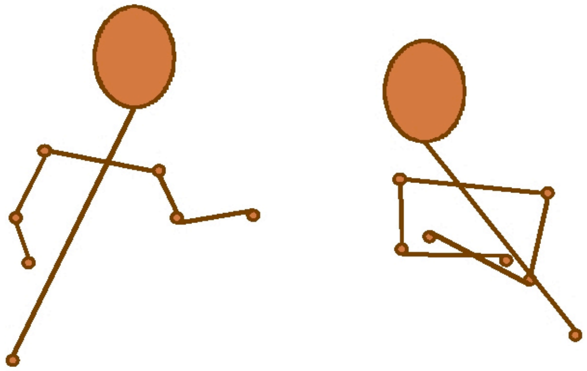


June 2010

Lessons Learned from the Rakontu Project

by Cynthia F. Kurtz

Rakontu is a free and open source web application small groups can use to share and work with their stories. This report explains the motivations behind the Rakontu project, results from building and using the software in 2009, and what I learned by doing the project. More information on Rakontu can be found at http://www.rakontu.org.

Table of Contents

Part One: The Story of Rakontu

1999 - The Idea

After previous careers in ethology (specializing in the evolution of social behavior) and educational software design (specializing in helping people learn about the environment), I took a job in 1999 as a temporary technical writer at IBM Research. This job choice was serendipitous and had more to do with location (my husband was also working at IBM Research) than subject matter. The offer was to write for John C. Thomas in the Knowledge Socialization research group.

At that time, although I had always loved telling and reading stories, I had never *worked* with stories before. I soon discovered that my background and interests were an excellent match for the field of organizational narrative. It is to John's credit that he let me run with this and do some real research in the field, no matter what my official job title said. I spent the next two years researching ways to help people and organizations gain benefit from telling and listening to stories, and embarked on a new career that has outlasted any other phase of my professional life.

The idea for Rakontu came after only a few months of exploration into the field. Helping people share stories seemed a glaringly unmet need in the world of social software, and I had some ideas on how to fill it. My first presentation in this area, in the summer of 1999, called "Story-colored glasses" (after which I named my blog), contained the seeds of the idea that eventually became Rakontu. Here I'd like to pull out only a few of the ideas from that presentation, those that I think pertain to why I've wanted to build Rakontu so much ever since.

First, as a former ethologist, I thought about the "natural habitat" of stories and how to foster their growth. I said:

> *Let's take it for granted that stories are an important mode of informal communication in organizations. Using biological terms, then, stories are important to the health of the organization as an ecosystem. Now let's assume that stories are becoming endangered in an organization. Say you wanted to create a story preserve, where stories could flourish and recover in a supportive habitat. What sort of habitat would you build?*
>
> *I suggest that stories are most likely to survive and prosper only where people are found. And not just one person—many people, in groups large enough to interact in interesting ways but small enough to share a common culture and history. The stories would also need a certain type of interaction between the people: they would need for the people to interact often (so stories can be passed on before they are forgotten) and over a long period of time (so stories can live a long life).*

(I've written much more about the analogues between narrative and biological ecosystems in other places since then: check *Working with Stories* for more on this idea.)

Then I talked about what I thought was the best way to support storytelling in small groups.

> *I speculate that a virtual community that supports informal information exchange through storytelling should have these qualities.*
>
> 1. *It should connect people* ***frequently*** *and* ***persistently****.*

2. *It should enable the creation of* ***traditions*** *for social contact in (virtual) space and at specific times.*

3. *It should create populated libraries, not empty ones—people and information should be* ***mixed****, not separated.*

4. *It should allow stories to be not only told but also* ***created*** *by the actions of people in the virtual community.*

5. *It should provide a* **place**, a ***time****, a* ***reason****, and an* ***opportunity*** *for telling stories.*

Finally I talked about how existing social software didn't meet the needs of a storytelling support structure. In what was probably my favorite part of the presentation, I contrasted the image of a person facing a computer above which was a speech bubble saying, "Type in everything you know so we can fire you!" with another image: same person, same computer, but this time the speech bubble said, "Welcome to our group! Here's what *we* wished we'd known when *we* started." Essentially, I suggested moving the emphasis in knowledge exchange from ***take*** to ***give*** (and ***give back***). I said that the main goal of the "thought project" I was presenting was "to build a virtual community based on storytelling that helps people to leave behind informal knowledge for others to find—in an interesting and compelling way."

The purpose of the "thought project" in this presentation was to explore how best to support storytelling for mutual knowledge exchange (though of course it has many other functions). The reason I thought new tools were necessary was because storytelling lives in the space of tacit, unstructured, informal knowledge exchange—yet with its own ***ancient structure and function***—and this means it requires a different means of support than was available. It requires more emotion, more connection, and more emergence of collective meaning. The ***tools of storytelling***—metaphors, allegories, oblique references, fantasy, changing perspectives, shared symbologies, imagination—are useful in group formation, knowledge transfer, and the creation of behavioral rules and norms. These tools are seldom available to people using computer-mediated communication tools.

1999 to 2008 - The Need

Between 1999 and 2008, I worked in the field of organizational narrative, doing one project after another and reaching somewhere around fifty projects. Some projects were research oriented and involved software prototyping and/or method development. Others involved helping people collect stories and use them to discover trends, help people learn, and pursue many other goals. Through these projects, as I worked with many thousands of stories of personal experience, I began to feel a growing sense of discomfort.

Why did the project planners and funders (and me myself) get to read and learn from all of these wonderful stories, while the people who *told* them never got to see them? I had done some projects, earlier on, when more stories were shared, but somehow those got crowded out, partly because it was always a hard sell getting the project planners to want to spread the stories around. The fact is, more people were willing to pay me to work on projects where stories were collected and never let out again than projects where stories were helped to get to other people. That bothered me more and more as the years went by.

At some point I encountered a wonderful book called *Where There Is No Doctor*. I think I picked it up in a bookstore. The following statement in the introduction of the book created a deep connection with the discomfort I had been feeling about my work with stories.

> *This book was written for anyone who wants to do something about his or her own and other people's health. However, it has been widely used as a training and work manual for community health workers. For this reason, an introductory section has been added for the health worker, making clear that the health worker's first job is to share her knowledge and help educate people.*
>
> *Today in over-developed as well as under-developed countries, existing health care systems are in a state of crisis. Often, human needs are not being well met. There is too little fairness. Too much is in the hands of too few.*
>
> *Let us hope that through a more generous sharing of knowledge, and through learning to use what is best in both traditional and modern ways of healing, people everywhere will develop a kinder, more sensible approach to caring—for their own health, and for each other.*

This was an exact match for my feelings. I decided to do something about those feelings. First, I completed the first edition of the book *Working with Stories* in May of 2008 and released it for free on-line. That helped with the transfer of ***knowledge***. But I also felt that people needed ***tools*** they could use to share and learn together from their own stories.

2008 - The Manifesto

I heard about the Knight Foundation's News Challenge in the summer of 2008. This was, and is, one of the very few funding opportunities open to businesses that are not non-profits. (I did consider forming a non-profit to build the software, several times, but decided it would take too much of my time and energy away from actually doing the work.)

