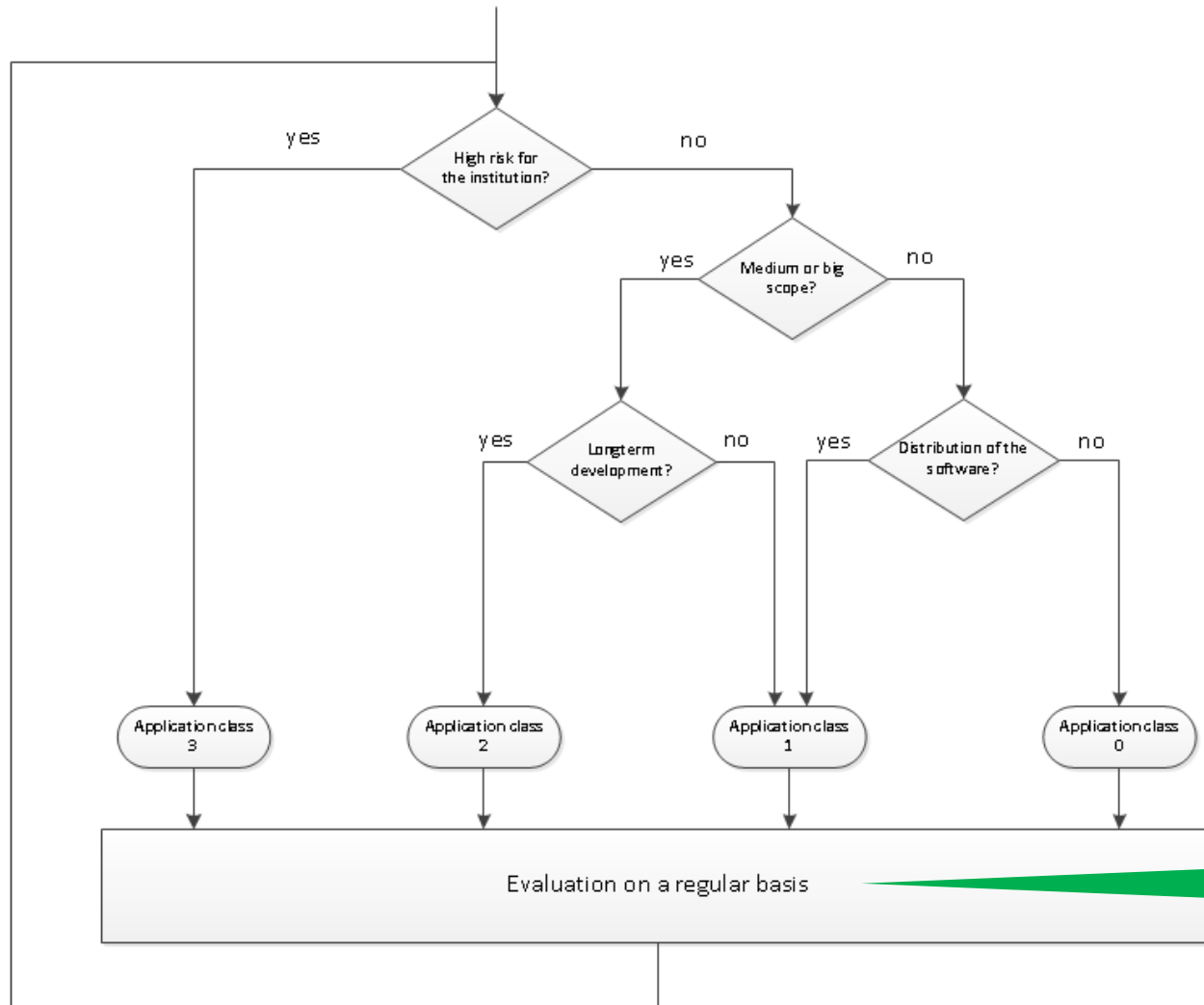
Gegenstand des Änderungsmanagements<sup>11</sup> ist, systematisch und nachvollziehbar Änderungen an der Software durchzuführen. Ursachen für Änderungen sind beispielsweise Anforderungen, Fehler oder Optimierungen. Das Änderungsmanagement unterstützt dabei, den Überblick über den Entwicklungsstand zu 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

## Application classes

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!



But how does this look like in practice?



Knowledge for Tomorrow



# The DLR Software Engineering Guidelines

## An Example

### Python script for calculation of characteristics of a set of sample values

- Software is a small tool and used by other internally.

The software fits well into the application class 1.

### Summary of the generic recommendations:

- Manage your code using a **version control system**
- Apply a **basic coding style**, strive for a **modular design**, **avoid code duplication** and **over-engineering**
- **Automate creation of an executable, usable version**
- **Provide essential documentation:** software purpose, user and developer information, constraints and central concepts, known problems and ideas
- **Internal release: test** your software and assign a proper **release number**
- **Public release:** check the **open source guidelines**

Recommendations have to be mapped into the concrete development context.





# The DLR Software Engineering Guidelines

## Possible Implementation

Git repository which contains code, examples, build script, and documentation

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

- Releases correspond to tags
- Release package download

Add license Set up CI

or / +

Find file

History



345241ea

Schlauch, Tobias committed 9 minutes ago

Name	Last commit
examples	Adds examples.
src	Provide initial version.
CHANGES.md	Adds the change log.
CONTRIBUTING.md	Adds information on how to contribute.
README.md	Fixes typos.
setup.py	Adds a build script for creation of the dist...

