2008-01-01

Knowledge swapshop: implementing a small but scalable knowledge management initiative in a software development team.

Glenn Higgins
Technological University Dublin, brendan.tierney@tudublin.ie

Knowledge SwapShop: Implementing a small but scalable knowledge management initiative in a software development team

Glenn Higgins

A dissertation submitted in partial fulfilment of the requirements of Dublin Institute of Technology for the degree of
M.Sc. in Computing (Knowledge Management)

March 2008

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Knowledge Management), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: _____

Date: ***31st March 2008***

1 ABSTRACT

Never before have so many organisations relied on their IT infrastructure for their survival and growth. In this environment the need for organisations to efficiently and cost effectively manage their computer systems is very clear.

Knowledge Management is increasingly being called upon to aid in the search for efficiency in the software development industry.

This project proposes the introduction of a Knowledge management initiative in a small software development team. The initiative culminates in the introduction of an automated knowledge sharing system which is intended to facilitate easier collaboration among members of the team, while at the same capturing knowledge for use in the longer term.

Before and after the implementation of the system primary research is carried out within the team firstly to establish requirements, and then post-implementation to evaluate the implementation.

Key words: *Knowledge Management, Knowledge Sharing, Wiki, Software development*

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor Brendan Tierney for all his help and advise. I would not have been able to complete this project without his assistance. I would also like to thank all the staff in the school of computing who helped me over the course of my studies.

A special thanks also to my friends and family who put up me during the course of this project. I couldn't have done it without your support.

TABLE OF CONTENTS

TABLE OF TABLES

1 INTRODUCTION

"Software Engineering is knowledge-intensive work; the main asset in software companies is what has been called the intellectual capital."
(Dingsoyr & Royrvik, 2003)

Software development projects can be very expensive endeavours, and are fraught with difficulty(Vitale, 2000). From an organisational point of view large software development projects can be a daunting and potentially dangerous prospect. The potential impact of a failed software development project can be anywhere for massive financial losses to organisational meltdown. (Mangione, 2003). One such catastrophic IT project failure lead to bankruptcy for the FoxMeyer Drug Co. in 1996. Before the failed IT project this has been a $5 billion wholesale drug distribution company. (Charette, 2005)

There is a cost to an organisation if its software developers do not effectively communicate what they know to each other. Developers in a production support role may spend time analysing a system defect, when the same analysis had been done previously within the team and a solution or workaround found, but just not effectively communicated to those who would benefit from this knowledge. Effective communication among team members is essential in order to reduce rework which is costly and doesn't add value. (Gopal et al, 2002)

This project investigates the need for the effective sharing of knowledge within a software development team, and looks at some technological solutions that can be used to assist with the sharing of knowledge across space and time within software development teams.

1.1 Background

The focus of the project is on the introduction of a knowledge sharing initiative in a software development team which is part of a large Irish financial services organisation. Increased competition in the Irish Market has lead to a squeeze on

margins for organisations in the financial services sector(O'Connor, 2008). As a result of this cost savings in all areas of the business need to be investigated.

The intention of this project is to investigate if the introduction of a knowledge sharing initiative can facilitate easier collaboration and knowledge sharing among the members of the software development team thereby contributing to faster turnaround on work carried out and ultimately to reduced costs for the organisation.

In any software development project, even in a production support context, it is vital that developers learn from each other and mistakes are not made twice, or solutions designed twice. The re-use of knowledge in software development project is of critical importance. (Petter, Mathiassen & Vaishnavi, 2007)

The project will be carried out initially as a pilot among a small team of developers involved in a focussed area of the organisations IT department, however it would be useful to allow for future expansion of the initiative should it prove to be a success so the ability to effectively scale any technological solution delivered is an important requirement for the project.

The software development team at the center of the project consists of 15 IT analysts/developers varying in skill level from inexperienced to expert. Staff are usually recruited as trainees and trained and developed within the team. This is achieved by assigning a mentor to each new trainee for their first one to two years depending on progress. In addition to this more senior staff would coach staff of intermediate technical level.

This team provides mainframe support and development for one of the organisations mainframe based applications. The core skill required by team members is COBOL although other skills are essential within the team. Developers need to be able to interact with the database, need to understand how messages are passes into and out of the mainframe and need to understand and be able to write Job Control Language. In addition to the technical IT skills required it is vital that developers understand the business and the organisations products. This can often be a challenging learning task for new staff as it is not a part of any IT course and can be difficult to master.

The team is part of an IT division which supports all the IT systems of the life assurance arm of the organisation. The mainframe system which the team supports is one link in a complex chain of IT systems which support the business needs of the life assurance business. On a working day there are approximately 1000 people dependant upon the availability of these systems from administrators in the organisations various offices to financial planners who would access the systems remotely while face to face with the customer.

1.2 Research problem

Currently the focus of the literature on knowledge management within the software development arena seems to be on the open source paradigm, especially due to the complex geographically dispersed nature of open source development. Many of the issues experienced in the open source paradigm can also be felt to a greater or lesser extend among more traditional software development teams whether they be collocated or geographically dispersed. One prime example is the need to document and record valuable expert knowledge so that it can be retained after the departure of an experienced developer. (McManus, Wilson, Snyder, 2003)

The team that is the focus of this project is a traditional development team, but due to the fact that some team members are geographically dispersed (one full time team member is based in Australia, some others work from home on an ad-hock basis) the team does experience some crossover with the issues of the open-source development community. In addition to this some of the tool adopted by the open source development community such as communication tool and Wiki type tools for example would appear to offer excellent potential to any group reliant on collaboration.

This project will investigate the requirements of the team and attempt to provide some automated way to assist this collaboration and knowledge sharing across barriers of space and time. Using modern technologies such as Wiki based systems to collaborate it is the intention of this project to provide a framework within which the team members can share their knowledge and assist each other without greatly increasing their workload. The Wiki style solution is a good fit in this case as it provides ease of

interaction and operation, which easily facilitates "*mass collaborative authoring*". (Lin et al, 2007)

Currently there is some knowledge sharing happening in the team, but there is an opportunity to leverage this newer technology to build on what already takes place in the team. This situation is a common one and was experienced by Boston law firm Bromberg & Sunstein LLP. According to their Chief Technology Officer Monroe Horn:

"As our workload and staff grew, however, our old methods of sharing knowledge – document, spreadsheets and word-of-mouth – became increasingly inadequate. We could not get quickly the information people needed to move forward on tasks. We had to find new ways to collaborate." (Horn, 2007)

This is a good parallel to the team in question here in the sense that the current processes are working but increasingly becoming under pressure to deliver in a busy modern environment.

This project focuses on a small subset of the broader area of knowledge management and another research problem was to site the project within the broader context of the field of knowledge management.

1.3 Research objectives

The following objectives have been achieved throughout the dissertation and contributed to the overall outcome:

1. Carry out a literature review in the area of knowledge sharing in organisations, and then focus in on the area of knowledge sharing in the context of software development both in traditional software development teams and open-source software development teams or groups. Following this carry out further literature review in the area of available technologies for knowledge sharing in organisations and the implementation of these technologies. Also carry out literature review in the broader area of knowledge management in order to inform and direct the more focussed research into the area of knowledge

sharing in organisation, knowledge sharing in software development teams and knowledge sharing technologies.

2. Carry out a literature review of available case studies in order to understand how people had previously gone about the deployment of automated knowledge sharing technology.
3. Through primary research find the opinions of the software development team on what might improve knowledge sharing. This process feeds into the selection of technologies that would help to facilitate collaboration. Gather requirements also by reference to the literature, attempting to include the best aspects of similar implementations and avoid issues experienced in them.
4. Review the available and suitable candidate technologies which can feed into the process of selecting the best fit solution.
5. Selected a candidate solution and arrange for this to be installed and tested. Once in place it will be pre-populated and presented to staff in the development team. They will be asked to use it if possible and appropriate.
6. Carry out a further round of primary research after the implementation phase to assess whether the implementation was a success. The opinions of team members will inform this research but will also feed into the recommendations for future work.

1.4 Research methodology

Both primary and secondary research was carried out to achieve the research objectives. The primary research involved structured interviews of the members of the software development team at various stages of the project. In addition the collection of statistical information about the level of use of the technology was recorded and analysed in conjunction with the outcome of the structured interviews.

The secondary research that was used to achieve the research objectives was the review of relevant literature. Initially the broader area of knowledge management was the focus followed by the area of knowledge transfer in organisations especially in the context of software development. Finally the focus narrowed further to the area of

technology to assist and facilitate the transfer of knowledge especially in the area of Wiki style solutions.

The richest sources of research material were conference proceeding, papers and journals. In addition some books were useful and websites of various organisations especially the websites of vendors of knowledge management technologies.

1.5 Resources

One key resource which was required for the completion of this project was the co-operation of the management and staff of the software development team. From the outset the management indicated that they would support the project, and without this support the project would not have been possible.

Another fundamental resource was the library facility's both on-site in DIT and also those facilities available over the internet in particular the electronic journals (e.g. IEEE, Emerald Journals).

Access to personal computer facilities on-site in the organisations premises, on-site in DIT and at home were essential. The availability of email access and internet access at all these points was also a crucial resource.

The support of the project supervisor was also essential resource and the project would not have been possible without the regular meetings and assistance provided in person or by email.

1.6 Scope and limitations

This project will carry out primary research in a software development team, and using the results of that research as well as the outcome of a literature review, the project will attempt to deliver an automated knowledge sharing system to the development team. The system will not be rolled out to any other teams within the scope of this project although the ability to do that at some future stage without rework is part of the project deliverables, hence the requirement for the system to be scalable.

The project will then monitor the system in operation and collect further primary research among the user group to evaluate the performance of the system.

The goal of this project is to improve the knowledge transfer processes of a specific software development team. Results obtained from research within this specific team may not be applicable to other similar teams or to the software development industry as a whole.

The time and resources available to the project limit the work that can be done in the implementation, and limit the work that can be done post implementation. Any useful actions identified to improve the knowledge sharing/collaboration system post-implementation may need to be included as future work and addressed as such either by the author or by the organisation itself.

1.7 Organisation of the dissertation

The dissertation is organised in chapters with each chapter being broken down into section and subsections as appropriate. The current chapter is intended to provide a brief overview of the project, introducing the concepts and areas to be discussed in greater detail later.

Chapter two is based on the outcomes of the literature review and starts with some discussion of the area of knowledge management as a whole. Section 2 of this chapter focuses in on knowledge transfer in the context of software development, and discussed various references which are useful to the proposed implementation in this project. In section 3 some case studies are examined which highlight knowledge initiatives undertaken in real world situations. These are very useful as they help to highlight potential pitfalls and how & where certain technologies have been a success in the past.

Chapter three discussed in more detail how the project will be brought to completion. In this chapter the steps that are required to achieve the goal are discussed individually in detail.

Chapter four will outline the requirements for any technological solution that could be used to help improve knowledge sharing among members of a software development team. We then consider the available technologies that could be implemented. This section will present each of the options which were short-listed, and analyse their strengths and weaknesses against the requirements. The final section of the chapter will identify the best fit solution and outline the reasons why it was chosen.

Chapter five looks at how this system is actually implemented in the development team, looking firstly at the research that was necessary to judge the cultural readiness of the organisation and then moving on to look at the technical tasks of physically installing the system. This chapter will then look at the less technical tasks that needed to be carried out in order to deliver this system to the development team such as encouraging people both formally and informally to use the system and demonstrating potentially useful aspects of the system to the user group.

Chapter six will attempt to evaluate the success of the knowledge sharing system that was discussed in chapter 5, and attempt to evaluate the success of this project as a whole. This chapter looks at a number of sources of primary research such as a survey of staff and management as well as usage information collected during the system pilot.

Finally chapter seven looks at the project as a whole, revisiting the research objectives and research problem before outlining the addition that this project has made to the body of knowledge. It finishes with a discussion of future work which is indicated by this project.

2 LITERATURE REVIEW

2.1 Introduction

This section contains a discussion of the literature that was reviewed at the outset of the project. Initially the broader area of knowledge management was reviewed before the focus was narrowed to areas more directly related to the project at hand. This can be seen in section 2.2 of this chapter.

Following on from the field of knowledge management the specific area of knowledge sharing was examined, especially in the context of knowledge capture and transfer for software development teams. This review lead on to the review of specific technologies that can be used to facilitate knowledge sharing and collaboration amongst software developers. This can be seen in section 2.3 of this chapter.

The final part of the literature review involved investigating case studies of actual knowledge management initiatives in an attempt to isolate positive learnings and potential difficulties in their implementation.

2.2 Knowledge Management

Although knowledge management in industry and academia has in recent years become very popular as a discipline, humankind has shared knowledge from their earliest days. Over the course of history much of the knowledge shared amongst humans would have been tacit knowledge, that being knowledge which is passed on from person to person, usually face to face in some social context like master to apprentice or parent or child. This type of knowledge is distinct from explicit knowledge which is held outside of the person for example in written form or electronically in a computer system.

