

Wiki i skolen

Christian Rudi Zachariasen

Master i informatikk
Oppgaven levert: Mai 2010
Hovedveileder: Terje Rydland, IDI

NTNU 2009/2010

Wikis in Schools

Discovering the source of motivation in
collaborative learning.

Christian Zachariasen

Problem Description

The master's thesis follows the analysis, introductory work, deployment, development and research project that has been performed during the master's program at NTNU in the fall of 2009/spring of 2010.

Abstract

Children growing up today generally have a decent amount of computer experience. In schools, however, this experience is seldom put to good use – some work is done on computers with word processors, but most students still take notes using pen and paper. Helping young children acquiring vital internet skills is becoming more and more important, as the world of information just keeps expanding at an ever-increasing rate.

In this study, an experiment was carried out where a custom-produced Wiki was installed on a school server, with the aim of monitoring and improving on the software as students used it. An attempt was also made to model the structure of contribution and how social networks form on larger and more established social websites in contrast with our implementation. In the paper, the differences between these is discussed. In addition, the motivation for contributing to such sites without receiving any monetary value in return is explored. The goal of this paper is to find a better way of using Wikis in schools.

Preface

The software development and research described in this Master's Thesis was performed during the fall semester of 2009 and the spring semester of 2010 at the Norwegian University of Science and Technology, NTNU.

The work was conducted by a student of Informatics with the supervision of assistant professor Terje Rydland. I would like to thank Margaret Tangen, Per Johnny Teigene, and all the students in class 10A at Orkanger Ungdomsskole for letting me use their study time for my experiments.

Table of Contents

Problem Description	2
Abstract	4
Preface	5
Introduction	10
The Wiki	10
A History of Wikis	10
The Memex	10
The ZOG system	11
Structure	12
The Basic Type: an Article	12
The link	13
Special pages	14
Security and Philosophy	15
Sabotage	16
Adopting the Wiki model	17
Research questions	18
Method	20
Previous research	24
Children and learning	24
Collaborative learning	24
Traditional model	24
Modern research on Wikis in schools	25
Motivation in collaborative projects	27
The social component of social networks	30
Structure of contribution and why it works	31

Social networks and their influence on knowledge systems	32
Bringing the homework “back to the school”	33
The project	35
Deployment	37
Implementation.....	39
Exploration of the existing product	40
Database agnosticism	41
Most common operations	41
Plug-ins	42
Further development.....	44
Project deployment versions	44
Unit tests	44
Source code analysis	44
First data collection	47
Questionnaire	47
First interview.....	48
Second data collection	51
Sabotage	51
Core group.....	53
Other usage statistics	53
Third data collection / push for quality improvement	53
Uncategorized	54
Categorized.....	54
Fourth data collection	56
Results	59
Motivation	59
Sabotage	60
Content.....	61
Content vs. motivation	61

Similarities between programming projects and the Wiki system	61
Content ownership	62
Incentive.....	64
Models for Wiki growth.....	68
Exponential model.....	68
Logistic model	69
Wiki configuration	71
User account management	71
Further steps.....	71
Transitioning a school account system to a Wiki account system	71
Making it easier to create new accounts.....	72
Anonymous accounts	72
Automatic feeds.....	72
Conclusion	74
Sabotage	74
Research questions	74
Would there be any motivation to use the Wiki as opposed to the traditional way of pen and paper?.....	74
How can we motivate users to not only do what they have to do (i.e. homework), but maintain articles on a voluntary basis?.....	74
Is it possible to identify a core group of students who are active when it comes to maintaining the site?	74

Part 1

Introduction

Introduction

The Wiki

Today, Wikis are used for just about anything one could imagine. Wikipedia, the free encyclopedia might be the best example, and functions as a simple and quick web-based lexicon. The name Wikipedia is combined from the two words *Wiki* and *Encyclopedia*, with *Wiki* being a Hawaiian word meaning "quick" or "fast".¹ In the context of the web, the word *Wiki* has the following definition²:

A wiki is a website that uses wiki software, allowing the easy creation and editing of any number of interlinked Web pages, using a simplified markup language or a WYSIWYG text editor, within the browser.

Wikis are often used to create collaborative websites, to power community websites, for personal note taking, in corporate intranets, and in knowledge management systems.³

A History of Wikis

The Memex

One might think that the invention of the Wiki is a modern occurrence, but this is not completely true. Given the fact that the Wiki technology does not add anything revolutionary to our "normal" way of browsing the web, and considering that the definition of a Wiki is relatively loose, we can trace back the history of our current Wikipedia to the concept of the *memex* proposed by Vannevar Bush in 1945:⁴

The owner of the memex, let us say, is interested in the origin and properties of the bow and arrow. Specifically he is studying why the short Turkish bow was apparently superior to the English long bow in the skirmishes of the Crusades. He has dozens of possibly pertinent books and articles in his memex. First he runs through an encyclopedia, finds an interesting but sketchy article, leaves it projected. Next, in a history, he finds another pertinent item, and ties the two together. Thus he goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to a particular item. When it becomes evident that the elastic properties of available materials had a great deal to do with the bow, he branches off on a side trail which takes him through textbooks on elasticity and tables of physical constants. He inserts a page of longhand

¹ "Wiki". <http://en.wikipedia.org/wiki/Wiki>. Retrieved 31 May, 2010.

² "Wiki". <http://en.wikipedia.org/wiki/Wiki>. Retrieved 31 May, 2010.

³ "Wiki". <http://en.wikipedia.org/wiki/Wiki>. Retrieved 31 May, 2010.

⁴ "As We May Think". Bush, V. (1945). <http://www.theatlantic.com/doc/194507/bush/4>. Retrieved 31 May, 2010.

analysis of his own. Thus he builds a trail of his interest through the maze of materials available to him.

When the concept of the memex was first invented in 1945, there was of course no such thing as the Internet. Bush's proposal assumed that the sources of information, that is the pictures, texts, comments and so on would run on a mechanical device. Thus, instead of there being one global memex where thousands of users would contribute to millions of different content pages⁵, Bush expected each user to have their own memex and purchase readily-available cartridges of microfilm that they could use to fill it with information. In this way, each user would have a personal memex – presumably applicable only to the subject matter that they wished to research. To draw a parallel to today's Wikipedia, each user can indeed only study the subject that they want to (for example, Turkish bows) by reading up on the Turkish bow⁶ article, and continuing their study with that article as their base. If the user would like to read up on how it compared to, for example, a Mongol bow, he would not have to purchase additional memex tapes – he could simply click on the link and the article would be available. The user could also insert his own notes in the same way as Bush proposed – although those notes would probably be edited and adjusted so that they fit into the Wikipedia article style⁷.

The ZOG system

Another concept that is worth mentioning is the ZOG system, a collaborative hypertext database system developed at Carnegie Mellon University in the 1970s. Without going into the details of this system, it is important to consider that it was in this system the “frame” or “card” model was used, later popularized by HyperCard, an application made by Bill Atkinson for Apple Computer.⁸ HyperCard is another step on the road towards the Wikipedia we know today, and was in fact much of the inspiration for the first Wiki created by Ward Cunningham⁹:

Cunningham [made a card that] had three fields: Name, Description and Links. The fields in HyperCard were WYSIWYG editors, but linking was a pain that involved moving between both cards. Cunningham abandoned regular stack links and used search-on-demand instead. Normally one would type links into the Links field. When using the card, each link had a button that would take you to the card if it existed, or beep otherwise. If you held the

⁵ “Special:Statistics”. <http://en.wikipedia.org/wiki/Special:Statistics>. Retrieved 31 May, 2010.

⁶ “Turkish Bow”. http://en.wikipedia.org/wiki/Turkish_bow. Retrieved 31 May, 2010.

⁷ “Wikipedia:How to Edit a Page”. http://en.wikipedia.org/wiki/How_to_edit_a_page. Retrieved 31 May, 2010.

⁸ “HyperCard”. <http://en.wikipedia.org/wiki/HyperCard>. Retrieved 31 May, 2010.

⁹ “History of Wikis (The influence of Hypercard on Wiki inventor Ward Cunningham)”. http://en.wikipedia.org/wiki/History_of_wikis#The_influence_of_HyperCard_on_wiki_inventor_Ward_Cunningham. Retrieved 31 May, 2010.

button down, it would relent and go make the card for you. (One can recognise here the traditional wiki feature by which a new page is opened for editing whenever one clicks on any new word formed in camel-case. In Wikipedia the equivalent feature is called "red links".

There are some differences between these systems compared to the Wikipedia we know today. The Memex and the ZOG system assumed that the application was personal to the user (or limited group of users). The user would buy a copy that was his, and sharing the information with other users was difficult. This assumption was further reinforced by the fact that, in the Memex, the user would purchase proprietary containers of information regarding the subject matter he wished to study, to load his Memex with. ZOG was a collaborative knowledge-based system, but limited itself to the information that the group of users wanted knowledge about. Wikipedia makes no such distinction between subject matters – it “lives” on a huge database, with all main articles available to all users.

But the main difference between a modern Wiki and the concepts or systems invented earlier is that, though some of the older systems focused on collaboration, they did not run on a popular or standardized protocol. Much of the power of the modern Wiki is that they run on the hypertext protocol, allowing them to be easily set up and used. In addition, mostly anyone has access to a computer with internet access nowadays. This fact vastly increases the accessibility of modern Wikis as compared to the older systems - all the user needs is a web browser and he can interface with the Wiki.

Structure

The structure of the Wiki model is incredibly simple, yet extremely powerful. Wikipedia will be used as the example throughout the explanation of the model.

The Basic Type: an Article

An article in Wikipedia is defined as *a page that has encyclopedic information on it*.¹⁰ The article is the basic unit of information in a Wiki, and could be viewed as a vertex in a directed graph, with the edges between vertices being the links of the Wiki article¹¹. Usually, most of the entries in a Wiki that are accessible to users are articles. They can contain text and images. Text is inserted into the article by writing words and sentences surrounded by a simple *markup language*. Images are uploaded to the Wiki's file storage and are put into the article by using a special *tag* provided by the markup language syntax that describes to the article parser that an image should be placed in that position when the article is rendered to

¹⁰ “Wikipedia:What is an Article?”. http://en.wikipedia.org/wiki/Wikipedia:What_is_an_article%3F. Retrieved 31 May, 2010.

¹¹ Capocci, A., Servedio, V. D. P., Colaiori, F., Burial, L. S., Donato, D., Leonardi, S. and Caldarelli, G. *Preferential attachment in the growth of social networks: the case of Wikipedia*. http://www.inf.ufrgs.br/~burial/papers/Physical_Review_E_06.pdf. Retrieved 31 May, 2010.

the browser. Several other things can be placed into an article using different tags, for example:

- The ref tag to indicate that a footnote should be placed here pointing to a reference at the bottom of the page.
- The table tag, which indicates that a structure for holding tabular data should be inserted into the text.
- The math tag, which supports inserting LaTeX-style mathematical formulas in the text. An example is the formula `$\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$` which

turns into
$$\frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}}$$
 when rendered through the Wikipedia markup engine.



The article as a basic unit of information.

An article is, by definition, a completely dynamic object in the Wiki model. Its contents are defined by user-made contributions, and the only things that remain static when viewing an article is the frame around the article text, and the way the article is rendered to the browser.

The link

One of the concepts within a Wiki which is of immense importance to the structure of the Wiki as a whole is the link. A link in a Wiki behaves much in the same way as a normal HTML hyperlink, and is designed in the way that we would link to something when writing a web page. It behaves the same in the way that, when clicked, it leads the user to a different page. Links can be either internal or external, meaning that they can either point to another page within the space of the Wiki, or to a page that might be anywhere else on the web. Users add

links to pages by using the special *brackets* tag. The brackets can describe the location of the page that the link points to and an optional description of the link by using the following syntax:

[LaTeX] will link to the internal page on LaTeX.

[click this link|LaTeX] will do the same, but the link title will read as “click this link” when viewing the page on which the link has been put.

The link concept as used in Wikis is explored further when discussing the difference in the mental model one may have of how a link normally works, versus how a link in a Wiki normally works.

Special pages

A Wiki needs a way to display pages that are *static* in the way that it is pre-defined by the Wiki code. A user cannot make edits to these pages, and they are named *Special pages* because they usually help an administrator or normal user to fulfill a task in some way. There are several examples of such special pages:

- An “All articles” page
- A search field or page that allows the user to search for articles
- The history log for a page

# Revisjon		Bruker
	Aktuelle Versjon - 12/02/2010 09:46	hakongi
16	12/02/2010 09:41	hakongi
15	11/02/2010 13:58	hakongi
14	11/02/2010 13:50	hakongi
13	11/02/2010 13:43	hakongi
12	11/02/2010 13:43	hakongi
11	11/02/2010 13:21	hakongi
10	11/02/2010 13:21	hakongi
9	11/02/2010 13:15	hakongi
8	11/02/2010 13:15	hakongi
7	11/02/2010 13:15	hakongi
6	11/02/2010 13:15	hakongi
5	11/02/2010 13:15	hakongi
4	11/02/2010 13:04	hakongi
3	11/02/2010 13:04	hakongi
2	11/02/2010 13:04	hakongi
1	11/02/2010 13:03	hakongi
0	11/02/2010 13:00	hakongi

Security and Philosophy

It is hard to describe how a Wiki works without mentioning the underlying philosophy, and thereby the security that is used to ensure that the quality of the information available is reasonably high. This is to ensure that people who only want to read articles “get what they came for”. A person can enter something into the search field of Wikipedia, get more information on the subject he or she is interested in, and leave. But that person might never even consider that by reading an article they are made a user. Not in the classic sense that a user has a username and a password to authenticate themselves and log in to the site, but that they are always provided with the possibility to make changes to the text that they are consulting.

At the core, a Wiki is readable and editable by all users, even unauthenticated users. This means that anyone can enter an article on Wikipedia; click *Edit this page*, and write whatever they want. They can delete all the text in an article, vandalize it by writing whatever they want, and generally mess up the information contained within so that it is no longer readable or accessible to users.

This is the core philosophy of Wikis – they are collaborative information databases that encourage anyone to contribute. It should be noted that Wikis generally do *encourage* users

to register and create a user name and password in order to authenticate themselves in the future, and most active users do

When a user makes an edit to an article without being authenticated, an entry is saved in that article's *history log*, along with information on what changed, at what time the edit was made, and the user's *IP address*. An IP address is a set of numbers that can be used to contact the computer that is being used. A user can create an account if he wishes to. In doing this, his name will be on every edit that he makes.

Over time, some users are promoted to what Wikipedia calls *sysops* (**system operator**) or Administrators. These users receive the rights to use restricted technical features like page protection, deletion and blocking. *Protection* of a page entails that an administrator *locks* the page for further editing. This usually happens when a large number of users want to collaborate on an article, and is relatively common on political articles or articles regarding current events and news. By locking the page, no users can edit it until an administrator *unlocks* it. *Deleting* a page means that the article and all entries in that page's history logs are removed from the Wiki. Usually a backup exists so that the article can later be restored. To *block* means that an administrator can choose to *ban* a user from editing articles in the Wiki. This feature is commonly used when a user vandalizes one or several articles.

Sabotage

In Clay Shirky's words¹², the most complex question about Wikipedia's continued success is "How does it survive both disagreement and vandalism? Openness, division of labor, and the multiple motivations of its users drive its rising average quality, but none of those things explain why articles on contentious subjects aren't damaged by editing wars among rival factions, or simply destroyed by vandals, who can delete an entire article with the click of a button. Why don't these sorts of things happen?"

The *Tragedy of the Commons* is an interesting concept to describe here. According to Wikipedia, the Tragedy of the Commons "describes a situation in which multiple individuals, acting independently, and solely and rationally consulting their own self-interest, will ultimately deplete a shared limited resource even when it is clear that it is not in anyone's long-term interest for this to happen".¹³

When there is no common feeling of idealism and self-interest that would help sustain environments that are vulnerable to the tragedy of the commons, it is intuitive that an

¹² Shirky, C., *Here comes Everybody – How change happens when people come together*. Penguin, 2009. p. 135.

¹³ http://en.wikipedia.org/wiki/Tragedy_of_the_commons

authority or federation is required. “In a typical example, governmental regulations can limit the amount of a common good available for use by any individual.”

The closest we get to such rule-enforcing on Wikipedia is the possibilities that administrators have to “lock down” an article. This closes the article for further editing. According to Shirky, articles on subjects such as Islam and abortion often have vandalism or even complete deletion applied to them. Articles on contentious subjects like this are subjected to locking the most. At any given point in time, around 0.5% of Wikipedia’s articles are subject to lockdowns.¹⁴

Adopting the Wiki model

The Wiki model is an important basis for making a user able to understand how to make use of the technology. Many users have difficulties adapting their mental model of an encyclopedia to the Wiki model. As an example, the hyperlink on a Wikipedia page:

History

Main article: History of wikis

WikiWikiWeb was the first site to be called a wiki.^[6] Ward Cunningham started developing WikiWikiWeb in 1994, and installed Airport counter employee telling him to take the “Wiki Wiki” shuttle bus that runs between the airport’s terminals. According to Cunningham was in part inspired by Apple’s HyperCard. Apple had designed a system allowing users to create virtual “card stack change one another’s text”.^{[2][9]} In the early 2000s, wikis were increasingly adopted in enterprise as collaborative software. Com wikis as their only collaborative software and as a replacement for static intranets, and some schools and universities use wikis On March 15, 2007, wiki entered the online Oxford English Dictionary.^[10]

In this image, one can see several different links. Let’s examine the link pointing to the *schools* article as an example. A user with some amount of internet experience, especially if he or she has read blog articles, would look at the link to *schools* in the context of history, and intuitively think that the link would point to a list of schools that use Wikis. It is normal to think of links on the internet as a way to get to a page that shares some kind of context to the page from which we are coming.

The Wikipedia model works in a different way – the link examined above actually leads to the Wikipedia article concerning schools, that is, an explanation of what a school actually is. In this way, a Wikipedia link is normally context-less. In other words, if a link can be viewed as a directed edge from vertex A to vertex B, the edge *does not carry any information concerning vertex A to vertex B*.

If one is to introduce the concept of Wikis to users with varying degrees of competency regarding technology, one needs to be sure that a common way of writing and maintaining

¹⁴ Shirky, C. (2008). p. 137

articles is shared between the users. A set of best practices should be conveyed to the users, in order to stimulate a structured understanding of what to do at different points in the article writing process, and why they should do these things. It is indeed possible to introduce different standards for writing than Wikipedia uses, in the case that other standards suit the style of the Wiki better. But in order to eliminate any unwanted variables from the research experiment, we chose to go for the Wikipedia standards as much as we could.

Let us examine the idealistic and practical model that a Wiki depends on. It is important to note that some of the points mentioned here are not rigid laws that need to be followed with regards to implementing a Wiki, they are simply modifying guidelines that point to the way that modern, successful Wikis are being used today. For example, the point that all users should be able to edit all content is a pointer to the openness of the model, not a requirement for a Wiki's operation.

What does this entail when seeing the Wiki implementation in the context of a school setting? It is obvious that we're not looking to replace the use of Wikipedia in the school – Wikipedia is still a very powerful tool students can use in their exploration of their subject matter. In addition, it would not be possible for the students using the local Wiki implementation to in any way keep up with the development on the official Wikipedia – there are just too many contributors to Wikipedia.

We are not looking to replace the students' textbooks either, as each text book will contain different views and ways of writing about a subject that may still be valuable to the student's learning of the curriculum. What we are trying to do is to insert the Wiki as an external tool to help in the student's understanding of the subject matter. Specifically, through encouraging the students to write about what they are learning, we are hoping to make them understand the subject matter better. By editing other student's contributions and commenting on what is good and what's not so good, along with correcting sub-par articles, we hope to encourage students to gain an understanding of the idealistic side of using Wikis, or knowledge-based repositories in general.

Later in the text, we will be taking a look at how different study methods have an effect on the student's understanding.

Research questions

My research questions were as follows:

- Would there be any motivation to use the Wiki as opposed to the traditional way of pen and paper?

- How can we motivate users to not only do what they have to do (i.e. homework), but maintain articles on a voluntary basis?
- Is it possible to identify a core group of students who are active when it comes to maintaining the site?
 - Can we find a pattern in their usage of the site to more closely identify them?
 - Are there similarities between the core group here and core groups on other Wikis?

One problem today seems to be that, for adolescents, there is little or no integration with new technology in the school, while at the same time more and more learning is performed at home in front of the computer, via the internet. It is important to stay relevant in our digital age, and students do not always learn how to work with computers at school. By putting technological education into another, more structured perspective; one might be able to take advantage of the motivation that arises when a student learns something new, while it can lead to higher and more focused productivity when students use other, perhaps related technologies.

There could also be an underlying factor when it comes to the teachers in schools. Some seem unwilling to accept change at the pace that digital technology is changing. Some technology that was brand new only five years ago feels completely outdated and useless today.

I hope that this master's thesis shows that some of the problems mentioned above can be solved using Wikis. Being a relatively modern technology, Wikis seem to be intuitive in the way that a user does not have to be very experienced with computers and the Internet in order to start using them. It might take a while before users feel that they are working with Wikis in the "right" way, but with the plethora of examples on how to "do it right" that are available online (see ¹⁵, ¹⁶, and ¹⁷), interested users soon get the hang of it. In the case of using a Wiki in Orkanger ungdomsskole, it has been presumed that the students will learn the technology on their own, but with a certain amount of guides and examples written for them to help them on their way.

