

Table of Contents

1. Introduction	P. 3
2. Collaboration Models	P. 3
3. Scenarios: Collaborative Projects	P. 5
3.1 Scenario 1 - Dynamic Team Composition	P. 5
3.2 Scenario 2 – Knowledge across Multiple Groups	P. 5 - 7
4. Methodology: Adopting and Adapting Confluence	P. 7 - 9
5. Conclusion	P. 9
6. References	P. 10

1. Introduction - What is a *Wiki*?

Derived from the Hawaiian expression for 'quick', a *wiki* is a type of website, which allows users to easily add, remove, or edit content, sometimes without the need for registration. This ease of interaction and operation makes a wiki an effective tool for collaborative authoring (Wikipedia 2006).

In essence, a wiki simplifies the process of creating HTML pages, combined with a versioning feature that records changes to pages over time, so that pages can be reverted to their previous states. A wiki system may also include various tools, designed to provide users with an easy way to monitor the constantly changing state of the wiki as well as a place to discuss and resolve the many inevitable issues (such as disagreements over the wiki content). However, despite their potential as an effective collaborative tool, most wiki systems are not enterprise ready. Many do not have the capability to support multiple teams and projects. They usually cannot support rich content, they are not extensible, they do not integrate well with other applications, and perhaps most importantly, they do not consider how collaborative authorship works as a process.

This white paper discusses an advanced enterprise strength wiki system - *Atlassian Confluence*. It will explore the collaboration possibilities that are built into Atlassian Confluence and show how this was done by adding 'groupware' features, which make this wiki unique. It will also highlight the process of applying wiki technology in order to support, team-based, knowledge intensive collaborative projects, and it will introduce some basic features of Atlassian Confluence. Finally, an overview of a simple methodology for wiki development and evolution, showing how various advanced features in Atlassian Confluence can be used to support the process of collaborative authorship in organisations will be highlighted.

2. Collaboration Models

In this section we describe various forms of computer-mediated communication and the degree to which these technologies can support groups who are separated by geographical distance and varying time zones. We emphasise the potential of wiki technologies in team support.

There are four types of computer-mediated communication technologies; those that provide communication at the same time or at different times, at the same place or at different places. These options are presented in *Figure 1*. While web-based technologies support asynchronous, remote communications needed by teams they do not necessarily support this communication effectively. When we think about how to support communication within and between groups, we can use a framework similar to *Figure 2*. In this diagram, participants are team members, and the term Artefacts refers to the content we want to create and manage. There are three kinds of communication commonly associated with computer supported cooperative work or groupware applications, *Figure 2*. Communication between Participants can be used to act *upon* artefacts (Control and Feedback).

	Same place co-located	Different place remote
Same time synchronous	Face-to-face conversation	Telephone Teleconference Fax
Different time asynchronous	Post-it-note	Letter Email Voicemail Most Web Technologies

Figure 1: Time/space matrix
(Modified: Dix et al 1998, 465)

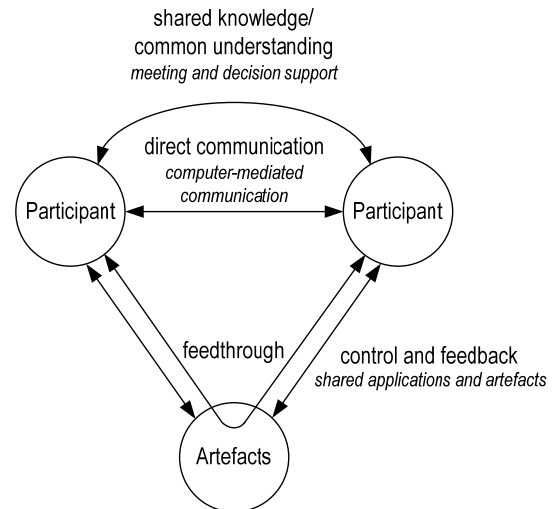


Figure 2: Cooperative work framework
(Modified: Dix et al 1998, p. 495)

Communication can also be *about* these artefacts (Direct Communication). A third kind of communication is referred to as 'feedthrough'. This occurs when a Participant becomes aware of the actions of others who are adding, deleting or modifying the Artefacts- this is communicating *through* the artefact. Most organisations try to support team collaboration using traditional technologies like e-mail and document repositories. Using *Figure 2* we can see why most of these technologies are not successful. Email does not promote good Control and Feedback, while word processing technologies support only limited forms of communication about Artefacts that do not scale up to large teams with rapidly changing membership.

