# Wiki Customization to Resolve Management Issues in Distributed Software Projects

Jamie Hohman
*Garmin International*

Dr. Hossein Saiedian
*University of Kansas*

*This article introduces methods of wiki modification that can assist software project managers in addressing the communication problems associated with distributed projects. It outlines some of the best practices from industrial projects that employ distributed teams and discusses how such practices can be implemented via the wiki processes.*

Software development projects are constantly becoming more complex. As customers continue to shift more of the risk to the development organization, project managers search for ways to mitigate this risk while still providing the requested functionality in a timely manner. Many managers increase the size and capability of their development organization by expanding teams across sites and even across continents. These distributed project teams introduce a new set of challenges to traditional project management.

Most of these challenges involve the level of communication that grows non-linearly compared to the increase in project team size. The addition of geographically remote teams that may span time zones, cultures, and languages complicates communication even further. Effective tools are necessary to make this communication overhead manageable.

One communication tool which has been around for over a decade is a wiki. Wiki means *quick* in Hawaiian and consists of two things: 1) a program that makes editing Web pages extremely quick and easy, and 2) a philosophy regarding how users should go about that editing [1]. The wiki way philosophy allows any user to edit any page created by another user with the belief that fixing incorrect content is easier than painstakingly ensuring content is always correct the first time. Companies as diverse as Motorola and SAP are already using wikis in their day-to-day business operations[1]. Furthermore, there is already an annual international symposium on wikis and in particular, their use in software engineering and management[2]. The site highlights a quote by Shashi Seth, senior product manager at Google: "This company runs on wikis," referring to the uses of wiki tools such as GooWiki TWiki, and Sparrow.

This article focuses on the use of wikis in project management and, in particular, as a tool for project development oversight and as a communication tool by project management personnel and development team leaders.

Wikis have been in use for some time as a project management tool, but the very freedom and flexibility that make them powerful also make them difficult for large scale use. With some simple guidelines and modifications, this power can be harnessed. For a wiki to serve as an effective central knowledge repository, it must be easily searchable. Adding semantic information in the form of metadata tags is important.

The most powerful aspect of wikis is their flexibility, but project management and software development processes require some level of discipline. Adding light constraints to the wiki helps provide guidance for team members and improves monitoring of project status by management.

## Distributed Project Management Challenges and Current Solutions

### *Software Project History and Evolution*

As steady hardware development has produced more machine capability, solutions to more complex problems are desired that utilize this increased hardware capability. These complex solutions tend to exceed the customers' technical knowledge, so their focus shifts from requirements to objective capability. Projects without clear, precise requirements increase the risk for the developer. The increased complexity of the solution often requires development of a system of systems (i.e., an enterprise) rather than one isolated system. Ensuring that these systems all communicate and operate together correctly increases the technical complexity of whatever solution is designed. To design and build these large, complex solutions in a timeframe required by the customer, the project manager must often utilize strategic teaming, outsourcing, and acquisitions [2].

Just as the software development community was starting to come to grips with the *software crisis* of the 1980s and early 1990s, the Internet opened previously unheard of communication channels. These new channels were quickly leveraged to build new distributed systems that created their own unique set of management challenges. Development teams soon realized this new, unique form of communication could also be utilized to create distributed project teams with significant advantages over traditional teams as shown in the following [2]:

1. Creating multiple, independently located teams results in a larger workforce than can be contained in any single location.
2. With teams around the world, time zone differences can be utilized to increase the number of productive hours during the day. Scarce or expensive machines (or other non-human resources) can be utilized at a higher rate. Artifacts produced by one team can be consumed by another team within the same day rather than waiting until the next day.
3. Knowledge experts and other specialized resources can be utilized regardless of where they reside.
4. Less expensive labor can be utilized in some geographic regions without experiencing a drop-off in expertise or productivity.