This distinction between explicit and tacit knowledge is examined by Nonaka and Takeuchi (1995). We are told that western philosophy struggled between the two schools of empiricism and rationalism. Empiricism being the viewpoint that what we

know is based on what we experience as contrasted with rationalism which would state state that what we know is based more on some inherent truth.

By examining the work of Bateson as well as Argyris and Schon Nonaka and Takeuchi proposed a representation of knowledge creation as a "dynamic spiral" (Nonaka & Takeuchi, 1995). From this representation Nonaka and Takeuchi go on to develop their model for knowledge creation and transformation which is shown below:

Tacit to Tacit *Socialisation*: Sharing knowledge and experience between people. e.g. on the job training	**Tacit to Explicit** *Externalisation*: Express or articulate tacit knowledge explicitly e.g. Written documents.
Explicit to Tacit *Internalisation*: To absorb the knowledge held in some explicit means e.g. Learning by using a manual.	**Explicit to Explicit** *Combination: Sorting or combining knowledge e.g. Formal education.*

TABLE 2.1 – REPRESENTATION OF NONAKA'S KNOWLEDGE SPIRAL.

To efficiently manage knowledge an organisation must facilitate knowledge sharing which allows for tacit knowledge to be externalised, but also facilitates the internalisation of this by its members and absorption of knowledge by socialisation. (Nonaka & Takeuchi, 1995)

In the organisation under review in this project there is a very high level of skill and experience embedded in tacit knowledge. It would be fair to say that a high percent of what the organisation knows leaves at the end of each day in the minds of its staff. There are good mechanisms in place within the IT department for tacit to tacit knowledge transfer, with new staff receiving good on the job training and good support in general from their colleagues and supervisors. This tacit to tacit knowledge sharing is *"more likely to be internalized by the receiver than, for example, person-to-document-to-person knowledge transfer"(Endres et al, 2007)*. Having said that it would still be desirable to have this tacit knowledge recorded so that there is some chance of internalisation in the absence of the expert.

Another important body of work in the area of knowledge management is the work done by Thomas Davenport and Laurence Prusak. Unlike Nonaka this work had its foundations in the United States of America, and so sprang from a very different cultural environment. This difference in national culture undoubtedly had an effect of the research and conclusions that the author arrived at.

Three knowledge management processes are outlined which we are told are key to success: generation, codification, and transfer of knowledge. (Davenport & Prusak, 1998)

Knowledge Generation we are told can happen in 5 modes: acquisition; dedicating resources; fusion; adaptation; and building knowledge networks. Knowledge generation modes can harness the knowledge already in the organisation such as dedicating resources or building knowledge networks. Knowledge can also be brought into the organisation from outside for example by hiring individuals or by acquiring or merging with another organisation. (Davenport & Prusak, 1998)

In terms of knowledge codification Davenport and Prusak outline four principles which can be used to facilitate the process:

1. Managers must decide what business goals the codified knowledge will serve.
2. Managers must be able to identify knowledge existing in various forms appropriate to reaching these goals.
3. Knowledge managers must evaluate knowledge for usefulness and appropriateness for codification.
4. Codifiers must identify an appropriate medium for codification and distribution.

(Davenport & Prusak, 1998)

Finally knowledge transfer is identified by Davenport and Prusak as one of the most challenging aspects of any knowledge management programme. They identify the cultural aspects of knowledge transfer as the most likely source of difficulties in any programme, and outline a number of barriers to knowledge transfer which may be found within an organisations culture.(Davenport & Prusak, 1998)

2.3 Knowledge Transfer in a software development context

Effective knowledge transfer among team members is vital to enable the team as a whole to benefit from lessons learned by individuals. The team cannot afford the same mistake to be made repeatedly by team members due to non communication of the pitfalls of various operations.(Voit & Drury, 2006) Implementing a "validated and accessible organisational repository" which is underpinned by learning from previous experience can help an organisation to improve its "project planning, implementation and control". (Wei, Hu, Chen, 2002)

The sharing of knowledge among members of a software development team is essential in order for them to efficiently achieve their goals. According to Chau, Maurer and Melnik (2003), there is a need for knowledge sharing to support:

1. Effectively sharing of domain expertise between the customer and the development team.
2. Identifying the requirements of the software system.
3. Capture non-externalised knowledge of the development team members
4. Bringing together knowledge from distributed individuals to form a repository of organisational knowledge.
5. Retention of knowledge that would otherwise be lost due to the loss of experienced staff.
6. Improvement of organisational knowledge dissemination.

Due to the nature of the work done by the software development team in which this project is sited, and following on from discussions with the management group of the team the more important points above that would be relevant to this project are points 3, 4 and 5.

The third point about the capture of non-externalised knowledge is important as the team hold a lot of specialised knowledge and often it is hard to know who knows what, or sometimes it is even hard for people to be aware of the full extent of what they themselves know. To provide help and support with the capture of this type of knowledge is important for this project.

The fourth point is important to this team, as not all team members are based on the same site, and not all team members work the same hours. There is one expert employee based off-shore, and in addition to this the organisation operates a system of flexible working hours which may mean that certain knowledge is not available to the team at certain times. To facilitate easier knowledge sharing across these barriers of space and time would be of great advantage to the team.

Finally point 5 about the retention of knowledge is an important one as the developers on the team are experts in their areas and if a developer leaves the team, this knowledge is lost so there is a need for some means of aiding the capture and further sharing of knowledge within the team. This point feeds into the requirements for the system which can be seen in section 4.2 of this dissertation.

The capture of non-externalised or tacit knowledge would be an essential part of this initiative. According to Stenmark, "*Tacit knowledge constitutes the major part of the body of knowledge and it is therefore important for organisations to sustain and exploit this asset*". (Stenmark, 2000)

A good example of where this would be directly applicable in the team would be where a developer makes a change to a programme or job in order to fix a defect with the system. At the time the developer will have done analysis of the problem, and will have some idea of the impact of the change. This information should ideally be recorded at the time when it is fresh in the developer's mind, as opposed to them struggling to remember when asked by a colleague months or years later.

Ideally these nuggets of knowledge, as well as other lessons learned and other more formally gathered knowledge would be brought together in some type of knowledge repository. This repository would help to overcome the temporal or geographical problems of accessing knowledge that other developers may experience when seeking knowledge.

In the context of this project though the most crucial point from those outlined above is I believe the ability to retain knowledge in the team after the loss of experienced staff.

In the team under research the rate of staff attrition among senior staff has been slow up until 2 years ago but since then due to various management opportunities and group wide expansion, this rate of attrition has increased. This further underlines the need for some method of storing knowledge long term and making it accessible to people who need it. According to Pickett (2004), *"the recording and retention of the knowledge and experience of people who leave is increasing in importance to organisations around the world"*.

Access to knowledge too is important, with knowledge often being available somewhere within the team but not accessible to those who need it. A good example of this lack of access is where a team member has stored knowledge in their email folder. If another team member knows this and can ask them for a copy this knowledge is easily shared but otherwise this knowledge is effectively wasted as other team members may end up chasing down this knowledge all over again, with the consequent cost to the organisation. This problem is acknowledged throughout the industry and is sighted by Cristal and Reis as an example of how incorrect media can lead to loss of expertise:

"One example is e-mail usage. In certain projects team members exchange valuable information for their organization through e-mails. A lot of important information can be just forgotten in some team member's inbox instead of being saved in a common entity to be shared across the entire organization."

(Cristal & Reis, 2006)

Of course knowledge sharing is not only influenced by the technological tools or facilities in place to facilitate it. The culture of the organisation will also have a massive impact on knowledge sharing within that organisation and the potential success or failure of any knowledge sharing initiative that may be undertaken.

Some organisations may have a very innovative culture which facilitates high levels of knowledge sharing between individuals, however an organisations culture can inhibit knowledge sharing initiatives by encouraging individuals to *"resist searching out or receiving knowledge or to resist efforts to move knowledge out of their heads"*. (Ladd & Heminger 2002).

It would be possible to think of knowledge sharing as a spectrum with organisations placed along this spectrum depending on various factors, such as for example culture, size/scale or industry. On the one side we could consider the notion of the knowledge market as outlined by Davenport and Prusak. In this conceptual framework knowledge sharing is thought of as a transaction which occurs between buyers and sellers in the knowledge market and can be facilitated by brokers. This view would appear to indicate that people take part in these exchanges to achieve some personal advantage, as with any market analogy.(Davenport & Prusak, 1999)

The contrasting perspective of how knowledge sharing might occur in an organisation is that offered by Nonaka and Takeuchi, where knowledge is shared via the knowledge community. In contrast to the knowledge market this view tends to focus more on the personal and social aspects of knowledge sharing. Instead of being motivated entirely by self interest, In the knowledge community a person may participate in knowledge sharing to achieve a shared goal or indeed out of concern for each other. (Nonaka & Takeuchi, 1995)

Where an organisation stands in the knowledge sharing V knowledge market is subjective and different people in the organisation might feel the point could be differently placed for their organisation. This only serves to highlight the cultural issues that are ever present in knowledge management programmes, and which must be dealt with in order to achieve success in any knowledge management initiative. In fact culture is a vital aspect to any knowledge management programme, and is one of the most commonly mentioned obstacles to knowledge management programmes(Hall & Goody, 2007).

The sharing of knowledge in an organisation is very different to information sharing as knowledge is *"enriched with insights and is socially embedded in organisational practice"*(Chua, 2003). It is clear then that the culture of the organisation under the influence of which the socially embedded practises occur will have a serious impact on an organisations level of success at sharing knowledge.

To gain a deeper understanding of an organisation we need to look at how that organisation is structured. One useful way to view an organisation is as a series of

interconnected and co-operating communities of practice. The level of this connectedness and co-operation could indeed influence the overall efficiency of the organisation. According to Cristal and Reis(2006), an effective organisation can be seen as a *"constellation of interconnected communities of practice"*. These communities of practice are vital to knowledge organisations, and can be a very effective facilitator of socialisation of tacit knowledge within an organisation. Unlike purely functional teams communities of practice emphasise what and how people learn from each other rather than who they report to. (Cristal & Reis, 2006)

This concept could prove very useful to the research at hand, as the pilot implementation of the sharing and collaboration system will be rolled out in a development team which could be described as a community of practice. Indeed the IT department in which this team is situated could well be described as series of communities of practice which operate independently but work together to achieve their own and the departments objectives, and ultimately to play their part in the organizations' objectives as a whole.

Within these communities of practice knowledge creation, accumulation and sharing is essential. According to Yang, Chen and Su (2007) it can be seen that these communities of practise use knowledge sharing to help all members achieve their goals, in fact this knowledge sharing is they say the main reason people become involved in these communities of practise. Knowledge sharing requires *"intensive interaction between participants"*(Yang, Chen & Su, 2007). In the development team where this project is being carried out there is already considerable interaction between developers, however there may be an opportunity to collect articulated knowledge and make dissemination and re-use of such knowledge more efficient and easier for team members.

Communities of practise would traditionally have used socialisation and documentation to share their knowledge but there are now many technologies available which are specifically designed to aid knowledge sharing and collaboration among people and groups or indeed across organisations.

One such technology which is widely used and has great potential for expansion is the use of Wiki software within an organisation. The use of a Wiki within an organisation can allow people who wouldn't normally have had the opportunity to publish through normal means to find a voice within the organisation. (Charles & Ranmi, 2007)
The use of Wiki systems In organisations to facilitate collaboration offers great potential benefits. This technology is *"rich in potential for communication, information sharing, and collaborative endeavors"*. (Ramos & Piper, 2006)

Wiki systems are often though of as an "*evolving shared knowledge repository*", but there usefulness may be underestimated by some observers. Currently there exists the potential for Wiki technology to be *"a tool to support a contemporary, yet challenging view of corporate KM that is participatory, holistic, collective and contextual."* (Hasan et al, 2007)

2.4 Case studies

Perhaps the best way to understand the implementation of knowledge sharing and collaboration technology in industry is by reference to case studies. In this section 3 such case studies will be analysed, including the Variation-reduction advisor employed by general motors, the Wiki system which replaced the intranet at Jannssen-Cilag Australia and the roll out of a Wiki system at Dresdner Kleinwort Wasserstein.

2.4.1 The general motors variation reduction advisor

This case study (Morgan et al, 2002) is an excellent example of a successful knowledge management initiative which gained the support of its users and actually delivered practical and real benefits to the organisation.

The general motors variation-reduction advisor was initially developed as a case based reasoner to help with knowledge capture and transfer, especially in the area of lessons learned, within the dimensional management team. This team investigates cases where the basic vehicle frame is found to be variant from specification. This variation can be very minor but can lead to issues later in the vehicle assembly process.

Initially the variation reduction advisor captured knowledge about faulty vehicle frames as cases in a case based reasoning system, which were structured with templates and "domain specific ontology's". As the project evolved the use of templates was relaxed, and users could enter lessons learned along side messages, and other un-validated input. This lead to a trade off between control and usefulness, but ultimately users found this beneficial.

