

DEVELOPING INFRASTRUCTURAL SOFTWARE FOR PRESERVATION: REFLECTIONS OF LESSONS LEARNED DEVELOPING THE PLANETS TESTBED

Brian Aitken, Matthew Barr
Humanities Advanced
Technology and
Information Institute
University of Glasgow

Andrew Lindley
Safety and
Security Department
Austrian Institute
of Technology
Vienna

Seamus Ross
Humanities Advanced
Technology and
Information Institute
University of Glasgow
and
iSchool at
University of Toronto

ABSTRACT

The Planets Testbed, a key outcome of the EC co-funded Planets project, is a web based application that provides a controlled environment where users can perform experiments on a variety of preservation tools using sample data and a standardised yet configurable experiment methodology. Development of the Testbed required the close participation of many geographically and strategically disparate organisations throughout the four-year duration of the project, and this paper aims to reflect on a number of key lessons that were learned whilst developing software for digital preservation experimentation. In addition to giving an overview of the Testbed and its evolution, this paper describes the iterative development process that was adopted, presents a set of key challenges faced when developing preservation software in a distributed manner, and offers a real-world example of how lessons can be learned from these challenges.

1 INTRODUCTION

Planets (Preservation and Long-term Access through Networked Services)¹ was a four year project, partially funded by the European Community, that ran from 2006 until 2010. Its primary goal was to build practical services and tools to help ensure long-term access to digital cultural and scientific assets [7]. The sixteen consortium members brought together and further developed a huge knowledge-base of digital preservation research, with expertise pulled from national libraries, archives, leading research universities and technology companies.

Planets developed software that addressed several aspects of the digital preservation challenge. A variety of preservation action services were released to actively aid

¹<http://www.planets-project.eu>

in the process of the preservation of data. These include services for migrating data, such as the SIARD suite of tools for migrating relational databases to XML [8], and services for presenting data in emulated environments, such as GRATE [14]. Planets also focussed on the development of characterisation services which could extract properties from data and perform automated comparison of such properties, and the XCL [3] Extractor and Comparator were the principal outcomes of the project in this respect. A further aspect of digital preservation that the project addressed was the need for preservation planning services that can assess an organisation's specific preservation requirements and capabilities to help define a suitable preservation plan. The Plato [2] application was developed for this purpose.

In addition, Planets also identified the need for a Testbed for digital preservation experimentation, a collaborative research environment where preservation tools and services could be systematically tested and empirical evidence on their effectiveness and applicability could be gathered, analysed and shared. The need for such a research environment can be traced back to two related projects, the Dutch Digital Preservation Testbed project [12] and the DELOS Testbed for Digital Preservation Experiments [6]. These studies identified the need for research into digital preservation to be more engineering focussed, with a clearly defined rationale and methodology and an emphasis on a controlled set of experimentation to provide justification and validity to the choice of preservation approaches and services. Planets significantly developed and refined the underlying principles of these earlier projects, resulting in the web-based Testbed application that is now available to all interested parties for preservation experimentation.

Through the Testbed's online interface² the outputs of Planets are made available for experimentation, from preservation action and characterisation services through

²<https://testbed.planets-project.eu/testbed>

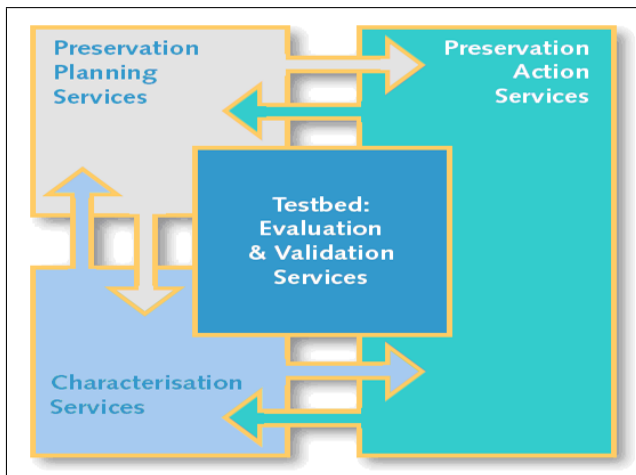


Figure 1. The Planets Software Components

to the executable preservation plans generated by the Plato application. Figure 1 demonstrates how the other software outputs of Planets interact with the Testbed application. The overall aim of the Testbed is not merely restricted to validating the success of Planets-developed software; the remit of the Testbed is considerably broader and a wide variety of third party preservation focussed tools are also made accessible for experimentation.

