Using Behavioral Driven Development (BDD) in Capstone Design Projects

Dr. Ana Elisa E. Goulart, Texas A&M University

Ana Elisa Goulart received a bachelor's degree in electrical engineering from the Federal School of Engineering of Itajuba (EFEI), in Brazil. While working in the industry, she received a M. Sc. degree in Information Systems Management from the Pontificial Catholic University of Campinas, in 1997. She moved to the United States in 1997 where she earned a M. Sc. in Computer Engineering at North Carolina State University, Raleigh, NC; followed by a Ph.D. in Electrical and Computer Engineering at Georgia Tech, Atlanta, GA, in 2005. She is currently an Associate Professor in the Electronics Systems Engineering Technology program at Texas A&M University, in College Station, TX. Her research interests include protocols for real-time voice and video communications and their performance, IP-based emergency communications, last-mile communication links for the SmartGrid, rural telecommunications, and behavior-driven development.

Using Behavioral Driven Development (BDD) in a Capstone Design Project

Abstract

To assist undergraduate engineering students define the functional requirements and acceptance criteria for their capstone design project, behavioral driven development (BDD) was applied for the first time with a team of three students. This team designed a complex system for health care monitoring consisting of hardware and software elements. This case study is presented in this paper, which includes an overview of BDD, how it was implemented in a two-semester Capstone course sequence, and the lessons learned from this new approach. This paper is written from the point of view of faculty advisor and industry sponsor, who closely followed the use of the BDD approach by the student team.

1 - Introduction

The Requirements Phase is the most unstable lifecycle component of a product. Many more assumptions are made about a product at the requirements phase than at later stages. However, only at later stages the features become better understood. This volatile aspect is a leading cause of ambiguous, incomplete, or logically inconsistent feature specification. Engineers design and implement based upon these weak definitions. This propagates requirement decisions and errors into later stages. Unfortunately, fixing the errors at later stages costs more, sometimes exponentially more, as shown in Figure 1.

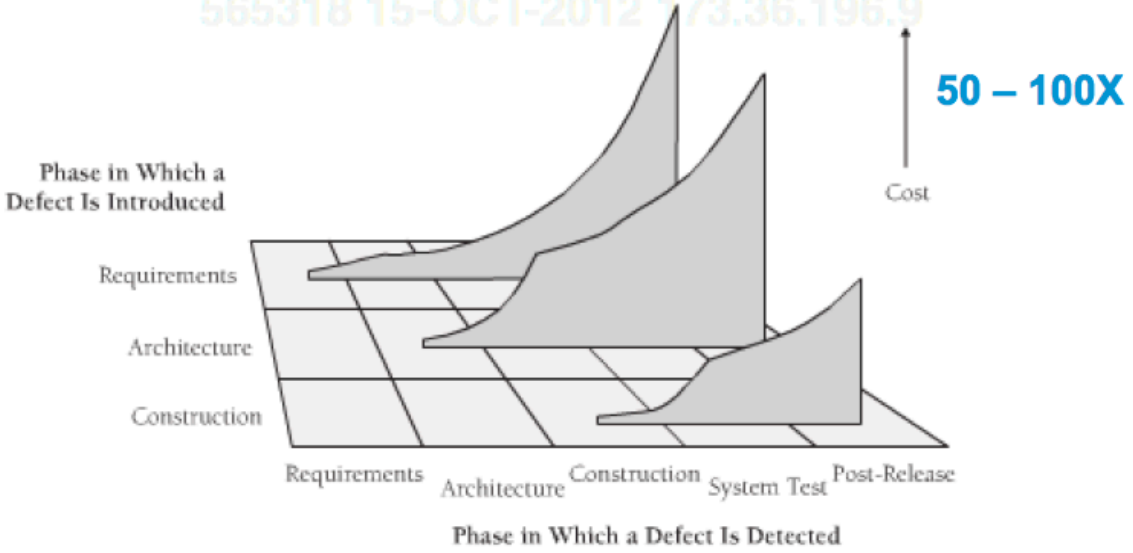


Figure 1 - According to McConnell, the cost to fix defects increases significantly when defects are detected at later phases in the project [1].

Behavior Driven Development (BDD) is a new way to address this problem. By engaging the entire design team during requirements definition, the BDD discussion process clarifies expected behavior of a product's features. Using plain, non-technical language of the business domain, the team writes scenarios that specify how the end user will use their final product. By defining these user-centric scenarios, the product's features can be clearly specified.