Since team members of distributed projects are no longer necessarily part of one large, homogenous group, some new development issues arise [2]:

1. It becomes difficult to leverage and track available resources. Even when a functioning system is in place, it can be time consuming to maintain.
2. Different organizations have different processes and capabilities so estimating effort and synchronizing work between sites is difficult.
3. The Internet allows communication and collaboration at levels never experienced before, but this communication is always less rich than face-to-

face communication, increasing the risk of miscommunication. The simplicity of Internet communication, however, also increases the volume of communication between team members which can lead to overload.

4. Different sites have different cultures so soft skills of employees and managers must be changed to conform to one standard of behaviors, processes, and technologies. Trust must be established at these different sites in the new processes as well as with other team members with whom they have not worked before.
5. Transmitting all of this knowledge over an inherently open system requires some additional security policies to protect the confidentiality and privacy of customer information.

A combination of the increased size and complexity of the system to be developed, as well as the new challenges listed that come with managing a distributed project, may make it extremely difficult for a project manager to control and coordinate all of the activities associated with modern software development projects. This necessitates a shift in his/her work away from the role of a benevolent dictator in a top-down hierarchical structure and develop new skills as an orchestrator of interconnected relationships [2]. Wikis are helpful tools for this orchestration. In fact, wikis provide such a great degree of openness, tolerance, and empowerment of the individual, that the previous dictatorship model would be nearly impossible to achieve. By their very nature, wikis delegate control and promote cooperation based on trust [1], forcing a more democratic approach to project management that is necessary in a distributed environment.

### *Current Distributed Project Management Methods*

Current research in the field of distributed project management has outlined some key areas of focus that should be addressed to lead to more successful distributed projects. First, the analysis and effort that goes into creating project artifacts (such as effort estimation, resource allocation charts, and decision rationale) is as important to keep as the artifact itself. Next, the communication between team members during the design and development phases must be stored and available to everyone on the teams for future reference. Also, management and other team members must have visibility into the status and progress of development activities. This allows peer review at all levels as well as more accurate progress tracking. Every project should also have an electronic project repository where all relevant project information is stored to eliminate data re-entry costs, eliminate duplication and information loss, standardize work processes, and increase management's ability to efficiently retrieve accurate information leading to greater management control [3]. A collaborative project management architecture that could be implemented using a wiki is also proposed.

A suggested number of practices useful in increasing collaboration during distributed projects include synchronization of main milestones (more granular is unnecessary), frequent deliveries to create visibility in the project, establishment of peer-to-peer links to eliminate management bottlenecks, problem solving practices (especially at the project planning phase), informing and monitoring practices for management and employees, and relationship building practices to establish trust between teams [4].

### *Wiki Usage in Collocated Project Management*

Wikis are already in use in project management roles, particularly in the open source community. The flexible nature of wikis allows use across the project management activity spectrum including project selection, cost estimation, effort estimation, scheduling, and progress tracking. They are also used, among others, for the following activity-independent tasks [5]:

- Creation and storage of deliverables.
- Meeting agendas and outcomes.
- Status reports.
- *Great ideas* saved for later.
- Standard and best practices.
- New and innovative methods and processes.
- Recording claims for usage of shared machines and other resources.
- Vacation schedules[3].

Current research shows that these wikis are standard, completely open repositories with little in the way of guidance for the users. While it has been demonstrated that wikis are sustainable and increases are shown in contributions and accesses as they mature, the true effectiveness of wikis will be demonstrated as greater numbers of users are obtained [5]. It has also been shown that a dedicated administrator will decrease the apparent chaos of the wiki and lead to greater organization [6]. They also believe that user training is essential for a well-structured, organized wiki. If a wiki has support from a methodology advocate and from upper management, the collaborators will be more effective in creating and using content. As will be shown in later sections, light constraints imposed on the wiki will allow user training and methodology conformance to occur informally and continuously.

## Wiki Modification: Recommended Practices

Standard usage of wikis is already common in project management. The ad-hoc nature of wiki development is one of the most powerful aspects of wiki usage. When supplemented by a few standard project management wiki items, the wiki becomes even more useful.

