Shared Understandings in e-Science Projects

Peter Darch
Annamaria Carusi
Sharon Lloyd
Marina Jirotka
Grace de la Flor
Ralph Schroeder
Eric Meyer

A report from the ‘Embedding e-Science Applications: Designing and Managing for Usability’ project (EPSRC Grant No: EP/D049733/1)
# Table of Contents

Executive Summary .................................................................................................................. 3

1. Introduction .......................................................................................................................... 6

2. Case Studies: myGrid, preDiCT and HubLab ................................................................. 8
   2.1 myGrid .......................................................................................................................... 8
   2.2 preDiCT ......................................................................................................................... 9
   2.3 HubLab ......................................................................................................................... 9
   2.4 Methodology ............................................................................................................... 10

3. Findings .............................................................................................................................. 11
   3.1 Reward structures ....................................................................................................... 13
   3.2 End-users in competition with each other ................................................................... 15
   3.3 Designing for a range of users ..................................................................................... 16
   3.4 Conceptualizing the work contexts of other project members ..................................... 17
   3.5 Models of software development .............................................................................. 19
   3.6 Hierarchies ................................................................................................................... 20

4. Conclusions and Recommendations .................................................................................. 21

Acknowledgements ................................................................................................................. 25

Bibliography ............................................................................................................................. 25

Appendix ................................................................................................................................... 29

Author Bibliographies ............................................................................................................. 31

Author Contact Information .................................................................................................... 32
**Executive Summary**

This report forms part of the ‘Embedding E-Science Applications: Design and Managing for Usability’ project. Through case studies of three e-Science projects, it focuses on the development of shared understandings within the context of software engineering practices and development lifecycles between partners in e-Science project.

The Introduction (Section 1) introduces what is meant by the term *shared understandings*, explaining that partners in a collaborative project may have shared understandings about a variety of aspects of a project, including the project's overall goals and visions, the tools and software under development, and how to proceed with the work of developing these. The Introduction then explains the benefits to collaborative projects of achieving these understandings. These include: fewer costly changes in the direction of the project and of project partners’ work; the ease of anticipating and prediction what others are doing; a greater likelihood that the work produced by different project members will mesh effectively; and a reduction in frustration and conflicts can result from the achievement of shared understandings.

Finally, the Introduction argues that the particular distributed, multi-institutional, and multi-disciplinary nature of e-Science projects might pose particular challenges to the fostering of shared understandings amongst the members of such projects, yet there is little existing literature which deals with these distinctive challenges: this report is an attempt to fill this gap.

Section 2 introduces the three e-Science consortia which comprise the case studies for this report (myGrid, preDiCT and HubLab), and describes the qualitative methodology employed. For each case study, a series of semi-structured interviews was conducted with various members of the consortium, and an analysis of documents produced by the consortium was undertaken.

Section 3 reports the findings from these case studies. There were two broad underlying issues which were found to impede the development of shared understandings in the case studies. The first is that the distributed nature of the consortia meant that it was often necessary to communicate using Information Communication Technology (ICT) methods. This is covered in greater depth in another report for this project, *Collaboration and Communication in e-Science Projects*.

The second is that, because project members come from diverse institutions and scientific disciplines, they may be embedded in very different work contexts. This contributes to a number of issues which are discussed in this Section. These issues are:

1) **Reward structures.** Different project members are often embedded in different reward structures, which can lead to misunderstandings regarding the role played by end-users and their requirements in the software development lifecycle;

2) **End-users in competition with each other** (for instance, some end-users may be commercial rivals, whilst others may be research groups who are in competition for funding). This can result in reticence on their part in communicating their requirements as they may not wish certain information to become known to competitors;

3) **Designing for a range of users.** There may be different groups of potential end-users who, given that they work in different contexts, may have very different requirements. This can mean that it may not be possible to accommodate all of these requirements
within the limited timeframe and resources of an e-Science project, or that designing the tools and software to meet the requirements of one group of end-users may render it unsuitable for the needs of others;

4) **Conceptualizing the work contexts of other project members.** Developers may have difficulty in conceptualizing the contexts in which other developers are working. This can result in difficulties in implementing decisions if they are taken by project members who do not share the contexts in which these decisions are to be executed. It may also mean that developers find it difficult to anticipate what others require of them;

5) **Models of software development.** The variety of models that exist for software development may cause difficulties in integrating the work of project members. For instance, new project members may need to become accustomed to the models being employed by the project. Furthermore, groups from institutions who are working closely together may employ different models; and

6) **Hierarchies.** The different partner institutions involved in a project may have very different pre-existing hierarchical structures. There may therefore be differences between project members in terms of what they are used to in terms of how authority is exercised. Furthermore, they may feel the need to respect and conform to the existing hierarchies within their home institutions in addition to the project’s structures and lines of authority, which can create frustrations.

In light of these challenges, Section 4 concludes the report by detailing a numbered list of recommendations for practice. These are:

1) **Employing new members for a probationary period.** This provides a strong incentive for directing work towards the project’s goals because if someone is unable to demonstrate how their work contributes to the project, their contract may not be renewed;

2) **Research days.** These are days dedicated to allowing researchers to work on developing their research in a form that it can be presented to others in their field rather than focussing on working towards the project’s goals (Warr et al. 2007b);

3) **Assigning to each junior researcher a more senior member of the project to act as a “Champion”**. A Champion is a more senior member of the consortium whom the individual can approach for support and advice, and who can act as their advocate during meetings. This may help to increase a junior researcher’s confidence that their interests are being represented in project decision-making;

4) **Personnel exchanges and visits to other sites.** Even a short visit can promote understanding of the work context at other sites, for instance regarding the technologies and facilities available and the dynamics of the relationships between the project members;

5) **Workshops** which bring together project members, sometimes including end-users, from different institutions and get them to work together on tasks;

6) **Hackathons.** Developers from different partner institutions come together in one place to work collaboratively on software development for a period of days. This allows them to understand less familiar methods of software development they may be employed by others working on the project;

7) **Peer-programming.** This is where developers work in teams of two and develop a piece of code together. A new member of a project might be paired with a more experienced
member, and this could prove a good way for the former to get used to the model of software development employed in the project. Projects might also pair a developer at one of the partner institutions with a developer at another, in order to foster convergence in software development practices between the two institutions;

8) **Embedded users.** Having an end-user collocated with the developers will help to enable the end-user to understand the constraints upon developers. This end-user is possibly more able than developers to communicate these constraints to other end-users;

9) **Adopting a collaborative wiki approach to developing plans of work,** and allowing project members, including developers, the opportunity to contribute. This might enable those doing the day-to-day work to contribute their greater expertise regarding how the work is to be carried out, as well as being able to give realistic feedback about proposed timescales;

10) **One-on-one meetings with end-users.** For instance, developers might meet with representatives from one group of end-user at a time because they may feel more comfortable giving feedback about how the prototype meets their needs when competitors are not there. Additionally, an individual developer might be assigned to work closely with people from each group of end-users throughout the course of the project. This may help the development of trust between end-users and the developers through providing opportunities for building personal rapport;

11) **Involving representatives from commercial in the design of the requirements gathering process,** because they will be able to advise on how best to elicit commercially-sensitive information from commercial partners;

12) **Developing blueprints for future work beyond the end of the project.** This may allow end-users to remain fully engaged with the project even if they feel their needs are not currently being met because it can demonstrate how the work of the project might be built upon to better meet their needs in the future.
1. Introduction

e-Science projects are large-scale, multi-disciplinary, multi-institutional research projects, with the goals of developing technological systems and middleware to support highly-distributed computationally-intensive scientific research. They have been set up to support in a variety of academic disciplines, spanning the life and physical sciences, the social sciences, and the humanities. Typically, e-Science projects involve software engineers and developers working in close conjunction with researchers in particular fields in order to develop software and computational tools that closely match the needs of these researchers.

