|  |
| --- |
|  |
| SSW  Management Report |
| Knowledge Management Systems - Wikis |
| **10286084**  **Glenn O’Sullivan**  **10407500** |
| **Philip Olsen** |
|  |

# Executive Summary

There are two aims of this report; to understand the roles of and differences between the differing forms of knowledge management systems (KMS) compared to a wiki and to research and recommend a KMS that will meet all of SSW’s requirements.

Wikis are a collection of web pages that can be accessed and then edited by users through a simple mark-up language or “What You See Is What You Get” (WYSIWYG) editor.

The report discovered the following general principles:

* Blogs are best used for current news items, and opinion based information sharing
* Wikis are best used for objective and general knowledge sharing amongst a user community
* Discussion Boards are for detailed questions/discussions and can be either opinion based or objective
* Knowledge bases are for expert information about a specific topic or product

Following this research, we analysed the requirements of SSW and its environment and selected the most popular wiki solutions that were likely to meet the requirements. These requirements were:

**Required (In Priority Order)**

1. Edit button on each rule
2. Ability to design and customise the site appearance (e.g. CSS)
3. Category per rule
4. Comments per rule

... and 14 more

Using a comparison table, we directly compared the 4 solutions; SharePoint, Confluence with SharePoint, MediaWiki and TWiki. SharePoint was selected as the base to compare to the other solutions as it was the best option in a direct comparison.

The analysis found that despite TWiki having more complete functionality through user-made plugins, SharePoint was a better option for SSW by virtue of its development skills and ability to leverage SSW’s access to Microsoft resources as a Gold Partner.

Implementation of SharePoint with the option of adding Confluence on top in the future if the functionality of SharePoints built-in wiki was unsatisfactory for SSW’s needs was recommended.

# Contents

Contents

[Executive Summary 2](#_Toc201740017)

[Contents 4](#_Toc201740018)

[Introduction 5](#_Toc201740019)

[Requirements 6](#_Toc201740020)

[Scenarios 7](#_Toc201740021)

[Current 7](#_Toc201740022)

[Comparison 9](#_Toc201740023)

[Wikis 9](#_Toc201740024)

[Blogs 11](#_Toc201740025)

[Discussion Boards 14](#_Toc201740026)

[Knowledge Bases 15](#_Toc201740027)

[Analysis 18](#_Toc201740028)

[Recommendation 23](#_Toc201740029)

[Implementation 25](#_Toc201740030)

[Conclusion 31](#_Toc201740031)

# Introduction

The popularity of knowledge management systems (KMS) has grown rapidly in recent years as businesses strive to manage and streamline the creation and display of information. Some common examples of KMS are wikis, blogs, discussion boards and knowledge bases. Recently trends have shown the increasing popularity of “social” KMS solutions, such as wikis. By having user friendly, intuitive interfaces, wikis appeal to users who then add and share their knowledge with others.

The goal of this report is to analyse the suitability of various popular KMS for implementation at SSW. As SSW is a Microsoft Gold Partner and it has .NET-centric skill sets, the requirements of the system are focused towards a .NET or Microsoft solution. The report first aims to determine what features wikis lack in comparison to other KMS (such as blogs, discussion boards and knowledge bases) and determine if a wiki can be a holistic solution for knowledge management or if several systems must be used in conjunction with one another.

Using the requirements elicited from the client, the report will then compare and contrast the top 3 suitable solutions. This analysis will examine each solutions key features, extensibility, ease of implementation and maintenance as well the degree to which it meets the requirements.

From this detailed analysis, the report will present its recommendations for SSW given its requirements and infrastructure environment. These recommendations will then be used to form a management implementation that compares the “out of the box” functionality available with developing customised code in a requirement by requirement review.

This document should be used for the basis of a release plan and the development of the final solution.

# Requirements

There are a number of Wikis available each with differing functionality, purposes and cost. As such, it is vital to understand SSW’s needs and expectations of a wiki system. After a discussion with SSW management, the following is a comprehensive list of requirements and features that are to be delivered by the wiki solution, in priority order:

**Required (In Priority Order)**

1. Edit button on each rule
2. Ability to design and customise the site appearance (e.g. CSS)
3. .NET language
4. Category per rule
5. Comments per rule
6. Comments are hidden until user opts to view them (e.g. Separate page or AJAX)
7. Ratings per rule (agree and use)
8. Rules are separated by category and can be ordered by
   1. Agree rating
   2. Use rating
   3. Importance
   4. Order
9. Tasks per rule (e.g. Shown as Tasks (X) ~~Tasks Done (Y)~~)
10. WYSIWYG editing
11. Publish all rules to PDF for printing with publishing service
12. User registration and logon
13. Workflow for user edits
14. Permissions levels per rule (open, semi-protected, protected)
15. Visible version number per rule
16. Reports:
    1. tasks complete per category
    2. tasks to be completed per category
    3. agree per rule per category
    4. agree per category
    5. used per rule per category
    6. used per category
17. Support for embedded SQL statements

**Desirable (In Priority Order)**

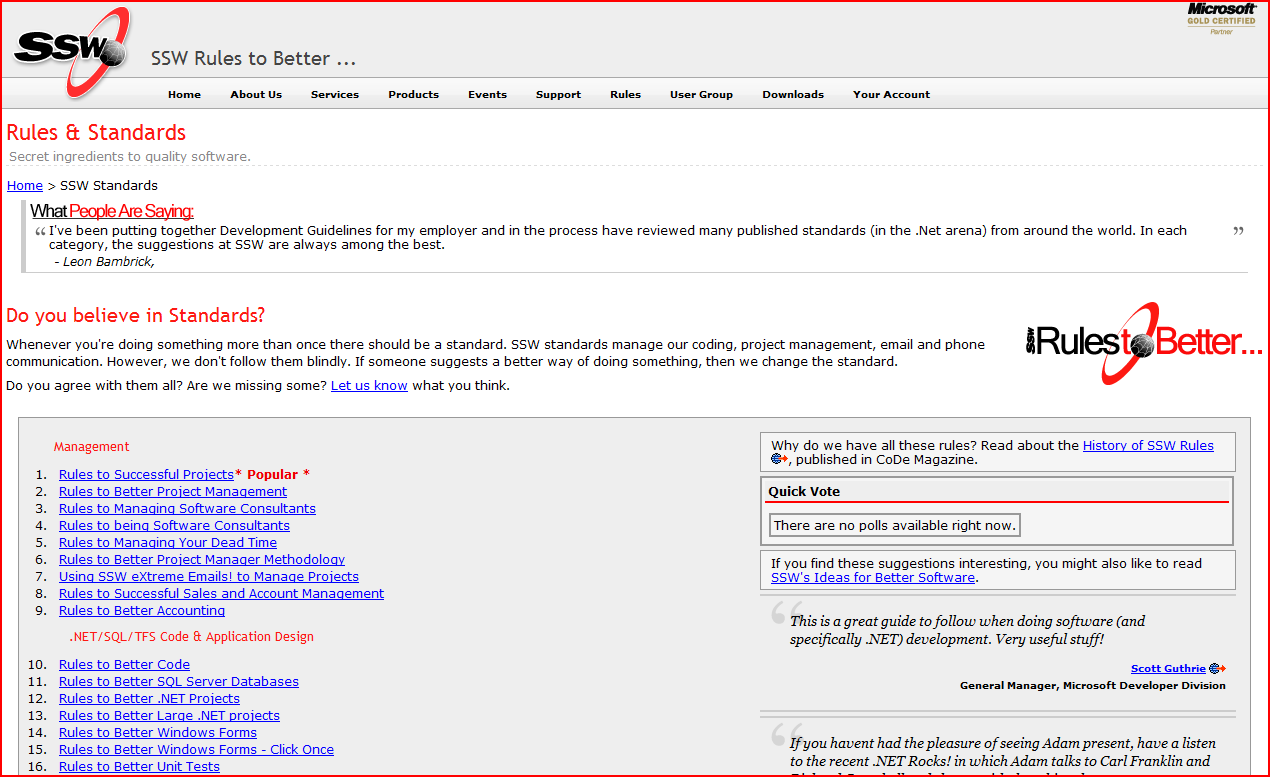
1. Viewable history per rule
2. Trackbacks per rule
3. Related rules (e.g. Phil’s Rules or by using tags)
4. Workflow for trusted user edits (e.g. users become trusted after 3 accepted edits)
5. Edits can be made offline and uploaded later

Preliminary research indicates that no single wiki will be able to meet all of these requirements in a default configuration. However, many wikis offer the ability to create additional functionality through plug-ins. The ability to code new features, and the time and skills required to do so, will be taken into account during both the initial and detailed analysis of the systems.

