REPORT ON OpenDreamKit DELIVERABLE D1.4

Innovation Management Plan v1

BENOÎT PILORGET ET AL.

Due on	28/02/2017 (M18)
Delivered on	28/02/2017
Lead	Université Paris-Sud (UPSud)
Progress on and finalization of this deliverable has been tracked publicly at: `https://github.com/OpenDreamKit/OpenDreamKit/issues/20`	

DELIVERABLE DESCRIPTION, AS TAKEN FROM GITHUB ISSUE #20 ON 2017-02-28

- **WP1:** Project Management
- **Lead Institution:** Université Paris-Sud
- **Due:** 2017-02-28 (month 18)
- **Nature:** Report
- **Task:** T1.3 (#16): Innovation management
- **Proposal:** p. 34
- **Final report**

This is the first version of the Innovation Management Plan for OpenDreamKit. This report contains:

- ✓ Some context about the Virtual Research Environment and the end-user / developer relation;
- ✓ A description of transversal innovations OpenDreamKit is pushing for;
- ✓ Notes on open source (mathematical) software development processes, with a focus on some systems the VRE is developing;
- ✓ Elements of strategy for reaching out for a wide range of end-users.

CONTENTS

1. CONTEXT

OpenDreamKit (Open Digital Research Environment Toolkit, ODK for short) is a four-year Horizon 2020 European Research Infrastructure project (#676541). It provides substantial funding to the open source computational mathematics ecosystem, and in particular popular tools such as LINBOX, MPIR, SAGE, GAP, PARI/GP, LMFDB, SINGULAR, MATHHUB, and the JUPYTER interactive computing environment.

From this ecosystem, ODK will eventually deliver a flexible toolkit enabling research groups to set up Virtual Research Environments (VRE), customised to meet the varied needs of research projects in pure mathematics and applications, and supporting the full research life-cycle from exploration, through proof and publication, to archival and sharing of data and code.

The primary end-users are researchers in (pure) mathematics; however, thanks to the modular design the work will benefit a wide range of end-users, including teachers, engineers, in academia, public institutions, or enterprises.

A rather unique feature of ODK is that it emerged from an ecosystem of community-developed software where the prevalent development model is "by users for users". And indeed, many of the ODK participants are themselves end-users: academics wearing several hats – developer, researcher, teacher, in mathematics or computer science, etc. – that came to develop the computational tools they needed for their daily research. ODK itself was born from two dual movements, end-users pushing innovation and developers pulling innovation, often led the same people wearing both hats. This, combined with long dissemination experience from the said people, is a strong asset for delivering that actually benefits a wide range of end-users.

Reciprocally, ODK works hand in hand with the communities; some innovations brought to the toolkit are partially or entirely accomplished outside of the ODK project. This is not a problem in itself as the communities and the project share the same values and objectives.

End-users will benefit from innovations brought by ODK in the multiple pieces of software forming the opensource VRE. In the case of ODK the innovation will be the unification of opensource tools with overlapping functionality, the simplification of the tools for end-users without coding expertise, and the development of user-friendly interfaces.

The following document will first explain the innovations that ODK will bring to end-users, then explain the processes enabling the implementation of the said innovations, and finally will explain how ODK will target its end-users.

2. INNOVATIONS ODK IS BRINGING TO THE RESEARCH COMMUNITY

Recall that ODK has several transversal objectives, each of which brings innovation to the research community:

- Develop and standardise math soft and data for VRE: WP3, WP4, WP5, WP6
- Develop core VRE components: WP3, WP4, WP5, WP6
- Bring together communities: WP2, WP3
- Update a range of software: WP3, WP5
- Foster a sustainable ecosystem: WP3, WP4, WP5, WP6
- Explore social aspects: WP7
- Identify and extend ontologies: WP6
- Effectiveness of the VRE: WP2, WP7
- Effective dissemination: WP2, WP7

We now detail a bit two typical areas of innovations:

Best practice and tools for correct and reproducible research: We refer here to the excellent talk "Is your research software correct" by Mike Croucher which highlights crucial best practice whenever software is used in research, including open code and data sharing, automation, use of high level languages, software training, version control,

pair programming, literate computing, or testing. A lot of the work in ODK relates to disseminating this set of best practice (WP2: "Community Building, Training, Dissemination, Exploitation, and Outreach"), and enabling it through appropriate technology (WP4: "User Interfaces"). Just to cite a few examples, D4.6: "Tools for collaborating on notebooks via version-control", and D4.8: "Facilities for running notebooks as verification tests" enable respectively version control and testing in the JUPYTER literate computing technology, while Mike's talk is and will be delivered in several of ODK's many training events (see D2.2: "Community building: Impact of development workshops, dissemination and training activities, year 1" for the list of training events in year 1).

Multisystem architecture: Nowadays, research requires more and more the combination of several computational, database, and user interface components. We explore novel ways to combine software (WP3: "Component Architecture", WP4: "User Interfaces"), while taking advantage of parallel features (WP5: "High Performance Mathematical Computing"), sharing data and semantic in a sound way (WP6: "Data/Knowledge/Software-Bases"), and fostering collaboration between systems (WP7: "Social Aspects").