An in-depth case study of an e-Science project, eDiaMoND, identified the achievement of shared understandings within the context of software engineering practices and development lifecycles between e-Science project partners as a critical challenge for embedding e-Science applications (Warr et al. 2007a). The nature of e-Science projects may substantially increase the complexity of this challenge as shared understandings need to be fostered not only amongst project members who are collocated (i.e. who work in the same geographic location) but also amongst project members who are separated by team, institutional, disciplinary, and sometimes international boundaries.

This report aims to build on the eDiaMoND case study in order to:

- Identify issues arising in e-Science projects which may impede the achievement of shared understandings amongst project partners; and
- Suggest techniques and methods for addressing these issues.

Three case studies of distributed, multi-disciplinary e-Research projects have been carried out to try and achieve this. These projects will be introduced in Section 3, and Section 4 will present issues which have arisen in these projects which have posed challenges to the achievement of shared understandings and how (and to what extent) they were addressed. Finally, Section 5 will conclude by offering recommendations for addressing these issues.

First, however, the remainder of this Introduction will give consideration to what is meant by the term shared understandings, and why the achievement of these understandings is an important contributor to the success of collaborative projects.

In the existing literature regarding collaborative projects, 'shared understandings' is an umbrella term. In the case of e-Science projects, these aspects include the project's overall goals and visions (Warr et al. 2007a), the tools and software under development, including the features and user requirements that will be incorporated into them (Lloyd & Simpson 2005), and how to proceed with the work of developing these tools (such as timescales and deadlines, the project members' roles and responsibilities, and how these roles fit together, and the methods and work practices being used to develop the tools) (Mathieu et al. 2000; Hinds & Weisband 2003).

---

1 eDiaMoND was a pioneering flagship project which ran from 2002 to 2005 and was funded as part of the UK e-Science programme (www.ediamond.ox.ac.uk).
2 Only a brief summary of some aspects of a project about which shared understandings might be achieved is given here: a fuller picture will emerge during the presentation of findings from the case studies.
In addition to considering the aspects of a project that partners might have a shared understanding of, it is also important to distinguish between different senses in which two groups of people might be said to have achieved a shared understanding of a particular aspect of a project. The first is that the two groups have the same or a similar understanding of the aspect from the outset. This is evident in their way of talking about it, and in the actions that they think are appropriate with respect to it.

The second sense is that the two groups understand each other's perspectives on the aspect in question, although they do not necessarily share that perspective. For example, each understands what the other's goals with respect to that aspect are, but they have different goals themselves.

There are a number of ways in which the achievement of shared understandings can promote the success of a collaborative project. The first is that it can result in fewer costly changes in the direction of the project and of project partners’ work (Spencer et al. 2008). When a project’s goals are imprecise or poorly understood, this can lead to shifts in people’s direction as their understandings of these goals are liable to change over time.

The achievement of shared understandings also makes it easier to anticipate and predict what others are doing (Hinds & Weisband 2003). This means that collaborators can take action without having constantly to check and monitor the action of others. It also results in less duplication of work, because project members are more likely to trust that the work of others will meet expectations.

Furthermore, it can mean that the work produced by different project members is more likely to mesh effectively, resulting in more rapid development of end-products (Hinds & Weisband 2003). A shared understanding of the specifications required of the end-products and of the work practices and conventions of others is less likely to result in project members producing work that is inconsistent or incompatible with the work done by others. For example, developers very often require information from potential end-users in order to parameterise tools, and a shared understanding of the stage at which the data are required, what they will be used for, and the form it should take, will help to ensure that data is presented to developers in a timely fashion and in a format that can be used by the developers; conversely, it is useful for potential end-users to know when to expect, for example, to be presented with a prototype so that they can plan for its evaluation (for instance, ensuring that a sufficient number of end-users are recruited for testing, or arranging the venue where it is to be tested).

Finally, a reduction in frustration and conflicts can result from the achievement of shared understandings, thereby increasing the satisfaction and motivation of project members. (Spencer et al. 2008). If project members believe that there is a shared understanding of relevant aspects of the project, they are more likely to have confidence that the project’s end goals will be met and that the work they are doing is not wasted. For instance, if end-users have expectations that are unlikely to be met, this may lead to disappointment and impact upon their willingness to continue to be involved in the development process (for instance, in being part of focus groups or prototype testing).
2. Case Studies: myGrid, preDiCT and HubLab

We have conducted case studies of three consortia for this report. These are myGrid\textsuperscript{3} (which aims to support scientists working \textit{in silico}\textsuperscript{4}), preDiCT\textsuperscript{5} (which is in the field of pharmaceutical research), and HubLab\textsuperscript{6} (which focuses on the social sciences and humanities), and each will be described in more detail later in this section.

These consortia were chosen because it was felt that studying them would uncover a broad range of issues which might impede the achievement of shared understandings in e-Science projects, given that all three involve a number of barriers (including institutional, disciplinary, and international) across which shared understandings need to be fostered. It was also felt that they would reveal strategies and techniques undertaken within these consortia to address these issues, which would help inform the recommendations made in this report.

Furthermore, the three consortia were felt to complement each other well, for two reasons in particular. The first is that these projects began at different points in the past. preDiCT is a relatively new project (the case study was carried out in the first eighteen months of its life) so its members would be discussing issues (and strategies and techniques for addressing these) that were either arising at the time or were very fresh in their memory. By contrast, both myGrid and HubLab commenced a longer time ago (2002 in the case of myGrid and 2007 in the case of HubLab), therefore allowing for greater reflection on the part of stakeholders about issues relating to shared understandings, and the success (or otherwise) of steps taken to address them. Both projects have developed to the point where lessons learned are being reincorporated into future development.

The second reason is that the projects cover a wide range of different disciplines, in which researchers may have very different work practices into which computational tools might be integrated, and different habits regarding the use of computational technologies to support their work. HubLab allows for interesting contrasts with both myGrid and preDiCT in that it involves the social sciences and humanities, and therefore raises a different set of issues relating to a user community which is overall less used to working with computational technologies.

2.1 myGrid

The myGrid consortium was originally formed in 2001, and since its formation, has been coordinated at the School of Computer Science at Manchester University. The consortium has been through a number of phases, each involving Manchester University with a variety of partners, both academic and commercial:

\begin{itemize}
  \item \textsuperscript{3} www.mygrid.org.uk.
  \item \textsuperscript{4} An \textit{in silico} experiment involves the processing of experimental data using computational methods.
  \item \textsuperscript{5} www.vph-predict.eu. The VPH initiative is a collection of projects. Its main vision is the development of both patient-specific computer models to support clinicians to predict the course of a patient’s condition and to plan a course of treatment, and computer simulations of human physiology and disease (www.vph-noe.eu).
  \item \textsuperscript{6} See www.virtualknowledgestudio.nl.
\end{itemize}
- First phase (2001-2006). Manchester University plus five other academic partners and eight industrial partners (main funding from this phase came from the Engineering and Physical Sciences Research Council (EPSRC)\(^7\));
- Second phase (2006-2009). Manchester University plus one other academic institution (main funding for this phase came from the Open Middleware Infrastructure Institute (OMII-UK)); and
- Third phase (2009-2014). Manchester University plus one other academic institution (main funding for this phase came from the EPSRC).