# Scenarios

## Current Solution

The following figures illustrate the current solutions. They all share a lack of the required functionality described in the requirements.



SSW Rules main website

### Summary of Missing Functionality

* No edit button
* No task tracking button
* No WYSIWYG editor
* No publishing to PDF/printing service
* No editing permissions
* No workflow for approving changes
* No comments
* No SQL statement support
* No trackbacks
* No versioning (this can be viewed in Team Foundation Server but only by developers)
* No support for editing pages in Microsoft Office

It is this lack of functionality that SSW seeks to resolve through the implementation of a new KMS such as a wiki.

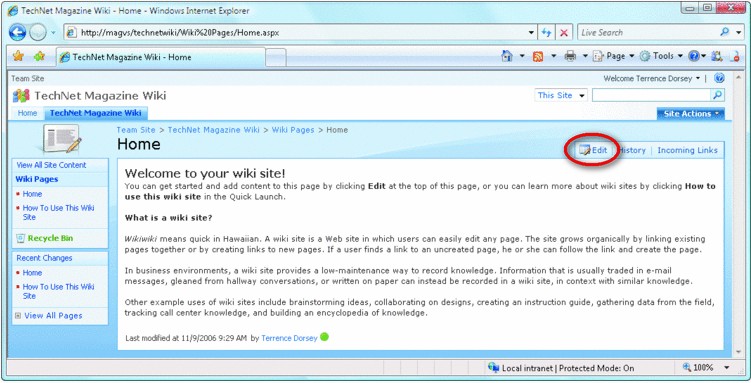
# Comparison of Technologies

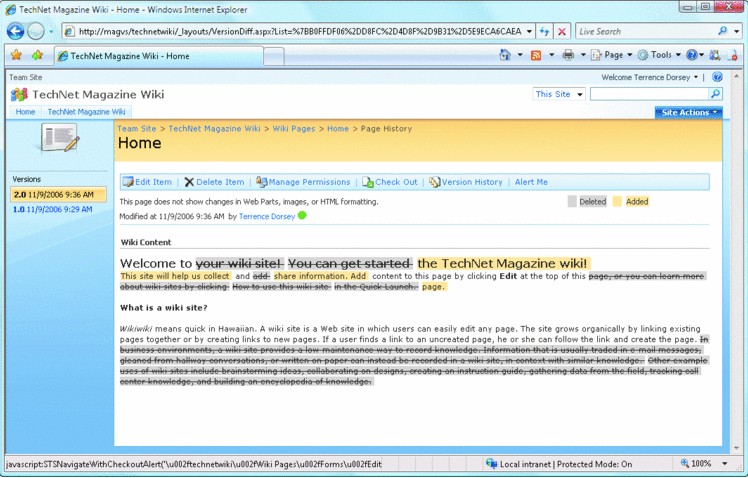
In order to fully realise the potential functionality of a wiki, it is pertinent to examine the alternative types of knowledge management systems and compare their functions with those of a wiki.

## Wikis

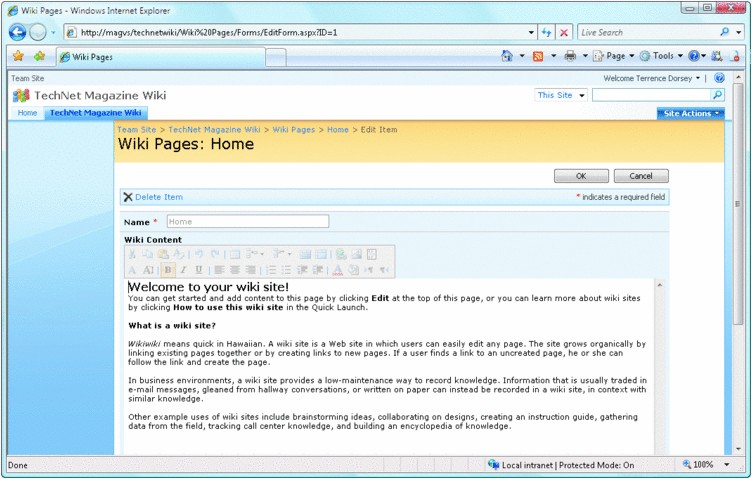
Wikis are a collection of web pages that can be accessed and then edited by users through a simple mark-up language or “What You See Is What You Get” (WYSIWYG) editor.

They operate on the concept that users will share knowledge and information with one another which will then be corrected if/when necessary by other users. By their nature, wikis require little maintenance on behalf of the host.

 A wiki with an edit button displayed on each page



Editing a wiki page with change tracking turned on



Editing a wiki page with a WYSIWYG editor

### Downside of Wikis

Wikis are particularly susceptible to “vandalism” due to the fact that any user can edit the content. This can be limited through use of a login system, but makes the wiki less appealing for legitimate users to edit.