From a technological perspective, a Wiki is relatively trivial to adapt to a school's needs. One of the problems with former studies on the use of Wikis in schools¹⁸ has been that the

¹⁵ "HowStuffWorks: How Wikis Work", <http://computer.howstuffworks.com/internet/basics/wiki.htm>. Retrieved May 31, 2010.

¹⁶ "Wikis in Plain English", <http://www.commoncraft.com/video-wikis-plain-english>. Retrieved May 31, 2010.

¹⁷ "Why Wiki Works", <http://c2.com/cgi-bin/wiki?WhyWikiWorks>. Retrieved May 31, 2010.

¹⁸ Kuznetsov, S., "An Informal Survey of New York University Students", April 2006

researchers have not had any possibility to change the Wiki depending on the user's needs. With regards to the version of the software that was rolled out on Orkanger ungdomsskole, software was chosen that had been in development for several years. This means that the software could be presumed to be mature in the sense that most of the software errors, or *bugs* typical with very young pieces of software had been removed. It was very helpful to be able to start using something instead of having to start from scratch, writing a new implementation of the Wiki model – this could have led to so much work that the process of deploying and researching the Wiki as it was being used would have been impossible.

The reason for the software being easy to modify was, as mentioned before, that the software had been licensed under the GPL. This provided an extra dimension of flexibility when it came to the research – it became possible to not only be able to monitor a human-machine interaction process, but observing the effects of any changes made directly to that process.

Method

From the time that planning started, the project was going to be based on the Design and Creation methodology as defined by Oates (p. 108). Design and Creation follows five basic guidelines:

- **Awareness** is the recognition and articulation of a problem, which can come from studying the literature where authors identify areas for further research, or reading about new findings in another discipline, or from practitioners or clients expressing the need for something, or from field research or from new developments in technology.
- **Suggestion** involves a creative leap from curiosity about the problem to offering a very tentative idea of how the problem might be addressed.
- **Development** is where the tentative design idea is implemented. How this is done depends on the kind of IT artifact being proposed. For example, an algorithm might need the construction of a formal proof. A new user interface embodying novel theories about human cognition will require software development. A system development method will need to be captured in a manual that can then be followed in a systems development project. A new approach in digital art might require the development of an art portfolio tracing the development of the artist's creative ideas.
- **Evaluation** examines the developed artifact and looks for an assessment of its worth and deviations from expectations.

- **Conclusion** is where the results from the design process are consolidated and written up, and the knowledge gained is identified, together with any loose ends – unexpected or anomalous results that cannot yet be explained and could be the subject of further research.

The software project was to be released in three different versions, with post-release interviews taking place after each deployment. In addition I would host a teaching session after each version educating students on how to use the new features that had been developed in the previous release.

For my interviews I decided to use a semi-structured approach where I wrote up some questions regarding the usability and general thoughts on how the Wiki technology worked, and would ask the interviewee follow-up questions to further elaborate on a subject which I found interesting. I would not necessarily interview all the students in the class, but pick the ones which I, through general observation on the Wiki, could see had either performed above average amounts of work, or less than average amounts of work. The variable of how much work a certain student had performed was measured purely in quantity, not bringing in the variable of quality of work into the measurements. The reason for this is discussed by Chay Shirky, writing about the photo-sharing service *Flickr*:

“Given that everyone now has the tools to contribute equally, you might expect a huge increase in equality of participation. You’d be wrong. (...) The Wikipedia articles for asphalt had 129 contributors making 205 total edits, but the bulk of the work was contributed by a small fraction of participants, and just six accounted for about a quarter of the edits.”¹⁹

What we can surmise from this, and other examples given by Shirky, is that the simpler the actions are, the better. Consider the example of Nupedia, an initiative founded by Jimmy Wales, creator of Wikipedia. Nupedia was an English-language Web-based encyclopedia whose articles were written by experts and licensed as free content. (...) Nupedia lasted from March 2000 until September 2003, and is mostly now known as the predecessor of the free wiki encyclopedia, Wikipedia²⁰. This is how things turned out when the medium required *experts* to write the articles, instead of amateurs:

“(...) Since anyone can act, the ability of the people in charge to kill initiatives through inaction is destroyed. This is what befell Nupedia; because everyone working on that project understood that only experts were to write articles, no one would even begin an article they

¹⁹ Shirky, C. (2008). p. 123.

²⁰ “Nupedia”, <http://en.wikipedia.org/wiki/Nupedia>. Retrieved May 31, 2010.

knew little about, and as long as the experts did nothing (which, on Nupedia, is mostly what they did), nothing happened.”

It seems that simply lowering the bar for when a user is allowed, and indeed, *encouraged* to contribute to the repository is enough. Wikipedia seems to work because the users are the experts, or rather that there are no experts, only users. Thus, a clear goal of this study must be to enable the students in the group to think of themselves as the authority on the articles they write.

Also, in order to make sure that the knowledge repository remains self-maintaining, it seems necessary to introduce some factor that serves as the replacement for *critical mass*. Critical mass is discussed later in the paper, but in simple terms it can be seen as *the amount of users that gets the ball rolling*. In other words, in a self-maintaining project it is necessary to recruit as many users as one needs so that the effects of simple actions build on themselves. One example is the one given above regarding asphalt – one person creating a stub article on asphalt was enough to get the ball rolling, and the reason that this was possible was critical mass.

Part 2

Prestudy

Previous research

Children and learning

Lev Vygotsky, a Soviet psychologist, thought that intellectual development and thought was deeply rooted in social activity²¹. The social aspect of education and development is the most important factor to consider, while the individual aspect is less important. Vygotsky, in this manner, had a socio-cultural view of education. In order to adjust this view to the idea of learning through collaborative tools, it is possible to view a Wiki as a social framework for education: Either a draft is produced by the individual, and then further developed by the group, or the entire product is developed by the group. This further leads to all the individuals in the group achieving learning through social interaction. One can see here how the basic thoughts of the socio-cultural model has a place in the use of Wikis in schools.

We can also try to apply the John Dewey “Learning by doing” principle to the idea of Wiki learning. Dewey said that the active collaboration by the individual is the central requirement for learning. When we encourage the students to improve their product through collaboration – in this case by the act of writing or helping to write an article – we enhance the participation of the individual and help the student to learn more effectively.

Collaborative learning

Collaborative learning in general refers to a group of people working together to achieve a common goal, as opposed to a number of individual working by themselves to achieve a common goal. What follows is a discussion of the traditional model of collaborative learning, and how the Wiki model fits in.

Traditional model

Cooperative/collaborative learning is interactive; as team members the students:

- Develop and share a common goal
- Contribute their understanding of the problem: questions; insights and solutions
- Respond to, and work to understand, others' questions, insights and solutions. Each member empowers the other to speak and contribute, and to consider their contributions
- Is accountable to others, and they are accountable to the student
- Is dependent on others, and they depend on the student

²¹ Imsen, G. *Ellevens verden. Innføring i pedagogisk psykologi*: Universitetsforlaget, 2005.

Modern research on Wikis in schools

Generally speaking, previous research seems to base itself on the use of existing wikis. The tendency seems to be that the studies explore the differences between traditional learning and learning with the help of Wikis – *collaborative* or *cooperative* learning. This means that the studies that base themselves on these pillars need to consider the fact that the Wiki model in the way that it is used in whatever Wiki software they use needs to be followed, with the positive and negative aspects that might be connected to that model. When the context is changed from being a pure encyclopedia like Wikipedia and many other Wikis are, and into an educational platform, there is the possibility that one might lose a little bit of the positive aspects such a model has to offer.

Many other studies bring up a point which was not handled in this one – control groups. In many studies, the control groups are used to explore the differences between the group where the researchers modify the environment (in this case, introduce a Wiki to facilitate collaborative learning), and a group where no change has been made (a group that continues with traditional learning). Whether a control group would be interesting to use in this study is something we will not debate any further here, but considering the fact that the only group that participated in the study used a Wiki that was continually changing relative to the group's needs, it is possible to compare the productivity and learning results of the modifications by measuring these values between software iterations. For example, one can measure the educational value of the software using traditional "tests", hosted on the Wiki. These traditional tests would then try to pinpoint both learning in general, and whether the users have learned to apply the new techniques that were introduced at the last point of software deployment.

In one study²², a series of post-project interviews were performed, in which a control group was used to explore how the students felt that the Wiki project had gone. The researchers asked the students what kinds of experiences they had had with traditional learning versus collaborative learning. The results were overwhelmingly positive to the Wiki way of education – 12 students thought that the experience had been a positive one, while 6 students thought it had been negative. 3 students thought the traditional way of learning was sufficient, while 16 thought it was not sufficient. The researchers of the study do not provide any reason that

²² de Pedro, X., Rieradevall, M., López, P., Sant, D., Piñol, J., Núñez, L. and Llobera, M., *Writing documents collaboratively in Higher education using Traditional vs. Wiki methodology*, http://uniwiki.ourproject.org/tiki-download_wiki_attachment.php?attId=98&page=Uniwiki-Congressos. Retrieved May 31, 2010.

the students seemed so negative to the traditional way of teaching, but they do mention some reasons that the students in the Wiki group were so positive to it (p. 10):

- It gave them an opportunity to collaborate without having to travel to meet.
- It let them observe the development in the other students' work.
- It was a dynamic communication protocol.

The negative feedback regarding the Wiki mostly concerned some students not being able to connect to the internet. Another reason given by the students was that it was hard to learn the syntax of the Wiki markup language.

In the same study, a set of *best practices* for working with a Wiki is mentioned. The authors, among other things, talk about users often being nervous about the fact that everything is so open on a Wiki – they are afraid to think someone will read what they are working on before they are finished with it. Traditionally, a student's work is often restricted to that student, and follows a pattern like the following:

1. The student performs the work.
2. The student hands in the work to the teacher.
3. The teacher evaluates the work and gives the student feedback on the quality of the work.
4. The student receives the evaluation and the feedback.

When a student follows this process, he or she normally does not need to be afraid that other students will see his work and evaluate it. Many people worry about being exposed to criticism because the work is sub-par, or feel that their contributions are worth less than what other students' contributions are worth.

When one crosses over to using the Wiki model for education, the roadblock between individual students disappears. The process changes in the following way:

1. The student performs the work, and saves it. Note that at this point, the work is completely accessible to all other users – students and teachers.
2. The teacher evaluates the work and can provide written or oral feedback.

It is important to note that the Wiki process almost mirrors the traditional group process in the school. In traditional group work, it is normal to use a group of students that collaborate in making a product that is then performed in plural or delivered to the teacher. There are, however, two main differences between this process and the Wiki process:

- Traditionally, it is unusual that a student directly changes another student's work, that is edits his text. The group's product is, in this way, often a mix of several students' work, resulting in a product where each individual part is different from the previous part. A wiki makes accessible all parts of the product for editing, and even encourages it if it can provide a better end result. This is how most articles at Wikipedia are built up; through successive collaboration and "helping" an article is made. This then results in a "final" product that is the deliverable.
- The process has traditionally been somewhat opaque to the teacher or project leader – the students work on a project, and the teacher might be involved in some way with the evolution of the final product, but in a limited way. With the Wiki process, every step and transformation of a draft is documented with the use of the history logs. This allows the teacher to watch how the final product evolves from start to finish.

The first point mentioned here will be termed the "one user per article trap". This is a phenomenon that has been witnessed in other Wikis – the number of contributors to an article seems to correlate with how "exotic" the subject matter is. Consider the *asphalt* example, from earlier in the paper. Asphalt is a relatively common concept with which most people have some kind of relationship. One could even argue that it would not take much effort to read up on asphalt from other dictionaries and fill in a summary of that information into the article on Wikipedia. Now consider another article, on Hontanar, a municipality located in the province of Toledo, Spain²³. Admittedly, the article's very specific subject matter has been chosen in order to prove a point, but this is representative of many articles on Wikipedia. The Hontanar article has only one contributor. There are several more edits, mainly done by so-called bots, which go through articles fixing normal mistakes. It would be hard for most people to contribute to the Hontanar article simply because they don't know anything about it. Similarly, it is unlikely that there are many sources on the subject matter available through the internet or in published works. Articles with very few contributors tend to require experts to write about them.

Motivation in collaborative projects

The study "Motivations of Contributors to Wikipedia" written by Stacey Kuznetsov²⁴ discusses the reasons why some people decide to contribute their knowledge to a site without getting any reward that can be measured in money back. The study also touches upon Wikipedia contributions and their relation to open source projects. In addition, my group in Experts in Team (EiT) in 4th year on NTNU examined open source group and how motivation to

²³ "Hontanar", <http://en.wikipedia.org/wiki/Hontanar>. Retrieved May 31, 2010.

²⁴ Kuznetsov, S., *Motivations of Contributors to Wikipedia*. <http://www.staceyk.org/personal/WikipediaMotivations.pdf>. Retrieved May 31, 2010.

contribute can be explained in such cases. These two studies will now be compared and further examined.

According to Kuznetsov, Wikipedia totaled around 1.25 million registered contributors at the time of writing (2006)²⁵. This did not include an unknown large amount of users that do not register themselves on the site. The number of named accounts (which directly translates to registered contributors) has since grown to about 12.3 million²⁶ - a *ten-fold* increase, given that the numbers are correct, in just 4 years. This is a testament to the explosion in popularity that Wikipedia has enjoyed since its start-up in 2001.

Kuznetsov continues by examining the motivations of the *stakeholders* of Wikipedia – defined in the research paper as being the users who contributes to Wikipedia in a “constructive manner”, that is, “edits that add to the integrity of Wikipedia as an encyclopedia with a neutral point of view (NPOV) and do not include malicious edits or deliberate vandalism such as false information, propaganda, or offensive content.”

The motivations of people who contribute to “no-pay” projects such as Open source projects have been studied relatively extensively (paraphrased):

The prevailing majority (over 78%) of the 2784 survey respondents in the FLOSS study²⁷ section IV) indicated that the reason why they joined FLOSS projects was to “learn and develop new skills”. The second most popular reason was to “share my knowledge and skills” (49,8%), followed by the motivation to “participate in a new form of cooperation” (34.5%).

(...) Most FLOSS developers where not driven by financial or otherwise materialistic incentives, as can be seen from the fact that reasons to “make money” and “get a reputation in the OS/FS scene” received the lowest rankings.

In the Open Source study done by the EiT group²⁸, a questionnaire was posted on a public news/discussion site, *Reddit*. The questionnaire contained questions such as what motivates people to work on open source projects. The responses were many and varied; they are summarized in the list below:

²⁵ “Wikipedia:Wikipedians”. http://en.wikipedia.org/wiki/Wikipedia_users. Retrieved May 31, 2010.

²⁶ “Wikipedia:Wikipedians (Demographics)”.

http://en.wikipedia.org/wiki/Wikipedia_users#Demographics. Retrieved May 31, 2010.

²⁷ Ghosh, R. A., Glott, R., Krieger, B. and Robles, G. *Free/Libre and Open Source Software: Survey and Study*. <http://www.infonomics.nl/FLOSS/report/Final4.htm>. accessed through waybackmachine.org because the web page had been removed. Retrieved 31 May, 2010.

²⁸ Gudal, K., Werstad, S. T., Bratland, K., Coskun, M. and Zachariasen, C. *Hva motiverer utviklere av åpen kildekode?*. (2009).

- Learning new technologies
- Being able to extend one’s CV (some employers value participation in an open source project to a very high degree)
- Helping others without charging money for their time.
- “Get a name” on the Internet. That is, being recognized as a separate person than the masses of people on the Internet.
- “Giving something back” to society or the services that they have enjoyed.
- The thought of someone else using and enjoying their work.

Admittedly, few of these reasons are attributable to the reason for user motivation behind contributing to Wikis. The last two reasons, however, play into the idealistic nature of contributors that we have and will describe further in this paper.

Kuznetsov also refers to studies done of participants in the SETI@Home project^{29 30 31} as well as a pilot survey conducted throughout the New York University Campus³². The survey results have been coded and entered into a table. The sample sizes for the FLOSS study, the SETI@Home study and the Wikipedia study were 2784, 139254 and 102, respectively. Please note that in the FLOSS survey, respondents were allowed to select multiple reasons for their motivations.

General reason	FLOSS projects	SETI@Home	Wikipedia
Furthering or sharing knowledge (for-the-good-of-humanity reasons)	49,8 %	58.51 %	48,89 %
Learning new skills (my-own-benefit reasons)	78 %	N/A	17,78 %
Giving back to the community (why-	N/A	17,44 %	15.56 %

²⁹ "SETI@home". http://en.wikipedia.org/wiki/Seti_at_home. Retrieved May 31, 2010.

³⁰ "SETI@home Poll Results". <http://seticlassic.ssl.berkeley.edu/polls.html>. Retrieved May 31, 2010.

³¹ "Why people are running SETI@home", accessed through waybackmachine.org because the SETI website was down. <http://setiathome.ssl.berkeley.edu/motivation.html>.

³² Kuznetsov., S. (2006)., p. 1

not? reasons)

Other	34 %	24 %	17,66 %
-------	------	------	---------

As is evident from the table, most respondents indicate that they are motivated to perform the kind of work they do because they subscribe to the idealistic idea that these projects convey. Very few responded that the motivation behind their work was for selfish reasons. Indeed, in Kuznetsov's words, *Most FLOSS developers were not driven by financial or otherwise materialistic incentives.*

Another interesting thing to point out is that, in the New York University Campus study, "more than 81% of the respondents stated that they would correct an error, such as a spelling or grammatical mistake, false or bias information, or offensive content in Wikipedia. Respondents were significantly less willing to correct the same types of errors in printed sources, as only 16% of respondents said they would contact the publisher if they noticed a mistake in print."³³

This result is interesting because it shows us that, if the results are generalizable to the rest of the population, readers are incredibly unlikely to notify the author of a printed text of an error. They are, however, extremely willing to correct errors if they can be allowed to do it themselves. It is assumed that this has to do with two things: the time it takes and how accessible the error-correction is.

In the case of Wikipedia, all one has to do is edit the article with an error, correct the error, and then save the article. In the case of a printed work, one has to compose a letter or e-mail to the author or the publisher, and wait for the error to be corrected.

The social component of social networks

There is always a social component to social network, however subtle or prominent that social component is relative to the system it operates within. Wikipedia has a social component to it, but one could say that the social part of Wikipedia is a rather small part of the ecosystem itself. On the other hand, as evidenced later in the text, the social component of sites like Stack Overflow or Facebook is absolutely critical for the system's well-being. Systems like this simply would not survive if not for their social interactions; they are what drive the system.

³³ Kuznetsov, p. 2

It would certainly be possible to exploit the subtle social effects of taking a group where everyone knows each other (like a classroom full of students) and putting it into a Wiki setting, more so than Wikipedia does. The reason for this is that by analyzing the group of students, we can derive some assumptions on how they will interact online based on how they interact in real life.

Structure of contribution and why it works

Clay Shirky has an excellent example of how incremental the knowledge model in Wikipedia is:

“Here’s how it works. Someone decides that an article on, say, asphalt should exist and creates it. The article’s creator doesn’t need to know everything (or indeed much of anything) about asphalt. As a result, such articles often have a ‘well, duh’ quality to them (...). Once an article exists, it starts to get readers. Soon a self-selecting group of those readers decide to become contributors. Some of them add new text, some edit the existing article, some add references to other articles or external source, and some fix typos and grammatical errors. None of these people needs to know everything about asphalt; all contributions can be incremental. (...) Every edit is itself provisional.”

Shirky goes on to say that *this works in Wikipedia’s benefit partly because bad changes can be rooted out faster, but also partly because human knowledge is provisional.*³⁴

It does, however, seem very natural that in the case of privately implemented Wikis like our one, it is necessary with some kind of *boost* early on with the simple goal of increasing the value of the Wiki in itself. The hypothesis here is that knowledge-building using collaborative networks is automatic *beyond a certain point*. It does not happen overnight – that is, one can’t set up a Wiki on anything and just expect it to grow by itself. There needs to be focus, first of all. In our case the focus is on any information that is curricular and relevant to the students that are going to be using the Wiki. In Wikipedia’s case, the focus is any information that could be relevant to anyone. If one analyzes the relative slow growth of Wikipedia after it started up in 2001³⁵, one reason that intuitively plays a role is that few people knew about the site, so few people would contribute. This is of course an important point to make. But the primary factor here is that the amount of information was so low. That is, there was no data *to build on*. As we will see, there is a need to get the proverbial ball rolling in any form of network that has a social component to it, no matter how thinly spread-out that social

³⁴ Shirky, C. (2008). p. 118-119.

³⁵ “Wikipedia:Modelling Wikipedia’s Growth (Two-phase exponential model)”. http://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia%27s_growth#Two-phase_exponential_model. Retrieved May 31, 2010.

component is; *if improving the site is dependent on its users, we need critical mass*. This is reminiscent of the chicken-and-egg problem: critical mass is absolutely necessary, for the following reasons:

- Users will feel more committed to improving a site the larger it is, because bigger means more influential and more relevant.
- Users will feel more rewarded for improving a site the larger it is, because of reason number one.
- In turn, more users will flock to the site on recommendation from fellow users.
- As a result, the site will grow larger.
- The process repeats itself.

The word “large” in this context simply means the combined amount of data and active users that a site has. This is a process that occurs in one form or another in just about any social network.