Given the benefits of the BDD approach for product development, can a team of undergraduate students apply BDD in their Capstone Design project? Would the BDD approach help them specify the features and scenarios of their final prototype? Would the BDD scenarios help students communicate with their industry sponsor?

This paper will address the above questions, and present an application of BDD in a Capstone Project that began in Spring 2013. The project in this case study is a health monitoring system. It involves both software and hardware development, for it integrates an IP-based emergency calling system, an Android application, and a heart-rate sensor. The project's goal is to provide health and emergency personnel with the end user's real-time vital signs during an emergency call.

The students have included BDD scenarios as one of their deliverables, and have worked closely with the sponsor on these scenarios. This paper shows an evaluation, from the point of view of faculty advisor and industry sponsor, of the use of BDD as a project management tool that can be taught to undergraduate engineering students. It also discusses the impact of the BDD approach in the Requirements and Testing phases of the current Capstone project.

The remainder of this paper includes an introduction to the BDD approach (Section 2), followed by an overview of the Capstone project in Section 3, with details on how BDD was used in the planning, design and testing phases of the project. Then, in Section 4 we conclude this paper with a few learned lessons from this case study.

2 - What is Behavioral Driven Development (BDD)?

In this section, the BDD concept is explained, showing how it is used for both software and hardware design projects. We also discuss related approaches that engineering instructors use in Capstone projects.

2.1 – Overview of BDD

In software engineering, test-driven development (TDD) is a well-known methodology in which the tests are defined during the design phase. Developers and testers work together to define the tests early in the design. Testcases or test scenarios are coded and usually run in an automated fashion. The research in [2] has shown a 40 percent reduction in defects for a project in which developers used the TDD approach.

Other studies confirm that most software defects come from early phases in the project, such as the Requirements phase and Design phase. McConnell [1] explains that the defect injection rate can be about 56 percent in the Requirements phase, and 27 percent in the Design phase. The problem is that usually these defects are only found at later stages in the project, sometimes when the final product is available to the end user. If we have a technique that can detect the defects early in the design, the cost to fix the defects will be reduced. If defects are not detected, they are shipped to the customer. Thus, quality control and cost reduction are the main motivations for the BDD approach.

The BDD approach combines the concept of TDD with the idea of writing test scenarios during the Requirements phase, so that requirements and functional specification of the product can be well defined by developers and testers. For instance, if a requirement is vague and nobody knows how to test it, that means that this requirement needs to be further discussed with architects and system engineers. Differently than TDD, the BDD test scenarios are written in a high-level or natural language, usually plain English language (or any other idiom), which facilitates the discussion with all project stakeholders, including marketing and business people. The BDD test cases represent scenarios that show how the end user would use the final product. For instance, the scenarios can be written as a “should” statement, or a “Given, Then, When” format [3].

Figure 2 shows an example of a default BDD scenario for the Capstone project described in this case study. The Capstone project consists of a hardware component (wristband sensor) and a software component (cellphone application), as shown in Scenario 0 in Figure 2. This scenario shows a high level definition of the requirements, similar to the level of functional testing.

Additionally, although BDD is mostly used in projects focused on software development, note that the students have used BDD to specify the hardware requirements as well. For instance, we can see "Scenario 0: HelpAlert system default behavior" in Figure 1, where it says "It should collect data samples from wristband sensors" and "It should have an LED that displays Green to indicate normal operation and good battery", which are mainly hardware features.

Scenario 0: HelpAlert system default behavior
Wristband Sensor:
    It should collect data samples from wristband sensors
    It should send collected data to a paired nearby cellphone via Bluetooth
    It should have an LED that displays Green (to indicate normal operation, and good battery)
    It should have an LED that displays that is transmitting to cell phone
Cellphone App:
    It should receive and store the data sent by the wristband
    It should periodically update a webserver (default: each hour)

Figure 2 - Example BDD scenario used in the Capstone project. This scenario was defined early in the Spring of 2013, when the students were beginning their Capstone design course.