### Popular Wiki Engines

MediaWiki

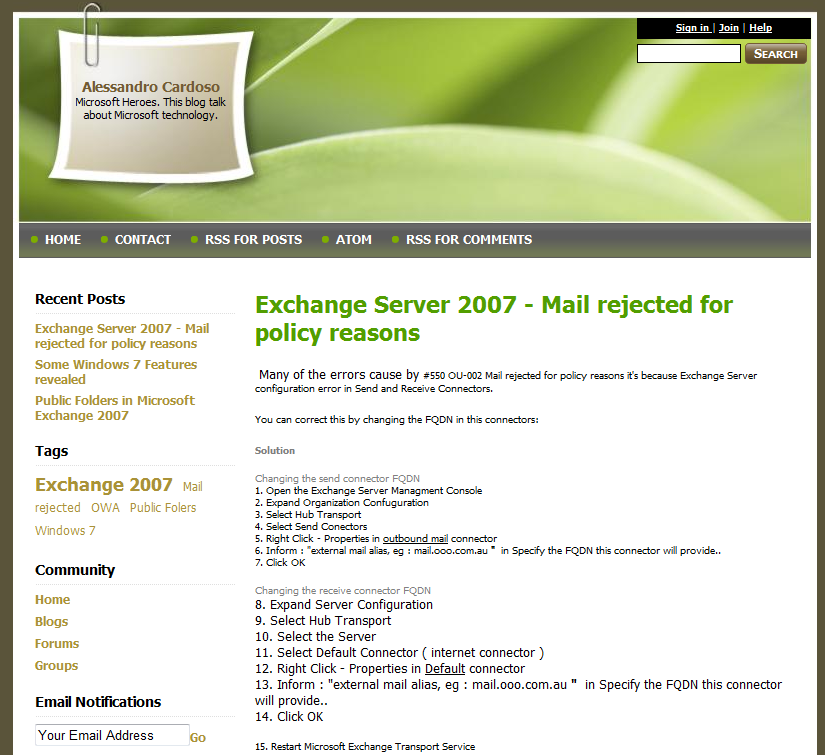
TWiki

Confluence

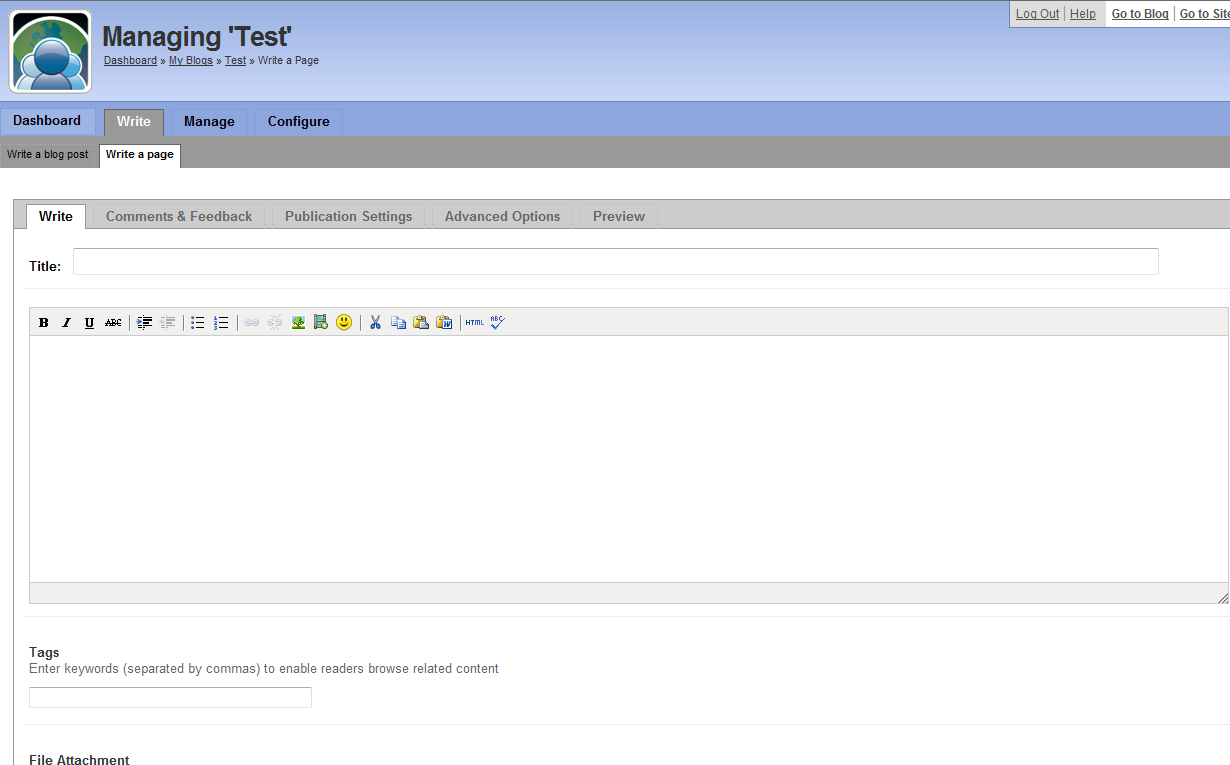
Microsoft Sharepoint

## Blogs

Blogs are a website, usually written by a single or small closed group of authors, with regular posts on current topics. As they tend to be written about news items, many are organised in chronological order. Most blogs also feature a comments system in which readers can post feedback, questions and answers to the blogger and/or other readers.



Example of a blog post on Live Writer



Example of creating a blog post in Live Writer

### Getting traffic

Increasingly, blogs are offering ‘Social Networking Buttons’ that allow the reader to flag the post in a number of social sites (Digg, Slashdot, Technorati, etc) or make their own blog about the post. This can also be done on wikis, however it is not as common.

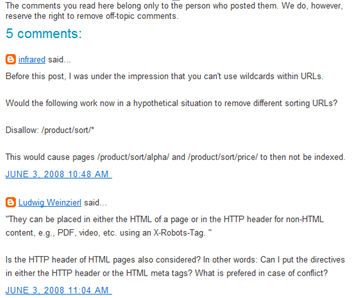


Example of Social Networking Buttons from PeerIT

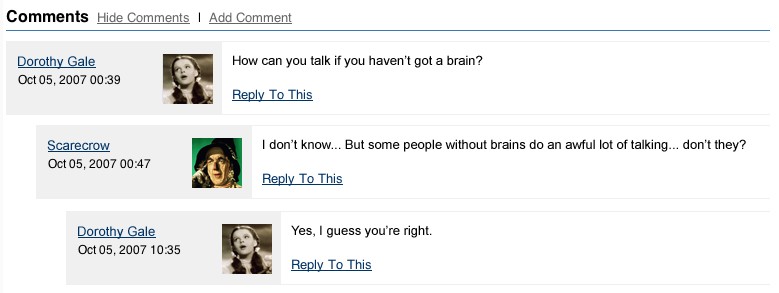
### Comments

Blogs and wikis tend to have a similar comment systems, these are:

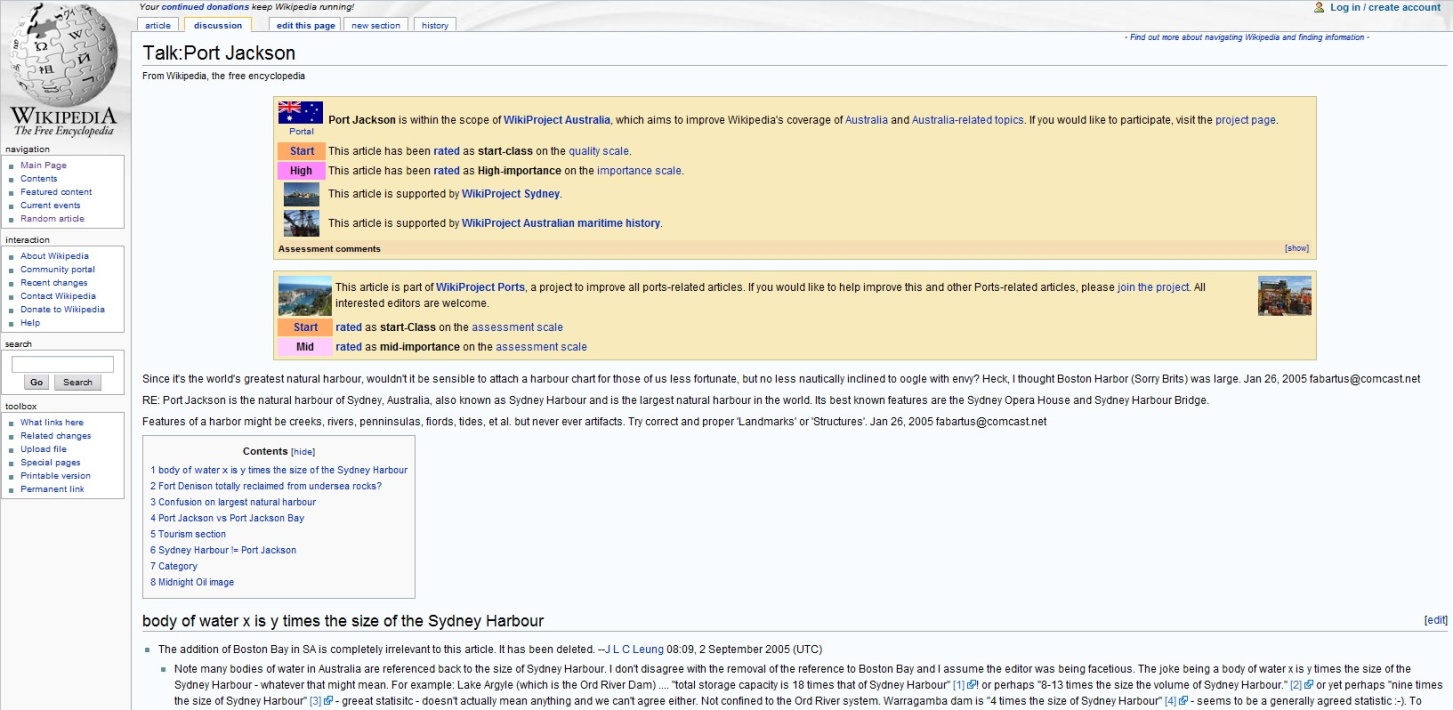
* Inline comments (e.g. Blogspot)



* Threaded comments (e.g. Confluence)



* Single page (e.g. Wikipedia)



### Downside of Blogs

Blogs tend to fall out of date and/or relevance. Blogs also tend not to be updated because of their opinion based nature, authors do not edit their old posts, instead they create a new one.