The principal aim of myGrid is to support knowledge- and data-intensive *in silico* scientific work, for instance that carried out by bioinformaticians. To this end, it aims to build applications which will allow scientists to plan and execute *in silico* experiments, and to analyse and publish the results of these experiments. In addition to this core work, myGrid project members are also involved in associated projects where they may be working with partners, in some cases based in other countries, from a particular scientific field to help incorporate myGrid applications into their work\(^8\). One of these is myExperiment, which seeks to support and facilitate the sharing of data and workflows between researchers, and is a collaboration between teams at Manchester and Southampton Universities\(^9\).

### 2.2 preDiCT

preDiCT is a three-year project, commencing in June 2008 and funded by the European Commission as part of the Virtual Physiological Human (VPH) Initiative\(^10\). It is coordinated at the University of Oxford, and also involves four other academic institutions from a variety of European Union countries, three pharmaceutical companies and one data management company.

preDiCT’s aim is to produce software which will enable the simulation of drug interactions to predict the performance of pharmacological compounds on the heart’s rhythm, which could be used by pharmaceutical companies and academic scientists in drug development, to offer improvements upon current methods for determining whether candidates for drug development will have harmful side effects on the heart.

### 2.3 HubLab

Hublab was a project that ran from November 2007 until June 2008, based at the International Institute of Social History (IISH) in Amsterdam, Netherlands. This Institute has created a hub comprising of five international collaboratories, which are all in the area of social and economic

---

\(^7\) Note that The myGrid consortium has also received sponsorship from Microsoft, the Joint Information Systems Committee (JISC), the Biotechnology and Biological Sciences Research Council (BBSRC), and the Economic and Social Research Council.

\(^8\) An example of such a project is caGrid, funded by the US National Cancer Institute, which aims to support the sharing of data and collaborations amongst institutions researching cancer in the USA (www.cagrid.org).

\(^9\) www.myexperiment.org

\(^10\) The VPH initiative is a collection of projects. Its main vision is the development of both patient- specific computer models to support clinicians to predict the course of a patient’s condition and to plan a course of treatment, and computer simulations of human physiology and disease (www.vph-noe.eu).
history. These collaboratories are designed to allow researchers (typically demographers, sociologists, economists, and social and economic historians) to build, access, and analyse diverse and globally-distributed data sets. HubLab aims to study the working practices and the experiences of users within these collaboratories (Kok 2008).

HubLab is funded under the SurfFoundation’s Collaboratory Programme\(^\text{11}\), and is thus co-funded by the Virtual Knowledge Studio for the Humanities and Social Sciences and the hosting Institute.

\section*{2.4 Methodology}

All cases studies consisted of:

- Semi-structured interviews conducted with members of the consortia; and
- Analysis of documents produced by these consortia, including:
  - Funding proposals;
  - Project progress reports (such as annual reports); and
  - Papers and presentations produced by members of the consortia.

Drawing on data from a range of sources allowed for triangulation to take place (O’Donoghue & Punch 2003). It has been observed that the content of texts (including interview transcripts, reports etc.) is contingent on the context in which they are produced (e.g. the interaction between interviewer and interviewee) rather than simply reflecting an underlying reality. Triangulation involves the cross-checking of data from different sources (and thus which have been produced in different contexts) which helps to ensure that conclusions drawn from the data are not biased by the context in which the data are produced.

A total of seventeen interviews were conducted with various members of the preDiCT, myGrid, and HubLab consortia. The themes covered in the interviews\(^\text{12}\) were selected with the aim of drawing out interviewees’ experiences regarding (for a detailed schedule of interview questions, please see Appendix A):

- How they and their collaborators converged on shared understandings;
- The implications of failures in achieving shared understandings;
- How such failures were recognized as such and addressed; and
- The extent to which the moves taken to foster shared understandings were successful.

Interviewees were selected in order to provide a cross-section of consortia members. These included:

- Principal investigators;
- Project Managers;
- Application developers;
- Postdoctoral researchers;
- Doctoral students; and
- End-users of applications.

\(^{11}\) See \url{www.surffoundation.nl/en}.

\(^{12}\) For a detailed schedule of interview questions, please see Appendix A.
The data was analysed using a grounded theory approach (Strauss 1987). Interview transcripts and other documents were read closely, and a number of issues relating to shared understandings emerged from this. These documents were then coded according to these issues, allowing for both the frequency with which each issue arose to be seen, along with the context in which they arose. This meant that it was possible to see the ways in, and the extent to, which each issue impacted upon shared understandings in the consortia studied. Adopting a grounded theory approach meant that the findings in this report are data-driven, in the sense that they emerged from the empirical research rather than being imposed upon the data in a top-down fashion.

The findings from these case studies are presented in the following section.

3. Findings

This section presents a series of issues arising in the e-Science consortia studied that may have impeded the development of shared understandings amongst their members. For each issue, there is an explanation of what the issue is (supported by examples from our case studies), and the sense in which it makes the fostering of shared understandings more difficult.

There were two broad underlying issues which emerged from the case studies which were found to impede the development of shared understandings. The first is that the distributed nature of the consortia meant that it was often necessary to communicate via Information Communication Technology (ICT) methods (such as email, telephone, teleconferences), all of which have limitations compared with face-to-face meetings where all attendees are co-present. This is covered in greater depth in another report for this project, *Collaboration and Communication in e-Science Projects* (Darch et al. 2010). Briefly, this is a major issue because shared understandings are established from information conveyed during the course of conversations and communications, and the flow of this information is impeded by some of the features of ICT-mediated communication (Olson & Olson 2000; see also Clark 1996). For instance, many of the cues (visual, tone of voice, body language etc) that are present in face-to-face communication can be used by the individual who is speaking (for example, in order to add emphasis, indicate humour, or to solicit a response from the listener), but these are often missing in ICT-mediated communication (for example, visual cues are missing in the case of teleconferencing) (de Rooij et al. 2007; Veinott et al. 1999).

The second broad underlying issue is that, because project members come from diverse institutions and scientific disciplines, they may be embedded in very different work contexts, and it is this issue that will be focussed on in this section. It has been found that somebody working in a collaborative project will tend to conceptualize a task in terms of their own immediate work context; hence there may be differences in the way that collaborators embedded in different contexts approach a task (Beckkey 2003). There are many ways in which such contexts can differ. One is that there are differences in the work practices and methods employed at different sites. These include the model of software development that is used (for instance, a Waterfall method as opposed to more agile methods) (Goguen et al. 1992), the way in which decision-making processes (such as the level of the organization at which particular decisions are made, who is able to contribute and the nature of the contribution they are permitted to make, and how such processes are resolved). Furthermore, there may also be differences in the technologies and technical support available at different sites, which
can constrain the work that can be carried out at these different sites, and the methods that can be used to perform this work (Hinds & Weisband 2003).

There are also significant differences between the work carried out in different scientific disciplines, for instance regarding the phenomena under investigation, and the experimental methods and modes of reasoning employed to arrive at scientific knowledge, as well as there being differences in the extent to which scientists in different disciplines collaborate and the typical size of such collaborations (Knorr-Cetina 1999).

A third difference in work contexts is that there are also likely to be substantial differences in the goals that the various institutions possess and are seeking to pursue through their involvement in the project. For example, a commercial company’s overriding goal is likely to be the production of saleable goods, whereas an academic institution is driven by the need to secure funding for future research. The hierarchies and communication systems within these institutions are then likely to be organized in a way that it is thought best allows the institution to pursue its respective goals.

A final difference in the work contexts of individual project members is in the career and incentive structures in which a project’s members are embedded. These can vary for a number of reasons, including the stage of an individual’s career, the tasks and accomplishments for which an individual might be rewarded (this can vary across institutions and disciplines), and the form this reward might take (for instance, a qualification, a publication, promotion, a chance of future research funding).