Social networks and their influence on knowledge systems

Shirky writes³⁶ that, contrary to the belief that Wikipedia works because of collectivism and an idealistic approach to the rest of society, it is argumentation that is “driving” the site:

“The people most enamored of describing Wikipedia as the product of a free-form hive mind don’t understand how Wikipedia actually works. It is the product not of collectivism but of unending argumentation. The articles grow not from harmonious thought but from constant scrutiny and emendation.”

This counters the studies done on the motivation of contributors – a large majority of respondents indicated that the reason for their involvement was indeed idealism and helping out others. Still, we will discuss the opposite argument for the sake of including it in the study.

Shirky continues: “The idea behind Nupedia was that it should be possible to improve on traditional encyclopedias by keeping the process but dropping the commercial aspect. This turned out to be a bad idea, because much of the process for creating a traditional encyclopedia has less to do with encyclopedias than with institutional imperatives. Once you dispense with the institutional dilemma, as Wikipedia does, it is possible to dispense with much institutional process as well. (...) Wikipedia (...) does not have to be efficient – it

³⁶ Shirky, C. (2008)., p. 139.

merely has to be effective. If enough people see an article, the chance that an error will be caught and fixed improves with time. Because Wikipedia is a process, not a product, it replaces guarantees offered by institutions with probabilities supported by process: If enough people care enough about an article to read it, then enough people will care enough to improve it, and over time this will lead to a large enough body of good enough work to begin to take both availability and quality of articles for granted, and to integrate Wikipedia into daily use by millions.”

Bringing the homework “back to the school”

Some schools, both Norwegian and international, have been experimenting with making students do their homework at the school, instead of taking the homework home with them. The planned effect of this is that more students will complete their homework because they are able to collaborate with other students and, perhaps even more importantly, a teacher, who is present at these after-school sessions.

One such Norwegian school was Brandengen skole.³⁷ They expanded their normal school day by 1.5 hours to accommodate for the extra study session in which students would do their homework. They measured that by using the traditional model; only around 70% of the homework would be completed. With the new model, they managed to bring this number up to about 100%. Some students did not have the necessary support at home to be able to do their homework – some students had parents that did not speak Norwegian and could not help with their children’s homework.

Through the use of Wikis, this is made possible not necessarily by expanding the work day, but encouraging students to work on their homework in a collaborative fashion, as in their study sessions.

³⁷ “Mindre stress med lekser på skolen”. (Norwegian).
<http://www.dagsavisen.no/innenriks/article337588.ece>. Retrieved May 31, 2010.

Part 3

Deployment

The project

The goal of the project was to offer students a digital alternative to the traditional method of using pen and paper for their school work. In accordance with this goal, class 10 A at Orkdal Ungdomsskole was selected for participation in the project.

The proposed solution was to implement a piece of Wiki software on a server hosted at the Norwegian University for Science and Technology (NTNU). Building the Wiki software from scratch would have been too much of a challenge, especially with the added burden of testing, deploying, and maintaining the software. It was decided that the project would employ a mature, well-tested open source solution, readily available for download on the internet.

Considering the fact that part of the master's project was to add new functionality to the Wiki in order to test this new functionality on the motivation of the students to use it, it was decided to go for an open source package that had been developed in a programming language I was already familiar with. I had heard good things about ScrewTurn Wiki in the past, and so that ended up being my choice.

ScrewTurn is licensed on the GNU Public License (GPL). This could have been a slight issue, considering the fact that any changes made to GPL software needs to be made available to the public. My supervisor assured me this was not a problem.

Work started early in the fall. The first thing to do was to download a version of ScrewTurn that had the source code included. The domain name chosen for the Wiki was <http://www.skolewiki.net>. I registered this address and deployed a basic version of ScrewTurn onto a server hosted by NTNU, then forwarded the address to the server's external IP address. After this was done, I met with my contact at Orkanger Ungdomsskole, where I demonstrated the basic functionality of the Wiki. The goal of the meeting was to show the Wiki model to my contact, in order to discover how proficient she was with the technology at hand, and also to request any comments as to how the page looked and behaved.

- [Hovedside](#)
- [Tilfeldig side](#)
- [Lag en ny side](#)
- [Alle sider](#)
- [Kategorier](#)

- [Administrasjon](#)
- [Filbehandling](#)

- [Logg inn/ut](#)
- [Språkvalg](#)
- [Din profil](#)
- [Lage konto](#)

Søk



[Avansert søk »](#)

Per Olsen

Endret: 24/09/2009 18:48 av [admin](#) - *Ikke kategorisert*

[\(4\)](#) » [The Sandbox](#) » [Hvordan skrive din første artikkel](#) » [Per Olsen](#)

Jeg heter Per Olsen, og jeg er 15 år.

ScrewTurn Wiki  version 2.0.37. Some of the icons created by [FamFamFam](#) .

The meeting went very well. I showed her the basic version of the Wiki, and had a little discussion concerning how the technology would be used by her students. I had a couple of ideas regarding how the students would be allowed access to it, but was unsure of how this would play out because I didn't have any data on how new users adopt the Wiki model.

At this time there were still a couple of areas in the Wiki not fully translated into Norwegian. In addition, there was no way for the students or teachers to log in. My contact and me agreed that I would generate user names and passwords for the students in the class before we opened the Wiki page for normal use.

It is worth mentioning that the most basic user in ScrewTurn, like most other Wikis, have access to editing, creating, discussing and viewing the history of any article on the Wiki, unless the page has a special status as set by a moderator. In general, this is how the Wiki model works, and is thought to encourage cooperative learning. In ScrewTurn's case, however, there seems to be no possibility to create a more fine-grained security system without major reworking of the code. To explain further: The three types (or roles) a user can have are Guest, User and Admin. Depending on how the Wiki has been configured (i.e. either *Public* or *Private* access), a guest can either have the same rights as a user, or be locked out altogether. When the site has been configured to only allow logged-in users, a guest will be "turned away at the door", that is he will not have access to any part of the site except for the login page. A user can always log in, no matter how the Wiki has been configured. An Admin has the same rights as a user, but can also configure the Wiki by adding new users, installing plug-ins, and so on. He can also lock and delete articles.

Deployment

It was arranged for the Wiki to be deployed on Monday the 28th of September. In addition, a user manual would be written describing how a user could create his or her first article on the Wiki. As the Wiki was written in the programming language C# and running on top of the ASP.Net web framework, it was simple to fix the little things that would come up before deployment. The only things that needed attention were that some of the translation work in the readily available Wiki was lacking in certain areas.

Using a build script, deploying the solution to the production server could be done in less than five minutes. This was a huge help when it came to fixing bugs and getting a new version out in as little time as possible. A backup script was also developed, running nightly and making copies of the production site's articles and user database. These backups were then uploaded to a web-based cloud storage service, so any changes to the Wiki would be kept safe in case the production web server should crash.

The contact at Orkanger reserved 1.5 hours for teaching the students how to use the Wiki. This session was assisted by user guides that had already been uploaded to the Wiki beforehand. From my position as a site administrator I could view, almost in real-time, the changes that were made to the wiki along with a timestamp for each individual change. Before the session started I uploaded some user guides to the wiki for the students to follow. I had estimated that these prewritten user guides would help the students pick up the technology, but it soon became apparent that I had underestimated their technical abilities. At the start, I'd see a couple of students walking off the beaten path and trying out functionality that I hadn't described in my user guides. Students quickly learned how to link to external and internal pages, uploading images and linking to them, among other things.

"OVERSKRIFT".

Dette er *egentlig* ganske **basie** stoff. Om du fortsatt synes dette er avansert, [KLIKK HER](#), [HER](#) og [HER](#).



OG SLIK ER VÆRET

ScrewTurn Wiki  version 2.0.37. Some c

After a little while, students started editing other student's articles. On some occasions, students would delete text that other people had written. The underlying model of the wiki inherently discourages such malicious actions, because of the built-in version control of the text, and the simplicity with which one can replace vandalism with older versions of the text. Even though it is possible to delete text in an article and save that article so it becomes the current version, it is impossible to physically delete an article. Most of the students seemed unaware of that fact, and complained to the teacher that other students were tampering with their text.

Of course, a user with a more administrative role than a student would like to keep track of which students do vandalize. In planning for the deployment of the wiki, we had anticipated that some students would probably try to ruin other student's contributions. This is why we required each user to be registered with a user name and password in the wiki. A user guide was written describing how to administer these aspects of the application. When users have an assigned user name, that user name is associated with every single action they perform within the context of the wiki. This also means that when they vandalize an article, that action is logged in the application, and we can view this log to get an overview of what different users have done at different times.

Sidehistorikk: ██████████

Sammenlign sideversjoner

8 ▾ Aktuell ▾ Sammenlign

# Revisjon	Kommentar	Bruker
Aktuelle Versjon - 28/09/2009 10:04		noreha
8 28/09/2009 10:04		jornbr
7 28/09/2009 10:04		atledy
6 28/09/2009 10:03		atledy
5 28/09/2009 09:11		noreha
4 28/09/2009 09:09		noreha
3 28/09/2009 09:05		noreha
2 28/09/2009 09:03		noreha
1 28/09/2009 09:00		noreha
0 28/09/2009 08:53		noreha

After this first session, a meeting was held with my contact at the school. She thought that the session had gone a bit out of control in the way that she no longer could keep track of what everyone was doing. In a way this could also be viewed as a positive thing, in the way that the students learn much more by trying out the functionality of the wiki themselves, instead of just sticking to a pre-written recipe. It is, however, a fact that different students have different levels of technological competency, and some of the students might feel that it is difficult to learn the technology on their own. But in many cases, and especially within a technical context, learning seems to be much more effective for the users who try to test the system's limits.

My contact seemed positive to the thought of being able to get an overview of what had been written along with a timestamp of when it had been written. After the first session she had been left with a feeling of not having any idea of what the different students were doing. We arranged to have a meeting on the following Monday, so that I personally could manage the students while they were working with the wiki, and to better help them along if they were stuck with something.

Implementation

Early in the project, it was decided that the Agile Unified Process would be used for the maintenance and development of the existing software package I had been using. There are articles describing the use of this methodology for existing projects, although normally it is only used for projects started "from scratch".³⁸

The Agile Unified Process consists of the following steps³⁹:

³⁸ Wessberg, M., "Using IBM Rational Unified Process for Software Maintenance". <http://www.consolidate.se/docs/Using%20RUP%20for%20Software%20Maintenance.ppt>. Retrieved May 31, 2010.

³⁹ "Agile Unified Process". http://en.wikipedia.org/wiki/Agile_Unified_Process. Retrieved May 31, 2010.

1. **Model** – understand the business of the organization, the problem domain being addressed by the project, and identify a viable solution to address the problem domain.
2. **Implementation** – transform model(s) into executable code and perform a basic level of testing, in particular *unit testing*.
3. **Test** – perform an objective evaluation to ensure quality. This includes finding defects, validating that the system works as designed, and verifying that the requirements are met.
4. **Deployment** – plan for the delivery of the system and to execute the plan to make the system available to end users.
5. **Configuration Management** – manage access to project artifacts. This includes not only tracking artifact versions over time but also controlling and managing changes to them.
6. **Project Management** – direct the activities that take place within the project. This includes managing risks, directing people (assigning tasks, tracking progress, etc.), and coordinating with people and systems outside the scope of the project to be sure that it is delivered on time and within budget.
7. **Environment** – Support the rest of the effort by ensuring that the proper process, guidance (standards and guidelines), and tools (hardware, software etc.) are available for the team as needed.

Personally, I have used AUP in the development of other projects within groups of other programmers with a large amount of success. In comparison with other models like the waterfall model, AUP is very flexible, and one gets a feeling that it exists to support the development process instead of limiting it. In addition, considering the fact that this is a one-man project, one needs to prioritize certain parts of the process over other.

Exploration of the existing product

When looking through the code at first, one can see that it is relatively well written. According to experience this trait is often apparent in many open source projects. These projects in a way follow the Wiki model quite closely, through the fact that basic things exist, like complete openness in relation to other users and version control.

The group of developers that have made ScrewTurn has, in other words, made a pretty good job. One negative point one could make is that most of the code has been written so that the result is to be compatible with older versions of the development tools they have used, and that it should be compatible with older web browsers. They have used version 2.0 of the .NET framework, and version 2.0 of the C# programming language. This is both positive and

negative; it is positive in the way that it makes the source code accessible even to users who only have old versions of the development tools. It does not require the developer to purchase the newest development tools if he or she wants to contribute to the product. It is negative in the way that there exists newer versions of the development tools that simplify the development process. There are also newer web based technologies that would simplify and enhance the experience of the end user. A lot of performance improvements could be performed here through the commitment to newer technologies like AJAX.

Database agnosticism

ScrewTurn Wiki is *database agnostic* in that it can communicate with several different types of databases through the use of configuration changes in the system. An administrator can choose to migrate all of the Wiki's user-created content and data to another database implementation (e.g. from Microsoft's SQL Server to SQLite) at the click of a couple of buttons. The language needed to talk with the underlying database is defined in the *storage provider*, which translates the internal query language of the Wiki to the query language of the database that the system is communicating with. The underlying database or data provider will from here on be called the data store.

Most common operations

The most common operations that are performed on a Wiki by a user are as follows (sorted by how common they are):

- Search
- Read (that is, the user requests an article through his browser, and the Wiki reads the article text from a data store).
- Update (editing an existing article)
- Create (making a new article)

Search

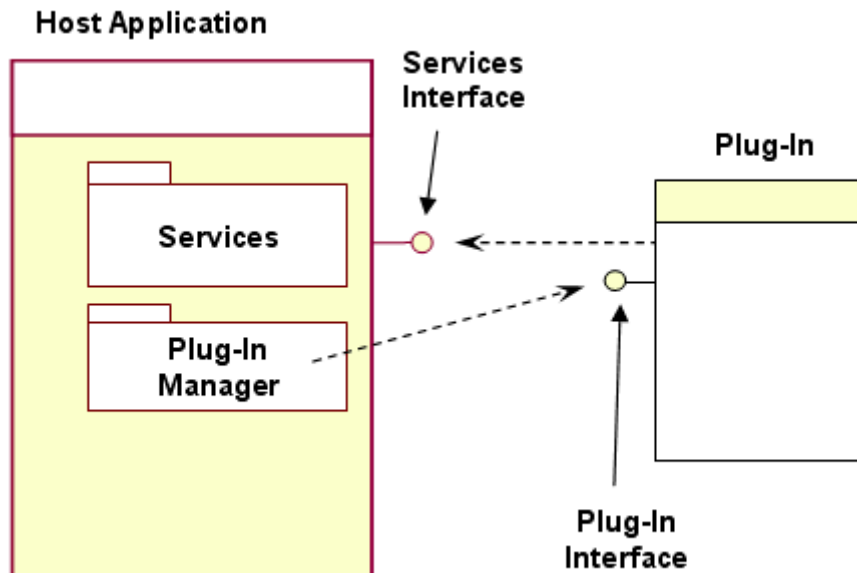
A search is instantiated by accessing the deployed site's search page. The search page is a so-called *special page* (refer to the section on Special Pages for more information), which allows a user to enter in a search query and view the results of running that query against the Wiki's data store.

Read

The read operation is instantiated when a user clicks on a link pointing to a page.

Plug-ins

One important trait that the system has is that it is extendable through the use of custom made plug-ins.



Most systems that support plug-ins, like ScrewTurn, work in the following way (refer to the sequence diagram for an example):

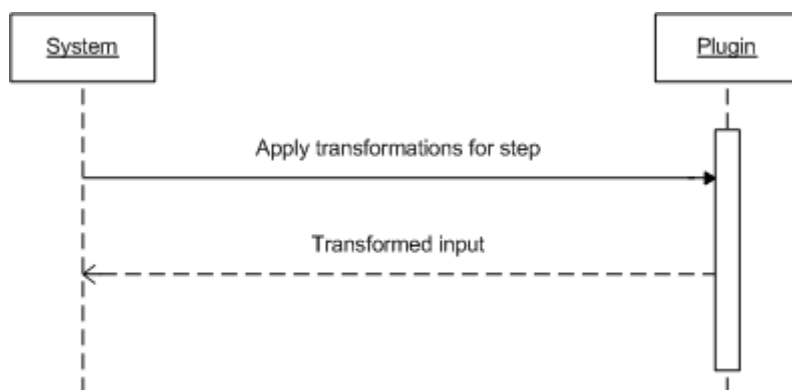
1. The system administrator acquires a plug-in in the form of a DLL or an archive file. This file contains the compiled code of the plug-in, written to implement an interface. For this example, let's call the interface IPlugin.
2. The systems administrator then has to register the plug-in to the system. Note that this can happen automatically with the administrator putting the plug-in in a specified folder or something similar. This allows the system to "know about" the plug-in in some way, so that it can dynamically instantiate it at a later time.
3. The IPlugin interface exposes one or more *hook methods* for applying the code in the plug-in to the work flow of the system. One can say that any plug-in that implements IPlugin is allowed to alter the way that the input to the system is treated, so that it results in a different output.
4. As a part of the application workflow, the system calls the plug-in manager to instantiate any registered plug-ins and run their hook method(s), sending any already-processed data into the plug-in as a method parameter.
5. The plug-in returns its altered output to the plug-in manager.

In ScrewTurn's case, there are three steps in the rendering of an article. After the user queries the system for an article by clicking on a link pointing to an article, or entering an

address to an article in the browser's address bar, the system goes to the database and looks for an article by that name. If the article is found in the database, the system starts the process of rendering the page to the user. This is done by converting it from the internal article format to normal HTML. Any tags in the original format are converted into their respective HTML representatives, and any registered plug-in transformations are applied to the output. This is done in three *phases*:

1. Performed before the wiki engine formats the content with its own formatter.
This phase is suitable to process custom tags, markup or other content that might be otherwise misinterpreted by the integrated formatter; overriding the behavior of existing tags is also possible.
2. Performed after the wiki engine formats the content with its own formatter.
This phase is suitable for processing custom tags that do not conflict with the built-in markup.
3. Performed before sending the content to the browser, at every web request.
This phase is suitable to process tags whose output is dynamic and potentially changes at every request.

At every step, or phase, the following action is taken by the system, allowing a plug-in to alter the article output before it is displayed to the browser.



Due to simplicity, every plug-in must implement the methods PerformPhase1 through PerformPhase3. Functionality at each phase can be turned on or off by the use of the PerformsPhaseN flag, where N is one of the three phases. If PerformsPhase2 is set to false, the system will not call the PerformPhase2 method during execution.

Through the use of the plug-in system, it is relatively trivial to design and implement changes that can be turned on or off in the administration interface in the Wiki. It would also be possible to change the actual Wiki by altering the code, although by doing this one loses some amount of modularity, and there is a possibility of introducing errors into the main code.

Further development

Project deployment versions

It is quite normal when developing software to organize the deployment into three different *classes*: Development, Testing, and Production. This means that the project's source code will exist in up to three different versions at the same time.

- Development deals with the latest source code that is developed, and is not to be dealt with as if it was *production* code, in that it is deemed too unstable to be used by end-users. In some cases, it might not even work to compile the code into a finished product. The product itself is usually represented by clear-text source code in this stage.
- Testing or staging describes code that is finished, where bugs have been fixed and it is ready for testing. The code has at this stage been compiled and deployed to some server installation, for example on the local development computer or on a server hosted somewhere else.
- Production describes code that has been tested and deemed stable and ready enough for usage by end users. This is the compiled version of the product that will be used by the students at the school.

Unit tests

Unfortunately, the existing product did not come pre-packaged with any unit tests. This is presumably due to the developers wanting any further development to happen with the use of plug-ins, instead of modifying existing code. On the other hand, some of the items that were to be implemented in this Wiki were hard to do without actually modifying the core code of the Wiki, and having unit tests would have been a massive help with this. This would have allowed us to *fork* the Wiki, that is, separating the Wiki code from the main source code branch and further developing our own version for use in the school.

Source code analysis

Several code metric analyses were run on the existing product, giving us numbers for *cyclomatic complexity*, class coupling, and lines of code for the core of the product. The analysis was run with the use of Visual Studio 2008. Cyclomatic complexity is a software metric that is used to indicate the complexity of an application, and gives us an indication of how hard it is to extend or fix errors in the product. It signifies the number of independent linear paths through the application, in the form of *if* blocks and *for* loops. According to

Wikipedia⁴⁰, there is a strong correlation between cyclomatic complexity and the number of defects that exist in a module. The class coupling describes how tightly coupled different classes are. The results of the code metric analysis are described below.

Maintainability index	Cyclomatic complexity	Class coupling	Lines of code
74	1773	141	4725

As we can see, the maintainability index for this project is very good, indicating that the coding style is well enforced and that there is a tight team producing the software.

⁴⁰ "Cyclomatic Complexity". http://en.wikipedia.org/wiki/Cyclomatic_complexity. Retrieved 31 May, 2010.

Part 4

Data collection

First data collection

The first interview and the questionnaire were performed after the Wiki had been in production for three weeks. No alterations had been done to the product at this point, but as the technology would be new to all the students, and because the intent of the research project is to not only study the Wiki, but also how any changes would affect performance and usage of the Wiki, it was deemed best to release the base version and see how the students reacted to this before introducing any changes. It was important that the students learned how to use the technology before introducing any changes in order to measure the effects of those changes.

Questionnaire

The introductory questionnaire was handed out to the students before the interviews proceeded, and collected when the interviews were over. The questionnaire was designed to be easily answerable and only touch on the surface of the experiences that the students would have had with the Wiki so far. We can sum up the results in the table listing students' positive and negative feelings about the Wiki.

