Recommendations on the

Development, Use and Provision of Research Software

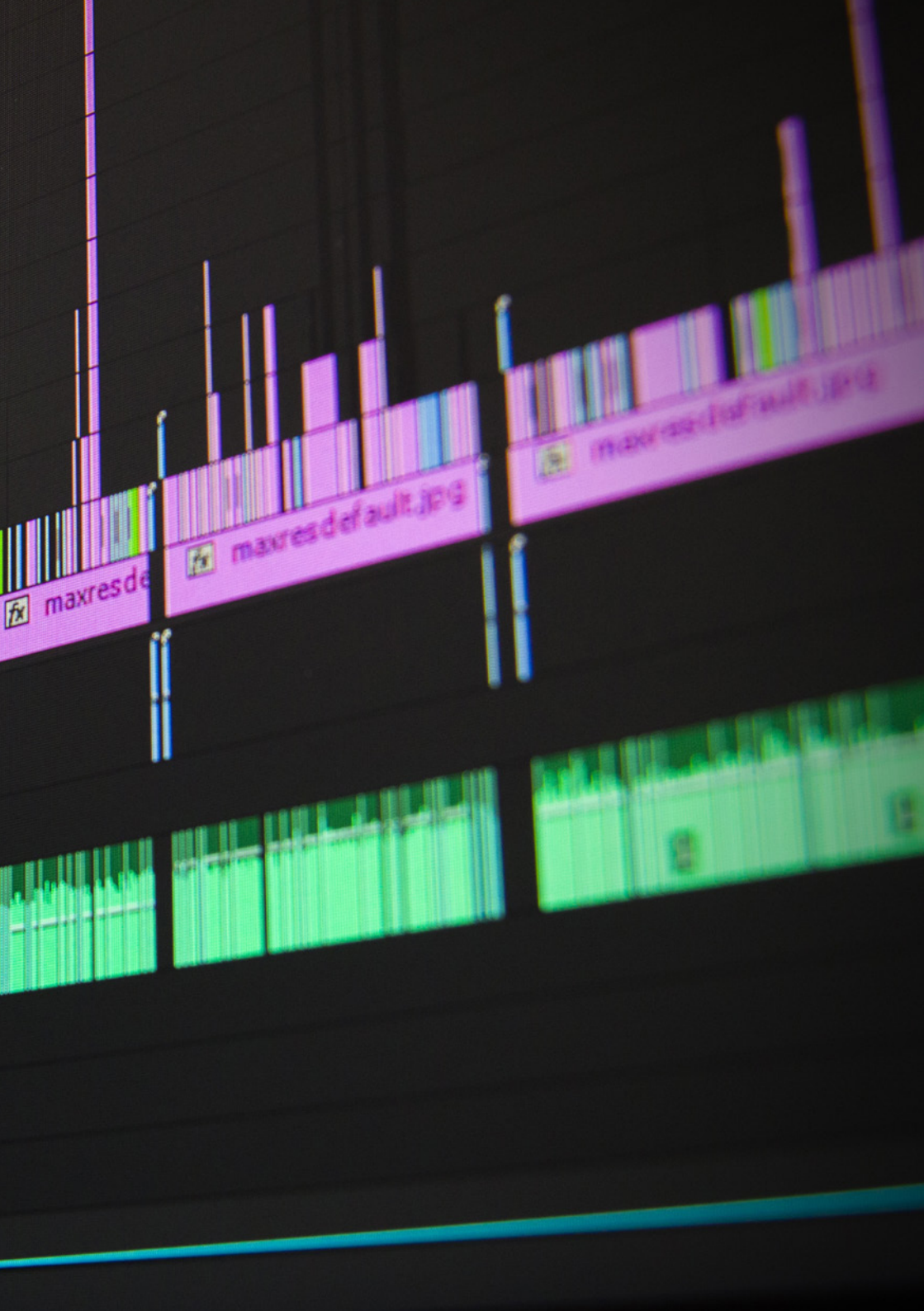
Research Software Working Group
in the Priority Initiative Digital Information
of the Alliance of German Science Organisations

This document should be cited as:
Katerbow, Matthias; Feulner, Georg (2018): Recommendations on the Development, Use and Provision of Research Software (Handreichung zum Umgang mit Forschungssoftware). Published by the Research Software Working Group as part of the priority initiative Digital Information of the Alliance of German Science Organisations. With the collaboration of Mathias Bornschein, Björn Brembs, Michael Erben-Russ, Konrad Förstner, Michael Franke, Bernadette Fritzsch, Jürgen Fuhrmann, Michael Goedicke, Stephan Janosch, Uwe Konrad, Dennis Zielke.

