Lessons from the Document Management Trenches

Susan K. Charles
Global Library and Information Services
HP Laboratories Fort Collins
HPL-2001-230
September 24th , 2001*

document management, knowledge management, EDMS, internal documents, software engineers

This case study examines a document management project implemented to provide storage and accessibility to internal HP software documents. The Kalliope project was undertaken to increase knowledge sharing, collaboration, and communication between software engineers in various project teams and labs at four or more HP locations worldwide. Design, planning, and implementation of the project are described. Emphasis is placed on several factors that affected the success of the project. Important factors considered are: obtaining senior management support, creating a motivation/incentive framework, building the necessary organizational infrastructure, and developing a knowledge sharing culture. This paper features a practical perspective on the "lessons learned" during the implementation of this project.

* Internal Accession Date Only Approved for External Publication
To be published in Online: The Leading Magazine for Information Professionals, January 2002 issue

Lessons from the Document Management Trenches

Susan K. Charles, Information Research Analyst, Hewlett-Packard Company (Fort Collins, CO)

1. Introduction

It has been well documented that the role of the information professional, especially in corporate settings, is evolving rapidly. We are no longer just librarians, the gatekeepers of traditional information collections, or online searchers who skillfully navigate and extract high-value information from complex database systems. Because of our skills in categorizing and finding information many people outside of our profession think of us when they face their own "information" problems. These opportunities may present themselves as complex knowledge management projects.

As an information professional working for a large, high-tech company I was approached with just such an opportunity. An informal contact at our Information Desk with a manager in a large R&D lab developed into a query about our willingness to participate in creating a repository of internal documents written by software engineers in support of their software development activities. This seemed like an unprecedented opportunity for our library to be involved in creating a knowledge access tool that would be of great benefit to many engineers. Furthermore, my experience as a technical researcher, former cataloger, administrator for our online library system, and webmaster seemed custom tailored to just such an undertaking. A discussion with my manager, coupled with a firm belief in the obvious benefit of helping people find and use their own information, convinced me that I was the right person for this job. It was agreed that I would participate in the lab's project as a part-time library consultant. It seemed that I had become an "accidental knowledge manager" [1].

One query at our Information Desk had developed into a primary role in a complex document repository project that was envisioned to capture thousands of documents and serve 1,500 software engineers across multiple labs in at least four geographical locations. As an initial team of one, I was unaware of what was to transpire over the next couple of years. I would like to share my experiences with this project and also provide, what I hope, are valuable insights for any information professional who undertakes a knowledge management project.

2. Why Create a Document Repository?

Where had the initiative to create an internal document repository originated? The software development organization that would use this tool consisted of teams grouped into sections within large labs. These labs, in turn, were organized into large technology centers that were part of the decentralized business structure that Hewlett-Packard was known for at this time. At a leadership development meeting, managers and engineers from the technology centers had identified a number of areas for improvement including communication. The action item associated with communication improvement was to "rent a librarian" to help organize information for better communication, specifically, to "define and create a storage and search mechanism". A caveat was to identify what information was important for sharing.

Engineers and project teams had been using the corporate intranet to build websites to link to documents they needed within their development groups. Software developers are natural users of the Web and they took the initiative of creating these on-the-fly websites. Busy programmers are often immersed in the process of creating and improving code and, unfortunately, the maintenance of these websites was being neglected. As I interviewed engineers and asked them where they kept documents I was pointed to a number of websites, only to find that many document links were dead. Several successful but small Web repositories did exist and I was cognizant of the need to consult with these teams and find out what they needed in a more encompassing tool.

Many documents had been created and stored on numerous lab servers and individual computers. Documents were accessible to the people or groups involved in their creation but there was no cohesive way to locate and share them. If an engineer left a team or the company, documents stored on his or her individual computer would, most likely, disappear. Documents existed in a variety of formats including: plain text, HTML, various word processing formats, and more arcane formats such as Framemaker, TROFF, and Latex. Such a diversity of formats inhibited document sharing.

The labs had already created several tools that were used for some specific knowledge-sharing activities. These included a technical documentation repository supported by a large team of technical writers and programmers, a software defect tracking tool, a peer review database where work progress was monitored, a customer information database, and a website that provided access to software development tool information. There was no standard approach to the implementation of these knowledge resources and metadata, when used, was rudimentary. Ideally, our tool would interact with these existing resources.