From its initial existence in one team in their flint manufacturing plant the variation advisor began to roll-out to other manufacturing facilities. Perhaps uniquely in this case however this roll-out was actually requested by other facilities who had heard about the system and could see a place for it within their facility. Those who used the system from the outset were proud of it as they had a major influence on its development and direction.

There are many useful lessons from this case study which can influence the implementation of any knowledge sharing and collaboration system but for the purposes of this project, one of most interesting aspects of the case study would have to be the grass roots approach taken by the organisation in their implementation. This has some very important influence on their ultimate success for example low initial cost meant higher level management would not be reluctant to allow the project to happen, and aiming the system at the grass roots level meant the developers could canvass opinion amongst the actual end users and actually include this practical feedback into the system as it was developed.

This grass roots approach also had an effect on scale. The system which evolved from this project was a useful tool, fit for purpose which provided benefit to its users. The proposal for this project is to implement a small scale system, in order to deliver something which actually adds value in the team in which it is to be used. For this project scalability will be a key part of the design from the outset, meaning that a small scale practical application can be delivered which lends itself to expansion without the need for re-work.

Another very useful point is the evolutionary nature of the system. It can be seen from the case study that the organisation allowed the system to evolve in a way that made it

most effective, and this brought the system in a direction which could not have been envisaged at the outset (i.e. move from templated content to free form content). This free evolution contributed to the creation of a system that was actually beneficial to the users and helped to contribute to the high level of user buy-in which is essential to the introduction of any new knowledge system.
(Morgan et al, 2002)

2.4.2 Case Study of Wiki Adoption at Janssen-Cilag Australia

Janssen-Cilag is an Australian company with more than 300 employees and is one of the 250 Johnson and Johnson operating companies(Wallace, 2007). In 2006 they replaced their static HTML based Intranet site with a Wiki system, based on the Confluence product from Atlassian.

Their implementation would appear to have been a great success according to the case study, and there are a number of interesting lessons that can be taken from this case study which could help with my own knowledge sharing system implementation.

They didn't necessarily broadcast that the system being rolled out the users was a Wiki. It was just an updated Intranet site. The name used was JCintra which again doesn't draw attention to the underlying Wiki. Often people could have a preconceived notion of a Wiki due to references to Wiki in popular culture and the widespread reporting of Wikipedia misuse in the media. According to Wallace users of the JCintra site *"don't view it as a Wiki with all the anarchistic overtones that brings. Rather they see the simplicity and flexibility as a natural evolution of Intranet technology"*(Wallace, 2007)

There are similarities also between the business requirements which were gathered prior to the JCintra implementation and the knowledge sharing system requirements for this project. One of the key requirements for the JCintra system was simplicity, *"whatever we do has to be* simple" (Wallace, 2007). This too was a requirement within the project at hand and would seem an intuitive part of facilitating collaboration between individuals. A further requirement was that content could be trusted. This seems a strange requirement at first in the context of Wiki based implementation

however many Wiki products now come with various security features as standard such as LDAP integration and user login for content addition and update.

Another interesting point raised is that Wiki systems are by no means a perfect fit for all situations, but in this case it was felt that the Wiki model was fit for purpose, and provided what the organisation needed at that point. Naturally some people raised concerns about the security and reliability of content which could be added or updated by anyone, but this was addressed in the following 3 ways:

1. The use of "social force" as opposed to "technical force" to ensure that people created and managed content appropriately. If people needed to login to add or update content, then the wider group would see that content and there would be a certain group judgement of misbehaviour.
2. Anyone could monitor any content they were concerned about, in the case of the system used here the method expressed was automatic email alert with changes however for this project this would be handled by RSS feeds showing updated content. Once concerned individuals are aware of changes they can make further changes or roll-back the content very quickly and easily
3. In some cases it was essential that content was rarely or never changed, and this could be achieved by "locking down critical content" and setting out clearly who owns the content and who is responsible for its maintenance.

The concept of ownership is also explored, and is an important area to deal with in any collaborative system such as a Wiki where users can add and update content themselves. One interesting point raised in the case study is that if someone is not willing to maintain content the its obviously not that important to the business. If content is important to someone or a group of people they will "watch" for updates and update it if they come across it in searches and notice it is out of date. In this case study with ownership of the content pushed out to the relevant departments the *"quality of content in an area is a reflection of that owner and not of the intranet itself"*. (Wallace, 2007)

Currently in the IT department under investigation documents are stored on shared network drives and documentation is very difficult to search and re-use. If documents get forgotten about then the value that they could potentially add is lost.

(Wallace, 2007)

2.4.3 Dresdner Kleinwort Wasserstein (DrKW) Case Study

This case study (Socialtext.com, 2008)is based on the implementation of a Wiki system at this investment bank with its headquarters in London and Frankfurt but with offices in New York, Paris, Luxembourg, Tokyo, Singapore and Hong Kong.

Prior to the initiative in 2004 there had been a Wiki in place in the organisation but only in the IT function, this initiative was designed to roll out Wiki use across the entire organisation. The organisation chose to use the socialtext Wiki System.

An interesting point about their roll-out of the Wiki was that they pre-populated the Wiki with useful and interesting content. This meant that users had something that they could search immediately, as opposed to gradually building up content in the system. This would appear to be a sensible precursor to any Wiki launch if possible.

As adoption of the Wiki grew the organisation brought in a social software consultant to help manage support and adoption. For the purposes of this project due to the scale involved it is unlikely any outside assistance would be required for the pilot roll-out however it is interesting to see the strategy that the consultant employed.

The consultant carried out *"informal training sessions to introduce new users to the Wiki, explain best practice and teach Wiki mark-up. These voluntary sessions lasted less than an hour and showed users how to navigate the Wiki, create and edit pages, upload files, and change settings."*(socialtext.com, 2008). There is a parallel here with the GM case study where the knowledge management initiative was started as grass roots initiative. Clearly the best way to proceed with this project if to roll-out the system on a small scale at first, and to help people get involved a short demonstration sessions to give people the basics would appear the best approach. This is facilitated by the intuitive nature of the Wiki systems and the ubiquitous nature of the internet in modern Irish society.

Some of the areas that are mentioned as being popular uses of the Wiki are:

i. Managing Meetings

   People use the Wiki to compile and distribute agendas and meeting minutes. Use of the Wiki helps to reduce the amount of emails required to manage meetings and to reduce the amount of critical information that is sitting in peoples in-box unavailable to others.

ii. Brainstorming & Publishing

   The Wiki has been useful in facilitating the generation of ideas, whereby people would add lots of random ideas to a page and over time this page would evolve into a firm document filtering out unworkable ideas and arriving at ideas that perhaps would not have been thought of.

iii. Creating Presentations

   Creating presentations in the Wiki as opposed to Powerpoint allowed users to concentrate on the content rather than the look of slides etc. which meant they could produce effective presentations quicker.

Sometimes the cost savings delivered by a collaboration system can be difficult to measure, but in this case the organisation felt there was a real cost saving delivered by the Wiki. According to the report:

"At DrKW, Wiki users have seen demonstrable value. Their meetings run more smoothly and are more productive; unnecessary barriers between teams are being broken down; the quality of product specifications and documentation is improving; presentations are being written faster and more effectively; and the risks posed by staff leaving is reduced.".

For this project the ability to retain knowledge after someone has left the development team is a important factor.

Finally a very interesting point that is raised in the case study in the area of regulation. According to the author *"Investment banking is a highly regulated and conservative industry,yet DrKW have built an open, collaborative, live intranet that is robust, up-to-date, and continuing to expand."*.
(Socialtext.com, 2008).

2.5 Conclusion

Knowledge Management can deliver benefits to modern organisations in terms of improved efficiency, reduced cycle times and better re-use of knowledge/less rework. In the software development context collaboration tools can also provide efficiency's. The Wiki tool is one which facilitates fast and simple collaboration between people and groups, and its use as a tool for collaboration appears to have been successful for organisations in the past.

3 PLAN TO DELIVER PROJECT

3.1 Introduction

This section will further investigate the motivation behind the project. It will explore the need for improved knowledge management in the organisation in question as well as in general. It will also discuss the benefits that can accrue from more efficient management of the knowledge assets of the organisation.

In the sections that follow the tasks that are required to complete the project as examined, and a brief description is included with each of what is required to complete these tasks.

3.2 Why Manage Knowledge

According to Sean Dorgan of the IDA: *"Ireland can now convincingly claim to be a knowledge economy"* (Dorgan, 2003). This is exemplified by an increase in foreign direct investment and in increase in Research & Development spending by both home grown and multinational organisations. The Irish economy is changing moving away from a more traditional manufacturing and agricultural based economy and towards a knowledge economy where the provision of services to the domestic market and abroad forms a large part of our GDP.

In this context it is essential that organisations efficiently manage their knowledge assets in order to remain viable and to grow. According to Wikramasinghe et al (2004):
"It can be argued that knowledge management, in addition to making informed decisions, will result in an improvement due to a uniform procedure, and increased learning capacity, a reduction in costs due to greater efficiency and improved communication".

According to Grutter et al (1999) Organisations are *"discovering knowledge as their critical asset to successfully compete in emerging global markets"*, organisations that

are involved in the knowledge economy are finding that they are *"subject to increasing returns, whereas conventional energy-based industries exhibit diminishing returns"*.

In light of these changes, it is essential that any organisation that wants to grow and prosper must be conscious of their knowledge management needs and put in place plans to effectively manage their knowledge assets.

3.3 Outline of project to completion

There are many aspects to this project, theoretical and practical. To successfully complete the project it was necessary to manage several aspects simultaneously. The literature review section was carried out initially to help drive and inform the remaining tasks. A part of the project was to investigate and implement a system which would help to facilitate knowledge transfer within the team. The steps which were followed to implement this technological solution are discussed further in section 3.4 of this document.

To successfully implement this system and change the knowledge sharing patterns of the team it was also necessary to carry out some less technical tasks. The first of these was meeting with people and asking for their inputs, what they wanted from a knowledge sharing system. The output from this process fed into the requirements section of the project. It was not possible to survey all staff in the time available but I did meet with a representative sample of staff and management.

Just before the roll-out of the system to the team, it was necessary to carry out some staff training. This was not particularly involved as the system itself is quiet intuitive and the staff in this team are of a good or higher technical level of ability. The first step was to carry out a presentation to staff on what was being implemented, and how it should be used. At this point it was a key point to emphasise the benefits to the team of this initiative. Without the buy-in of the team members this initiative would be doomed to fail from the outset.

Following the roll-out to the team it was necessary for the author to be an "evangelist" for the system within the team, and lead by example so using the system myself and communicating with people through the system were ways I could highlight its potential uses to the members of the team.

3.4 Steps involved in practical implementation

As part of this initiative an automated tool was implemented to facilitate easier knowledge sharing, and collaboration. In this section I will discuss each of the stages that was necessary to bring about this implementation successfully.

We can break the steps required to carry out this implementation into 4 phases:

1. Assess the cultural readiness of the organisation, and gather together the requirements for the system.
2. Install the solution that was selected, and carry out some basic testing to ensure it is operational.
3. Roll out the system to the developers in the team. Monitor the use of the system, and champion its use within the team. During this phase also gather data and feedback which will help to decide whether the implementation was a success.
4. Bring together all the output from the previous phase along with further primary research output and review the use and usefulness of the system to conclude how the implementation went. Decide if any future work would be of use in light of the findings.

3.4.1 Cultural Readiness

Before it was possible to look at implementing a solution it was necessary to asses the culture of the team in which the project was taking place. For the purposes of this project this investigation was restricted to the team directly impacted by the initiative, however for a project with broader scope, it would be beneficial to investigate the organisations culture as a whole and see how the IT department and this development team sits within that culture. Such deep analysis is however beyond the scope of this project.

The culture of the team was assessed by primary research, by structured questionnaire designed to find out whether the proposed knowledge sharing system would work in the team, given the existing culture.

It was necessary to look at the culture from two sides. Firstly from the management perspective, and then also from the developers perspective. The team manager, and a senior manager responsible for the team were surveyed, as well as a representative sample from the development team.

Some of the areas of interest from a management point of view were:

- Did the management feel there was a need for change?
- Did they "believe" in using new technology to achieve more efficient knowledge sharing and collaboration?
- Were they prepared to support the change initiative?
- Are they comfortable with the centralisation of knowledge in a manner accessible to others?

Some areas of interest from the point of view of the developers were:

- Did they feel there was a need to improve on the current process?
- Is there a good spirit of teamwork and collaboration in the team?
- Are the team members technically competent with modern social computing applications e.g. Wiki interaction.
- Would they be prepared to share their knowledge in a way that would permanently capture it?

The outcome of this research is discussed further in chapter 5 and chapter 6 of this document, but broadly speaking the outcome was that there was an opportunity to improve efficiency through more efficient management of knowledge, and an automated system to facilitate knowledge sharing would be a welcome initiative.