The background to the Testbed and an overall description of the facilities it has to offer has already been published in a number of papers [11, 1]. The primary focus of the current paper is to firstly give a general overview of the final version of the Testbed that was released during the Planets project, and then to investigate more closely the issues involved when engaging in a distributed preservation software development project. As the domain of digital preservation matures it is likely that an increasing number of preservation tools and services will be developed, both by research projects and by commercial organisations. By presenting and analysing some of the issues encountered during the development of the Testbed it is hoped that future development projects can learn from these issues and be prepared for certain challenges that are likely to emerge during the development process.

2 OVERVIEW OF THE FINAL VERSION OF THE TESTBED

The final version of the Testbed that was released during the Planets project was unveiled in April 2010, and a screenshot of this version can be viewed in Figure 2. The culmination of four years of development through eight point releases and several sub-point releases, this version of the Testbed provides a solid base for preservation experimentation through an easy to use web-based interface. To enable experimentation on preservation tools, access to these tools must be provided through the controlled experimentation environment. Within the final version of the Testbed roughly fifty preservation tools are available, each of which can be executed by an experimenter



Figure 2. The Planets Testbed version 1.2

using nothing more than a web browser and an internet connection. Each preservation tool is published in the Testbed via a web-service wrapper which exposes certain aspects of a tool's functionality, specifically those aspects that have particular relevance for preservation tasks. This 'networked services' approach is a core principal of the Planets project and it offers a standardised means of accessing preservation tools, providing users with the ability to execute experiments on tools that have a disparate set of hardware and software requirements, all from a standardised web-based access point.

Preservation tools which are wrapped as services and deployed in the Testbed are split into different categories depending on their function, thus enabling experiments of different focus to be designed and executed. Services offered include migration services, such as OpenOffice, Gimp and SIARD, characterisation services, such as the New Zealand Metadata Extractor and the XCDL Extractor and emulation services (identified within the Testbed as 'CreateView' services), including Qemu and GRATE. Other service types, such as identification and validation, are also offered and a complete list can be found through the Testbed website.

Access to sample data is also critical to successful preservation experimentation within the Testbed. The Testbed enables users to define a dataset for their experiment in three ways: by providing their own data, by accessing the Testbed corpora of sample data, or by combining these two approaches. Access to several corpora of test files are made available to experimenters through the Testbed interface. These corpora, comprising over eleven gigabytes of files, have been collected by the Testbed team during the course of the project and provide a broad range of test files that cover not only the major office, image, sound and video formats but specific versions of such formats where applicable. In addition, the corpora include a variety of 'edge case' files, such as GIF files that have experienced bit-rot. To ensure corpora files are ideally suited for experimentation, the properties of each file are documented us-

ing XCDL, with these measurements being stored alongside the files within the corpora.

In order to test the effectiveness of preservation tools the Testbed provides facilities to measure and analyse properties relating both to the tools and the digital objects that are manipulated by tools during experimentation. Property analysis represents the principal manner in which preservation tools are evaluated in the Testbed. Properties relating to a tool include its execution time and the success of its invocation while properties of digital objects include a very broad range of properties that can vary depending on the file type and the file contents. Example digital object properties include file size, bit depth, character encoding and sample rate.

The Testbed offers a variety of services that can automatically extract and measure properties for particular file formats, including the XCL tools, the New Zealand metadata extractor, Droid and Jhove. In addition, properties can be manually measured and the Testbed provides a predefined selection of properties plus facilities enabling experimenters to define new properties. By comparing the properties of the original digital objects with the post-preservation action digital objects, and taking into consideration properties of the preservation action tool during execution, it is possible to gather a detailed understanding of the effectiveness and suitability of the tool in question. The final version of the Testbed also provides facilities to evaluate individual property measurements, thus making it more straightforward to pinpoint strengths and weaknesses encountered during an experiment.