I had some gaps between paying work—not many, but a week here and there—and I decided to try for some funding to build software tools that would help communities share their stories. The first thing I did was ask several experienced colleagues to join me in a group that would seek the grant together. (The list, in no particular order, was John Caddell, Stéphane Dangel, Steve Barth, Rob Peagler, Shawn Callahan, Stephen Geiger, Terry Miller, Thaler Pekar, Kelvin Saldern, Dave Snowden, and Paul Fernhout. Later, Stephen Shimshock provided helpful feedback as well.)

A few of the members of this group spent time on the phone with me talking through what the software should do and how to get funding for it (thanks guys). Based on those conversations and looking back over my notes on the topic for the previous several years, I wrote a 100-page document that described in detail why I wanted to build this software and what I wanted it to do. This took about six weeks in the fall of 2008. (I also chose the name Rakontu around this time.)

We didn't get the grant, though we did get to the second level of scrutiny. But thinking through my vague ideas about what needed to be built was of great value anyway. After I finished writing the document, I knew what I wanted to build, why I wanted to build it, what it was and wasn't going to be, and what it could (and couldn't) help people do. I also had some good documentation of the ideas, which I split up into the white papers on the Rakontu site.

Early 2009 - The Mockup

The next thing that happened was good and bad. I lost a major consulting client, so a huge chunk of “free” time opened up. I had a bit of savings, so I decided to use the time to build the first version of Rakontu. I did this for three reasons: because I thought the world needed it; because I had wanted to build it for ten years; and because it might potentially, eventually, bring in some new business.

I started the work by thinking about what I would like the Rakontu interface to look like in an ideal world. The result was a visual mockup of a Rakontu interface, at least the main parts of it. I finished and released this in May of 2009. I got feedback from some colleagues and friends, and made improvements.

Mid 2009 - Building Version One

The next thing after the mockup was to start building actual working software. I knew I wouldn’t have time to build anything like the beautiful vision I had written about in the grant application or even the fancy mockup I had drawn out. I had to start small. So I decided to build an “ugly” version of Rakontu, one that would fit into the time I had available, and see what happened next. I chose only about a quarter of the functionality I would like Rakontu to have someday, the bare minimum, and prepared to build it.

As always one of the hardest hurdles to cross in building software is to choose a platform. This loomed next. My main choices came down to something on the web or something on the desktop. I had many discussions with my husband and colleagues about options. I read a lot of opinions online. I tried several brief “playing around” attempts in both environments. I was very familiar with desktop software development, and not so much with web development. But I wanted Rakontu to help people share stories across the internet, and the options for connecting desktop to desktop, though they were many, were all difficult to set up for computer novices. I was keen on keeping the bar low and helping as many people as possible use Rakontu.

As part of all this exploration, the Google App Engine (GAE) came up. This was a free service; anyone could get a free account; it sounded as if it would be fairly easy to administer; I could write in Python, which I know well; it used the django template libraries, which I had used before; and Google was behind it, so it was probably not going to fold immediately. I tried the “Hello world” application for the GAE and was very impressed. Essentially, I was able to get a database-driven, interactive, free web application set up within a few hours. This seemed to be the best choice: on the web, easy to use, easy to install, easy to administer, easy for me to get started: the smoothest curves and lowest barriers all around.

So I plunged into developing Rakontu on the Google App Engine. I worked steadily on the software all summer. Things didn't go as well as I had hoped (and there's lots about that in the lessons learned part further down), but I got it done. Rakontu 1.0 was complete.

Late 2009 - Using Version One

After much building and testing, I released the first public version of Rakontu in September of 2009. I filled up the Rakontu web site (http://www.rakontu.org) with information, including a video tour of the software and screen shots.

I advertised on the Rakontu site for beta groups to test the software. I also contacted everyone who had ever told me they were interested in Rakontu, to tell them it was ready to look at and to ask if they wanted to have a beta test group. In total there were five beta test groups, but only one of them had any content. Most of the people who said they wanted to use Rakontu ended up not having time, or for whatever reason they did not get a group of people together to use it. All of the beta-test managers gave me exceedingly valuable feedback and support, and I thank them heartily.

The site that proved to be the best test of Rakontu was the Mistake Bank Rakontu, managed by John Caddell. I first met John because he was the very first person to send me an email thanking me for writing *Working with Stories.* We corresponded, and he ended up writing a case study for the book. John calls himself a "business generalist," having worked in many fields related to the business of having businesses. At the time he was getting into using narrative to help people address problems such as with customer service and product development.

John's Mistake Bank ("learning from faux pas, slip-ups and decisions gone wrong") was and is an experimental social project he started to explore the issue of people learning from each other by sharing stories of mistakes, in business and in life. He started a Ning site for the Mistake Bank in—well, I'm not sure exactly, but maybe 2007 or 2008. When I was starting to build Rakontu I asked John what was lacking in using Ning for story exchange, and I incorporated his wish list into Rakontu's design.

John wanted to try Rakontu as an alternative/additional home for the Mistake Bank, so I set up one of the beta test groups for him, and he invited people from the Ning group to come over and try it. This turned out to be our only "active" Rakontu in the beta test period. I put "active" in quotes, because in fact it was mostly John and myself who did the posting. There were 31 people and 45 stories on the Mistake Bank Rakontu by the time we stopped using it in January of 2010.

Also during this time, I applied for two additional grants for Rakontu funding (no joy there), started my blog, and started working on the second edition of *Working with Stories*. In January of 2010, I finally got some new paying projects and my attention switched over to working on them. I had told the beta test groups that the beta period would be over in January (then I extended it to March), so when that time came I looked for another block of time to write up this summary, shut down the Rakontu beta sites, and end the development project.

One other positive development is that as of this writing, Stéphane Dangel has nearly completed a French translation of the enRakonttire Rakontu interface. All of the French translations are available with the source code, so the software is nearly ready to be used entirely in French. I put special effort into making Rakontu as easy to translate and as internationalized as possible, so it should be possible to expand it into more languages in the future.

The future of Rakontu

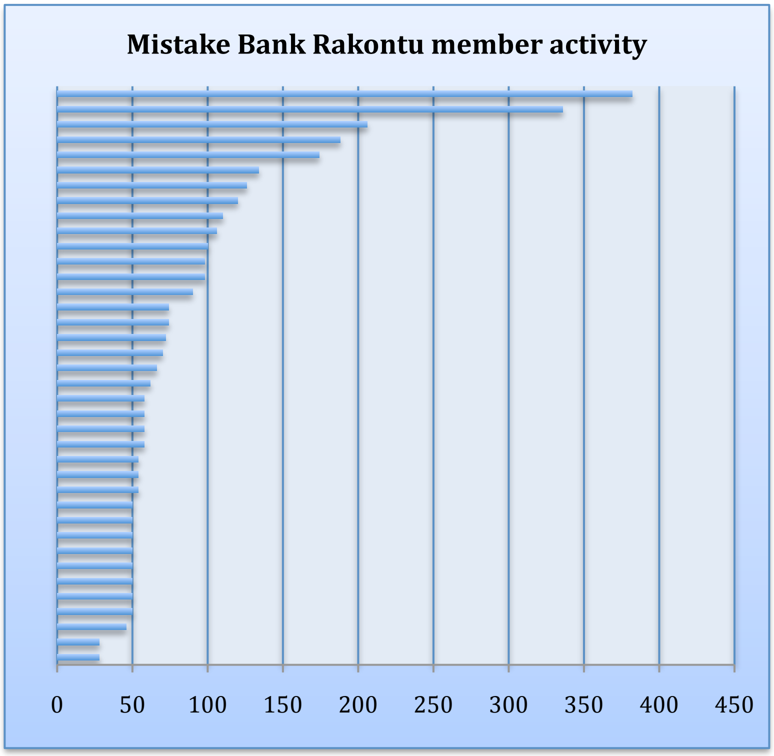
As of this writing, I like to say that Rakontu is "resting." The source code is available, and it is free to use on the Google App Engine. Anyone who can install and administer a Rakontu site is free to use it, though my familarity with it and ability to help with it are of course fading with time. I am not free to put more time into Rakontu at the moment, between paying work and trying to get the second edition of *Working with Stories* completed. I would love to come back to active development on Rakontu, but this will depend on getting more funding for the project. I welcome collaborators who have the technical skills to work on Rakontu, either in Google App Engine or to port it to another system and continue to work with it. That's the point of open source software, after all: Rakontu is not mine, it's ours.