The details of the survey are discussed briefly below but for a full discussion of the questionnaire see chapter 5 of this dissertation. To see the full text of all questions and answers see Appendix 1.1.

There were 6 questions asked in this survey and 7 respondents took part made up of 5 developers of varying levels of experience/expertise and two management level respondents who ultimately have responsibility for the team.

The questions were designed to find out what people thought the current state of play was in the team, in terms of knowledge sharing and knowledge management. The questions were structured to begin by asking people about how well we do things at present. This meant that we could embrace what worked well and ensure it was not overlooked as any solution was designed.

The questions then turned to what we don't do well in the team. The reason for this is that the best people to ask about a particular system or process are often those who are working with it every day.

Finally the questions sought to find out if people felt there was a need for change and if they would support change if it happened. Often peoples or cultural reluctance to change can destabilise a change initiative so it was important to find out peoples opinions from the outset. If at that point there had been a major concern it could have informed the rest of the project or been dealt with at that point.

The next piece of work that was carried out was to analyse and decide upon the requirements for the system. This was done in the first instance by reference to the literature review, attempting to avoid problems experienced by previous implementations and learn from their lessons. The requirements are discussed further in chapter 4 of this document particularly in section 4.2. They also refer back to the literature review section in particular to section 2.4 which focussed on case studies of previously implemented solutions.

Once requirements were established the next step was to select a candidate solution. This is dealt with in detail in chapter 4 of this document in section 4.3. The steps required then are listed and briefly described below. These steps are expanded in the implementation section of this document in chapter 5.

3.4.2 Install and test software

The software necessary had already been downloaded to facilitate the product comparison so the next step was to install it on a networked machine that the team could access. The initial round of testing was simple just to ensure that the Wiki could be connected to from network locations as well as the 'localhost'. The table below shows the tests that were carried out at this point and the results.

	Add User	**Login**	**Add Page**	**Retrieve Page**	**Add Attachment**	**Retrieve Attachment**	**Perform Search**
'localhost'	Ok	Ok	Ok	Ok	Ok	Ok	Ok
Remote 1	Ok	Ok	Ok	Ok	Ok	Ok	Ok
Remote 2	Ok	Ok	Ok	Ok	Ok	Ok	Ok

TABLE 3.1 – DESCRIPTION OF INITIAL WIKI SYSTEM TESTING

This testing although simple was essential as the users confidence would have been undermined if there was a major issue with the system right from the outset. To get user buying this project needed to retain peoples confidence and needed to show people just how effective this system could be.

Once the software was up and running it was necessary to pre-populate the system with some useful information. This involved deciding upon what was relevant and useful knowledge and moving it over to the Wiki so that the users could access it when ready. This step was carried out based on the lessons learned from the case study of Dresdner Kleinwort Wasserstein namely that "*You need to add interesting content to the wiki to engage people and show them how it is relevant to their team. Then they'll start experimenting - editing and creating pages. If you give them a blank wiki it will never take off because they will never find the time to learn how to use it if they don't see the value.*" (Socialtext.com, 2008).

One other task at this point was to add company logo and team name details to the Wiki screens. This was not a key deliverable but a certain level of visual completeness was necessary.

3.4.3 Roll-out to developers with demo.

A presentation and FAQ was prepared to circulate to the development team and the author spend a few minutes with each developer running through the operation of the system, and answering any questions they had. This was a luxury I could afford due to the scale of the implementation.

Following the roll out of the system it was necessary to monitor peoples usage of the system and champion its use. This was done in two ways, firstly monitoring the pages added to the Wiki, and the logs to see how it was being used. Secondly it involved speaking to people informally and finding out how they were finding the system and if there were any obvious issues that were creating barriers to its effectiveness.

In addition to the informal discussions with team members It was possible to champion usage of the system by using it myself and attempting to communicate with people through the system as often as was possible, and relevant. In so doing I was attempting to show people, without forcing them, that the Wiki system would facilitate collaboration among team member without a huge amount of work being needed to get up and running.

During this phase as well as carrying out informal discussions some results were gathered to feed into the next phase. This involved monitoring how many pages were currently on the Wiki, how often people were logging into the system, and how many searches were being carried out.

At this point also it was necessary to carry out some further primary research. This consisted of two methods of data collection. Firstly creating and circulation a structured questionnaire to all the users, and following up on any interesting outcomes from this process. See appendix 1.2 for questionnaire. The results of this survey are discussed further in chapter 6 of this document.

There were two questionnaires used to assess the success of the implementation as I felt it was important to look at it from two perspectives, firstly management were surveyed to attempt to find the organisational viewpoint on how useful the system was.

In this questionnaire the intention was to ascertain if the management felt the tool provided good return on investment. Although there was no cost to the organisation in terms of installation, licensing or maintenance costs at this point there would still have been a cost in terms of the time spent by people reading and writing pages on the Wiki.

The second group surveyed were the users of the system, the software developers within the team. From them the intention was to ascertain how often they used the system and how useful they found it. I was also interested to know if they found it a good use of their time or found it a burden to document their work in another place.

The second source of data was more quantitative and involved interrogating the server logs and the database behind the Wiki to look for usage information and assess how much people were using the system and did that match with the answers given to the questionnaire.

There were three key areas measured here, the quantity of pages stored in the database which sits behind the Wiki, the quantity of attachments stored in the underlying file system, and the amount of searches that had been carried out. This would provide an accurate picture of the amount of content being added to the system, and how regularly people were searching for it.

All these results are shown and conclusions drawn in chapter 6 of this document.

3.5 Conclusion

This chapter brought the focus back onto the project following the introduction and literature review sections. It clarified the motivation behind the project further and showed some of the benefits of managing knowledge effectively. Next it looked at the steps that were required to carry out this project, and focussed then on the practical implementation of the knowledge sharing tool that was deployed as part of this initiative. This chapter serves as a high level guide to what will follow in the document, and points to the areas where more detail can be found on the various topics.

4 PRODUCT SELECTION

4.1 Introduction

In Chapter 2.3 of this document examples of where knowledge management systems had been implemented in the real world were discussed. The next phases of the project was to combine the lessons learned from this section with the outcome of discussions with developers and management from the team in order to come up with a set of requirements that the knowledge system should fulfil. This chapter deals with those requirements and investigates potential solutions to meet them. In section 4.3 the three short-listed applications are analysed to decide whether they meet the requirements. Finally in section 4.4 the most suitable application is selected with a discussion of the reasons why.

4.2 Selection of candidate solutions - Requirements

In order to prevent any issues with budget or implementation delays in the organisation it was decided at an early stage that any candidate solutions would need to be very cost effective. In the table below the issue of cost of considered because even some of the open source solutions considered do have modules and functionality which can be added on for a fee, should the organisation wish to expand its use of the application. These considerations can be seen in the the product comparison matrix(figure 4.1) under the heading *cost*.

Particularly relevant then when we consider open source products or indeed enterprise applications is the issue of support available. The downside to selecting apparently free open source software can sometimes be the lack of good quality support available for these products. This is not always the case of course, and the issue of support has therefore been included as a factor for consideration in the selection of the candidate solution.

A good example of a well supported Wiki implementation is the Dresdner Kleinwort Wasserstein (DrKW) Case Study. Although not a open source tool, there was a good

level of support provided by Socialtext for this implementation and it did appear to feed into the success of the initiative. These considerations can be seen in the the product comparison matrix(figure 4.1) under the heading *support*.

A further requirement was that the solution would be easy for the end users to learn and use. This requirement was decided upon with reference to the case study from Janssen-Cilag Australia, where a simple to use Wiki system proved very effective.

Naturally this selection criterion may necessitate a compromise between ease of use and functionality available to the users but due to the intuitive nature of Wiki interaction this trade off should not pose a significant problem. These considerations can be seen in the the product comparison matrix(figure 4.1) under the heading *usability*.

One aspect of usability that may not be immediately obvious is the strength of the applications search abilities. There is little point going to the effort of adding content to a system if it cannot then be retrieved by the users. The ability to find the correct and most relevant results to a given search are an important factor.

Prior to this implementation the team stored any documentation generated on folder directories on the LAN. This was not always easy to search, and as a result people were less likely to search for documentation, they would be more inclined to go directly to an expert for solutions, which is perhaps not the most efficient manner of operating.

These considerations can be seen in the the product comparison matrix(figure 4.1) under the heading *search*.

Having reviewed the literature on the area it seems appropriate to start with a “grass-roots” implementation and scale this up as necessary depending on the success of the initial pilot. This is particularly informed by the example of the variation reduction advisor from general motors which although developed and rolled-out among a small group of people was eventually scaled up into use across many geographically

dispersed manufacturing facilities. These considerations can be seen in the the product comparison matrix(figure 4.1) under the heading *scalability*.

Again stemming from the literature review another requirement is to be able to pre populate the system with relevant and useful knowledge. As outlined in section 2.4.3 this process has helped previously in getting users to buy in to the concept of using the Wiki system, giving them some value upfront instead of the users needing to enter all the knowledge themselves. Of course any Wiki system can be pre-populated given enough time and resources so I have decided to score this based on how good the tools that are supplied with the application are at facilitating pre-population. This can be seen in the product comparison matrix(figure 4.1) under the heading *pre-pop.*

Stemming both from literature review and from primary research carried out in the team is the requirement to capture knowledge long term so that it may be usable by team members when other team members with that knowledge are not present either for temporal or geographical reasons.

From the primary research in the survey for question 3 one of the management group gave the response below which would appear to support this requirement:
"When developers leave the team it can sometimes leave a knowledge gap in the team"
(Appendix 1.1, question 3)

All the candidate solutions store the content added to them and can be backed up and effectively stored indefinitely. For this reason this factor has not been shown in the product comparison matrix but it will be considered when deciding upon the success or otherwise of the implementation.

So to sum up the requirements for this project in words the need in the development team is for a *low cost* system which can *permanently store* knowledge in a manner that is *easy for other to access*. The system needs to be *scalable* so that it can be rolled out to further user groups in the future, and with the ability to *pre-populate* the system with any available relevant knowledge

4.3 Candidate Solutions

4.3.1 Socialtext

Socialtext is a Wiki solution that can run in the client organisation or be hosted on the vendors server. It provides easy addition and editing of content in the users browser using a supplied rich text editor.

In terms of cost Social text can be used under a personal license by up to 5 users without charge. Above 5 users the cost varies depending on how many users are involved but according to Socialtext that the costs are roughly as follows:

"Groups of around 25 have been in the range of about £100-300 per month. With groups of 100, the pricing has ranged between £800 and £1,500 per month. Typically, the more users there are, the larger the project and the more custom work to be done. This means that pricing for a project of over 1,000 users is not easy to quantify without understanding the scope of the work to be undertaken."
(Socialtext.com, 2008)

There are various other interesting features which will be discussed below:
Personal Dashboard
Socialtext can provide the user with their own dashboard showing things of interest to the user such as:

- Recent changes made by your team
- Pages you are watching
- Workspaces to which you belong
- "Announcements and links" to bring to your group's attention
- "Your notepad" with personal notes and links

(Socialtext.com, 2008)

Email In and Out
Socialtext seems to provide very strong integration with email, allowing users to send pages out of the Wiki by email, but also and more impressively allowing users to email

pages directly into the Wiki creating new pages or updating old pages. This is done by sending emails to the wiki email address with various parameters.

File Management

Socialtext can also handle files, and can attach them to any page in the Wiki. Files can be added into the Wiki by email or by the user browsing to the file location on their network.

Search Facility

Socialtext has a very robust search architecture and can search the following file types:

- Microsoft Word and Excel, Adobe PDF and Postscript, HTML, RTF, and XML
- Zip archives
- MPEG-3 audio file metadata in ID3 tags
- Customize searches for page titles, or search for text in pages with a given tag.

(Socialtext.com, 2008)

Weblog and RSS

Socialtext provides RSS feeds that allow users to track change in content which they are interested In using their RSS readers. In addition users can post to weblogs either online or via email, and can watch other weblogs using rss feeds.

4.3.2 Confluence

This is an enterprise Wiki solution provided by Atlassian software. It allows users to add and update pages in their browser through the use of the rich text editor supplied. There are a host of additional features provided which are discussed below.

The application can be run locally in the organisation or can be hosted by Atlassian for a monthly cost from $49 to $249 depending on the amount of users. The cost of purchasing the application to run in the organisation varies between $1200 and $8000 depending on the number of users involved. Up to 25 users can be licensed for $1200 and for $8000 unlimited users can use the system. This license entitles an organisation to perpetual use and 12 months support. It is optional whether people renew their

license, and would probably depend on the level of support that they required, and whether or not they needed to keep up to date with the latest release of the product.

There are many additional features provided in confluence but some of note are:

Update users in changes within the Wiki

Confluence can notify users by email or RSS fee of any changes to the content in the Wiki that the user is interested in. Users can watch certain pages or entire areas of the Wiki for changes and notified when they are updated.

File Management

Files can be attached to pages, and can be then linked to or searched. Confluence also allows manipulation of these files via windows explorer which is a very useful facility.

