What recent scientific papers are telling about Test Driven Development

Felipe A. P. De Lima

Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 - Rebouças CEP 80230-901 - Curitiba - PR - Brasil

felipepastoree@gmail.com

Abstract. *This paper aims to introduce test driven development and provide an literature review by gathering some relatively recent empirical studies with scientific bases, published from years 2008 to 2012, and check what they are telling about test driven development.*

1. About Test Driven Development

Test driven development is a technique that relies in short cycles of repetitions, where the developer writes a unit test case that defines a new functionality or an improvement, than writes the code to make the test pass. When the tests passes, the developer than refactory the code, making it as good as possible, with the assurance that the tests will show if anything goes wrong.

Because users are forced to think about the functionality before writing the code, this technique isn't just a code technique, it influences in design too.

2. What studies are researching about

The research found five principal topics about test driven development on studies collected, they are: advantages to quality in development and class design, the impediments on adoption, the mistakes on adoption and student adherence on test driven development.

	Advantages to Quality	Advantages to Class Design	Impedments on Adoption	Mistakes on Adoption	Student Adherence
Factors Limiting Industrial Adoption of Test Driven Development A Systematic Review.			X		
Most Common Mistakes in Test-Driven Development Practice: Results from an Online Survey with Developers.				X	

Does Test-Driven Development Really Improve Software Design Quality?	X				
Como a Prática de TDD Influencia o Projeto de Classes em Sistemas Orientados a Objetos: Padrões de Feedback para o Desenvolvedor.	X	X			
Exploring Influences on Student Adherence to Test-Driven Development.	X				X

3. The Advantages to Quality

The goal of a new technique for software development, such as test-driven development, is to bring improvements in quality, performance, software or both.

Janzen, D., Saiedian, H. (2008) made experiments with students and industry developers, analysing the impact of test driven development and test last approach. They found that developers using test driven development had a possible tendency to write smaller, simpler classes and methods. Because smaller classes and methods tend to create more connections between objects, the study showed that developers writing tests first tend to write code with more coupling. However, according to the authors, possible increases in abstractness might indicate that the higher coupling is a good kind of coupling, resulting in more flexible software.

Aniche, M. F., Gerosa, M.A. (2012) in their experiments affirms that test driven development on his own don't guides the developer to a better class design. On the other hand, they affirm that test driven development gives fast feedbacks that allows developers to see if the code design needs more improvements. When developers find hard to write the test is because something is wrong with the class design.

4. The Impediments on Adoption

Causevic A., Punnekkat S. and Sundmark D. (2011) made a deep literature research to find the limiting factors on adopting test driven development. They found seven primary factors. Increased development time: the authors found that the majority of researched studies reported negative experience with development time. Insufficient test driven development experience/knowledge: two industrial experiments attributed problems of implementing test driven development to lack of test driven development education/experience. Insufficient design: the authors say that there are a handful of studies reporting problems regarding lack of design in test driven development, particularly in the development of larger, more complex systems. Insufficient developer testing skills: the authors found in the studies evidences of insufficient developer testing skills as a limiting factor. Insufficient adherence to the test driven development protocol: the authors found that developers sometimes were abandoning the test driven development protocol because of time pressure, lack of discipline, and shortage of perceived benefits. Domain- and tool-specific limitations: the studies found that the

automatically testing GUI applications was the most reported technical problem related with test driven development. Legacy code: Since test driven development, in its original form, does not discuss how to handle legacy code, adoption of test driven development might be problematic in large development organization.

5. The mistakes on adoption

Aniche, M. F., Gerosa, M.A. (2010) made an online survey with developers to check and measure the common mistakes in test driven development practice. The authors says that, besides test driven development is theoretically a simple technique, in practice the steps of test driven development are not that easy to follow as programmers need to be very disciplined. They say that this might reflect why programmers are induced to make some mistakes, which might lead code to a poor quality and unexpected behaviours. The main mistakes they observed on developers using test driven development was to forget the refactoring step, building complex test scenarios, and refactor another piece of code while working on a test.

6. Student adherence to test driven development

Buffardi K., Edwards, S. H. (2012) made a research exploring influences on student Adherence to Test-Driven Development. The authors collected data of the student practicing test driven development, using a tool named Web-CAT, and the authors asked students questions about skills in Computer Science, behaviours on developing programs, their experience with test-driven development, the impact of test-driven development in the quality of their code and design, and the experience with Web-CAT. The results found, when positive response to test driven development on questions, was significant correlations between self-reported adherence to test driven development and perceived helpfulness of test driven development, students who adhered to test driven development also benefited with significantly better test coverage and solution correctness, and test driven development principles of test-first and unit testing also correlated with adherence to the good practice of starting work early.

7. Conclusion

The studies colected showed positive relation when using test driven development on code and design quality, for both industry and academic developmers. However studies pointed several impediments on adoption and mistakes on using test driven development that means that developers are not always following test driven development correctly, it opens a lack to not get the most of the technique.

References

Aniche, M.F., Gerosa, M.A. (2010). *Most Common Mistakes in Test-Driven Development Practice: Results from an Online Survey with Developers*. Software Testing, Verification, and Validation Workshops (ICSTW), 2010 Third International Conference on.

Aniche, M. F., Gerosa, M.A. (2012). *Como a Prática de TDD Influencia o Projeto de Classes em Sistemas Orientados a Objetos: Padrões de Feedback para o Desenvolvedor*. Simpósio Brasileiro de Engenharia de Software (SBES 2012).

Causevic A., Sundmark D., Punnekkat S. (2011). *Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review* . Software Testing, Verification

and Validation (ICST), 2011 IEEE Fourth International Conference on.

Janzen, D., Saiedian, H. (2008). *Does Test-Driven Development Really Improve Software Design Quality?*. IEEE Software.

Buffardi K., Edwards, S. H. (2012). *Exploring influences on student adherence to test-driven development*. ITICSE12.