In a slide presentation created to elicit senior management support, we listed such impacts as loss of engineering productivity, frustration, rework, diminished product quality and longer lifecycles, high product design costs, customer dissatisfaction and loss of loyalty, potential liability, and loss of sales and revenue. We estimated a very favorable return on investment (ROI) based on engineering costs, expert opinions on the time involved in information seeking activities, and case studies from other companies that had implemented similar projects.

In a 1996 knowledge management case study at HP, it is indicated that HP's decentralized structure could hinder corporate-wide knowledge initiatives [2]. This particular project more readily fit the category of being what Davenport, De Long, and Beers [3] refer to as a "more bottom-up, decentralized" project. I often thought of our project, named Kalliope, as a grassroots effort. Unfortunately, this bottom-up approach resulted in a lack of upper-level management awareness and support, with a subsequent lack of dedicated resources. In the initial stages, I mistakenly assumed that it would be apparent to engineers and management how important this initiative was and that they would readily buy into the need to preserve, find, and share their knowledge. Phil Perkins of Pillsbury calls this attitude the "Field of Dreams" trap and states "Don't assume that if you build it, they will come" [4]. Unknowingly, I was experiencing Peter Dorfman's "mismatch between reality and expectation; goals that make sense in the abstract may be absurd in practice."[1]

3. Getting Started

As a team of one, with a manager as my sponsor, I needed to quickly get a handle on this project. At inception, the timeframe for the project and the resources that would be made available were fairly undefined. The basic directive was to get it done "as fast and as inexpensively as possible". I wrote a detailed project plan that included a project description and goals, and attempted to identify all of the critical components that needed to be considered.

A short version of the initial project description reads "Create a knowledge system which all personnel can use to access, create, and store information in order to enhance knowledge sharing, collaboration, and expedite decision-making. This knowledge system will include all information/documentation which contributes to the knowledge base and will consist of the following major components: 1) information creation, 2) storage, 3) access, 4) retrieval, and 5) information tracking and control. This system will be a real-time, dynamic Intranet tool that is well-organized, and very easy to use and maintain. It will be a critical knowledge collaboration tool." The list of project goals additionally included: furthering the business goals of the labs, meeting the knowledge-sharing needs and behaviors of the users, improving upon current knowledge sharing processes, and providing (computer) platform independent information creation and use.

I undertook a crash course in current document management theory, practices and implementation that included attending a major industry conference and reading Bielawski and Boyle's book, "Electronic Document Management Systems" [5]. I talked to other practitioners who were implementing document management systems. I also felt that it was important to contact people working on similar projects within HP to get their advice, find out about resources and tools, and understand the current, and ongoing, state of document management within the company. There was no one set of tools or processes that could be leveraged at a corporate level to help with a smaller project. There were, however, a number of people with exceptional document management skills that they had gained by working on teams to implement major projects. Several of these have been described in the literature [3].

4. Getting to Know End-Users

According to Stan Lockhart, president of Lockhart Consulting Group, “Knowledge transfer among software developers is always a tenuous proposition” and at one software company “developers met the announcement of a KM initiative with vocal skepticism” [4]. I considered it crucial to talk to the engineers who would be using this tool so that I could understand what they wanted (and needed). After studying the organization charts for the targeted labs, I talked to local engineers face-to-face. I selected people that represented several teams that my sponsor had identified as key groups for creating useful documents. I also looked for engineers who were involved with the existing team Web repositories and for people who seemed to have a genuine interest in this type of project. Beyond gathering information, I wanted to develop some enthusiastic champions who would help their colleagues understand and embrace the project. I had a basic set of questions to ask at interviews but also wanted the engineers to feel free to express their feelings and concerns about finding and sharing their knowledge. These interviews resulted in a wide range of suggestions. Some people understood and supported the broad purpose of the project while others expressed concerns or made specific practical suggestions. Inevitably though, when asked if they would like to be involved at some level with the project, they declined citing a definite lack of time. Everyone acknowledged that despite the need this would be a complicated project to implement. In retrospect, I realize that there was confusion about the management support and commitment for this project. It had also become apparent to me that any knowledge tool was going to require engineers to accommodate a change in work processes and that resistance would exist unless this could be done with minimal disruption. Ease of use was going to be critical for success.

