

INVESTIGATING SOFTWARE
DEVELOPMENT QUALITY THROUGH
IMPLEMENTATION OF STUDENT
TEAMWORK PROJECT SYSTEM

Michał Smigiel

May 2017

Contents

1	Introduction	5
1.1	Title	5
1.2	Background	5
1.3	Main Research Question	6
1.4	Justification and benefits	6
1.5	Feasibility	6
	1.5.1 Technical requirements	7
	1.5.2 Human Resource Considerations	7
1.6	Methodology	7
2	Literature Review	9
2.1	Frameworks overview	9
2.2	Testing Methodology	10
3	Appendix A: Project Plan	15
	List of references	17

Chapter 1

Introduction

1.1 Title

Project entitled

“INVESTIGATING SOFTWARE DEVELOPMENT QUALITY THROUGH
IMPLEMENTATION OF STUDENT TEAMWORK PROJECT SYSTEM”

conducted under the supervision of Dr. Matt Smith

1.2 Background

One of the biggest challenges during studies for both students and lecturers are group projects. Students often do not know how to properly log and report group meetings. This situation has a negative impact on the quality and final evaluation of their projects.

From the perspective of teachers it is extremely difficult to assess the involvement of students in the group project. (Presence at the meetings, creativity, performance of scheduled tasks). Control of the progress of the project is limited in most cases to oral reports. In the case of supervision of several projects this situation can lead to a lack of control over their progress.

The aim of this project is to address this issue by designing and developing a web based system for managing Student teamwork project.

The system will be built in such a way to allow:

- Planning meetings – creating agenda.
- Managing student’s presence.
- Recording minutes of meeting.

- Managing and controlling progress of agreed actions/tasks.

1.3 Main Research Question

To evaluate the use of open source tools and techniques for software validation and testing during developing an online system for group project meetings.

The complexity of current software projects forces developers to use several solutions for managing, testing and improving source code. The aim of this research is to find proper solutions, apply them and compare during development cycle.

1.4 Justification and benefits

The benefits of this project should be considered on two levels. From the system client's, user's and from the author's perspective.

System user:

- The system allows for better communication within the project between its members.
- The system will allow better communication between the team and the supervisor.
- The system will accelerate and unify planning and reporting meetings.
- The system will allow for simplifying the supervision of performance, dependability and reliability of the individual team members.

Researcher:

- Understanding and evaluation of various software solutions for managing, testing and improving coding quality.
- Improving knowledge of full stack web development with use of Symfony framework
- Improving knowledge of programming with use of OO PHP, HTML5, Java Script, CSS
- Improving knowledge of designing a relative database MySQL.

1.5 Feasibility

“A Feasibility Study is the first stage of the product or service development cycle. Its aim is to analyse the viability of a proposed project, product or service.” (Overton 2000)

1.5.1 Technical requirements

- For system user:
 - Device with installed internet browser and broadband connection to Internet
- For developer
 - Hardware
 - * PC with 1024X768 minimum screen resolution
 - Software
 - * IDE – PHP Storm
 - * Git repository hosting service – GitHub
 - * Local www server – Laragon

With the available budget, the failure of the author's computer may result in delay.

1.5.2 Human Resource Considerations

This project is an individual work and will be carried out by the author. The final result may be affected by many human factors, such as illness or being overwhelmed by the amount of work.

1.6 Methodology

The project was conducted in a way to satisfy Agile software development life cycle (Figure1) (blog.capterra.com 2016). Agile is a methodology created to describe software production cycle process when the designer, developer and tester are working together in recurring cycles. For an individual project one of the Agile techniques which can be used are Test Driven Development (TDD) or Behaviour Driven Development (BDD).