2.2 – Applications of BDD in industry and research

In software projects, the BDD scenarios can be automated. There are tools or frameworks that can convert the natural language testcases into actual code that run the test. An example of such tool is Cucumber [4], which uses Gherkin as an interpreter that accepts multiple idioms in plain text. Each statement in the scenario is converted to a "step definition" which will have actual code that executes the parts of the project being tested. For instance, Cucumber uses the "Ruby" programming language to code the steps. A comparison of different BDD frameworks that use other programming languages such as "Java" is presented in [3].

There are also other BDD applications at the research level that combine hardware and software projects [5, 6], including automated ways to convert the natural language testcases into step definitions for both software and embedded hardware products. Some keywords in the BDD scenarios can automatically call methods or functions in the code to run the tests. In the special case of projects focused mostly on hardware, Diepenbeck *et al.* [6] showed that the BDD approach can be used for hardware modeling as they defined BDD scenarios early in their design flow.

Industry has also used BDD testcases as "acceptance criteria" in software projects managed using the Agile framework [7]. For instance, managers and product owners can use the BDD scenarios to evaluate when a feature is completed. If the scenario runs and passes the test, then the feature is accepted as completed. Sometimes this is called "definition of ready". It helps non-technical personnel to evaluate if the software developer completed a given requirement of the project.

Our industry sponsor has been using the BDD approach in some pilot software projects, and also in research projects with our university [8]. For instance, they have used BDD to test a new web-based graphical user interface (GUI) of one of their products. Even before the web pages were developed, the BDD scenarios were written. Since the web GUI can be tested using automated tools, developers and testers worked together in getting the testcases ready for testing the new application before the final code was completed. This process helped detect ambiguous requirements and missing specifications early in the project.

To summarize the main reasons why the BDD approach is being used in industry, research projects, and in the case study addressed in this paper, Figure 3 shows a diagram depicting the benefits BDD brings to a project. Among them, BDD as a communication tool that helps clarify objectives and specification, and helps explain limits of what work is expected shows that it can be applied beyond projects where software development has a central focus, but on hardware projects as well.

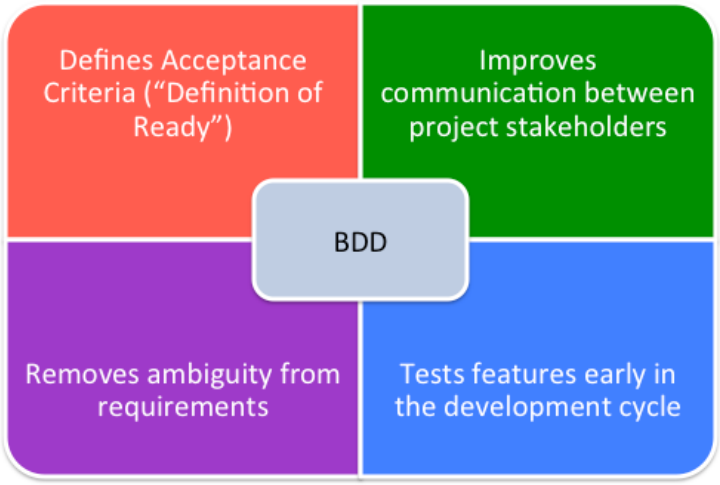


Figure 3 - A summary of the main advantages of using behavior driven development (BDD).

2.3 – Related BDD work in Capstone Projects

To the best of our knowledge, the use of the BDD approach is novel in undergraduate education. However, the concept of improving the quality of the Requirements phase in the Capstone courses has always been an issue. Several engineer instructors advocate the idea that the students communicate the requirements and functional specification clearly with industry sponsor and faculty advisors. As an example, the work in [9] shows a detailed approach, including assessments, that emphasizes the requirements phase of a software project for computer science students.

3 - Case Study: "HelpAlert" Capstone Project

As a suggestion to the undergraduate student team, the sponsor proposed that the students work on BDD scenarios at the beginning of their first semester in Capstone, which in our university is composed of two semesters. The two-semester capstone projects require groups of three to four students to form teams (startup ventures), identify a problem or opportunity, secure an external sponsor, and select a technical advisor (faculty member). The two-semester project is very sequential, as the work is divided in the two semesters as follows:

- **Semester 1**: the teams learn and use project management techniques and tools to plan the project and complete an initial design.
- **Semester 2**: the teams implement, test and validate, document, and present a fully functional prototype that meets all performance requirements.