3 EVOLUTION OF THE TESTBED

As previously mentioned, the notion of a Testbed for digital preservation experimentation had its roots in the Dutch and DELOS Testbeds. From these relatively modest beginnings Planets aimed to significantly expand the capabilities of a digital preservation Testbed, providing web-based access to experiments, online experiment execution and a shift in emphasis to the automation of tasks such as experiment execution and property measurement. These core aims of the Testbed remained relatively static over the four-year duration of the project, but the details shifted markedly as understanding of the concepts grew and knowledge of the capabilities and limitations of the architecture developed.

Rather than leaping blindly into one single, lengthy and chaotic development period, the Testbed team followed the principals of iterative software development, with an initial period of detailed requirements capture feeding into a prototype, which in turn was tested, with feedback leading to a refinement of certain requirements that were then the target of a subsequent release. This process was repeated several times, with each release resulting in a greater level of functionality and a better understanding of the underlying requirements, which may have evolved significantly since the initial period of requirements elicitation. The iterative approach adopted by the Testbed developers

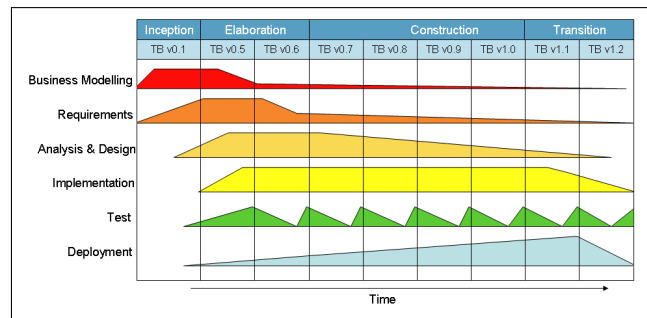


Figure 3. Development of the Testbed versions within RUP

was the Rational Unified Process (RUP) [10], and Figure 3 demonstrates how the incremental releases of the Testbed fit into the four phases and six disciplines of RUP.

At the beginning of the project, members of the Testbed group engaged in a period of requirements elicitation. This involved several face-to-face meetings where members of the team met and discussed the goals of the project and their role within it. This included a hands-on session with the Dutch Testbed software and the involvement of representatives from other strands of the Planets project in order to ensure that their notions of a Testbed were represented during the critical phase of requirements definition. This period lasted roughly six months and during this time documents were created that helped refine the initial direction of Testbed development. This began with a set of interviews with the content holding partners within the project, which gathered information on the facilities and functionality each partner desired from a Testbed environment, for example one partner defined a scenario involving the upload of a dataset, the passing of this data through a characterisation service, then through a migration service and finally through another characterisation service in order to compare the input and output results.

From the interviews a series of user scenarios were formulated, representing a distillation of the core functionality required by the various partners. From the user scenarios a further abstracted set of use cases and potential actors was defined, with each use case consisting of such items as ID, title, actors, preconditions and success scenarios. Roughly 60 use cases were defined for such tasks as uploading data to the Testbed and defining experiments. The next step in the Testbed design process was the creation of functional and non-functional requirements documents, which deconstructed the information contained within the use cases into short, demonstrable statements covering every aspect of intended functionality. The requirements document followed an industry standard template [9], with each requirement being assigned a unique ID, a priority level and references back to the originating use cases. The document could be referenced by members of the Testbed group and the wider Planets project to gain an understanding of the feature-set the developers hoped to be able to develop.

The Testbed requirements document defined what the

developers aimed to achieve during the course of development. However, it was not the intention at this stage to define exactly how these requirements should be implemented. The final stage in the initial design phase was the construction of a software design document, where formal definitions of the software components of which the Testbed would comprise were first formulated, class diagrams were mapped out, initial mock-ups of the Testbed front-end were proposed and the intended development environment and pre-existing software implementations were decided upon.

The initial detailed design phase of the Testbed lasted roughly six months, and by the end of this period a comprehensive set of requirements and design documents had been created, discussed, and refined. Following on from this the developers spent a further six months on the initial development of the Testbed API and the Testbed front-end, resulting in Testbed version 0.1, an HTML mock-up of the main pages of the Testbed that exhibited no real functionality but represented with a fair degree of accuracy the overall structure and layout of the final Testbed product.