Pro	Con
Text written is collected in a central storage area.	Rarely time to use the Wiki
Can access from anywhere	Other people can delete your work
Alternative way of learning	Wiki's layout is boring
Easier to share what one is writing, and learn from others.	Too advanced to learn
Motivation to write better when one knows it is viewable to others	Most of what we're writing already exists on Wikipedia
Cooperate to create a better product	Hard to navigate
Correct errors in other student's writing	Author of an article should be able to delete the article
Fun to work with computers	Bad name (referring to "SkoleWiki")
Useful to learn about Wikis	Bad security
Quicker to type with computers than handwriting	
Linking to other articles	

Almost like a conclusion of the curriculum	
Discussions	

To condense the most important points from the questionnaire:

- Access from anywhere – through the use of Wikis, homework can, in a sense, be shifted from an individual activity to a group activity, with separate students working on the same project from home. Considering the fact that any work saved on the Wiki stays there, students are not required to coordinate time amongst them – one student can edit an article at 18.00, and another student can continue the edit at 20.00. We already see schools trying to incorporate homework into the actual school day, instead of giving out homework for the students to do at home. (Dagsavisen reference)
- Centralized storage of work – in the traditional educational model, there is a vast amount of work that is produced by the individual for his or her own use; written notes, homework, marking stuff in the curriculum books, and so on. With a Wiki, it is possible to encourage students to write notes into shared areas instead of in their own note books. This makes it possible for any student to access the class' own notes for the course.
- Rarely time to use the Wiki – it was important when introducing the Wiki to the students that it would be a tool to use both in class and at home. It is thought to be an alternative to the traditional homework that the student is already doing. In this research project, it was harder to make this possible because we only had a four hour time slot allocated each week – the students had other homework that they had to attend to and for which they could not use the Wiki.

First interview

Seven students were interviewed, in order to examine their thoughts on the use of the Wiki. The interview was organized in a semi-structured manner, and revolved around the following topics:

- How much experience do you have with computers?
- Have you used a Wiki before? Have you edited a Wiki article in the past?
- Why do you believe the Wiki is open to everyone, allowing anyone to edit anything?
- Is it OK that other people can read what you have written?
- Why do you think we keep a history record of all changes on the Wiki?
- What do you think would be needed for the Wiki to be used outside of school?

- Can you compare the Wiki to your school's internal education software?

What follows is a summary of the more relevant quotes, grouped by the individual questions.

How much experience do you have with computers?

All of the seven students indicated that they had a lot of experience with computers. They all used the internet regularly, with different amounts of time spent on common internet pastimes – Facebook, Youtube, and so on.

Have you used a Wiki before?

The students had all used Wikipedia both for personal and educational purposes. Two of the students had edited a Wikipedia article, but done so not to enhance or improve the quality of the article, but as a practical joke.

Why do you believe the Wiki is open to everyone, allowing anyone to edit anything?

One student mentioned that this was a negative aspect, in that other students could enter the Wiki and essentially sabotage the original article by overwriting it with garbage and saving it. The student being interviewed quickly rebutted this point by saying that since it was possible to see who had done the edit through the history log, and also to revert the changes back to the original state, this was not really an issue.

Two students seemed to really grasp the collaborative aspect of the Wiki, but thought it was a bad thing that there was only one class using it. They thought that if more people would edit and contribute to the Wiki, the quality of the Wiki would ultimately improve. They also pointed out that “everything we've written is better on Wikipedia”.

The same students commented that, while the openness of Wikis made sabotage possible, it was alright to contribute or criticize other people's articles, as long as the criticism was constructive. On the aspect of sabotage, the student commented that if someone sabotaged his article, he'd just sabotage their article back. For a person who consistently sabotaged other people's writing, the students thought that that person should get a lower grade.

Is it OK that other people can read what you have written?

Five out of seven students did not mind the fact that other people could read what they wrote on the Wiki. As discussed earlier, the fact that one's individual or group contributions shift from being a strictly student-teacher relationship to a student-everyone relationship is a necessary break with the traditional way of learning, and can seem strange or difficult at first.

This was also encountered in the case study conducted by Lyndsay Grant⁴¹, where the students tried to adopt traditional borders around their individual work like they do in the traditional way of learning (paraphrased):

The students [were] relating their work in wikis to the wider discourse of school education, comparing this work to other written work completed in lesson time and being judged by the content of writing produced.

As mentioned earlier, it is vital to encourage the users of the Wiki to alter and improve the quality of the written articles. One student mentioned that it was a motivating factor that other people could read what one had written.

Why do you think we keep a history record of all changes on the Wiki?

As some students had mentioned, the history record functionality was needed to control the changes that people do, in order to discover who performs sabotage and so on. All students seemed to understand this.

What do you think would be needed for the Wiki to be used outside of school?

One student said that if they were given out tests to perform at home, that would encourage them to use the Wiki outside of school. Another student said that it was possible to use the Wiki to read up on test material that they themselves had written.

Can you compare the Wiki to your school's internal education software?

The school's education software was It's Learning, which is a content management system (CMS) and not a Wiki. The reason for this question was that it was not the intention to compete with this system. Therefore, it was important that the students understood the differences between their CMS and the Wiki.

One student felt that the overall writing experience in It's Learning was better than on the Wiki. They also didn't see the possibility for getting your articles on the Wiki judged by the teacher and getting grades on them. Another student saw that since writing articles on It's Learning was impossible, he learned much more by just using the Wiki as his learning tool. He also thought it positive that the students now had a platform which they controlled.

⁴¹ Grant, L. *Using Wikis in Schools: a Case Study*. (2006). http://www.futurelab.org.uk/resources/documents/discussion_papers/Wikis_in_Schools.pdf. Retrieved 31 May, 2010.

Second data collection

The second data collection took place two months after the first, and focused on trying to identify a core group of users that improved the quality of the Wiki. It also focused on trying to graph the occurrence of sabotage within the Wiki.

Sabotage

The data collection was performed simply by examining the history log through the Wiki APIs, and yielded some pretty interesting visuals. The graphs represent the counts of sabotage over time from the date that the Wiki was deployed to the school. Sabotage is, in this document, defined as malicious editing of other people's articles. Due to the number of students in the class, and various uncontrollable variables like groupthink and threats to fail the students if they kept it up, along with the knowledge of the history log itself keeping track of all the saboteur's actions.

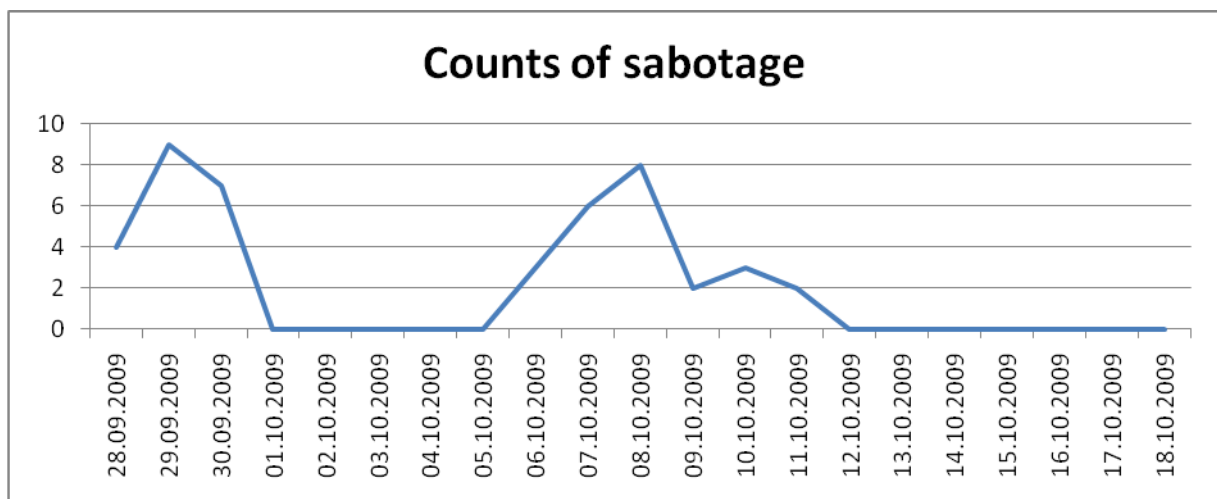


Figure 1 – Counts of sabotage starting on deployment date (28.09.2009) and ending three weeks into the experiment.

Now let's put this data into a little more perspective, bringing in the counts of sabotage performed after the end date.

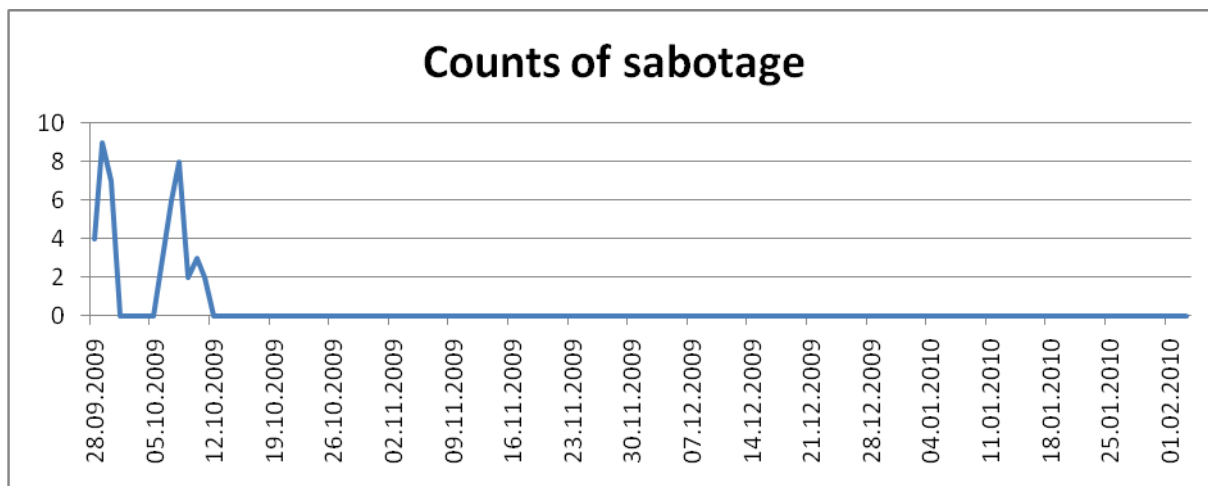


Figure 2 – Counts of sabotage starting on deployment date (28.09.2009) and ending about four months into the experiment.

As one can easily see, there was no more sabotage performed after the initiation period of the Wiki, that is, after the students learned how to use the Wiki. The sabotage done within the initiation period was restricted to such acts as overwriting what others had written, or writing derogatory remarks on an article that someone else had written.

23 06/10/2009 11:13	mohamadan
22 06/10/2009 11:12	juneaa
21 06/10/2009 11:12	torilannema
20 06/10/2009 11:12	juneaa
19 06/10/2009 11:12	mohamadan
18 06/10/2009 11:11	mohamadan
17 06/10/2009 11:10	mohamadan
16 06/10/2009 11:10	juneaa
15 06/10/2009 11:05	andersod
14 06/10/2009 11:05	mohamadan
13 06/10/2009 11:05	andersod
12 06/10/2009 11:05	admin
11 06/10/2009 11:05	juneaa
10 06/10/2009 11:04	mohamadan
9 06/10/2009 11:03	andersod

It was made a point to educate the students that there was a history record of everything that had been written and who had written it. Some of the students had initially believed that an article was restricted to the person who had written it, and any malicious things that were written in the article would be believed to be written by that same person. We can see a parallel to this in the Grant study, where one student said the following:

“There should be locks on it. Because they could write rubbish stuff and then you’d get told off for it.”

When the students found out that what they wrote was kept on record for as long as the Wiki stayed operational, the acts of sabotage seemed to stop. In a later section we discuss possible alternative reasons for this, and also match the results with the sabotage that happens on Wikipedia.

Core group

The data set of history records was sifted through in order to identify a core group, or even a set of individuals who repeatedly contributed to other people’s articles. In Grant’s study, these individuals were met with nasty comments, and it would be interesting to find out whether these results could be replicated in this experiment.

No such group could be found – the only person who seemed to be editing other student’s articles was the teacher, who helped the students with spelling mistakes and so on. It was important that students help each other as well, and so a point was made to educate the students on how to help each other with the articles, instead of seeing each article as a “one-person project”.

Other usage statistics

It was observed that most students made heavy use of the “save often” principle, in order to be able to keep a permanent record of their work as they were working on it, as well as being able to go back to previous revisions of their work if necessary.

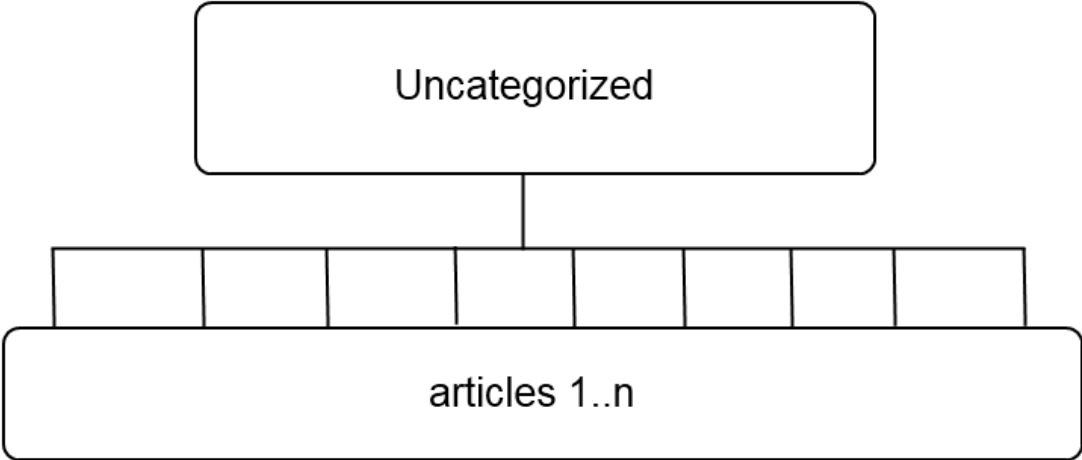
Third data collection / push for quality improvement

During the time between the second data collection and the third data collection, it became relatively evident that the quality of the articles published on the Wiki was not good enough. Some examples of this were that many of the articles had fallen into the one-user-per-article trap – with some different articles containing the same text, or slightly different versions of that text. Another example was that there were no categories for the different pages. Because we were interested in getting the users to collaborate and contribute to each other’s articles, a push for quality improvement was necessary. In this way, the third data collection also served as a workshop for the Wiki. The goal was to continually improve the structure of the articles such that only one article per subject existed, and that that one article was the authoritative article on that subject. Further changes to that article should come in the form of edits to that article alone, instead of duplicating pieces of text across articles. We also

wanted the students to categorize all articles so that they would be more easily browseable. Consider the following ways of structuring pages:

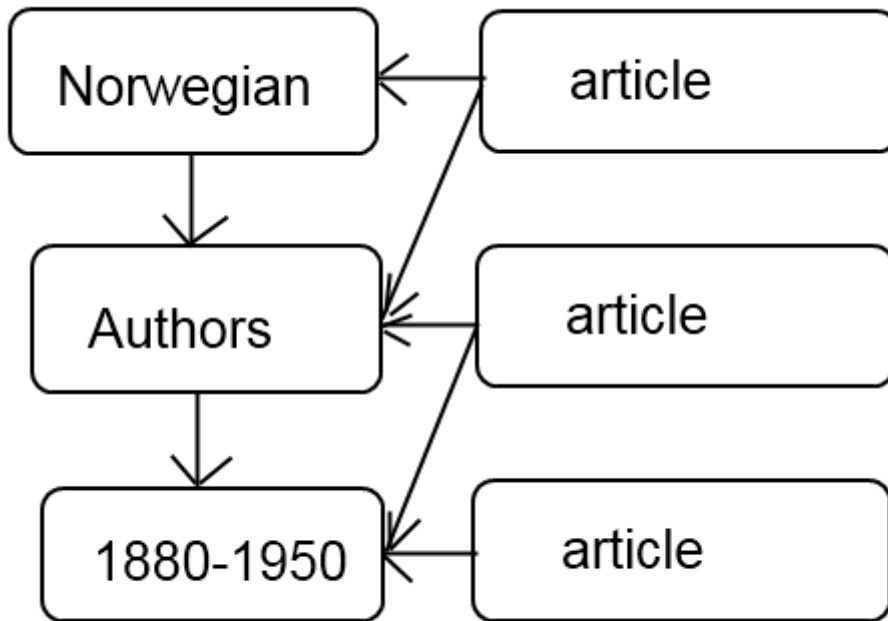
Uncategorized

Without categorizing, all articles are at the top level with regards to category. There is no way of browsing all the articles regarding Norwegian literature history without going through all articles in the Wiki.



Categorized

When categorizing articles, one can easily browse to whatever category one wants and see all the articles that fall into that category.



Categorized – articles are categorized by subject, with subcategories being more specific than their parents.

A plan was laid out to do these quality improvements in parallel steps:

- One group of users had the task of generating categories and subcategories for articles. This would result in better-categorized articles, attempting to group each set of articles per category in the students' curriculum, instead of the chaos of articles that were uncategorized before.
- Another group had the task of picking the best version of two or more articles concerning the same subject, and merge all the text in those articles into the best version. Then delete the lower-quality articles. This approach served as a kind of quality improvement to the point that we try to match the Wikipedia model as much as possible, but this also would have gone in direct conflict with the work of students not necessarily experienced with the Wikipedia model (there is usually no single author of an article, the article consists of various contributions from various stakeholders); there was a risk that students could feel unfairly treated because their article had not been considered the best one in the collection.
- Make the home page of the Wiki more presentable. Because of the low amount of distinct articles on the Wiki, and because it was so hard to find them (and there was

no real browsing functionality implemented), we wanted to make the home page a starting point for browsing down into the different subjects. In other words, instead of a 1-to-many relationship with the start page, we wanted a hierarchical way of finding articles, so that a student could browse his way to an article instead of having to know the name of the article to search for it. This is relatively different from the Wikipedia model, where there are usually articles regarding related subjects to the article one is reading, such that one can browse through the different articles in a general subject in a linear fashion. This break away from the Wikipedia model was justified by the fact that the articles were few and far between, meaning their content described wildly different subjects and were not strictly related by links, like Wikipedia articles are.

The workshop was functional in the way that the students performed the tasks asked by them, showing a better understanding of the Wiki model than previously experienced. Several new categories were created, and all but 13 articles were successfully categorized. In addition, many articles were rewritten or merged so that there was only one authoritative source on a subject on the Wiki.

Fourth data collection

In the fourth data collection, the final semi-structured interviews with the students were performed. The use of the Wiki had declined, based on the activity logs. This was partly due to changed priorities because it was the students' last semester at the school, but it was also interesting to find out why there had been no "takeover" of the Wiki by the students; why did they only use the technology when they were "forced" to?

Eight students were interviewed. The list of questions revolved around the following topics:

- Do you feel that the Wiki has helped you in any way with your studies? If so, how?
- Would you be positive or negative to the idea of starting out with the Wiki from grade 8 instead of 10?
- Do you feel that the Wiki is a tool for the students or for the teachers? Who gets the most benefit?
- What would you feel if your grade in the different courses were directly tied into your performance on the Wiki?

The responses to all the questions varied:

Do you feel that the Wiki has helped you in any way with your studies? If so, how?

The students all indicated that the Wiki had helped them in a way, but they were unclear as to what degree it had been helpful.

One student answered that it's helpful to be able to write about the things they were studying, but was unhappy about the amount of time they had been spending on the Wiki in general. He didn't feel that it was worthwhile to read up on the summaries that the students had been writing about the curriculum, because there was so little information, and the information that was indeed there was spread too thin.

Would you be positive or negative to the idea of starting out with the Wiki from grade 8 instead of 10?

The interviewed students were all positive to this idea of being taught how to use the Wiki as they enter the 8th grade and continuing to use it as one source of information (among others like books and articles) until they graduate from the 10th grade. Follow-up questions were asked about whether the students would be scared of editing articles that other people had written, but few interviewees thought this would be the case. One student even thought it would be easier to edit other people's articles if the original authors were not in one's class, because there would be less of a chance that one person would get mad at you for editing his or her work. The student added that editing would probably occur as follows:

- 9th grade edits 8th grade work
- 10th grade edits 9th grade work

And not the other way around. This seems fairly intuitive given the fact that younger students would not normally have learned the curriculum of the students older than them. It is also a possibility that younger students would be afraid of some kind of reaction from older students when editing their work, simply because they are younger.

Do you feel that the Wiki is a tool for the students or for the teachers? Who gets the most benefit?

Most of the students thought the Wiki was a tool for both the students and the teachers, but was heavily favored towards the students. The students were the ones who were supposed to learn from this.

What would you feel if your grade in the different courses were directly tied into your performance on the Wiki?

One follow-up question was whether the Wiki should be more tied-in with one of the students' courses, which would of course further the connection to the teachers, because they would

have to analyze the contents of articles in order to give the students a grade for their work. The answers to this question was more varied than the previous ones – some students did not like this idea, because they thought the grades would differentiate between the students who were skilled at computers in general and the students who were not that experienced, even though the Wiki model should be understandable even to a person who has little or no experience with computers.