### Rapid Feedback Areas

The project management wiki should provide areas for quick turnaround on evaluation of work products, including integration and testing reports, code reviews, and documentation approval. This keeps the process and artifact clear in the developer's mind and provides enhanced motivation. Both management and peers provide this evaluation of the work products. As the manager sees that the developer is doing good work (via these evaluations), the manager starts to trust and respect the developer's abilities [4]. Peer evaluations breed trust between co-workers, which is difficult to develop in a distributed project environment; efforts should be made to enhance trust whenever possible.

### Peer-to-Peer Links

Peer-to-peer links should be established so each level of the project teams know who to contact on remote teams and what is the easiest method of contact. These links should be established at three levels: upper management, project managers, and developers [4]. As project teams grow, proper communication of all relevant knowledge becomes more burdensome. One way to help ease this burden is to eliminate redundant and misdirected communication. If the message gets to the right person the first time, communication burdens are eased.

### Role Descriptions

The wiki should include role descriptions for enhanced communication between personnel that are not necessarily peers. A vital role here is *problem solver*. A problem solver is a highly experienced team member who either knows the answer to the problem or knows where to find it. There should be one on each team, and everyone on other teams should know who that person is. If this role goes unfulfilled, team members may spend more time trying to find someone to help them on a remote team than they would actually solving the problem [4]. These types of problems can build to disastrous proportions at the end of projects when time constraints are most critical. It is also important for project managers to know who the designated problem solver is on the team because answering these questions can take a large amount of this individual's time. The project manager should schedule accordingly.

### High-Level Reports

High-level project status and progress reports should be included on the wiki. This provides a window for both upper management and developers to recognize priorities and gain motivation from a project that is scheduled well. If the communication process in other areas breaks down, these high-level reports can serve as a backup so that remote teams have a chance to learn of changes or decisions made to a project. They can also serve notice that artifacts required for further project work are now available [4]. For example, if a development team is waiting on a requirements specification from a business analyst to begin their development work, the development team would be able to see that the business analyst has posted his work on his wiki page and they can begin working on it sooner.

### Face List

To further build the trust level between remote teams, face lists and organization charts should be posted. Knowing who someone is, what their role is in the organization, and receiving feedback from that person creates trust and eliminates the feeling that remote team members are unknown and easily disregarded partners [4]. This also encourages more targeted communication, which eases the burden on project managers and problem solvers who often become communication bottlenecks. It must be noted that not all organizations, especially those within the military and intelligence communities, would want to provide as much detail and identifying information for their personnel and organizational structures.

### Project Dictionary

A project dictionary where key terms, jargon, and methodology are defined is useful. This benefits both developers (for enhanced clarity) and other stakeholders who may not possess as much technical knowledge (i.e., management, sales, etc.) [3]. As this dictionary matures it will provide project context, background, and boundaries that would otherwise be lost as memories fade.

### Business Rules and Policies

Project members should post a list of business rules and policies that are not specified in more formal documents or a link to these documents. This leads to enhanced standardization across all sites and quicker adaptation by new team members [3].

## *Semantic Wikis*

The combination of ad-hoc information contributed by team members and the project management information contributed by leadership produces a large amount of data that most readers have neither the ability nor the motivation to search through while trying to complete their day-to-day work. Adding semantic data to wiki entries can make this process easier.

Adding metadata to wiki pages is most effective when the granularity of the metadata is chosen carefully. All of the pages should include a vertical metadata tag, a horizontal metadata tag, or both. The vertical tag would indicate a *describes-feature* relationship. For example, the notes from a meeting that planned the overall database schema of a project would have a describes-feature property that relates to database schema. The horizontal tag would indicate what step in the project management plan this document pertains to, or a *plan-level* tag. For example, a Gantt chart that shows the resource allocation plan would have a plan-level tag that relates to scheduling. The schema planning meeting notes described above would also have a *plan-level* tag that relates to architecture planning. One flavor of this idea is in use in Wikipedia where typed links are highly encouraged. Typed links means that the author gives information about the link while creating it. For example, when creating a link from Germany to Berlin, the link is typed as: *is-capital-of*.

