Scientific Software Center

IWR Interdisciplinary Center for Scientific Computing

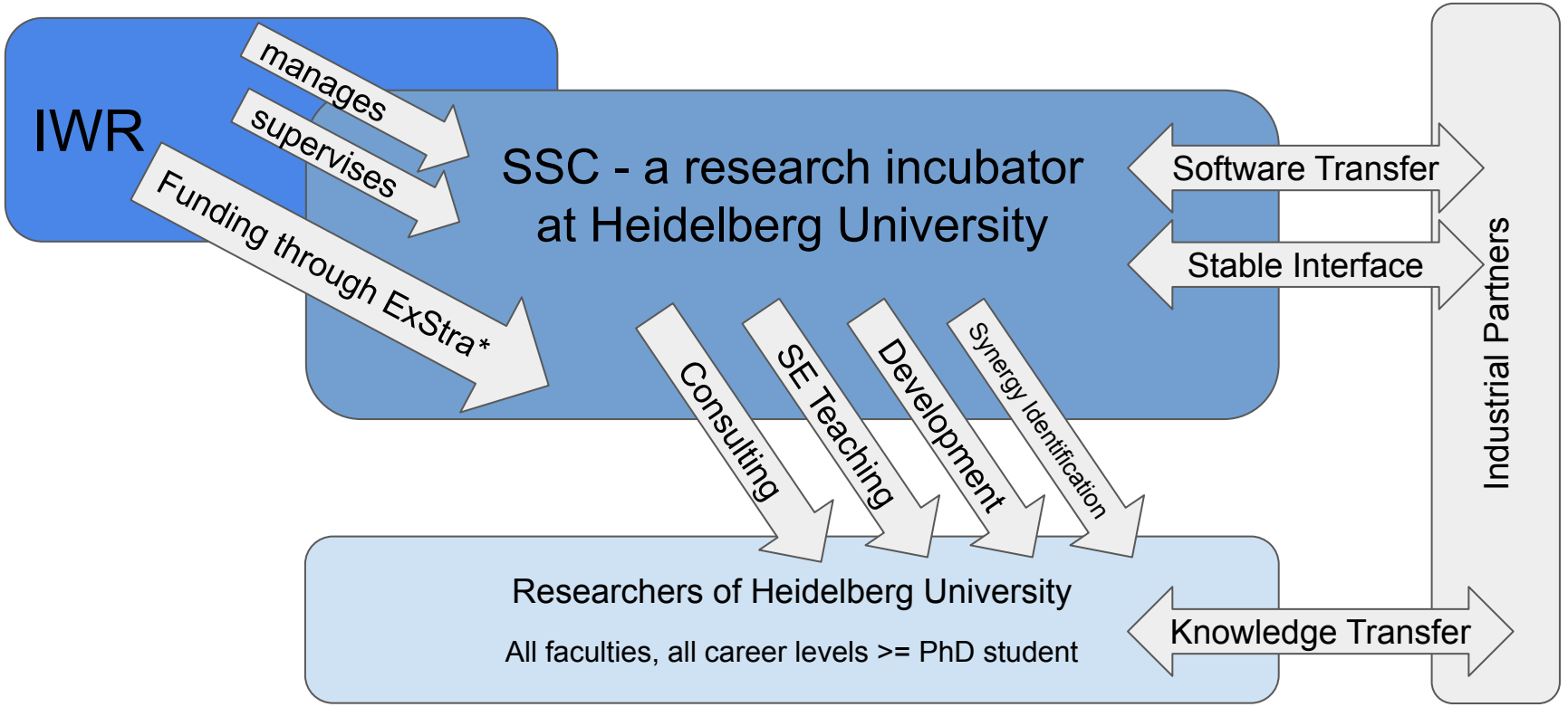
UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

