20040302MarcLaporteInterview

Interview with Marc Laporte (Tiki CMS/Groupware)

Language: English

March 2, 2004, 1700h CET, interviewer: Stefan Haefliger

Link to the audio file (MP3)

Below is the original transcript. It has be corrected and hyperlinks have been added.

S: [intro, presentation of project, research goals, interview format] could you start with telling something about how you got involved in this project?

M: i've been in websites for several years as my job, i was doing classical websites with software installed on your computer, loading the files, and i was doing some websites for non-profit organizations for which i was volunteer, and often in those organizations i ended up having to do all the updates myself. and i felt it was becoming time consuming and it wasn't something that was going to get better. if people are using the website a lot i was going to have to spend more time. i wanted something where i could, once installed, people could take ownership of the website and of the project and contribute themselves, and have me out of the loop. i tried out different software, looking what was available, found some cms, phpnuke, postnuke, so i did some pretty thorough analysis of what was available and at that time i came to the conclusion that postnuke was the best alternative for what i was looking for: portal type software with lots of different features, add-ons, forums, and all kinds of things. so i started with that, and i worked for that almost two years and it was pretty good, one point the development was doing pretty well, somethings happened within the postnuke community, they didn't agree with things, it wasn't going as well anymore, also around that time i started to get interested in the wiki concept, forums are nice for certain things, but they are not good for building content. you have to go over the whole forum to see what the conclusions are. so i found wikis and thought that's interesting to build knowledge together, so i installed that and played with it. i tried PhpWiki, and a few wikis, and that's how i found tikiwiki. it was just starting out, the team was very enthusiastic and open about my suggestions, and they did many, and the ones they didn't do they said we keep them on a to do list, i really liked the team, that's what got me interested. at that time tikiwiki wasn't as good as others, e.g. twiki.org in perl – the most mature project, but i prefered in php. so since the tikiwiki project seemed to go in the right direction and figured that would be my wiki and i would combine that with postnuke for the portal stuff. also i was using Tutos for the groupware features. that's how it started out. and tiki was adding more and more features, to the point where i figured tiki could not be my wiki but also for my portal needs. at that time the postnuke situations was getting complicated, there was some forks, Xaraya, they seemed like having a nice plan but they said they would only release something stable in 6 months, and i didn't really want to wait, so i tried with these guys. that's how i got involved in the beginning, there were Luis and Garland running it, and at one point Garland faded away and wasn't so available and even luis wasn't available and lots of people wanted to contribute some patches so i sending mails to luis saying this guy is suggesting this patch what do you think, and luis was overwhelmed and asked me to help him out. i said sure, why not, that's my first experience in an open source project from an administrator perspective, and i figured sure why not.

S: you're basically involved in everything right now as an administrator?
M: yes and no. i'm not a programmer, i just hack around and i'm able to find bugs and correct bugs, i'm not a programmer per se, i'm an IT manager. that's why i really want to focus on getting more developers, that's something i don't do and typically in os projects there's not much marketing done and i decided we had a really good project and needed to get it more known. if people would know about it they would join and make the project better. so i focused on getting more and more programmers on board, normally in os it's release early, release often, what i try to implement is recruit early, recruit often. anytime somebody comes up with a patch i say, why don't you work with us in the CVS instead of sending patches. people

were sending patches but weren't getting any feedback, it was just laying there. i felt like they should be a closer part of the team.
[10:55]

S: can you talk a little bit about your experience in the project, when did you observe code to have been imported, and what type of code is being imported and what have you taken from other projects?
M: i think that depends on where you come from. as i'm not a programmer and it's not something i can do easily, my first reaction is to get, well somebody has done it before, how can we get somebody else on the team to work on it or use their experience, but when for programmers sometimes they feel it's easier to do it themselves, if they like programming, that's often what happens, the programmer's first reflex is how could i write it, and it's probably easier and simpler for them, and for me it's, why don't we get a partner to collaborate. most of the time when we want new features, and everybody agreed on, we want tiki to be able to be used by different people, different websites, the long debates if we should include a third party or write our own, and sometimes i was in favor of using another php forum, but the programmers explained to me, look it's too integrated, too complicated, it's better to write on our own. and try to link the wiki syntax. and i just trust their judgment, and say ok. we looked at that option and it looked simpler, easier, better in the long term to do it that way.

S: do you have an example for that?
M: there was cases where it was better to have a third party program, ok, for example JGraphpad, which is a java applet to have graphics. we wouldn't necessarily have the expertise on the team to do that, or the time or the interest, but that was a perfect match between what they were doing and what we were doing. actually, they were even looking for a wiki type project to integrate to. so that worked great.