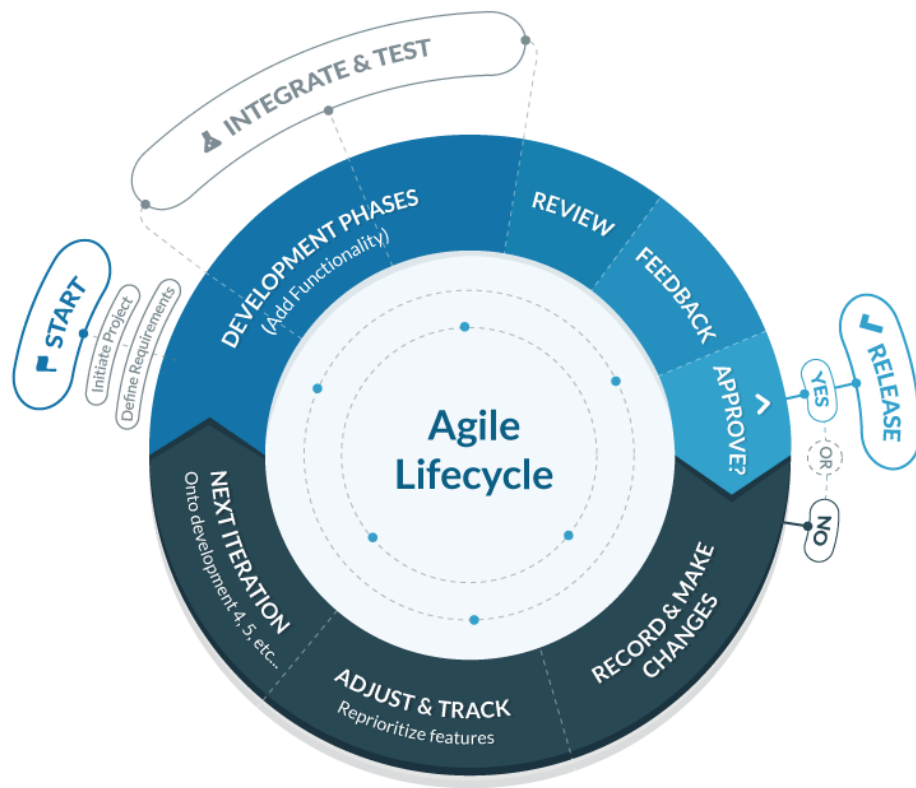


Figure 1.1: Agile Lifecycle.

Chapter 2

Literature Review

The modern programming techniques are accompanied by a suite of tools. To research them I divided this project into three parts. First part when the application is designed. During this process the developer should decide which framework to use. Second part, the development process when the application is written and tested, and the third part when the application's performance is analysed.

The process of performance analysing is called profiling. To use any of the profiling tools like Xdebug (<http://xdebug.org>) or XHPProof (<http://xhproof.ie>) the developer must know all performance issues and those can be discovered by benchmarking tools like Apache Bench (<http://bit.ly/apache-bench>) or Siege (<http://joeedog.org/siege-home>). (Lockhart 2015)

The foundation of this project is to focus on selecting and exploring PHP frameworks and developing the application with support of the testing tools so this literature review will be focused on selecting frameworks and testing tools.

2.1 Frameworks overview

Since Web 2.0 the border between web pages and standard applications has become blurred and they have a shared name web apps. Web apps not only provide information but more importantly provide services to the user. All these changes have significantly influenced the complexity of modern apps forcing developers to find solutions for it. (M. Salas-Zárate and Cuadrado 2015) The solution to this problem is to use frameworks.

PHP frameworks have several common features (Sklar 2016):

- Routing – translate user request and invoke specific function to send response

- Object-Related Mapping (ORM) – let the programmer use a database treating the database record as an object and providing set of functions to operate on the database.
- User management – Mechanisms for managing users, authentication and authorization.

Like most programming languages PHP have their own frameworks. Three most popular are Laravel, Symfony and Zend Framework.

Laravel

“Love beautiful code? We do too. The PHP Framework for web Artisans” (Laravel 2016)

The distinctive feature of this framework is cloud solution Forge, when the programmer can have access to preinstalled Laravel.

Zend Framework

“Professional PHP packages ready for PHP 7 Focused on Simplicity, Reusability, and Performance” (Zend 2016)

Zend Framework is sponsored by Zend and Rouge Wave Company with contributions in many features with Google and Microsoft.

Symfony

“Symfony is a set of reusable PHP components... and a PHP framework for web projects” (Symfony 2016)

It is an Open Source PHP framework. Created by SesionLab and published in 2005 on MIT Open Licence. It is worth a mention the Symfony components are base for many other open source projects like Drupal, phpBB, Laravel, Joomla! and Composer.

The common feature of these projects is that they are made up of independent modules, packages or libraries. In order to manage them use a separate tool Composer. Composer (Getcomposer.org 2016) is a package manager on a per-project base. Its two major tasks are:

1. It enables you to declare the libraries you depend on.
2. It finds out which version of which packages can and need to be installed.

