



Wikidsmart turns your enterprise wiki into an information collaboration dashboard, delivering: structure to your wiki; semantic capture of information within the wiki; deep interoperability with other applications and tools; exposure of information from other tools and applications; and smart search across all tools and applications.



zAgile Wikidsmart: Just plug in your tools and applications for a comprehensive information collaboration dashboard across your teams.

Information collaboration is the deep integration of information across teams, tools, and applications, with a common semantic web based infrastructure, enabling comprehensive organization-wide collaboration.

Challenges of the Traditional Enterprise Wiki

Whereas a wiki provide an easy mechanism for users to collaborate, develop and maintain documentation and other content within an enterprise, its inherent wiki characteristics also come with some natural limitations:

- **Consistency of content:** First there is no ability of teams to collaborate and create content in a consistent manner. Whether it is information related to projects, processes, people or some other domain-specific information, how consistently is that information captured? How easy is it to create content in a consistent manner across teams?
- **Search precision:** Search is limited to string matches within the content and can best return non-specific result sets. As the number of pages increases, this search method becomes increasingly inefficient.

Wikidsmart's semantic wiki engine turns your enterprise wiki into an information dashboard:

- » Find precise information easily, across the wiki as well as other applications and tools
- » Generate and maintain new pages with embedded queries, pulling content from the wiki and other applications and tools
- » Achieve consistency of content across the wiki and other tools and applications
- » Easily capture semantic annotations of existing content
- » Create new content using semantic templates and forms, and populate other systems if desired

Wikidsmart for
ATLASSIAN
CONFLUENCE®

“zAgile seemed too good to be true, but they proved it to me. Before zAgile, we had the right tools, but we saw opportunities to gain more efficiency. ... now we are even more productive across engineering, professional services, and product management teams.”

Niall Murphy
VP of Engineering, Market6

- **Cross-referencing across pages:** Maintaining cross-references of information across pages is typically a tedious process that involves manually embedding links using wiki markup. These links, if not updated, can easily become stale, rendering the content unreliable and outdated
- **Integration of external information:** Information Integration with other applications and vice versa is mostly limited to data sharing via RSS feeds. If you are capturing information about product requirements in a wiki, then you may also want to integrate them with corresponding test cases, tasks, check-ins, etc. And you may want to do it both ways, i.e., integrate information from other tools into wiki pages but also pull some information from the wiki into other tools or applications. However, aside from page, space and section-level URLs, there is no other mechanism for such integration. Furthermore, the lack of integration with other applications and the lack of attribute level support limit the wiki to a static knowledge repository. If specific information changes on one page, the change is not automatically reflected everywhere else it may also appear. It must be manually updated, and if not, the content quickly loses integrity.
- **Context associated with pages:** The content categorization may be implied in the page hierarchy but there is no inherent structure to support it. Does a page depict a requirement, a process, a profile of a team member, some information about business partners, or instructions for a holiday party? There are limited mechanisms to define it through the use of labels but it is manual, arbitrary, and not structured. A page's content reflects some information or knowledge - but what is its nature? What is it describing? What are its relationships with other information concepts? You may be able to glean that off of the page title, its place in the page hierarchy, labels or its Space container but the page itself does not contain any attribute-level information.
- **Content organization is limited to mostly page-level hierarchies**
This limitation is quite analogous to binders on a shelf. Designing and maintaining the page hierarchies, cross-referencing pages across spaces, tracking and updating them with the most current information and maintaining cross-reference links all require significant manual work. In the absence of this level of meticulous support, the wiki content quickly becomes outdated and unreliable. Within the context of Atlassian's Confluence enterprise wiki, the most typical method of content organization is through the use of Spaces which may imply or mimic some topic.
- **No wiki page attributes or metadata.** Information access may be accomplished via searching by keywords, navigating through the Space and page hierarchies or looking for a specific page title or a URL. But because there is not an easy mechanism to store page-level attributes or properties within the context of the page itself, you cannot so easily find a page written by a specific author, for a particular project, depicting a

Limitations of Other Semantic Wikis

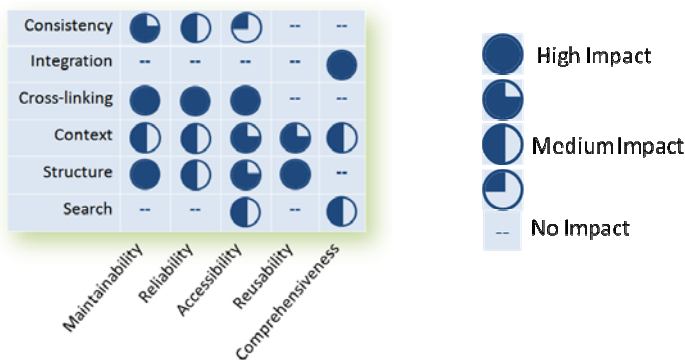
- » *Semantic wikis that are built natively* from ground up do not offer much to organizations that have existing investments in wikis, unless they migrate their content and adopt the new wiki technology. They also prevent users from leveraging the community-level support and feature extensions available in widely popular wikis.
- » *Semantic extensions:* this approach supports popular wikis but have made the wiki itself a semantic repository, as opposed to one of many sources of semantic content

zAgile Wikidsmart provides semantic enablement of wikis and leverages the following high-level functional components:

- » zAgile's Semantic Repository consisting of a set of ontologies and metamodels specific to a domain of interest. Ontologies are for Software Engineering are included, and others are available through zAgile professional services or other ontology providers.
- » zAgile's Semantic Interface Layer provides wikis with access to the Semantic Repository via connectors
- » zAgile's Semantic Plugin for Confluence provides the interface between Confluence and the zAgile Semantic Layer and supports macros for creating templates and forms in Confluence for annotating wiki pages.

specific process or containing a specific section. This limits the level of granularity of the information or knowledge that is being captured and accessed within the page content. For example, a page may depict information on business partners of the organization, use cases related to a specific requirement or roles effective within a project. But it is still mostly accessible as a page and at best, as sections within the page.