S: what type of format does this have: is this LOC, a project or a library?
M: it's another open source project, in java, they have an applet version and a desktop version, the data is saved in xml, and that's how it's possible for the information to be exchanged between the two programs. another good success story is with the HAWHAW library, it's a toolkit to make your webpages available to cell phones, via WAP, that's something i was very interested in having, when i was going around the web i didn't really see anything, software, which was doing this at a reasonable level. so i started looking around and i hunted this one down quite a bit, most of the stuff is on sourceforge or freshmeat, but this one wasn't anywhere, and when i found it i said, wow, this is exactly what we need: basically, it's a toolkit but by itself you can't do anything with it, you have to be able to program to integrate it in your applications. so it's not something that some poweruser can download and install. so i contacted the developer, we exchanged emails and finally he joined the team and plugged in his toolkit, into tiki, and now since then, the last two months, tiki has been cell phone accessible. so that's a perfect match between what he does and what we do, and what's nice is that someone now goes to HAWHAW and say i want to have a cell phone accessible website. well they can just download tiki and use it.

S: is that a project of itself? and how is it spelled?
M: h-a-w-h-a-w. for all the partners you can look at: http://tiki.org/Tikipartner

S: what is about the time frame that you talked about before, when you joined, and this just happened a few months ago. were you talking about the last two years? just to have an idea.
M: the project started in october 2002, it's a very young project. there were lots of releases in the first few weeks: 8 releases in the first 10 weeks, something that impressed me a lot. then i started getting very involved somewhere in may 2003, when i started to be very active, almost full time. somewhere in june, mose also became very active, and there was al brown, and we started working on the 1.7 version, at that time luis was becoming less available.

S: the examples you gave me were basically whole projects, that you got to partner, integrate and work together. are there examples of smaller pieces, modules or LOC?
M: yes. the basic foundations of tiki, we're using php, that's our programming language, the main thing, for the template engine we're using smarty, that allows tiki to have any "look & feel" you want.

S: what does that mean? what is a template engine?
M: smarty template engine.

S: what does a template engine do?
M: it separates the business / application logic from the presentation. it permits you to have graphic artists which working on the "look & feel" of the sites, working on the templates and not touching any of the programming code, so they can't mess anything up. they can only play with the "look & feel". on the other side, the programmers can concentrate on the php files where they can work out the business logic.

S: smarty, what type is it and how is it being integrated?
M: i think it's the most popular and biggest template engine in php. it's pretty big.

S: you're basically reusing it and it's used by other projects and comes from the php distribution.
M: mhm. it's used by a lot of projects. this is something else that helps. we have quick development, we've been using very good components. the other main component is the database layer. we used to use pear:DB, and now we use ADOdb, version 1.8. it permits us to use any database, so we can use mysql, sqllite, oracle, basically any database, by using ADOdb it's an excellent component that we're using, so we're just benefiting from their experience and test and research.

S: what was your reason to change?
M: we wanted really to have database abstraction, it was there in theory in tiki 1.7, but it wasn't really tested and not all the queries, and wasn't done everywhere in tiki thoroughly and properly so it worked on different databases. mose and Florian (redflo) from Germany, decided to pick this up and make sure we have database abstraction, and did all kinds of research and tests on what abstraction layer were the best, there were about 5 available, and the best ones seemed to be pear:DB and ADOdb, and after lots of mailing list discussions ADOdb was picked, then flo and mose spend several weeks rewriting all the queries in tiki to make them work with ADOdb, and that's why we have database abstraction today. and that's also used by lots of different applications, postnuke also e.g. uses ADOdb.

[21:48]

S: can you talk about how you learn from other projects before you program, if you go to see how other projects they do it?
M: i personally do that a lot. in the case where the other project has the same license as we do, what's really great is that we're allowed to copy their code as they're allowed to copy ours. and what's really interesting is not the project with the exact same license in terms of user interface, and what features are available. so that was very interesting. so typically, that's the role i have been playing, a bit of a product manager role. for example calendar. i think it would be a good feature for us to have: a good work group calendar. so i have been looking at dozens of os applications and listing all the features that they have and making one master to do list: this would be the perfect calendar. In terms of discussions with other projects, I think there are way too many open source content management systems. There are hundreds of open source CMSs in php. I think that's a waste of talent and energy, but that's how it is, what can you do? But hopefully, instead of starting their own little project, people will join larger projects and we'll have more high quality projects. What I try to do is to have discussions and exchanges with other projects. e.g. the Horde project, which is an excellent email and groupware project, they're working on syncML, so we could synchronize information with cell phones, e.g. your address book. They're doing some work there and i hope we could collaborate with them and maybe some other projects, so we could all work together on certain projects. I don't think some day some projects will say "i'm closing my door and not going to do any development anymore, I'm going to merge my project with another project". I don't think that's very realistic, but to have different projects working together on common tools, that's very possible. For example the syncML project, that's something everybody would need and the users would like it if they were able to exchange data from one place to another.