2.2 Testing Methodology

In typical types of application life cycle, testing was a separated part of the cycle often conducted by a separate team. Because of changing business requirements the developing life cycle process had to change as well. Testing the application entered into a new dimension where testing is parallel to coding. In 2001 a group of seventeen programmers published a set of rules for new methodology in

software development, Agile was defined (Agile 2016). To fulfil Agile methodology, programmers started with Test-Driven Development (TDD). TDD means you writing test before you start write application code (Lockhart 2015)

Kent Beck in his book define two simple rules for Test-Driven Development (Becks 2002):

1. Don't write a line of new code unless you have a failing automated test.
2. Eliminate duplication.

This mean first you write the test then you write just enough code to past this test, then you refactoring this code. This process is repeated many times. In TDD programmers are using PHPUnit programmer-oriented testing framework for PHP. It is an instance of the xUnit architecture for unit testing frameworks (Phpunit.de 2016)

PHPUnit prints results of the test in red colour when it fails and in green when the test is passed. This is why TDD cycle is called red/green/refactor. See figure 1 (Kondas 2016) below.

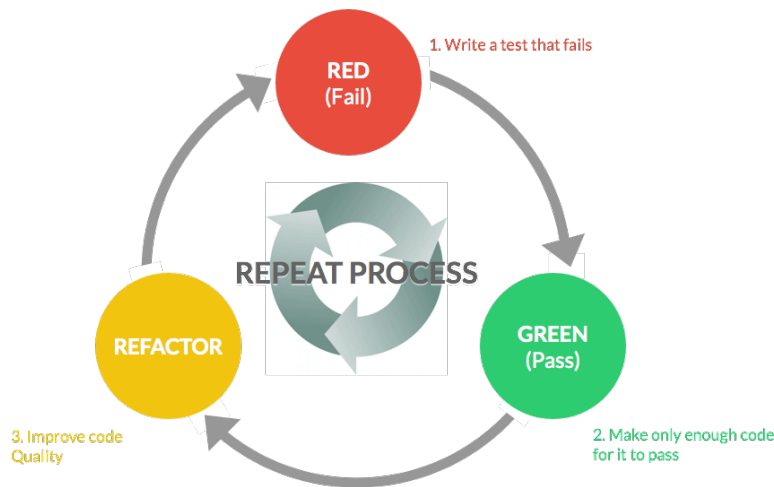


Figure 2.1: Red, Green and Refacto.

PHPUnit test should be used to test every public class, method and function. Testing of each of these elements is conducted in isolation from other methods and functions. The outcome of TDD using Unit test is a granular and decoupled set of classes and functions. (Lockhart 2015)

Despite the use of TDD, it is a big step forward in software production methodology, the problem of communication between team members is still unsolved because there is no language in which business team members or customers can

directly communicate with the programmers. PHPUnit is too technical for this. (Laracasts 2016)

Another testing methodology in software development is Behaviour-Driven Development (BDD). The programmer writes a story to describe the behaviour of an application. BDD has two types, SpecBDD and StoryBDD. [15]

SpecBDD technique uses PHPSpec framework. (Phpspec.net 2016) SpecBDD is derived from Test-Driven Development, the different between SpecBDD and xUnit testing is a language, SpecBDD uses human-friendly language that makes it more focused on behaviour and code design. StoryBDD describes application business logic. Josh Lockhart in his book gives a simple and clean example to spot differences between StoryBDD and SpecBDD scope.

“StoryBDD resembles something a project manager would write (e.g., “this should generate and email me a report”). A SpecBDD test reassembles something a developer would write (e.g., “this class method should receive an array of data and write it to this PDF file”)” (Lockhart 2015)

Behat framework is used for StoryBDD methodology. “A php framework for auto testing your business expectations” (Behat.org 2016) Behat is a tool build of Symfony components. Behat scenarios are written in Gherkin which is a Business Readable, Domain Specific Language. Language that describes behaviour without describing implementation. The figure 2 [gherkin_code] below presents snippets of Gherkin code.

```
1: Feature: Some terse yet descriptive text of what is desired
2:   Textual description of the business value of this feature
3:   Business rules that govern the scope of the feature
4:   Any additional information that will make the feature easier to understand
5:
6:   Scenario: Some determinable business situation
7:     Given some precondition
8:       And some other precondition
9:     When some action by the actor
10:      And some other action
11:      And yet another action
12:     Then some testable outcome is achieved
13:       And something else we can check happens too
14:
15:   Scenario: A different situation
16:     ...
```

Figure 2.2: Example of Gherkin code.