### Popular Blog Engines

Wordpress

Blogspot

Blogger

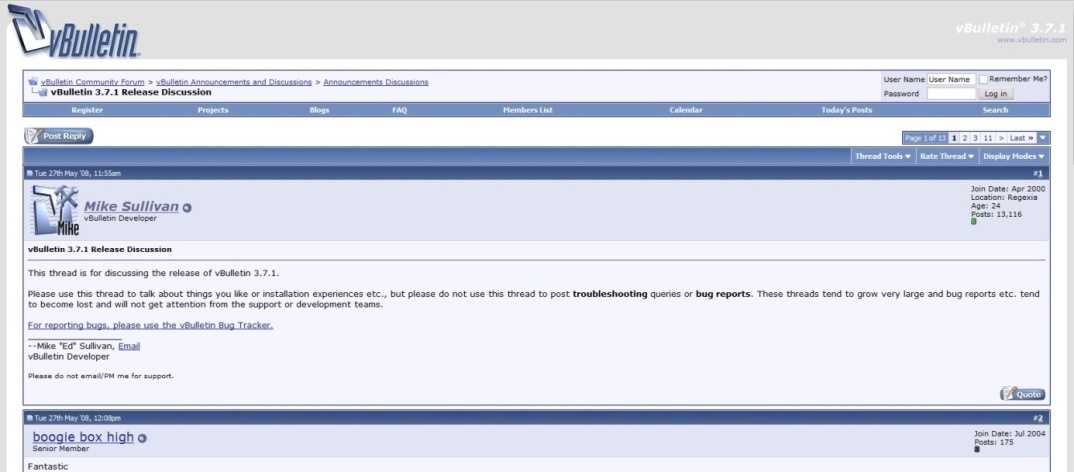
### Summary - Functionality Matrix

|  |  |  |
| --- | --- | --- |
| Functionality | Blogs | Wikis |
| Chronological Ordering | Primary ordering system, may also include categories | End user driven, usually topic based |
| Comments | Out of the box | Out of the box – could be as:   * inline comments * threaded comments * single page |
| Social Networking Buttons | Out of the box | End user customisation of pages/master pages or plug-ins |
| Authors | Closed group or individual | Social |
| Objectivity | Opinion based | Objective viewpoint |

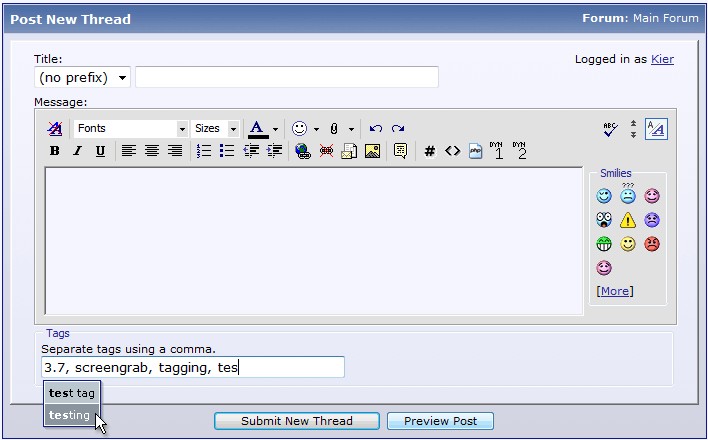
As the table shows, blogs offer no real advantage or functionality that cannot be implemented within a wiki solution. Wikis provide a social based and objective presentation of knowledge as well as the ability to be organised by topics instead of dates.

## Discussion Boards

Discussion boards (A.K.A forums) are an application for posting questions and information on various topics, ordered by most recently active posts.



An example of a discussion board



Creating a post on a discussion board

### Credibility

Typically, users must register a username before they are allowed to post and only closed group of moderators and administrators are able to edit posts by other users. This limits the potential for malicious users to cause damage to the discussion board as users can only edit their own posts.

### Getting Information

Discussion boards are good for asking a specific question that is too detailed or niche to be included in a FAQ or wiki page.

### Downside of Discussion Boards

Unfortunately, the increased credibility of a discussion board this means that there is an increased maintenance cost as administrators have to delete bad posts. Sometime, poor search functions can result in the same or very similar questions being asked and responded to multiple times.

### Popular Discussion Board Engines

VBBulletin

phpBB

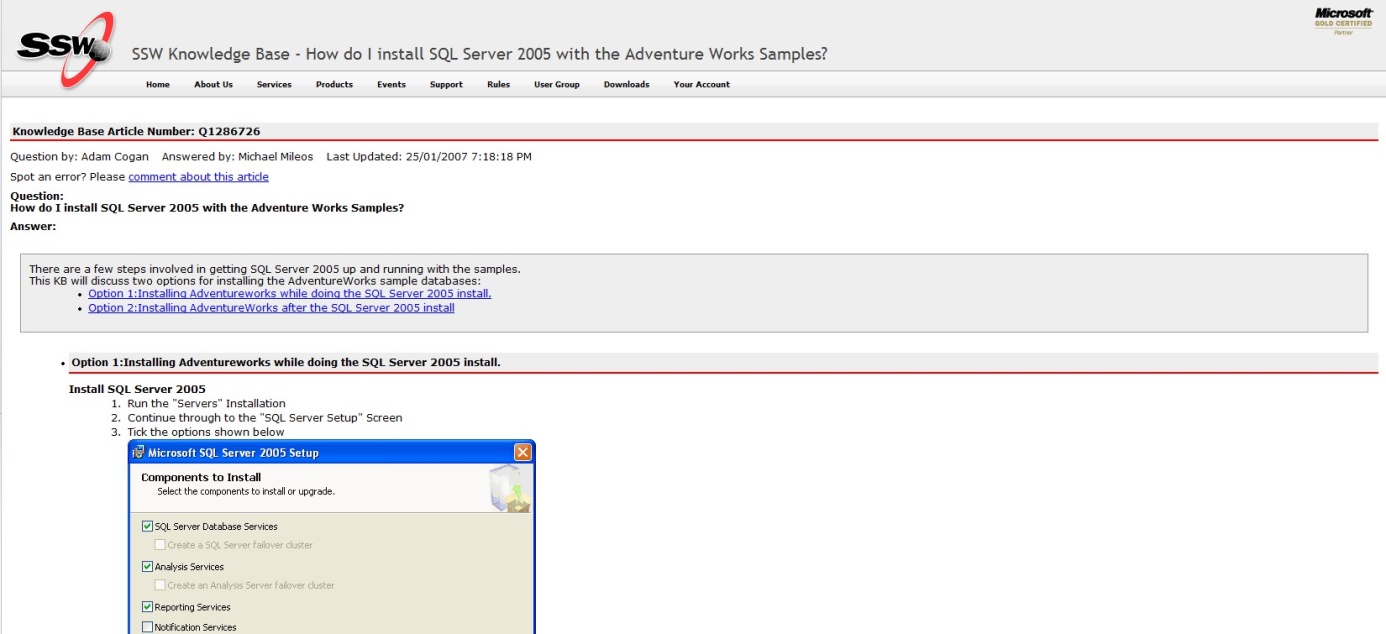
### Summary - Functionality Matrix

|  |  |  |
| --- | --- | --- |
| Functionality | Discussion Boards | Wikis |
| Ability to ask specific questions | Yes, but it may have been answered before | Not on the main wiki page, but can be asked within the comments system which may be a discussion board |
| Chronological Ordering | Topic based but most recent posts are shown first | End user driven, usually topic based |
| Credibility | Requires user to register. Users cannot edit other’s posts. | Depends on configuration. Can use protected/semi-protected pages to limit this. If unprotected, other users can fix vandalism without administrator/moderator permissions |
| Objectivity | Can be both objective and opinion based | Objective viewpoint |

A discussion board offers no relevant advantages over a wiki solution with appropriate access permissions applied. A Wiki will prevent information being posted multiple times and allows for less maintenance by the host.