2025 Open Call of the Scientific Software Center

Liam Keegan, Dominic Kempf, Inga Ulusoy, Harald Mack

ssc.iwr.uni-heidelberg.de

The Scientific Software Center



*the SSC receives its core funding through the *Exzellenzuniversität* initiative

The SSC's target group

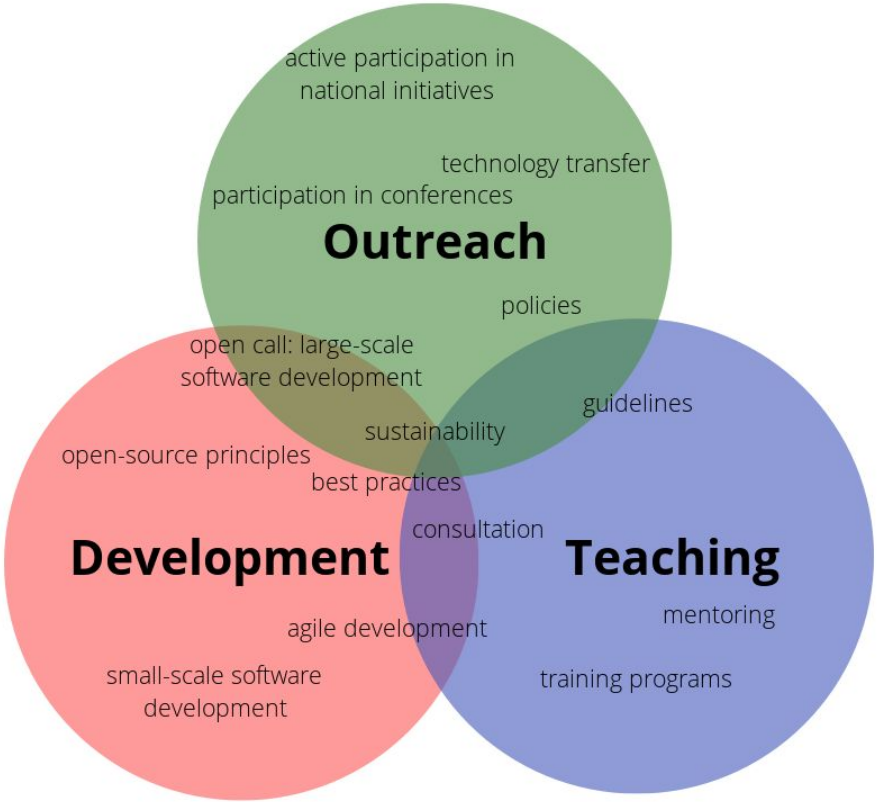
Users of specialized software/libraries

- Research critically depends on digital tools and specialized research software
- Adaptation of software to specific needs
- The generated data is used to provide scientific insight

Developers of own specialized software/libraries

- Software development and method development are core concepts that the research is based on
- The developed methodology and generated data are used to provide scientific insight

SSC mission and team



SSC Team

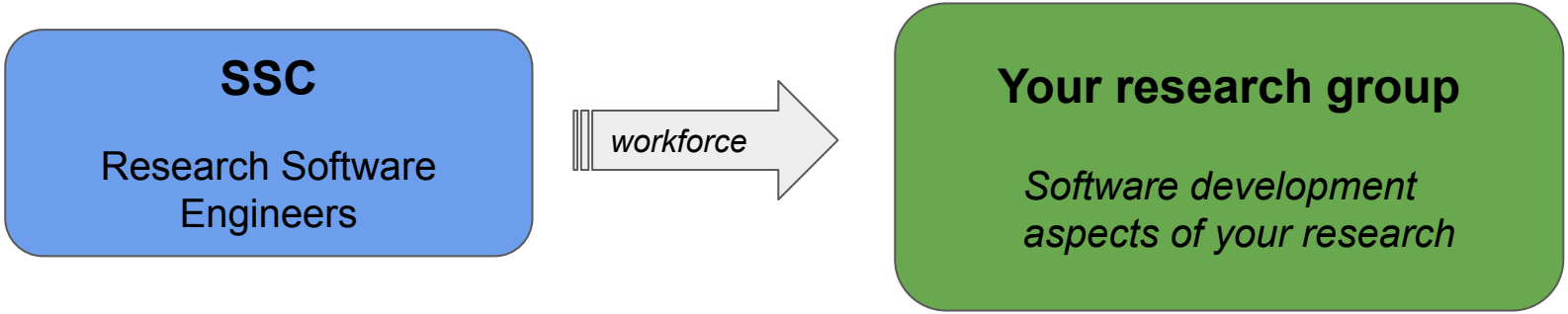
- Currently eleven Research Software Engineers (RSEs)
- Shared leadership:
  - Liam Keegan
  - Dominic Kempf
  - Inga Ulusoy
- System Administrator
- Administrative Support