Wiki technology supports a cooperative framework and enables Artefacts to be collaboratively developed, promoting shared knowledge and understanding. Control and Feedback is facilitated by allowing team members to easily add notes and comments. 'Work-in-progress' documents can be improved and refined over time, while maintaining 'accountability' in the form of full versioning. Full versioning enables document changes to be progressively undone allowing previous versions to be recovered. This is particularly useful when teams are collaboratively developing large and complex documents. While it is possible to find these features in other standalone wiki products or in portal environments that have wiki modules, part of the reason Confluence is described as an 'enterprise strength' wiki is that Atlassian have paid considerable attention to how collaborative authorship using wiki technologies should work as a process. In the next section we show how wiki technology can be used to support team-based, knowledge intensive, collaborative projects, and introduce some 'enterprise strength' wiki features in Atlassian Confluence.

3. Scenarios: Collaborative Projects

In complex and agile organisations, work groups are faced with continually changing situations. This continuous change in the work environment affects how team members interact, collaborate, share knowledge and ultimately how projects are delivered. As a result, individuals who are separated by geographical distance and varying time zones, whose skills and services are in high demand and as a result, belong to multiple groups, will require and benefit the most from the support, which collaborative wiki systems offer.

Scenario 1 – Dynamic Team Composition

Teams are often project-oriented and therefore environment-driven. *Figure 2* shows the situation for a single project where team membership is stable. The reality of collaborative projects is that team members will be reassigned, there will be a loss of experience in the team, and there will be the need to train and orient new team members. Despite this volatility, teams must nevertheless still achieve their goals and project deliverables.

Scenario 2 – Knowledge across Multiple Groups

Specialists, whose expertise must be shared across multiple projects, will also require collaborative groupware support. They need to know specific information that can quickly orient them to the current status of their newly assigned project. In addition, they will need to provide other team members with various deliverables, artefacts, or services that make teams more self-sufficient and self-reliant. This in turn enables them to concentrate on more important matters.

Wiki technologies are a great solution to the above scenarios because they can assist by allowing shared knowledge and common understandings to be represented in various document forms, such as project plans, resources, media, and guidelines. These document forms then facilitate direct communication between participants, and provide control and feedback on the status and history of various documents.

The following are forms of Atlassian Confluence functionality, which are unique to collaborative models and which support the fluid scenarios outlined above:

- **Confluence's Spaces** (*Figure 3*): features are used to specify team and project assignments, each have their own space, content and artefacts can be moved between them, and each space is independently managed. Teams and the specialists they share must create, refine, and version documents of various kinds (resources, media documents, guidelines and knowledge bases) that enable their expertise to be used on projects even if they cannot be there to provide it first hand.