More details about the innovations ODK is developing and implementing will be available in the second version of this document at Month 45, when the VRE is much more mature.

3. IMPLEMENTATION PROCESSES OF THE INNOVATIONS

The success of the software in the ecosystem ODK builds upon owes much to the organization of the work process of the innovation. Their communities have, over the years if not decades, developed and accumulated a strong expertise in the social engineering aspects of community software development, pulling general ideas from the open source movement (e.g. public development, early releases, ...), and adapting them to their specific contexts. They are heavy users of the usual collaborative software development tools and best practice such as mailing lists, wikis, collaborative editing pads, online chat rooms, version control, issue tracker, continuous integration, regression testing, code reviews, coding sprints, etc.

We describe below some striking aspects of the implementation process in some of the software systems involved in ODK.

3.1. Implementation process of SAGE

All the development happens in the open (public mailing lists, bug tracker, ...). There is no specific sustainable leader in the development of SAGE. Its "community" of developers bases its work on the consensus of the group and on the availability of its members to tackle issues and work on the software development. If a decision doesn't create a consensus, there can be a vote but that seldom happens. When a development is brought to the software, it is reviewed by at least one other member of the community, more often than not with others looking over the shoulder.

Experience has shown that the success of SAGE, and in particular the biggest achievements and best innovations, owe much to a long track of focused week-long workshops, called Sage Days (about ten per year since 2005), where developers get together and form small groups for focused coding sprints. This method is actually close to the two-pizza team rule from Jeff Bezoes, the founder and CEO of Amazon. According to Mr Bezoes, beyond 6-7 people, the more you add people to a group, the less the group is agile and innovative because too much effort is put into communication and management.

An example of such workshop is Sage Days 77, organised by Nicolas Thiéry in April 4-8 2016, in a guest house far from urban civilization and with a solid internet connection. About 15 people joined this workshop throughout the week, and split into three or four constantly self-reorganizing teams. Concerning the impact, proper packaging and distribution has been a

recurrent issue for SAGE and is a major task for OpenDreamKit (**T3.3**: "Modularisation and packaging"). Major brainstorms occurred during the week to clarify the needs, isolate the core difficulties, and explore potential approaches to tackle them. The outcome was posted on the Sage Wiki, to be shared and further edited by the community. This fostered tighter collaboration between the packaging efforts for various Linux distribution, and triggered major progress on the Debian packaging side. Several small workshops such as this one are to be organised all project long to speed up the software development process.

3.2. Implementation process of JUPYTER

The JUPYTER project is driven by the Jupyter steering council, as described in the Jupyter governance documents. This council is composed of 15 members, one being Benhamin Ragan-Kelly who is leader of the Work Package 5 (User Interfaces), at least two being regular participants to ODK events.

The larger scale mission of the project is decided by the steering council. In most cases, the wider JUPYTER community decides what should be done by contributing proposals on an individual basis. Most proposals come in the form of pull requests on GitHub, but larger proposals can be discussed as enhancement proposals beforehand, and must be approved by the steering council. In general, decisions for additions are handled by the maintainers of existing packages, who are longstanding members of the community or delegates thereof (such as those hired under the ODK grant). If there is conflict, decisions can be resolved by the steering council.

Concerning the developments reviews, the JUPYTER community does it publicly. The maintainers of each project take on the bulk of the review responsibility. If there are conflicts, they can be brought to the steering council for resolution.

3.3. Implementation process of SINGULAR

Wolfram Decker and Hans Schönemann from the University of Kaiserslautern are considered to be the leaders of the SINGULAR software. There is no specific process to decide how the software should evolve. It can be individual decisions of a single person in need of a service for their research, or decision taken by vote by the core of SINGULAR developers during a meeting. Nevertheless the new code is essentially reviewed by Hans Schönemann.

3.4. Implementation process of GAP

Individuals who have helped or been helping in the development, the maintenance, the advisory and in the support for users are referred as the GAP Group. Furthermore, the GAP council was formed in 1995, which is currently consisting of 19 senior mathematicians and computer scientists with Leonard Soicher as chair. The GAP council is not a representative body but an editorial board. In that sense, the Council chairman acts as editor in chief, and the other members function as "associate editors," after the fashion of some journals.

In GAP, most issues concerning development are decided after discussion among the developers or the support team. Most of the discussions are happening in GitHub issues and pull requests, primarily in a former repository. Larger proposals can be discussed beforehand in the Open GAP development mailing list. Sometimes discussions may fail to reach a conclusion, either because the issues are too complex or because people simply can't agree. In this case a smaller core group will discuss and make decisions which have to be in the general interests of GAP.

For GAP packages, the decisions are taken by their authors and maintainers. Many of currently 132 packages redistributed with GAP adopt open development model which facilitates interaction between them and participation of the developers of the core GAP system in technical discussions. For the core GAP system, reviewing and evaluation is mostly done via public code review on GitHub and regression tests. For GAP packages redistributed with GAP, there are regression tests and checklists to use when a new package is submitted for the redistribution with GAP.