Search Functionality

Confluence allows easy searching of all content, i.e. Pages, News Items, Comments, Bloggs, emails, attachments, people. The following attachments can be searched by confluence:

- Microsoft Word documents
- PDF files
- Microsoft Excel spreadsheets
- Microsoft PowerPoint presentations
- HTML, XML, source and text files
- ZIP files

(Atlassian.com, 2008)

Discussions

By adding comments on pages, users can engage in discussions about the content on those pages and this itself can often be as valuable a tool for collaboration as the pages themselves.

Workspaces

Workspaces is a way in which confluence helps to organise content for larger organisations. For a large organisation each section or function may want to have its

own workspace within the Wiki, however users can have a dashboard showing all workspaces if necessary and linking across workspaces is easy.

Security

Confluence has a permissioning system which allows the organisation to control exactly what each user can do. This fine grain control allows confluence to be very dynamic in that it can be used for extranet as well as intranet applications.

4.3.3 Deki Wiki

Deki Wiki is an open source tool available from Mindtouch software. It is an open source product and so can be downloaded and installed free of charge. Unlike some open source products however Deki Wiki does have the advantage of reasonably priced support plans from the creator, which may help to encourage larger organisations to place their trust in it.

Deki Wiki provides the ability for user to easily create, update or delete content (depending on access). It also provides a whole host of useful features which help to make it more useful and secure. The following are the more useful features:

Versioning and Reversion

A version history of each page includes when and by whom the page was altered.

A version history of each attached file includes when and by whom the file was attached. Page versions can be compared for changes. After comparing two versions of a page, a user can revert to the oldest version in one-click.

Access Control

Sites can be made public or private, with anonymous access or not. Permissions can be set for individual users or groups of users. Permissions can be applied to entire hierarchies to create private or non-editable workspaces, or to single pages. We can easily restrict pages for editing, or restrict pages for viewing. Leverage existing user account management systems, such as LDAP, Active Directory, Drupal, Wordpress, and Joomla.

Alerts and Notifications

We can view an automatically generated log of all changes globally sorted chronologically. A log can also be viewed of any user's changes sorted chronologically. Users can create their own watch lists of pages and view the changes in a simple and intuitive interface that sorts the changes chronologically. Users are emailed by the system automatically at critical times, such as at site registration or if they reset their passwords. Users can subscribe to site-wide changes through the global RSS feed, or subscribe to user-specific changes through their user RSS feed. Users also have the ability to subscribe to page-specific changes through their watch list RSS feed.

Advanced search

Deki Wiki uses Lucene.net, a powerful search engine that is developed by the Apache Foundation. Users can search for terms or use sophisticated queries to pinpoint the desired information. Search terms are automatically highlighted in found pages. Users can also search contents of commonly used files including Microsoft Office, OpenOffice, Adobe PDF, TXT, HTML and more. Users can also search attachment descriptions.

Support Available

There is not upfront cost to Deki Wiki, however if an organisation wanted a support package from MindTouch there are various levels of support available. The basic level of support provides access to Community support portal, Alert notifications and updates and Software updates and security patches. In addition with this level of support emails can be directed to a support centre however the number of “support tickets” is limited to two per year. The cost of this level of support is $495 per year.

The silver support level builds on the above adding unlimited support tickets and the ability to contact for real-time chat as well as by email. The response time is halved also from 2 to 1 business days, and there is also partial support for installation included. The cost for this level of support is $2495 per year.

The Gold level support plan is the highest level of support available, and offers telephone support as well as email and real-time communication. The turnaround time

for queries is reduced this time to 4 hours (business day only). There is complete support for installation and configuration, and there is an unlimited amount of support tickets available. The cost for this level of support is $4995 per annum.

Clear for the benefits of any pilot or testing work a support plan would not be necessary however if this tool was implemented in a production environment and users began to rely on it for knowledge sharing and collaboration then an organisation would be well advised to consider the gold level of support obviously depending on the organisations scale and use of the system.

4.4 Product Selection

Product	Cost	Support	Usability	Search	Scalability	Pre-Pop
Socialtext	£100 +	Available	Simple	V. Good	V. Good	Ok
Deki Wiki	Free	Available	Simple	V. Good	V. Good	Ok
Confluence	$1200 +	Available	Simple	V. Good	V. Good	Ok

TABLE 4.1 PRODUCT COMPARISON MATRIX

The figure above was populated following my own pilot install and test of these products on a standalone machine with a small set of data. As can be seen from the figure and indeed the descriptions in section 4.3 all three of the candidate solutions came close in terms of features. All were capable of carrying out the required tasks and were relatively simple to install and run.

The application which was decided upon for installation in the organisation was Deki Wiki. It was found that Deki Wiki could do what the other applications could do, but ultimately cost less to install. It runs on a virtual machine so is very easy to install but can also be run on UNIX so can easily be scaled up, and in the event that the organisation ever needs full time technical support for it that option is available. On balance Deki Wiki provided the best value to the organisation for the purpose of this project although all three applications were impressive in their ability's and simplicity to work with.

4.5 Conclusion

This chapter dealt with the requirements that the best fit solution needed to fulfil. These requirements came from both primary research within the software development team and secondary research in the for of literature review of other implementations of the type being undertaken here.

A number of potential solutions were presented and their potential advantages and disadvantages were discussed. The three solutions that were investigated all appeared to be rich with features and appeared capable of fulfilling the requirements for this project.

Finally a suitable solution was chosen and a brief explanation was provided. The final choice was to go with the Deki Wiki application which met with the required functionality and appeared following investigation to be the best value solution for the organisation.

5 FACILITATING KNOWLEDGE SHARING

5.1 Introduction

The focus of this chapter will be on how the Wiki system was put in place and how people were encouraged to interact with it. As outlined in section 3.4 of this document, there were three distinct phases that took place in order to complete this project. Those phases were:

1. Assess the cultural readiness of the organisation, and gather together the requirements for the system. ***Phase 1***.
2. Install the solution that was selected, and carry out some basic testing to ensure it is operational. ***Phase 2***.
3. Roll out the system to the developers in the team. Monitor the use of the system, and champion its use within the team. During this phase also gather data and feedback which will help to decide whether the implementation was a success. This involved further primary research in the form of a survey of users and review of system output from the system itself. ***Phase 3***.

5.2 Phase 1 – Cultural Readiness and Requirements

As a member of the software development team under investigation the author was of the opinion that the organisation would benefit from more efficient knowledge sharing, however it was necessary to canvass opinion within the team in order to understand the readiness or otherwise of the team for this initiative.

In addition to the opinion of the team and perhaps of even greater importance was the opinion of the team management group, without who's permission the initiative could not go ahead and without the support of whom the initiative would surely fail.

The activity of this phase was broken down into two tasks. Firstly to approach the management group informally to discuss the proposal and gauge opinion. At this point

the objective was to ascertain would they allow this type of research to be undertaken in their department, and would they participate in this research.

These informal approaches proved successful and the management group of the team confirmed they would be happy to be involved in the research and that they would be happy to have members of the team participate in the research. One key point that emerged from these discussion was that the management group were happy to commit some of their own and their team members time to participate in the research and potentially participate in any knowledge sharing initiative which may follow, but they were very clear that there was no budget available at this point. This stipulation did feed into the requirements section of this project which can be seen in section 4 of this dissertation.

With this agreement in principal in place the next step was to was to survey both management and members of the development team independently using a structured questionnaire, and then follow up any particularly interesting answers by further interviews. The same questionnaire was used for both management and staff, and can be seen in appendix 1.1 of this dissertation.

A representative sample of staff members was taken with some senior developers and some junior developers surveyed in order to ensure good coverage of the sample space. In total 2 Managers were surveyed and 5 staff members. The table below shows the names that were assigned to these individuals and shows their position in the organisational hierarchy of the team. Manager M1 is responsible for this team and several others, so Manager 2 reports into them. All developers in this team would report into Manager 2, however for project related activities(i.e. Breakdown and supervision of work) Developer 4 would report into developer 1 and developer 5 would report into developer 2.

	Manager 1 (M1)	
	Manager 2 (M2)	
Developer 1 (D1)	Developer 2 (D2)	Developer 3 (D3)
Developer 4 (D4)	Developer 5 (D5)	

TABLE 5.1 TEAM HIERARCHY

Over the following pages the answers supplied to each question will be examined and an attempt made to gauge the cultural readiness of this team for the proposed change to their work practices. The actual list of questions and answers can be seen in appendix 1.1.

5.2.1 Question 1

Can you give some examples of where we need to share knowledge in the team?

This question was designed to find out what people understood about knowledge sharing that already takes place within the team, and find out about where people felt knowledge sharing was important to the team. The most common area to emerge was that of staff training. The majority of people felt that knowledge sharing was necessary to facilitate training of staff both new and existing members of the team.

One interesting theme that emerged was that of learning from lessons learned, and sharing these lessons among the team. A number of respondents mentioned this as an area where knowledge sharing was necessary. The final point of interest here was that of knowledge retention in the team after expert developers leave.

Judging by these answers it would appear that that the team see a role for knowledge sharing in practice and it would appear they would see practical applications for further future expansion of that knowledge sharing.

5.2.2 Question 2

In what ways do we currently share knowledge well within the team?

This question was designed to find out what the team feels they currently do well. It is important in any change initiative not throw away things that work well simply because it represents the old way of operating. Of course on the other hand the team may feel certain practices are working well but to the outside observer this may not appear to be the case.

The largest single theme to emerge from the survey was that of mentoring of staff. All new staff are assigned a mentor and work closely with their mentor to learn the skills necessary to carry out their role. This type of interaction is an example of tacit to tacit

knowledge sharing where the knowledge is passed directly from one human to another. It is also known as socialisation.

One interesting answer makes reference to project documentation, and perhaps what is most interesting is that documentation is only referred to by two out of seven answers. From the answers we can see that documentation in some from exists and is created regularly, but how often it is referenced is not clear.

Judging by the answers to this question there would appear to be a good atmosphere of collaboration in place already within the team, although it appears very focussed on the tacit to tacit sharing of knowledge. The answers to this section appear to support the need for some mechanism to assist tacit to explicit, and explicit to tacit knowledge sharing.

5.2.3 Question 3

What restrictions are there currently to knowledge sharing in the team?

This question was designed to find out was there any obvious blockers to knowledge sharing that could easily be removed to facilitate knowledge sharing. It was also hoping to find out if there were any restrictions/additions that should be put on the new system to make it more useful.

The overriding theme in the answers here appears to be that of time constraints. People are busy and cannot always dedicate the time they would like to to knowledge sharing. This concern would need to be brought into the requirements of any knowledge sharing system (see section 4.2 of this document), and any potential system would need to deliver results with the least possible burden on the time of the users.

Another area that is raised is that of expertise being held in a small group of people(functional areas of the system), and that expertise being lost when they leave the team. This being raised as an issue would appear to support the need for a centralised repository of knowledge that could be consulted indefinitely by people in the team as they needed it.

5.2.4 Question 4

Does knowledge impact on performance?

This question was designed to find out if people felt there was a link between the amount of knowledge a person had at their disposal and a persons effectiveness in the team. The ultimate aim of this knowledge initiative is to improve efficiency so some correlation between the two is essential.

The answers to this question were unanimously that knowledge did have an impact on performance so it should follow that given access to quality knowledge in a timely manner, the performance of the team would improve. Following on from this a question about the teams performance post implementation was included on the questionnaire which was circulated after the system was rolled out. The answers to that question are discusses in chapter 6 of this document.

5.2.5 Question 5

Do we need to make any changes to how knowledge is shared in the team?

This question was designed to find out if people felt change was needed, which could indicate how likely they would be to support change. It was also hoped that in answer to this question people might make suggestions or comments which would be beneficial in thinking about the actual implementation.

All respondents endorsed change in the team to varying degrees. The focus of the developers interest seemed to be on having access to the knowledge they needed, as quickly as possible. This would facilitate them in their role, and support them in getting the job done. From the management perspective the view point seems to be that they would be interested in any initiative that helped to improve efficiency.

Interestingly both sides of the staff/management divide are looking for efficiency, and support in getting the job done as quickly and as well as possible. They are just coming at it from slightly different perspectives as we is perfectly understandable.

5.2.6 Question 6

Would you support an initiative to improve knowledge sharing in the team?

This question was designed to spell out clearly that an initiative would be taking place and find out if people would support such an initiative. The question is deliberately phrased in order to elicit an problems people might have with involvement in *yet* another change program.

Broadly speaking the answers received to this question were very encouraging. Every participant indicated that they would support a initiative to improve knowledge sharing. There were some caveats mentioned but these were expected and indeed encouraging as they were entirely treatable.

The first issue mentioned was from the management group and was a restriction in cost terms. This had already been outlined by the management at the earlier meetings so was not an issue. The second concern mentioned was that any knowledge sharing system doesn't become a burden on its users. This comment is factored into the requirements for the system in section 4.2 of this document. It stands to reason that a system to improve knowledge sharing and collaboration should be lightweight and not a burden to the users, however for designers of these solutions it is useful to bear this in mind.