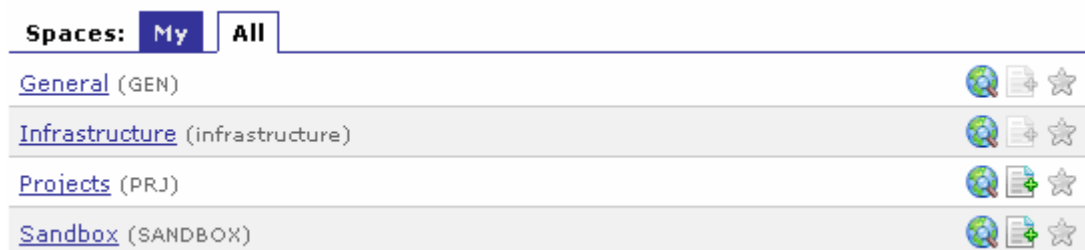
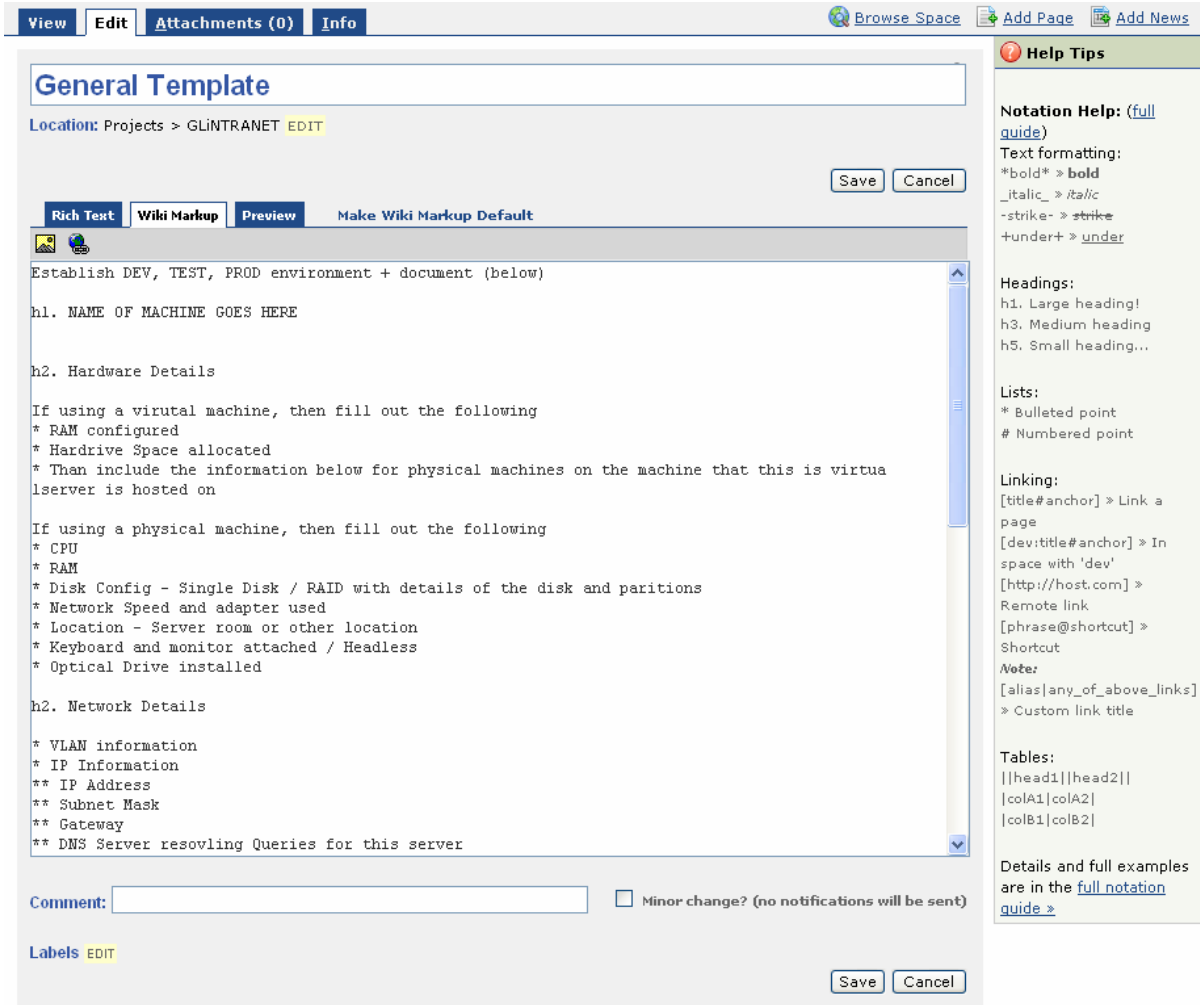


Figure 3: An example of Confluence 'Spaces' using high-level categories. These categories drill down into sub-categories (Children 'Spaces')

- **Supports Multiple Notations** (*Figure 4*): while supporting Rich Text, Confluence also has its own *notation*, a simple, yet powerful mark-up convention to create and edit content headings, text effects, text breaks, links, anchors, images, lists and tables.