Digital edition
A digital edition of this document is available via
http://doi.org/10.5281/zenodo.1172988

German language edition
The original document is written in German and available via
http://doi.org/10.5281/zenodo.1172970

Table of contents


maxresde
maxresdefault.jpg

Introduction

Aim

These recommendations are primarily intended for senior researchers and research managers at universities, non-university research institutes, funding organisations and infrastructure facilities, as well as developers and users of research software. They provide information about the topic of research software, presenting challenges and possible solutions on the following three levels:

1. Development of research software
2. Use of research software
3. Provision of research software as a service to the research community

Background

Against the background of the digital transformation, research software has become an essential element of research activity. This includes the development and use of research software for simulation, for the generation, processing, analysis and visualisation of data and for the control of research equipment and experiments. Research software is therefore of fundamental importance, and this gives rise to new challenges in relation to good scientific practice, with respect to the transparency, traceability and reproducibility of research results. These requirements are best satisfied through open accessibility to software, so access to research software should be designed in line with the *FAIR principles* (Findability, Accessibility, Interoperability, Reuseability). In the debate surrounding open accessibility to scientific results, the connection between research data, research software and publications is therefore coming more and more to the forefront. Although the connection should be obvious – particularly with regard to the traceability and reproducibility of research results – there is still an imbalance on an organisational and a technical level. This can be seen, for instance, in the fact that retention periods have been defined for research data and that access arrangements are being discussed or even already implemented, whereas in many cases research software is not even managed or made available. However, research software is also associated with specific challenges and these are the subject of these recommendations.

Software categories

In the academic use of software there are three categories:

- *Self-developed research software:* Particularly where researchers have developed their own research software, open accessibility to the source code is necessary in order for others to verify how it works. This access should only be restricted (temporarily if appropriate) where there are clear utilisation interests. Access to the source code should therefore be emphasised in addition to the provision of executable files, for example.
- *Software applications for research:* This includes established software applications used by researchers (e.g. MITK), free open-source software (e.g. R), commercial software with licence costs (e.g. Matlab) and free software from commercial providers (e.g. yEd).
- *Infrastructure software and services:* Infrastructure software includes publicly funded web services to support research (e.g. Zenodo), repository software such as D-Space or Fedora, and commercial services (e.g. Dropbox, Figshare). In this document, infrastructure software is only discussed in terms of its importance to the long-term development and use of research software.

In the three sections that follow, challenges relating to the *development, use* and *provision* of research software are described and the responsibilities of senior researchers and research managers, software users and developers, and infrastructure facilities are formulated. The Research Software Working Group presents short-, medium- and long-term recommendations for each area of challenges and for all stakeholders.

Development of research software

Description of the problem

For the most part, academic software development currently takes place in an environment where temporary contracts, shortages of skilled staff and time pressures all hamper professional software development. The development environment is also subject to rapid change. In addition, unlike publications, the development of research software is not adequately recognised as an academic achievement. Finally, in terms of maintaining good scientific practice, there is a lack of established standards for the publication of source code, persistent identification of versions, and documentation.

Management responsibility

For senior researchers and research managers, the main challenges in academic software development are the recruitment of qualified staff, support for institutional software development and the training of academic and academic/technical staff. Recognition, career paths, training opportunities and incentives for ongoing software development must be improved both within individual institutions and throughout the research system as a whole. For the publication of the source code and software, the legal situation must be checked, licences for research use must be defined and arrangements must be made for transfer for commercial use. Wherever feasible, and if it cannot be commercially exploited, research software should be made available for free use as open-source code on a trustworthy infrastructure.

Developer responsibility

With respect to the basic principles of good scientific practice, academic software developers (which includes both academic-technical and academic staff) have a responsibility to comply with best practice in quality control during the software development process. As well as the publication of the source code, this includes the use of version management systems, comprehensive documentation of the code, the use of appropriate infrastructure software for quality management, standardised tests and, ideally, independent review of the source code.