All projects are product-oriented and generally include embedded hardware/software development. System integration is very important too, for the projects usually include sensors, signal conditioning, algorithm development, wireless/wired communications, and specialized packaging of the prototype. In this case study, our student team worked on their Capstone project during Spring and Fall of 2013. Their capstone project is called HelpAlert, which will be described next, followed by how BDD was used throughout the project.

3.1 – HelpAlert Conceptual Design

The student team and sponsor agreed on a project focused on the "Internet of Things", in which devices or machines communicate automatically to other machines, over the Internet, for different purposes such as monitoring or data collection. The students selected the topic of health care monitoring, and they targeted the critical first moments of an emergency. Usually, there is no information for doctors or first responders to access before arriving on the emergency scene. By having access to a person's vital signs measurements, first responders can be better prepared for a medical emergency.

Thus, the proposed system, as shown in the diagram in Figure 3, incorporates heart rate monitoring and IP-based emergency calling system, wristband with hear rate sensor, cellular phone application (i.e., Android app), and a web server. The system collects, stores, and displays sensor information, as well as makes emergency calls or regular doctor calls. It is supposed to be a personal use system, with rechargeable battery, that can be used and accessed 24 hours a day, 7 days a week. The heart rate sensor is located on a wristband (Figure 4), which will be worn on the user's wrist. The sensor periodically measures the user's heart rate, and it must be in direct contact with the user's skin. The cellular phone should remain within close proximity to the user. Thus, this product will allow users be monitored, but at the same time to have mobility and pursue their daily schedule.

During normal operation, sensor data is periodically transmitted from the wristband to a web server. The wristband, shown in Figure 5 in a dedicated packaging designed by the students, also has two physical buttons that will enable a patient to contact their medical doctor or 9-1-1 emergency personnel. When activated, these buttons will cause the Android device to initiate a phone call with the doctor or emergency personnel utilizing either a regular phone call or an IP-based call, using the Session Initiation Protocol (SIP). The latter case allows video calls between patient and doctor, which the students showed in their final demonstration.