5.2.7 Phase 1 – Conclusion

It seemed that all those surveyed had a positive outlook on change and were open to the efficiency gains possible through improved knowledge sharing. The team did appear from their answers to have a good team work and collaboration ethic albeit engrained in tacit to tacit knowledge sharing at present. This would appear to indicate that they would be prepared to share the knowledge they each held.

Finally the management were supportive of the initiative and seemed eager to investigate the potential cost savings and efficiency's that the project could deliver.

5.3 Phase 2 – Software Installation

The application selected for this implementation was Deki Wiki, as discussed in chapter 4 of this document. One of the benefits of this application is that it is very

easy to install and get up and running. The version installed was the Deki Wiki (Hayes version 1.8.1).

There are two options to run the Wiki. It can be run on a PC using a virtual machine to simulate a Unix environment (Vmware), or alternatively the application can be installed on a Unix server along with MySQL and a server in order to run it there.

For the purpose of this project the Vmware installation was deemed to be the best option as it offered very easy installation, and didn't require any intervention from teams outside the software development team in order to begin using it. In addition to this, there was no need for dedicated hardware or support resources when the machine was run on a PC in the team, and due to the fact that cost was a factor for this project this method of implementation was ideal.

In addition to the basic install of the Deki Wiki application, there were a number of add-ons that were necessary. Both the add-ons were supplied free by Mindtouch software, and made using the Wiki easier. The first was the windows desktop connector, which allowed the Wiki file system to be viewed in windows explorer and allowed for the drag and drop large amounts of files onto the Wiki, making them accessible to the users. This add-on greatly aided the pre-population of the Wiki with useful material from the existing LAN directory's.

The second add-on was the outlook connector. This allows users to publish emails directly into the Wiki from their outlook mailbox. Users can also publish mails that already have attachments, and these pages and their attachments then become searchable by other users, so this is a very powerful tool, and greatly improved the user experience, and ease of publishing.

Despite the fact that no support package had been purchased for Deki Wiki from Mindtouch software, there was still extensive information and assistance available through the Deki Wiki website, where most of the issues encountered during the install had previously been experienced by other users and answers posted by the user community.

One issue in particular which caused difficulty was in connecting the outlook connector to the Wiki. At first this did not appear to work despite the fact that it had installed cleanly. At the time the outlook connector was an alpha release and had some known issues, so it was a little more difficult to find the solution to this issue. Eventually thanks to support from Deki Wiki's own support site a solution was found whereby Outlook service pack 3 and internet Explorer version 7 were installed on the machines which needed access to the Wiki prior to the users attempting to add emails directly into the Wiki.

Once these infrastructural components were in place, It was necessary to make some changes to the appearance of the Wiki. This was not difficult as there is support in the application for basic look and feel changes. Extensive changes are possible, and libraries of skins exist, but for the purposes of this project the only change necessary were the addition of a company logo, changing the colour scheme to match the company's and changing the name of the Wiki homepage.

Once this was complete testing commenced and as no major issues were encountered during the test the next phase commenced.

5.4 Phase 3 – Roll-out to developers & Collect Usage Details

Often IT projects are technically successful but ultimately fail in the organisation for cultural reasons. For this reason this phase was probably one of the most crucial. It was not enough to tell developers there was a tool in place, they needed to buy-in to it. They needed to be convinced that it would benefit them, as well as the organisation and that its use would reduce workload as opposed to increase it.

The first step was to create a presentation on what the system was designed to do and how it worked. In the presentation there was no mention of the system being a Wiki, following the example from the case study in Jansen-Cilag (see section 2.4.2). In that case study the author mentioned that users of the JCintra site *"don't view it as a wiki with all the anarchistic overtones that brings. Rather they see the simplicity and flexibility as a natural evolution of Intranet technology*"(Wallace, 2007).

This presentation was given to the development team at an already scheduled team meeting. The presentation was short as the concepts were simple, but the main purpose of the presentation was to bring the idea into peoples conciousness and show them some of the clever things this system could do very easily.
Following this presentation discussions were held with each developer individually to ensure they understood what the system was and crucially what it wasn't. I then made sure to use the system myself and attempted to encourage others to use it by arranging some communication through the Wiki rather than traditional means.

An example of how this was handled was following a design meeting between myself and 3 other developers, I created a design document, but instead of sending it to the other developers by email (which would have been the prior process) I sent a link to a page which showed the design in the Wiki. I asked people to add their comments to the page rather than by reply email to me, and this proved to be a useful example when showing others the benefits of collaboration in this way.

This type of open collaboration when possible really does facilitate easier knowledge sharing, as in this case other team members at the time and in the future would be able to view not just the design document but the discussions that occurred around that document. These discussions would previously have been held out of sight in someone's email box, so this alone is a step forward.

Over the course of the pilot implementation usage statistics were recorded on a weekly basis such as the amount of content on the Wiki, the frequency of user activity, and the quantity of searches that were carried out. In addition to this people were encouraged to use the system and assisted with any issues they had. The details of the usage statistics and the users perceptions are looked at more closely in chapter 6, however it is interesting to note that as the pilot went on some developers began to really see the benefits of the system and to experiment with adding new sections and changing sections of the Wiki to make it more useful to them. This progress was encouraged as It was felt that this tool would evolve best with the input of the actual users (grass-roots support).
The final step in this phase was to carry out another survey of the users to find out what they thought of the system itself and what they thought of the state of knowledge

sharing in the team post implementation. This survey can be seen in appendix 1.2. For a further discussion of the responses to the survey and the conclusions drawn see chapter 6 of this document.

5.5 Conclusion

This chapter looked at the practical implementation of a Wiki based knowledge sharing system in a software development team. The focus of the chapter was on the phases that were involved in the process.

The first phase looked at the culture of the team and by survey sought their opinions on the current position of the team and how open they were to change. The outcome of this section would appear to support the potential for an automated knowledge sharing system.

The next phase was focussed on the actual install of the candidate solution. This phase was also concerned with the addition of any necessary add-ons and the initial test of the system to make sure it was correctly installed and ready for use.

The third phase was concerned with the roll-out of the system to developers after it was installed, and all the work that was required to try to help people find their way around the new system and see how useful it could be. During this stage also usage statistics were collected and a survey was carried out to inform the decision on whether or not the implementation was a success.

6 EXPERIMENTATION & EVALUATION

6.1 Introduction

This chapter reflects on the success or otherwise of the implementation as judged against the requirements set forth in chapter 4.1. It also investigates the success of the project from an organisational point of view as well as from a personal point of view for the developers on the ground.

6.2 Experimentation

The main experimentation involved in this project was the implementation of a knowledge sharing system as outlined in chapter 5 of this document. Before and following this implementation surveys of the staff involved were carried out, and the usage of the system was monitored. Below the outcome from the second of these surveys as well as the usage information that was gathered is discussed in more detail. The conclusions drawn from these sections then lead into section 6.3 and 6.4 where the overall outcome of the project is evaluated.

6.2.1 Usage Statistics

During the implementation of the system It was possible to record some information which would help to determine how much it was being used. The first set of data presented below relates to user activity. This was simple to collect but quiet useful in assessing peoples regularity of usage. Following on from that there is some data presented which relates to the amount of content in the Wiki. This information is presented in the table below (marked table 6.2) showing the various figures over the course of the pilot. Each heading is explained below. The tables represent the position at the end of a week starting when the system was rolled out and ending after 4 weeks. The first line represents the position just before the system was rolled out as there had been an exercise carried out to pre-populate the system with relevant available knowledge (e.g. Existing documentation or saved emails).

One additional statistic which gives a picture of peoples usage of the system is the user screen which is available to all Deki Wiki users by clicking tools, then users. This screen shows a list of all users currently set up in the system and when they were last active. It is also a useful screen for users of the system as it allows them to link to other peoples contributions.

The table below shows a representation of the amount of users, and when they were last active. These details were collected each Friday afternoon during the pilot.

Day Last Active	Mon	Tue	Wed	Thu	Fri	Total Users
Week One	0	0	6	3	5	14
Week Two	1	4	2	7	3	17
Week Three	3	1	5	3	6	18
Week Four	1	3	6	5	3	18

TABLE 6.1 USER STATISTICS FOR KNOWLEDGE SHARING SYSTEM

What these statistics show is that all those who were asked to participate in the pilot did register as users and did add some content each week during the pilot. Perhaps the reason for peoples continued use of the system during the pilot could be traced back to the encouragement that was provided by the author, but nonetheless they did log on and use the system. These results are particularly encouraging and would tend to indicate that the implementation was a success and that people would use the system in the future provided perhaps that there was still as much encouragement and support for them system in and around the team.

The next usages statistics of interest are those collected from the server side of the Wiki system. Deki Wiki runs on an Apache Server, and so does leave considerable information available to the administrators of a Deki Wiki installation. In addition to the server logs it was also of interest to monitor the quantity of information stored on the MySQL database which sits behind the Wiki system and the file system which holds the attachment files for the Wiki. The quantities of these attachments would indicate how many of the created pages had content attached to them such as word documents or text files.

The following table shows some key indicators of the usage of the system, the individual columns are explained further below. The five rows represent the position after pre-population firstly, then the position at the end of each week during the pilot. Note the number in the box represents the running total on the right with the amount added for that week shown in the box on the left.

	Pages		Attachments		Searches	
	Add	**Total**	**Add**	**Total**	**Add**	**Total**
Before Week 1	175	175	90	90	0	0
End Week 1	32	207	5	95	73	73
End Week 2	40	247	13	108	140	213
End Week 3	49	296	7	115	208	421
End Week 4	45	341	12	127	206	627

TABLE 6.2 CONTENT STATISTICS FOR KNOWLEDGE SHARING SYSTEM

The column marked pages in the table relates to the amount of rows returned when the query below was executed within the MySQL database which runs behind the Deki Wiki system (wikidb is the name of the database in MySQL):

SELECT * FROM PAGES

In Deki wiki each page has a record in this table which holds necessary information so this table represents the quantity of content in the wiki.

The column marked searches represents the amount of searches carried out by the system users. TO find this out the unix command below is issued and the matches that are returned each represent a search being carried out.

cat access.log | grep “search=”

This looks through the log file which is produced within the Apache server and picks out all entries which relate to a search request by the users. As an alternative to this command the log file could be exported to windows and manipulated using some tool, for example excel.

The column marked attachments relates to the quantity of documents in the Wiki. This can be easily seen by looking in the file system where these attachments are stored, or

alternatively executing query in MySQL looking at the attachments table which is how the physical files are linked back to the pages on which they are to appear. To find out the amount of attachments the following should be executed in MySQL:
SELECT * FROM ATTACHMENTS;

These statistics show a steady increase in the amount of content available on the system. This would appear to indicate that people are using the system and adding their own content to the system, which would in turn indicate a sucessful implementation.

The search figures show a gradual increase in the amount of searches weekly in the first weeks of the pilot but then a levelling out of the activity. This would be consistent with people finding their way around the system and getting comfortable with using it. Some users carried out more searches than others but again this is to do with peoples different patterns of working and technological adoption.

The figures were encouraging as the system did look to be genuinely useful, and the data collected did appear to be broadly in line with the opinions collected during the second survey. As with the user statistics these statistics would indicate that system usage needs to be monitored and users encouraged if it becomes evident that their usage of the system is beginning to tail off.

6.2.2 Post-Implementation Survey

Following the implementation of the knowledge sharing system, and it operation in the target organisation for a number of weeks (i.e. The period of the pilot), a survey of some of the users was carried out. The actual questions asked and answers submitted can be seen in appendix 1.2 of this document.

The group surveyed were the same as those surveyed in the cultural readiness survey which was carried out before system requirements were gathered and the system was designed. A description of this group can be seen in section 5.2 of this document but essentially there were 7 people in total surveyed, 5 developers from the team and two members of the management group.

It was decided this time to ask the management group different questions to the developer group. The reason for this was to try to understand the impact of the project from both an organisational point of view as well as from a personal point of view. Below each of the questions is examined individually and the answers analysed. It is important to highlight at this point that as the second survey was circulated the respondents were asked to make their answers as verbose as possible and avoid responding with yes or no answers. The reason for this is that some of the responses to the first survey were quiet short and perhaps therefore less useful than others.

Management Survey - Question 1

Has the team performance improved since the introduction of the initiative ?

This question was designed to find out if the knowledge management system had lead to any tangible improvement in performance as measured using the performance metrics the management team would normally use to judge team performance.

The consensus of the respondents would appear to be that no there was no tangible improvement in performance that could be attributed to the system. An improvement in the future though through use of this system is not ruled out and will be monitored.

Management Survey - Question 2

Do you think the system provides good return on investment for the developers time?

Although there was no cost in implementing this system in the organisation there is a cost in terms of developers time in populating and searching the knowledge system, so this question was designed to see if management felt this was warranted and would continue to support it going forward.