The second level of granularity for metadata tags would be chosen by the individual page authors conforming to the wiki way. As a team works on designing or implementing a certain feature, the team members decide what aspects of their documents in the wiki they want to highlight and add metadata tags consistently and appropriately. Good communication would allow ideas to naturally cross-pollinate between teams as best practices emerge. This would lead to some level of uniformity across the organization.

One important aspect of adding semantic data to wikis is that these additions should *always be left as optional*. One aspect of what makes wikis powerful is the naturally low entry barrier. Any requirements that raise this barrier will discourage participation and should be rejected [7].

As all of this metadata is collected in the wiki, it must be able to be queried or it will not be useful. Simple queries that use the suggested describes-feature and plan-level tags should be readily available. Another mechanism that evaluates arbitrarily complex queries should also be available for the advanced user. Two other methods to easily formulate queries of medium complexity are provided in [7]. One is keeping track of the links the user traverses during normal navigation and prompting with these types when entering the query. The other is taking feedback from the user concerning query results to further refine the search.

## *Light Constraints on Wikis*

A method of adding useful structure to a wiki without constricting the features that make wikis so powerful, namely the low entry barrier and tremendous flexibility is outlined in [8]. Even though wikis are tremendously flexible, it is generally desirable to have a structure that is enforced throughout the wiki to maintain a consistent look and feel throughout the wiki as well as forcing similar pages to have similar content and structure. In the case of a project management wiki, the management process may require certain sections and content, for example, a section entitled "Feasibility Study" or one entitled "Nominal Costs," or one entitled "Effort Multipliers/Drivers" to be included in all cost estimation pages. In traditional wikis, the maintenance of this structure is accomplished through copy-and-paste creation, manual refinement, and checking by WikiGnomes (editors who work quietly to fix minor nuisances). To begin to automate this process, a proposed set of rules enforced by the wiki engine points out to editors and viewers when something about the content of the page is unacceptable [8]. This can be something as simple as violating the HTML or the company-wide cascading style sheet (CSS) rules, or introducing an acronym for the first time without providing appropriate references for it. These rules are guidelines that the wiki page ought to satisfy. They are generally identified by recognizing properties that pages of the same type should share.

Complete freedom from editing is retained in two ways. First, any page can be saved and viewed at any time, even if it does not abide by all of the rules set forth for that page. However, the rules broken will be illustrated on that page at all times, both to the editor and the viewer. If an editor attempts to save a page that is breaking these rules, the system should issue a prompt before allowing the save to verify that these rules are going unfulfilled. Secondly, any of these rules can be added, modified, or deleted at any time. If so chosen, the wiki administrator could set up a group of *tailors*. The tailors would be the only users allowed to edit the rules (e.g., the CSS rules). This would provide for heavier constraints which may be desirable in a specific wiki.

Adding light constraints to wiki entries helps the project manager in multiple ways while still keeping the traditional benefits of the wiki way:

1. Light constraints allow the project management authority to specify and modify standard formats for required project artifacts that are stored in the wiki. If these standards change, the project managers need to only make these changes in one place, and the people responsible for these artifacts will automatically see that they need to be updated.
2. Light constraints allow the author of the artifacts to easily see what aspect of their artifact is unacceptable, yet still be able to save it and work on it later. This removes many of the major inconsistencies between artifacts developed by different people while alleviating the burden of standards enforcement from project managers.
3. These constraints allow the project manager to view the status of required artifacts. This will almost be in real time. When a standard format must be modified, the status of every artifact that uses this standard will automatically be updated by the wiki add-in software that enforces the light constraints..
4. These constraints should be tailored so that each page is required to have appropriate metadata tags as described previously. Without having some way to remind authors that these tags are important, it would be easy to overlook their addition and maintenance.