Previous Open Call projects

	Field	Project Type	Year	
AFWizard: Adaptive ground point filtering for aerial archaelogy Python framework to facilitate complex ground point filtering workflows in archaeologic prospection	Archaeology	Open Call Project	2021	→
Corpus and argumentation Open-source software to automatize annotation and database encoding of linguistic texts	Language Studies	Open Call Project	2021	→
Neuroscience data processing and analysis Deploying and improving analysis pipelines on HPC resources	Biology	Open Call Project	2021	→
py4dgeo: Change detection 4D point cloud data Library of algorithms for the spatio-temporal analysis of 3D point cloud data. Provides an extensible, open Python framework based on a high performance C++ core.	Geography	Open Call Project	2021	→
Systems biology parameter estimation GUI Extend parameter fitting functionality and integrate into the existing GUI application	Biology	Open Call Project	2021	→
Dust polarization observations Improve portability and quality of code and tooling	Astrophysics	Open Call Project	2022	→

https://ssc.iwr.uni-heidelberg.de/open-call-projects

What can you apply for in this call?

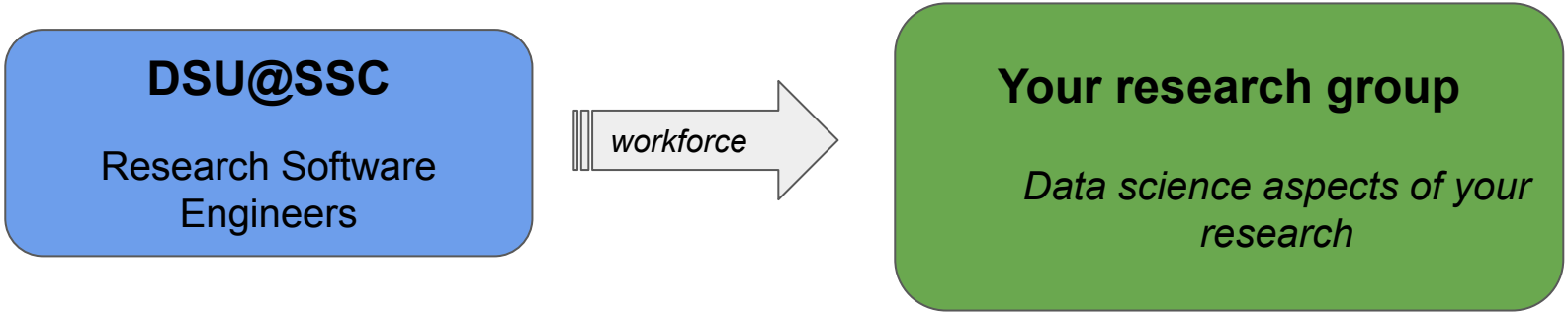


- Custom development of **new scientific software** for a research project
- Addition of **new functionality** into existing research software
- Development that **increases the software quality** of existing research software (e.g. performance, scalability, portability, usability, reusability etc.)
- Adoption of **best practices** for the development of scientific software

What can you apply for in this call?

New!