The responses were positive and the only proviso was that similarly to question 1 above, things would need to be monitored on an ongoing basis to ensure this return on investment continued.

Management Survey – Question 3

Has collaboration across the team among team members improved as a result of the initiative?

This question was designed to find out if the management felt the softer side of the performance agenda had improved. Did the management feel that the team were working together better and more easily after this implementation.

The responses indicate that management were aware of/perceived an improvement in the team collaboration dynamic. The system had not lessoned or replaced the need for face to face interaction but did offer something else which hadn't been there and was useful.

Management Survey – Question 4

Has this initiative been a success? Would you recommend and initiative such as this to other team leads/Departments?

This question is designed to judge the managements overall feeling about the outcome of the project. In addition the question about whether they would recommend this implementation to other teams does impact on the future work section of this project.

The responses to this question would appear to indicate that the project has been a success. As with previous responses the caveat of maintaining current performance/usage into the future is mentioned but generally speaking the sentiment is very positive.

Developers Survey – Question 1

What was good about the knowledge sharing system. Did you add much content and did you use it much?

This question was intended to find out what people felt worked well about the system and find out how much people were using the system. If people felt it was useful but didn't use it much it might indicate some further work was required to make the system more accessible to people.

The responses to this question generally point to the system being easy to use and easy to add and retrieve content to, so this is a very positive comment in light of the requirement for a user friendly system which people could learn quickly.

In terms of usage of the system all those asked indicated regular usage with some indicating daily use. This again is an encouraging response, and can be cross checked for accuracy against the usage statistics collected.

Developers Survey – Question 2

Did this system add to or reduce your workload?

For this system to be a success it is essential that it got grass roots support from the users, and this wouldn't be possible if they felt it was a chore, or if it was just another form of documentation that they had to complete after/while they were going their development. This question was intended to show how much of a burden if at all the system was for developers.

Some of the respondents felt it was a slight increase in workload but the consensus seems to be that it was a worthwhile addition to the work practices and that generally speaking time invested in the system was delivering a benefit down the line.

Developers Survey – Question 3

Did you find it easy to search? Were the results returned relevant to what you were searching for?

This question was designed to find out how effective the retrieval of knowledge from the system was. This part of the system functionality was crucial as otherwise it would end up being just a dumping ground that people would quickly grow tired of searching.

The responses here were encouraging in the most, with most respondents indicating that searching was easy and reasonably fast. There does however appear to be an opportunity for future work here to ensure that content is properly organised, and that the search function is as efficient as it possible.

Developers Survey – Question 4

What could be done better? What should we add to it?

This question was attempting to find out from the users if they had any ideas to further improve the system that had been implemented. The responses from here could again feed into the future work section of this project.

The most common response was that the content available needed to be increased, and this would naturally happen as people used the system and added the content themselves as they had it available. Some future work may be to find out if there was a particular type of content that would useful and that wasn't being added naturally, and investigate a piece if work to add that content.

In addition it was suggested that the size of the user group be expanded and that is again something that would happen as the system evolved naturally and was rolled out to other teams and departments.

The final suggestion that would be of interest for future work was to automatically populate the system with relevant data from the organisations mainframe, and this would certainly be something that could be investigated as part of future work. A regular extract from the mainframe such as a text file could be automatically loaded into the Wiki system, but this was beyond the scope of this project.

Developers Survey – Question 5

Did this initiative improve collaboration in the team? What could help improve it more?

This question was intended to gauge if the developers felt that the system had delivered better collaboration within the team. These responses would feed into the overall success of the project. Similarly to question four this question was also intended to isolate any opportunity's for potential future work which developers might be able to suggest.

The general consensus among the respondents was that this system had made collaboration in the team easier. There are a number of suggestions for imprvements that would help to further facilitate collaboration in the team, perhaps the most interesting of which is communication. Maybe the system itself could be reviewed to investigate the ways in which it can facilitate communication, and this could be build into a training course or presentation for staff. This process could even happen informally within the team as one developer begins to utilise the system in a given way they could then let the other users know about that ability.

One final point that deserves particular mention is that point that was raised about tying collaboration into peoples objectives. Often peoples performance objectives are for very job specific things but the concept of building teamwork and collaboration into peoples individual objectives as a way to help improve collaboration would appear to be an interesting one.

6.3 Evaluation

This project can be evaluated from two points of view. Firstly the evaluation of the practical knowledge sharing system which was built as part of this project, and secondly the evaluation of the other aspects of the project and the project as a whole.

In terms of the knowledge sharing system that was implemented, its success can be evaluated using the initial requirements discussed in detail in chapter 4 of this document. It can also be evaluated with reference to the second survey that was carried out in the development team. The initial requirements for this system were introduced in chapter 4 under the headings below:

- Cost
- Support
- Usability
- Search
- Scalability
- Pre-Population

Cost

The system that was implemented was Deki Wiki, and it is supplied as an open source application by its creators. For this reason it clearly met with the requirements for this project. If the organisation was to expand usage of the system to additional teams and departments then it would probably be sensible to purchase so level of support from the vendor, however for the purposes of this project it can be considered to have successfully met with the requirement to be cost free.

Support

This product does have the option to purchase a support plan however that was not relevant or appropriate for this pilot implementation. There was a need to consult the organisations Wiki site when installing and configuring the system, and this website was helpful and provided the required solutions. For this reason and the fact that support plans are available this requirement was successfully met.

Usability

A key requirement for this system to ensure people would use it was that it would be uncomplicated and easy to use without a need for much training. When the survey was carried out it appeared to indicate that the system users felt that it was easy to use. In fact in answer to question 1 all respondents indicated in some way that the system was easy to use and learn (see appendix 1.2). For this reason it would appear that this requirement was successfully met.

Search

The search functionality was also an important consideration as it was useless to have a large repository of knowledge if people could not find what they needed from it. The speed of the search and relevance of results in particular where what was of interest The performance of this aspect of the system was measured in the second survey to find out what the users thought.

The consensus of opinion among the staff seemed to be that the system could efficiently locate the relevant content for them, however there were a number of reservations on this criteria, and it is something which would perhaps benefit from some future work. On balance the requirement would appear to have been met but with the proviso that the content organisation be monitored to ensure it is optimul going forward.

Scalability

In the pilot roll-out of this system it was only to be used by one team, and the intention was that if successful it could be rolled out to futher users. This roll-out would be entirely possible and actually quiet simple in technical terms, with the only major job involved to move the application to a dedicated server if needs be for better performance.

For the purposes of this pilot the Wiki system was run as a virtual machine on a PC connected to the local area network. This was acceptable for the purposes of the pilot as the user group was less than 30 people, and the performance of the system was acceptable. If the system was to be rolled out to a large user group it might be best to consider setting it up on a unix server, with the consequent performance improvement helping response times for users.

Apart from that the system is completely scalable so it would appear this requirement was adequately met.

Pre-Population

One of the requirements for the system was that it would facilitate some form of mass automated loading of knowledge so that it could be pre-populated with useful content. This Wiki system does have such a tool which was used in this organisation, the Deki Wiki desktop connector which allowed the drag and drop of existing documentation/content. In light of this fact it would appear that this requirement has been met.

Overall it would appear that all the requirements that the system needed to meet were delivered to a very satisfactory level so in pure system terms it would appear the project has been a success.

In terms of the project as a whole the initial intent was to investigate if the introduction of a knowledge sharing initiative could facilitate easier collaboration and knowledge sharing among the members of the software development team thereby contributing to faster turnaround on work carried out and ultimately to reduced costs for the organisation.

Judging by the answers to the survey people felt that the system was useful and they found it simple to use. They felt it did make the sharing of knowledge in the team easier so the project would appear to have helped facilitate easier collaboration thereby fulfilling that aspect of the initial plan.

One key point mentioned in the overall project aims was to ultimately reduce costs for the organisation. Although there is still potential for this to happen in the future as the use of the system continues, there is currently no evidence that that has been achieved. According to the management respondents in the survey when asked about the performance of the team post-implementation:

"In terms of performance measured against closed work items this has been constant, but developers use of the system has been impressive and future usage of the system and team performance will be monitored."

"The quantity of work items delivered has not improved significantly however there can be many factors that effect the speed of delivery on these items e.g. Complexity. Overall we feel that this system is adding value and should continue for the foreseeable future."

So these answers do indicate that in terms of measurable cost reduction this has not yet been achieved but they are still positive about the initiative and feel that there may be some gain possible down the road with continuing use of the system.

One area where this system has brought a noticeable change is in the area of searching for content. The prior process was that once completed documents were placed on a shred LAN directory structure, but this became difficult to search, and often people would avoid searching it as they did not have confidence in finding what they needed. The Wiki system on the other hand is easy to search and returns relevant results from pages and also from within attachments. If the content on the system is maintained in a tidy and up to date manner, then this change will be a significant improvement in allowing developers access to documentation.

6.4 Conclusion

In this chapter the second survey of managers and system users was discussed and the answers provided were analysed. The survey outcome generally was quiet positive with users finding the system easy to use and fit for purpose.

The system was then evaluated in relation to the initial requirements, and it appeared to perform well against those requirements. One area that might require future work was the search functionality, indeed it is suggested that this is an area that would require vigilant supervision to ensure its continued success.

Finally the project as a whole was evaluated against the initial intention, and the results of the survey appeared to indicate that the project had only partially delivered its goal at this point but that in time this could be achieved with continued use of the system.

7 CONCLUSION

7.1 Introduction

This chapter looks at the project as a whole, looking at what the research problem was again and investigating how this research was approached. It then looks at what this research has contributed to the body of knowledge. It looks at the experimentation that was carried out and any limitations that may be necessary to point out. Finally it proposes future work that is indicated having completed this project.

7.2 Research Definition & Research Overview

This project uses a small software development team to investigate the need for efficient knowledge sharing in the software development context. It involved consulting both the team themselves and the body of knowledge on knowledge sharing and knowledge transfer to decide upon a set of needs that the organisation and particularly the team had.

From that point a solution was implemented using modern Wiki technology which was designed not to change how the team operated, but to assist and make knowledge capture and sharing easier.

The next piece of primary research was a post-implementation survey of some users and management. This in conjunction with the usage statistics captured from the Wiki system gave an indication of how much the system was being used and how useful the users found it to be.

7.3 Contributions to the Body of Knowledge

This project would be useful to anyone who was considering carrying out a similar project in a similarly sized team, or in a small development team. The actual implementation in this case was quiet successful so perhaps other practitioners could follow aspects of the implementation as appropriate.

More so than that though perhaps the most useful contribution of this project was to further backup the viewpoint that Knowledge management initiatives benefit from being implemented in the small scale rather than the big-bang approach. This project achieved success in part due to its scale and the fact that the author could monitor progress regularly and encourage and assist users of the knowledge system. This view that grass-roots (user focussed, small scale initially) initiatives are most successful was inspired with reference to the literature for example the case study which is investigated in section 2.4.1 of this dissertation, the General Motors variation reduction advisor. This initiative was very successful, but began in local small teams(communities of practise), and got great grass-roots support.(Morgan et al, 2002)

7.4 Experimentation, Evaluation and Limitation

The main experimentation carried out as part of this project was to install and pilot a Wiki bases knowledge sharing system in a software development team. Looking at the requirements for this system and the results of the implementation it would appear to have been a success.

One major limitation was the time available. The initiative didn't have time to impact on actual measured team performance during the pilot of the knowledge sharing system. Perhaps given time this could have been more closely investigated. In addition to this time and resources restricted the amount of people who could be interviewed for the primary research part of the project. To have interviewed all team members as opposed to a selection may have delivered some unseen interesting results.

Another possible limitation was the fact that the primary research was carried out among a user group who were familiar with the author. There is a chance that this may have affected the respondents, perhaps not being as honest as if it was an external entity asking the questions.

7.5 Future Work & Research

In the case of this project the system that was implemented was only introduced on a pilot basis into one team in an IT department. Following on from the apparent success of the implementation it may be necessary in the future to investigate further

expanding the use of the system within the department. This is beyond the scope of this project but the author will more than likely be involved should that work go ahead in the organisation.

One interesting point that emerged from the post-implementation survey was that although people were able to search for what they wanted, a number felt this was an area which could be improved or at least needed to be monitored to prevent the system being swamped with content. This question of how best to organise content and keep it relevant is a very complex one and would merit further investigation.

7.6 Conclusion

With the move happening in Ireland towards a Knowledge economy, it is essential that organisations efficiently manage what is fast becoming one of their most valuable assets, their Knowledge. This knowledge is presently in many places printed pages, embedded in business processes even peoples heads. To be successful and survive organisations need to know what they know, they need to know and understand all the blocks of knowledge that support their existence. If they are to survive in this increasingly competitive environment, Organisations must make every effort to reduce cost and improve efficiency by effectively managing knowledge.