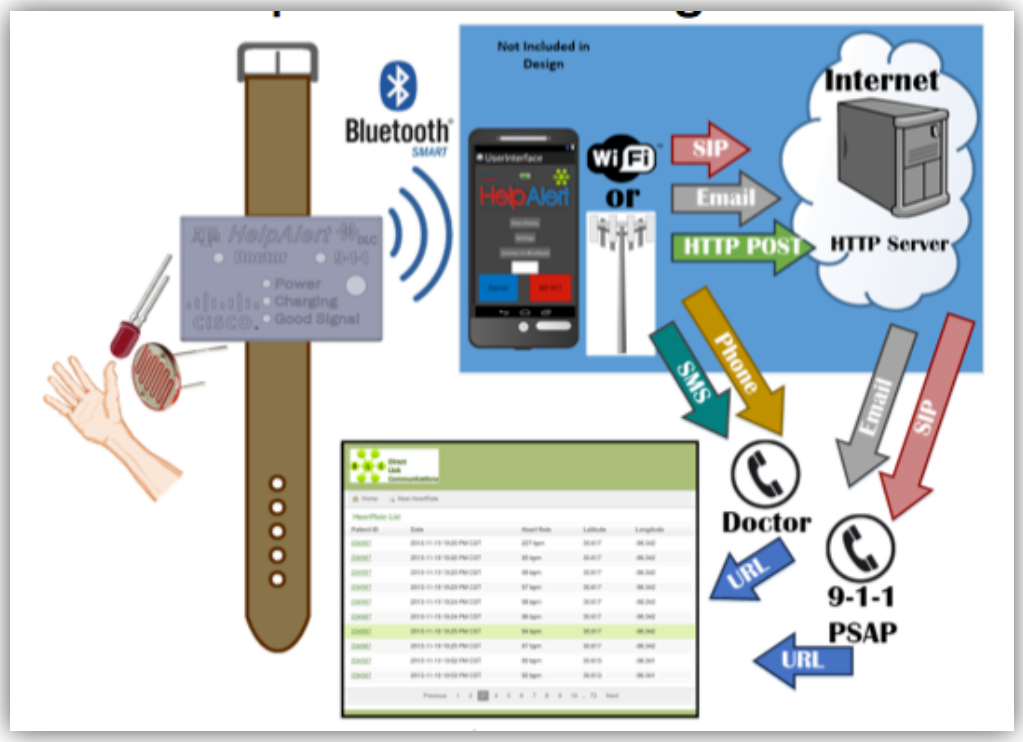
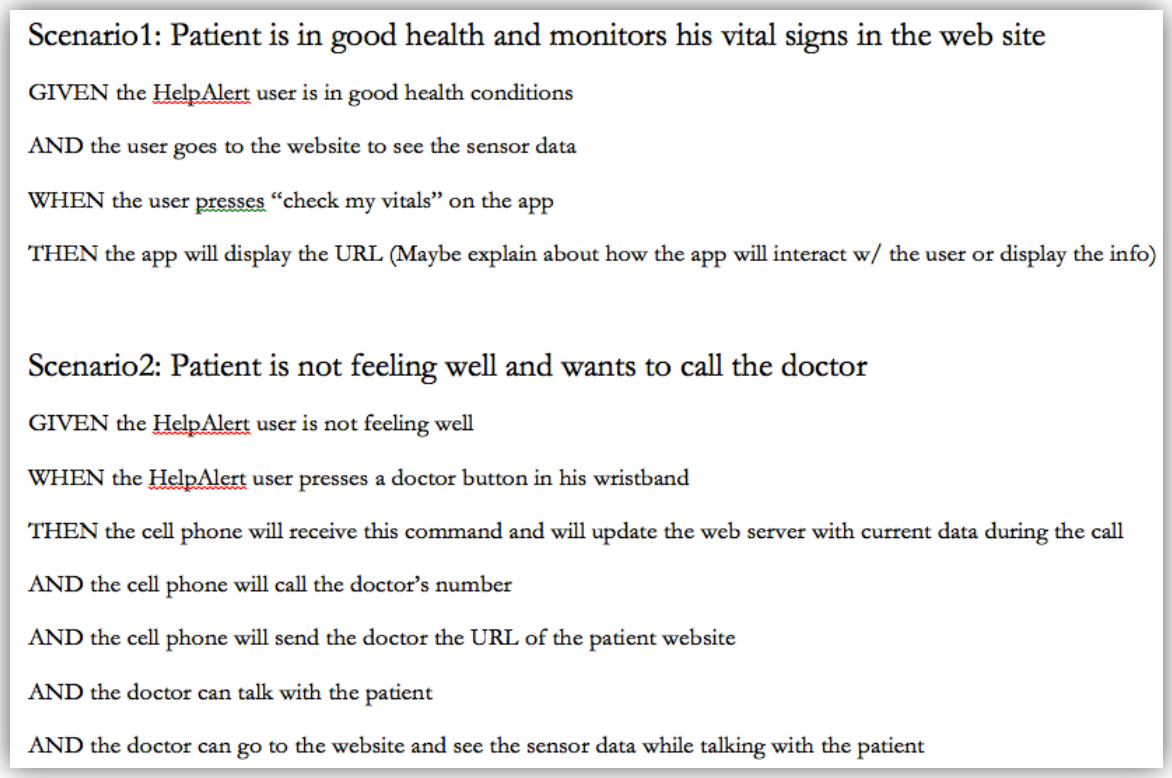


Figure 4 – Capstone design project case study – "HelpAlert" - and its conceptual diagram.

Figure 5 - Wristband prototype developed by our team of undergraduate students.

3.2 – BDD in the Planning Phase

The default scenario previously shown in Figure 2 (Section 2.1) was first written in late January of 2013, and sent to the sponsor for review. By mid February, the sponsor approved multiple scenarios, after a few meetings and iterations with the students. The sponsor did not require the use of automated tools to convert the BDD scenarios into actual code for the testcase, for the project involved hardware and software. The sponsor's main idea was to use BDD in the Requirements phase, update it as needed during the design and implementation phase, and run all the scenarios during the testing phase. Therefore, the BDD scenarios were constantly used throughout the two semesters of the project. A sample of other scenarios defined early in the project is shown in Figure 6, which includes feedback from the sponsor to the students (inside parentheses in the last line of Scenario 1). Additionally, negative scenarios to handle problems and alert users of problems (such as lost of connectivity from wristband to cellular phone) were part of the BDD scenarios.