To provide another communication path for the engineers, especially with those working at remote lab locations, I decided to try to enable discussions on the corporate intranet. I set up a project website that included a project description, Q&A, existing documents, a graphical model of the project (FIGURE 1) and a section where I attempted to initiate an interactive discussion of the project. I sent an email message out to a target group of engineers known as technical contributors. The purpose was to involve a high-level group of engineers across all the labs in a discussion of the project and their expectations. We were also looking for project champions at other geographic locations.

The reactions to the email and website were interesting. I was underwhelmed with the number of responses but those engineers who responded had a wide variety of concerns and questions. Opinions ranged from full support to some very critical reactions from people who were wondering why they hadn’t heard about this project from management and who lacked trust in the purpose of the project. The lack of “poor vertical communication” cited by Dorfman as a common stumbling block was becoming apparent [1]. Additionally, I received feedback from a group of engineers that had extensive concerns about usability for sight-impaired software developers. Several respondents had studied the project model description and made point-by-point comments. At least one respondent felt that continuing to use the current approach of linking documents from on-the-fly websites or developing a tool from scratch within HP was preferable to using any third party tool. Kalliope would later provide the capability of allowing end-users to create websites with links to repository folders or documents.

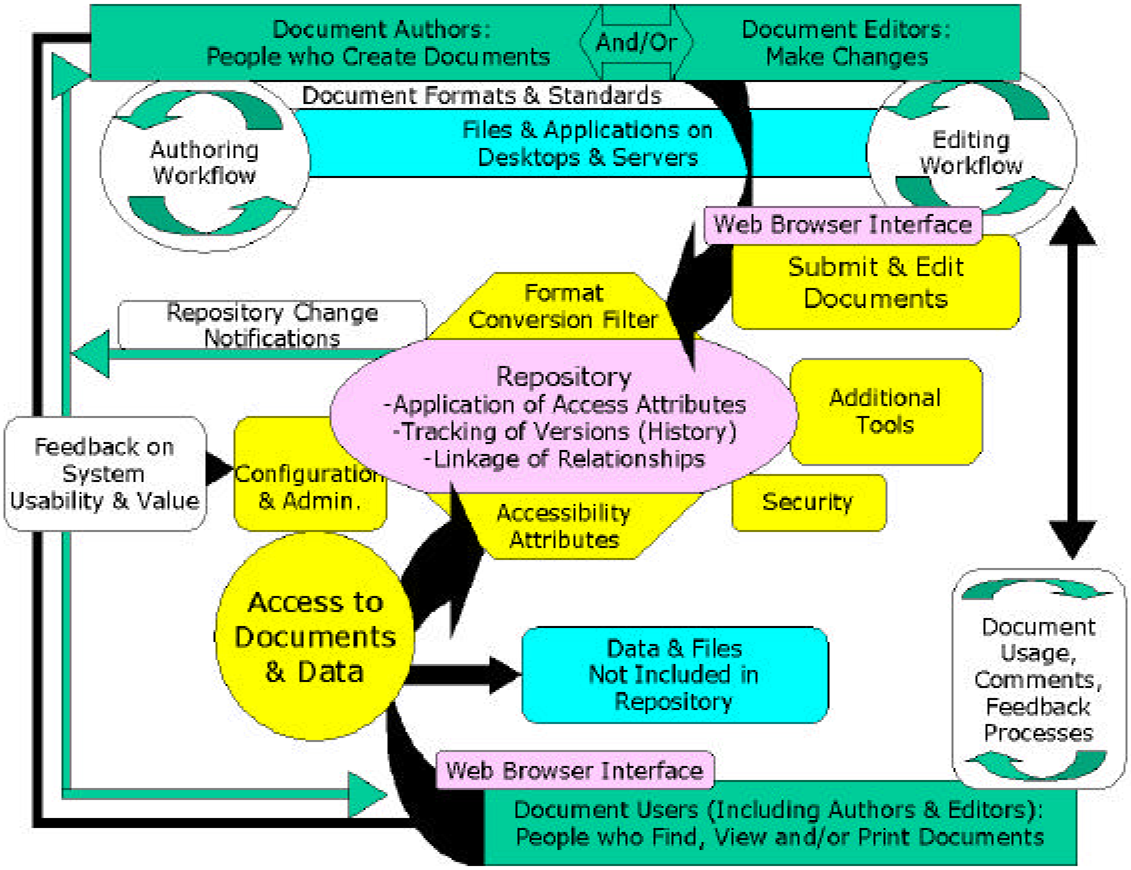
Document Authors:
People who Create Documents
And/Or
Document Editors:
Make Changes
Document Formats & Standards
Authoring Workflow
Files & Applications on Desktops & Servers
Editing Workflow
Web Browser Interface
Submit & Edit Documents
Format Conversion Filter
Repository Change Notifications
Repository
-Application of Access Attributes
-Tracking of Versions (History)
-Linkage of Relationships
Additional Tools
Feedback on System Usability & Value
Configuration & Admin.
Accessibility Attributes
Security
Access to Documents & Data
Data & Files Not Included in Repository
Document Usage, Comments, Feedback Processes
Web Browser Interface
Document Users (Including Authors & Editors):
People who Find, View and/or Print Documents