Responsibility of infrastructure facilities

Central facilities (such as data centres and libraries) should actively support the software development and publication process not only by making adequate development and test environments available but also by advising on licensing, software publication and long-term storage and updates, thus providing assistance to academic software developers.

Recommendations

The recommendations below are aimed at **senior researchers** and **research managers** with a view to improving working conditions for academic software development:

- *Human Resources policy and funding:* Establishment and design of career paths for software developers and in particular the creation and active support of software development jobs. Enabling academic staff to take training courses with the aim of professionalising their skills in the development of research software. Support for developer networking activities. Creation and recognition of adequate time in research projects for the quality-assured development and, in particular, documentation of research software.
- *Recognition* of software development and documentation as an academic achievement, with documentation and measurability of this achievement for the future career development of researchers and academic software developers.

- *Automation:* In collaboration with central facilities (e.g. data centres), standardised working environments should be created for software developers in which the developer environment is linked as seamlessly and smoothly as possible to downstream infrastructures, allowing the FAIR principles to be efficiently implemented. Best practices should also be developed and publicised within the institutions.

The recommendations below are aimed at **academic software developers:**

- *Software quality:* International best practice standards should be followed in the development, packaging and documentation of research software so as to ensure maximum technical interoperability and long-term usability. Developers should take advantage of training and networking opportunities offered to them. Validation and acceptance procedures should be established and review processes should be integrated at an early stage.
- *Cooperation:* During the development process, regular dialogue between academic software developers and research users, and also between developers, should be facilitated for the purposes of quality assurance. The reuse of established and validated software components also contributes to productivity and quality; the usual citation standards should of course be followed.

The recommendations below on the professionalisation of academic software development are aimed at **infrastructure facilities:**

- *Software development:* Data centres should support research institutes in the establishment and provision of (standardised) development and test environments and, in particular, facilitate close links to established infrastructures and services for the publication of source code and research software.
- *Software publication:* Libraries, in particular, should set up advisory services on the licensing, persistent referencing and citation of research software and provide support to academic software developers on these issues.

Use of research software

Description of the current problem

Although the use of software is a core aspect of modern research work, there are still no established standards for the transparent representation of software use in the methods section of publications. This includes the detailed description and citation of the software version used and its configuration, the parameters chosen and the software and hardware environment in which the software was used. In the case of web applications, the fact that earlier versions are often not kept available only adds to the difficulty. Furthermore, the lack of established repositories for research software makes it difficult to find existing software solutions, potentially resulting in the unnecessary duplication of development work. Finally, the use of commercial software and services results in growing dependence on commercial providers and as yet unresolved aspects of data ownership regarding the decentralized (commercial) storage of data.

Management responsibility

Senior researchers and research managers have a responsibility to continuously develop the rules of good scientific practice in relation to research software and implement them in their institutions. With respect to commercial software and services, savings can be made through the acquisition of central licences (e.g. national licences). Special attention should be given to the risk of dependence on individual providers. However, as a general rule more emphasis should be placed on the use of academic research software. In particular, for reasons of cost-efficiency and data ownership, consideration should be given to the use of non-commercial and open-source applications and services.

User responsibility

Users of research software are confronted with the challenge of making an informed choice from among the software solutions available and understanding as far as possible how they work. The use of software must also be appropriately documented in publications to allow other researchers to follow how the results were obtained. In their role as reviewers, researchers can also help to ensure that corresponding standards are upheld. Due to the rapid pace of software development, management and users should remain in regular contact to ensure the best possible user support.

Recommendations

The recommendations below are aimed at **senior researchers** and **research managers** with a view to improving the use of research software:

- *Good scientific practice:* Refinement and implementation of university- or institution-specific rules of good scientific practice regarding the use and citation of research software and promotion of a culture of academic recognition for the development of research software.
- *Use and citation:* Development of rules for describing the use of research software (e.g. parameters, configuration) in the methods section of publications, establishment of citation guidelines for software and enabling documentation by users; raising awareness of research software in the peer review process. Use of packaged documentation options (e.g. Docker or other runtime environments) in situations where solutions of this kind constitute best practice.
- *Selection of research software:* When selecting and using research software, preference should ideally be given to freely available open-source software. Where older software is used, reasons should be given for this choice.
- *Funding for academic research software:* Enable quantification of the use of research software, encourage user participation in the software development process (e.g. through feedback) and offer support (financial if appropriate) to developers.