Scientific Software Center

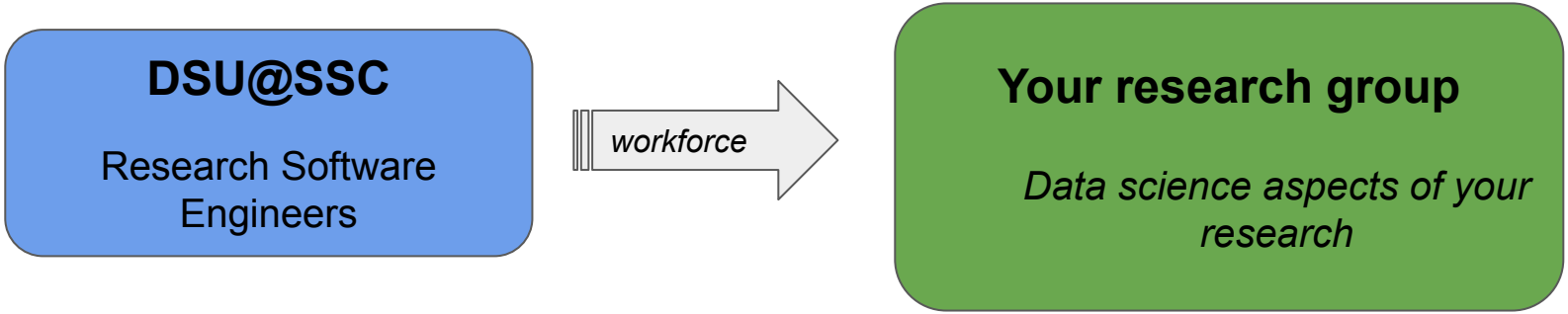


- **Develop new** data science solutions from scratch for specific questions
- **Augment or enhance existing** data science workflows with new models or functionality, increase reproducibility, scalability and trustworthiness
- **Make machine learning results interpretable** - go from black box results to communicable insights

What can you apply for in this call?

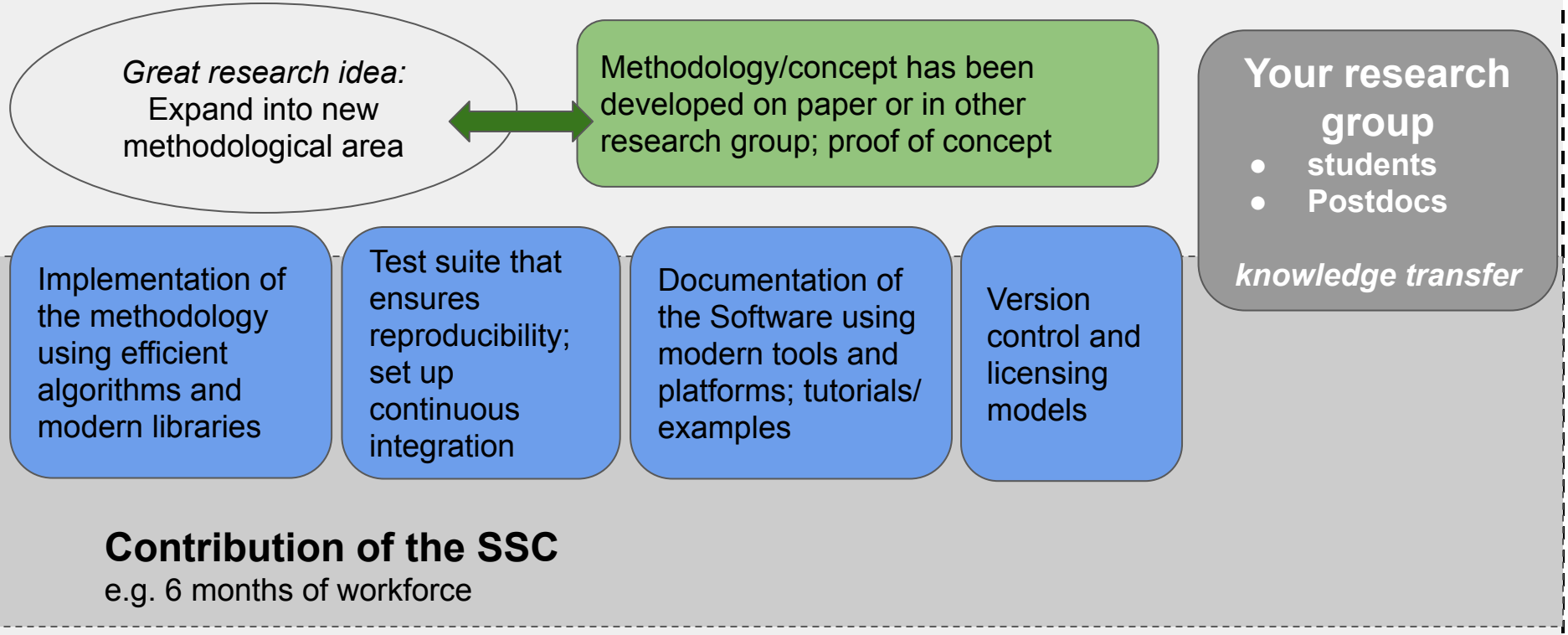
New!

