

How to write a spec that is productive? [closed]

Asked 16 years, 4 months ago Modified 2 years, 3 months ago

Viewed 991 times



2



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 5 years ago.

[Improve this question](#)

I've seen different program managers write specs in different format. Almost every one has had his/her own style of writing a spec.

On one hand are those wordy documents which given to a programmer are likely to cause him/her missing a few things. I personally dread the word documents spec...I think its because of my reading style...I am always speed reading things which I think will cause me to miss out on key points.

On the other hand, I have seen this innovative specs written in Excel by one of our clients. The way he used to write the spec was kind of create a mock application in Excel and use some VBA to mock it. He would do things like on button click where should the form go or what action should it perform (in comments).

On data form, he would display a form in cells and on each data entry cell he would comment on what valid values are, what validation should it perform etc.

I think that using this technique, it was less likely to miss out on things that needed to be done. Also, it was much easier to unit test it for the developer. The tester too had a better understanding of the system as it 'performed' before actually being written.

Visio is another tool to do screen design but I still think Excel has a better edge over it considering its VBA support and its functions.

Do you think this should become a more popular way of writing spec? I know it involves a bit of extra work on part of project manager(or whoever is writing the spec) but the payoff is huge...I myself could see a lot of productivity gain from using it. And if there are any better formats of specs that would actually help programmer.

project-management

specs

Share

Improve this question

edited Aug 16, 2017 at 9:14

Follow



Vadim Kotov

8,274 ● 8 ● 50 ● 63

asked Aug 22, 2008 at 18:18



Krantz

6,273 ● 3 ● 26 ● 18

-
- 2 This question appears to be off-topic because project management questions are no longer on-topic. See pm.stackexchange.com. – LittleBobbyTables - Au Revoir Jun 6, 2014 at 13:53
-
- 3 I'm voting to close this question as off-topic because it's about project management. – EJoshuaS - Stand with Ukraine Aug 16, 2017 at 14:12
-

4 Answers

Sorted by:

Highest score (default)



[Joel on Software](#) is particularly good at these and has some good articles about the subject...

5

A specific case: [the write-up](#) and [the spec](#).



Share Improve this answer

edited Sep 1, 2022 at 10:21



Follow



Milo Fultz

17 ● 1 ● 5



answered Aug 22, 2008 at 18:22



TheEmirOfGroofunkistan

5,644 ● 8 ● 39 ● 55



Two approaches have worked well for me.

3



One is the "working prototype" which you sort of described in your question. In my experience, the company contracted a user interface expert to create fully functional HTML mocks. The data on the page was static, but it allowed for developers and management to see and play with a "functional" version of the site. All that was left to do was replace the static data on the pages with dynamic content - this prototype was our spec for the initial version of our product. The designer even included detailed explanation of some subtle behavior in popup dialogs that would appear when hovering over mock links. It worked well for our team.



On a subsequent project, we didn't have the luxury of the UI expert, but we used similar approach. We used a wiki to mock a version of the site. We created links between the functional aspects of the system and documented each piece of functionality in detail. Each piece of functionality could, in turn, link to detailed design and architecture decisions. We also used to wiki to hold our to list feature list for each release (which became our release notes). These documents linked back to the detailed feature page. The wiki became a living document - describing our releases and evolution of our system in great detail. It was an invaluable resource.

I prefer the wiki to the working prototype because it's more easily extensible - growing and becoming more valuable as your system evolves.

Share Improve this answer

answered Sep 2, 2008 at 18:48

Follow



Vinnie

12.7k ● 15 ● 61 ● 80



2

I think you may have a look about Test-Driven Requirements, which is a technique to make executable specifications.



There are some great tools like [FIT](#), [Fitnesse](#), [GreenPepper](#) or [Concordion](#) for that purpose.



Share Improve this answer

answered Aug 29, 2008 at 8:49

Follow



paulgreg

18.9k ● 18 ● 48 ● 56



0

One of the Microsoft Press books has excellent examples of various documents, including an SRS (which I think is what you are talking about). It might be one of the requirements books by Weigert (I think that's his name, I'm blanking on it right now). I've seen US government organizations use that as a template, and from my three work experiences with the government, they like to make their own wherever they can, so if they are reusing it, it must be good.



Also - a spec should contain NO CODE, in my opinion. It should focus on what the system must do, should do, and can not do using text and diagrams.

Share Improve this answer

answered Aug 22, 2008 at 18:22

Follow



Thomas Owens

116k ● 99 ● 317 ● 436