The issue of diverse work contexts gives rise to a number of sub-issues within the case studies. Each of Sections 3.1-3.6 will deal with an individual sub-issue, describing the issue and its particular impact upon the development of shared understandings (supported by examples from our case studies), before detailing some recommendations for addressing the particular issue.

Section 3.1 deals with the impact of the different incentive structures in which different project members might be embedded, explaining how it may lead to misunderstandings regarding the role played by end-users and their requirements in the software development lifecycle. Even when shared understandings have been reached regarding this, however, there may still be difficulties in achieving shared understandings of what end-users’ requirements actually are. One issue which might create such a difficulty is that different groups of end-users may be in competition with each other, which can result in reticence on their part in communicating their requirements: this is explained in Section 3.2.

It should also be noted that there may be different groups of potential end-users who, given that they work in different contexts, may have very different requirements which it may not be possible to accommodate within the limited timeframe and resources of an e-Science project: Section 3.3 discusses the impact of this upon shared understandings.

Section 3.4 will consider the difficulties that a group of project members might have with conceptualizing the work contexts of other project members.

Finally, Sections 3.5 and 3.6 will explain the effects that two particular differences that may exist between various project members’ work contexts have on such members’ abilities to work together across these contexts: Section 3.5 deals with the differences in models of software development that
3.1 Reward structures

Stakeholders may be embedded in very different incentive structures and this can impact upon the extent to which shared understandings develop amongst project members. Two particular reasons for differences in incentive structures will be considered here. The first is that there are variations in the typical career structures between the scientific disciplines in which project stakeholders work. The second is that different project members may be at different stages of their careers and therefore the short-term incentives they face may vary. This problem has notably been encountered in both myGrid and preDiCT in the case of doctoral students and postdoctoral researchers.

These may lead to differences in the behaviour of, or tensions between, project members as they orient their work towards pursuing their own particular career goals (Spencer et al. 2008). The nature of these differences, the impact they have upon the development of shared understandings, and how they might be addressed in e-Science projects will be discussed in the following two sections which deal in turn with the two reasons listed above.

However, it may also occur that different career paths are being opened up by e-Science and related projects. For example, the very important role of the project manager in the HubLab project seems to indicate that this is occurring.

3.1.1 Software engineers and research scientists

A project’s stakeholders are likely to be seeking to further their careers within their own disciplines. There may well be variations between disciplines regarding which accomplishments or work are rewarded, with the effect that project members from different discipline may be seeking to accomplish different things within the project, as the following extract from an interview illustrates:

“There are different agendas. So there’s the agenda of a computer science researcher, which is different to the agenda of a software engineer, which is different to the agenda of a [domain scientist].” Interview 02/iv-a

For instance, as Bos et al. (2008) describe, the incentive structure for computer scientists is such that they are usually rewarded for technological innovation, rather than for the functionality and applicability of their software in the actual work context where it is intended for use; conversely, scientists working in the domain for which the tools are being developed are incentivized to produce original scientific results, and would like developers to provide them with software that supports them doing this with little concern about whether the software is innovative (see also Ribes & Bowker 2008). This is illustrated in the following interview extract:

“Even today, it’s a struggle to make sure that everybody keeps on track about what are the requirements, and we still make mistakes about this, despite the fact that we have structured mechanisms of focus groups...and mailing lists and all sorts of things. It’s still difficult to make sure that software engineers don’t go off and build something that is interesting or refactor all their code in an interesting way when the real problem is something else altogether” Interview 01/i-a
This can potentially lead to differences between project members in their understandings of various aspects of a project, because they may conceptualize these in terms of their own interests and reward structures. In the cases studied here, there were instances of misunderstanding between developers and end-users about the overriding vision of the project and, as a result, the nature of the software under development. Whilst potential end-users wanted a product that would do nothing more than support their work, there was often a tendency on the part of the developers to focus on producing software that was as accomplished as possible in technical terms rather than in terms of usability. For instance:

“We found out [when we met the end-users] that they don’t actually need very, very fancy software tools... We’ve been learning that the tools being developed will not replace completely existing methods, but rather will complement them” they need us to make [their work] easier” Interview 02/ii-a,

Here, the developers initially believed that their task was to develop *in silico* tools that would completely replace the scientists’ existing *in vivo* and *in vitro* methods, and it took some time for them to appreciate that the scientists instead intended to integrate these tools into their existing work practices. The developers did not share the work contexts of the end-users, and therefore may not at first have appreciated the extent to which (amongst other things) the end-users trusted their existing methods to provide accurate data and to enable them to make sound scientific conclusions, and desired computational tools which would enhance these conclusions rather than replace their existing methods wholesale.

3.1.2 Doctoral students and postdoctoral researchers

Junior researchers are often very focussed on developing their careers within their academic discipline, and pursuing this may result in their work diverging from the project’s work.

In the case of doctoral students, one interviewee stated that, although they felt the students had a clear grasp of the goals of the project and of the particular part of the project in which they were working, some did not integrate fully into the project:

“You have DPhil [doctoral] students who are doing their own research and they are sitting at their computer and not interacting with others” Interview 02/i-a

A reason for why this has been the case is that the incentive structure for doctoral students can cause conflict and tensions. A doctorate is an increasingly necessary first stage in the career of an academic researcher or a scientist working in industry. Typically, a student is awarded funding to pursue a doctorate for a fixed length of time, with the goal of producing a thesis which makes an original contribution to their field of study, as well as producing journal articles and making conference presentations to others in their field. These demands can compromise a student’s willingness to direct their work towards the project’s goals in at least two ways. One is if they do not believe that doing so will allow them to produce original research. Another is that arranging and presenting their work in a way that is acceptable for their thesis and journal articles is unlikely to be the way in which their work needs to be presented to be meaningful to others in their project: prioritizing the former will compromise the time available for the latter.

In the case of postdoctoral researchers, some similar issues have arisen. Under pressure to establish themselves within their field, these early-career researchers also need to publish articles and present
at conferences. Furthermore, they may also desire a certain degree of independence in order to develop their careers in the direction they wish. The following excerpt from an interview suggests that tensions between this and the needs of the project may arise:

“If you recruit someone straight from their DPhil...and what you want when you recruit them on to a project is to pull them in the direction of your research agenda. So you’re pulling them away from this independence, whereas their natural reaction would be to try and grow their research area themselves” Interview 02/iv-a

If the overriding incentives for doctoral students and post-doctoral researchers are not related to those of the e-Science project, then this may compromise their motivation to gain a thorough understanding of many things, including the visions and goals of both the project and the particular part of the project within which they are working, and how their own work is intended to fit into the work of the project (such as how it related to the software and tools the project aims to produce; and the plans of work, such as how their work relates to the work of others, including presenting their work in a way that is timely and meaningful to their colleagues).

### 3.2 End-users in competition with each other

Another issue arising from diverse work contexts that different groups of end-users may be in competition with each other (such as commercial organizations, or research groups in the same field who may be in competition for funding) and this can impede the sharing of information amongst these groups or sharing of information with developers, impacting upon the ease with which developers can gather users’ requirements.

For instance, commercial stakeholders in e-Science projects may wish, for commercial purposes, to keep competitors from gaining information about the work practices of those within their organization or details of the products they have under development. This can impact upon their willingness to share requirements with developers or to give feedback on prototypes or new releases of tools, as the following quotation illustrates:

“It must be remembered that the companies are competitors so they do not feel comfortable in sharing their views of what should be incorporated into the tools in front of representatives from other companies because this may be commercially-sensitive information” Interview 02/iv-a

Here, it seems that the representatives from the companies felt inhibited by the presence of members of other companies from giving feedback to developers regarding the prototype. They did not feel comfortable discussing how the product under development might, for instance, be integrated into the existing work practices of people working in their own company.