## Knowledge Bases

A knowledge base is a data base of known solutions and procedures, primarily for use in training and troubleshooting. It is written by a closed group of users with expert knowledge of the topics selected. Its purpose is to list all known problems and solutions so that users encountering already documented problems can easily troubleshoot and resolve them.



Example of SSW knowledge base

### Credibility

Only experts have access to post information on a knowledge base and this greatly increases the accuracy and credibility of the information provided.

### Downside of Knowledge Bases

There is no social information sharing, other than users providing the host with private suggestions or bugs which the host then has to confirm and post on the knowledge base. This has a higher maintenance cost than other KMS.

### Popular Knowledge Base Engines

ASP

JSP

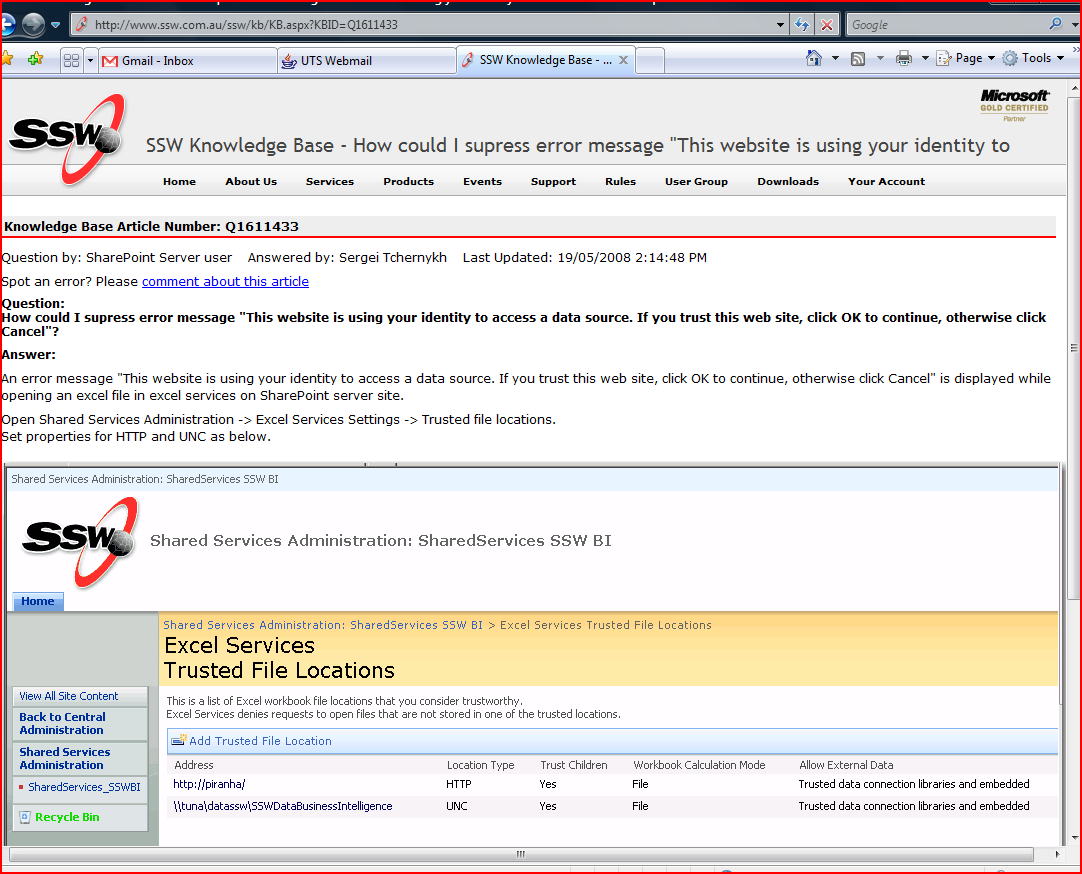
### Summary - Functionality Matrix

|  |  |  |
| --- | --- | --- |
| Functionality | Knowledge Bases | Wikis |
| Credibility | Closed group of experts | Depends on configuration. Can use protected/semi-protected pages to limit this. If unprotected, other users can fix vandalism without administrator/moderator permissions |
| Comments/questions | No. May be possible to email the owner for the knowledge base | Yes, through discussions/comments system |
| Maintenance | High | Low due to social nature of wikis |

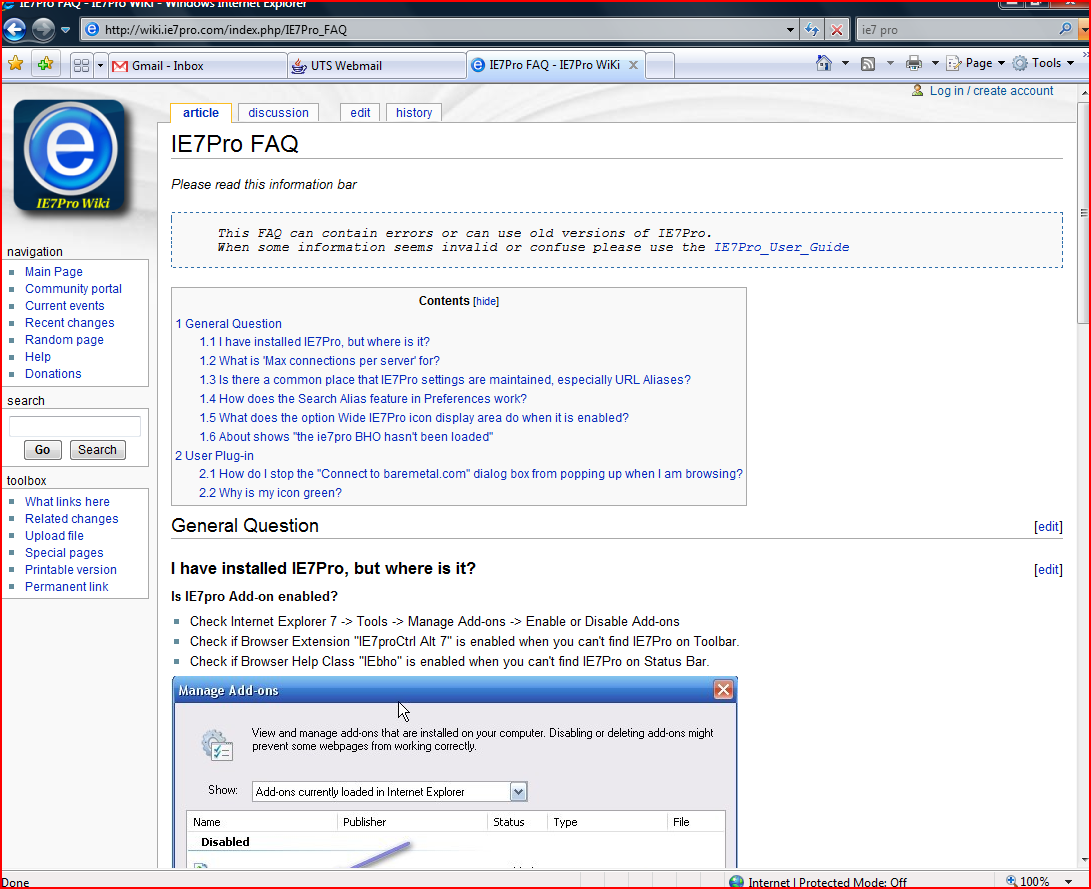
Knowledge bases provide no relevant advantages over a wiki solution. Wikis can use change approval workflow to increase credibility and still have a lower maintenance cost. Finally, wikis support publicly viewable comments with knowledge bases do not.