General Template

Location: Projects > GLINTRANET [EDIT](#)

[Save](#) [Cancel](#)

Rich Text **Wiki Markup** **Preview** [Make Wiki Markup Default](#)

```

Establish DEV, TEST, PROD environment + document (below)

h1. NAME OF MACHINE GOES HERE

h2. Hardware Details

If using a virutal machine, then fill out the following
* RAM configured
* Hardrive Space allocated
* Than include the information below for physical machines on the machine that this is virtua
lserver is hosted on

If using a physical machine, then fill out the following
* CPU
* RAM
* Disk Config - Single Disk / RAID with details of the disk and partitions
* Network Speed and adapter used
* Location - Server room or other location
* Keyboard and monitor attached / Headless
* Optical Drive installed

h2. Network Details

* VLAN information
* IP Information
** IP Address
** Subnet Mask
** Gateway
** DNS Server resovling Queries for this server
  
```

Comment: Minor change? (no notifications will be sent)

[Labels](#) [EDIT](#)

[Save](#) [Cancel](#)

Help Tips

Notation Help: (full guide)

Text formatting:
 bold > **bold**
 italic > *italic*
 -strike- > ~~strike~~
 +under+ > under

Headings:
 h1. Large heading!
 h3. Medium heading
 h5. Small heading...

Lists:
 * Bulleted point
 # Numbered point

Linking:
 [title#anchor] > Link a page
 [dev:title#anchor] > In space with 'dev'
 [http://host.com] > Remote link
 [phrase@shortcut] > Shortcut
 /note: [alias|any_of_above_links] > Custom link title

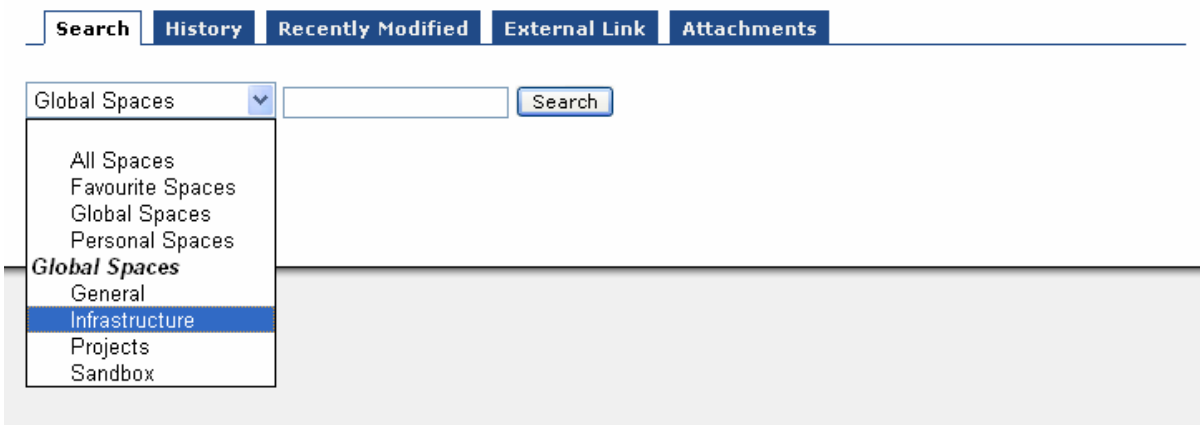
Tables:
	head1		head2	
colA1	colA2			
colB1	colB2			

Details and full examples are in the [full notation guide](#) >

Figure 4: An illustration of the wiki Mark-up language used to layout the structure of the page.

- **Confluence Comments:** are used to facilitate discussion between team members these can be flat (chronologically listed) or threaded (forum like).
- **Bi-directional email support:** facilitates direct communication with team members.
- **Easy Linking (Figure 5):** once created and named, pages can be *linked* together easily along with news items, mail messages, attachments or external sites.

Choose a Page



[Search](#) [History](#) [Recently Modified](#) [External Link](#) [Attachments](#)

Global Spaces [Search](#)

- All Spaces
- Favourite Spaces
- Global Spaces
- Personal Spaces
- Global Spaces**
 - General
 - Infrastructure**
 - Projects
 - Sandbox

Figure 5: Easy Linking provided by the "Choose a Page" popup window

- **Confluence's News features:** are used to define and publish to team members any relevant information of urgent nature - news can be added, viewed, edited and linked.
- **Confluences RSS Feed features:** can be used to provide summaries of information concerning changes to web content; this is used by Confluence to track updates to its own contents and assists in providing control and feedback.