Scenario1: Patient is in good health and monitors his vital signs in the web site

GIVEN the HelpAlert user is in good health conditions

AND the user goes to the website to see the sensor data

WHEN the user presses "check my vitals" on the app

THEN the app will display the URL (Maybe explain about how the app will interact w/ the user or display the info)

Scenario2: Patient is not feeling well and wants to call the doctor

GIVEN the HelpAlert user is not feeling well

WHEN the HelpAlert user presses a doctor button in his wristband

THEN the cell phone will receive this command and will update the web server with current data during the call

AND the cell phone will call the doctor's number

AND the cell phone will send the doctor the URL of the patient website

AND the doctor can talk with the patient

AND the doctor can go to the website and see the sensor data while talking with the patient

Figure 6 - Sample of scenarios written in BDD's syntax of "Given, When, Then", which means that given a context or pre-condition, when an action happens, the system will perform something.

3.3 – BDD in the Design Phase

The design and testing phases happen most in the second semester of our capstone two-semester course. The main deliverables to the sponsor are submitted in the second semester. Therefore, to enforce that students keep the BDD scenarios updated, the

sponsor required an official document with all the BDD scenarios to be submitted in the Fall semester, as shown in the deliverables timeline of Figure 7.

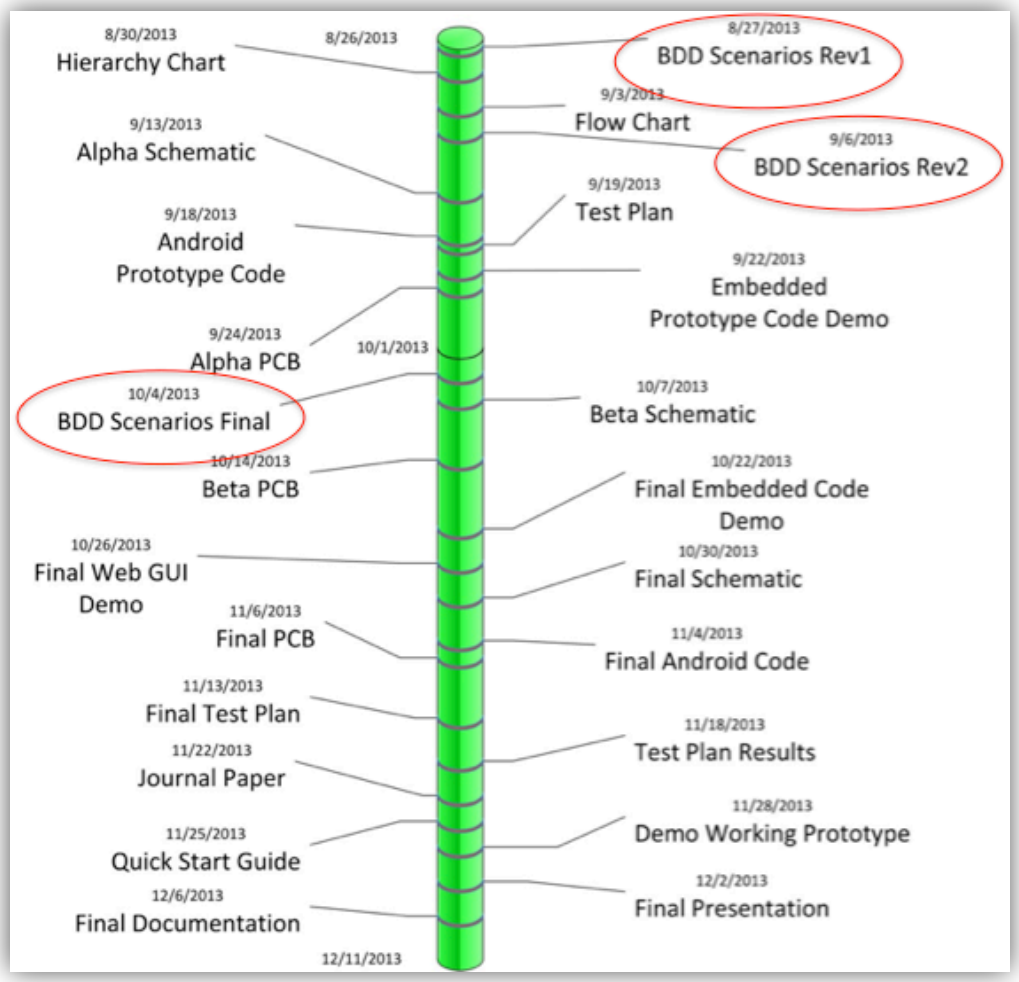


Figure 7 - Students proposed a deliverables plan that included two preliminary versions of the BDD scenarios and the final BDD scenarios.