This poses clear risks to the prospect of developing tools that will prove usable. In particular, this issue poses a challenge to the development of shared understandings between developers and users about a number of aspects relating to the development of the tools, including the users’ requirements and needs in the sense that developers may be more likely to have an incomplete understanding of users’ needs, the features that will be incorporated into the tools, in the sense that developers and users may be less able to discuss freely and come to agreement about what should be incorporated, and the potential benefits that the tools may offer to potential users, because these
users may feel inhibited from asking questions about how the tools might fit into their existing work practices.

### 3.3 Designing for a range of users

An e-Science project typically involves designing applications for a community of end-users who may have a divergent range of requirements: there can be significant differences in the work practices of research scientists working in different institutions and disciplines, and they may be embedded within very different hierarchical structures and organizational cultures (Knorr-Cetina 1999). Indeed, as will be seen below, even if the group of potential end-users is at first sight somewhat homogeneous, there may still nevertheless be substantial differences in their requirements that are challenging for developers.

Two particular impacts of this have been identified in our case studies. The first is that it may not be possible to meet all of the expectations of end-users. The second is that it may result in developers focusing on designing and building for one particular group of end-users, with the possible result that what is developed is unsuitable for other end-users. These two impacts are discussed in turn in the following two sections.

#### 3.3.1 Resource constraints: not all end-user requirements can be met

One consequence of having end-users with different requirements is that it may not be feasible for an e-Science project to be able to deliver applications that meet all requirements of end-users, given the limited resources available and time constraints. An example of this occurred in preDiCT, where the end-users (pharmaceutical companies) initially wanted a suite of tools which they could deploy internally on their own computer systems and on which they could use their own data. However, despite all being pharmaceutical companies, they each have different methods of formalizing and curating data, and hence the tools would need to be fine-tuned for each individual company. This, however, has not proven feasible given the resources available, and instead the developers are working towards a system hosted at the University of Oxford Computing Laboratory, using publicly-available data (see Table 3.1).

<table>
<thead>
<tr>
<th>What pharmaceutical companies initially desired</th>
<th>What the developers believed was feasible</th>
</tr>
</thead>
<tbody>
<tr>
<td>A system they could deploy internally</td>
<td>A system hosted within the University</td>
</tr>
<tr>
<td>To use their own data</td>
<td>Using non-commercially sensitive, publicly-available data</td>
</tr>
</tbody>
</table>

*Table 3.1 End-users’ initial expectations of what preDiCT would produce, compared with what is considered feasible by preDiCT’s developers*

#### 3.3.2 Different end-users may have very different needs

Another possible consequence of having a diverse community of potential end-users is that if an application is developed according to the requirements of one group of end-users, it may prove unsuitable for the needs of other groups. Instances of this have arisen in myExperiment. The pilot phase of myExperiment aimed to support four groups of scientists, including bioinformaticians based in Manchester and chemists based at Southampton University. The tools that were initially
developed by developers working at myGrid in Manchester to support sharing were finely tuned to
the needs of the bioinformaticians in Manchester:

“myGrid [in Manchester] is focused more on biologists and bioinformaticians...The sharing model that we
developed was very much about biologists working in silico” Interview 01/iv-a

This model allowed individual bioinformaticians (including junior researchers) choice of with whom
(e.g. specific individuals, particular teams, or to everybody) to share their data and workflows, but
this model of sharing proved less suitable for the chemists, because they tended to be embedded in
hierarchical structures in which junior researchers had less autonomy regarding whether to share
with others:

“[The chemists] have a much more hierarchical world with the students and the professors...and they would
prefer a system that was much more organised in terms of people and hierarchies of people and hierarchies of
sharing, so someone might share with their professor who would then decide who to share it with after that”
Interview 01/iv-a

There are a number of possible reasons given regarding why this initial sharing model better met the
needs of one group of end-users than another. One is that the developers in Manchester had a
greater familiarity of the developers in Manchester with bioinformaticians than with the chemists in
Southampton, because they had more prior experience of developing for bioinformaticians given
their other work on myGrid and because bioinformaticians and developers were located in the same
University, which enabled easier communication flows between bioinformaticians and the
developers in Manchester, including communication of requirements and feedback from
bioinformaticians on the software under development. Furthermore, given the time and resource
constraints, it was not possible to develop for all groups of end-users and therefore the needs of one
group were privileged over those of another.

In the HubLab project this issue was also experienced in that the adaptations that had to be made in
the initial platform chosen for Virtual Research Environment and the end users. The platform
(LifeRay) had been designed for business purposes and contained many more capabilities than the
researchers were actually interested in. This made it more difficult for researchers to learn to use the
platform and was offputting since researchers felt that they did not have time to spend learning how
to use an over-complicated piece of technology on top of their research.

The fact that the potential end-users of a project’s tools form a very heterogeneous group can
impact on shared understandings, in particular regarding the features of the tools under
development and the work practices they are intended to support. Failing to achieve a convergence
between the expectations of the potential end-users and the developers regarding what is actually
being developed could cause frustrations and diminish the willingness of potential end-users to
remain involved in the project and continue to communicate requirements and evaluate prototypes.

3.4 Conceptualizing the work contexts of other project members

Another way in which diverse work contexts can impact upon shared understandings is that it can
make it difficult for people to conceptualize the work context of colleagues in other institutions,
disciplines or geographic locations. Two major effects of this have been observed in these case studies, and will be explained in this section.

The first is that decisions about what work is to be carried out and about what particular features the tools under development should include may be taken by people who do not share the work contexts of those who will have to do this work. This may result in difficulties in implementing these decisions effectively, as this following quotation illustrates:

“Sometimes the decisions taken in higher positions in the hierarchy...somebody takes the decision but it's not all about analysing it, seeing what it implies, seeing which requirements it generates, deciding the tasks, assigning the tasks to different researchers, so I think that sometimes decisions are taken by Work Packages or managers but... we only get the overall concept, but that needs to be translated more into tasks that can be identified and implemented, the time that it takes to implement them and so on.” Interview 02/v-a

This is attributed to decisions being made by members who are more senior in the project hierarchy than those who must implement the decisions, and it may be further complicated in e-Science projects because those making decisions may be remote from those who have to carry out the work to implement them in a number of senses. One is that those making the decisions are geographically-remote from those who might implement them which may reduce opportunities for decision-makers to observe, and thereby understand, the work practices and organization of developers. Additionally, they may come from different institutional or disciplinary backgrounds, and may therefore have little experience of conceptualizing their decisions in terms of the work required for them to be implemented.

The second effect of this is that it can make it difficult for developers themselves to understand how their work might fit in with the work of others, as the following quotation suggests:

“Sometimes [another work package] require new functionalities and we get requirements from them...in the initial estimate of effort, nobody really took into account requirements coming from other work packages. The problem is that sometimes we've got really tight schedules in order to write the deliverables and implement the functionality, but maybe they didn't allow enough time to consider requirements that may arise throughout the project.” Interview 02/i-a

Here, it was difficult for one group of developers to anticipate what other teams required from them in order to perform their own work, and this made it difficult for this group to plan their work in order to ensure smooth collaboration with others. In general, smooth collaboration could be facilitated by greater awareness between groups of what is required from group A so that group B can fulfil its tasks, and when group B require this, and what input can be expected from group B in order to facilitate group A’s meeting of this requirement, and when they can expect it to be produced.