Over the course of the remaining three years of the project eight major Testbed point releases were made, each of which expanded upon and refined the functionality found in the previous release. The implementation period for each release was between four and six months in duration and for each point release an implementation plan was formulated. Each implementation plan expanded upon the initial design documentation based on an increased understanding of the field, the capabilities of the software, feedback and requests from other project partners, and feedback from more formal testing sessions arranged by other members of the Testbed team.

The domain of digital preservation is not static; new research is constantly being published and the Testbed facilities which content holders desired and considered to be of the highest importance changed markedly over the course of the Planets project. Where possible a face-to-face meeting of all involved parties was held prior to the formulation of an implementation plan to ensure that feedback from the previous release could be gathered, areas where a divergence of understanding between developers and content holders could be pinpointed and addressed and the focus of the implementation period could be defined. The relatively short implementation periods and focussed point releases enabled the Testbed developers to address specific issues in each release and by publishing all implementation plans, minutes and supporting documentation on the project wiki it was the developers' intention to ensure that the decision making process and development status were as transparent as possible. As Testbed development progressed the iterations became gradually shorter and more focussed, taking on many characteristics from an agile software development framework such as Scrum [4], where a small focussed development team prioritises requirements and adapts to changing requirements through regular team meetings and updates.

4 CHALLENGES ENCOUNTERED AND LESSONS LEARNED

Throughout the four year development period of the Testbed the team noted some specific challenges and difficulties, some of which are unique to the domain of digital preservation, others which are more generally applicable to distributed software development projects. Each of these challenges has been a learning process and in the majority of cases the team identified a means to meet each challenge, or learned how to better address a similar situation in future. In this section a selection of these challenges and the lessons learned are presented.

4.1 Developing a preservation system for a variety of stakeholders is difficult

Planets involved a variety of different organisations, including national libraries and archives, research universities and technology companies. Different types of organisation and even different organisations of the same type had dissimilar and at times conflicting requirements for and demands from the Testbed software. Reaching a consensus as to the direction of development when 16 partner organisations are involved is difficult. From a logistical point of view it is infeasible to gather representatives from all organisations in one physical or even virtual location with any degree of frequency and even if such a gathering can be managed it is difficult for agreement to be reached.

This difficulty may be further exacerbated by a number of factors, as experience from the Testbed can demonstrate. Firstly, as a research project Planets involved many researchers from an academic background. Active and at times heated debate is crucial to the formulation of new ideas and to defend existing points of view, especially when researchers from different backgrounds interact. What is deemed less critical for such researchers is to reach a consensus on each discussion point, yet for software developers a conclusion to debates and a very definite pathway to follow is hugely important. Secondly, different representatives from partner organisations may be present at different meetings, and there is no guarantee that each partner institution will have a shared internal vision of the importance of certain aspects of the preservation software. Thirdly, the opinions of the stakeholders are not static; they evolve and change over time. Features that a stakeholder may consider of the utmost importance in year one of a project may easily become of minor consequence by the fourth year.

The Testbed team had to contend with these issues over the course of the project. Due to the conflicting nature of some requirements it was impossible to please everybody. For example, some partners deemed it of critical importance that certain Testbed experiments could be performed 'in private', with no experiment data being shared with other experimenters, thus enabling users to practice with the Testbed without exposing their mistakes or sensitive data to others. Conversely, other partners considered it vital that all experiments should be shared with other

users in order to build up the knowledgebase, the concern being that if users were given the option of experimenting in private then few experiments would be made publicly available and some experiments that ended in failure, but which still contained ground-breaking findings, would be hidden from view.

In order to address these issues the Testbed team attempted to find a middle ground that suited a majority of stakeholders where possible. As alluded to earlier, each implementation period featured a phase of internal testing where Planets partners could give their feedback on the current iteration, and building this feedback loop into the development period helped to minimise the risk of partners having unrealistic expectations of the software. The dissemination amongst partners of all plans and minutes also helped to alleviate this issue, and the iterative design method that was adopted ensured that requirements and overall goals were fluid enough to deal with a shift in focus over time.