I don't actually mind that much if Rakontu the software dies. The project was successful because I learned a lot and I was able to explore some interesting and useful issues related to helping people share stories over the internet. My hope at this point is that the ideas behind Rakontu, the ones I started thinking about in 1999, will work their way into things that get built for the internet, somewhere, somehow, through some somebodies, and have a positive impact on how people can tell each other stories. If that happens, I'm very happy.

Part Two: Patterns of Rakontu 1.0 use and reception

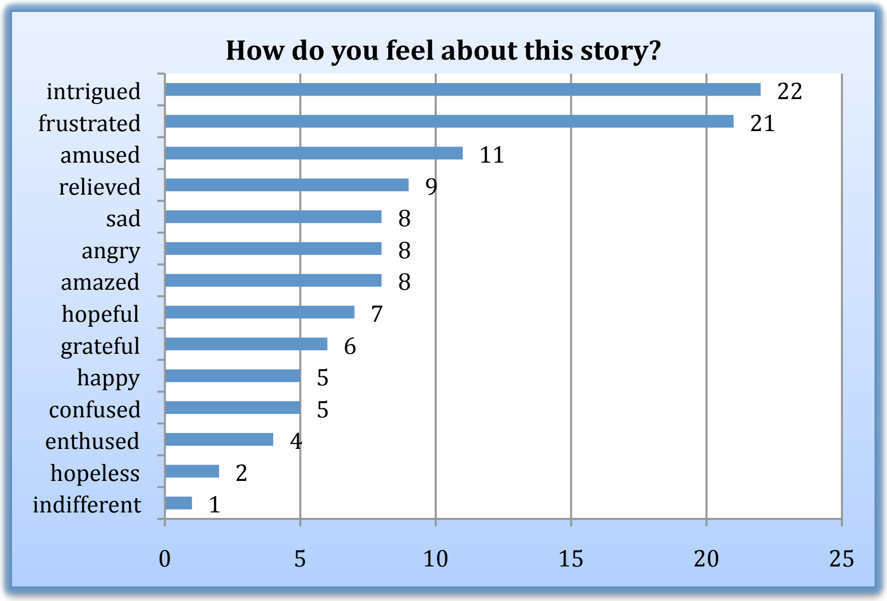
Patterns in the Mistake Bank Rakontu

A Rakontu keeps track of the things its members do by accumulating "nudge points" which people can use to rank stories as to their utility for various purposes. These nudge points can show us how active people were in the system.

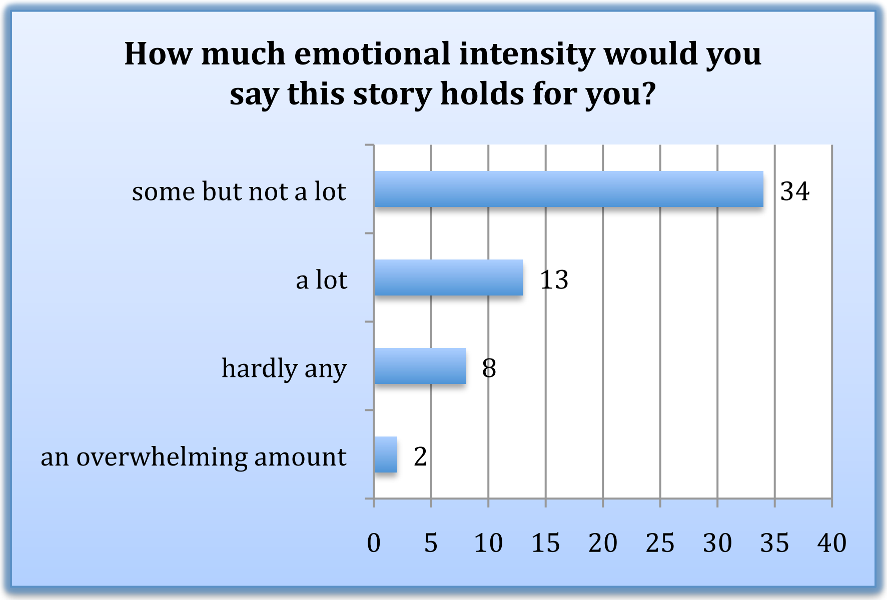


In the Mistake Bank Rakontu, if we look only at users who were not myself or John Caddell (the site managers), we can see that their activity levels vary pretty widely. The two highest accumulators of points were managers of two of the other beta-test groups.

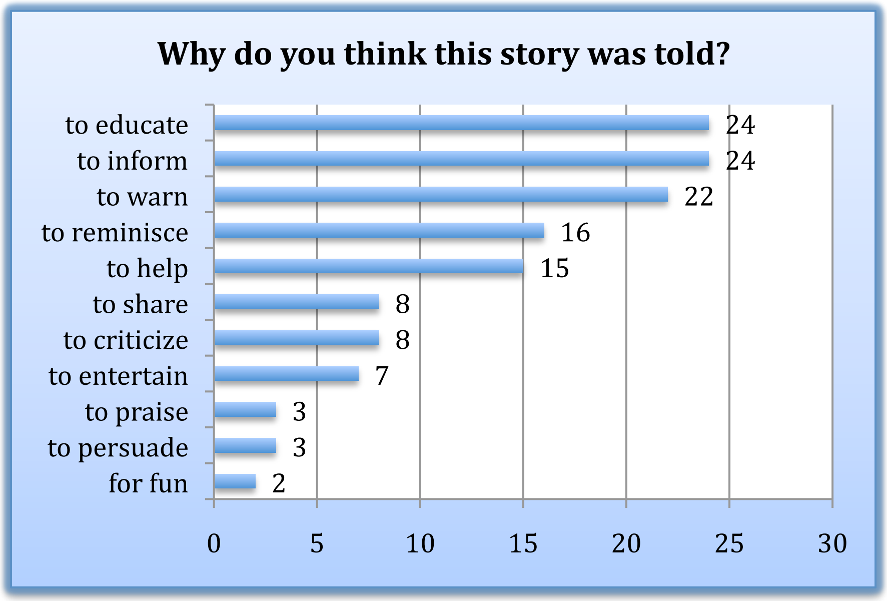
For the the 45 stories told, 62 sets of answers were collected to questions about them. (People can answer questions about stories other people told. Some of these were me, testing the system and showing people you could do this; but some were bona fide.) The answers to questions can show us a glimpse of the sorts of patterns eventual users of Rakontu could see.