Provision of research software

Description of the current problem

While the challenges outlined above relating to the development of research software are relevant to an individual research project, a group of researchers or a project involving multiple teams, the requirements are multiplied as soon as individually or institutionally developed software is offered to a larger group of colleagues. This changes the responsibility from the original role of developer to the role of provider. Provision should be based on a community need identified in a bottom-up process such that a quality-checked selection process has already taken place. The change of role of a developer to the role of a provider requires considerable professionalisation in order to satisfy user needs in terms of quality, updates and ongoing development. The new role is also associated with additional challenges, primarily licensing and other legal considerations (such as copyright), a business or funding model for the long-term provision and maintenance of the software, and user support.

Management responsibility

Senior researchers and research managers must make the basic decision as to whether the institute or facility wishes to act as the software provider so as to have control over the professionalisation process. If so, then the first main challenge is to enable and organise the establishment of the provider role. Following the establishment phase, it is necessary to enable the long-term funding of the provider role, for example through a funding or business model (e.g. licence costs for commercial use or a membership model). In many cases coordination will be required between the academic software developers and supporting central infrastructure facilities.

Responsibility of infrastructure facilities

Central facilities such as libraries and computing, data or media centres can create services with which to support research institutes in the provision of research software. These may be information infrastructures to implement the *FAIR principles* for research software, (national) reference systems for research software or the creation of metadata sets for research software. Infrastructure facilities should also contribute to the development of generic solutions for the provision, referencing and searching of research software.

Provider responsibility

In the role of academic provider of research software as a service to the research community, there are three main challenges to be addressed: firstly, the management of 'customer' requirements and expectations (installation and setup, support, ticketing, needs analysis etc.); secondly, the ongoing maintenance and development of the software; and thirdly, ensuring the long-term availability of persistent, citable and documented versions of the source code including dependencies on reused software components and, if relevant, executable files of compiled programs.

Recommendations

The recommendations below are aimed at **senior researchers** and **research managers** with a view to professionalising the provision of research software as a service to the research community:

- For the ongoing development and maintenance of research software it is necessary to create support for a *developer community* for the collaborative (further) development and curation of collaborative developments.
- The development of *funding or business models* for the software provider role should begin early on so that a strategy can be implemented for the long-term assurance of software availability (source code and executable files) and citability. This assurance is especially important should the provider cease to function in a provider role.

- The *integration* of central infrastructure facilities and/or national infrastructures should also be taken into consideration at an early stage.
- Software *licensing* should be investigated and implemented prior to the first publication of the software.
- Tutorials, documentation and – if appropriate – training opportunities should be envisaged for software users.

The recommendations below are aimed at central **infrastructure facilities:**

- The *technical infrastructure* for the publication and collaborative ongoing development of the source code and persistent citation of the research software, e.g. for version management, should be made available. This includes the provision of runtime environments so that users or reviewers do not have to perform complex installations.
- Options should also be provided for the *quantification* of user numbers and – if appropriate – citations of the software. In addition to purely quantitative indicators, the academic use of the software should also be monitored.
- Infrastructure facilities should assist software developers and research institutes with the clarification of *legal matters* (e.g. licences and pricing).
- For new tasks, consideration should be given to the development and creation of *new roles* (e.g. software librarian).

Key recommendations

Senior researchers and *research managers* have a special responsibility to set in motion the prerequisites for the long-term development of research software in terms of *HR policy and funding*, to promote *compliance with good scientific practice* in relation to the use of software in their institutions and to develop the required *funding and business models* for the provision of research software.

Academic software developers should critically evaluate their own development processes in terms of *quality assurance* and comply with international standards for software development, documentation and testing.

Most importantly, *users of research software* must aim to achieve transparent *usage documentation and citation* of the software they use to make sure that research results remain reproducible and to give appropriate recognition to the work of external developers.