The issue highlighted above can impact upon shared understandings, in particular regarding how the day-to-day work of the project is to be planned and executed (including: how the work might be broken down into various tasks that need completing, and how these tasks are interrelated; the work practices and methods that might be used to perform these tasks; the resources required to perform these tasks; which project members have the competencies to perform various tasks, and how the
roles assigned to different members might fit together; and realistic timescales and deadlines, both in terms of end-deliverables and interim milestones).

### 3.5 Models of software development

The process of developing software involves the integration of some or all of the following phases:

- Gathering requirements from potential end-users;
- Designing the software;
- Writing the code;
- Testing for bugs;
- Testing with potential end-users; and
- Maintenance of software.

A number of models of software development have been developed, which integrate the above phases in different ways (Sommerville 2000). For example, in the Waterfall model, the phases listed above occur sequentially, so that each phase is completed and its outcomes fixed before the next stage occurs (Royce 1970). A contrasting model is *agile programming*, in which software is produced using a series of relatively short iterations, typically a matter of weeks, with each iteration involving some combination of the above listed phases and frequent communication between the developers involved (Abrahamsson et al. 2002). Different institutions may choose to employ different model. The choice of a particular software development model makes a great difference to the organization of the development team: for instance, a waterfall method lends itself to a more structured, top-down hierarchy, whereas teams using agile methods usually need to be more self-organizing (Abrahamsson et al. 2002).

The existence of a wide variety of models for software development among the partners of collaborative projects can pose significant challenges for developing shared understandings in e-Science projects. These projects involve bringing together different institutions, which may have divergent ways of organizing the software development process. For instance, there were differences in myGrid between the partners based in Manchester and their colleagues at the European Bioinformatics Institute (EMBL-EBI), as the following interview extract illustrates:

"[The EMBL-EBI are] much more structured, they have a much more organizational/managerial style…from the EBI's point of view, they weren't used to the same level of Skype chats every day [as us], telecons regularly, and reporting and reprioritising. A big part of this was the Agile nature of the way we developed software…I don't think the EBI had that initially” Interview 01/iv-a

Because different partners may come to a collaboration with divergent conceptions and experiences of how software is to be developed, it may be challenging for the partners to arrive at shared understandings of plans of work, for instance agreeing upon what tasks need to be done and the deadlines for completing these tasks, assigning roles and responsibilities to various individuals and determining how these roles are to fit together.

The issue of software development models also arises when new people are recruited to a project since they may be used to a very different way of working to that employed by the project. Ensuring
that new developers adapt to the project’s models is very important if developers are to understand their role and responsibilities and to integrate well into the project.

3.6 Hierarchies

Different institutions involved in an e-Science project may have very different organizational structures and hierarchies. There may therefore be differences between project members in terms of what they are used to in terms of these and how authority is exercised. Furthermore, they may feel the need to respect and conform to the existing hierarchies within their home institutions in addition to the project’s structures and lines of authority. These factors can give rise to a number of issues which can impede the development of shared understandings amongst project members.

One such issue is that it can impede the flow of communication within a project, complicating lines of communication and delaying flows of information. This is covered in greater depth in another report, *Collaboration and Communication in e-Science Projects* (Darch et al. 2010).

Another major issue is that involving participants from a range of institutions can often complicate project hierarchies. This can lead to a number of problems in achieving shared understandings. One of these is that it can lead to delays in the resolution of disputes and decision-making processes, possibly leading to frustrations, as the following quotation illustrates:

“In [the other institution], the hierarchy was much more important so if the project leader wasn’t there, the people underneath would be waiting for an answer from them” Interview 01/vi-a

Here, two groups from different institutions were working together and, in such cases, decisions about how to proceed with the day-to-day project work need to be taken across institutional boundaries. The problem here is that people may seek to resolve such decisions according to the existing hierarchical practices in each institution before resolving them across institutional boundaries, leading to delays in decision-making.

This can impact upon shared understandings amongst project members regarding the work that various individuals or teams are performing. For instance, somebody may be under time pressure to complete their work but may be expected to wait for a decision to be taken before they can proceed: if decision-making is delayed, they may decide they cannot afford to wait any longer and go ahead with their work, thereby contributing to misunderstanding and confusion regarding what they are actually doing.

As well as causing delays in decision-making, bringing together participants from a range of institutions can also give rise to complex hierarchies. This has not been mentioned in the interviews for our case studies, but has been widely discussed in the literature (for instance, de Rooij et al. 2007). For example, a participant may have to report to more than one superior (somebody in their home institution and another person in the project who may be from outside their home institution), which can cause uncertainty if these superiors have diverging interests or conflicting understandings of the project’s goals and vision. This is further complicated when the participant is working with somebody from another institution, who may have to report to a different superior with yet another understanding of what the project vision is. These factors can thus inhibit the development of shared understandings amongst project participants.
4. Conclusions and Recommendations

The fostering of shared understandings within the context of software engineering practices and development lifecycles is critically important in the success of e-Science projects (Warr et al. 2007a), and the contribution that achieving shared understandings can make to collaborative projects was discussed in the Introduction. The distributed, multi-institutional, multi-disciplinary, and even international, nature of e-Science projects can cause particular challenges in the fostering of such understandings.

Since the establishment of the UK e-Science Programme in 2000, a wide range of such projects have been set up involving a variety of disciplines that spans the life sciences, the physical sciences, the social sciences, and the humanities (Hey & Trefethen 2002). This has given rise to a vibrant e-Science community, with many opportunities for project members to share their experiences of managing and working in such projects, including at conferences (such as the annual UK e-Science All-Hands Meeting13), and in project reports14.

Nevertheless, the three case studies of e-Science consortia that were conducted for this report showed that e-Science projects still encounter a number of issues that impede the development of shared understandings about a range of aspects of the project. This conclusion discusses these issues, and offers recommendations for addressing them.

Section 3.1 described some issues which result from different project members often being embedded in different reward structures. This has been found to lead to misunderstandings between developers and end-users regarding the role played by the requirements of the latter in the software development lifecycle, and whether the primary goal of an e-Science project is to produce novel technological systems or to produce tools that support the work of domain scientists. Issues of reward structures have also been found to impact upon the understandings of some developers regarding the purpose of their own work and how it fits in with the work of their colleagues: this arose particularly in the case of early-career researchers (such as doctoral students and postgraduate researchers), who might be oriented primarily towards establishing their careers in their fields (for instance, through producing a doctoral thesis, journal articles and conference papers) rather than towards the goals of the project.

There are a number of ways in which these particular issues might be addressed. For instance, projects should change the incentive structures for developers in order to direct their work towards the project’s goals. They might also provide space within the project for developers to work on their own research in order to give them an opportunity to pursue their own interests without compromising the project’s focus on producing highly-usable tools. Finally, they should take measures which will increase individuals’ confidence that working towards their project’s goals will also help to further their own interests. It is with these considerations in mind that these first three recommendations have been made:

1) **Employing new members a probationary period.** This provides a strong incentive for directing work towards the project’s goals because if someone is unable to demonstrate how their work contributes to the project, their contract may not be renewed;

13 www.ahm.org.uk.
14 For a list of many such reports, see www.nesc.ac.uk/technical_papers/uk.html.
2) **Research days**\(^\text{15}\). These are days dedicated to allowing researchers to work on developing their research in a form that it can be presented to others in their field rather than focussing on working towards the project’s goals (Warr et al. 2007b); and

3) **Assigning to each junior researcher a more senior member of the project to act as a “Champion”**\(^\text{16}\). A Champion is a more senior member of the consortium whom the individual can approach for support and advice, and who can act as their advocate during meetings. This may help to increase a junior researcher’s confidence that their interests are being represented in project decision-making.

