

Maintaining a Programmer Wiki

[closed]

Asked 16 years, 1 month ago Modified 11 years ago Viewed 5k times



20



Closed. This question needs to be more [focused](#). It is not currently accepting answers.



Want to improve this question? Update the question so it focuses on one problem only by [editing this post](#).

Closed 11 years ago.

[Improve this question](#)

I was recently put in charge of the wiki for the development team. The wiki is still in its infancy, so I have a lot of room to work with. Its goal is to house internal to the development team. Currently, the main piece of information that the wiki holds is Coding Standards.

- What are some best practices your dev team uses for its internal wiki?
- What information is important to have on a dev wiki?

- If you were to go to the wiki for your dev team what information would you expect to see?
- Is there some information that shouldn't go on the wiki even though it seems like a good idea?

-- edit --

- Also, is there a good way to organize the information? (such as by layer (data, ui), by project, or other)

wiki

Share

Improve this question

Follow

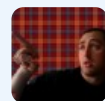
edited Dec 12, 2013 at 6:37



Kara

6,206 ● 16 ● 53 ● 58

asked Oct 29, 2008 at 18:51



wusher

12.4k ● 22 ● 73 ● 96

What wiki software are you using? The capabilities and therefore possibilities can be limited by the tool.

– [Thomas Owens](#) Oct 29, 2008 at 18:57

Also, what other tools do you have? If you don't have a bug tracker, you might want to use your wiki for that (better to get a bug tracker, of course). – [Jonathan Adelson](#) Oct 29, 2008 at 19:00

2 Noooooooooo, don't use a wiki for a bug tracker..... I've seen it happen before, it is NOT pretty. – [swilliams](#) Oct 29, 2008 at

11 Answers

Sorted by:

Highest score (default)



9



- Introduction to the source base for new programmers
- General documentation (not the API documentation per-se, but more tutorial like things)
- Lists of staff / who's doing what and how to reach them
- Notes / resources / articles that explain concepts used in the software
- Documentation of the build process and the filesystem layout of the codebase

Other things I usually put up there are

- Planning / todo lists
- Information that is interesting for others to read
- Everything else that I feel should be shared

Share Improve this answer

edited Jan 28, 2009 at 3:59

Follow

answered Oct 29, 2008 at 18:56



Jasper Bekkers

6,809 ● 36 ● 46



7



Share Improve this answer

answered Dec 9, 2008 at 4:38



Follow



[lillq](#)

15.3k ● 20 ● 54 ● 58

-
- 1 Link is dead. Current site: stewartmader.com/wikipatterns
– [cyclingLinguist](#) Aug 28, 2023 at 12:02
-



6



We had a development wiki and it was a great tool. We used it for *everything*!

- When brainstorming new ideas, we'd capture them on the wiki. The low friction nature of the wiki made it easy for anyone in the organization (we were a small startup) to add ideas as they thought of them. We had a high level "brainstorming" page which linked to detailed pages containing a thorough description of each idea.
- For each iteration, we'd "move" feature idea items from the "brainstorming" list to the feature list for that iteration. The details of the feature were flushed out to include design and implementation details.
- As features were completed, the iteration page became our release notes page - which also

included the release tag from our version control system.

- We had a bug page that worked very similar to the feature pages. Bug fixes were added to the iteration/release pages as they were worked on/complete.
- We also created our user documentation on the wiki and exported those pages it with the release.

As time went on. This tool was viewed more and more valuable. We wound up creating new wikis for different the products the company was working on.

I hope you find your development wiki as useful as we did!

Share Improve this answer

answered Oct 29, 2008 at 19:45

Follow



Vinnie

12.7k ● 15 ● 61 ● 80



2



One thing that we stress on our dev wiki is that it is updated when things change. We don't want our wiki that is intended to provide information and be a central source of collected knowledge to become so out of date that it is useless. As the code is updated, developers are requested to update any related information on the wiki.



Other than Coding Standards, we keep tips and tricks for working with our code base, setup information for new hires, and general environment information.

Share Improve this answer

answered Oct 29, 2008 at 19:00

Follow



CalvinTreg

446 ● 1 ● 4 ● 11

One of my roles as "wiki master" is to make sure that everything is up to date and get those in charge of out of date content to update their section. – [wusher](#) Oct 29, 2008 at 19:04

At my last company another developer and I made a Perl script to check for out of date wiki pages, and flag them along with the name of the last person to edit them. People could then decide to delete the page, or update it with new info. – [KeyserSoze](#) Oct 29, 2008 at 20:11

That sounds like a cool side project – [wusher](#) Oct 29, 2008 at 21:16



The hardest part is getting developers to use your wiki. I have some long standing suggestions here:

2

<http://possibility.com/wiki/index.php?title=GettingYourWikiAdopted>



Getting a Wiki Adopted is Tough



Have a Champion



Remove Objections

Create Content

Enmesh the Wiki In Company Processes

Evangelize

Don't Give Up

Consider Not Using Wiki For Conversations

Just Do It! Don't Wait For a Budget

Have a Transition Plan

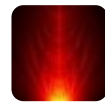
Promoting Your Wiki

One good practice is to have the entire documentation and source code for each build available through your wiki. Then developers will go to wiki to access build info and that makes it invaluable.