Part 5

Results and further research

Results

Motivation

It was a sobering experience to discover that it was hard to replicate Wikipedia with its self-maintaining process that just has seemed to evolve out of nothing. Some suggestions as to why this didn't happen are described later in the text, but more research is necessary. In this implementation of the Wiki, there was close to no editing outside of school time. There was plenty of excitement about the new technology and getting to use computers more at the school. The models for motivation seem to match up in some way with Shirky's description of why people contribute to Wikipedia⁴², although some seemingly significant exceptions are the user base, which is much, much larger on Wikipedia, and the time that was given to work on the Wiki. In our case, a relatively small amount of time was allotted to work on the Wiki, because that's all the time that was available for doing research. In further experiments, it would be interesting to see what would happen were the students given a larger time scale (say, three years), and a more even distribution of time spent on improving articles, instead of a workshop here and a workshop there.

The results on motivation are, however, consistent with the results produced in the study done by Lyndsay Grant⁴³, and seemed to be one of the main reasons that the Wiki in that study was only used for small personal or group work by the students. Building up and maintaining a Wiki containing information that is actually relevant to the people who are writing the articles is a more difficult endeavor, and was seemingly a limiting factor in this experiment. In order for a Wiki to really be useful for the students, it would need to contain curricular information instead of serving as a place to host the students' group work. In other words, in the Grant experiment, the researchers made an effort to see *how* students carried out work on a Wiki. In this experiment, we are trying to integrate the Wiki as a tool for studying. The main difference being that we are asking the students to participate and entering the information they feel is important, instead of giving them a project that they *have* to work on. This follows more in the Wikipedia spirit.

⁴² Shirky, C. (2008)., p. 139.

⁴³ Grant, L. (2006).

It would be interesting to see whether the functionality of the Wiki could be increased if we created a group of people who started writing the main articles and delegating tasks out to the students to do before each class. This group could consist of students who volunteered for this task, as well as teachers who were interested in furthering the Wiki. In a way, instead of relying on the students to create all the articles themselves, they would be swept up in an ongoing process and might be able and willing to contribute more to the Wiki.

In essence, increasing the amount of available material as well as keeping the Wiki active would probably be a good way of researching motivation in those students who were not a part of the group.

Sabotage

It was very interesting to see the graph describing the counts of sabotage. After the initial testing period had ended, the amount of sabotage dropped down to zero. The amount of user activity closer toward the end of the semester started dropping as well, but there was no noticeable correlation to the amount of sabotage performed.

Why was there no more sabotage? Shirky writes that the Wikipedia article history system works in a way that fundamentally damages sabotage⁴⁴. He uses the example with graffiti – if removing graffiti from a brick wall required only the click of a button, not only would brick walls everywhere be cleaner – it would take away the motivation for tagging on any brick wall. Graffiti artists would have to find legitimate places to put their art, and people wouldn't bother to write their names with spray paint anymore.

It is unclear whether we can adapt this model of sabotage to our own Wiki. There are many variables in play:

- On Wikipedia, a saboteur very rarely knows the author of the article he is sabotaging, while on this Wiki he will almost certainly be in the author's class in school.
- The teacher made it clear from the start that any sabotage would be dealt with by lowering the saboteur's grades. While the disciplinary action was never performed, this probably served as a good way of discouraging the students from sabotaging each other's articles.

Another Wiki feature that probably discouraged sabotage in the long run is the versioning feature, along with the fact that all student accounts were registered on the Wiki. This made it possible to not only save a copy of each version of an article, but also to attach the editor's name to each version. If we further Shirky's metaphor of graffiti, it would be even more

⁴⁴ Shirky, C. (2008)., p. 120.

discouraging to the tagger if his full name was printed next to each of his tags, along with a date describing when it was performed.

Content

In general, I would have liked to have seen either a tighter focus on one of the subjects and writing more about one specific thing, instead of spreading out the information too thinly. As some of the students indicated in their interviews, they didn't really feel that the Wiki was a good source of information, because there were so many holes in the information. A tighter, more coherent Wiki could have been more successful in the long run – students could add articles as part of larger projects to extend the Wiki with new information from the curriculum. For example: If we had focused our effort to just the subjects of Knut Hamsun, Henrik Ibsen, and Jonas Lie in this semester, and then the students in the next semester would write about three other authors, or about Norwegian Literary History in general, this could have helped increase the coherency of the information on the Wiki and ensure that the articles that were available were more tightly linked by context.

Content vs. motivation

It would have been interesting to do a study where we split the students up into two groups – one group would use an empty Wiki, while the other group would use a Wiki where many of the articles had already been filled in from their curriculum. It would even be possible to add obvious mistakes to the articles and measure whether the students corrected these errors. Studying the activity over time between these two groups, and seeing whether we could derive a core group of students who took it upon themselves to maintain the Wiki would be an interesting idea for further research. Such a core group is probably hard to come by, but is almost vital for a Wiki with such a small user base as ours – compared to Wikipedia which has a very diverse user base, with most of its users not really writing that much, as evidenced earlier in the text – most users only make one or two edits during the course of their Wikipedia use.⁴⁵

Similarities between programming projects and the Wiki system

When one considers the fact that a Wiki is a relatively new concept, it seems necessary to examine whether we can draw parallels between studies of this new technology and studies concerning older technology, or even older models encouraging cooperation and knowledge-building. One such concept is that of programming projects. The most interesting one is open source development. Participating in an open source project expects all developers to share some common abstract goal. It is possible to *fork* a project – the process of copying an open

⁴⁵ Shirky, C. (2008)., p. 123.

source project and making custom changes in the copy instead of in the main version. These forks, however, usually do not succeed as well as the main version.

Content ownership

In programming, the concept of *code ownership* is one that would be advantageous to compare to the “owning” of a Wiki article. Martin Fowler writes:

“There are various schemes of Code Ownership that I’ve come across. I put them into three broad categories:

- **Strong code ownership** breaks a code base up into modules (classes, functions, files) and assigns each module to one developer. Developers are only allowed to make changes to modules they own. If they need a change made to someone else's module they need to talk to the module owner and get them to make the change. You can accelerate this process by writing a patch for the other module and sending that to the module owner.
- **Weak code ownership** is similar in that modules are assigned to owners, but different in that developers are allowed to change modules owned by other people. Module owners are expected to take responsibility for the modules they own and keep an eye on changes made by other people. If you want to make a substantial change to someone else's module it's polite to talk it over with the module owner first.
- **Collective code ownership** abandons any notion of individual ownership of modules. The code base is owned by the entire team and anyone may make changes anywhere. You can consider this as no code ownership, but it's advocates prefer the emphasis on the notion of ownership by a team as opposed to an individual. (...)

“

The AuthorShip page on the Community Wiki site lists a couple of strategies for documenting the authorship and contributions on a specific article. The page discourages thinking of articles as having a single author, even though in practical circumstances there might very well be only one author per article. In the school setting, this is also what we would like to discourage, and we try to develop ownership strategies similar to weak code ownership as described by Fowler. One could say that the most intuitive strategy one would use when starting out working on a Wiki would be strong code ownership. As a function of getting used to other people reading what you have written, and maybe starting out editing other people's articles, a progression could be made from strong code ownership to weak code ownership.

Here is a list of the models described on the AuthorShip page:

- **Spatial authorship model** – different parts of a Wiki page are authored by different contributors.
 - Advantages: Each part has a single author.
 - Disadvantages:
 - Does not treat a page as a single entity.
 - Contributions that don't result in persistent artifacts (like deletion and refactoring) are not recognized.
 - Contributions of ideas that are later re-stated (edited to have a different wording) are not recognized.
- **Temporal authorship model** – each contributor creates a brand new work based on the previous version of the page. Each page version is “authored” by the person who made the last edit, with “contributions” or “co-authoring” by all the people who have made edits previously.
 - Advantages:
 - Authorship is assigned to a whole page, not to words, letters, and punctuation marks.
 - People who make contributions like refactoring and deletion are recognized on the article as contributors.
 - People who contribute ideas that are eventually restated are recognized.
 - Disadvantages:
 - It's hard to justify calling contributors who make small contributions the “primary” author.
- **Mixed mode model** – combines both temporal and spatial authorship models. The page is first logically split up into different areas. Then, for each area, the primary author of that area is the last person to make changes to it and the contributors to that area are all the people whose edits have (directly or indirectly) effected it.
- **Group authorship model** – considers the set of all editors on the page to be “co-authors” or “contributors”. In other words, the page is an indivisible unit, created, blended, remixed and refactored by all the people who took the time to edit it. Each contributor gets equal credit as co-author. What is important about one's role as co-author is not *what* one contributed, but that one did.
 - Advantages:
 - Authorship is assigned to a whole page, not to words, letters, and punctuation marks.
 - People who make contributions in the form of editing and refactoring are recognized.

- People who contribute ideas that are eventually restated are recognized.
- Disadvantages:
 - One contributor could make 95% of the text, while another made 5%, and they would both get equal amounts of credit.

Incentive

The incentive to maintain a Wiki's articles should come from the idealistic concept that a well-maintained Wiki is good in itself. The common goal should, in other words, be to create and maintain high-quality material. The incentive for creating and maintaining the articles should not be to gain respect or acknowledgement for writing a good article, but rather that the *collective consciousness* works on a product to increase the value of that product.

This would be true in the ideal case. There has, however, been some research on point-based systems for constantly-expanding social networks and knowledge-based repositories. Stack Overflow, a knowledge-based question/answer site, specifically engineered for asking and answering computer-related questions ⁴⁶ employs a point-based system for its users. A user may gain so-called *reputation points* if a question he asks is *upvoted* by another user, and may lose these reputation points if another user *downvotes* that question. In many of these upvote/downvote sites, the ratio between upvotes and downvotes results in an 1:1 score, that is, any upvote will increment the user's score by 1, and a downvote will decrement that user's score by 1. Stack Overflow has a different ratio – an upvote earns the recipient 10 reputation, while a downvote subtracts 2 reputation.

The screenshot shows the Stack Overflow homepage. At the top, there is a navigation bar with the Stack Overflow logo and links for Questions, Tags, Users, Badges, and Unanswered. There is also an 'Ask Question' button. Below the navigation bar, there is a 'Recent Questions' section with a filter for 'active' and a count of 118. The questions listed are:

- HTML text field not displaying decimal places of SQL money value (0 votes, 3 answers, 24 views, 8s ago, Kosta 108)
- Creation date column in SQL table (1 vote, 4 answers, 17 views, 20s ago, paxdiablo 79.9k)
- Python loop | "do-while" over a tree (2 votes, 6 answers, 248 views, 24s ago, Neha 1)
- Type or namespace name could not be found (0 votes, 2 answers, 39 views, 25s ago, Tejs 331)
- C# analog for sql in operator (1 vote, 2 answers, 23 views, 27s ago, Amitabh 459)

On the right side, there is an 'Interesting Tags' section with an 'Add' button. Below that is an 'Ignored Tags' section with an 'Add' button. At the bottom right, there is a 'Recent Tags' section with a list of tags and their counts: c# (177), sql (52), datetime (2), and sql-server-2005 (13). There is also a job listing for a .NET Developer at Stayhealthy, Inc.

⁴⁶ "FAQ – Stack Overflow". <http://stackoverflow.com/faq#reputation>. Retrieved May 31, 2010.

This seems to have had a positive effect on the site. Users admit to maintaining, editing, and answering questions simply to achieve more points⁴⁷. It does not matter that the points do not translate into any form of material good (like money), but the site's model rather subscribes to the fact that a sense of reward is associated with receiving these points, almost as if they are confirmations that one has done something that is inherently "good" or "right". Other users do not subscribe to this idea, but say that they are doing it in order to build the site so that it will get better. In this sense, knowing that the site quality improves with each correct answer is the primary reward, and the point system is a secondary reward.

Though the points in themselves do not really earn the user any objective value, they do give the user more "power" on the site.

The following list describes in more detail what abilities the user will get when he amasses a certain amount of points.

- 15 Vote up
- 15 Flag offensive
- 50 Leave comments†
- 100 Vote down (costs 1 rep)
- 100 Edit community wiki posts
- 200 Reduced advertising
- 250 Vote to close or reopen your questions
- 250 Create new tags
- 500 Retag questions
- 1000 Show total up and down vote counts
- 2000 Edit other people's posts
- 3000 Vote to close or reopen any questions
- 10000 Delete closed questions, access to moderation tools

Another incentive is the badge system, directly modeled after the Xbox badge system⁴⁸, where a player can earn a badge as a reward for completing various tasks. The badge awards are easy to achieve at first – for example, there exists a badge for finalizing your profile on Stack Overflow, that is, filling out all the details of your profile. Sites and games

⁴⁷ "6 Simple Tips to Get Stackoverflow Reputation Fast". (2009).

<http://www.codexon.com/posts/simple-tips-to-get-stackoverflow-reputation>. Retrieved May 31, 2010.

⁴⁸ "Browsing Retail Games". <http://www.xbox360achievements.org/games/retail/>. Retrieved May 31, 2010.

that employ techniques like this for rewarding their users can be described as *meta-games*. Let's consider a presentation on Meta-games made by Amy Jo Kim⁴⁹ (summarized and paraphrased).

Meta games are everywhere. In the Scouts, one accomplishes certain things in order to receive a badge for that accomplishment. In karate, people try to develop their skills in order to get new ranks, which translate to prestige. Airlines' frequent flyer programs award the traveler by transferring a certain amount of points to that traveler's frequent flyer account after each flight. In contests, people perform an action, which could be anything from simply entering the contest to answering a question. Some of these things directly translate into a certain amount of value for the user. In Farmville, the immensely popular Facebook application⁵⁰ (which currently has around 80 million users – a measured 30 million people using it **every day**), users can grow their crops, tend to their animals, and invite their Facebook friends to visit their farm and comment on it by putting up signs. Accomplishments in Farmville do not translate into any form of value, monetary or otherwise. So why would people spend their time on these things? Amy Jo Kim describes three types of points a user can receive:

- Experience Points (XP)
 - Earned directly via player's actions
 - Track & reward socially/economically useful player actions
- Skill points (Score, rank)
 - Earned via interacting with the system
 - Based on mastery of the activity or the game
- Influence points (rating, reputation)
 - Earned via the actions of other players
 - Proxy for quality/reputation/influence
 - Track & reward socially valuable contributions and actions

⁴⁹ Kim, A. J., "Reward Systems that Drive Engagement". (2010). <http://www.startuplifeblog.com/2010/03/09/reward-systems-that-drive-engagement-via-amy-jo-kim/>. Retrieved May 31, 2010.

⁵⁰ "FarmVille". <http://en.wikipedia.org/wiki/FarmVille>. Retrieved May 31, 2010.

An important point about any reward system is whether the users can spend their points in any way. If we consider the Stack Overflow example from before, users are allowed to spend their karma points by assigning a *bounty* to one of their questions. This means that if the user asks a question and the question goes unanswered for a certain amount of time, the user can then put up a bounty of any amount of points that the user might have. Whoever answers that question will then receive that amount of points into his or her account.

In other systems, points or rewards cannot directly be spent. Consider Reddit⁵¹, another point-based web site that encourages users to post new articles that they have found around the web. Other users will then upvote or downvote these articles. Users can also post comments on the articles if they would like, and these individual comments can also be upvoted or downvoted. A popular comment tends to have a large amount of upvotes and will rank higher on the list of comments. The points each user earns cannot be spent on anything. They simply serve to show how popular or well-received a user's contributions have been.

A system like this would be interesting to experiment with on an implementation like a Wiki. SkoleWiki might benefit from such a point-based system. Students would then be allowed to upvote or downvote a user's contribution to an article, whether that contribution is an edit or an insert. Maybe the users could be encouraged to earn more points simply because of this added feature. This would serve as an incentive for the users to contribute more the Wiki.

Another factor to consider is the one concerning the context of the site. Incentive is a very powerful factor alone, but simply adding a reward system to an otherwise "boring" site (as perceived by the user) probably won't do any good. Wikipedia talks about the phenomenon of *critical mass* in the context of social systems⁵²:

“Critical mass is a sociodynamic term to describe the existence of sufficient momentum in a social system such that the momentum becomes self-sustaining and fuels further growth.

Social factors influencing critical mass may involve the size, interrelatedness and level of communication in a society or one of its subcultures. Another is social stigma, or the possibility of public advocacy due to such a factor. Critical mass may be closer to majority consensus in political circles, where the most effective position is more often that held by the majority of people in society. In this sense, small changes in public consensus can bring about swift changes in political consensus, due to the majority-dependent effectiveness of certain ideas as tools of political debate.”

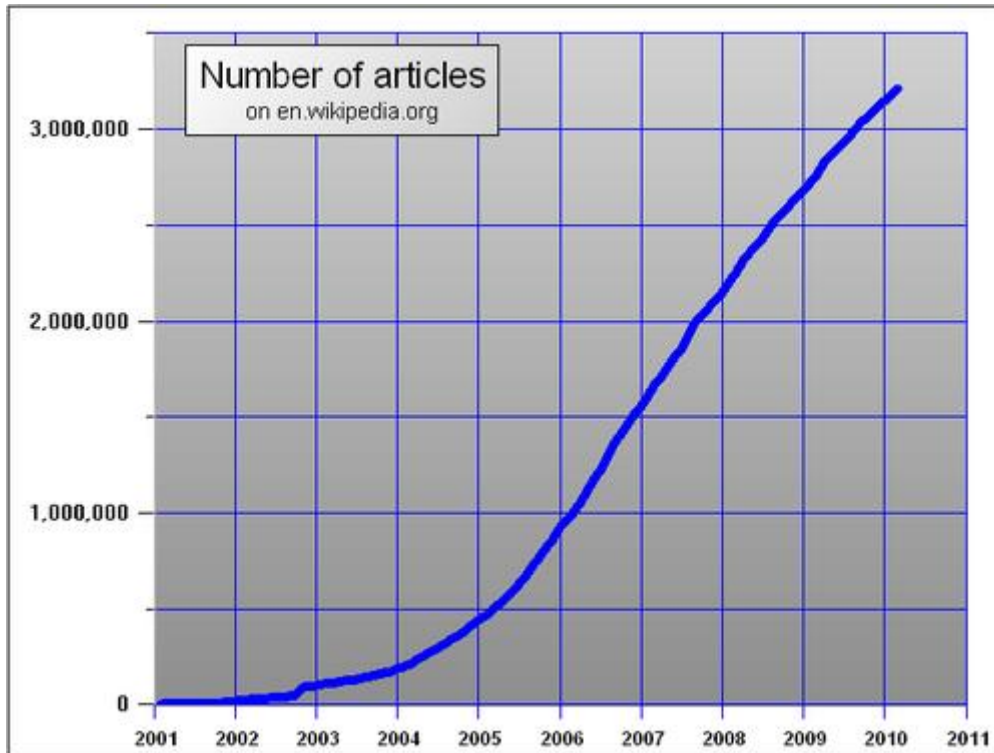
⁵¹ “reddit.com: what's new online!”. <http://www.reddit.com>. Retrieved May 31, 2010.

⁵² “Critical mass (sociodynamics)”. [http://en.wikipedia.org/wiki/Critical_mass_\(sociodynamics\)](http://en.wikipedia.org/wiki/Critical_mass_(sociodynamics)). Retrieved May 31, 2010.

The dynamics of critical mass are inherently altered by the context of the social network.

Models for Wiki growth

Wikipedia is seen as the largest and most complete of the Wikis, with current page counts being over 3 million. With this data as their base, a couple of growth models have been devised to further explain why Wikipedia grows, as well as predicting future growth.⁵³



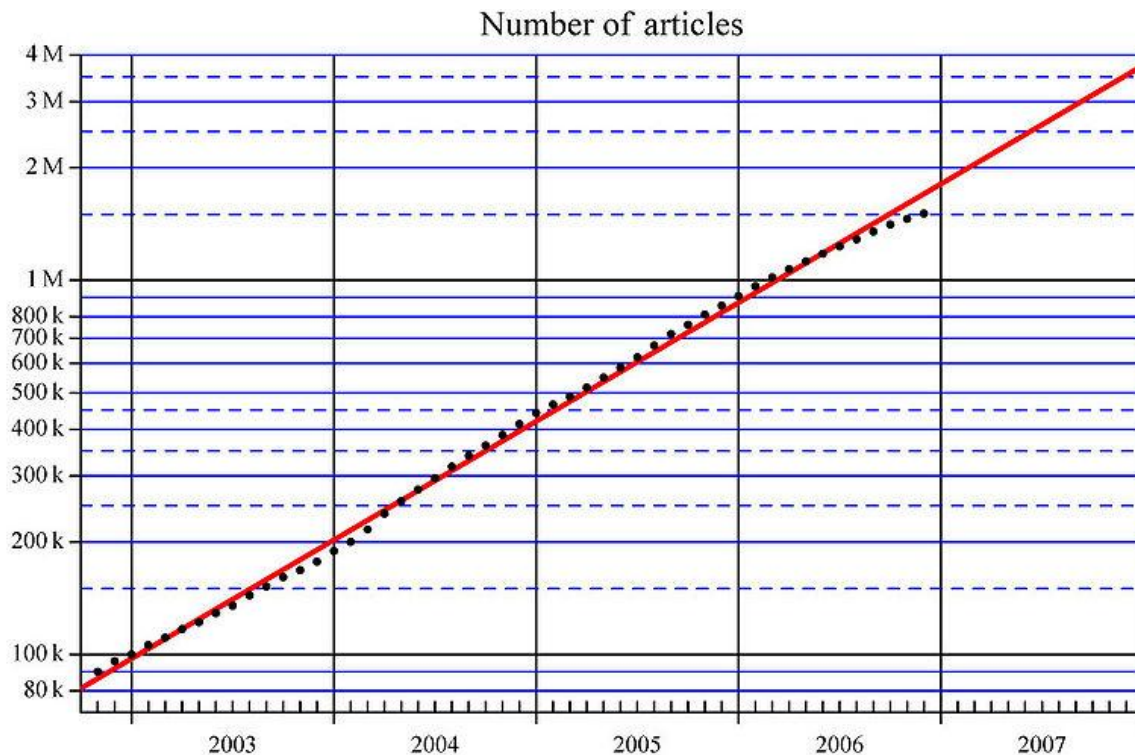
Exponential model

The exponential model describes Wiki growth such that:

- More content leads to more traffic
- Which leads to more edits
- Which generates more content

This model gives us the following graph

⁵³ "Wikipedia:Modelling Wikipedia's Growth".
http://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia's_growth. Retrieved May 31, 2010.