# Analysis of Wiki Solutions

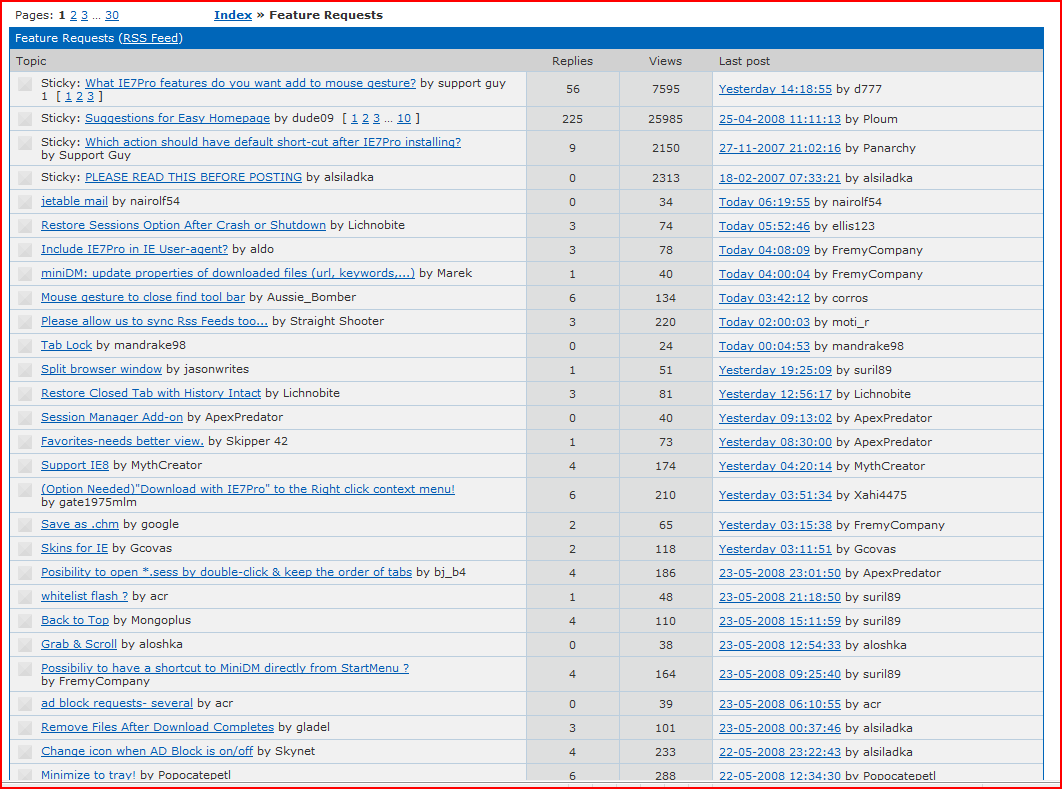
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reqs/Wikis** | SharePoint | Confluence w\ SP | MediaWiki | Twiki |
| License | Commercial | Commercial | GPL | GPL |
| License Fee | ~$30,000 | US$4/8k (500/∞ users) | $0 | $0 |
| Language | C#, VB | Java | PHP | Perl |
| Data Storage | MS SQL | SQL DB (inc. MS SQL) | MySQL | Text files/RCS |
| Intended Users | Enterprise | Enterprise | Education | Enterprise |
|  |  |  |  |  |
| **Features** |  |  |  |  |
| Per Section Editing (1) |  |  |  | Plugin |
| Task Tracking (2) |  |  | Plugin (Semantic) |  |
| Page Revisions (2) |  |  |  |  |
| Revision Diffs (2) |  |  |  |  |
| WYSIWYG (3) |  |  | Plugin |  |
| XML Export (4) |  |  |  | Plugin |
| PDF Export (4) | Plugin |  | Optional |  |
| Page Permissions (5) |  |  |  |  |
| Access Control Lists (5) |  |  |  |  |
| Change Notification (6) |  |  |  |  |
| Comments (7) |  | Threaded | Discussion Pages | Threaded |
| Dynamic SQL Statements (8) |  |  |  |  |
| Trackbacks (9) |  |  |  |  |
| Change Summary (10) |  |  |  |  |
| Email Post Generation (11) |  |  |  |  |
| Extensibility (12) |  |  |  | Up to 20,000 pgs |
| Blog Feature |  |  | Plugin | Plugin |



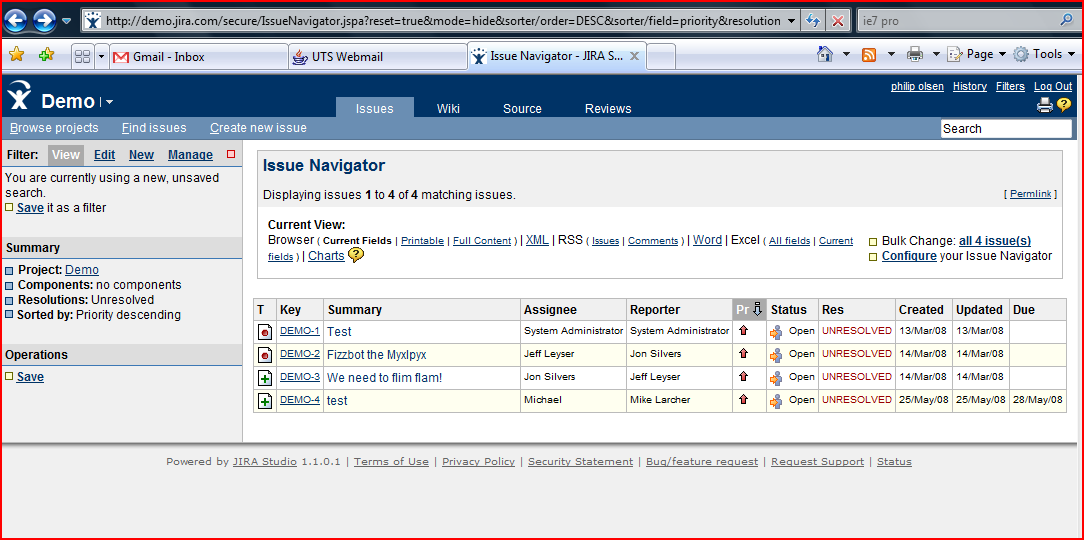
This page has a question and answer clearly shown. It also in categorised into multiple applicable categories.