Scientific Software Center



- **Apply data science to specific scientific questions**
  - Build a new data science system based on research project
  - Augment or enhance existing systems to tackle new questions
- **Integrate data science into the scientific process**
  - Make machine learning systems more interpretable
  - Data science for knowledge extraction or hypothesis generation

Example 1: Custom development of new scientific software



Example 2: Custom development of a data science system

Great research idea:
Expand into new methodological area

Idea/concept has been developed on paper or in other research group; proof of concept

Your research group
- **students**
- **Postdocs**

knowledge transfer

Build scalable data processing- and augmentation pipeline

Select and evaluate model architectures and paradigms together with research group

Train, evaluate and verify selected model to ensure reproducibility and trustworthiness

Test and document produced software using best practices

Contribution of the SSC
e.g. 6 months of workforce

Example 3: New functionality for existing software

New Research Question

New User Base

Changes to State of the Art

New Requirements

Existing piece(s) of research software

- Self-developed program/framework
- Open Source Project
- Lab Code/Script
- Data Processing Workflow
- Database System

Contribution of the SSC, e.g. 1-3 months

Many important new features do not touch on the scientific core of a piece of research software, but rather require pure engineering work.

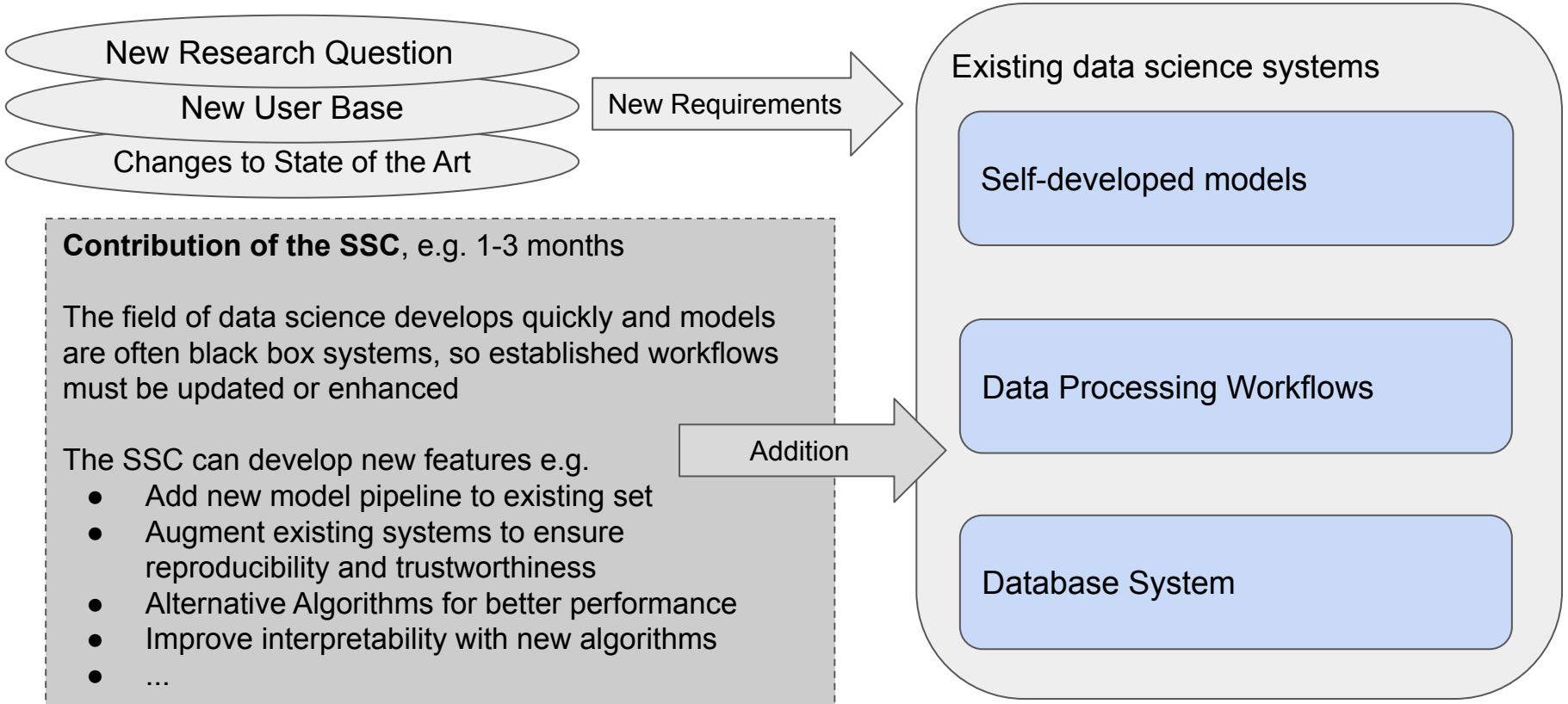
Addition

Enhancement

The SSC can develop new features e.g.