Between the time of October 2002 and October 2006, this graph mirrors the actual growth of Wikipedia to a very good approximation⁵⁴. According to Wikipedia, if they had kept up with this trend, the number of articles by December 2006 would have been 1,900,000, by June 2007 2,800,000 and by December 2007 4,000,000, although there has been a slowdown of the growth and Wikipedia has apparently ceased growing exponentially.

By the end of 2006, when there were 1.5 million articles, the projection was already overestimating the growth by 10-15%, and the prediction of over 3 million articles by the end of 2007 is significantly more than the actual figure of about 2.1 million articles.

Logistic model

As the exponential model no longer fits the growth of Wikipedia, a logistic model was developed in 2007, which better describes the growth:

- More content leads to more traffic, which in turn leads to new content
- However, more content also leads to less potential content, and hence less new content
- The limit is the combined expertise of the possible participants

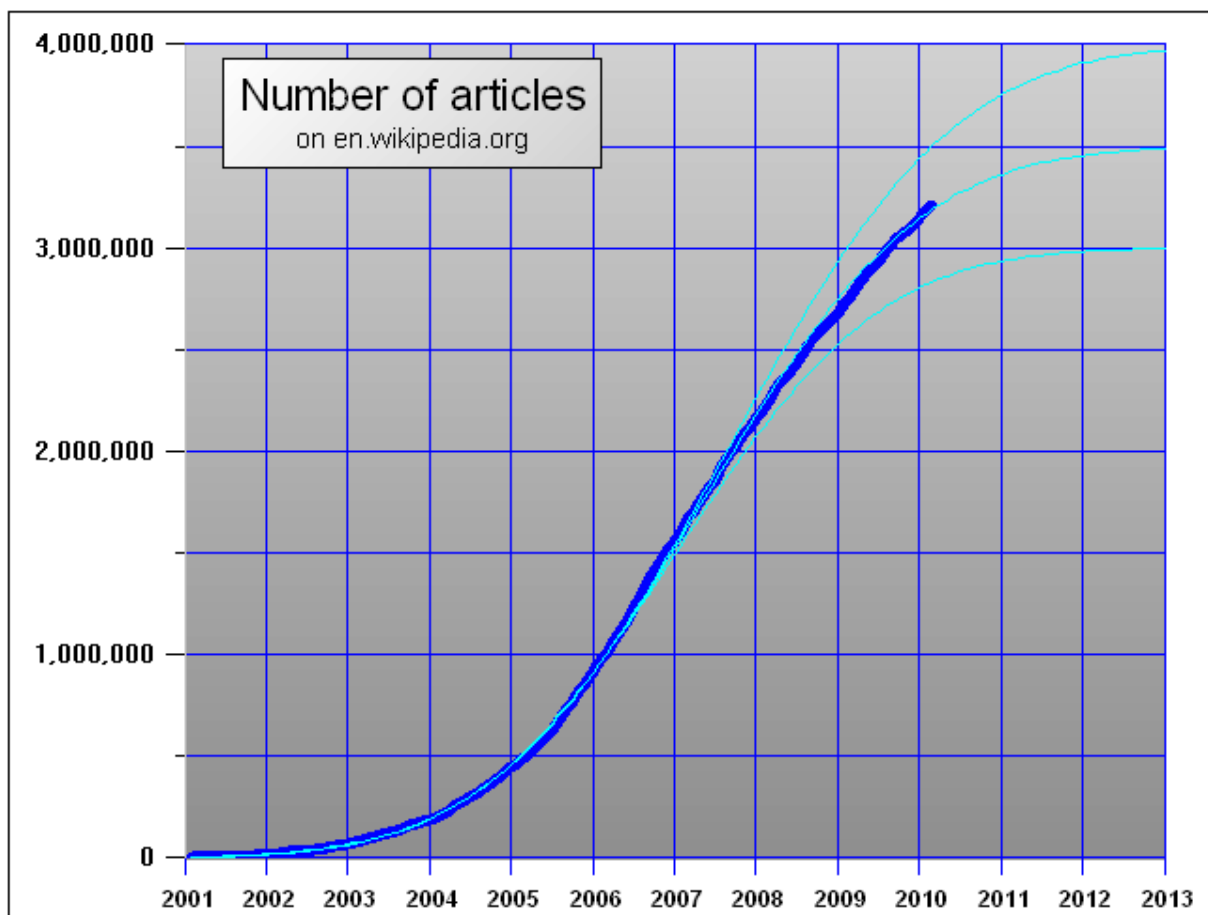
Some characteristics of this model are:

⁵⁴ "Wikipedia: Modelling Wikipedia's Growth". http://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia's_growth. Retrieved May 31, 2010.

- There will be a maximum to the number of articles. On Wikipedia one can hardly imagine this as there will be new events and people to describe in the future. Compared to the large number of existing articles this is a very small effect though.
- At the end the growth is zero.
- At the pivot point (halfway the maximum) the growth is at its peak. For the [English Wikipedia] this might have been in August 2006 with 60,000 new articles a month.

Although this model describes the *quantity* of new articles as having a maximum given a specific point in time, the *quality* of articles may still improve.

This gives us the following graph, displaying article growth until around March 2010, with extrapolations to the marks of 3, 3.5, and 4 million articles.



More models for explaining and predicting Wikipedia growth exist, but are not strictly relevant to the subject matter discussed in this paper. The function of growth is, intuitively enough, in all models attributed to the activity of the Wiki's users. Some amount of change can be ascribed to automatic bots working on the site, but almost all content available on Wikipedia is due to users contributing that content.

Wiki configuration

In order to be able to deploy the Wiki in a normal school setting, it is necessary to have an understanding of how the Wiki works – not just the Wiki model that almost all implementations adhere to, but more specifically the details of the implementation and the correct way to configure it.

In the ScrewTurn implementation that has been deployed on the school, there are a few common tasks that need to be done, which we'll describe here.

User account management

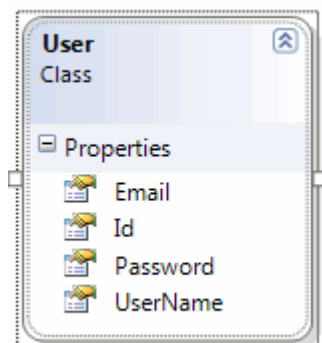
For day-to-day operations, it is important that administrators, whether they be the IT administrator at the school or a teacher, know how to generate new accounts, recover lost passwords, deleting old and unused accounts, and suspending and re-enabling accounts.

Further steps

This section discusses some further steps one could take in order to further integrate the Wiki into the school IT system.

Transitioning a school account system to a Wiki account system

In a school context, it would in most cases be possible to write a plug-in (as previously detailed) that could data mine the existing student database and generate users for the Wiki, so that each student would be given his or her own account. Let's take a look at the main properties of the account model for an example Wiki implementation.



As one can easily see, the model is so simple that a transition from any standard school account system would be trivial, provided the account system is accessible through a standard interface like API bindings or even a standard SQL database.

This simple tool would be set up to run every year, creating new accounts based on the data that exists for the currently active students, and disabling the accounts of students that have graduated from the school or otherwise do not take part in any classes anymore. It would

also not override the settings set up by the Wiki administrator who might have disabled some accounts because of sabotage or otherwise.

Making it easier to create new accounts

In case the transitioning system is too difficult to properly implement, either because of a non-existent programming interface to the existing school account infrastructure, or because of other challenges, it is necessary to have an easy interface for creating new accounts en masse. This would be interesting to do every new semester, when an influx of new students requires new accounts on the Wiki in order to be able to work with it. This problem would not exist if the Wiki were to be completely anonymous and not require an account to modify, but as was decided for this implementation, keeping track of who did a change makes it easier to record counts of sabotage and directly connect those offenses to the person who did it. Private means, as mentioned in the administrator's user manual, that the Wiki requires a user to log in before making any changes.

Anonymous accounts

For any given article in a private ScrewTurn implementation, it is trivial to retrieve a list of user names having edited that article by using the history feature. The user name was in this implementation directly connected to the account user's first and last names, and so it is easy to infer which student a certain username belongs to. This could in some cases interfere with whether a student will edit an article that, in a sense, "belongs" to another student, most often because that other student edited that article first.

An alternative to this is giving each user a "real" username which would only be visible to that user and users with superior privileges than that user (such as a teacher). This would still make it possible to connect any form of sabotage to that student's account, but only by a privileged user.

Automatic feeds

As a school Wiki would be geared towards several students working on articles, it would be possible to set up automatic feeds or notifications being sent to interested parties when an article was being edited. This would be useful when students would like to be notified of any changes that had been done to "their" articles.

This would be quite trivial, and plug-ins for this do in fact already exist and are available on the ScrewTurn Wiki web site⁵⁵. Notifications could come in the form of:

- Article title

⁵⁵ "User's Plugins". <http://www.screwturn.eu/Customize.UsersPluginsV2.ashx?HL=plugin>. Retrieved May 31, 2010.

- Name of contributor
- Time of edit
- Link to check out the diff between the current and the previous page version.

This idea ties into the subtext of the whole reason this implementation was deployed in the first place – as mentioned earlier in the text.

Conclusion

After the prestudy stage was finished, the Wiki was deployed at Orkanger school. The Wiki evolved in such a way that it was hard, at first, to keep track of the development of learning done by the students. It was soon apparent that some of the students took very easily to the newly introduced technology, while others had a harder time grasping the concepts.

Sabotage

It was interesting to witness how one of the most important reservations I had before deploying the Wiki played out: sabotage. The students did at first perform sabotage to other students' articles, but this soon vanished completely. This was probably due to several factors, but it was an interesting tie-in to our description of the Wikipedia model, the most relevant site to compare our local implementation with.

Research questions

Would there be any motivation to use the Wiki as opposed to the traditional way of pen and paper?

This question has, in part, been answered in the interviews done with the students. The answer is clearly yes. The interview answers were reflected and well thought out, and the students seemed to favor the idea of this new way of doing things. It is unclear how well the Wiki or other internet technologies in general would work as opposed to the traditional way, but for the students' sake, they were very interested in working with these kinds of tools.

How can we motivate users to not only do what they have to do (i.e. homework), but maintain articles on a voluntary basis?

This is a very difficult question to answer. In this paper, some of the possibilities are explored, but there was not enough time to research all the different ones. The conclusion is seemingly that a strong core group is definitely necessary in order to achieve some kind of momentum, as well as more education on the core values of the Wiki. In summary, we require not only the teachers and the researchers to be enthusiastic about the technology, but also the students.

Is it possible to identify a core group of students who are active when it comes to maintaining the site?

In this paper (sadly), the answer is a clear and resounding no. It was not possible to identify such a core group. As mentioned earlier, it is possible to artificially create such a core group, either consisting of teachers willing to work on the articles on the Wiki, or a group of students

who take it upon themselves to continually extend the Wiki with information that is relevant to the students.

Part 6

Appendices

Appendix A – Administrator’s User Manual (In Norwegian)

Brukermanual for administratorer

SkoleWiki

Innholdsfortegnelse

Introduksjon.....	83
Administrere brukere	84
Opprette en ny konto	84
Slette, aktivere, deaktivere en konto	85
For å slette brukeren	86
For å deaktivere brukeren	86
For å aktivere brukeren	87
Oppgradere/degradere en brukerkonto	87
Oppgradere en konto	87
Degradere en konto.....	87
Administrere artikler	89
Slette en artikkel	90
Slette diskusjon på artikkel.....	90
Låse en artikkel.....	90
Fjerne lås på artikkel.....	90
Døpe om en artikkel.....	91
Rulle tilbake til en eldre versjon	91
Kategorier.....	92
Opprette en ny kategori	93
Slette en kategori.....	93
Flette sammen en kategori med en annen	94
Døpe om en kategori	94
Systemlogg	95
Diskbruk	95
Sikkerhetskopiering	96
Generell konfigurasjon.....	96

Generelle valg	96
Sikkerhetsvalg	98
Avanserte valg	100
Avanserte funksjoner	100
Plugins og providere	100
Listen over providere/plugins	101
Legge til en plugin	102
Aktivere en plugin	102
Deaktivere en plugin	102
Konfigurere en provider/plugin	103

Introduksjon

Denne manualen beskriver hvordan man utfører ofte brukte administrasjonsoppgaver på SkoleWiki.net.

For å starte administrasjon, klikk deg inn på SkoleWikis hjemmeside,

<http://www.skolewiki.net>.



Logg så inn med en administratorkonto.

Administrasjon - Wikiens startside >

Nettleseren kan få problemer med å vise denne siden.

[Admin startside](#) [Brukere/administratorer](#) [Sider/sikkerhetskopier](#) [Kategorier](#) [Snippet'er](#) [Nav. stier](#) [Filredigering](#) [Systemlogg](#) [Diskbruk](#) [provider'e](#) [Konfigurasjon](#) [Sikkerhetskopiering](#)

Administrasjonsstartsidens

Online brukere: 1
Brukerkonto: 26 (0 Inaktive)
Sider i hurtigbufferet: 1/69
Wiki versjon: 2.0.37 (Funnnet nyere versjon: 3.0.0.333)
Tjenerens opptid: 3d 4h 11m 55s (May be inaccurate)

[Slett hurtigbufferet \(cache\)](#)

Start webapplikasjonen på nytt

Du kan utføre en shutdown-og-restart av webapplikasjonen. To ganger vil du bli bedt om å bekrefte restart'en. Web applikasjonen vil starte på nytt etter den første anmodningen.

Advarsel: alle åpne sesjoner vil gå tapt og brukere kan oppleve feil.

Bemerk: Omstarten vil bare påvirke denne webapplikasjonen.

[Start applikasjonen på nytt](#) [Bekreft](#)

GNU General Public License

This Program is released under the GNU General Public License. You cannot remove this statement.
[View the GNU General Public License](#) or visit the [GNU website](#).

Du har nå logget inn og blir presentert med administratorens startside. Det er herfra du vil kunne gjøre alle administratoroppgavene.

Administrere brukere

Administrasjon - Wikiens startside »

Nettleseren kan få problemer med å vise denne siden.

Admin startside **Brukere/administratorer** Sider/sikkerhetskopier Kategorier Snippet'er Nav. stier Filredigering Syste

Brukere og administratorer

Vis brukere Vis administratorer Vis aktive kontoer Vis inaktive kontoer

aleksanderhe (Aleksander222@hotmail.com) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:23	▲
aleksanderho (aleksanderho@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:31	
andersod (oddli@hotmail.com) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:24	
anitapr (anitapr@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:20	
anneof (anneofs@hotmail.com) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:24	
atledy (atledy@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:17	
elikn (elikn@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:18	
hakongi (hakongi@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:17	
inaau (inaau@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21	
jensro (jensro@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:20	
jornbr (jornbr@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22	
juneaa (juneaa@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:16	
kristinekv (zico_dov@hotmail.com) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:18	
larsto (larsto@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21	
malingu (malingu@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22	
margaretta, Admin (margaret.tangen@orkdal.kommune.no) [MSSQL Users Storage Provider 1.1.2] 24/09/20	
martinho (martinho@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:20	▼

Slett Deaktiver Aktiver Rediger konto Opprett brukerkonto Gruppe epost

Oppgrader til admin Degradert til bruker Send melding

GNU General Public License

This Program is released under the GNU General Public License. You cannot remove this statement.
[View the GNU General Public License](#) or visit the [GNU](#) website.

Klikk på **Brukere/administratorer** for å administrere brukere.

Opprette en ny konto

Klikk på **Opprett brukerkonto** for å opprette en ny konto. Du vil da komme til dette bildet:

Opprett i **MSSQL Users Storage Provider 1.1.2** ▼

Brukernavn:

Epost:

Passord:

Passord (Gjentagelse):

Administrator

Fyll ut skjemaet for å opprette brukeren. Ønsker du at brukeren skal være en administrator, krysser du av for Administrator på bunnen av skjemaet.

Klikk "Opprett brukerkonto" for å lagre den nye kontoen. Hvis noe av informasjonen er feil, vil du få beskjed om dette, slik at du kan rette opp feilen og prøve å lagre på nytt.

Hvis lagringen lyktes, vil du bli flyttet til listen over brukere. Du vil da kunne se at din nye bruker har blitt lagt inn i systemet.

Administrasjon - Wikiens startside »

Nettleseren kan få problemer med å vise denne siden.

[Admin startside](#) [Brukere/administratorer](#) [Sider/sikkerhetskopier](#) [Kategorier](#) [Snippet'er](#) [Nav. stier](#) [Filredigering](#) [Systemlogg](#)

Brukere og administratorer

Vis brukere Vis administratorer Vis aktive kontoer Vis inaktive kontoer

hakongi (hakongi@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:17
inaau (inaau@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21
jensro (jensro@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:20
jornbr (jornbr@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22
juneaa (juneaa@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:16
kristinekv (zico_dov@hotmail.com) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:18
larsto (larsto@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21
malingu (malingu@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22
margaretta, Admin (margaret.tangen@orkdal.kommune.no) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22
martinho (martinho@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:30
melissamo (melissamo@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:19
minbruker (minbrukers@epost.no) [MSSQL Users Storage Provider 1.1.2] 08/04/2010 10:36
mohamadan (mohamadan@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21
noreha (noreha@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:17
odach (odach@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22
runaos (runaos@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:19
studals (studals@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22

GNU General Public License

This Program is released under the GNU General Public License. You cannot remove this statement.
[View the GNU General Public License](#) or visit the [GNU website](#).

Slette, aktivere, deaktivere en konto

For å endre en bruker, navigerer vi oss fram til listen over brukere, slik som i begynnelsen av dette kapittelet.

Velg brukeren du ønsker å endre.

Administrasjon - Wikiens startside »

Nettleseren kan få problemer med å vise denne siden.

Admin startside **Brukere/administratorer** Sider/sikkerhetskopier Kategorier Snippet'er Nav. stier Filredigering

Brukere og administratorer

Vis brukere Vis administratorer Vis aktive kontoer Vis inaktive kontoer

hakongi (hakongi@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:17
inaau (inaau@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21
jensro (jensro@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:20
jornbr (jornbr@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22
juneaa (juneaa@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:16
kristinekv (zico_dov@hotmail.com) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:18
larsto (larsto@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21
malingu (malingu@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22
margaretta, Admin (margaret.tangen@orkdal.kommune.no) [MSSQL Users Storage Provider 1.1.2] 24/09/20
martinho (martinho@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:30
melissamo (melissamo@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:19
minbruker (minbrukers@epost.no) [MSSQL Users Storage Provider 1.1.2] 08/04/2010 10:36
mohamadan (mohamadan@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:21
noreha (noreha@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:17
odach (odach@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22
runaos (runaos@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:19
sturdala (sturdala@skolewiki.net) [MSSQL Users Storage Provider 1.1.2] 24/09/2009 19:22

Slett Deaktiver Aktiver Rediger konto Oprett brukerkonto Gruppe epost
Oppgrader til admin Degrader til bruker Send melding

GNU General Public License

This Program is released under the GNU General Public License. You cannot remove this statement.
[View the GNU General Public License](#) or visit the [GNU website](#).

For å slette brukeren

Ved hjelp av denne knappen raderes all informasjon om brukeren fra systemet.

OBS! Ved å slette en bruker, kvitter systemet seg med all data om denne brukeren. Denne handlingen kan ikke rulles tilbake. Vurdér hvorvidt det vil passe bedre å bare deaktivere brukeren.

Klikk **Slett**.

En dialogboks med teksten "Er du sikker på at du vil fortsette?" vil poppe opp. Klikk **OK** hvis du er **helt sikker** på at du vil slette brukeren.

Brukeren vil fjernes fra brukerlisten, og teksten "Brukeren er slettet" vil stå i grønt under brukerlisten.

For å deaktivere brukeren

Deaktivering av en bruker betyr at den brukeren ikke får lov til å logge seg inn på Wikien.

Klikk **Deaktiver**.

Brukeren vil nå ligge i listen slik:

minbruker, Inactive (minbrukers@epost.no) [MSSQL Users Storage Provider 1.1.2] 08/04/2010 10:36

(Merk ordet **Inactive** etter brukernavnet)

Teksten "Brukeren er deaktivert" vil stå i grønt under brukerlisten.

For å aktivere brukeren

Denne handlingen forutsetter at brukeren er deaktivert. Ved hjelp av denne handlingen vil brukeren igjen kunne logge seg inn på Wikien.

Klikk **Aktiver**.

Teksten "Brukeren er aktivert" vil stå i grønt under brukerlisten.