Light constraints are already in use in practice to help spell-check wiki content. Two examples of this are tools named MoinMoin and DokuWiki. Also, a project called Miki is attempting to make an infrastructure for collaborative authoring of formalized mathematics. Light constraints are used to encode consistency of mathematical concepts [8].

### *Motivations for Wiki Use*

Even if a first-class wiki is designed, if it is not accepted by the project team or is viewed as potential overhead that takes time away from core project work, the community will not build it and the design effort is not worthwhile. Team members must be presented with factors such as the following to motivate them to use the wiki:

1. Encourage posting of vacation schedules, best practices, innovative ideas, great ideas saved for later, etc. Each team member could make their own section and personalize it with information about that person in which the rest of the team would be interested.
2. Archive absolutely *all* information. Include meeting agendas, meeting notes, formal status reports, all information that led to a decision that was made, and real-time capturing of meeting information. Eventually, memories fade and important project-related information will be lost.
3. Project authorities must use wikis, encourage others to use them, and demonstrate successes. Leaders should point out to team members that if the requested information was on the wiki, everyone's communication overhead would be lessened. As more content is added, the utility of the wiki will become more pronounced, starting a cycle of adding more useful content to a useful system.
4. Recognize personality types who are *adders* versus those who are *synthesizers* and encourage each person to pursue that direction. Adders are utilitarian and are more concerned with efficiently finishing the work by adding content rather than their reputation or task novelty. Synthesizers are more interested in impacting their tasks by finding a novel solution and enhancing their reputation by synthesizing already existing content [5].
5. Provide a mobile, device-friendly interface. Team members on distributed projects travel frequently and richly enabled mobile devices are becoming common. The downtime provided while traveling (such as at airports and in hotel rooms) provides a great opportunity to catch up with the status of other areas of the project that they may not be familiar with. Formal documentation does not translate to mobile devices easily and it is often out of date.

> *"High-level project status and progress reports should be included on the wiki. This provides a window for both upper management and developers to recognize priorities and gain motivation from a project that is scheduled well."*

## Wikis Serve a Distributed Project Management Need

Clearly, the manner in which current distributed project management is approached can be improved. Most of these deficiencies center around the challenges of effective communication between geographically separated project teams. Many project managers seek communication systems that support real-time, multimedia collaboration. This is not useful for developers who generally work alone. Also, as teams spread across continents and even across multiple continents, time zone differences between team members often make real-time collaboration unwieldy. The use of wikis as project management tools resolves many of these issues.

Impromptu, casual communication (such as e-mail, telephone calls, and Internet chat) is critical and can generate great amounts of high-value information, but that information is lost as soon as the telephone call or chat session is ended. That information is also never successfully disseminated across the project team if it remains in one person's private e-mail box or chat log. Semantic wikis help organize this extreme amount of information and facilitate searching it so more knowledge is present and readily available to those who need it. Oftentimes, knowledge management systems and expert systems provide this service. Unfortunately, they are generally costly to implement and maintain, have a high entry barrier for first-time users, and are not well suited to span geographically remote locations. None of these restrictions exist in wikis.

The flexible nature of wikis allows project managers to shape and monitor the flow of information among distributed team members (for example, for the presence of certain page headings, preamble, or prologue information, in a particular color and font size, and within a predefined line length that would enhance readability and comprehension). Adding light constraints to the system will automate some of this monitoring without restricting the flow. It will also increase conformity to standards as the wiki can demonstrate where a critical document is insufficient.

Faster problem solving through both wiki content about the project as well as information posted on the wiki about whom to contact on remote teams will alleviate what is often a major communication bottleneck in project work. It will also alleviate work for the designated problem solver on each team who is usually an experienced team member and whose time is very valuable. As problems are solved more efficiently across teams, greater trust and reliance will be created between teams [8, 9, 10].