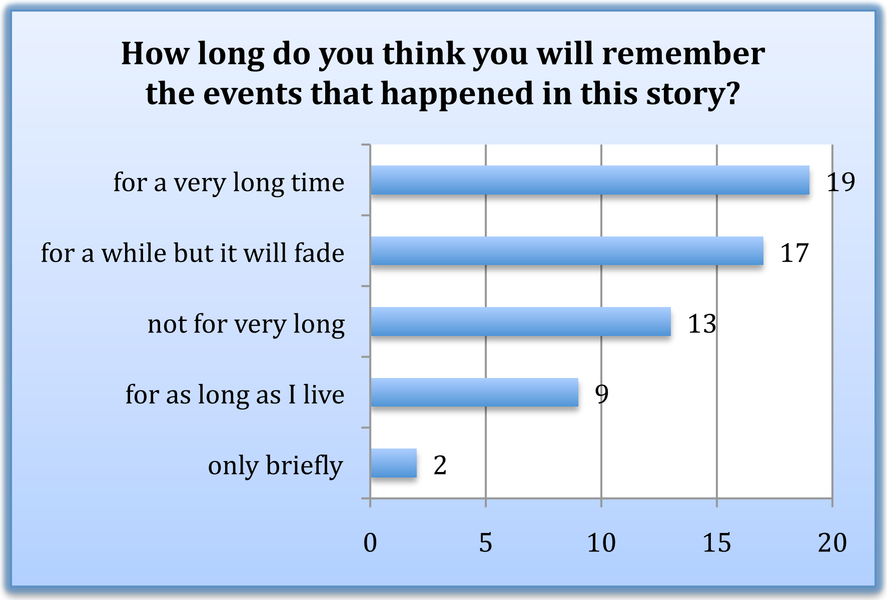
The most common emotions described were intrigue and frustration, followed by amusement and relief. These make sense if the stories are about mistakes: frustrating but interesting.



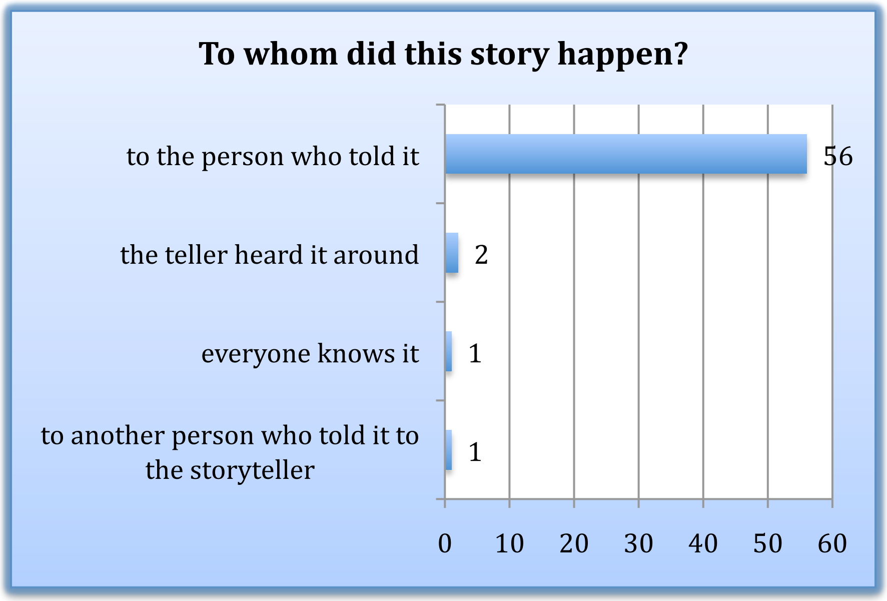
Emotional intensity was mostly mid-range, which is relatively high for a business context. I'd guess that mistake stories are likely to range higher in intensity than other stories.



The most often "why was this story told" answers were business-like: to educate and inform. That is typical for stories told in a work context or for a purpose. However, the high incidence of "to warn" stories indicates the nature of the stories being told, about mistakes.



Memorability was fairly high. This might be expected when people are talking about things they regret.



The stories were overwhelmingly in the first person, though some of that is historical in nature. The Ning Mistake Bank advised against talking about other people's mistakes, in order to avoid flame wars.

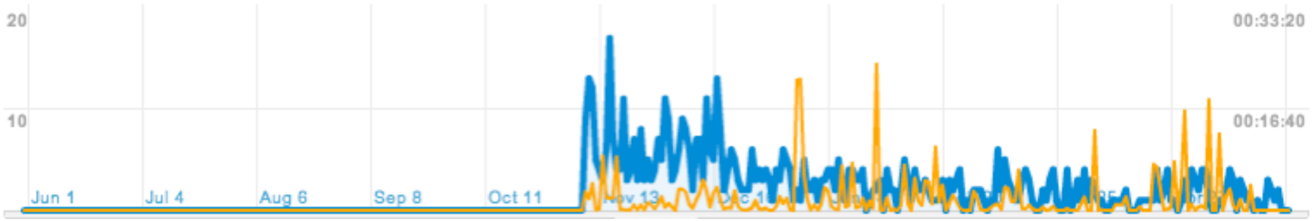
Taking this all togetther, if I was in a group looking at patterns in the stories we had collected and trying to use that information to benefit our goals, I'd conclude that:

- People tell about mistakes to warn others not to make them again. However, mistake stories do more than warn others. They may provide tips and ideas beyond a simple "don't do what I did" lesson. This may mean that we should make sure to label our mistake stories with more than just "what not to do" but also elements of contexts, things they learned, surprises, and other less central elements to the story.
- Mistakes are simultaneously upsetting and valuable. We should emphasize the valuable part of the equation!
- People remember their mistakes for a long time. This is a good thing, because even very old mistakes can still help others. We should encourage people to talk about mistakes from far back into their experience. Also, since stories about mistakes are memorable, it might be a good idea to ask people to recount stories of mistakes their parents and grandparents told them, because those might still be useful to pass on.

These patterns would help us improve our story sharing and collective sense-making. This is a little bit of the sort of revelation that I think would happen when people compiled their stories about a topic (or place or problem or idea), and that Rakontu could help them make sense of.

