E-mail exchange with Tim Starling, WMF software developer

The Bridgespan Group

E-mails to Tim:

I'm part of the Bridgespan part of the project team for the Wikimedia strategy project. We are in the process of preparing fact bases and analyses for each of the up-coming Task Forces to work on through December. A couple of the Task Forces will deal with technology and we are a little concerned with the divergent input and significant concern around scalability, stability, and innovation we've gotten from our interviews with Board, Advisory, staff, Wikipedians, and external experts re: the wiki platform, the network infrastructure, UI, QA, etc. I'm looking for a clinical assessment to help me balance the views we've heard to date. Philippe suggested you were the best person to talk to.

Specifically, I would like your opinion of the following:

- Network infrastructure:
  - scalability
  - network configuration
  - stability
  - redundancy
  - management
  - maintenance costs (too high vs "about right")
  - needed upgrades
  - reporting, analysis, and tools
- Wiki platform and in particular MediaWiki:
  - scalability
  - usability / adaptation across projects
  - limitations
  - patch upgrades vs wholesale upgrades needed
- UI
  - editing tools, developer tools, etc (basically tools)
  - editing interface
  - mobile platforms (reading & editing)
- Development & QA processes
  - development plan
  - balance of upgrades vs bug fixes vs patches, etc
  - release schedules & process
  - QA process

We can either set up a time to talk, or you can send me your thoughts via e-mail... Alternatively, are there documents, wikis, lists, etc that you can point us to that might help paint a more comprehensive picture?

More specifically, re: 2-4, what I'd like to know are your thoughts at a macro level on Wikimedia's ability to continue to expand and innovate in these areas to:

- continue to meet the growth and proliferation of projects
- continue to be technologically relevant (especially with respect to UI and tools for non-tech users); and
- potential issues or pitfalls with either current processes or systems such that scaling or innovating could be a problem.

In that you have specific case examples to point to as potential pros or cons, fantastic. I am not so well-versed in the particulars of the wiki platform and its processes, but it strikes me that the MediaWiki is on old architecture with a lot of patchwork fixes moving it forward, that the UI as everyone knows is not user-friendly, and that developer network for new platforms like Twitter, Facebook, and iPhone seem to multiply overnight, but that Wikimedia's developer networks aren't enjoying the same growth or importance within the community. (Though you might get rich with a Twitter tool - but is there still opportunity for Wikimedia to grow developers and it isn't happening?). Please let me know if you agree with these very broad, sweeping assessment - if not, please let me know.

Thanks, Tim and any insights you might have would be greatly appreciated.

Tim's responses:

There's a school of thought in software engineering that has it that incremental development from an existing code base is the best way to bring new features, and that rewrites are expensive and risky. There are several examples from both the commercial and open source world where rewrites have led to catastrophic failure. A previous discussion among senior MediaWiki software developers and community members showed that among the people who care to think about such issues, this school of thought is considered to be good, and that incremental development is the best long-term plan for MediaWiki.

This approach has certainly shown its flexibility, since developers (such as those at Wikia) have been able to bring great usability improvements to MediaWiki. No obstacle to this progress is currently evident, except the limitations of the language (PHP), which I think should have little impact.

The rate of progress on MediaWiki should be judged with respect to its tiny budget, and your comparison with large companies such as Facebook and Apple is not a fair one. Note, for instance, that we don't have any paid technical writers. Properly-written manuals could go a long way to encouraging community development.

If the strategy project identifies software development as a key need to meet our wider goals, then expansion of the software development team should be the primary response. Decisions on the implementation details (language choice, release schedule, etc.) can be made by that expanded team, in the course of the usual design process.