Figure 1: Preliminary model of the Kalliope document management process.

I was receiving valuable information about my target users but there was the usual less than enthusiastic response about actual project participation. In the article "Trust Matters", Phat X. Chiem states "Building trust begins with creating a shared sense of commitment across the enterprise." [6]. It seemed that I would need to take a different approach to gain trust and project participation.

5. Off and Running

It was becoming apparent that the Kalliope Project required a larger team. My sponsor hired a Business Process Specialist who would work closely with me, eventually take over the project, and administer the repository. My new colleague was enthusiastic, very skilled in computer and Web-based technologies, and we developed an excellent working relationship. It seemed that we might be able to implement this tool after all!

We used our plan to create a shortlist of three electronic document management systems (EDMS) that might be viable alternatives. Vendors were invited on-site to make presentations and answer our questions about their products. We then wrote a detailed Request for Proposal (RFP) to describe our requirements. Critical requirements included the need for easy, cross-platform accessibility so that users could access and update the repository from Unix workstations and MS Windows PCs. Our users would also have a variety of Web browsers and these needed to be supported. The RFP was sent to two vendors that, after the on-site visits, seemed the best fit for our project. We had also explored the possibility of a partnership within HP and an affiliated lab, with an existing HP-customized system, also responded to our RFP. After receiving vendor responses to our RFP we became painfully aware of the realities of how much a system that met our requirements would cost. We needed to reassess our needs and resources.

6. A System is Born

According to Steve Barth, "KM applications ranked last out of 22 categories in terms of user satisfaction" [4]. We were aware that we were trying to implement a complex tool. Potential vendors would tell us that their products could be implemented "out-of-the-box". Collectively, we had some experience with complicated systems, viewed these claims with skepticism, and continued to question the implied ease of implementation. At the time, the document management/KM software industry was young and the transition to Web technology integration was incomplete. We were told that existing products would meet our requirements but we might need to use additional consulting. Consulting fees did not fit into our limited budget and with the hubris of experienced high-tech knowledge workers we decided that we could solve any problems we encountered.

After preparing a presentation for upper management to galvanize support and justify resources we waited. Previously, we had made a contact in one lab where an engineer was implementing a small repository using one of our candidate products. In a twist of fate, this engineer was leaving the company and there was no support structure in place at his remote site for his fledgling repository. He was happy to transfer the product license to us so that we could use the components for our project. This did not negate the need for us to spend money on additional licenses, database resources, and product support but did effectively make our initial costs considerably more appealing to management. After obtaining additional software and licenses we were ready to implement our new tool.

With little formal training in project management and a timeframe of "as soon as possible" we eagerly plunged forward. A difficult struggle to get computer hardware, operating system, and database software operational, even before we installed our new tool, quickly established that we were in for a long haul. Once we started installing and configuring Kalliope, we realized that we were facing proprietary programming languages and customized scripts, a daunting task indeed for a small, part-time team. Schedule slippage can occur during complex projects and we were to experience this first hand. Our efforts to decipher the complexities of our EDMS were punctuated by prolonged head banging interspersed with shouts of joy and relief. We were further frustrated by delays in software upgrades that promised better integration with Web technologies, solutions to bugs, and the fulfillment of requirements detailed in our RFP.