Chelimsky in his book presents a testing system assembled from RSpec and Cucumber. (Chelimsky and others 2010) Two BDD framework for Ruby, in PHP respectively PHPSpec and Behat. He describe BDD cycle when at each green test in RSpec cycle the programmer checks the Cucumber cycle. If the outside cycle

is red the test result should lead back to the inner cycle, where the programmer refactors the code till it passes. When result is green the programmer repeats Cucumber cycle by writing another failing test. See figure 3 (Chelimsky and others 2010) below.

In this example Cucumber focuses on the business logic and RSpec focuses on single function behaviour. Both these tools can replace each other's role or work separately. Conclusion Using frameworks and components for PHP development the programmer can be sure that every component of this framework is well documented and tested, and this makes his work faster and free of errors. My personal choice is Symfony 3 framework with the Composer as a package manager. For testing purpose I will create a system proposed by Chelimsky. Story Driven Development with use of Behave framework and SpecBDD with use of PHPSpec. Both testing frameworks installation is supported by Composer and their website documentation confirms compatibility with Symfony.

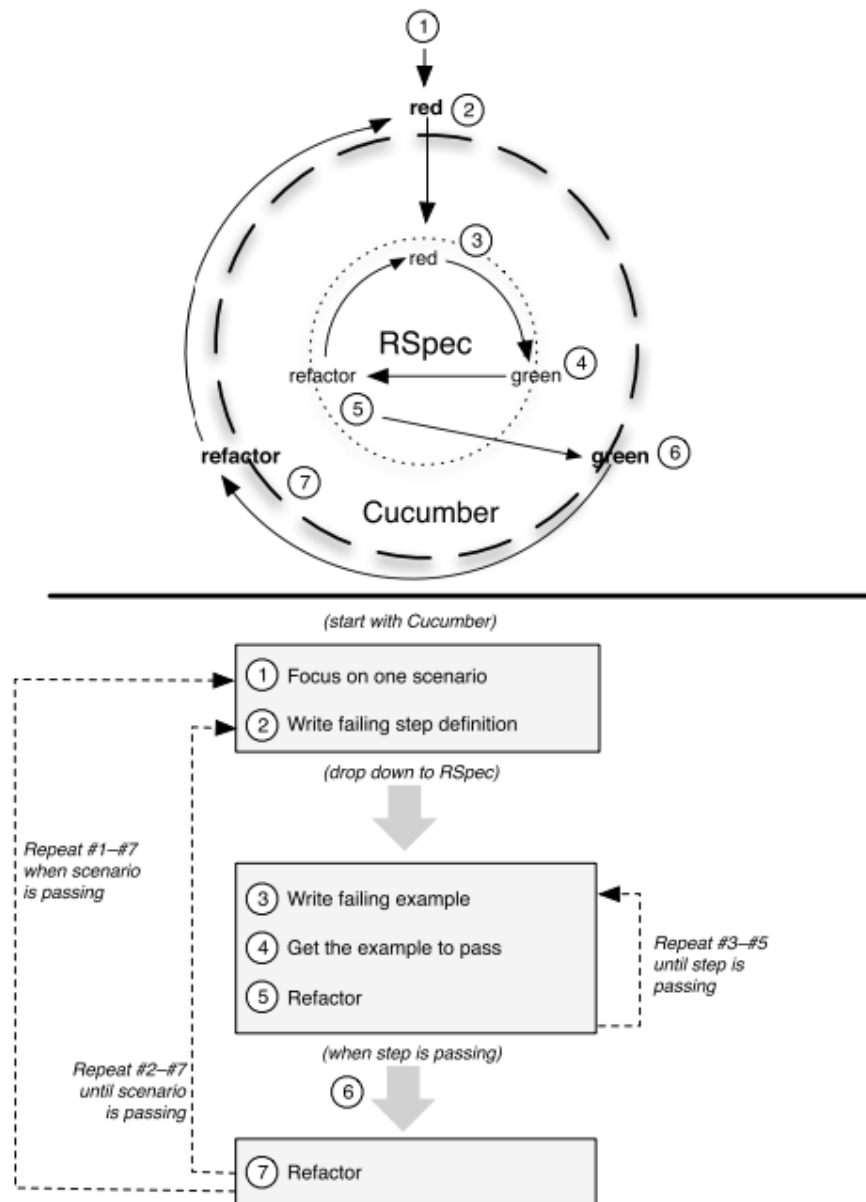


Figure 2.3: BDD Life Cycle.