- New output/input formats
- Bridging two pieces of research software
- Alternative Algorithms
- Restructure software to enable new research methodology
- ...

Example 4: Augmenting existing machine learning pipeline



Contribution of the SSC, e.g. 1-3 months

The field of data science develops quickly and models are often black box systems, so established workflows must be updated or enhanced

The SSC can develop new features e.g.

- Add new model pipeline to existing set
- Augment existing systems to ensure reproducibility and trustworthiness
- Alternative Algorithms for better performance
- Improve interpretability with new algorithms
- ...

Example 5: Increase quality of existing research software

Involvement of students/postdoc knowledge transfer

Existing research software that has been developed in your group

- Low performance
- Low parallel scaling
- Dependence on a specific set of libraries (possibly outdated)

Contribution of the SSC
e.g. 3 months of workforce

- Analysis and improvement of serial performance
- Analysis and improvement of parallel performance
- Suggestion and implementation of interface to alternative libraries; updated build system

Other examples could involve:

- portability of the software (different environments/operating systems)
- usability (user interface, code structure)
- reusability (generalization of the software)
- …

Example 6: Adoption of best practices

Status Quo: A research group develops a lab code that is passed on from PhD generation to generation. Everybody adds functionality and examples from their work.

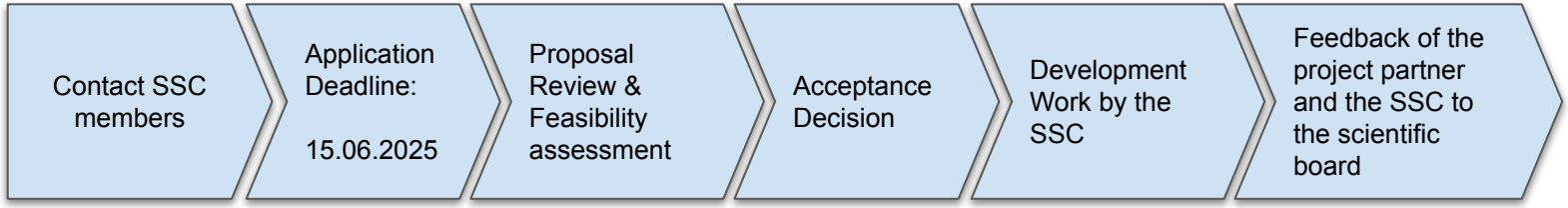
- Code divergence
- Insufficient Testing
- “Grown” software design
- Lack of Documentation

- Introducing Git + GitHub/GitLab
- Setting up CI + Introducing testing frameworks
- Code refactoring: Software Design consultation + Sprint supervision
- Introducing documentation tools E.g. Sphinx + Doxygen

SSC Contribution: A developer joins the group and introduces best practices that follow the state of the art in software development.

- *Implementation*: We set up tools and configurations for the group
- *Training*: We teach the group to enable long term improvement

How to apply



- Applications can be submitted online:

  www.ssc.uni-heidelberg.de/en/development/the-sscs-open-call

  https://limesurvey.urz.uni-heidelberg.de/index.php/956616?lang=en
- Applications can be written in English or German.

Evaluation criteria and process

The decision is taken by the scientific board of the SSC.

Criteria:

- Feasibility statement of the SSC
- Scientific merit of the proposal
- Clarity of what is expected from the SSC
- Leverage of project outcome
- Criteria about applicant:
  - Enabling interdisciplinary collaborations
  - Supporting Early Career Researchers
  - Aiding equal opportunities

Scientific Software Center

Question Time!

Get in touch!

By email:

- ssc@uni-heidelberg.de

Or have a closer look at our services on our website:

- https://ssc.uni-heidelberg.de