- Examples provide reference input values and results

- Code is broken into small functions
- Coding style recommendations applied

- Build script for packaging and installing
- Release numbers follow semantic versioning approach
- CHANGES.md explains user-understandably major changes



# The DLR Software Engineering Guidelines

## Possible Implementation (cont.)

### 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](#):  
main documentation
- [CONTRIBUTING.md](#):  
contributor information

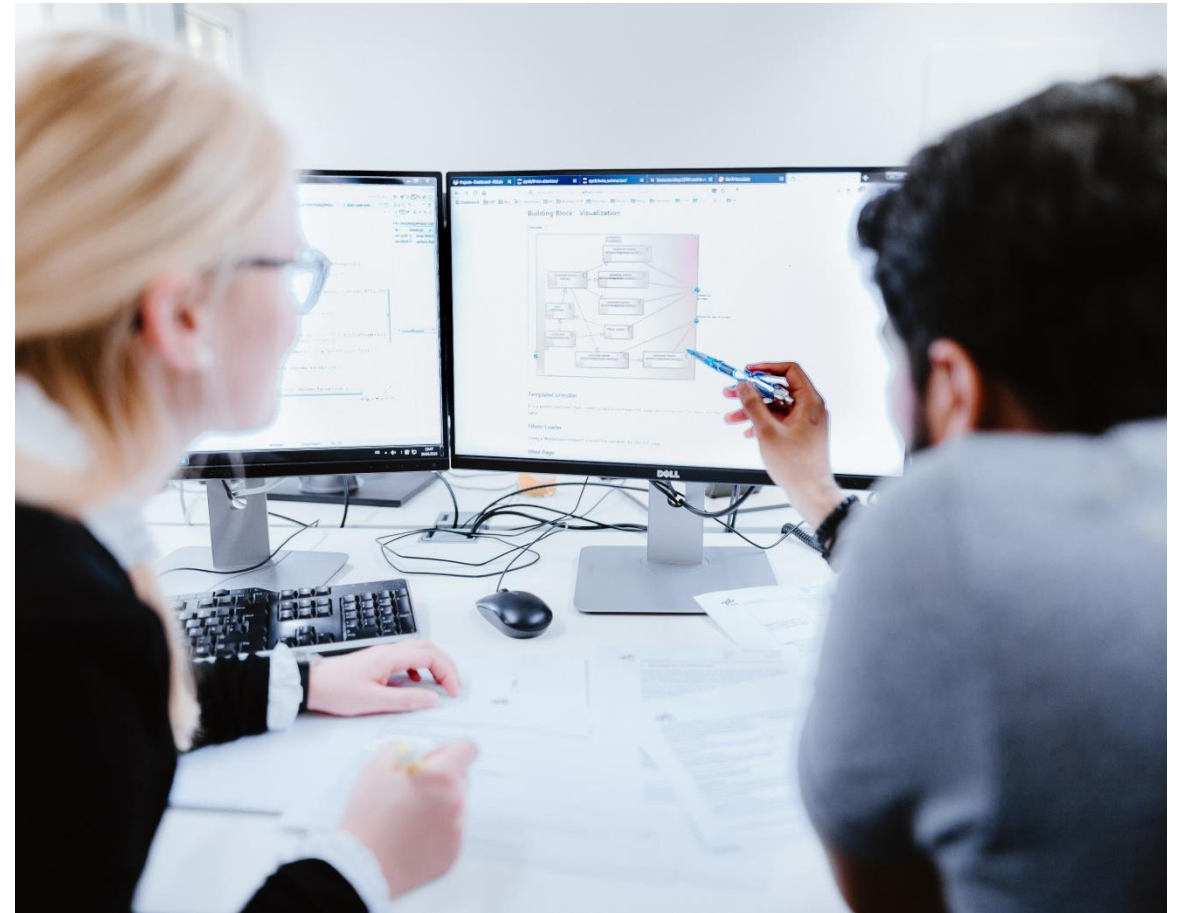
- Explanation of the [software purpose](#) (what?, for whom?, why?)
- Overview of the [main features](#)
- Important [usage constraints](#) and [conditions](#)

- Basic [installation](#) and [usage information](#)
- Future plans and ideas



# The DLR Software Engineering Guidelines Summary

- Checklists are project sensitive
- Generic recommendations → Solution Suggestions
- Covering the Basics
- Reevaluate regularly



That's not all folks.



Knowledge for Tomorrow



**DLR**

Our approach

## Software Engineering Initiative of DLR

Guidelines

Trainings

Knowledge  
Provision

Collaboration

Experience  
Exchange



# We are not the only ones...

## eScienceCenter (NL)

Netherlands eScience Center Guide

https://guide.esciencecenter.nl/#/

netherlands  
eSciencecenter  
guide

Type to search

Introduction

Best practices

Checklist

Version Control

Code Quality

Code Review

Licensing

Communication

Testing

Releases

Documentation

Standards

UX - User Experience

Language Guides

Bash

Dead-link checker status (does not actually build guide itself): build error

DOI 10.5281/zenodo.4020565

## Guide

This is a guide to projects and software development at the Netherlands eScience Center. It both serves as a source of information for exactly how we work at the eScience Center, and as a basis for discussions and reaching consensus on this topic.

**This Guide is a work in progress**

Read this book online here: <https://guide.esciencecenter.nl>

If you would like to contribute to this book see [CONTRIBUTING.md](#).

To see who is responsible for which part of the guide see [chapter\\_owners.md](#).

The guide is best read online, but there is a [PDF document](#) for offline reading.

- Similar topics
- Language guides
- No application classes



## There is a whole movement...

- RSE UK (<http://rse.ac.uk/>)
- de-RSE (<http://www.de-rse.org/>)
- Helmholtz Task Group „Wissenschaftliche Software“ (<https://os.helmholtz.de>)



# Questions?

carina.haupt@dlr.de  
@caha42

**#WIRROCKENSOFTWARE**