4.2 Distributed development is more of a challenge than development at a single location

When engaging in the development of a preservation system, especially within the context of a research project where development is frequently entering into unknown territory, having developers working in isolation at different locations is not the ideal situation. Although there are many online collaborative tools that can help alleviate this issue, nothing is as effective as sharing an office with other developers and having the option of bouncing ideas back and forth.

The principal developers of the Testbed were based at three different organisations in three countries. In order to ensure effective communication the developers conducted weekly conference calls where open issues of a technical, design or organisational nature could be discussed and solutions could be formulated. In addition to this, the developers made frequent and efficient use of instant messaging systems to keep in contact and the use of a Subversion code repository ensured that code developments could be regularly distributed to other developers while minimising the possibility of conflicts within the code.

Effective use of such online collaborative tools was crucial to the successful operation of a distributed software development team, yet regular face-to-face meetings still proved to be essential. These helped to bolster the relationships between the developers leading to a stronger and more unified group, they improved developer morale and motivation and they also proved vital to problem-solving and decision-making. Having a day-long face-to-face developer meeting every few months provided a significant boost to productivity and was absolutely critical to the success of the Testbed, and on average three such meetings took place each year of the project. In addition to this the Testbed developers engaged in occasional longer 'exchange' visits, where a developer from one organisation travelled to and worked at another organisation for several

days. These visits also proved to be highly valuable to the development of the software.

4.3 Preservation software development can require a significant outlay of developer effort

Estimating resources for a software development project is a tricky business. This problem is not limited to the development of preservation software or to distributed software development, but it must be taken into consideration when a project is being planned. If a project plan detailing workpackages, effort and timescales must be created and agreed upon before the official launch of a project and if project-specific requirements elicitation and systems design tasks cannot commence until after an initial plan has been compiled it is unlikely that any initial software development timescales will be accurate.

The difficulty of estimating required effort was encountered within Planets with respect to the Testbed. In the initial plan it was assumed that the Testbed would be released within the first 18 months of the project, and that this release would be stable, fully tested, documented and usable by both project partners and external parties. This estimate proved to be unrealistic, which had an impact on a range of other project activities that had been planned. In retrospect, the reaction to delays in the release of the Testbed was perhaps not as prudent as it could have been. Workpackages and deliverables that relied upon a fully operational Testbed were not redesigned to take into consideration the updated circumstances and this led to some parts of the project being less effective than they otherwise might have been.

During the course of Planets new versions of the project plan were compiled every 18 months and to a certain extent the need for more developer effort for the Testbed and the need for more realistic timescales were reflected. However, developer effort proved to be a continuing point of difficulty for the Testbed throughout the project. Overall developer effort assigned to the Testbed as an average throughout the project was less than two full-time equivalents, and this was generally split between several individuals who were working part time for the Testbed. The final release of the Testbed demonstrates just what is possible to achieve with such a limited amount of developer effort but future projects should recognise that software development does require a significant amount of developer effort, and that a degree of flexibility must be built into timescales, deliverables and follow-up activities.

4.4 Staff turnover will be an issue for a project with a multi-year duration

A project that lasts four years and involves sixteen organisations cannot possibly expect to maintain the same staff for the duration of the project. It is inevitable that staff will move on and new members will join. This can have both positive and negative impacts on the project. New members can bring new ideas and innovative ways of looking at previously established practices and concepts, however

there is also the risk that staff who leave do not pass on their knowledge and expertise, and that the project is unable to find suitable replacements.

During the development of the Testbed both positive and negative aspects relating to staff turnover were encountered. Within the first 18 months of the project two Testbed members left, resulting in a period of several months where the involvement of certain partners was ambiguous. Thankfully another project partner offered to provide effort for Testbed development and the supplied member of staff proved to be extremely beneficial to both the development of the application and the refinement of the core Testbed concepts. The existence of an extensive body of documentation about the Testbed, both in terms of design documentation and wiki-based plans and definitions was crucial for ensuring new staff members could gain a detailed understanding of the Testbed in the shortest possible time.