Also on Figure 7, note that the sponsor required several preliminary demonstrations, such as:

- Android prototype demo code
- Embedded prototype demo code
- Final Embedded code demo
- Final Android code demo prototype

The combination of preliminary code demonstrations and BDD scenario updates was very important to the flow of the project. It helped students make small-steps in their project and demonstrate their work to the sponsor for feedback. Moreover, it helped students start testing their project early in the design, and not in the final weeks of the second semester, when the final test results are due. Early testing ensured that defects were detected early in the semester. As supporting evidence, we can only say that the project was delivered on time and within scope of the main BDD scenarios.

3.4 – BDD in the Testing Phase

Traditionally, student teams are required to create a test matrix and design their test plan in the second semester of their capstone course. The initial test matrix for this project is shown in Table 1, which shows in the first columns the main features of the project.

Table 1: Test Matrix

	Battery power	Session Initiation Protocol	9-1-1 Mode	HTTP	Groovy	Android Application	Bluetooth	LED feedback	GPS	Doctor Mode	Default Mode
Run HelpAlert™ on battery power	X						X	X			X
GPS is available					X	X			X		
GPS is not available					X	X			X		
Make Doctor call					X	X	X			X	
Make SIP 9-1-1 call		X	X		X	X	X		X		
Cell phone Default mode					X	X	X				
Pair wristband to phone						X	X				
Remote connect to web server				X	X	X					
Make Doctor call but no answer						X	X			X	
Test 20' Bluetooth range						X	X	X			X
App notification of wristband battery status	X					X	X	X			
Wristband is turned off	X	X				X					X
Cellphone configuration		X	X			X			X	X	
Cellphone memory storage				X	X	X	X		X		
Security				X	X	X	X				
Run 12hr runtime	X							X			

Initially, the test plan had no connection to the BDD scenarios defined in the Requirements phase. Although many of the tests included some of the BDD scenarios, the test plan did not cover all scenarios. Therefore, sponsor and faculty advisor guided the students to map each BDD scenario in their test plan. Their final test plan contained not only a detailed procedure of the tests, but also a reference to the BDD scenario it corresponded to, as shown in the example in Figure 8.

Students also documented each scenario by video recording their test procedure. Brief video clips were created, organized by BDD scenarios, which the students used in their final demonstration. Thus, their final demonstration was very straightforward and sponsor and faculty advisor could clearly "accept" the features of the project as completed. In other words, it helped them determine if the students met the required acceptance criteria for the project. In Figure 9, we can see the BDD scenario checklist given to sponsor at their final demonstration.

1.2.6 Remote connect to web server

Reflects: Scenario 1 of the Behavioral Driven Development

Description: The HelpAlert™ System will be tested to insure the patient can monitor his or her vitals remotely. The HelpAlert™ System will be tested to make sure technicians can connect to the server remotely to perform maintenance and also the URL sent out during the call is able to locate the right information.

Procedure:

- Log on to the web server from any computer
- Log off
- Press "Check my vitals" on cellphone
- Check heart rate
- Log off

Pass	Fail
• Server displays the correct users heart rate • Sign in works • Log off works	• The URL links does not work • Sign in fails • Log off fails • Server not accessible

Results: Passed and made a little user friendly with the button saying "Data History"

Figure 8 - A section of the student's test plan showing the mapping to a BDD scenario.

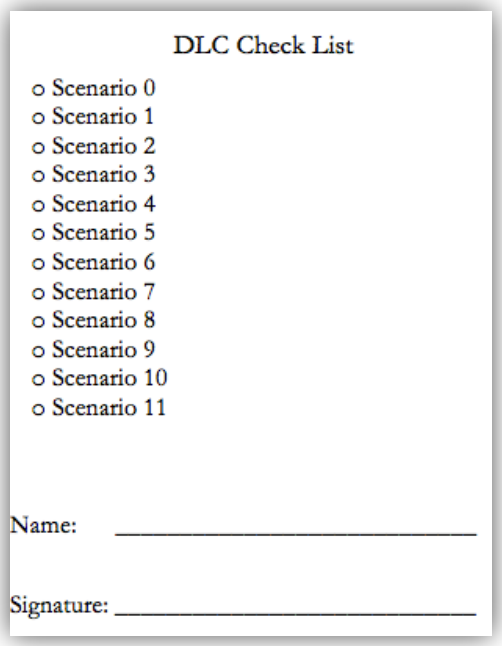


Figure 9 - On how the BDD scenarios were used in the final demonstration: for each scenario, a detailed video demonstration was presented to sponsor.