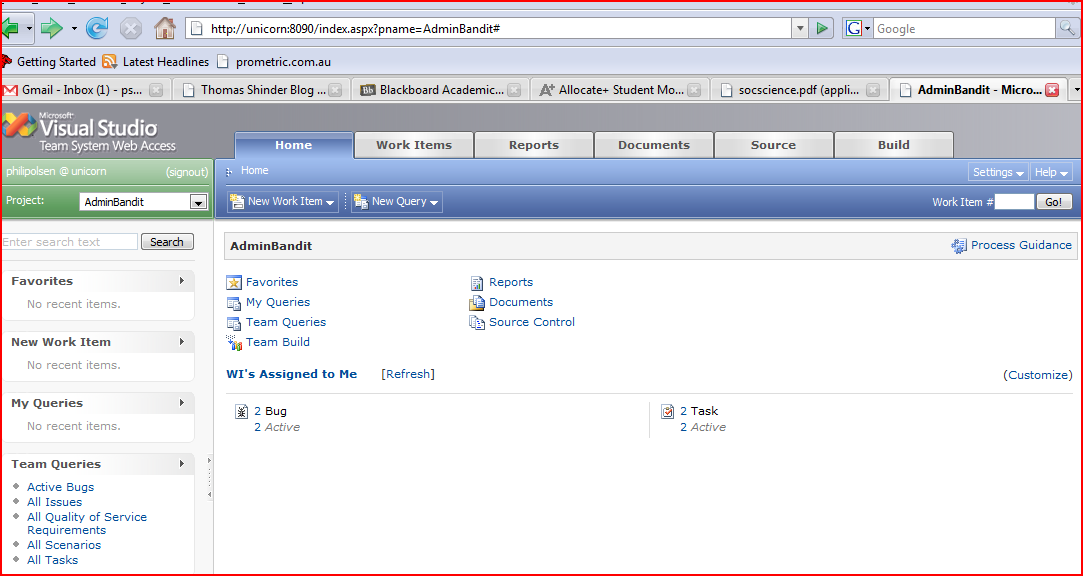
Shows the following features; edit button, rich HTML editing, image support and discussion history



Pro: Allows content to be rated, features an alerts system, questions and answers Neg: category (single), no support for html or images

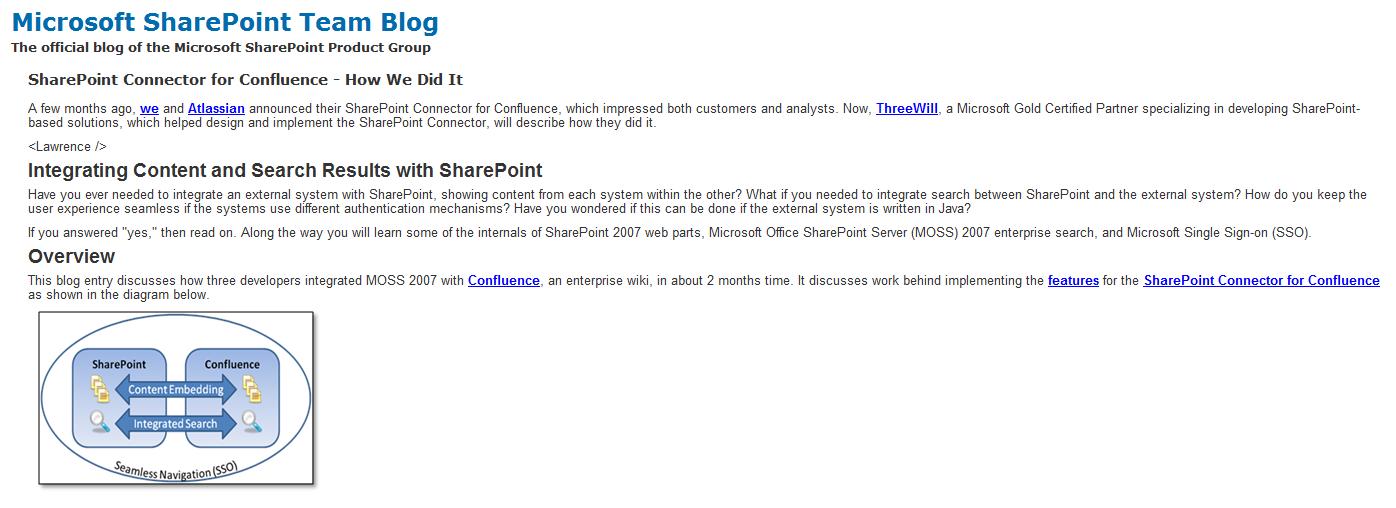


Example of task tracking in Jira



Task tracking in TFS

Should be able to track tasks across a number of pages. This could be achieved by having a search page and passing a URL as a parameter if the user wished to look at an individual page's tasks.



# Recommendation

Through our analysis of possible solutions SharePoint is the best path to follow in SSW’s case. SSW’s future goal of hosting their entire website, including the rules pages and staff blogs, will require functionality that only SharePoint can provide. Further, this decision was influenced by the ability to extend SharePoint with SSW’s development skills and leverage SSW’s access to Microsoft resources as a Gold Partner.

The recommendation is to initially use just SharePoint’s inbuilt functionality for all required areas (i.e. web pages, blogs, wikis, discussion boards) and build custom web components for the requirements that aren’t covered in the standard installation (e.g. self publishing, advanced approval workflows, etc). As this is the only solution that offers discussion boards, blogs, etc.

In terms wiki functionality, it is recommended that three months after the implementation of a SharePoint solution, the functionality be reviewed for all pages that are on the SharePoint wiki model. If further functionality is required on these pages, such as the wiki abilities provided by Confluence, it is recommended that Confluence be added to the SharePoint solution to manage all wiki pages. This can be integrated directly with SharePoint through the SharePoint Connector to provide the same functionality of SharePoint for content management and workflows with the added features of Confluence’s superior wiki functionality.

Other benefits of this solution over others are the ability to cover all foreseeable requirements for SSW’s entire website (i.e. above and beyond wikis) and the ability to translate the knowledge gained directly to client projects with SharePoint implementation.

Implementation should be completed in parallel phases, migrating sections of the site at a time to SharePoint and then adding the customer web parts to them. During this time, the existing website should be maintained and any changes should be replicated to the migrated SharePoint content. This will ensure that the content will work and be displayed correctly with full functionality before cutting over to the new SharePoint solution as the primary website.

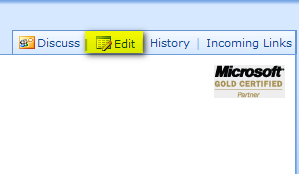
# Implementation

Following our initial recommendations to the client, a meeting was held to discuss alternative configurations of SharePoint that would allow for more customisation of the solution to fit SSW’s requirements. In general terms, customised code is always an option for achieving the exact needs of a project, but the cost in terms of resources can outweigh the benefits over “Out of the Box” (OOTB) functionality. Given this, an analysis was conducted to understand how closely two non-customised solutions could meet the requirements (wiki and blog base configurations), which were then collaborated and compared with a customised solution. Below is a requirement by requirement analysis of an OOTB solution vs. customised coding.

**Required (In Priority Order)**

1. **Edit button on each rule**

This comes OOTB for all SharePoint pages and meets requirements perfectly. This will be implemented OOTB.

  
Figure: Edit button on a wiki page

1. **Ability to design and customise the site appearance (e.g. CSS)**

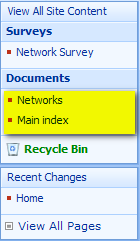
This is a part of SharePoint’s core functionality and comes OOTB. The solution will use custom CSS to change the look and feel of the pages.

1. **.NET language**

This is a part of SharePoint’s core functionality and comes OOTB. The solution will use .NET language for any customised web parts.

1. **Category per rule**

After discussion with the Client, this requirement was updated to allow rules to have a many to many relationship to categories. There was no facility to have this relationship OOTB and given that this functionality only required an estimated 15 minutes of development time, it should be implemented through customised code.

  
Figure: A single category per rule in a wiki site.

1. **Comments per rule**While there is a good 3rd party package that can installed for discussions per rule in a wiki configuration, it forces the user to have a wiki site as the site theme. This essentially rules it out as a solution as the custom master page and page layouts, required for the other requirements, cannot be the use a wiki site. Therefore a custom developed list is required. This is estimated at 24 development hours to create the comments web part and comments list.



Figure: Link to discussion in a wiki site

1. **Comments are hidden until user opts to view them (e.g. Separate page or AJAX)**As comments will be custom developed, this will be a part of the development specifications.