## Conclusion

This article has outlined a set of useful pages that should be included in distributed project management wikis by leadership and grown by the team. They include areas used to communicate quickly between developers and management, areas used to communicate between peers on distributed projects to increase trust and productivity, and areas that provide increased visibility on project status and progress to all stakeholders. Adding semantic metadata in both the vertical and horizontal direction to this large repository helps to efficiently search and maintain it. Light constraints on the wiki aids project managers in their quest for conformity (when necessary) and status monitoring without increasing the entry barrier to the wiki and, thereby, discouraging use. Like other communication tools, modified wikis will not be adopted by the team at-large without concerted management support and demonstrated effectiveness. Once this buy-in is achieved, the wiki will organically grow and mature into an indispensable tool.◆

## References

1. Louridas, P. “Using Wikis in Software Development.” IEEE Software Mar./Apr. 2006: 88-91.
2. Nidiffer, K.E., and D. Dolan. “Evolving Distributed Project Management.” IEEE Software Sept./Oct. 2005: 63-72.
3. Chen, F., J.F. Nunamaker, N.C. Romano, and R.O. Briggs. A Collaborative Project Management Architecture. Proc. of the 37th Annual Hawaii International Conference on System Sciences, 2003: 12-24.
4. Paasivaara, M., and C. Lassenius. “Collaboration Practices In Global Inter-Organizational Software Development Projects.” Software Process: Improvement and Practice Sept. 2004: 183-199.
5. Majchrzak, A., C. Wagner, and D. Yates. Corporate Wiki Users: Results of a Survey. Proc. of the Association of Computing Machinery (ACM) International Symposium on Wikis. Odense, Denmark: 2006.
6. Raygan, R.E., and D.G. Green. Internet Collaboration: Twiki. Proc. of the 2002 Institute for Electrical and Electronic Engineers Southeast Conference. Louisville, KY.
7. Kiesel, M., and L. Sauermann. “Towards Semantic Desktop Wikis.” Upgrade: The European Journal for the Informatics Professional 6.6 (2005): 30-34.
8. Di Iorio, A., and S. Zacchiroli. Constrained Wiki: An Oxymoron? Proc. of the ACM International Symposium on Wikis, Odense, Denmark: 2006.
9. Da Lio, E., L. Fraboni, and T. Leo. Twiki-Based Facilitation In a Newly Formed Academic Community of Practice. Proc. of the ACM International Symposium on Wikis, San Diego, CA: 2005.
10. Haake, A., S. Lukosh, and T. Schummer. Wiki Templates: Adding Structure Support to Wikis On Demand. Proc. of the ACM International Symposium on Wikis, San Diego, CA: 2005.

## Notes

1. See <http://twiki.org> for details.
2. See <www.wikisym.org>.
3. This may affect privacy concerns for some employees. It is a suggestion only for organization where employees do not object to publishing their schedules.

## About the Authors

**Jamie Hohman** works as a software engineer for Garmin International and has previously worked as a software development consultant in Kansas City, Kansas. Hohman is currently pursuing his master’s degree in computer science from the University of Kansas. He received his bachelor’s degree in computer science from Kansas State University in 1997.

**Garmin International, Inc.**
**1200 E 151st ST**
**Olathe, KS 66062**
**Phone: (913) 397-8200**
**E-mail: jamie.hohman@garmin.com**

**Hossein Saiedian, Ph.D.,** currently is a professor of software engineering at the University of Kansas. His primary area of research is software engineering and, in particular, models for quality software development both technical and managerial. Saiedian has published more than 100 articles on a variety of topics on software engineering and computer science. His past research has been supported by the National Science Foundation as well as regional organizations. Saiedian is a senior member of the Institute for Electrical and Electronic Engineers. Saiedian holds a doctorate degree in computing and information sciences from Kansas State University.

**University of Kansas**
**Department of EECS**
**Lawrence, KS 66045**
**Phone: (913) 864-8812**
**Fax: (913) 897-8682**
**E-mail: saiedian@ku.edu**