Brukeren er nå aktivert.

Oppgradere/degradere en brukerkonto

Brukerkontoer kan oppgraderes til Administrator eller degraderes tilbake til vanlige brukere.

Kontoer med administratortilgang kan aksessere konfigurasjonssiden som beskrevet i dette dokumentet. Vanlige brukere kan ikke aksessere denne. Vanligvis er det lærere eller IT-admin som har administratortilgang, mens elever har vanlig brukertilgang.

Du må befinne deg i oversikten over brukere for å utføre de følgende handlingene.

Oppgradere en konto

OBS! Dette vil gi brukeren full tilgang til alle konfigurasjonsinnstillinger – dette inkluderer sletting av brukere eller sider! Vær helt sikker på at du velger riktig bruker før du klikker på denne knappen.

Velg brukeren i brukerlisten.

Klikk **Oppgrader til admin**.

Brukeren vil nå ligge som følger i brukerlisten:

```
minbruker, Admin (minbrukers@epost.no) [MSSQL Users Storage Provider 1.1.2] 08/04/2010 10:36
```

I tillegg vil teksten "Brukeren er oppgradert til Administrator" stå i grønt under brukerlisten.

Degradere en konto

Klikk på brukeren i brukerlisten.

Klikk **Degrader til bruker**.

Brukeren vil nå ligge som vanlig bruker i brukerlisten. I tillegg vil teksten "Administrator er degradert til bruker" stå i grønt under brukerlisten.

Administrere artikler

Klikk på menyvalget **Sider/sikkerhetskopier** for å komme til listen over artikler. Alle artikler på denne Wikien har [MSSQL Pages Storage Provider 1.3.6] skrevet etter seg. Dette betyr at artikkelen er fysisk lagret i en database (MSSQL) i stedet for i en fil.

Administrasjon - Wikiens startside »

Nettleseren kan få problemer med å vise denne siden.

Admin startside Brukere/administratorer **Sider/sikkerhetskopier** Kategorier Snippet'er Nav. stier Filredigering Systemlogg Diskbruk

Sider og sikkerhetskopier

Vis låste sider Vis normale sider Vis offentlige sider

(1833-1908) [MSSQL Pages Storage Provider 1.3.6]
Aleksander Kielland [MSSQL Pages Storage Provider 1.3.6]
Amalie Skram [MSSQL Pages Storage Provider 1.3.6]
An-Magritt Av Runa Osli [MSSQL Pages Storage Provider 1.3.6]
Anne Ofstad [MSSQL Pages Storage Provider 1.3.6]
Artikkel Nyrealismen [MSSQL Pages Storage Provider 1.3.6]
av oda [MSSQL Pages Storage Provider 1.3.6]
AVIIIIS [MSSQL Pages Storage Provider 1.3.6]
Avis [MSSQL Pages Storage Provider 1.3.6]
Avis av Håkon [MSSQL Pages Storage Provider 1.3.6]
Avis i skolen, av Tormod [MSSQL Pages Storage Provider 1.3.6]
Avis oppgaver [MSSQL Pages Storage Provider 1.3.6]
avisoppgaver [MSSQL Pages Storage Provider 1.3.6]
avisoppgaver - runa [MSSQL Pages Storage Provider 1.3.6]
avisoppgaver - runaaa [MSSQL Pages Storage Provider 1.3.6]
Avisprosjekt [MSSQL Pages Storage Provider 1.3.6]
Bruke historikk [MSSQL Pages Storage Provider 1.3.6]

Gå til Slett Slett disk. Sidestatus: Normal Sett status Slett alle sikkerhetskopier Slett eldre sikkerhetskopier Rull tilbake
Døp om til

GNU General Public License

This Program is released under the GNU General Public License. You cannot remove this statement.
View the GNU General Public License or visit the GNU website.

Det første man må gjøre for å kunne administrere en artikkel er å klikke på den artikkelen i sidelisten. Den høyre listen vil da oppdatere seg med en liste over revisjoner for den valgte artikkelen.

(1833-1908) [MSSQL Pages Storage Provider 1.3.6]	Rev 3, 2009/10/05 09:36:21
Aleksander Kielland [MSSQL Pages Storage Provider 1.3.6]	Rev 2, 2009/10/05 09:33:58
Amalie Skram [MSSQL Pages Storage Provider 1.3.6]	Rev 1, 2009/09/28 09:30:31
An-Magritt Av Runa Osli [MSSQL Pages Storage Provider 1.3.6]	Rev 0, 2009/09/28 09:28:21
Anne Ofstad [MSSQL Pages Storage Provider 1.3.6]	
Artikkel Nyrealismen [MSSQL Pages Storage Provider 1.3.6]	
av oda [MSSQL Pages Storage Provider 1.3.6]	
AVIIIIS [MSSQL Pages Storage Provider 1.3.6]	
Avis [MSSQL Pages Storage Provider 1.3.6]	
Avis av Håkon [MSSQL Pages Storage Provider 1.3.6]	
Avis i skolen, av Tormod [MSSQL Pages Storage Provider 1.3.6]	
Avis oppgaver [MSSQL Pages Storage Provider 1.3.6]	
avisoppgaver [MSSQL Pages Storage Provider 1.3.6]	
avisoppgaver - runa [MSSQL Pages Storage Provider 1.3.6]	
avisoppgaver - runaaa [MSSQL Pages Storage Provider 1.3.6]	
Avisprosjekt [MSSQL Pages Storage Provider 1.3.6]	
Bruke historikk [MSSQL Pages Storage Provider 1.3.6]	

Man kan navigere seg fram til artikkelen man har valgt ved å klikke **Gå til**.

Slette en artikkel

OBS! Ved å slette en artikkel, vil all historisk data som gjelder den artikkelen forsvinne. Denne handlingen kan ikke reverseres – vær helt sikker på at du vil slette artikkelen før du gjør dette.

Velg en artikkel ved å følge prosedyren nevnt over.

Klikk **Slett**.

En boks vil dukke opp med teksten "Er du sikker på at du vil fortsette?" Klikk **OK**.

Teksten "Siden er blitt slettet" vil dukke opp i grønt under artikkellisten.

Slette diskusjon på artikkel

OBS! Denne handlingen kan ikke reverseres og bør brukes forsiktig. Vær helt sikker på at du har valgt riktig artikkel og faktisk vil slette den artikkelens diskusjon.

Funksjonen vil medføre at all diskusjon omkring en sides innhold vil fjernes.

Velg siden det gjelder og klikk **Slett disk**.

Låse en artikkel

I noen tilfeller kan man låse en artikkel, i det tilfelle nevnte artikkel blir sabotert eller misbrukt på en eller annen måte.

Låsing av en artikkel medfører at vanlige brukere ikke kan redigere den artikkelen.

Velg artikkelen det gjelder

testside [MSSQL Pages Storage Provider 1.3.6]

Sett **Sidestatus** til **Locked** i stedet for **Normal**. Klikk så **Sett status**.

Sidestatus:

Artikkelen det gjelder vil da stå som følger

testside (Locked) [MSSQL Pages Storage Provider 1.3.6]

Og teksten "Siden er satt til låst" vil stå i grønt under artikkellisten.

Fjerne lås på artikkel

Følg oppskriften over, men velg **Sidestatus Normal** i stedet for **Locked**.

Klikk så på **Sett status**.

Teksten "Siden er satt til normal" vil stå i grønt under artikkellisten.

Døpe om en artikkel

Denne funksjonen kan brukes hvis en ønsker å gi en artikkel et nytt navn.

Velg artikkelen det gjelder i artikkellisten.

Skriv inn det ønskede navnet i tekstboksen ved siden av **Døp om til**.

Døp om til	testside2
------------	-----------

Klikk **Døp om til**.

En dialogboks vil komme opp med teksten "Er du sikker på at du vil fortsette?". Klikk **OK**.

Artikkelen vil vises med det nye navnet i artikkellisten, og teksten "Siden har fått nytt navn" vil stå i grønt under artikkellisten.

Rulle tilbake til en eldre versjon

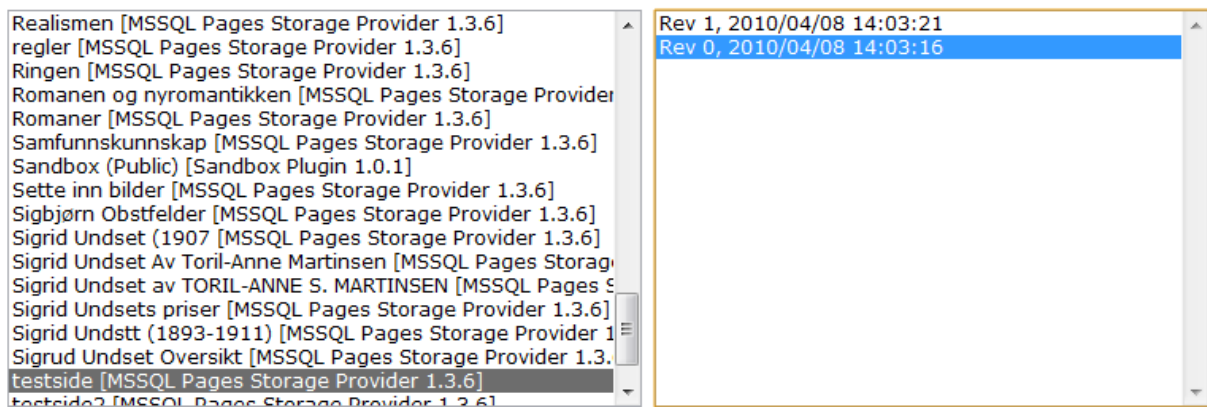
OBS! Denne funksjonen vil rulle tilbake en artikkel til en eldre versjon. Dette medfører at all historikk registrert etter denne eldre versjonen vil forsvinne. Handlingen kan ikke reverseres (man kan ikke rulle fram igjen til en nyere versjon). Vær helt sikker på at du vil utføre denne handlingen før du gjør det.

Denne funksjonen tillater administratoren å rulle tilbake nåværende artikkelrevisjon til en eldre revisjon. Dette kan være nyttig i tilfelle en artikkel har blitt sabotert av en bruker.

Velg siden du vil rulle tilbake.

Realismen [MSSQL Pages Storage Provider 1.3.6] regler [MSSQL Pages Storage Provider 1.3.6] Ringene [MSSQL Pages Storage Provider 1.3.6] Romanen og nyromantikken [MSSQL Pages Storage Provider 1.3.6] Romaner [MSSQL Pages Storage Provider 1.3.6] Samfunnskunnskap [MSSQL Pages Storage Provider 1.3.6] Sandbox (Public) [Sandbox Plugin 1.0.1] Sette inn bilder [MSSQL Pages Storage Provider 1.3.6] Sigbjørn Obstfelder [MSSQL Pages Storage Provider 1.3.6] Sigrid Undset (1907) [MSSQL Pages Storage Provider 1.3.6] Sigrid Undset Av Toril-Anne Martinsen [MSSQL Pages Storage Provider 1.3.6] Sigrid Undset av TORIL-ANNE S. MARTINSEN [MSSQL Pages Storage Provider 1.3.6] Sigrid Undsets priser [MSSQL Pages Storage Provider 1.3.6] Sigrid Undsttt (1893-1911) [MSSQL Pages Storage Provider 1.3.6] Sigrid Undset Oversikt [MSSQL Pages Storage Provider 1.3.6] testside [MSSQL Pages Storage Provider 1.3.6] testside2 [MSSQL Pages Storage Provider 1.3.6]	Rev 1, 2010/04/08 14:03:21 Rev 0, 2010/04/08 14:03:16
--	--

Velg revisjonen du ønsker å rulle tilbake til.



Klikk **Rull tilbake**.

En boks vil poppe opp med teksten "Er du sikker på at du vil fortsette?". Klikk **OK** til dette.

Siden er nå rullet tilbake.

Kategorier

Kategorier hjelper en bruker med å sette en side inn i en kontekst. En side kan ha flere enn en kategori. Eksempler er: Historie, Norsk, Samfunnsfag, osv.

For å begynne, klikk menyvalget **Kategorier** på toppen av siden. Listen over kategorier i Wikien vil vises.

Merk at ved siden av hver kategori er et tall i parentes. Dette tallet viser hvor mange artikler som har denne kategorien.

Kategorier

Her kan du behandle kategoriene i denne wiki'en.

Historie (8) [MSSQL Pages Storage Provider 1.3.6]
Norsk (49) [MSSQL Pages Storage Provider 1.3.6]
Samfunnsfag (2) [MSSQL Pages Storage Provider 1.3.6]

Slett

Døp om til eller **Opprett ny** i MSSQL Pages Storage Provider 1.3.6 ▼

Flett sammen med <-- Den nye kategorien får navnet til denne kategorien.

GNU General Public License

This Program is released under the GNU General Public License. You cannot remove this statement.
[View the GNU General Public License](#) or visit the [GNU website](#).

Opprette en ny kategori

Kategorier kan opprettes slik at brukere kan kategorisere artiklene sine.

Begynn med å skrive inn navnet på den nye kategorien i feltet til høyre for **Døp om til**-knappen.

Slett

Døp om til Min kategori eller **Opprett ny** i MSSQL Pages Storage Provider 1.3.6 ▼

Flett sammen med <-- Den nye kategorien får navnet til denne kategorien.

Trykk **Opprett ny**. Du trenger ikke å gjøre endringer i feltet der det står MSSQL Pages Storage Provider.

Teksten "Kategori er lagret" vil stå i grønt under kategorilisten.

Slette en kategori

OBS! Dette valget sletter kategorien og alle tilknytninger mellom en artikkel og den kategorien i Wikien. Vær helt sikker på at det er dette du vil gjøre før du bruker denne funksjonen. Denne handlingen kan ikke reverseres.

Velg kategorien du ønsker å slette.

Klikk **Slett**.

En boks med teksten "Er du sikker på at du vil fortsette?" vil dukke opp. Klikk **OK**.

Teksten "Kategorien er slettet" vil stå i grønt under kategorilisten.

Flette sammen en kategori med en annen

Denne funksjonen lar deg flette sammen en kategori med en ny. Dette medfører tap av data, i det at alle sider med den gamle kategorien nå vil få den nye kategorien i stedet. Vær derfor sikker på at det er dette du vil før du utfører denne handlingen.

Kategorier

Her kan du behandle kategoriene i denne wiki'en.

Historie (8) [MSSQL Pages Storage Provider 1.3.6]

Min kategori (0) [MSSQL Pages Storage Provider 1.3.6]

Norsk (49) [MSSQL Pages Storage Provider 1.3.6]

Samfunnsfag (2) [MSSQL Pages Storage Provider 1.3.6]

Slett

Døp om til eller i

Flett sammen med <-- Den nye kategorien får navnet til denne kategorien.

Velg kategorien du ønsker å flette inn i en annen kategori. Her har vi valgt **Min kategori**.

Velg kategorien denne skal flettes inn i. Her har vi valgt **Historie**.

Klikk **Flett sammen med**.

En dialogboks vil poppe opp med teksten "Er du sikker på at du vil fortsette?". Klikk **OK**.

Teksten "Kategorier er flettet sammen" vil stå i grønt under kategorilisten.

Alle artikler med kategorien Min kategori vil nå få kategorien Historie i stedet.

Døpe om en kategori

Velg kategorien du ønsker å døpe om.

Skriv inn det nye navnet i teksten til høyre for **Døp om til**-knappen.

Klikk **Døp om til**.

Teksten "Kategori er gitt nytt navn" vil stå i grønt under kategorilisten.

Systemlogg

Systemloggen aksesseres ved å klikke **Systemlogg**-knappen i administratormenyen.

Systemloggen gir et overblikk over de siste hendelsene som har skjedd på Wiki-serveren.

Vanligvis er disse ikke særlig relevante hvis man ikke leter etter feil.

Systemlogg		
<input checked="" type="checkbox"/> Vis meldinger	<input checked="" type="checkbox"/> Vis advarsler	<input checked="" type="checkbox"/> Vis feil
Dato/Klokkeslett	Bruker	Melding
13/04/2010 10:21	SYSTEM	Category Min kategori 5 removed
13/04/2010 10:21	admin	Category removal requested for Min kategori 5
13/04/2010 10:21	SYSTEM	Category Min kategori renamed to Min kategori 5
13/04/2010 10:21	admin	Category rename requested for Min kategori (Min kategori 5)
13/04/2010 10:20	SYSTEM	Category Testkategori 2 merged into Min kategori
13/04/2010 10:20	admin	Categories Merging requested for Testkategori 2 into Min kategori
13/04/2010 10:20	SYSTEM	Category Testkategori 2 created
13/04/2010 10:20	admin	Category creation requested for Testkategori 2
13/04/2010 10:12	SYSTEM	User admin logged in through cookie
13/04/2010 03:00	SYSTEM	ScrewTurn Wiki is ready
13/04/2010 03:00	SYSTEM	Provider Basic Statistics Plugin 1.3.3 loaded (Enabled)
13/04/2010 03:00	SYSTEM	Provider Multilanguage Content Plugin 1.2.1 loaded (Enabled)
13/04/2010 03:00	SYSTEM	Provider MySQL Pages Storage Provider 1.3.5 loaded (Disabled)

Det kan likevel være lurt å gjøre seg kjent med systemloggen for når man skal feilsøke.

Diskbruk

Diskbruk velges ved å klikke på **Diskbruk**-knappen i menyen.

For en IT-administrator kan det være nyttig å vite hvor mye plass Wiki-installasjonen tar på serveren, slik at man vet når man må oppgradere for å skalere til en voksende Wiki.

Klikk **Beregn**.

Diskbruk

Total bruk av diskplass: **4,87 MB**

- Sidens bruk av diskplass: **3,48 KB**
- Diskplass brukt av filer: **3,29 MB**
- Loggens bruk av diskplass: **63,93 KB**
- Midlertidig filers bruk av diskplass: **0 B**

Beregn

Du vil få opp et detaljert bilde over hvor mye diskplass som brukes av hva.

Sikkerhetskopiering

Administrator aksesserer sikkerhetskopiering-siden ved å klikke **Sikkerhetskopiering** i menyen.

Sikkerhetskopiering

Klikk på knappen nedenfor for å laste ned en fullstendig sikkerhetskopi av alle data i wiki'en. **Bemerk:** handlingen kan ta en tid.

Last ned Sletter mappen Temp

Ved å klikke **Last ned** blir en sikkerhetskopi av alle Wikiens innstillinger lastet ned til administratorens datamaskin.

Merk at dette ikke tar med den viktigste informasjonen – den som ligger i databasen. For dette må man inn på SQL-databasen.

Det er anbefalt å sette opp automatisk sikkerhetskopi av SQL-databasen.


Generell konfigurasjon

Klikk **Konfigurasjon**-knappen i menyen for å få opp Wikiens konfigurasjonsinnstillinger.

Generelle valg

Rediger wiki konfigurasjon og klikk deretter lagre knappen.

Generelle valg

Wikittittel	<input type="text" value="Orkanger Ungdomsskole Wiki"/>
Hoved URL	<input type="text" value="http://www.skolewiki.net/"/>
Tema	<input type="text" value="Default"/>
Kontakpersonens epost adresse	<input type="text" value="chrizach@gmail.com"/>
Epost adresse til avsender	<input type="text" value="no-reply@skolewiki.net"/>
SMTP tjener	<input type="text" value="smtp.server.com"/>
SMTP bruker/passord/port	<input type="text"/> <input type="text"/> <input type="text"/>
	<input type="checkbox"/> Aktiver SSL for SMTP tjeneren
Standard side	<input type="text" value="MainPage"/>
	<input type="checkbox"/> Aktiver dobbeltklikk sideredigering
	<input type="checkbox"/> Brotkrümelnavigation deaktivieren
	<input type="checkbox"/> Process Single Line Breaks
Dato- og klokkeslett format	<input type="text" value="dd '/' MM '/' yyyy ' ' HH ':' mm"/> 
Standard språk	<input type="text" value="Norsk, bokmål (Norge) - Norwegian, Bokmål (Norway)"/>
Standard tidssone	<input type="text" value="(GMT+01:00) Central European"/>

Hvis man ser forbi litt mangler i oversettelsen, er det et par ting her man kan bruke for å konfigurere Wikien til sine behov.

- Wikitittel – Wikiens tittel. Endre denne til navnet på din installasjon.
- Hoved URL – Endre denne til adressen du bruker for å aksessere Wikien.
- Tema – Endrer farger og stiler for det grafiske grensesnittet. Prøv deg fram med de innebygde temaene, eller lag nye. Denne brukermanualen detaljerer ikke hvordan man lager nye temaer, men man kan få et raskt innblikk ved hjelp av denne nettsiden: <http://channel9.msdn.com/posts/howarddierking/Extreme-ASPNET-Makeover-iQuery-Re-Theming-ScrewTurn-Wiki/>
- Kontaktpersonens epost-adresse – Epost-adresse til kontaktperson/administrator.
- Epost-adresse til avsender – epostadressen som vises i Fra-feltet på utsendte eposter.
- SMTP tjener – adresse til datamaskin som skal brukes for utsending av e-post. Sett denne til din interne SMTP-tjener.
- SMTP bruker/passord/port/Aktiver SSL for SMTP tjeneren – samme som over.
- Standard side – Startsidene når en bruker navigerer inn på Wikien.
- Aktiver dobbeltklikk sideredigering – denne innstillingen gir brukerne mulighet til å dobbeltklikke med musen på en seksjon for å gå inn i redigeringsvisningen for denne seksjonen. Vanligvis må de klikke på linken "Rediger denne siden".
- Deaktivere brødsmler (Brotkrümelnavigation deaktivieren) – Når man navigerer i Wikiens artikler, vil man se en liste over sider man tidligere har besøkt. Man kan klikke på sider i denne listen for å komme tilbake til disse. Dette konfigurasjonsalternativet lar administratoren slå av denne funksjonaliteten.
- Prosessere enkelt linjeskift – Når brukere redigerer en artikkel vil denne funksjonen avgjøre om et enkelt trykk på Enter vil gjenkjennes eller ikke.
- Dato og klokkeslett format – Denne innstillingen avgjør hvordan Wikien prosesserer tid og dato. Det forhåndsinnstilte formatet er vanligvis OK. Hvis en avansert bruker ønsker å endre dette, kan denne referansen være grei å ha:
 - dd: Dag i måneden (to siffer)
 - MM: Måned (to siffer)
 - yyyy: År (fire siffer)
 - HH: Time (to siffer)
 - mm: Minutt (to siffer)
- Standard språk – Her kan administrator endre hvilket språk som vises for hjelpetekster over hele Wikien.