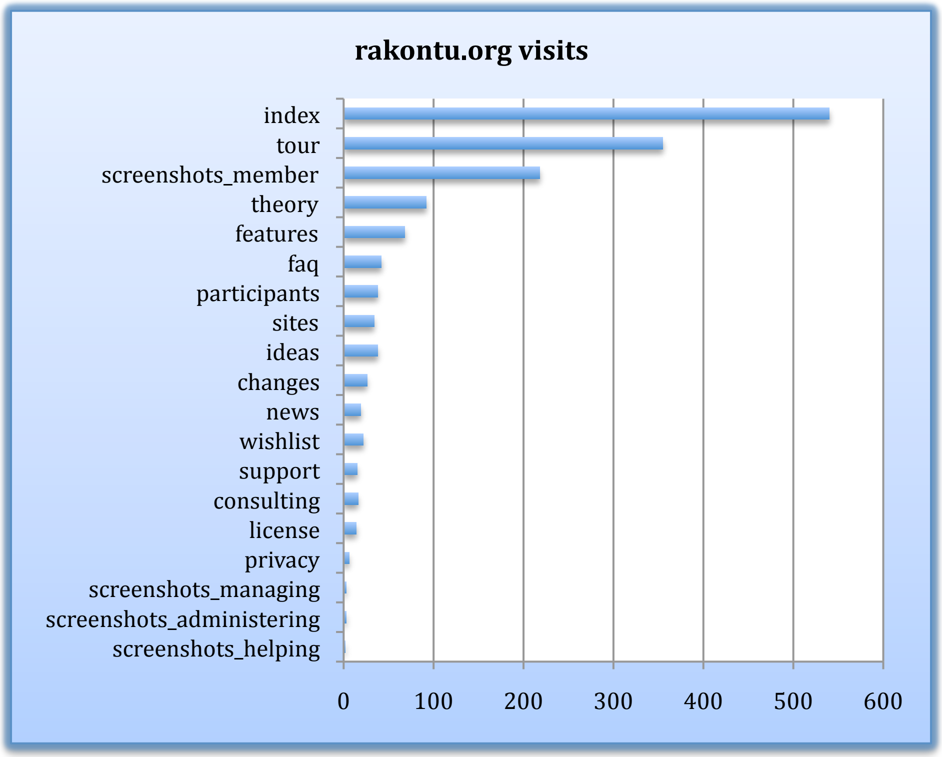
Patterns of rakontu.org visits

I connected rakontu.org to the Google Analytics system in November of 2009. Looking at visits to rakontu.org doesn't say anything about Rakontu as as system, but I think it says some interesting things about the uptake of new techological ideas.

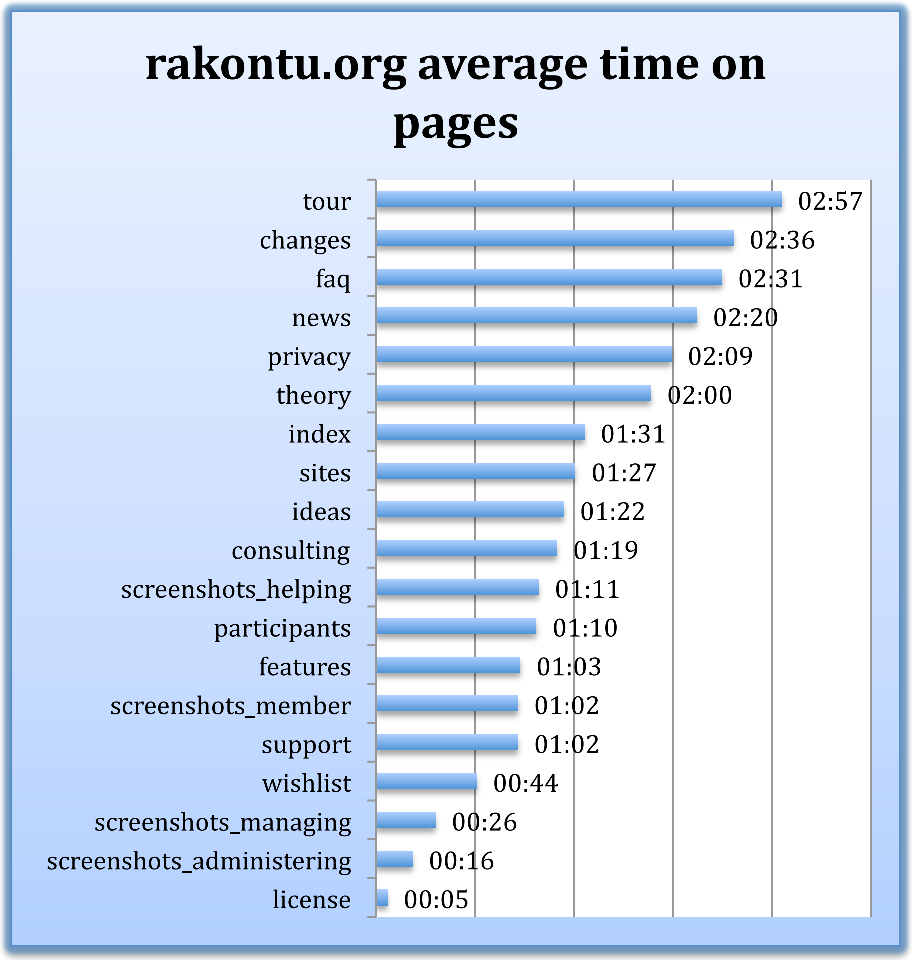


This graph shows the number of unique visitors (in blue) and the average time spent on the site (in orange). I'm not sure how accurate these statistics are, or how the system decides whether a visitor is unique and how long they stay on the site. But it's interesting that when I first set up the analytics service the number of visitors was relatively higher, and then gradually sloped down. The time on the site, however, started out uniformly small but peaked later with several longer averages.

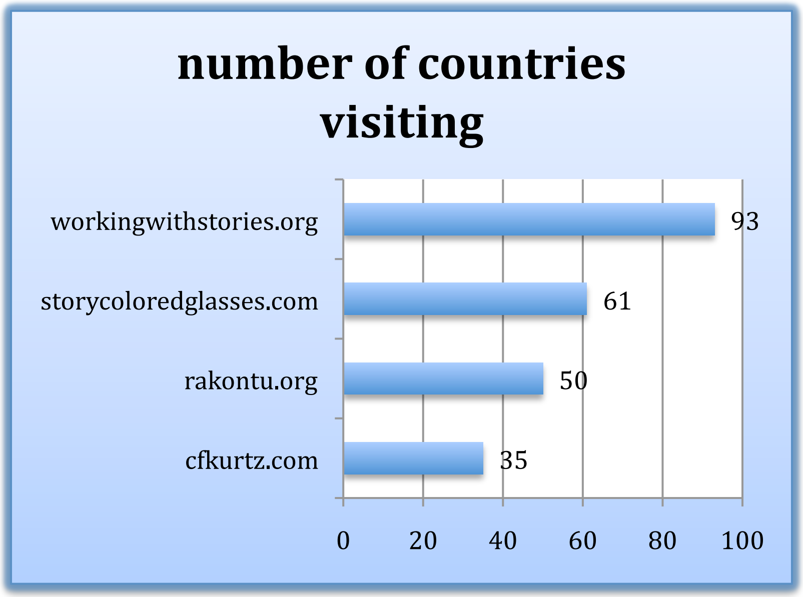
What this seems to say to me is that when the site first came out (and I listed it and mentioned it in a few places -- a pitiful marketing campaign, but not *nothing*) it attracted the attention of some random web-watchers, who skimmed and moved on. Later it was noticed by people who were more serious about the topic and wanted to explore the issues more fully (but not very many of them).