I was also becoming frustrated by those aspects of the project related to understanding, categorizing, indexing, and applying search functionality to the lab documents. It was hard to find a wide range of sample documents. Collections available electronically were often related to very specific aspects of a technology. Many documents were unavailable electronically, lost, or in remote locations. An accurate estimate of the number of documents that were targeted was never available; it was suspected that thousands existed somewhere. No formal standards had been applied across labs to create documents; some samples were very structured while others were simply charts or text with no titles. Despite my expertise in selecting concepts for inclusion in knowledge structure and metadata, I needed some assistance from a subject expert (programmer) to interpret some of the terminology. The lack of time cited by many engineers to avoid involvement left me struggling to define an overall knowledge structure. Our sponsor provided a useful perspective including some preferences for basic structure.

I was quite comfortable with the rigorous attention needed to establish taxonomies that end-users could use to refine their ability to extract documents from a repository. I had been a cataloger and the process of creating structure and controlled vocabularies was second nature to me. A recent Outsell Briefing notes, “A taxonomy provides a powerful way to create structure and access to unstructured electronic information by bringing context to content, thus it is an important piece to an overarching knowledge management system. Taxonomies help bring order to the chaos.” [7]. My sponsors had initially recognized this concept but also wanted our tool to interact with existing knowledge resources some of which employed inadequate vocabularies. From my perspective, the labs needed to implement a cohesive strategy for a taxonomy that was complete, controlled, and applicable across all of their tools. This was an effort far beyond my assigned responsibilities and the time invested in this type of undertaking would have been substantial. I also felt that to successfully implement a useful taxonomy there would need to be an ongoing time commitment to provide permanent guidance, control, and maintenance. This was not a consideration. Furthermore, we were having so many difficulties configuring Kalliope that creating easy-to-use forms to gather metadata and designing a flexible search interface were challenging tasks. The system we had chosen supported full-text searching and this was eventually used as a fallback in place of the creation of a useful taxonomy.

The pressure was on to quickly produce a functional tool, to get practical results, and to gain support from key end-users. Researchers such as Hunsucker and Walheim note engineers often need “concrete demonstrations of the capabilities of a system to help reduce resistance to change” [8]. Our sponsor wanted us to provide a “dog and pony show” to gather feedback. We rushed to get Kalliope functional and prepared a demonstration of a tool that still lacked certain features that I felt were key to gaining support. Critical features, such as a plug-in that enabled end-user access from Unix workstations and broad support for a variety of Web browsers, were not yet available despite being key RFP requirements. Repeated attempts to get support for our Unix functionality were fruitless. We decided to inform end-users of these defects and our attempts to rectify the situation.

Presentations went as smoothly as could be expected with concerns voiced about basic features (including the lack of the Unix plug-in). A number of end-users still supported the project goals and seemed to accept the system with the caveat that several improvements were needed. Our sponsor had reiterated the need to find teams that would test Kalliope and several people expressed interest. We used our feedback to try to improve the tool for a test audience. Additionally, we prepared a project test plan and hosted our presentation online for potential users at other geographic locations. We made several contacts at remote labs to begin testing with target teams and began to prepare online training. Yet we continued to meet "Catch-22" type resistance illustrated by such attitudes as "we would use the system if there was already content in the repository" or "it is a good idea; build it and we'll use it if we like what we see." One team enthusiastically requested a customized mini-repository but when completed this special repository went unused. Unfortunately, the Kalliope team has yet to come close to our initial, ambitious project goals.

7. How Did This Happen?

I felt total empathy when I read Steve Barth's account of the Pillsbury scientist who sent e-mail invitations inviting participation in his KM project and "waited for the knowledge to flow. After six months, he was still waiting" [4]. There were a number of factors involved in Kalliope's languishment. Some obvious issues were: 1) vendor software upgrades continued to be delayed, 2) every upgrade created major system changes resulting in more months of configuration frustration, 3) teams seemed eager but were just too busy, and 4) without certain features volunteers had difficulty using the tool without a struggle. In retrospect, one factor had clear impact. There was just no motivation for engineers to participate and use Kalliope.