REFERENCES

Charette, R., “Why Software Fails”, IEEE spectrum 2005 (accessed 20th February 2008), http://spectrum.ieee.org/sep05/1685

Charles, R., Ranmi, A, “Wild, Wild wikis, A way forward”, Fifth International conference on creating, connecting and collaborating through computing.

Chau, T., Maurer, F., Melnik, G., "Knowledge Sharing: Agile vs. Tayloristic Methods", Proceedings of the twelfth IEEE workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise.2003.

Chua, A., "Knowledge sharing: a game people play", Aslib Proceedings, Volume 55, Number 3, p117-129, 2003.

Confluence product Information (Accessed 19th February 2008), http://www.atlassian.com/software/confluence/features/.

Cristal, M., Reis, J., “Leveraging lessons learned for distributed projects through Communities of Practice”, IEEE International Conference on Global Software engineering (ICGSE’06), 2006.

Davenport, T. & Prusak, L., (1998), “Working Knowledge”, Harvard business school press.

Deki Wiki product information (Accessed 10th February 2008), http://wiki.opengarden.org/Deki_Wiki

Dingsoyr, T.; Royrvik, E.; An empirical study of an informal knowledge repository in a medium-sized software consulting company, Proceedings. 25th International Conference on Software Engineering, 2003.

Dorgan, S., "Ireland is now a knowledge economy". http://www.idaireland.com/uploads/reports/BusinessIre_win03/index.html (accessed 16th February 2008)

Dresdner Kleinwort Wasserstein Case study available from http://www.socialtext.com/node/80 (last accessed 21st March 2008)

Endres, M., Endres, S., Chowdhury, S., Alam, I., "Tacit knowledge sharing, self-efficacy theory, and application to the Open Source community", Journal Of Knowledge Management, Volume 11, Number 3, 2007.

Gopal, A., Mukhopadhyay, T., Krishnan, M., "The Role of Software Processes and Communication in Offshore Software Development", Communications of the ACM, April 2002/Vol. 45.

Gruder, R., Stanoevska-Slabeva, K., Fierz, W., "Implementing a Knowledge Medium in a Multi-Centered Clinical Trial", Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999.

Hall, H., Goody, M., "KM, culture and compromise: interventions to promote knowledge sharing supported by technology in corporate environments", Journal of Information Science, Vol. 33, No. 2 , 2007.

Hasan, H., Meloche, J., Pfaff, C., Willis, D., "Beyond Ubiquity: Co-creating Corporate Knowledge with a Wiki", 2007, Proceedings of International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies(UBICOMM 2007).

Horn, M., "Promoting Internal Collaboration with an Enterprise Wiki", Peer to Peer Magazine, August 2007. Available from http://www.iltanet.org/ (accessed 20th March 2008)

Ladd, D., Heminger, A., "An Investigation of Organisational Culture Factors That May Influence knowledge Transfer", Proceedings of the 36th Hawaii International Conference on System Sciences, 2002.

Lin, C., Li, L., Hu, W., Chen, G., Liu, B., "Constructing an authentic learning community through Wiki for advanced group collaboration and knowledge sharing", Proceedings of 7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007).

Mangione, C., "Software Project Failure: The Reasons, The Costs", http://www.cioupdate.com/reports/article.php/1563701 (accessed 19th February 2008).

McManus, D., Wilson, L., Snyder, C., "Assessing the Business Value of Knowlede Retention Projects: Results of Four Case Studies", proceedings of 2004 Conference in Decision Support Systems (DSS204).

Morgan, A., Cafeo, J., Gibbons, D., Lesperance, R., Sengir, G., Simon, A., "The General Motors Variation-Reduction Adviser: An Example of Grassroots Knowledge Management Development", Practical Aspects of Knowledge Management, Vienna, Dec 2-3, 2002.

Nonaka, I. & Takeuchi, H., (1995), "The Knowledge-Creating Company". Oxford University Press

O'Connor, M., "Financial services firms of the future will be characterised by a pervasive customer-centric culture", http://www.finance-magazine.com/display_article.php?article_id=5995&node=1 (accessed 19th February 2008)

Petter, S., Mathiassen, L., Viashnavi, V., "Five Keys to Project Knowledge Sharing", IT Pro (published by the IEEE computer society), May/June 2007.

Pickett, L., "Focus on technology misses the mark", Industrial and Commercial Training Volume 36, Number 6, 2004.

Socialtext product information sourced from http://www.socialtext.com/products/tour (accessed 19th February 2008)

Stenmark, D., “Turning Tacit Knowledge Tangible”, 2000, Proceedings of the 33rd Hawaii International Conference on System Sciences – 2000.

Vitale, M., “Recovering from software development”, EDUCAUSE review January/February 2000.

Voit, J., Drury, C., “Supporting Vicarious Learning With Collaborative Lessons Learned Programs”, IEEE Transactions on Systems, Man and Cybernetics- Part A: Systems and Humans, Vol. 36, No. 6, November 2006.

Wallace, N., “Our Intranet, the Wiki: Case Study of a Wiki changing an Enterprise”, Accessed (10 Jan 2008), http://www.e-gineer.com/v2/blog/2007/08/our-intranet-wiki-case-study-of-wiki.htm, 2007

Wei, C., Hu, P., Chen, H., "Design and Evaluation of a Knowledge Management System", IEEE Software Journal May/June 2002.

Wikramasinghe, N., Gupta, J., Sharma, S., “Creating Knowledge based Healthcare organizations”, 2004.

Yang, S., Chen, I., Su, A. “Personalized Annotation Management: A Web 2.0 social Software for Enhancing Knowledge Sharing in Communities of Practice”, Seventh IEEE International Conference on Advanced Learning Technologies, 2007.

APPENDIX 1.1 – CULTURAL READINESS SURVEY.

Below are the questions and answers collected during the assessment of cultural readiness phase of the project. They are referred to at various points during the dissertation and included for completeness.

Question 1

Can you give some examples of where we need to share knowledge in the team?

Name	Answer
M1	Training new staff and developing existing staff.
M2	Cross skilling among team members to ensure we can cope with all requests. Retain knowledge after staff leave team.
D1	Among developers to prevent us repeating work/re-designing solutions.
D2	Staff training, new and existing staff. Spread expertise.
D3	Share lessons learned and hints and tip among developers.
D4	Staff training on how to support the system
D5	Share expertise on system functions.

Question 2

In what ways do we currently share knowledge well within the team?

Name	Answer
M1	Mentoring of new and developing staff helps them to grow and develop. Documentation is created for all projects
M2	Mentoring. Good atmosphere of helping each other. People bouncing ideas off each other. All developers are approachable for help.
D1	Working closely with other developers on various projects, and mentoring.
D2	New Staff are assigned a mentor who helps them learn the necessary skills.
D3	Projects are documented, and documents are accessible when needed. Newer staff are assigned a mentor to work with them initially.
D4	People are approachable and helpful in the team so it is easy to learn from each other.
D5	New staff are assigned a mentor who then allocates time to help with skills development and training.

Question 3

What restrictions are there currently to knowledge sharing in the team?

Name	Answer
M1	I would not be close enough to the team to answer
M2	When developers leave the team it can sometimes leave a knowledge gap in the team.
D1	Various developers are expert in different areas, and generally work on these areas of the system, so it can be hard for others to skill up in that area.
D2	Developers have challenging objectives and deadlines and cant always spare sufficient time for this.
D3	Project deadlines don't always allow time for knowledge transfer.
D4	Some of the experts are not as good at explaining what they know as others.
D5	Experts are not always available due to time pressure.

Question 4

Does knowledge impact on performance?

Name	Answer
M1	Customer service is effected by the speed and quality of our development so yes.
M2	Yes, I find the most productive staff are those with the greatest experience.
D1	Yes because people with knowledge or access to knowledge can complete tasks quicker with less work than those don't have access to that same info.
D2	Sometimes the two are linked
D3	Definitely has an impact
D4	Yes but not always, some people are lazy.
D5	Yes not having access to information can stop me progressing on work items

Question 5

Do we need to make any changes to how knowledge is shared in the team?

Name	Answer
M1	If efficiency's can be gained then these should be investigated.
M2	We need to take advantage of available technology to improve the efficiency of the team.
D1	We should try to make it easier for people to get the information they need.
D2	Better access to/organisation of documentation would be useful.
D3	We should make knowledge more visible to people, for example refocus

	people on creating and searching documentation.
D4	It would be good to have greater access to team knowledge.
D5	We need to help people by making the information they need available instantly.

Question 6

Would you support an initiative to improve knowledge sharing in the team?

Name	**Answer**
M1	Yes I would support a pilot initiative given certain cost restrictions
M2	Yes I would welcome and support this type of initiative
D1	Yes this would seem like a good initiative
D2	As long as it didn't put an additional burden on staff then yes.
D3	Yes
D4	Yes I would appreciate greater access to knowledge in the workplace
D5	Yes

APPENDIX 1.1 – POST IMPLEMENTATION SURVEY.

Below are the questions and answers gathered in the post implementation surveys carried out both among the management group and the developers. The survey was phrased differently to management and developers so their answers are shown together below. This was included as the survey is analysed and referred to in chapter 6 of this dissertation.

Management Survey - Question 1

Has the team performance improved since the introduction of the initiative ?

Name	Answer
M1	In terms of performance measured against closed work items this has been constant, but developers use of the system has been impressive and future usage of the system and team performance will be monitored.
M2	The quantity of work items delivered has not improved significantly however there can be many factors that effect the speed of delivery on these items e.g. Complexity. Overall we feel that this system is adding value and should continue for the foreseeable future.

Management Survey - Question 2

Do you think the system provides good return on investment for the developers time?

Name	Answer
M1	It seems to at the moment but usage will be monitored and reviewed over time to ensure this is the case.
M2	The system is easy to use and learn for staff, and can quickly deliver a result for a developer so it does seem to provide good value to the company.

Management Survey – Question 3

Has collaboration across the team among team members improved as a result of the initiative?

Name	Answer
M1	Developers seem to enjoy using the system as an alternative to meeting notes

	or emails so I think it is helping.
M2	There is still a lot of face to face interaction in this team, but the system does provide an extra facility which people didn't have previously.

Management Survey – Question 4

Has this initiative been a success? Would you recommend and initiative such as this to other team leads/Departments?

Name	Answer
M1	It appears to be a success at present but we will need to monitor it going forward to ensure that early gains are not lost. If the usage thus far is sustained I would definitely recommend its adoption in other teams.
M2	This system has proved very useful to our team and could help to improve efficiency in other teams also I feel. It is simple to use and delivers results so has been a success.

Developers Survey – Question 1

What was good about the knowledge sharing system. Did you add much content and did you use it much?

Name	Answer
D1	It was easy to use, easy to add and retrieve information. I used it most days and added any documents I created since it was rolled out.
D2	This system was easy to use and learn. I added copies of all production support email correspondence.
D3	It was easy to get the hang of and usually returned a useful result to searches. I used it daily added anything I felt as useful, I added to it regularly.
D4	The system was simple to use, and it helped me find relevant information. I used it regularly.
D5	It was easy to use as it felt like other web based systems I had used. It provided a first stop when looking for help. I used it almost every day from once it was available.

Developers Survey – Question 2

Did this system add to or reduce your workload?

Name	Answer

D1	Slightly increased due to the addition of content.
D2	I think the impact is neutral as it can help me find solutions quicker but can take time to add content.
D3	At the outset it took time to find my way around it but now I interact with It very quickly whether it be adding or searching. Overall I feel it provides more benefit that cost
D4	Reduced the time I spend searching for answers in some cases
D5	Has had only a small impact on workload.

Developers Survey – Question 3

Did you find it easy to search? Were the results returned relevant to what you were searching for?

Name	Answer
D1	The system carried out searches fast and usually returned useful results
D2	Searching was easy however we need to ensure contents is well organised and kept up to date.
D3	The search facility is easy and fast. The results if available are usually well structured and relevant.
D4	Searching was easy and I could quickly narrow down the search to the relevant stuff.
D5	Searching was easy and always returned the most suitable content.

Developers Survey – Question 4

What could be done better? What should we add to it?

Name	Answer
D1	Increase the size of the user group to other teams within the dept., that way we could tap into their resources as necessary. Add more content.
D2	The system will become more useful as the amount of content increases. Could we investigate auto upload of data from the mainframe (e.g. Job stats and daily error reports)
D3	The content needs to be kept up to date and well organised. As content increases refine searching to ensure results are relevant
D4	Better quality and quantity of content.
D5	Additional relevant content.

Developers Survey – Question 5

Did this initiative improve collaboration in the team? What could help improve it more?

Name	Answer
D1	It facilitates easier exchange of ideas across the team, so helps in that way. Effective communication in the team is essential for collaboration.
D2	The system helps collaboration within the team. To improve further collaboration training could be provided on further opportunity's to advance collaboration.
D3	Collaboration could be improved by continued adoption of advances in communication technology.
D4	This system did help collaboration in the team. To improve this some informal social activities could help people get to know each other better.
D5	The system did help towards improved collaboration. This could be further improved by building collaboration into peoples objectives.