

Allocating resources for project documentation [closed]

Asked 16 years ago Modified 15 years, 10 months ago

Viewed 243 times



1



Closed. This question is [opinion-based](#). It is not currently accepting answers.



Want to improve this question? Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 6 years ago.

[Improve this question](#)

What would you suggest for the following scenario:

A dozen of developers need to build and design a complex system. This design needs to be documented for future developers and the design decisions must be noted. These reports need to be made about every two months. My question is how this project should be documented.

I see two possibilities. Each developer writes about the things they helped design and integrate and then one person combines each of these documents together. The

final document will probably be incoherent or redundant at times since the person tasked of assembling everything won't have much time to adjust every part.

Assume that the documentation parts from each developer arrive just a few days before deadline. A collaborative system (i.e. wiki) wouldn't work properly since there wouldn't be anything to read until a few days before deadline.

Or should a few people (2-3) be tasked with writing the documentation while the rest of the team works on actually developing the system? The developers would need a way to transfer their design choices and conclusions to the technical writers. How could this be done efficiently?

documentation

project-management

Share

Improve this question

Follow

edited Feb 20, 2009 at 19:02



George Stocker

57.9k ● 29 ● 180 ● 238

asked Dec 13, 2008 at 20:52



JcMaco

1,278 ● 3 ● 17 ● 32

5 Answers

Sorted by:

Highest score (default)





1



We approach this from 2 sides, using a RUP style approach. In the first case, you'll have a domain expert who is responsible for roughing out the design of what you're going to deliver - with developers chipping in as necessary. In the second case, we use a technical author - they document the application, so they should have a good idea of how it hangs together, and you involve them right through the design and development process. In this case, they can help to polish the design, and to make sure that it matches what they thought was being developed.

Share Improve this answer

answered Dec 13, 2008 at 21:09

Follow



Pete O'Hanlon

9,146 ● 2 ● 31 ● 28



1



We use confluence (atlassian's wiki-like-thing) and document all kinds of different "things". The developers do it continuously, and we push each other for docs - we let peer pressure decide what is necessary. Whenever someone new comes along he/she is tasked with reading through everything and to find out what still is correct. The incorrect stuff is either deleted or updated as a consequence of this. We're happy when we can delete stuff ;)

The nice thing about this process is that the relevant stuff stays and the irrelevant stuff is deleted. We always "got away" from the more formalized demands by claiming that we could always construct the word documents they

wanted if "they" needed them. "They" never needed them.

Share Improve this answer

answered Dec 13, 2008 at 21:01

Follow



[krosenvold](#)

77k ● 33 ● 156 ● 209



1

I think alternative 2 is the less agile, because it means a new stage to the project (although it may be in parallel with tests).



If you are in an agile model, then just add documentation (following a guideline) as a story.



If you are in a staged approach, then I would nevertheless ask developers to work on documentation, following some guidelines, and review that documentation along the design and the code. Eventually, you may have a technical writer reviewing everything for proper English, but that would be a kind of "release" activity.

Share Improve this answer

edited Dec 13, 2008 at 21:37

Follow

answered Dec 13, 2008 at 21:05



[rob](#)

37.6k ● 2 ● 60 ● 66



I think you can use Sand Castle to document your project.

0

Check it out



[Sand Castle from Microsoft](#)



Share Improve this answer

answered Dec 13, 2008 at 20:55



Follow



[Tarik](#)

81.6k ● 86 ● 241 ● 351



0

It's not a complete documentation, but making sure that interfaces etc. are commented using [Doxygen-style comments](#) means writing code and documenting it are closer together.



That way, developers should document what they do. I still think a review by the architect(s) is needed to ensure consistent quality, but ensuring people document what they do is the best way to ensure they follow the architecture.



Share Improve this answer

answered Dec 13, 2008 at 21:01

Follow



[Peter K.](#)

8,128 ● 4 ● 51 ● 75