4 – Conclusions and Lessons Learned

This paper introduced the concept of Behavior Driven Development (BDD) in a two-semester Capstone Design course. It also included samples of how the BDD scenarios were developed and used throughout the project. Although no automated tool was used to map the scenarios to actual "step" procedures to test the project, the updated documentation of BDD scenarios was part of the project's deliverables to the sponsor, and used in the student's test plan not only to design the tests but also to demonstrate to the sponsor that all acceptance criteria were met.

In summary, these are the main lessons learned during this case study:

- **Mentor the students so they can understand the purpose of the BDD approach**: For the students, BDD was a new approach that initially they did not understand very well and thought that the advisor and sponsor were adding more work to their Capstone project. However, as the BDD scenarios were defined early in the first semester, they could see that it facilitated the communication with the sponsor, who was in a remote location and not available to attend all their meetings. Sponsor promptly edited or questioned each BDD scenario, and asked about new user scenarios.

- **Require students to document and periodically revise BDD Scenarios:** After the BDD scenarios were created in the first semester, sponsor required that students had three deliverables to update their BDD document in the second semester of the project (Figure 7). This ensured that students used the BDD scenarios often. It was a document constantly referenced in our meetings, such as when we reviewed the hierarchy and software flow chart. By having the scenarios defined in a high level, the students could now specify how they would implement each step in their design, in a more detailed level, and could communicate their design choices very clearly with sponsor and advisor.

- **Add in your Capstone course elements of the Agile project management model, such as acceptance criteria and early demonstrations.** Although our Capstone course uses the traditional and sequential "Waterfall" project management approach, by incorporating "Agile" management elements [7] such as acceptance criteria and BDD user scenarios, our students had to provide periodic demonstrations of preliminary versions of their project. In these demonstrations, as well as in the final one, they would always reference the demonstrations to a BDD scenario. We also believe that the students began the testing of each scenario early in the second semester, as they prepared for these demonstrations to the sponsor.

Capstone courses can evolve to use new tools that industry has introduced. In this case study, BDD helped a group of undergraduate students to have a successful Capstone project. Moreover, it taught them a new tool, which is still in its infancy, but may allow them to be open to new development approaches and bring innovation to their future employers.

5 - References

1. McConnell, S., "Code Complete," 2nd edition, Microsoft Press, 2004.

2. Williams, L., Maximilien, E. M., Vouk, M., "Test-Driven Development as a Defect-Reduction Practice," in the Proceedings of the 14th International Symposium on Software Reliability Engineering (ISSRE' 03), pp. 34, 2003.

3. Solís, C. and Wang, X., "A Study of the Characteristics of Behaviour Driven Development," in the Proceedings of the 37th EUROMICRO Conference on Software Engineering and Advanced Applications, 2011.

4. Wynne, M and Hellesoy, A., "The Cucumber Book – Behaviour-Driven Development for Testers and Developers," Pragmatic Programmers LLC, January 2012.

5. Soeken, M., Wille, R., and Drechsler, R., "Assisted Behavior Driven Development Using Natural Language Processing," Lecture Notes in Computer Science, vol. 7304, pp. 269-287, 2012.

6. Diepenbeck, M., Soeken, M., Große, D., and Drechsler, R., "Behavior Driven Development for Circuit Design and Verification," in the Proceedings of the IEEE International Conference on High Level Design Validation and Test Workshop, November 2012.

7. Crispin, L., and Gregory, J., "Agile Testing – A Practical Guide for Testers and Agile Teams," Addison-Wesley.

8. Goulart, A., Corti, C., and Hawkes, M., "A College-Industry Research Partnership on Software Development for Undergraduate Students," in the Proceedings of the ASEE National Conference, 2011.

9. Vijayalakshmi, M., Padmashree D., and Joshi G. H., "An innovative approach to problem identification and design for capstone projects," in the Proceedings of the IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA), 2012.