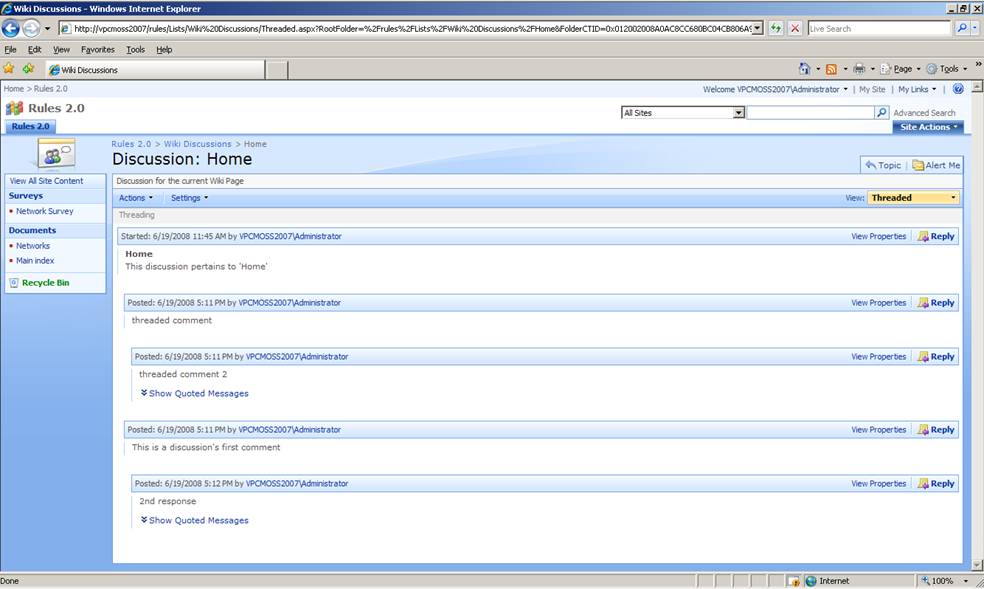


Figure: Comments on a wiki site

1. **Ratings per rule (agree and use)**

Through use of the survey webpart, rating system can be created although the system is slightly cumbersome. OOTB will be used in the early releases of the system with the possibility customised code to create a more personalised results chart in the future.

Figure: A survey webpart

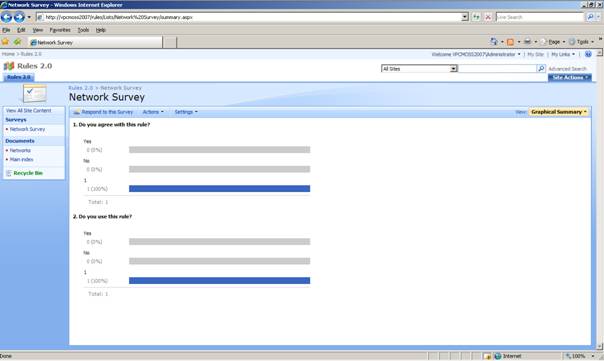


Figure: Results of the survey

1. **Rules are separated by category and can be ordered by** 
   1. **Agree rating**
   2. **Use rating**
   3. **Importance**
   4. **Order**

There is no built in provision for this ordering and will require customised code to meet the requirements. This will take an estimated 16 hours to develop.

1. **Tasks per rule (e.g. Shown as Tasks (X) ~~Tasks Done (Y)~~)**

While a task system can be created OOTB, centralised task tracking in TFS is a priority. In order to have these tasks integrated, customised code is necessary.

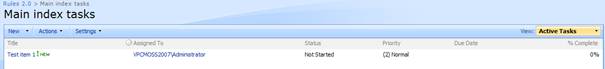
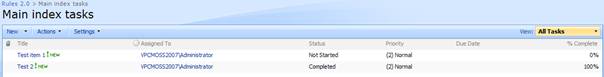


Figure: Various task reports from an OOTB task list

1. **WYSIWYG editing**

This is available by default in SharePoint, the only downside is that images cannot be copied and pasted directly into the input box. While a custom webpart could achieve this, the development time is too costly for the small benefit that would be achieved.

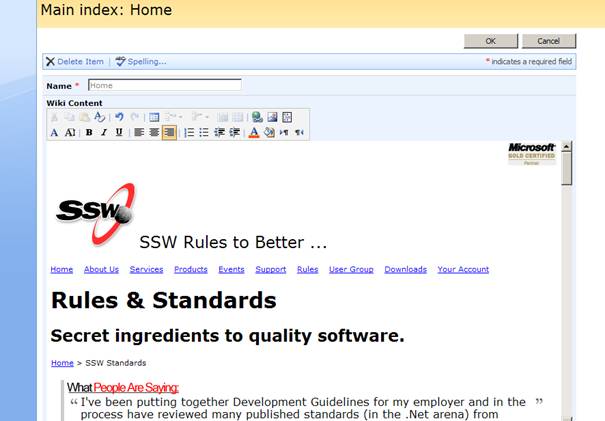
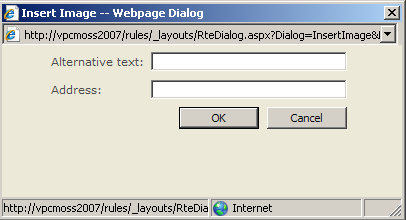
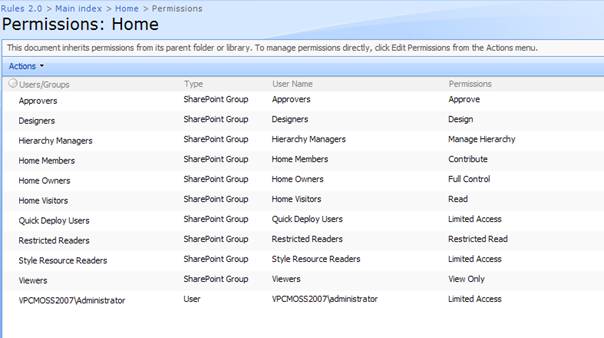


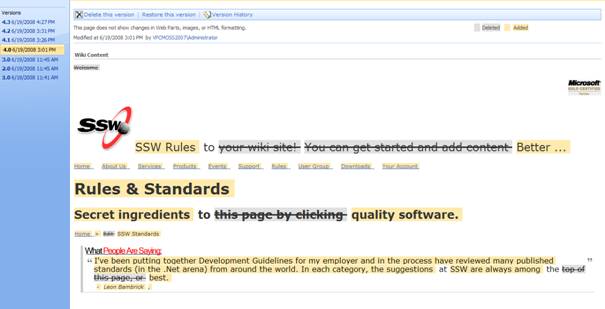
Figure: WYSIWYG editing



1. Publish all rules to PDF for printing with publishing service
2. User registration and logon
3. Workflow for user edits
4. Permissions levels per rule (open, semi-protected, protected)



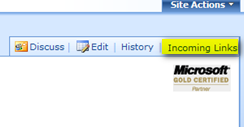
1. Visible version number per rule



1. Reports:
   1. tasks complete per category
   2. tasks to be completed per category
   3. agree per rule per category
   4. agree per category
   5. used per rule per category
   6. used per category
2. Support for embedded SQL statements

**Desirable**

1. Viewable history per rule
2. Trackbacks per rule



1. Related rules (e.g. Phil’s Rules or by using tags)
2. Workflow for trusted user edits (e.g. users become trusted after 3 accepted edits)
3. Edits can be made offline and uploaded later

# Conclusion

Through our report, we have discovered the similarities and differences between the various forms of knowledge management systems and explained the benefits and short fallings of each. Through this we have understood that each form of KMS has a specialty which will influence a decision on which type to use.

Taking a strictly functional viewpoint, any solution may be able to cater for all needs of the project, but usually will require either customisation or a sacrifice in elegance and quality of the solution. As such, we have found that it is better to use the best KMS tool for its desired purpose.

After a detailed investigation of possible wiki solutions, the idea of the best tool for the job leads us to recommend SharePoint. SharePoint’s diversity makes it the clear and obvious choice for SSW. Through its ability to provide wiki, blog, discussion board and webpage functionality, it is an all in one solution for SSW’s needs.