Section 3.2 described challenges to shared understandings which can result when end-users are in competition with each other (for instance, some end-users may be commercial rivals, whilst others may be research groups who are in competition for funding). This can result in reticence on their part in communicating their requirements as they may not wish certain information to become known to competitors.

In order to address this issue, e-Science projects need to ensure the confidentiality of their methods for gathering requirements from potential end-users, and ensure that there are mechanisms in place to ensure the trust of end-users in this confidentiality. To achieve this may require a number of steps. The first is that developers and others involved in requirements gathering need to be made aware of sensitivities, for instance what particular information the end-users wish to keep private from their competitors. The next step is to provide opportunities for end-users to make their requirements known to developers without them becoming known to their competitors. Finally, relationships of trust need to be built between developers and end-users so that the latter feel more comfortable with sharing their requirements and opening-up their work practices to the former. The following two recommendations should help in this respect:

4) **Conducting one-on-one meetings with end-users**, to complement meetings that involve a range of partners\(^\text{17}\). For instance, whilst workshops may provide a forum for developers to present prototypes of tools to potential users, developers might also meet with representatives from one group of end-users at a time because they may feel more comfortable giving feedback about how the prototype meets their needs when others are not there. Additionally, an individual developer might be assigned to work closely with each group of end-users. This may help the development of trust between end-users and the developers through providing opportunities for building personal rapport (Hinds & Weisband 2003); and

5) **Involving representatives of end-users not only in requirements gathering exercises, but in the design of the requirements gathering process itself**, because they will be able to advise on how best to elicit sensitive information from end-users.

The next section discussed the challenges to shared understandings that result when e-Science projects involve designing for a wide range of end-users. There may be different groups of potential end-users who, given that they work in different contexts, may have very different requirements. This can mean that it may not be possible to accommodate all of these within the limited timeframe and resources of an e-Science project, or that designing the tools and software to meet the

\(^\text{15}\)For instance, this has already been undertaken in preDiCT.

\(^\text{16}\)For instance, this has already been undertaken in myGrid.

\(^\text{17}\)For instance, this has already been undertaken in preDiCT.
requirements of one group of end-users may render it unsuitable for the needs of others. This can impact upon shared understandings between developers and end-users regarding what the tools under development are (for instance, for whom they are being designed and the scientific work that they will support). This may result in the latter group becoming frustrated and potentially disengaged from the project. The following recommendation should help to mitigate this:

6) Developing blueprint documents, detailing future work beyond the end of the project\(^8\). This may allow end-users to remain fully engaged with the project even if they feel their needs are not currently being met because such a document can demonstrate how the work of the project might be built upon to better meet their needs in the future.

Section 3.4 described the difficulties that those involved in management or development may have in conceptualizing other project stakeholders’ work contexts. For instance, this can result in difficulties in implementing decisions if they are taken by project members who do not share the contexts in which these decisions are to be executed, or it may also mean that developers find it difficult to anticipate what other, non-collocated developers require of them. This can contribute to misunderstandings regarding plans of work regarding the tasks to be accomplished and milestones for completing work, and how the roles and work of different developers fit together. Furthermore, developers can sometimes find it difficult to conceptualize the work contexts of end-users, which can result in disagreements amongst developers regarding the features that should be incorporated into the tools under development.

The variety of models that exist for software development may cause difficulties in integrating the work of project members, and this was discussed in Section 3.5. For instance, new project members may need to become accustomed to the models being employed by the project. Furthermore, groups from institutions who are working closely together may employ different models. This issue can contribute to difficulties in establishing shared understandings amongst developers regarding how the project work is to be carried out, such as the methods to be used, the frequency of communication between non-collocated personnel, and how end-users’ requirements are to be integrated into the software development lifecycle.

Finally, Section 3.6 dealt with the impact that different partner institutions having very different pre-existing hierarchical structures can have on shared understandings. These differences may result in differences between project members in terms of what they are used to in terms of hierarchies and how authority is exercised. Furthermore, they may feel the need to respect and conform to the existing hierarchies within their home institutions in addition to the project’s structures and lines of authority, which can create frustrations.

In order to address the issues raised in Sections 3.4–3.6, a greater understanding amongst project members of the contexts in which other, non-collocated colleagues and end-users work should be encouraged. Furthermore, more opportunities for non-collocated project members to work together and interact with each other face-to-face should be provided, in order to build rapport and increase trust between them (de Rooij et al. 2007; Carusi & Reimer 2010). The following six recommendations should prove useful in these respects. For instance, some may help developers to better understand the needs of their colleagues (Section 3.4). Working with project members from other backgrounds may foster a convergence in the methods and work practices employed (Section

\(^8\) For instance, this has already been undertaken in preDiCT.
3.5). Finally, these recommendations should help to dissolve pre-existing institutional boundaries between colleagues, which should help projects to overcome the issues that arise from project members being embedded in different hierarchies (Section 3.6)\(^{19}\). These recommendations are:

7) **Personnel exchanges and visits to other sites**, to encourage personal links between project members and to foster an understanding of how hierarchies in partner institutions operate (Hinds & Weisband). Even a short visit can promote understanding of the work context at other sites, for instance regarding the technologies and facilities available and the dynamics of the relationships between the project members;

8) **Workshops** which bring together project members, sometimes including end-users, from different institutions and get them to work together on tasks (either directly relating to the project's work, or team-building exercises). This should help in a number of ways. For instance, it can make project members feel more comfortable with working directly with opposite numbers in other institutions rather than having to go through their managers to make plans and decisions;

9) **Hackathons** (Sommerville 2001). Developers from different partner institutions come together in one place to work collaboratively on software development for a period of days. This allows them to understand less familiar methods of software development they may be employed by others working on the project;

10) **Peer-programming** (Sommerville 2001). This is where developers work in teams of two and develop a piece of code together. A new member of a project might be paired with a more experienced member, and this could prove a good way for the former to get used to the model of software development employed in the project. Projects might also pair a developer at one of the partner institutions with a developer at another, in order to foster convergence in software development practices between the two institutions.

11) **Embedded users**\(^{20}\). An *embedded user* is a domain scientist who is Having an end-user collocated with the developers will help to enable the end-user to understand the constraints upon developers. This end-user is possibly more able than developers to communicate these constraints to other end-users; and

12) **Adopting a collaborative wiki approach to developing plans of work**, and allowing project members, including developers, the opportunity to contribute. This might enable those doing the day-to-day work to contribute their greater expertise regarding how the work is to be carried out, as well as being able to give realistic feedback about proposed timescales. Furthermore, it should encourage non-collocated project members to make construct plans together, without going through their respective institutional line management.

\(^{19}\) The challenge of overcoming issues relating to hierarchies is well-expressed in the following quotation from one of our interviews:

"You have a team of developers and researchers...and if they’re distributed, you get that team to think of themselves as the project team irrespective of their institutions, so they become a unit that will stand and fall themselves and they ignore their institutional line management...so people become loyal to the project and not to their institution" Interview 01/i-a

However, in light of Section 5.1, it also needs to be borne in mind that there may be a tension between promoting identification with the project rather than institutions if the reward structures of project members are bound to institutions rather than projects.

\(^{20}\) For instance, this has already been undertaken in myGrid.
Acknowledgements

We would like to thank the interviewees for the myGrid, preDiCT, and HubLad consortia for their willingness to be interviewed. Furthermore, we would also like to thank the EPSRC for funding this research (Grant No: EP/D049733/1).

Bibliography


**Further Reading on Shared Understanding**


Appendix

Schedule of interview questions:

1) Describe the project:
   a. Participating organizations;
   b. People involved; and
   c. What was the product under development, and how is it being developed?

2) Describe your involvement in the project:
   a. How did you become involved in the project?
   b. What is your role in the project?
      i. Has this role changed over time, and how?