A further staff related issue that must be considered is the potential difficulty in attracting people with a suitable skill-set, especially if a project is part-way through its lifespan. The Testbed required developers with detailed practical experience of JavaEE³, the Java Server Faces web application framework⁴ and the JBoss application server⁵ and finding candidates with such expertise who were willing to work on a relatively short-term research project proved to be a challenge. During the final year of Testbed development a key developer was promoted within his organisation, which would have resulted in the end of his involvement with the Testbed. The organisation in question advertised for a suitable replacement to take over development responsibilities but was unable to find anyone who was considered appropriate. The organisation allowed the existing developer to continue his involvement with the Testbed on a part time basis, but this illustrates the difficulties that a potential project must take into consideration with regards to staff turnover.

4.5 When developing preservation software it is crucial that the end product is developed with long-term access in mind

When developing software it is imperative that the functional and non-functional requirements of the intended users are identified. Within the context of digital preservation it is vital that in addition to this the long-term access requirements are also taken into consideration. Digital preservation practitioners extol the benefits of adhering to software standards, utilising open, non-proprietary software and formats where appropriate and ensuring adequate documentation is recorded. Software developed for digital preservation must lead by example in this respect.

The Testbed, and indeed the majority of the software developed during the Planets project took these concerns into consideration. The Testbed was developed using the

³ <http://java.sun.com/javaee/>

⁴ <http://java.sun.com/javaee/javaxserverfaces/>

⁵ <http://www.jboss.org/>

widely available and platform independent JavaEE and Metro technology stacks, with the widely established MySQL⁶ database used for experiment data storage. The Testbed code is stored in a Subversion repository and has been released under an Apache2 license. It is possible for anyone to download, inspect and further develop the code from the Planets Sourceforge site⁷.

However, some problems were encountered with the underlying technology used by the Testbed during its development. Due to the requirements of the core functionality provided by the Planets Interoperability Framework, the Testbed was reliant on a very specific version of the JBoss application server for the majority of the development period. This in turn required any computer on which the Testbed was compiled to be running an out of date version of Java, with newer versions causing errors. This reliance on an outdated version of Java was identified as a potential problem and was addressed during the final project year, illustrating the need to keep up to date with software developments whilst ensuring backwards compatibility with older software versions.

4.6 There can be conflicts and dependencies between different parts of a large-scale preservation software development project

If a project is large enough to be developing more than one piece of software through individual software teams then care must be taken to ensure that any interdependencies between these pieces of software are well documented and that delays or difficulties encountered by one team have a minimal effect on other teams. If one piece of software requires the delivery of a component being developed by another part of the project then effective communication between the teams is required and contingency plans that ought to be followed in the result of delays should be specified. Also, if different software applications are being developed within a project care must be taken to ensure that there is a clear distinction between the applications and that duplication of effort is kept to a minimum.

The Testbed is one part of a suite of software that was developed by the Planets project, with other software development taking place concurrently, including infrastructural software that falls under the banner of the 'Interoperability Framework' (IF), preservation tools, and other online applications such Plato.

The IF team was responsible for developing the core functionality required by the Planets applications, such as data and service registries, single sign-on services, and the workflow execution engine. Each of these components was required by the Testbed yet IF development was undertaken simultaneously with Testbed development. In some respects this approach was very valuable; Testbed and IF developers collaborated closely and the requirements of the Testbed were well reflected in the IF output. However, problems were also encountered when IF devel-

⁶ <http://www.mysql.com/>

⁷ <https://sourceforge.net/projects/planets-suite/>

opments took longer than anticipated. In some instances the Testbed was unable to meet its deadlines due to unavoidable delays with the release of IF software, and in other cases the Testbed group had to create and rely upon mock-up functionality for the short term. Close collaboration between the two groups ensured that such delays were communicated as swiftly as possible but difficulties were still encountered when certain events such as formal testing sessions had already been scheduled. A more effective approach may have been to ensure that the core functionality provided by the IF was already available for use before the development of the Planets applications commenced.

