Why you should consider behavior-driven development

by Pablo Sebastian Soifer - Monday, May 22, 2017

http://blog.belatrixsf.com/why-you-should-consider-behavior-driven-development/

Behavior-driven development (BDD) is an increasingly popular software development methodology which evolved out of more traditional Agile development processes. However, many companies are reluctant or hesitant to change how they work, or to implement a new methodology. In this blog, I want to highlight some of the key benefits of BDD, and why you should consider it.

- **Better communication between business and development.** One of the central tenets of BDD is using a shared common language between technical and non-technical individuals. This means communication between the business, developers and quality assurance (QA) improves. In my experience, the cause of many problems in projects is simply communication. For the business, the priority is what the system will do – what is the outcome, while developers typically have the mindset of how the system can achieve this outcome. Interpretation problems can easily emerge. Using a common language means the developer will understand better what needs to be done, while people from the business side can better express what their expectations are.
- **Business people get more involved in product development.** As business stakeholders can write scenarios and communicate exactly what they want, they start to be more involved in the development process. In the beginning, this can be hard work as they are not experts in writing scenarios, but as they learn, and with proper guidance, they become better at writing them. In turn, they quickly see how this is a better way to communicate what they need. In my experience, once they understand how this works, they acknowledge that this is a great method and see writing scenarios as an excellent tool and a time saver.
- **Self-documenting.** As developers write code for different scenarios, the code is already documented. Or to put it another way, the scenarios that you have created, become your documentation. And this has an additional advantage, that as the business have been involved in writing the scenarios, everyone understands it.
- **Code that is easier to maintain.** As code is always documented, it is easier to make changes and verify that it is still running correctly after every change.
- **Better code.** With documentation and with the code easy to maintain, making changes to it should be simpler. As a result, as the system grows, adding new code becomes straightforward. This all makes it possible to code faster, without losing quality.
- **Easier to test.** For QA engineers, it is better to have scenarios written before testing cycles start. This means they can focus on what to test, how to test it and give priority to some scenarios as the behaviors have already been described. Reflective of this change, in BDD your QA engineers will talk in terms of the scenarios, rather than “tests”. It is unsurprising that there are numerous benefits for testing when using BDD, as the methodology evolved from test-driven development (TDD).
- **Visibility.** With described scenarios working it is simple to see how much work has been done, how was it done, what was done and what should be done. It is a great progress indicator.

I have highlighted some of the key reasons to implement BDD based on my experience with the methodology, but there are many more. If you want to find out more about how BDD could help your software project, please download our whitepaper.