**Limitations
of
Traditional
Enterprise
Wiki
Content**



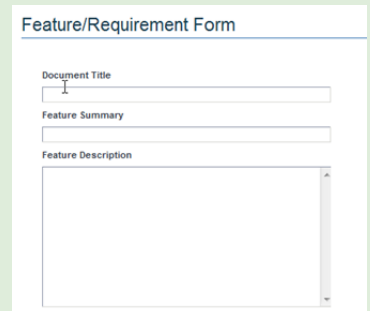
Previous Approaches to Semantic Enablement of Wikis

There have been a number of semantic wiki initiatives that have attempted to tackle some combinations of the limitations outlined above. They can be categorized into two groups, each with their own limitations:

- **Semantic wikis** that are built natively from ground up do not offer much to organizations that have existing investments in wikis, unless they migrate their content and adopt the new wiki technology. They also prevent users from leveraging the community-level support and feature extensions available in widely popular wikis.
- **Semantic extensions**, while supporting popular wikis, have taken an approach to making the wiki itself a semantic repository, as opposed to one of many sources of semantic content. While this provides an effective mechanism for creating 'knowledge' out of the 'relatively unstructured' wiki content, it still limits the users from integrating the wiki content with other applications. This integration needs to occur both ways, i.e. specific and semantically relevant wiki content be readily accessible to external applications, as well as wiki able to incorporate and integrate its semantics with external applications. These extensions are also inherently specific to a particular wiki and do not offer a mechanism of portability across different wiki technologies.

zAgile Wikidmart's Semantic Enablement of Wikis

zAgile Wikidmart's approach to semantic enablement of wikis addresses the natural limitations of wikis outlined earlier, and it also takes an architectural approach that overcomes the limitations of other implementations that strive to provide similar capabilities. The architectural benefits extend the ability of Wikidmart to become an enterprise information portal, or "information collaboration dashboard." zAgile provides the ability to semantically enable popular wikis, allowing users to:



Wikidmart enables you to create semantically enabled forms within your wiki



Content may automatically linked with external systems. This example shows links automatically connected to Jira

- Leverage their existing wiki-based content repositories
- Easily capture semantic annotations of existing content
- Create new content using semantic templates and semantic forms
- Draw information from other applications and tools into the wiki

zAgile's architectural approach:

- Separates the semantic repository from the wiki content so that the wiki functions not as the central and sole repository for both unstructured content and semantic data but as one of many applications that contributes semantically relevant data to it. This also allows users to have distributed and/or federated semantic databases.
- Facilitates two-way integration between wiki-based content and external applications that are also creators and consumers of related information. That is possible because the semantic repository is accessible to all applications, including the wiki, using the same interfaces. This level of semantic integration between the wiki and other applications provides a richer 'knowledge repository' that is capable of unifying an environment of disparate and heterogeneous applications and processes.

These extensions can be developed for any commercial or open source wiki, provided that the wiki supports a means for development of user extensions.

zAgile's solution for semantic enablement of wikis consists of the following high-level functional components:

- zAgile's Semantic Repository consisting of a set of ontologies and metamodels specific to a domain of interest. They provide a highly structured framework for annotation and capture of semantically relevant data. They also facilitate integration of this data across related metamodels, as well as allow for potential application of inferencing and reasoning for implied categorizations. The current offering focuses on the domain of software engineering and provides metamodels and ontologies pertinent to representing software processes and methodologies. Ontologies for other domains can also be similarly and appropriately implemented in this repository.
- zAgile's Semantic Interface Layer provides wikis with access to this repository. Via connectors, all applications and consumers of the repository can use a consistent API to access the semantic repository. This layer also supports standard semantic web query languages like SPARQL which can be embedded within wiki pages for querying and navigating the semantic graphs in the repository.
- zAgile's Wikidmart Semantic Plugin for Confluence provides the interface between Confluence and the zAgile Semantic Layer and supports macros for creating templates and forms in Confluence for annotating wiki pages. Templates are based upon the ontologies and metamodels defined in the semantic repository. They allow annotation of existing content and creation of new forms and pages.

About zAgile

Founded in 2006, zAgile is the open source leader in information collaboration. zAgile has solved one of the most long-standing and perplexing problems in collaboration: the problem of integrating teams, tools, processes, and knowledge. zAgile significantly reduces costs of collaboration on projects, instills consistency of methodologies across all projects, and instills confidence and predictability of delivery schedules.

zAgile Partners



World Headquarters

zAgile, Inc.
 101 California Street, Suite 2450
 San Francisco, CA 94111
 +1 (415) 946-8910 (Office)
 +1 (415) 358-5924 (Fax)
www.zagile.com
info@zagile.com