Within Planets there was also a certain degree of conflict between two of the applications being developed, namely the Testbed and Plato. Both applications shared a common origin, specifically the Testbed work carried out by DELOS. Under the umbrella of the Planets project a divergence of aims took place, with Plato focussing specifically on the generation and evaluation of organisation-specific preservation plans and the Testbed focussing on the benchmarking of specific technical capabilities of preservation tools under certain conditions within a controlled environment. Towards the beginning of the project the Testbed and Plato teams worked on their applications without a great deal of interaction and midway through the project it was observed that a certain degree of convergence had occurred, leading to some uncertainty and conflict between the two teams. Having identified the risk of convergence a greater effort was made to define clear boundaries between the two applications, a strategy that proved to be successful. From this point onwards the two teams engaged more closely and shared ideas and code more frequently, reducing any duplication of effort and ensuring both applications were interoperable where appropriate, specifically with results aggregation from the Testbed feeding into Plato and executable preservation plans from Plato being testable within the Testbed environment.

The Testbed group identified the lack of an overall software architect within the Planets project and would recommend such a role in a future project. The principal benefits of a software architect are twofold. Firstly s/he would be in a position to form an overall picture of the software developments and to a certain extent shape these developments and ensure that each independent development group is both aware of the work of others and can be presented with a distilled vision of where the work of their group is placed within the broader canvas of the project. Secondly s/he would be able to act as a buffer zone between the blue-sky research undertaken by academics and the software developers, who require very definite and clear plans for development.

4.7 Effective communication is a challenge within a large-scale project

In addition to communication challenges relating to a distributed development team as mentioned above, it was observed during the course of the project that communi-

cation between different workpackages and project areas was at times difficult to manage. With so many partners involved and such a wide variety of research and development activities being undertaken people tended to focus on their own silo rather than being able to formulate a complete picture of the project. This is a very difficult challenge to overcome in such a large project. The sheer number of publications, deliverables, wiki pages, and meetings means that simply keeping up to date with developments in one project area takes considerable time, and following the outputs of the entire project is much less feasible. This can result in synergies between different groups being missed and increases the risk of duplication of effort.

One area of Planets where this problem was effectively addressed relates to digital object properties. As mentioned earlier, these are vital to the evaluation of tool performance within the Testbed and for a long time different parts of the project were engaging in research into digital object properties independently and without much collaboration or awareness of each other's work. Towards the middle of the project members of the Testbed group became aware that a gap between different parts of the project needed to be bridged and moved to define a Planets-wide digital object properties working group. This working group brought a variety of project strands together and resulted in a shared Planets conceptual framework for digital object properties within the context of digital preservation, leading to some valuable research outcomes [5] and a standardised ontology based approach to properties that was adopted by the project as a whole.

5 CONCLUSION

Over the course of the four years of the Planets project the Testbed group successfully followed an iterative development approach to design, develop and refine a web-based application that both fulfilled the original remit and met the additional needs that were identified during the project. The end product is a stable and feature-rich web-based environment that can serve as a very solid base for research and experimentation within the field of digital preservation. The experiments database provides an extremely useful knowledgebase of the performance of digital preservation tools than can help broaden the understanding of digital preservation issues, and further experimentation can be continued through the application itself.

By the end of the Planets project more than one hundred external users had signed up as Testbed experimenters, with access to the Testbed's online presence being provided by HATII at the University of Glasgow. Active research into preservation using the Testbed has been carried out by Planets partners, for example one study performed research on the migration of a large corpus of TIFF images while another study investigated emulation, virtualisation and binary translation. External users have also begun using the Testbed to pursue their own research, and by the end of the project the Testbed environment had begun to receive positive online reviews [13].

As this paper has demonstrated, developing preservation software presents a number of challenges, especially when many disparate stakeholders are involved and the project duration spans many years. These challenges may be organisational in nature, such as issues relating to a distributed development team and the danger of conflicts and dependencies between development groups. They may relate specifically to staffing, such as the difficulty of managing staff turnover and attracting new staff with the correct skill-set. Challenges may also be of a technical nature, such as ensuring the software being developed follows best practice in digital preservation and ensuring a suitable development process is pursued. The Testbed team has addressed these challenges and has produced a stable product that can be further built upon and developed by subsequent projects.

Although Planets ended in May 2010, the Open Planets Foundation⁸ (OPF) has since been established to continue the innovative and highly beneficial digital preservation research and development that was spearheaded by Planets. The Testbed will continue to be managed, developed and supported by the OPF for the foreseeable future.