Chapter 3

Appendix A: Project Plan

The figure below shown the Gannt chart for our project plan.

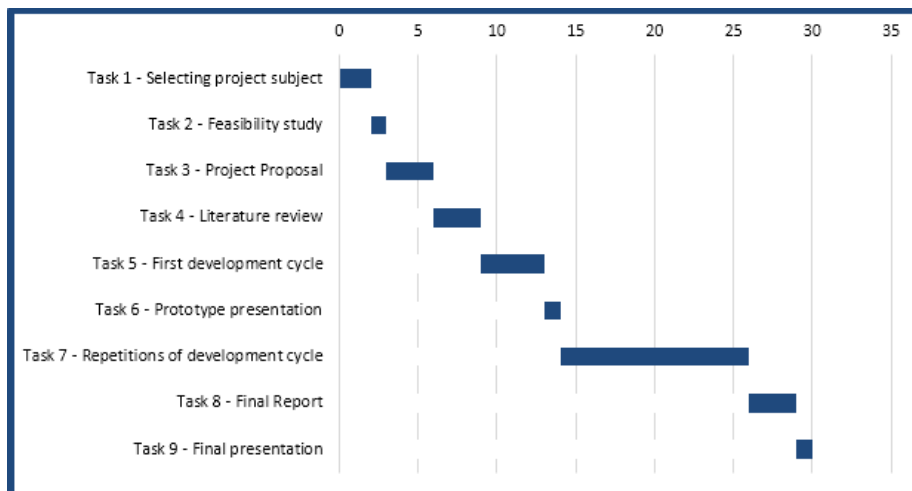


Figure 3.1: Gantt Chart.

The main phases of the project were:

1. Selecting project subject
2. Feasibility study
3. Project Proposal
4. Literature review
5. First development cycle
6. Prototype presentation
7. Repetitions of development cycle
8. Final Report

9. Final presentation

List of references

- Agile. 2016. “Agile Manifesto,” November. <https://agilemanifesto.org/>.
- Becks, Kent. 2002. *Test Driven Development: By Example*. Addison-Wesley Professional.
- Behat.org. 2016. “Behat — a Php Framework for Autotesting Your Business Expectations.” November. <http://behat.org/en/latest/index.html>.
- blog.capterra.com. 2016. “Agile Lifecycle.” November. <http://blog.capterra.com>.
- Chelimsky, David, and others. 2010. *The RSpec Book: Behaviour-Driven Development with Rspec, Cucumber, and Friends., the Bdd Cycle*. Pragmatic Bookshelf.
- Getcomposer.org. 2016. “Introduction - Composer.” November. <https://getcomposer.org/doc/00-intro.md>.
- Kondas, Arkadiusz. 2016. “Red Green Refactor.” November. http://itcraftsman.pl/wp-content/uploads/2016/04/red_green_refactor.png.
- Laracasts. 2016. “When to Use Phpspec, Behat and Phpunit.” November. <http://www.slideshare.net/joshuaswarren/phpspec-behat-two-testing-tools-that-write-code-for-you-phptek-edition>.
- Laravel. 2016. “Laravel.” November. <https://laravel.com/>.
- Lockhart, J. 2015. *Modern Php: New Features and Good Practices*. O’Reilly Media. <https://books.google.ie/books?id=rnSpBgAAQBAJ>.
- M. Salas-Zárate, R. Valencia-García, G. Alor-Hernández, and J. López Cuadrado. 2015. “Analyzing Best Practices on Web Development Frameworks: The Lift Approach.” *Science of Computer Programming* 102: 1–19.
- Overton, R. 2000. *Feasibility Studies Made Simple*. Australian Business Basics Series. Martin Management Services in conjunction with Parramatta Business Centre. <https://books.google.ie/books?id=SW-EAAAACAAJ>.
- Phpspec.net. 2016. “Introduction — Phpspec 3.x Documentation.” November.

<http://www.phpspec.net/en/stable/manual/introduction.html>.

Phpunit.de. 2016. “PHPUnit – the Php Testing Framework.” November. <https://phpunit.de>.

Sklar, David. 2016. *Learning Php: A Gentle Introduction to the Web’s Most Popular Language*. O’Reilly Media, Inc., USA.

Symfony. 2016. “Symfony.” November. <http://symfony.com/>.

Zend, a Rogue Wave Company. 2016. “Zend Framework.” November. <https://framework.zend.com/>.