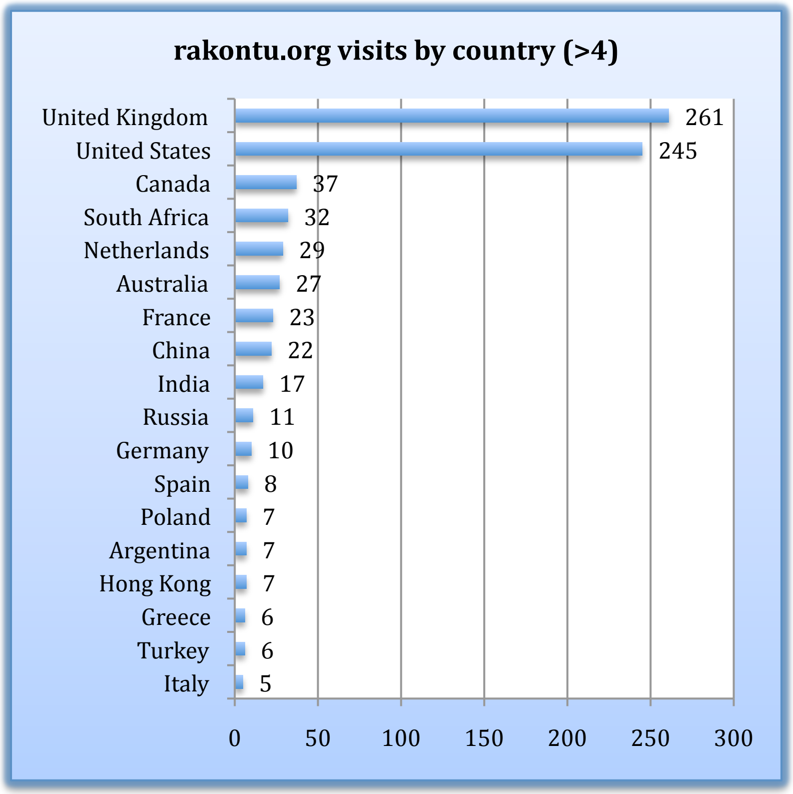
This graph shows how many unique visitors accessed each page on the rakontu.org web site. I was surprised how many people seemed to go directly to the video tour (tells you how important video has become on the web). Note also that many people accessed the screenshots for the member view, but very few of them also clicked on the other screenshot pages. The theory page (what Rakontu is for) was also often visited. I see that as a wonderful thing, because it means the ideas are getting out there.



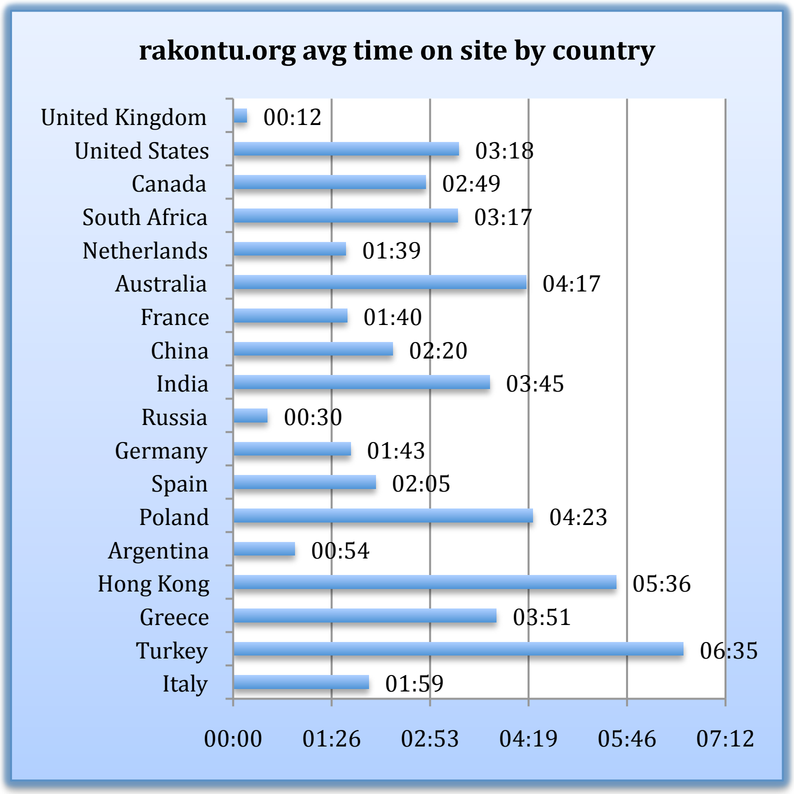
This graph shows how much time people spent on pages on rakontu.org. I'm not sure what Google Analytics does when people open up lots of web pages and leave them up, or walk away, so take this for what it's worth -- which is not that much. People spent the most time on the tour page, which means that a lot of people did watch the tour and didn't just see it and move away again. The second longest time is on the "changes" page, which I find a bit strange, because only a handful of people were actually *using* Rakontu. As far as I know. I think a few people have installed it to look at it, but nobody has told me that in person, so I'm not totally sure what's going on there. Also the FAQ and news pages were read longer (though less often). I'm wondering if this means that there have been more people using Rakontu than I think. Who knows.



There was also an interesting geographical dynamic to the rakontu.org site usage. I've got four web sites connected to Google Analytics, and rakontu.org falls pretty low in the number of countries represented in its readership. The *Working with Stories* web site is two years old now and is a textbook-type site, so it attracts a wide readership (it seems many people find it by typing in things like "collecting stories"). The blog (storycoloredglasses.com) is only a few months old, but it has outpaced rakontu.org, which went up a few months before that. And cfkurtz.com is simply a business-card and resume site, so it would not be expected to have broad coverage. What this says to me is that Rakontu is fairly undiscovered, as far as how many people have heard about it. I've put next to no time into "marketing" the software, and this pattern probably reflects that fact. If I was to work on Rakontu again, I'd put more time into getting the word out.

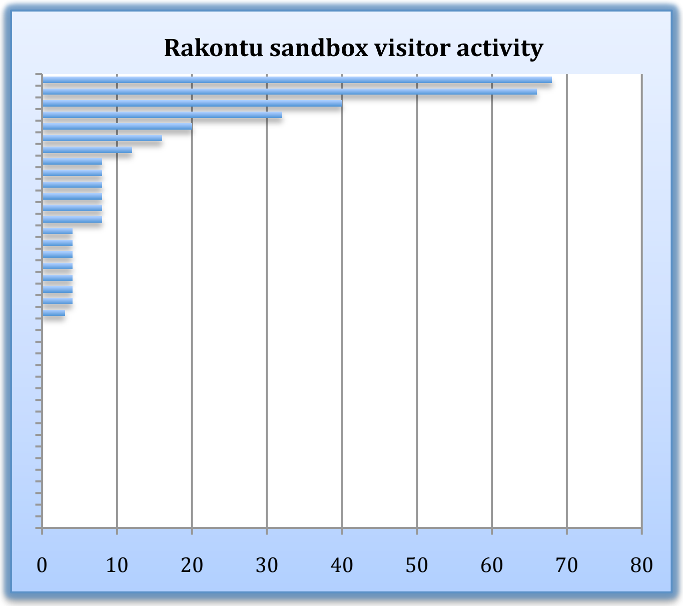


This graph shows how many people visited from the top countries. It looks from this graph like Rakontu is wildly popular in the UK. However,



from this graph it is clear that visits from the UK are very short, and are probably from some sort of automated system. Visits from pretty much everywhere else probably represent people actually reading what is on the site. The graphs for pages per visit and percentage new visits (not included here) show the same pattern. So the real picture is more like a bulls-eye around the US, with the usual Canada-UK-Australia-South Africa "Western World" conglomerate coming in second. What this says to me is that Rakontu is reaching people in the same way the internet is reaching people: mostly the technology-rich countries and not so much elsewhere. That is not my vision for Rakontu, but it is what is to be expected for people noticing new technological solutions.

Finally, let's look at people visiting the Rakontu sandbox, a fake Rakontu site I set up to show people what it's like.



Forty people have become members of the sandbox Rakontu in several months. The "nudge points" accumulated by sandbox members are as shown here. Note that almost half of those who joined (19) never clicked on anything and just left again. (I'm not surprised, I do that a lot too. Everybody does that on the web. It's a low-commitment world.) Of those who stayed, most read a story or two, and only a few did anything more than that.