Infrastructure facilities have a responsibility to provide developers and users of academic software with the *technical infrastructure* they need to comply with the FAIR principles and to support users.

Summary

This document describes challenges relating to research software and provide recommendations for the development, use and provision of this type of software. The *relevance of research software* to modern research should be clearly underlined, especially in the context of political debate on digital transformation in the sciences and humanities.

In the short term, the most important task is to raise awareness of the problems and existing potential solutions among stakeholders. In the medium to long term, shared standards should be developed (e.g. for the documentation of software use in publications and the citation of software).

Given the growing relevance of research software, *a greater degree of networking* between the various stakeholders is necessary, particularly across disciplines. One option for software developers is the national and international network Research Software Engineers. Networking should also be strengthened within institutions, including the passing on of knowledge and experience to early career researchers.

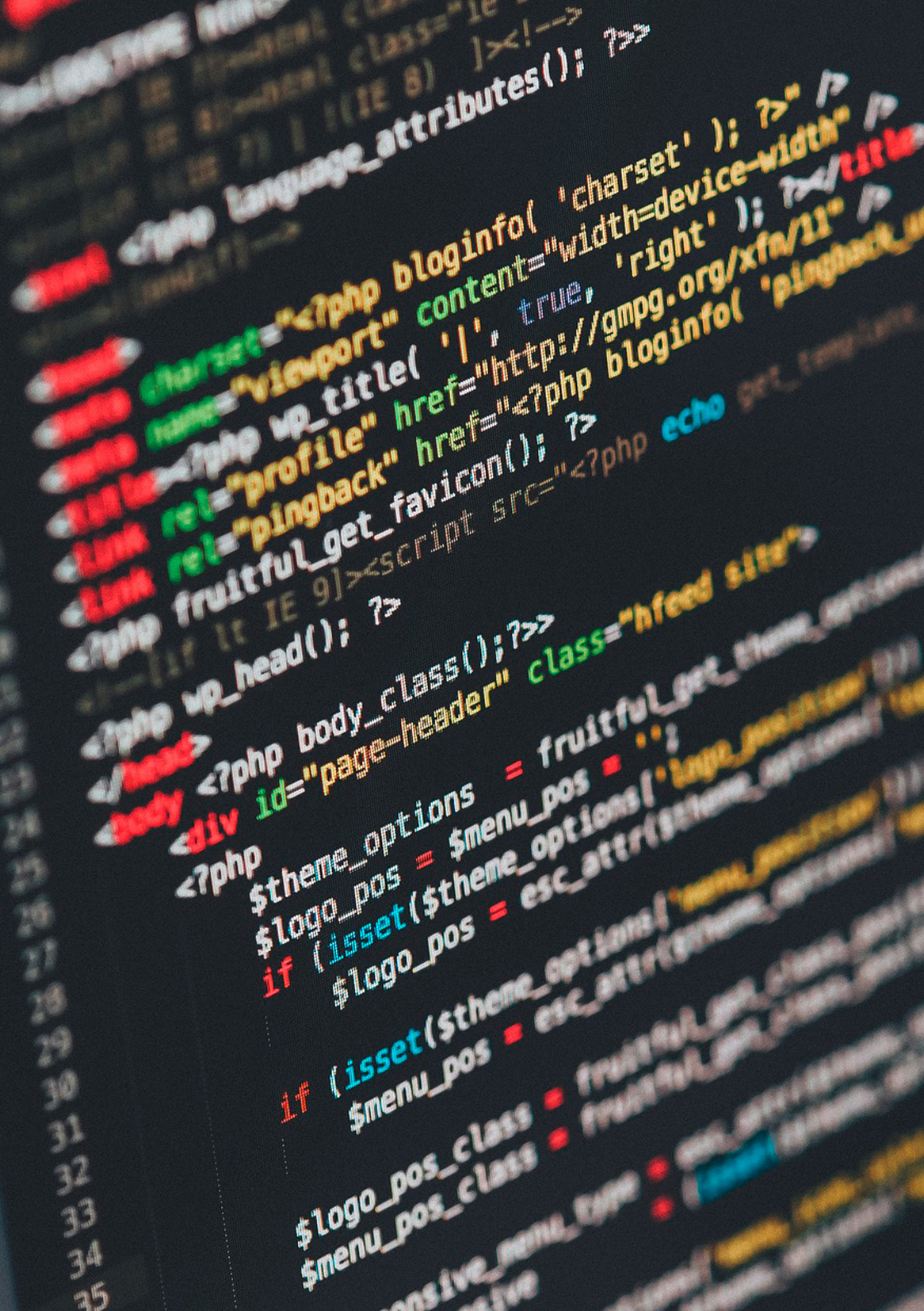
The *selection of research software* remains an ongoing challenge, but on principle preference should be given to open-source research software. Selection can be achieved through enhanced awareness in the review of articles and funding proposals, including for quality assurance purposes.