S: (some talk about OSCOM, and the people from Zurich)

M: There was a conference in boston, I was there actually, I think it's an interesting initiative, I'm in the oscom irc chatroom (#oscom on freenode.net) and there are not many people. I think it's a good idea but so far it doesn't seem to be working enough, it should be the place where the main Open Source CMSs debate and collaborate. So far it's not happening, i wish it would. I even have a list of different things on which I think Open Source projects should collaborate. Ex.: syncML, we should work on having more... calendar data sharing e.g. there is no really good standard. If I have a postnuke site with calendar information on my site, I should be able to share that information with somebody who has a Geeklog powered site. and it should be possible for us to have rss feeds, we have that for articles, we should have this for calendars, that's something we should work together. also to have a WYSIWYG editing applet, with spell-checking in every language, that's something everybody needs, and there are some ok solutions but no really good one right now. Then, spreadsheet applets, basically a mini java applet to do tables and spreadsheets. That's something that doesn't exist to my knowledge, I have done some research and lots of open source projects could use. Oscom has to be more a place to debate these types of ideas.

[27:59]

S: when you compare calendars, is it on the level of features that you do that, or on the level of the code: how did they solve this?
M: i always look on features, personally. i figure what i wanted to do and i just throw it to my team. and they figure out what's the best way, i trust their judgment. people doing their job, they should do it how they feel they should do it. i don't really get much involved in that.

S: over the history of tikiwiki so far, what do you observe how the importing and the acquiring of partners develop? was it more in the beginning, more right now, do you observe a change?
M: there's definitely a difference. i've always been interested in having partners, for the reasons discussed. what's different with time, as tiki is getting bigger and more known, it's a lot easier to start a partnership. recently, we started using phplayersmenu, that's a layer system which is not much used yet, but it's really nice and it's something lots of os applications could use. if i had written to this guy a year ago, i'm not sure he would even know what tiki is, and have a good response. now, when i write to a project, they know about tiki and it makes easier. people feel, that's an interesting project and we should look at it.

S: basically, there's your effort towards that, but the age of the project: did it come down to having more partners in recent times than when you started?
M: I'm not sure. because, earlier it was more like components: smarty, php, ADOdb, more stuff which was in the project, the low level, very programmer. stuff which we have now is more stuff which touches the user, more high, interface level. phplayersmenu, we have something now for the calendar, a javascript interface calendars. we try to get as many as we can – as soon as we need one, we look what's available. once you've done a feature, you can't redo it. in one sense i feel like the number of partners is slowing down, but it's also because it's hard to add when we had 8 releases in the first 10 weeks, it's tough to keep up that pace.

S: how would you say the license matters? is it important for importing knowledge?
M: that's a big point right now. we use the LGPL license, why i'm not sure that's how it was started and now that's the way it is. i don't think it's very realistic to change it, because there have been 100 people contributing code so it would have to get everyone to agree to change it. i'm not saying we want to change it, but it's not possible anymore. the difference between LGPL and GPL projects, basically, GPL projects can use our code but we can't use their code.

S: because yours would become GPL as well.
M: exactly. that makes it a bit more complicated for us. e.g. our webmail, i would like us to have a really good webmail interface. there are a few good projects, SquirrelMail, more.groupware's webmail is very good, IMP is very good, the problem is they're all GPL, more.groupware is another license and i don't think we're allowed to use it, so because we're LGPL it makes it a bit more difficult to include stuff. so that's a problem. however, there are advantages, well unclear word GPL, you know, if some people want

to use the code in commercial application, they can. that possibility may be a factor where some people may want to get involved. that licensing issue is always present in my mind when i look at applications. it could happen e.g. that we choose to partner with somebody not exactly with whom we would want to, we needed to be LGPL. it would be hard to have a module with a GPL project, because it's not always easy to know where the line is and we want things tightly integrated in the wiki syntax. we really like it, during the forum now, you can use wiki words, that's really nice, and it would be nice to have that also in the webmail. so but where would be the line to GPL in terms of changing the code. it's a touchy subject.

[34:39]

S: how does the language impact on what you can use from other projects?
M: you mean php?

S: yes, yours compared to others
M: i don't have much experience in the other ones, but why i chose tiki, in general when i look at applications, php looks like the ideal language. it's easy to use and fast and lots of people using it. so it's a good choice, and a definite advantage. it's easier to find programmers, easier to do patches, i think it's one of the factors why we're doing so well.