6 ACKNOWLEDGEMENTS

Work presented in this paper was supported in part by the European Union under the 6th Framework programme through initially the DELOS NoE on digital libraries (IST-507618), and then mainly through the Planets project (IST-033789).

7 REFERENCES

- [1] Aitken, B, Helwig, P, Jackson, A, Lindley, A, Nicchiarelli, E and Ross, S, "The Planets Testbed: Science for Digital Preservation" in. *Code4Lib*, vol. 1, no. 5, June 2008. [Online]. Available: <http://journal.code4lib.org/articles/83>
- [2] Becker, C, Kulovits H, Rauber, A and Hofman, H. "Plato: a service oriented decision support system for preservation planning" in, *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, New York, NY, USA: ACM, 2008, pp. 367370, [Online]. Available: <http://doi.acm.org/10.1145/1378889.1378954>
- [3] Becker, C, Rauber, A, Heydegger, V, Schnasse, J and Thaller, M. "A generic xml language for characterising objects to support digital preservation" in, *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing.*, New York, NY, USA: ACM, 2008, pp. 402406.
- [4] Cohn, M. *Succeeding with Agile: Software Development Using Scrum*, Addison Wesley, Boston, 2009.
- [5] Dappert, A and Farquhar, A. "Significance Is in the Eye of the Stakeholder" in *Research and Advanced Technology for Digital Libraries*, Springer, Berlin, 2009, pp. 297-308. [Online]. Available: <http://www.springerlink.com/content/r473n317t030/>
- [6] DELOS, "DELOS deliverable WP6, D6.1.1, Framework for Testbed for digital preservation experiments" in. 2004. [Online]. Available: http://www.dpc.delos.info/private/output/DELOS_WP6_D611_finalv2
- [7] Farquhar, A. and Hockx-Yu, H. "Planets: Integrated Services for Digital Preservation", *The International Journal of Digital Curation*, Issue 2, Volume 2, pp 88-99, 2007. [Online] Available: <http://www.ijdc.net/index.php/ijdc/article/view/45>
- [8] Heuscher, S, Jaermann, S, Keller-Marxer, P and Moehle, F. "Providing authentic long-term archival access to complex relational data," in *Proceedings PV-2004: Ensuring the Long-Term Preservation and Adding Value to the Scientific and Technical Data, 5-7 October 2004*, European Space Agency. Noordwijk, 2004, pp. pp. 241261. [Online]. Available: <http://arxiv.org/abs/cs/0408054>
- [9] IEEE *IEEE Recommended practice for software requirements specifications*, IEEE Std 830-1998, 1998.
- [10] Kroll, P and Kruchten, P. *The Rational Unified Process Made Easy: A Practitioners Guide to the RUP*, Addison Wesley, Boston, 2003.
- [11] Lindley, A, Jackson, A and Aitken, B, "A Collaborative Research Environment for Digital Preservation - the Planets Testbed" in, *1st International Workshop on Collaboration tools for Preservation of Environment and Cultural Heritage at IEEE WETICE 2010*, [Online]. Available: http://planets-project.ait.ac.at/publications/PlanetsTestbed_COPECH_08032010.pdf
- [12] Potter, M, "Researching Long Term Digital Preservation Approaches in the Dutch Digital Preservation Testbed (Testbed Digitale Bewaring)" in, *RLG DigiNews*, Vol 6 No 3, [Online]. Available: <http://digitalarchive.oclc.org/da/ViewObjectMain.jsp?fileid=0000070519:000006287741&reqid=3550#feature2>
- [13] Prom, C, Planets Testbed Review *Practical E-Records Blog*, 2010 [Online]. Available: <http://e-records.chrisprom.com/?p=1183>
- [14] von Suchodoletz, D and van der Hoeven, J. "Emulation: From Digital Artefact to Remotely Rendered Environments", *The International Journal of Digital Curation*, Issue 3, Volume 4, pp146-155, 2009. [Online] Available: <http://www.ijdc.net/index.php/ijdc/article/view/141>

⁸ <http://www.openplanetsfoundation.org/>