Priority initiative Digital Information

The priority initiative Digital Information is a joint initiative of the Alliance of Science Organisations in Germany aiming to improve the provision of information in research and teaching. With respect to digital information, the Alliance initiative aims to

- Shape and modify relevant operational structures and processes
- Respond to community-level initiatives and advance them to the implementation stage
- Reach agreement on direction-giving statements and policy papers and thus influence science policy debate
- Formulate joint positions of the research organisations on issues relating to digital information and introduce and represent these in science policy debate
- Promote sharing of information, agreement and communication on current issues
- Support and where necessary coordinate German participation in international initiatives
- Work together to draw up principles and guidance documents which are relevant to all research organisations

Between 2018 and 2022 the Alliance initiative will be faced with new challenges of the digital turn. Now that the digital transformation is well advanced, genuinely digital phenomena are beginning to play an essential role. This includes, for example, the international networking of research and teaching on digital platforms, new forms of digital publication and their evaluation as a research activity, machine analysis and interpretation of big data, and the importance of good scientific practice in the digital era.


charset="<?php bloginfo('charset'); ?>" />
name="viewport" content="width=device-width" />
<?php wp_title('|', true, 'right'); ?></title>
rel="profile" href="http://gmpg.org/xfn/11" />
rel="pingback" href="<?php bloginfo('pingback_
<?php fruitful_get_favicon(); ?>
<?php wp_head(); ?>
<?php body_class(); ?>>

<?php
$theme_options = fruitful_get_theme_
$logo_pos = $menu_pos = '';
if (isset($theme_options['logo_position'
$logo_pos = esc_attr($theme_options['
if (isset($theme_options['
$menu_pos = esc_attr($theme_options['
$logo_pos_class =
$menu_pos_class =

Imprint

Editors
This document was prepared by the Research Software Working Group, established in 2016, within the Alliance initiative.

The members of the working group are (in alphabetical order):
Mathias Bornschein, Prof. Dr. Björn Brembs, Dr. Michael Erben-Russ, Dr. Georg Feulner, Dr. Konrad Förstner, Michael Franke, Dr. Bernadette Fritzsch, Dr. Jürgen Fuhrmann, Prof. Dr. Michael Goedicke, Stephan Janosch, Dr. Matthias Katerbow, Dr. Uwe Konrad, Dennis Zielke

Lead authors
Dr. Matthias Katerbow
Deutsche Forschungsgemeinschaft (DFG)
Wissenschaftliche Literaturversorgungs- und Informationssysteme
Kennedyallee 40
53175 Bonn
E-Mail: matthias.katerbow@dfg.de
ORCID: 0000-0002-9770-7226

Dr. Georg Feulner
Potsdam-Institut für Klimafolgenforschung e.V. (PIK)
Postfach 60 12 03
14412 Potsdam
E-Mail: feulner@pik-potsdam.de
ORCID: 0000-0001-9215-5517

Digital Object Identifier
A digital edition of this document is available via
http://doi.org/10.5281/zenodo.1172988

German language edition
The original document is written in German and available via
http://doi.org/10.5281/zenodo.1172970

Production
Gabriele Wicker, Fraunhofer IRB, Stuttgart

Date
March 2018


DFG
Deutsche
Forschungsgemeinschaft


Fraunhofer


HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES


HRK German Rectors' Conference
The Voice of the Universities


Leibniz
Association


16
52
Leopoldina
Nationale Akademie
der Wissenschaften


MAX-PLANCK-GESELLSCHAFT


WR
WISSENSCHAFTSRAT