Finally, GAP features a longstanding formal package refereeing process overseen by the GAP Council. This process, rather unique and exemplary among academic software, aims at promoting high quality contributions by rewarding authors with credit similar to that of a published paper.

4. The end-users targeted at by innovations

As it was previously said, the originality of the ODK VRE is its "by users for users" model. Indeed, ODK's participants have many hats: they are mathematicians, computer scientists, developers, researchers, professors, end-users and sometimes all of this at the same time.

But since ODK is promoting open source, it is naturally aiming at reaching out as many end-users as possible. In order to do that ODK comprises two boards which can help reach out end-users outside the developer communities.

4.1. **Boards within ODK**

4.1.1. *Quality-Review board.* The Quality Review board (QRB) is composed of four members: Hans Fanghor (chair), Alexander Konovalov, Konrad Hinsen and Mike Croucher. They all have a long track of developing opensource research software and disseminating to end-users. Some of their objectives are to:

- Assess the quality of the software engineering aspects
- Identify best practice
- Improve future work within ODK and disseminate knowledge to a wider audience, i.e. potential end-users.

Thus, the work of the QRB can facilitate access of the end-user to the innovations brought by ODK by making sure best practices are respected and the software development is of high quality. However the board specifically turned towards the end-user needs is the Advisory board (AB).

4.1.2. *The Advisory board and its end-user group.* The AB is composed of seven members (Lorena Barba, Jacques Carette, Istvan Csabai, Françoise Genova, Konrad Hinsen, William Stein and Paul Zimmermann), and industry/end-users and academic representatives who understand broad 21st century needs for computational mathematics. The AB is to give an independent opinion on scientific and innovation matters, in order to guarantee:

- Quality implementation of the project,
- Efficient innovation management,
- Project sustainability.

Beside, the Board includes a small End-user Group (3 to 4 persons out of 7) which will be connected to an informal community of end-users. They will control the project execution from the point of view of the end-user needs and requirements, making sure that the outcome of the project indeed matches those needs.

4.2. **End-users targeted by ODK innovations**

Matching user needs is not sufficient. One additional challenge is to promote the VRE to potential end-users so that it actually gets put to use, in Europe and beyond, in established developer communities and across new users, in math and other relevant research fields. This challenge is being tackled by the WorkPackage 2 "Community building, training, dissemination, exploitation and outreach".

4.2.1. *A worldwide promotion.* As it was previously said, ODK is developing and promoting software that are opensource. The universal and cost-free distribution nature of opensource software allows ODK to target every country and continent notwithstanding any level of infrastructure development, economic performance or lack of a solid institutional academic network. In the first year of the project from Sept. 2015 to Sept 2016, 14 events (workshops, schools, etc.) were (co-)organised by ODK. Some of them were of course planned in Europe and North America, but others were planned in Africa, South America and Middle East. In the next years ODK will continue at the same pace to (co-)organise events in the same geographical regions. However in some areas where the internet connection does not allow for massive cloud usage, one must find alternative solutions such as distributing the required install files using USB sticks rather than online repositories. This simple trick enables dozens of undergraduates, master students, PhD students, postdoctorates, teachers and professors following the same given school to start working on a SageMath or Jupyter tutorial at the same time.

Because of the opensource nature of the ODK software, everything can infinitely be replicated and shared without any constraint. Therefore in addition and following the events organised by the project, ODK is counting on a snowball effect for the reaching out of end-users.

4.2.2. *Established communities.* Since ODK participants are all part of at least one of the well established opensource software communities, the communication on ODK's achievements comes naturally and easily. Furthermore, many workshops are organised all year-long, whereas they are (co)financed by the project or not, during which the developer communities join together their efforts to improve the software from the ODK toolkit.

4.2.3. *Research fellows outside of established communities.* Furthermore ODK was introduced at external events in order to reach out potential end-users outside of the regular opensource software developper communities. All these activities can be tracked in D2.2: "Community building: Impact of development workshops, dissemination and training activities, year 1".

All the pieces of software improved and developed by the project such as SAGE, SINGULAR, etc. were presented during these schools and workshops targeted at students and potential new users. Of course this software being originally developed for end-users who are potentially very skilled in computing and software development, one of the goals is to make it easy for relatively unskilled end-users to start using our opensource toolkit. For example, one of the main objectives in the development of SAGE in the attraction of new end-users is to develop the portability of SAGE (and therefore its dependencies) on Windows. With such a tool, the toolkit will enlarge its target to all Windows users in need of software like SAGE but without the time or knowledge to set it up themselves on their computers.

Three major conferences of about 60 to 90 participants are planned within the frame of ODK. One out of three was already jointly organised with the Collaborative Computational Project (CoDiMa) at the ICMS (International Center for Mathematical Sciences) of Edinburgh on the 10-16 January 2017 with about 50 to 60 participants. The topic was: Computational Mathematics with Jupyter. This conference aimed at enabling users and developers of GAP, SINGULAR, SAGE and JUPYTER to meet. Talks were given and coding hackathons were organised. The event is currently being evaluated by its organisers. The two other conferences will be organised each in 2018 and 2019.