One of the problems with showing people how a thing like Rakontu can work is that it doesn't work very well without a previously existing social group. How many stories people read in a fake community doesn't really tell you anything about how people would respond (and how the software would help) in a real community. So this is all kind of not very important. But it's interesting anyway, just in terms of reactions to a new product and idea.

Part Three: Lessons learned

This part of the report is me reflecting on what I learned by doing the project. It's personal, it's emotional and it's a bit whiny. But it's important. To make this lessons learned list, I simply thought about my time with Rakontu, paced, and talked out loud about all the surprises, excitements, disappointments and emotions I had encountered as part of it -- both mine and those of others. Then I clustered those elements into larger themes and expanded on each.

Note that this section relies on knowledge about what Rakontu is for and how it was built that you probably won't have if this is the first thing you are reading about it. For more information, look over the web site or the white papers (see the theory page on rakontu.org for all white papers).

The Rakontu Interface

What worked in the interface

I loved the roles (manager, curator, guide, liaison). I found them very useful to organize my thoughts about the things I wanted to do to support the site. Nobody *else* used them, but I liked them, myself. I think they work.

I loved the question asking system. It was just what I'd hoped it would be. You can easily ask questions, see the answers, change the questions, and so on. I wouldn't change much on that.

The tags were great. I used them much more than I had expected to. They became a shortcut for searching, to the point where I felt Rakontu needed to add more ways of navigating through metadata than the search filters I had set up. I felt something useful growing.

I liked the wiki-style formatting. It helped me to be more expressive in my stories.

The menus worked well. People seemed to understand them. Some things like that are so enmeshed into the culture that they come free, with no learning curve. I should have used more of those.

I had a ball playing with Rakontu, myself. I envisioned people loving it. I found administering and managing Rakontus to be pretty easy, for the most part. I could think of improvements but I didn't find it onerous. The people who managed the beta test sites asked questions, but they didn't seem to think managing a Rakontu was impossible.

What didn't work in the interface

I worked really hard on making the timeline interface move away from popularity or value-in-general assessments of stories. However, this totally did not work. I got sucked into thinking things near the top were "better" than things lower down, and so did everyone else. Saying the things at the top are more useful for some purpose *seems* like a good thing to do, but people have such a strong "up is good" bias that it's impossible to fight. It's possible that if I had been able to build the interface I mocked up, with the graphical "tails" on stories showing their histories, and as-you-click interaction, this might have not been such an issue. I'm not sure. It would require more testing.

Vertical-value bias aside, I found myself wanting to move off the timeline into other views after some number of stories had been contributed. I think I stopped caring about "what was happening" and just wanted to look at the "library" of stories. What this says to me is that you can look at story sharing as both a verb (an exchange) and a noun (a library). If you have a verb view -- the timeline -- you need a complementary noun view. I found myself wanting to take a stroll through our library. That may be the most significant finding from actually using Rakontu, that verb and noun want to be juxtaposed and complementary.

Several elements of the system, such as formatting your stories and creating search filters, required quite a lot of explaining, and hardly anybody used them. It became very clear that using Rakontu heavily would require spending quite a bit of time becoming familiar with it. The bar for usage was much higher than I had thought. If I was to go back and work on it again, I'd take a hard look at complexity and see what could be trimmed.

Social Aspects of Rakontu

What worked socially

I loved the exchange that got going between myself and John Caddell on the Mistake Bank Rakontu. For a few months we were trading a few stories a week. We were reminding each other of experiences, and making comments, and using the system in the way it should be used. In a way, the interaction between the two of us, over that short time, was the only real test of Rakontu. Rakontu is not meant to work for groups of people who don't know each other already. John and I knew each other, but I didn't know any of the other people in the Rakontu, and those exchanges were awkward. Not because it was anyone's fault, but because things like ratings and comments and views have a different meaning when you know people and when you don't.

In my writing I often reference Harrison White's model of human interaction where he distinguishes between selection activities (meeting people, making choices), mobilization activities (building coalitions, gathering converts) and commitment activities (working together as a team towards some goal). What I saw was that Rakontu worked really well for commitment activities -- John and myself talking together about what we've learned about mistakes. Rakontu didn't work well for mobilization or selection. This is a good thing, because I had meant Rakontu to work for commitment. I chose this quote in my first writings about it:

> *Never doubt that a small group of thoughtful, committed citizens can change the world. Indeed, it is the only thing that ever has. -- Margaret Mead*

So that's a positive thing. I felt, in this tiny little test, that Rakontu did help a small group of thoughtful, committed citizens do something together, even if it was only a tiny thing. That gives me hope that it can keep doing that, eventually, on a larger scale -- or that its *ideas* can help with that sort of thing.

A surprising thing with relation to people and Rakontu -- I'm not sure whether to put this into what worked or what didn't -- is that something like five or six people asked me if they could use Rakontu to collect some stories for short-term story projects. When I explained that it was meant to be used as part of an ongoing relationship and wouldn't work well for in-and-out story collection, they were disappointed. I suppose this is not very surprising, because my discomfort with the surgical nature of most story projects was one of the reasons I created Rakontu in the first place. It did make me wonder, however, whether I have been too purist about what should be created. For example, if Rakontu had the ability to do either short-term or long-term story collection, would the ideas spread more widely, and would it get more funding? Maybe. It's something to think about.

What didn't work socially

One of the most cringe-worthy moments while using Rakontu was -- and I thank this person heartily for participating, and mean no disrespect -- somebody sent me an email response saying something like "my story got three hits!" Sigh. So, in terms of people and social matters, the performance culture of today's storytelling, where people think they are on TV every time they open their mouths, was very much present even in the tiny attempts I made to try out story sharing. I wanted people to get away from the popularity rating that I belief precludes true sharing, especially of things as personal and emotional as stories, but I couldn't do it. You can't change cultural trends with software. I think any new work on Rakontu will have to confront that fact head-on and think of new ways to work through it.

The other thing that didn't work about people was that I have forgotten how little most people know about stories. There are several hurdles to getting people to understand why we are telling each other stories, what makes a story good or useful, what is a story, what isn't a story, what you do when you read a story, what you don't do, and why any of this matters. I tried to write succinct, helpful pieces for people just joining a Rakontu, but compared to typical social media applications, there is a much larger learning curve involved in storytelling.

One hope for getting people over this curve is that in a committed group, the people who are putting the energy into the project might be willing to explain the concepts in person and watch over things so that early misconceptions get put right. But that's a lot to ask. I was surprised how little the managers of the beta-test groups changed in the settings for their Rakontus. I set up Rakontu to be massively customizable. You can change what questions get asked, how nudge points are assigned, what it looks like, what you tell people about why they are there, and many other things. However, I found that all the managers basically left everything the way I had set it by default. This is nothing against them, but it did make me realize that the hump extends all the way up to the people who are in charge of the thing.

I also think I had not realized how overwhelmed people are by all the software they have to contend with today. When I first started writing software, a typical user would have to make sense of only a handful of programs. Back then, it was not hard to find people eager to plumb the depths of your new software. But today, not only do people have dozens of programs on their computers, every major web site is its own program! No wonder people are worn out and don't want to learn anything. I realized this when somebody asked me if I wanted to start using Diigo to share web links. I wanted to, and installed it, but when I dropped down the Diigo menu for first time, I got this overwhelming feeling of *too much*, and I couldn't do it. I've come to realize that today, it doesn't matter if your software is good. There is just so much software out there, good or otherwise, that people need a *very* compelling reason to try it and learn it, before they can commit the time to it. It's a different world than it was.