S: is there another factor you could think of that makes tiki particularly open for others and for features to be included? the size e.g.?
M: It's a bit like the highgrade sausage. (Advertising: "Everybody likes it
because everybody eats it and everybody eats it because every body likes"), where the more people using it it gets better the more people are using it. there's lots of really great projects out there, we don't know how many people are using which project, theoretically it's probably 80% of the people using 20% of the projects. since there's not much marketing done on os projects, lots of nice things don't get known. what was tough is getting past the initial point of getting registred at all the websites and getting recognition. it's a long process and not really dependent on you. you can do a good job on your code, but it doesn't mean it's known and there's reviews. one of the important things was version 1.7, that was like the first major version. up to 1.6. not that many people were using tiki because it was so new, maybe a few 100 sites. version 1.7, it was the first real big version, high visibility, and lots of features, so it was getting interesting. the fact that there's a wiki makes a big difference compared to all the other application. if you look at postnuke, zoops, geeklog, truepull, there's like dozens of really good projects but not many include a wiki. there may be a php wiki plugin or module, but it's not really tightly integrated, so the number of cms projects which really have tightly integrated wiki is pretty small. a year and a half ago, when it was called tikiwiki, but then it had so many features i felt it had to be called something else, so in version 1.7 we called it tiki cms groupware. i asked on the mailing list, and we had a few exchanges, but lets call it, we finally decided to call it tiki cms groupware. i decided that people don't really know what a wiki is, and those who do, we don't want them to think we're just a wiki. this is good, let's focus on the cms groupware. so an application that combines both in one is very interesting. there's lots of nice groupwar packages but not so many cms packages, packages that include everything, there's not that many. a year later, now wikis are getting more and more known, it's actually positive to be a wiki, in terms of marketing it becomes a differentiating factor. a look at it as market share and i want as many developers as possible. they have needs and they want a tool, my goal is to get them as much as possible to use wiki and to contribute their patches to tiki.

S: is there anything you could think of, what project- related factors influence the reuse of code and features?
M: it could be stuff like personal contacts. e.g. when we discussed using ADOdb, one of the reasons why we finally chose it, is we were in contact with one of the developers, because we were working on something else relating to ADOdb, and tried to get another module included. we were interested in one of the things he was doing, sometimes you don't have much to go on, so the fact that we liked one of the things he was doing, the other is probably also interesting. that was a factor. another example for contacts, we have a maps module now in tiki 1.8. the person who was working on an independent map

project, and he was friends with one of our developers, so that makes a lot easier, we didn't spend that much time looking at different map projects, he had one, he know our team, so that's how it happens.

S: does that mean the governance structure, basically the management style impact on that? have you made such an experience.
M: i don't know if i can call it management style or governance (laughs)

S: how you organize the project, or how people join...
M: it's very much a meritocracy. anyone who asks has access to the code. they just show us what they can do. one of our rules is, you can add anything you want as long as you don't break anything. and you make it optional. that's one of the things in the philosophy of the project, which makes it interesting for people to have projects, they know that if they need something specific, they just have to make it as generic as they can and add it to the code as an option. and it's in the code. and they don't have to worry about synchronizing versions later, they can just follow it, and somebody else may need that code at one point. obviously, this makes tiki very big. it's a big project, program, like a big boat. it's longer to steer, there could be performance or security issues, but so far we prefer to have it open and let people prove what they can do, and if there's a problem we will teach them or learn from them and correct. but so far it has worked out great, and i don't see many changes in the style. this is my first time in an os project, and at least two persons in the project, luis who started it, a very talented and experienced programmer who's written books on php, when he wrote tiki at the beginning, he did a great job at the basic infrastructure, and when mose joined and worked on 1.7. mose has lots of experience with os projects, and it wasn't just the code, but the development philosophy, how many branches we do and how we work in CVS, all that experience is really helping out. and the irc chatroom is also excellent, 24h /day, if you have questions, just pop on there and ask – somebody knows the answer. the people are very friendly. people are looking at different projects try a few out and come to our chatroom and get an answer, that could be the difference for them choosing tiki. since tiki has gone public, i could say, with v1.7. and the large distribution and diffusion, there's been lots of work on installation scripts, what used to be difficult to install, but since we get lots of support requests, the developers really worked on the install scripts to make it more robust and easier to use. that makes it easier for people to try it out and learn.

S: could you recommend me another programmer in tiki who i might also talk to?
M: i think you should speak with mose. we're six admins on the project, luis was the head developer, and for the last few months, mose has been the lead developer.
S: [thanks and outro]