- Standard tidssone – Endre denne til hvilken tidssone Wikiens installasjon befinner seg i.

Sikkerhetsvalg

Sikkerhetsvalg

	<input type="checkbox"/> Tillat brukere å registre seg
	<input checked="" type="checkbox"/> Require both Username and Email to reset a password
Kontoens aktiveringsmodus	New Account must be activated by Admins ▾
	<input type="checkbox"/> Deaktiver captcha
Diskusjonsrettigheter	Same as owner Page (default) ▾
	<input checked="" type="checkbox"/> La brukere få opprette nye sider
	<input type="checkbox"/> Tillat brukere å opprette nye kategorier
	<input checked="" type="checkbox"/> Tillat brukere å behandle sidekategorier
	<input checked="" type="checkbox"/> Tillat brukere å se filer som er laste opp
	<input checked="" type="checkbox"/> Tillat brukere å laste opp filer
	<input checked="" type="checkbox"/> Tillat brukere å slette filer
	<input checked="" type="checkbox"/> Privat tilgang
	<input type="checkbox"/> Offentlig tilgang
	<input type="checkbox"/> Tillat anonym filbehandling i offentlig tilgangsmodus
	<input type="checkbox"/> Deaktiver samtidig sideredigering
Tillatte filtyper (CSV)	jpg, jpeg, gif, png, tif, tiff, bmp, htm, html, zip, rar
	<input type="checkbox"/> Tillat script tag'er
Loggnivå	Alle meldinger ▾
Maximale Gröbe der Protokolldatei (KB)	64
	<input checked="" type="checkbox"/> Gi administratorene adgang til konfigurasjonssiden

- Tillat brukere å registrere seg – Denne funksjonen avgjør om brukere får lov til å registrere seg selv, eller om en administrator må gjøre det for dem. Vær sikker på at du vet hva du gjør før du endrer denne innstillingen.
- Kreve både brukernavn og epost for å resette et passord (Require both Username and Email to reset a password) – Hvis denne innstillingen er av, kreves bare brukernavnet til brukeren som ønsker å resette sitt passord.
- Kontoens aktiveringsmodus – forteller systemet hva brukere må gjøre for å kunne logge inn på systemet etter at kontoen deres er registrert.
 - **Bruker må aktivere via epost (User must activate via Email)** – Når bruker registrerer seg selv, mottar han en epost til den epostadressen han har oppgitt. Brukeren må deretter klikke på en link i denne eposten for å aktivere kontoen sin.
 - **Nye kontoer må aktiveres av admin (New accounts must be activated by Admins)** – Når en konto har blitt registrert, enten av en bruker eller en administrator (avgjøres av 'Tillat brukere å registrere seg'-innstillingen nevnt

tidligere), må en administrator gå inn på kontooversikten og aktivere denne før brukeren får lov til å logge seg inn på systemet.

- **Nye kontoer aktiveres automatisk (New accounts are activated by default)** – Hvis dette valget er gjeldende, vil en nyregistrert konto automatisk være aktivert. Dette er sannsynligvis det beste valget hvis 'Tillat brukere å registrere seg' er slått av.
- Deaktiver captcha – captcha er en sikkerhetsmekanisme som viser et bilde under kontoregistrering. Brukeren som skal registrere seg må lese noen bokstaver i bildet, for å å skrive disse inn i en tekstboks. Hvis bokstavene er skrevet feil får ikke brukeren registrert seg. Meningen med dette er å gjøre slik at automatiske registreringsagenter ikke registrerer seg på Wikien. Denne innstillingen har ingen effekt hvis 'Tillat brukere å registrere seg' er slått av, da administrator ikke trenger å fylle inn captcha-feltet.
- La brukere få opprette nye sider / Tillat brukere å opprette nye kategorier / Tillat brukere å behandle sidekategorier / Tillat brukere å se filer som er lastet opp / Tillat brukere å laste opp filer / Tillat brukere å slette filer – disse innstillingene er relativt selvforklarende og bestemmer hvilke rettigheter en vanlig bruker har. En administrator vil alltid ha disse mulighetene.
- Privat tilgang – Brukere må logge inn på Wikien for å kunne lese og redigere artikler.
- Offentlig tilgang – Brukere trenger ikke å logge seg inn på Wikien for å lese eller redigere artikler – disse handlingene kan utføres anonymt (dette er en mulig sikkerhetsrisiko).
- Tillat anonym filbehandling i offentlig tilgangsmodus – Hvis 'offentlig tilgang' er slått på, kan anonyme brukere behandle filene som er lagret på Wikien (dette er en sikkerhetsrisiko).
- Deaktiver samtidig sideredigering – hvis denne er slått av, kan flere enn en bruker redigere samme side samtidig. Hvis denne er slått på, kan kun en bruker ha åpen redigeringsvinduet om gangen.
- Tillatte filtyper (CSV) – en kommaseparert liste over filtyper som er tillatt å laste opp på Wikien. Hvis man ønsker å legge til en filtype her, bruker man filens endelse (for Word-filer som har .doc-endelser skriver man inn doc).
- Tillat script-tager – dette er en stor sikkerhetsrisiko, fordi det tillater brukere å legge inn JavaScript-kode direkte i artikkelen. Denne JavaScript-koden kan utføre handlinger på senere leseres datamaskin, og det er **ikke anbefalt å aktivere denne innstillingen**.

- Loggnivå – denne kan vanligvis stå på Alle meldinger, så lenge man ikke setter 'Maksimal loggstørrelse' for høyt.
- Maksimal loggstørrelse (Maximale Größe der Protokolldatei (KB)) – angir maksimal størrelse på loggfilen i kilobyte. Det forhåndsdefinerte 64 KB er greit å ha. Wikien vil automatisk kontrollere at størrelsen alltid er riktig, ved å slette gamle loggmeldinger ved behov.
- Gi administratorene adgang til konfigurasjonssiden – la denne innstillingen være på.

Avanserte valg

Avanserte valg (rediger forsiktig)

	<input type="checkbox"/> Deaktivere die automatische Versionsüberprüfung
Største filstørrelse (KB)	<input type="text" value="10240"/>
	<input type="checkbox"/> Deaktiver hurtigbufferet (cache)
Hurtigbufferets størrelse (cache)	<input type="text" value="100"/>
Hurtigbufferets klippestørrelse (cache)	<input type="text" value="20"/>
Antall forsøk på å åpne filer	<input type="text" value="10"/>
Ventetid mellom forsøk på å åpne en fil(ms)	<input type="text" value="50"/>
	<input type="checkbox"/> Aktiver 'viewstate' komprimering
	<input type="checkbox"/> Aktiver HTTP komprimering
Regex brukernavn validering	<input type="text" value="^[\\w\\ !\\\$@%\\^\\.\\(\\)]{3,25}\$"/>
Regex passord validering	<input type="text" value="^[\\w~!@#\\\$%\\^\\(\\)\\[\\]\\{\\}\\ \\.\\,=\\-_\\]{6,}"/>

Disse innstillingene bør vanligvis ikke endres.

Lagring av konfigurasjon

Når du er ferdig med å redigere konfigurasjonen kan du lagre eller avbryte.

Bruk **Lagre** eller **Avbryt**-knappene for dette. Teksten "Vellykket lagring av konfigurasjon" vil stå i grønt på bunnen av konfigurasjonsseksjonene.

Avanserte funksjoner

Plugins og providere

Klikk på **Provider'e**-valget i menyen for å komme til denne siden.

Denne siden tillater administrator å endre innstillingene til hvilke lagringsprovidere som er i bruk av Wikien. Det er også mulig å laste inn såkalte plugins, som endrer funksjonaliteten til Wikien på et bestemt vis.

På dette nettstedet kan man finne både offisielle og tredjeparts plugins som kan brukes i Wikien: <http://www.screwturn.eu/Customize.MainPage.ashx>

Providerbehandling

Her kan du behandle provider'ne som er installer i denne wiki'en.

[P] MSSQL Pages Storage Provider 1.3.6 [P] Sandbox Plugin 1.0.1 [U] MSSQL Users Storage Provider 1.1.2 [F] Multilanguage Content Plugin 1.2.1 [F] Basic Statistics Plugin 1.3.3 [P] Local Pages Provider - Deaktivert [P] MySQL Pages Storage Provider 1.3.5 - Deaktivert [U] Local Users Provider - Deaktivert [U] MySQL Users Storage Provider 1.1.1 - Deaktivert

Konfigurer Aktiver Deaktiver Last ut

Standard provider

Standard sideprovider: MSSQL Pages Storage Provider
Standard burkerprovider: MSSQL Users Storage Provider
Lagre Avbryt

Legg til nye provider'e

Her kan du laste opp DLL filer som inneholder provider'e, og eventuelt fjerne de DLL'ene som du ikke bruker. Dersom du fjerner en DLL så deaktiveres ikke automatisk de tilhørende provider'ne. Men laster du opp en ny DLL så vil provider'ne som filen inneholder, lastes automatisk.

PluginPack.dll	Browse...
	Last opp
Fjern	

Listen over providere/plugins

Her kan du behandle provider'ne som er installer i denne wiki'en.

[P] MSSQL Pages Storage Provider 1.3.6 [P] Sandbox Plugin 1.0.1 [U] MSSQL Users Storage Provider 1.1.2 [F] Multilanguage Content Plugin 1.2.1 [F] Basic Statistics Plugin 1.3.3 [P] Local Pages Provider - Deaktivert [P] MySQL Pages Storage Provider 1.3.5 - Deaktivert [U] Local Users Provider - Deaktivert [U] MySQL Users Storage Provider 1.1.1 - Deaktivert

Konfigurer Aktiver Deaktiver Last ut

Øverst på siden vil det vises en liste over installerte providere og plugins. Referer til seksjonene under for å behandle disse.

Det er verdt å merke seg hva bokstavene i klammer betyr:

- U: User provider – en provider som forteller Wikien hvordan den skal lagre og behandle brukerinformasjon.
- P: Page provider – en provider som forteller Wikien hvordan den skal lagre og behandle sideinformasjon.
- F: Formatter plugin – en provider som endrer hvordan tekst i en artikkel blir formatert.

Legge til en plugin

Vær sikker på at du har lastet ned pluginen du vil installere til din lokale datamaskin. Plugin-filen vil vanligvis komme pakket inn i en zip-fil. Denne pakkes ut, slik at man har en DLL-fil.

Klikk **Browse...**

Naviger fram til filen du vil installere. Klikk **Åpne**.

Klikk **Last opp**.

Les seksjonen om å aktivere/deaktivere providere for å behandle pluginen du nettopp lastet opp.

Aktivere en plugin

Her kan du behandle provider'ne som er installer i denne wiki'en.

[P] MSSQL Pages Storage Provider 1.3.6
[P] Sandbox Plugin 1.0.1
[U] MSSQL Users Storage Provider 1.1.2
[F] Multilanguage Content Plugin 1.2.1
[F] Basic Statistics Plugin 1.3.3
[P] Local Pages Provider - Deaktivert
[P] MySQL Pages Storage Provider 1.3.5 - Deaktivert
[U] Local Users Provider - Deaktivert
[U] MySQL Users Storage Provider 1.1.1 - Deaktivert

I listen over plugins og providere velger du pluginen du vil aktivere.

Klikk **Aktiver**.

Deaktivere en plugin

Velg pluginen du vil deaktivere.

Klikk **Deaktiver**.

Konfigurere en provider/plugin

Hver plugin/provider vil ha forskjellige former for konfigurasjonsmuligheter. Vi går derfor ikke inn på dem i denne brukermanualen.

Appendix B –User Manual (In Norwegian)

Brukermanual

SkoleWiki

Innholdsfortegnelse

Introduksjon.....	107
Logge inn/ut.....	107
Din første artikkel	107
Første steg	107
Andre steg	108
Tredje steg.....	109
Lenker	109
Bilder	109
Historikk	110
Se et bilde av en artikkel på et gitt tidspunkt	110
Rulle tilbake til en tidligere versjon	111
Sammenligne to artikkelversjoner	111
Diskusjon.....	112

Introduksjon

Denne brukermanualen detaljerer hvordan en bruker skal utføre de vanligste handlingene i SkoleWiki. Den er kun ment som en basis, slik at lærere eller IT-administrator kan lage en presentasjon for visning til elevene sine.

For å starte, åpne en nettleser og navigér deg fram til <http://www.skolewiki.net>.

Logge inn/ut

Logg inn

Her kan du logge inn til wiki'en. Du vil bli i stand til å redigere og op
Har du ingen konto? Du kan [opprette en](#).

Brukernavn:

Passord:

Husk meg

[Har du glemt passordet?](#)

For å kunne bruke SkoleWiki må du ha et brukernavn og et passord. Hvis du ikke vet hva brukernavnet eller passordet ditt er, kontakt en administrator. (Ikke bruk **Har du glemt passordet?**-linken)

Skriv brukernavnet og passordet inn i feltene og trykk **Logg inn**.

Du vil bli videresendt til **Hovedsiden**.

Din første artikkel

Første steg

1. For å starte, klikk "Lag en ny side" på menyen til venstre.
2. I feltet til høyre for "Opprett side" skriver du navnet på artikkelen. I dette eksemplet bruker du navnet ditt.

3. I feltet til "Sidetittel" skriver du det samme.
4. Hopp ned til det store tekstfeltet og skriv noe, for eksempel "Jeg heter Per Olsen, og jeg er 15 år".
5. Trykk Forhåndsvis, og så "Lagre element". Du vil få se siden din:

Orkanger Ungdomsskole Wiki

Navigation RSS

- [Hovedside](#)
- [Tilfeldig side](#)
- [Lag en ny side](#)
- [Alle sider](#)
- [Kategorier](#)
- [Administrasjon](#)
- [Filbehandling](#)
- [Logg inn/ut](#)
- [Språkvalg](#)
- [Din profil](#)
- [Lage konto](#)

Søk

[Avansert søk »](#)

Per Olsen

Endret: 24/09/2009 18:48 av [admin](#) - *Ikke kategorisert*
[\(4\)](#) » [The Sandbox](#) » [Hvordan skrive din første artikkel](#) » **Per Olsen**

Jeg heter Per Olsen, og jeg er 15 år.

ScrewTurn Wiki version 2.0.37. Some of the icons created by FamFamFam

Gratulerer – du har nå laget din første artikkel!

Andre steg

Nå skal vi gjøre siden litt finere ved å bruke formatering på teksten din.

1. Trykk på "Rediger denne siden"-linken. Du finner den oppe i høyre hjørne på siden.
2. Velg deg ut en effekt du vil ha på teksten din i verktøylinjen over tekstfeltet. Det ser slik

ut:




3. I dette tilfellet har jeg valgt fet skrift (den første knappen i verktøylinjen).
4. Merk noe tekst, for eksempel "Jeg heter".
5. Trykk på fet skrift-knappen. I tekstfeltet vil det nå stå "'Jeg heter'".
6. Klikk Forhåndsvis og bla deg nedover på siden for å se hvordan resultatet vil bli når du lagrer.
7. Prøv deg frem og lag tekst som bruker minst tre av de ti første knappene på verktøylinjen. Hvis du står fast kan du bare klikke på Avbryt-knappen.
8. Stykk opp teksten ved å bruke knappene H1 til H4 (overskrifter).
9. Klikk lagre element.


Tredje steg

Lenker

Nå skal vi lage en lenke til en annen side på Wikien.

1. Klikk først på "Rediger denne siden" så du kommer inn i redigeringsmodus igjen.
2. Klikk på -knappen for å begynne. Her vil du få opp en liste over mulige sider man kan lenke til.
3. Klikk "MainPage" for å lage en lenke til hovedsiden. En boks vil komme opp som spør deg etter lenkens tittel.
4. Skriv "Hovedside" her og trykk OK.
5. Du vil se at [MainPage|Hovedside] har kommet inn i tekstfeltet ditt.
6. Klikk "Lagre element".
7. Hold ctrl på tastaturet og prøv å trykke på "Hovedside"-lenken for å åpne et nytt vindu med hovedsiden i.

Bilder

1. Klikk på -knappen for å sette inn et bilde. Du vil få opp en slik boks:

Sett inn et bilde


Bildejustering
 høyre venstre Auto Intern (inline)

Bildets tittel

Bildets URL
 [Bla gjennom](#)

Lenke (valgfri)
 [Bla gjennom](#) - [Velg fil](#) - [Velg side](#)

OK - Avbryt

2. I det midterste feltet (ved siden av Bildets URL) trykker du "Bla gjennom"
3. Velg et av eksempelbildene og klikk OK.
4. Klikk OK. [imageright|bilde.png] vil vises i teksten din.

5. Trykk på forhåndsvis for å se hvordan bildet vil vises på siden din.
6. Du kan prøve å bytte ut "imageright" i teksten med "imageleft" eller bare "image", og så Forhåndsvis for å se hvordan bildet flytter seg.
7. Trykk på "Lagre element"

Historikk

Stå på artikkelen du ønsker å se historikk for.

Klikk på [Historikk](#) -knappen i øverste høyre hjørne for å begynne.

Du vil gå videre til listen over endringer for artikkelen.

#	Revisjon	Bruker
	Aktuelle Versjon - 15/04/2010 10:08	admin
36	15/04/2010 10:08	admin
35	08/03/2010 09:57	noreha
34	08/03/2010 09:42	margaretta
33	08/03/2010 09:41	margaretta
32	08/03/2010 09:41	margaretta
31	08/03/2010 09:38	margaretta
30	08/03/2010 09:38	margaretta
29	08/03/2010 09:38	margaretta
28	08/03/2010 08:54	margaretta
27	08/12/2009 12:48	hakongi
26	01/12/2009 13:31	atledy

Se et bilde av en artikkel på et gitt tidspunkt

Klikk på en av datoene for å få se hva som ble gjort akkurat da.

Sidehistorikk: Hovedside

Sammenlign sideversjoner

36 ▾

Aktuell ▾

Sammenlign

« [Eldre versjon](#) - [Tilbake til sidehistorikk](#) - [Nyere versjon](#) »

Sideversjon: 08/03/2010 09:38

Velkommen til **Orkanger Ungdomsskole Wiki!**

På denne wikien finnes det 69 sider.

Dette er hovedsiden til Skolewiki for Orkanger Ungdomsskole.

Notater

Du kan bruke linkene **Eldre versjon** og **Nyere versjon** for å navigere deg tilbake til versjonen før den du ser på, eller versjonen etter den du ser på.

Rulle tilbake til en tidligere versjon

For en vanlig bruker er det ikke mulig å rulle tilbake en artikkel til en tidligere versjon. Kontakt administrator for å få til dette.

Sammenligne to artikkelversjoner

Inne på Historikkoversikten for en artikkel vil du se følgende felt:

Sammenlign sideversjoner

3 ▾

Aktuell ▾

Sammenlign

Velg artikkelversjonene du vil sammenligne og trykk **Sammenlign**.

Et bilde som ligner på det under vil komme opp.

Sammenlign side *Bruke historikk* mellom revisjonene 3 og *Aktuell*

« [Tilbake til sidehistorikk](#) - Color key: added text / removed text.

```
1 == Bruke historikk ==
2
3 For å bruke historikken kan du klikke på Historikk-linken oppe til høyre på en side.
4 Med denne kan du:
5
6 * Sammenligne den aktuelle versjonen med en tidligere versjon
7 * "Rulle tilbake" den nåværende versjonen til en tidligere versjon
8 * Se hvordan en artikkel har utviklet seg fra den ble opprettet fram til nå
9 * Se hvem som har gjort hvilke endringer
10 * Endring
```

Merk at tekst som har blitt lagt til i *Aktuell* vil være merket med grønt. Tekst som har blitt fjernet i *Aktuell* vil være merket med gult.

Diskusjon

På alle artikler finnes det et diskusjonsområde. Knappen finnes øverst til høyre på siden, ved siden av **Historikk**-knappen.

[Diskuter \(3\)](#) [Rediger denne siden](#) [Historikk](#)

Meningen med diskusjonsområdet er å kunne snakke sammen om

Diskuter: Hovedside

Diskusjon for **Hovedside**.

06/10/2009 11:24 av [sturlala](#)

s

diskusjon?

06/10/2009 11:26 av [atledy](#)

Re: s

kjekt med en heftig diskusjon

11/03/2010 12:46 av [tormodsv](#)

halla

hei