Share Improve this answer

answered Oct 29, 2008 at 19:56

Follow



Todd Hoff

762 ● 5 ● 12



2



Wikis can be a valuable resource for software development teams but they are not a silver bullet. It is all too easy to create a Wiki that would rapidly fall into disuse or become grossly outdated.



In my opinion, the key to a successful Wiki is getting the entire team on board. That means getting people away from other resources (and in particular email archives) as knowledge repositories, and offering some incentive for people to contribute.

However, it's also important to not be a format czar: If you have a lot of documents that you generate in, say, MS WORD, it may be ideal to do them all in Wiki format but that takes time and may be annoying if you have diagrams, documents, etc. In those cases, it's better to compromise and let people keep it in word format, as long as the only way to access the newest version is through the Wiki.

If you're not a manager, you need to get a manager on board because it would require some "enforcement".

There has been accumulating research and experience on Wikis and their use in software engineering. You can search the ACM digital library, for example. I am a coorganizer of an annual workshop on wikis for SE and we had several interesting experience reports and there are additional materials in the international symposium on Wikis.

Share Improve this answer

answered Oct 29, 2008 at 19:56

Follow



Uri

89.6k ● 51 ● 226 ● 322



1



- Burndown charts
- common setup information for development environments (nice for when new people start)
- Specs
- Known issues and workarounds with development tools



Share Improve this answer

answered Oct 29, 2008 at 19:00

Follow



Gord

1,825 ● 3 ● 17 ● 22



1



Come up with some kind of style guide, and teach others how to style stuff. When I was in charge of a corporate wiki, all of the other developers would just write crummy prose that was barely formatted, and looked terrible.

Keep away from things that require discussion. I tried shoehorn in a book review section, but it was too difficult to have others comment on things.

Examples of in house libraries are good. And/or "storyboards" walking a user through a process when MethodX is called.

Share Improve this answer

answered Oct 29, 2008 at 19:00

Follow



swilliams

48.8k ● 27 ● 102 ● 130



1



What are some best practices your dev team uses for its internal wiki?

Make it look nice. I know it doesn't sound important, but if you spend a little time branding it pays off in terms of people actually using it. And uptake is key, or it will just wither and die.

What information is important to have on a dev wiki?

- General information about a Project, milestones, delivery dates etc.
- Summaries of design decisions/meetings. Important so that you don't re-visit the same areas time and time again.
- HowTo guides for general development of current projects (for example, how to develop a new Plugin)

If you were to go to the wiki for your dev team what information would you expect to see?

Project information, who is working on what etc. Design decisions. Also best practices and links to useful sites.

Is there some information that shouldn't go on the wiki even though it seems like a good idea?

Low-level task lists tend to fluctuate and not be kept up-to-date, and can be misleading. Also, critical communications between departments are better suited to e-mail, THEN the conversation can be copied to the wiki. It's too easy to ignore it otherwise!

Share Improve this answer

answered Oct 29, 2008 at 19:07

Follow



Duncan

10.3k ● 14 ● 66 ● 96



Remember that a wiki is interactive. If you're thinking about publishing, as in publishing burndown charts, then

1

you're not thinking far enough. Distributing that information is only part of it.



For instance, rather than having a "Current Burndown Chart" page, create a page for "Burndown Chart for Week of 10-27-2008" and then encourage people to comment on the chart, and what it means, and why you did so poorly that week.

Share Improve this answer

answered Oct 29, 2008 at 19:48

Follow



[Andy Lester](#)

93.5k ● 15 ● 104 ● 159

why assume they did poorly ;) – [xentek](#) Jan 2, 2011 at 5:40



1

We house and inhouse team wiki. And there we put all the necessary information for each project we are developing:



- repositories
- addresses for virtual machines
- passwords
- project documentations
- project overview
- project status

and anything else we fill needs to be written on a project. And it is the most useful web application we are running (besides [Mantis](#)) . On more general pages we put a

definition of every taxonomy we are using, general project guidelines, policies, coding and developing practices we use. It is there, it is simple and effective and I think every team should have one of those.

Share Improve this answer

answered Oct 29, 2008 at 22:09

Follow



Nikola Stjelja

3,687 ● 9 ● 39 ● 47
