

Managing user stories for a large project [closed]

Asked 16 years, 3 months ago Modified 13 years, 2 months ago

Viewed 4k times



13



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



Want to improve this question? [Update the question](#) so it's [on-topic](#) for Stack Overflow.

Closed 11 years ago.

[Improve this question](#)

We are just starting on a pretty big project with lots of sub projects. we don't currently use any kind of named process but I am hoping to get some kind of agile/scrumlike process in by the back door.

The area I will be focusing on most is having a good backlog for the whole project and, at least in my head, the idea of an iteration where some things are taken from the backlog, looked at in more detail and developed to a reasonable deadline.

I wonder what techniques people use to break projects down into things to go in the backlog, and once the

backlog is created how it is maintained and ordered. also how relationships between elements are maintained (ie this must be done before it is possible to do that, or this was one story now it is five)

I am not sure what I expect the answer for this question to look like. I think what may be most helpful is if there is an open source project that keeps its backlog online in some way so I can see how others do it.

Something else that would get +1 from me is examples of real user stories from real projects (the "a user can log on" story does not help me picture things in my project.

Thanks.

project-management

agile

scrum

user-stories

Share

Improve this question

Follow

asked Sep 22, 2008 at 22:45



Loofer

6,969 ● 9 ● 65 ● 106

-
- 1 I had to upvote you for two reason. I like to hear about people ideas on stuff like this, but also your reputation was sitting on 666! – [mattlant](#) Sep 23, 2008 at 1:17
-

7 Answers

Sorted by:

Highest score (default)





9



I would counsel you to think carefully before adopting a tool, especially since it sounds like your process is likely to be fluid at first as you find your feet. My feeling is that a tool may be more likely to constrain you than enable you at this stage, and you will find it no substitute for a [good card-wall in physical space](#). I would suggest you instead concentrate your efforts on the task at hand, and grab a tool when you feel like you really need one. By that stage you'll more likely have a clear idea of your requirements.

I have run several agile projects now and we have never needed a more complex tool than a spreadsheet, and that on a project with a budget of over a million pounds. Mostly we find that a whiteboard and index cards (one per user story) is more than sufficient.

When identifying your stories, make sure you always express them in terms that make sense to your users - some (perhaps only small) piece of surfaced functionality. Never allow yourself to slip into writing stories about technical details that you could not demonstrate to a user.

The skill when scheduling the stories is to try to prioritise the things you know least about first (plan for what you want to learn, rather than what you want to do) whilst also starting with the stories that will allow you to develop the core features of your application, using subsequent stories to wrap functionality (and technical complexity) around them.

If you're confident that you can leave some piece of the puzzle till later, don't sweat on getting into the details of that - just write a single story card that represents the big conversation you'll need to have later, and get on with the more important stuff. If you need to have a feel for the size of what's to come, look at a wideband delphi estimation technique called [planning poker](#).

The Mike Cohn books, particularly [Agile Estimating and Planning](#) will help you a lot at this stage, and give you some useful techniques to work with.

Good luck!

Share Improve this answer

Follow

answered Sep 23, 2008 at 11:07



[mattwynne](#)

2,198 ● 2 ● 16 ● 15



4

Like DanielHonig we also use RallyDev (on a small scale) and it sounds like it could be a useful system for you to at least investigate.



Also, a great book on the user story method of development is [User Stories Applied](#) by Mike Cohn. I'd certainly recommend reading it if you haven't already. It should answer a lot of your questions.



Share Improve this answer

Follow

answered Sep 23, 2008 at 2:01



[Bazman](#)

1,229 ● 9 ● 10



2



I'm not sure if this is what you're looking for, but it may still be helpful. Max Pool from codesqueeze has a video explaining his "agile wall". It's cool to see his process, even if it may not necessarily relate to your question:

[My Agile Wall \(Plus A Few Tricks\)](#)



Share Improve this answer

answered Sep 22, 2008 at 23:40

Follow



[brock.holum](#)

3,213 ● 2 ● 22 ● 15



2



So here are a few tips: We use RallyDev. We created a view of packages that our requirements live in. Large stories are labeled as epics and placed into the release backlog of the release they are intended for. Child stories are added to the epics. We have found it best to keep the stories very granular. Coarse grained stories make it difficult to realistically estimate and execute the story.



So in general:

1. Organize by the release
2. Keep iterations between 2-4 weeks
3. Product owners and project managers add stories to the release backlog
4. The dev team estimates the stories based on TShirt sizes, points, etc...

5. In Spring planning meetings the dev team selects the work for the iteration from the release backlog.

This is what we've been doing for the past 4 months and have found it to work well. Very important to keep the size of the stories small and granular.

Remember the Invest and Smart acronyms for evaluating user stories, a good story should be: I - Independent N - Negotiable V - Valuable E - Estimable S - Small T - Testable

Smart:

S - Specific M - Measurable A - Achievable R - Relevant T - Time-boxed

Share Improve this answer

edited Sep 23, 2008 at 1:14

Follow

answered Sep 23, 2008 at 1:05



Daniel Honig

4,418 ● 6 ● 28 ● 24



1



I'd start off by saying Keep it Simple.. use a shared spreadsheet with tracking (and backup). If you see scaling or synchronization problems such that maintaining the backlog in a consistent state is getting more and more time-consuming, trade up. This will automatically validate and justify the expenditure/retraining costs.



I've read some good things about [Mingle from Thoughtworks](#).

Share Improve this answer

answered Sep 23, 2008 at 1:47

Follow



Gishu

137k ● 47 ● 226 ● 311



here is my response to a similar question that may give you some ideas

1

[Help a BA! Managing User Stories ...](#)



Share Improve this answer

edited Sep 18, 2011 at 21:39



Follow



Andrew Whitaker

126k ● 32 ● 295 ● 308



answered Feb 3, 2011 at 16:08



Anup M

11 ● 1



1

A lot of these responses have been with suggestions about tools to use. However, the reality is that your process will be the much more important than the tools you use to implement the process. Stay away from tools that attempt to cram a methodology down your throat. But also, be wary of simply implementing an old non-agile process using a new tool. Here are some strong facts to consider when determining tools for processes:



1. A bad process instrumented with a software tool will result in a bad software tool implementation.
2. Processes will change based on the group you are managing. The important thing is the people, not the process. Implement something they can work successfully in, and your project will be successful.

All that said, here are a few guidelines to help you:

- Start with a pure implementation of a documented process,
- Make your iterations small,
- After each iteration talk with your teams and ask what they they would change, implement the changes that make sense.

For larger organizations, if you are using SCRUM, use a cascading stand-up mechanism. Scrum masters meet with thier teams. Then the Scrum Masters meet in stand-ups of 6 - 9, with a Super-Scrum-MAster responsible for reporting the items from the Scum-Master's scrum to the next level... and so forth..

You may find that have weekly super-scrum meetings will suffice at the highest level of your hierarchy.

Share Improve this answer

answered Sep 20, 2011 at 2:53

Follow



Mike Van

1,058 ● 7 ● 26