4. Methodology: Adopting and Adapting Confluence

Using concepts from Preece (2000) and Mok (1996), we highlight a simple methodology that could be used to support the evolving needs of collaborative project development, (Figure 6). In describing elements of this methodology, we take the opportunity to introduce some of Atlassian Confluence's outstanding features as an enterprise strength wiki.

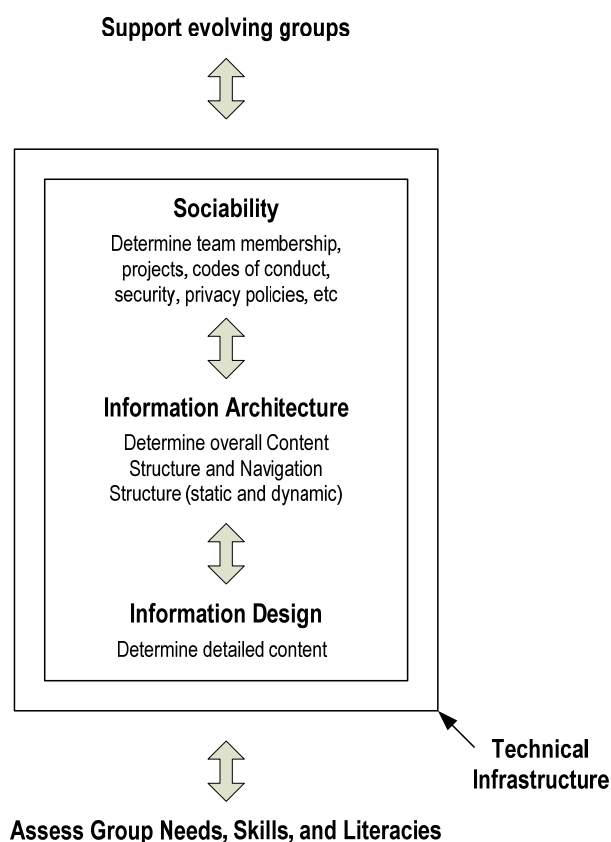


Figure 6: Methodology for Collaborative Project Support using enterprise ready wiki Technologies

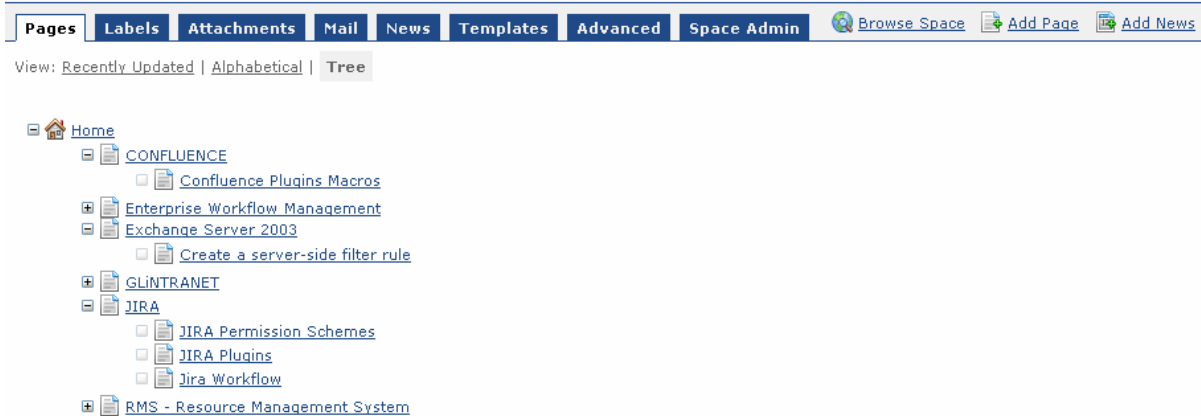
Sociability: Determine team membership, security, privacy policies.

- Use Confluence's *Spaces* features to determine project identity and team membership and creates a place for your team's content. Each team or project can have its own space and each space is independently managed. A portal-like *Dashboard* provides access to spaces with a site, either as *Team Spaces* or personal *My Spaces*.
- Amongst other things, *Spaces* are used to administer team membership and establish *access control settings*. Security is comprehensive can be applied to the space or page level, and permissions can be set page viewing, creation, editing and export; the addition and deletion of posts, comments and attachments; and mail management for example.

