Bare Demo of IEEEtran.cls for Conferences

Gabriel Oliveira, Sabrina Marczak
Computer Science School, PUCRS
Porto Alegre, Brazil
gabriel.pimentel@acad.pucrs.br,
sabrina.marczak@pucrs.br

***Abstract*—Behavior-Driven Development (BDD) is a set of software engineering practices which uses an ubiquitous language, one that business and technical people can understand, to describe and model a system by a series of textual scenarios. Those scenarios serve not only as the project documentation but also as executable steps that specialized tools use to verify if the product has an acceptable set of behaviors. BDD tools can be used on continuous integration environment, thus enabling the documentation to be more effectively used during development and guaranteeing that changes on it are proper reflected on the product tests. Thus, BDD is a practice that supports Continuous Software Engineering by providing documentation based tests to assist Continuous Integration process and straightening the boundaries between testing activities and coding to aid on Continuous Testing approach. Our intuition leads us to believe that the value of those documentation based scenarios is connected with how well they convey and document the details discussed by the team about the behaviors needed to fulfil customer needs. Therefore, making sure that only "good" scenarios are used by constantly inspecting them should be an important activity during a software life cycle. Still, there is no study addressing the problem of what makes a "good" BDD scenario, the definition of quality on a BDD scenario or how to inspect it properly. We take inspiration on the criteria used to evaluate other type of requirements (like use cases or user stories) and on the strategies to inspect them to guide us on the reflection about how those concepts can be useful to BDD scenarios. Additionally, this paper reports on our experience with novice requirements writers and the pitfalls involved on the evaluation of scenarios.**

Keywords*—*BDD, Behavior-Driven Development, requirements quality, quality inspection.

I. Introduction

Paragraphs ideas/order:

- what is BDD, what are scenarios, how are used
- BDD helps on Continuous Integration
- BDD helps on Continuous Testing
- Problems with bad writing (examples of how affect Continuous Integration, lack of trust on flaky tests, lack of requirements documents on agile)
- Quality of tests is important as they're the only documentation on agile

II. Background

blabla etc to introduce it further

A. BDD - scenarios, how to automate, how to write, examples using it on CI envs

[1] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

B. Requirements Quality - Use cases (BABOK characteristics) and User Stories (INVEST)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

C. Reviews and Inspections and Reading Techniques

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

III. Evaluating BDD scenarios trough quality attributes

Paragraphs ideas/order:

- Attribute list used and motivation

- What was needed when evaluating (presence/absence of attribute on a qualitative explanation, attribute interpretation)

IV. Study

Paragraphs ideas/order:

- Explain study design (2 products, one requirement format per product, cross design)
- First product description (allergy app)
- Second product description (allergy app)
- Participants evaluation of each other scenarios - design motivation on who evaluates who

V. Discussion

Paragraphs ideas/order:

- writing difficulties/easiness between use cases and scenarios
- rigour on writing importance,
- solution knowledge importance (write scenarios are easy, but not writing good scenarios that the team could use)
- bad attributes to analyze scenarios (maintainability, completeness, atomicity)
- results (?)

VI. Conclusion & Future Studies

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

References

[1] J. Smart, *BDD in Action: Behavior-Driven Development for the Whole Software Lifecycle*. Shelter Island, NY: Manning Publications, 2014.