



## Knowledge Management for Software Projects

Bruce Taylor

### Introduction

What is the most important product of your software development project, after the working software itself? If you'll reflect for a moment, you'll realize that the knowledge produced during the project is precious, because it provides guidance for maintaining and extending the product, and for planning follow-on projects. But in most organizations, this knowledge is discarded, or lives on only in organizational tradition and folklore. How can you effectively capture, organize, and retain this information? The most effective way to capture the information is to use communication tools that are appropriate for the situation.

### Email

Email has been around for a long time, and virtually everyone knows how to use it effectively. In the software development world, it is a great way to ask longer questions that don't require immediate answers: "We're getting lock contention on the employee table. Do you have any idea what's going on?"

But for knowledge management, email has several drawbacks. First, individual's email folders are not very reliable repositories; if your email system provides automatic archiving you should certainly use it. But a more serious problem is the low signal-to-noise ratio of email: only a very small percentage of a developer's email contains information worth preserving in an archive. Depending on the so-

phistication of your email system, you might be able to use filters to select messages with interesting keywords and copy them to the archive. Otherwise, you will have to rely on developers to archive important messages.

### Bulletin Boards

Bulletin board systems solve many of the problems of email for knowledge management because they are, by nature, well structured and persistent. The threaded discussion model of most bulletin boards is a natural fit for conversations about design issues, or debugging techniques, or any other topic that requires extended collaboration among a team of developers. On the negative side, bulletin boards are less easy to use than is email, and developers may have to get in the habit of checking the bulletin board regularly for new threads or replies to existing threads. Newer bulletin boards systems solve this problem by allowing developers to register for interesting topics, and some integrate with email, so that developers can carry on conversations in either format. If your bulletin board system has a sophisticated search capability, you can use it to answer interesting historical questions like, "Why in the world did we decide to put the employee's address in a separate table?"

### Wikis

Wikis are a relatively new mechanism for knowledge management. A Wiki is an unstructured collection of document pages

that can be created and edited by any member of the Wiki community. You can see an example of a well-run Wiki at <http://www.wikipedia.com/>. Wikis are ideal for capturing and storing information that changes over time, but not very rapidly: design and coding standards, architectural overviews, interface descriptions, and the like.

Wikis have the great advantage that they are not pre-structured and their organization can change to accommodate the changing needs of your project. But this strength can also be a weakness; Since Wikis are only loosely structured, they can degenerate into a chaotic jumble if the community doesn't follow a common protocol of creating, formatting, and linking pages. Most successful Wikis rely on a part-time administrator to keep their structure consistent.

## Document Repositories

Document repositories are the oldest form of knowledge management within the development community. In their simplest form, they can be a set of folders in a well-known location where developers can copy important documentation, meeting minutes, but lists, or any other data. Document repositories are cheap and easy to implement, and often work well enough as long as the volume and complexity of the data is low.

But the fact that document repositories have a static structure makes them very difficult to maintain. If a repository does not have a diligent administrator, it can rapidly degenerate into a random, chaotic collection in which documents disappear never to be seen again.

## Search Engines

No matter what form of knowledge storage you choose, it will work better if you add advanced searching to it. Although email systems, bulletin boards, and Wikis all provide title search capabilities, they don't provide the ability to look for keywords in the body of the documents, or to rank documents by usefulness. Really, what you want is a familiar search capability similar to Google or Yahoo!.

Luckily, you can easily add search capabilities to your knowledge base through freeware search engines like Swish-E (<http://www.swish-e.org>), or hardware like the Google search appliance (<http://www.google.com/enterprise>) With a search engine that can locate and catalog files, and spider an intranet, you are well on your way to providing effective knowledge management.

## About Bruce Taylor



*Bruce Taylor is the principal of CoachingProgrammers.com, an executive coaching firm located near Boston, Massachusetts. Bruce helps software organizations of all sizes to create low-stress, supportive, adaptable working environments, so that the engineers, leaders, and managers can work as effectively as possible. He provides executive coaching for senior managers who are creating superior organizations, management coaching for technical leaders who are adapting to new agile practices, and individual coaching for engineers who are upgrading their skills. Bruce has a Masters in Computer Science from Duke University, a Masters in Community Psychology, and a Certificate in Job Stress and Healthy Workplace Design, both from the University of Massachusetts.*