Information Architecture: Determine overall Content Structure and Navigation Structure (static and dynamic)

- Site structure can be specified as a *page hierarchy* (Figure 7) enabling as great deal of the wiki structure to be pre-specified.

List Pages - Tree View



The screenshot shows the Confluence Wiki interface. At the top, there are navigation tabs: Pages, Labels, Attachments, Mail, News, Templates, Advanced, and Space Admin. Below these are links for Browse Space, Add Page, and Add News. The main content area shows a tree view of the wiki structure. The tree starts with 'Home' and branches into several categories: CONFLUENCE (with sub-items like Confluence Plugins Macros), Enterprise Workflow Management, Exchange Server 2003 (with sub-item Create a server-side filter rule), GLINTRANET, JIRA (with sub-items like JIRA Permission Schemes, JIRA Plugins, and Jira Workflow), and RMS - Resource Management System.


Figure 7: Tree view of the wiki.

- Use Confluence's *Page Template* features to give a predefined structure to a page at the time when it is generated - the kind of work undertaken by the team, and the types of artefacts they will develop during the life of the project may be known in advance and specified as a *global templates* by site administrators or they may evolve over time and be can be defined as *space templates* by space administrators.
- Use Confluence's *Labels* feature to construct user or team definable tags. These can be used to develop additional user-relevant local navigation structures that enable pages to be categorised, and therefore identified, searched, and bookmarked. Labels can be used to provide a logical view of the site structure in the wiki, and can be used to filter page information during a search. Content structure may be explicitly marked up using labels making searches more domain specific.
- Confluence's *macros* can be applied to perform programmatic functions associated with navigation structure. For instance a macro can be used to integrate with your choice of Instant Messaging software alerting the user if the author of a particular wiki page is on-line and able to receive messages (Figure 8).

Macros

User macros allow you to do things like display your online status of your instant messenger. If you have some suggestions for these, or have seen some you would like implemented, please comment below.

So far the following user macros are implemented:

Macro	Description	Usage	Example
{jabber}	Will show your Jabber status on GLINTECH's Jabber server	{jabber}geoff.cobs {jabber}	 geoff.cobs

0 comments | [Add Comment](#)

Figure 8: An example of Macro usage. When user the Jabber Instant Messaging protocol.

- Use Confluence's *plug-ins* to provide extensibility - extend by adding new functionality or customise in order to change the behaviour of existing features.

Information Design: Design detailed content

- Customising the look, feel, and appearance, using *Themes* and *Colour Schemes* (Figure 9).

Choose Theme

To change the theme of this space, select one below.



Figure 9: Configurable look and feel.

- Confluence enables you to *attach* and *embed* rich media content including Flash (.swf), Quicktime movies (.mov), windows media (.wma, .wmv), real media (.rm, .ram), MP3 files (.mp3).
- Use Confluence's *macros* to perform programmatic functions within wiki pages including for example, generating complex and/or dynamic content. Existing macros can be used to provide useful media features including the auto generation of image thumbnails and galleries!

Sociability, Information Architecture and Information Design all change over time. Initial team planning shifts to supporting a constantly changing social structure. The overall Information Architecture originally created will require reorganisation to accommodate unanticipated work functions. The collaborative authorship continually improves the Information Design of content, which in turn influences the Information Architecture. Enterprise ready wiki technologies like Atlassian Confluence occupy the inner rectangle in Figure A. But this technology sits with a broader technical infrastructure, shown as the outer rectangle in Figure A. Atlassian Confluence emphasises integration with existing communication technologies for example, news aggregators to gather external feeds, the export of Confluence Pages and Spaces to hyperlinked PDFs, and full SOAP and XML/RPC interfaces.

5. Conclusion

Arguably, of all wiki technologies currently available, Atlassian Confluence best supports collaborative project team development in today's volatile and competitive business environments.

6.0 References

Dix, A.; Finley, J; Abowd, G. and R. Beale (1998) *Human-Computer Interaction* Second Edition, London: Prentice Hall Europe

Mok, C. (1996) *Designing Business: Multiple Media, Multiple Disciplines* California: Adobe Press

Preece, J. (2000) *Online Communities: Designing Usability and Supporting Sociability* John Wiley

Wikipedia (2006) "Wiki" <http://en.wikipedia.org/wiki/Wiki>