

UI Driven Development [closed]

Asked 15 years, 8 months ago Modified 8 years, 3 months ago

Viewed 4k times



6



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 7 years ago.

[Improve this question](#)

Does the idea of UI Driven Development make sense at all? Most of our clients like to convey their requirements in form of screens. For e.g. I want a screen to do THIS and THAT. Sometimes they even go as far as to dictate the layout of a screen by themselves (this may be because clients of today use software applications for most of their tasks already).

Also this method of requirement gathering seems to convey both the Data and Associated Behavior automatically.

What do you guys think?

[user-interface](#)[requirements](#)[Share](#)[Improve this question](#)[Follow](#)

asked Apr 2, 2009 at 9:15

[Ali Kazmi](#)

3,660 ● 6 ● 36 ● 52

5 Answers

Sorted by:

Highest score (default)



6

Actually it does make sense in this regard. Use cases differ here in that they cannot be used in the same way.

Use cases will be usefull to extract and formalize requirements.



I was workinng in a UI lab a while back and we toyed with this concept, though we did not call it as such. Basic idea here is that we would use agile iterative approach to development where we would use usability tesing to help converge on a desired solution.

Typical cycle would be :

- draft a few requirements and use cases, very limited in scope very focused.
- Create a test protocol that would allow us to gather data on this feature (usability, ease of use, acceptance, performace etc).
- Create or expand the application to include this feature

- have a few users test the application using agile adaptation of usability testing (see [Ruben's Handbook of Usability Testing](#)) where we would limit test sessions to 15 minutes with a 15 minutes debriefing.
- Extract useful data from testing to inject in future iteration.

This method was particularly useful when the users would either not know exactly what they wanted or could not tell us. We would therefore have to devise tests that would gather objective data on the actual usefulness of the software to the user and try to adjust the next iteration consequently. (many of our "customers" if I may call them so were heavily handicapped and could not talk so we had to be creative).

This way of working forced us to have a GUI available to the client very early in the software's development life and because it was the central point where we would direct testing one could say it was GUI driven design since the driving force for convergence was the user interactions with the system.

Although we developed this technique mostly for very specific cases we did some testing with normal users on a normal software and got very positive results. The design would converge very quickly towards the user's needs. Also the fact that they participated in this design approach had also very positive effects on the acceptance of the product in the target community.

Unfortunately internal strifes had the lab dismantled before we could publish our results and expand this line of research, a shame really.

So UIDD (if you pardon the bad acronym) would be a member of the TDD family of approaches towards software development where the iterations hinges around user interactions.

Additionnal reading on the subject :

<http://www.codinghorror.com/blog/archives/001091.html>

<http://cakebaker.42dh.com/2007/07/07/usability-driven-development/>

<http://www.springerlink.com/content/l413k76812896gnt/>

<http://www.agilemodeling.com/essays/agileUsability.htm>

<http://www.uxbooth.com/blog/how-test-driven-development-increases-overall-usability/>

Share Improve this answer

edited Apr 2, 2009 at 9:54

Follow

answered Apr 2, 2009 at 9:37



Newtopian

7,692 ● 4 ● 51 ● 73



2

what kind of development are you doing? When i develop web apps (flex or php) I work with drawings (wireframes) so I can get the client to know the flow/look of the app without writing any code.



Share Improve this answer

answered Apr 2, 2009 at 9:50

Follow



[Rick J](#)

2,703 ● 2 ● 22 ● 31



1

I find this is a reasonable way to gather requirements. I'd be careful actually building the screen and then developing functionality off that direct. You will find code reuse is low and you'll have a pretty looking app that's slow or doesn't work exactly right. It will work on some projects and not others depending on how complex the requirements and how good at foreseeing problems such as data bottlenecks, caching issues etc.



It sounds like it's similar to behaviour driven development, wireframing and user stories.

Share Improve this answer

answered Apr 2, 2009 at 9:20

Follow



[Rob Stevenson-Leggett](#)

35.7k ● 21 ● 91 ● 142



0

I think what you are really talking about here are **use cases** - see [this article](#) for an intro.

Share Improve this answer

answered Apr 2, 2009 at 9:18



Follow



anon



I disagree... Use case are a tool but they do not necessarily drive the development as would TDD or BDD do. Use cases are on the same level as unit tests in TDD, there is much more to TDD than unit tests, as there is much more to UI DD than use cases. – [Newtopian](#) Apr 2, 2009 at 9:40



0

Story Test Driven Development I think is quite a good idea as long as you take the automation part of it seriously. If you can't automate your tests then you're in trouble. Check out a paper on it here:



<http://www.industriallogic.com/papers/storytest.pdf> It doesn't sound like your clients are giving you automated UI tests but I think that's the best way to go.



Share Improve this answer

answered Apr 2, 2009 at 9:20

Follow



[Jonathan Parker](#)

6,795 ● 4 ● 46 ● 55