I think this overload of web applications meshed with the "ugly" nature of Rakontu, in this way. I think people use the interactivity level of a web site -- its bells and whistles -- as a way of evaluating whether it is worth spending time learning. I do this myself. If a web site looks like somebody wrote it in 1995, I often move on. I'm guessing that when people looked at Rakontu and didn't see interactive graphics, they said "probably not that great" and moved on. I think the expectations for web software have increased so much that -- I should have seen this -- in the time I had available, by myself, I just had no hope of being able to make something as interactive and up-to-date as people have come to expect on the web today. That doesn't mean I think people are unable to see quality without adornment, it just means people are so busy they have to be very choosy. I was facing a bar much higher than I had imagined, and I should have backed off and chosen another path.

Techological Aspects of Rakontu

What worked technologically

The choice of the Google App Engine didn't work out very well, as I will explain in the next section. Still, Rakontu works, and is mostly reliable, and people can use it. It can even be ported out of the GAE environment with some effort.

There *were* some good things about the Google App Engine. The user management aspect was very useful. It handled passwords and security well. I didn't think people felt their stories were vulnerable. I was able to get up and running very quickly. If the "gotchas" in GAE were worked out and its limitations were reduced, it could still be a viable platform for real web apps.

What didn't work technologically

At first things went very well with the Google App Engine ... but as with any architecture, things are not always as simple as they seem in the "Hello world" version. I spent weeks struggling with some of the aspects of the GAE that were not ready for prime time. Just a few examples:

- There are some limitations in GAE because it's a free service, and the documentation and introductions don't highlight these things. For example, because of some complicated thing about distributing server requests that I can't remember right now, Google recommends embedding every single get or put operation into an operation that retries several times, sort of to get the server's attention, before telling the user that there is a problem. By the time I found out about this little "gotcha" I had written hundreds of get and put operations in the Rakontu code. It would take forever to carefully go back and rewrite each one of them, and I ran out of time soon after I found this out. So, randomly, not often but enough to be irritating, users get error messages that they would not have seen if the GAE documentation had been more forthcoming about its flaws.
- GAE at the time had no reliable backup solution, and I felt this was an essential component of a story sharing system. A group needed to have a way to copy their stories off the Google site and keep them locally. I probably spent 80 or more hours trying three different approaches to site backup, finally settling on my own system that works fine, but requires more administrator technical ability than I'd hope to require. (I think GAE has recently improved their backup systems, so the time I put into this may not be required now.)
- I had hoped to support audio and video storytelling on Rakontu sites, but GAE's support for large binary objects lagged far behind my need.
- After using Rakontu for a while, I found that it needed email updating (telling people there was a new story to read, etc), but every time I read the GAE forums there were many messages about the "task queue" system having bugs and problems. At some point it just didn't seem worth trying it.

There were many other similar problems having to do with either fighting the limitations of the GAE system or dealing with the fact that it is still in development and still has many kinks to work out. That's fine if you have unlimited time to work, but I had only a little time, and wasting any of it on this sort of struggle was, well, heartbreaking. Eventually I got tired of fighting with it, and I got tired of finding out disappointing gaps and problems that only became apparent after it was too late. The support from Google was not great, either; a few times I spent weeks waiting for answers to questions.

I don't think the Google App Engine is a bad system; I just think it's not grown up enough to handle what I had wanted to do on it. If I had known that at the start I would have chosen another platform.

Another technological problem was that I spent a lot of my time reinventing the wheels of social media. I had no interest in supporting profiles and messages and notifications -- but those are the bones of social media, and I couldn't get away from them. At this point with so many social media platforms and systems out there, I wish I had chosen something that had more of the simple things built in. Even writing Rakontu as an add-on to an existing social media system might have been better. I thought of this but was stopped by wanting to keep it open source. That took out Ning and Facebook. I considered some open source platforms, but as I recall, none of them seemed compatible enough with my goals. (Hard to remember at this point.) Anyway, if I was to do this again I'd look harder for a way to get more of the basics done for me.

The thin client also did not work well for storytelling. The bandwidth of written text alone was too small for the complexities of story sharing. Audio and video were there in theory, but poorly supported in practice and hard to mix in elegantly. I felt that I kept hitting up against the walls of the thin client. For example, I was always having to rewrite sections of code because the requests took too long to come back from the database, and GAE would hit a limitation, and the user would be left hanging. On the desktop these sorts of things just don't happen. It felt a lot like the old days, like rewriting your code to fit into 8K of memory, or spending all day deleting files so you could fit into your expensive 1MB hard drive. I would not want to choose such restrictions again if I can help it.

Building Rakontu

What worked in building Rakontu

I built Rakontu for two reasons. I wanted to experiment with real working software in order to find out what would best support people sharing stories over the internet. I did that, if only in a small way. And I wanted to help other people think about how best to support people sharing stories over the internet. My hope is that the documents that have come out of the Rakontu project, if not the Rakontu software itself, will have a positive impact on the way people help other people share stories over the internet. So in that sense the project was a rousing success.

I think the *most* important thing that worked was that I got a chance to try out many of my ideas about story sharing in real software. I learned a *lot* building and using Rakontu. If I was to write Rakontu again, it would be vastly improved over what I could have done before because of all the things I have learned. My rule with building software, or writing books, holds for Rakontu: having done it, I'm ready to do it.

The other thing that worked was that I built something that works. I like Rakontu. I like using it. I would still be using it if I had a group that was using it. If I had written it for the desktop I would probably be using it for my own stories. (But watching over the GAE sites has taken too much time away from other things, so I can't really get back into that.) If I had more time to work on Rakontu, I would lean fairly heavily toward re-implementing it on the desktop and setting up some peer-to-peer or client-server capabilities on the back end, just to have more control over quality and reliability.

What didn't work in building Rakontu

As I may have said already, the choice of the Google App Engine didn't work out well. The thin client didn't work out well. I wish I had built it on the desktop. I chose the web mainly because I wanted a low bar to use. But I didn't end up with a low bar to use anyway. If I had accepted the inevitability of a high bar to use, I could have made it more useful to the few people who could scale the bar. If I had worked on the desktop, I could have used my time to build a much better interface than I did, because I wouldn't have spent so much time fighting the web environment and the GAE. Java and Python may be boring, but I know how to use them, and I've built lots of desktop interfaces. My ambition to “reach everyone” was too big. I wanted to reach people who could barely use the web and couldn't install anything on their computers. If I had started with the people who could install and maintain desktop software, I could have grown the user base later.

(Of course if I had built the desktop version first, this whole document might have been about how *that* didn't work out well. Hindsight is 20/20 vision.)

I also wonder if I should have concentrated so much on building a working system. I only had a little time to work with, and I wanted to build something people could really use, but I probably still should have held back and written more of a proof of concept rather than a working system. In a working system there is *so much* more work to be done, on the help system, on testing, on talking to beta testers, and on and on forever. Prototyping frees you up to explore. So I wish I had prototyped more.

The other thing I learned, even though I had lovely support from some people (and many thanks to them, because they know who they are), I didn't have as much support as I needed. Bugs are disheartening. Disinterest is disheartening. Rejection is disheartening. It’s hard to keep up a project by yourself. You need moral support, and a lot of it, to keep going.

Still, I'm *very* glad I had the opportunity to do the project, and I'm glad I did it, and I learned a lot, and I hope to return to it, or to something similar, someday.