3) What is your understanding of the overall goals of the project?
   a. How were these goals communicated to you?
   b. Has this understanding changed during the course of the project, and how?

4) What are your expectations regarding what you will personally accomplish within the project?
   a. Have these expectations changed over time, and how?
   b. How do these relate to the overall goals of the project?

5) Were there ever instances when you felt disengaged from the project in general?
   a. If yes:
      i. What happened? Why did you feel disengaged?
      ii. What action was taken to make you feel less disengaged:
         1. By yourself; and
         2. By others within the project?
      iii. Did these feelings of disengagement go away?

6) Who are your main collaborators within the project?
   a. For each one mentioned:
      i. Why are they a main collaborator?
      ii. Describe how you work with them;
      iii. How do you communicate with them?
         1. What is communicated?
         2. Frequency of communication & meetings
         3. Media used;
         4. Have there ever been any misunderstandings/conflicts/other failures resulting from a communication made?
            a. What happened?
b. How was the situation managed?
c. Was it resolved?

iv. What were your initial expectations regarding their contributions?
v. Did these expectations change over time, and why?

1. To what extent do you feel they shared your understanding of:
   a. Your own:
      i. Interests; and
      ii. Methods/way of working
   b. Your institution’s/discipline’s:
      i. Goals; and
      ii. Culture
   c. The project’s:
      i. Structure/management; and
      ii. Goals?

In all cases where understanding was felt to be missing:
   Ask for specific examples; and
   What attempts were made to achieve a shared understanding?

7) Describe instances where you were involved with making decisions regarding the project:
   a. What decisions were made?
   b. What other options were available?
   c. Who else was involved in the decision-making process?
   d. Any differences in opinion regarding the best decision to make?
      i. What were these differences?
      ii. What reasons were given by others who held different opinions?
      iii. How were these differences managed, and how was the process concluded?

8) Do you feel that the way the project played out was successful?
   a. To what extent:
      i. Was the project successful/unsuccessful?
   b. To what extent do you feel that this was due to:
      i. Your work within the project?
         1. Did you ever feel that the work you were doing had little or no impact on the project’s outcomes?
      ii. The actions of others.
   c. In hind sight, what do you think could have been done differently to improve the project’s outcomes:
      i. By yourself?
      ii. By others?
Author Bibliographies

Peter Darch is currently a DPhil student at the Oxford University Computing Laboratory, investigating how Volunteer Computing Projects (VCPs) negotiate both the technical and social challenges of engaging and retaining members of the public as participants. His research aims to formulate recommendations for future practice to improve such projects’ abilities to pursue their goals of public outreach and of producing worthwhile scientific results.

Annamaria Carusi is a Senior Research Associate at the Oxford e-Research Centre (OeRC). She studies social and philosophical aspects of computational technologies for research in science, social science and humanities. Her recent work has focused on ethics, trust and social epistemology as applied to digital data, data sharing and re-uses, computational visualisations and imaging, distributed access to medical images, and Web2.0 technologies for collaborative work. Prior to this, Annamaria conducted research on e-learning, textual studies and theory of interpretation, with specific focus on post-colonialism. She holds a Masters in Comparative Literature, and a PhD in Philosophy, and has lectured extensively in a wide range of topics: computers and society, general research skills for science and engineering students (including ethics and the responsible conducts of science), philosophy and critical theory.

Matteo Turilli is a Grid Systems Support Officer responsible for the NGS clusters at the Oxford e-Research Centre (OeRC). He recently completed his DPhil in Software Engineering under the supervision of Dr Marina Jirotka and Prof Lucas Introna. Also, he collaborates with the Centre for Ethics, Business and Economics (CEBE) at the Catholic University of Lisbon. Matteo's main interests are in parallel and distributed computing, software design, specifically as it relates to ethical requirements, formal methods and applied ethics.

Sharon Lloyd is a project manager and research facilitator at Oxford University. Sharon was project manager for the 2-year £4 million eDiaMoND project led out of Oxford in collaboration with IBM and 10 partner sites. She is now project manager for both Integrative Biology and NeuroGrid projects, which are led from Oxford, working with over 120 people over 25 sites. Before joining the University in February 2003, she worked in industry for 17 years where she gained extensive experience in project management in a commercial environment as well as people management through her management of IT teams. She has experience of both scientific and commercial IT development projects both in the UK and overseas.

Marina Jirotka is Director of the Oxford Centre for Requirements and Foundations, Lecturer in Requirements Engineering at Oxford University Computing Laboratory and Fellow of St Cross, Associate Director of the Oxford e-Research Centre (OeRC), and James Martin Research Fellow at the e-Horizons Institute. She has degrees in Social Anthropology and Psychology and Computer Science and Artificial Intelligence. She received her DPhil in Context Driven Requirements Capture from the University of Oxford Computing Laboratory in 2000. Her main areas of research have been developing novel methods and techniques for requirements capture drawn from the social sciences, focussing on workplace studies, practice driven requirements and design elicitation. Her research interests have long been concerned with developing systems that support everyday work and interaction, primarily to bring a richer comprehension of socially-organised work practice into the process of engineering technological systems. She developed the Requirements Engineering module that forms part of the MSc in Software Engineering and the MSc in Computer Science at
Oxford University. In recent years her research has focused on requirements for e-Science applications, particularly e-Health and she has worked on various industrially funded research projects, DTI, ESRC and EPSRC projects. She has been the requirements analyst on flagship e-Science projects such as eDiaMoND and is advisor on requirements methods and techniques for e-Science projects such as Integrative Biology and Virtual Research Environments for Humanities. She is a member of the UK e-Science Usability Task Force, Co-Director of the Oxford e-Social Science node that is investigating the Ethical, Legal and Institutional Dynamics of Grid Enabled e-Sciences and is Principal Investigator of the EPSRC project Embedding e-Science Applications: Designing and Managing for Usability.

Grace de la Flor is currently a DPhil student at the Oxford University Computing Laboratory studying the ways in which e-Science can change and improve the working practices of researchers and scientists. Her current research will assess how best to design and evaluate e-Science systems in support of the complex ways in which science happens; from 'in silico' experimentation, data sharing and visualization to new knowledge generation.

Ralph Schroeder is the Director of Research and a Senior Research Fellow at the Oxford Internet Institute (OII). Before coming to the OII, Ralph Schroeder was Professor in the School of Technology Management and Economics at Chalmers University in Gothenburg. He completed his PhD at the LSE in 1988. He has written extensively about virtual environments and recently completed a book manuscript 'Rethinking Science, Technology and Social Change'. At the OII he is continuing his research about how people interact in shared virtual environments. He is writing a book on this topic that will link virtual reality technology to other types of computer-mediated communication and new media.

Eric Meyer is a Research Fellow at the Oxford Internet Institute (OII). He studies the social implications of e-science and e-social science as part of the Oxford e-Social Science (OeSS) project. He brings an understanding of scientific collaboration from both sides: as a social scientist studying scientific behavior, and as a participant in the production of scientific knowledge. He completed his PhD in Information Science at Indiana University in 2007. For ten years prior to joining OII, he was based in the USA where he was the national data manager and a researcher working within a large scientific collaboration spanning twelve universities studying the genetic causes of mental health disorders.

Author Contact Information

For general enquiries please or write us following the details below:

The Embedding e-Science Applications Project
Oxford e-Research Centre
7 Keble Road
Oxford
OX1 3QG
UK

Telephone: +44 (0) 1865 610600
Fax: +44 (0) 1865 610612

E-mail: oerc-usability@mailinglist.ox.ac.uk

Website: http://www.oesc.ox.ac.uk/usability