Phil Perkins speaks of an experience at Pillsbury, "The fact was that there was no incentive for anyone to invest time and energy" and "the originator of the idea focused on the perceived benefits to the organization without considering what incentives would have to be offered to get people to contribute" [4]. Davenport, De Long, and Beers state, unsuccessful KM projects had "struggled to get organization members to contribute to repositories" and "the motivation to create, share, and use knowledge is an intangible critical success factor for virtually all knowledge management projects" [3]. In the Outsell Briefing "Knowledge Management: It's All About Behavior" four basic elements regarding successful KM cultures are listed including the "use of encouraging behaviors to facilitate more effective contribution/use behaviors" [9]. In this same report Outsell quotes a high-level knowledge executive, "We've set up an elaborate system of performance measures and rewards based on contribution and use, but it comes down to individuals – partners who love to share what they know and lead by example." During our project the only attempt to create an incentive for Kalliope participation was to discuss making document submittal a performance factor in the existing peer review database. Managers would encourage their engineers to use the system but we ultimately had to rely on future participants' perception of Kalliope as a "good and useful" tool.

8. An Overview of Lessons

Recognition of the success factors for KM projects is better understood today than at the inception of the Kalliope project. In summary, I present a list of the elements that I believe left Kalliope, though still in use, relegated to the category presented by Oz and Sosik where "the project has been completed, but the system is not fully used by its intended users" [10]:

- Lack of strong, consistent management support.
- Project scope was too large for the available resources.
- Limited resources (people, money, training) for a knowledge management project; skills had to be learned at the cost of time, alternative solutions were limited due to lack of money.
- Poor communication.
- No preparation for a "knowledge culture" or the incorporation of knowledge processes.
- Lack of a motivation/incentive system.
- Challenging software with poor vendor support (especially for our Unix functionality).
- No real timetable or budget, just "fast" and "cheap".

In their insightful article "Successful Knowledge Management Projects", Davenport, De Long, and Beers list eight major factors that contribute to the success of KM projects. They identify "having a knowledge-oriented culture, creating an organizational infrastructure, finding effective motivational tools, and developing senior management support" as the most critical success factors [3].

Kalliope is operational but has been scaled back. Kalliope's administrator continues to receive queries of interest from individuals in the software labs, indicating the ongoing need for a knowledge repository. It remains to be seen if the issues we encountered can be resolved and the success factors described above can be put in place to allow the full fruition of our original goals.

9. References

[1] Dorfman, Peter. "The Accidental Knowledge Manager." *Knowledge Management*. 4, n.2 (February 2001): pp. 36-41.

[2] Davenport, Thomas H. "Knowledge Management Case Study: Knowledge Management at Hewlett-Packard, Early 1996."

http://www.bus.utexas.edu/kman/hpcase.htm

[3] Davenport, Thomas H., David W. De Long & Michael C. Beers. “Successful Knowledge Management Projects.” *Sloan Management Review*. 39, n.2 (Winter 1998): pp. 43-57.

[4] Barth, Steve. “KM Horror Stories.” *Knowledge Management.* 3, n.10 (October 2000): pp. 36-40.

[5] Bielawski, Larry & Jim Boyle. Electronic Document Management Systems: a User Centered Approach for Creating, Distributing and Managing Online Publications. Upper Saddle River, NJ: Prentice Hall PTR, 1997. 332pp.

[6] Chiem, Phat X. “Trust Matters.” *Knowledge Management.* 4, n.5 (May 2001): pp. 50-55.

[7] Outsell, Inc. “Taxonomies: Structuring Today’s Knowledge Management Systems.” *Information About Information Briefing*. 4, n.26 (July 18, 2001): pp. 1-18.

[8] Hunsucker, John L. & Rex J. Walheim. “Work System Change in Technical Organizations: Problems and Prospects for Avoiding Them.” *IEEE Transactions on Engineering Management.* 39, n.4 (November 1992): pp. 303-311.

[9] Outsell, Inc. “Knowledge Management: It’s All About Behavior.” *Information About Information Briefing.* 4, n.3 (January 26, 2001): pp. 1-16.

[10] Oz, Effy & John J. Sosik. “Why Information Systems Projects are Abandoned: A Leadership and Communication Theory and Exploratory Study.” *Journal of Computer Information Systems*. 41, n.1 (Fall 2000): pp. 66-78.