# *Development Project II (420-E63-HR)*

# *Assignment 9 – Final Retrospective: Lessons Learned*

Date assigned: Friday, April 4, 2018

Date due: **Friday, April 11, 2018, 11:50PM**

**Learning Objectives**

Upon successful completion of this assignment, students will be able to:

* Identify the lessons learned throughout the development of the system
* Complete a self-evaluation of their performance on a project
* Evaluate the performance of their team mates

**Students are required to pass this assignment in order to pass the course.** Students who complete and fail this assignment before late marks are deducted may correct and resubmit the assignment for a maximum mark of 60% less the late marks on the original assignment.

To do:

This assignment is to be completed as an individual activity. Please update and submit to Moodle. It will be used to record the answers to the assignment.

**Part A - Lessons Learned**

The purpose of lessons learned is to identify any insights gained during a project that can be usefully applied on future projects. Experiences gained through projects, whether they were successes or failures, can teach a project team important lessons.

**Overall Project**

Answer the following questions related to the overall project. Double click on the checkbox that best matches your choice and change it to be checked and include any relevant comments:

|  |  |  |  |
| --- | --- | --- | --- |
| 1. How clearly defined were the objectives for this project? | | | |
| Very | Somewhat | Not Very | Not at all |
| If not well defined, what could have been done differently? Add any other comments.  **This was a real world development project, which meant that while objectives were well defined at the beginning they were constantly changing. I felt as though the requirements for the individual projects outside of the system development were very unclear most of the time.** | | | |
| 1. How clear was your role in the project? | | | |
| Very | Somewhat | Not Very | Not at all |
| If your role was not clear, what could have been done differently? Add any other comments. | | | |
| 1. How involved did you feel in project decisions? | | | |
| Very | Somewhat | Not Very | Not at all |
| If you did not feel involved, what decisions did you feel left out of? Add any other comments.  **I felt as though I was very involved in the decision making process, but in the end there was also a few things that it was made out to seem like it was up to us but wasn’t really.** | | | |
| 1. How efficient and effective was communication among the project sponsor (user), project manager and team members? | | | |
| Very | Somewhat | Not Very | Not at all |
| If not, what could have been done differently? Add any other comments. | | | |
| 1. To what degree do you feel the entire team was committed to the project schedule? | | | |
| Very | Somewhat | Not Very | Not at all |
| If not, what could have been done differently? Add any other comments. | | | |
| 1. Do you feel appreciated, recognized and rewarded for your efforts? | | | |
| Very | Somewhat | Not Very | Not at all |
| If not, explain what would have made you feel appreciated and rewarded? Add any other comments.  **I feel like the extent of the appreciation came from Alain expressing his liking of the system. I felt appreciation from teammates from time to time.** | | | |
| 1. How satisfied are you with the finished deliverable? | | | |
| Very | Somewhat | Not Very | Not at all |
| If yes, what is good about it? Add any other comments.  **It works well for what we were supposed to get done and there’s not a whole lot of bugs in the system. I’ll be very satisfied with the system if the presentation on Wednesday goes well and if the system ends up being implemented by the college.** | | | |
| If no, what’s wrong with it? Add any other comments. | | | |

**Specific Aspects of the Project**

For each of the following aspects of the project, write a short assessment addressing

* how well the team performed in the aspect
* how it could have been improved

Include a brief justification for the assessment of each aspect.

1. Agile development process
   1. sprint planning meetings

* **Our team did very well during the sprint planning meetings. We took the time to make sure that we thought we had time to do the entire thing, and if not, we made it clear that it would go into the next sprint.**
* **We could’ve done better on this if we knew just how much the requirements for the system would be changing.** 
  1. daily scrums
* **We did very well in our daily scrums for dev I and in the beginning dev II, but things eventually deteriorated once we finished the last development sprint.**
* **What we could’ve improved was continuing to do proper scrums throughout the bug fixing sprints once we’d finished development.** 
  1. Sprints
* **In our sprint, we divided up the work pretty well I found. Everyone on the team had a portion of the system that they knew very well, which for the most part worked out well.**
* **I think that we should’ve done more in-depth code reviews when approving people’s work however because there was parts of the system that only one person knew, even if they knew it really well. It’s always good to have multiple eyes on it, and we only had one person for a lot of things.**
  1. length of sprints
* **I think that the proper 2-week sprints worked very well for the development followed with the 1-week sprints. I don’t think that I would change how we did this.** 
  1. sprint review meetings
* **In the sprint review meetings things overall went well with Alain. He was always happy with the work we’ve done and the presentations all went well – except for the very first one.**
* **There were several times where we were scrambling to get things working moments before the meeting was supposed to start. Problems with the published version, problems with the database, problems publishing at all. We could’ve been more prepared many times.** 
  1. retrospective meetings
* **The retrospective meetings overall went well, everyone spoke their mind and was honest**
* **I felt like a lot of the time people weren’t thinking about the retrospective as deeply as they could’ve been? They were looking at things on surface level.**

1. Use of TFS to track the backlog

* **TFS worked well strictly as a work and backlog tracking tool. It was pretty simple to use and that was nice.**
* **We had no access to branching, tagging, code review, etc. TFS was frustrating to use and had a lot of issues merging. We were never really taught how to use TFS and I wish we’d just used GI**

1. Accuracy of estimates

* **We did fine at estimating most of the smaller tasks in the system, and as things went on we got better and better at it.**
* **I feel as though initially everyone in the team was very pessimistic about how long things would take to get done. People kept saying that deliverables wouldn’t be possible to achieve only to turn around and say yea, maybe we can do it.**

1. Distribution of work among team members

* **The distribution overall was pretty fair. We communicated to get things done and everyone seemed okay with the work distribution.**
* **There were sprints where I personally didn’t do much due to getting sick, but otherwise things were good. Every now and then everyone had a week where they took a little less work because they weren’t feeling the motivation and that was fine.**

1. Architecture/design of the system

* **The initial architecture of the system was pretty accurate to the final implementation of the system. At the beginning of the requirements gathering we had a decent idea of what the system would look like, but then we changed things up a lot with shifting requirements. The end product looks more like the original diagram than some of the previous iterations.**
* **I wish that our user requirements hadn’t changed so much, but that’s just what’s going to happen.**

1. Unit testing

* **The unit tests that were developed were high quality.**
* **We had very poor unit test coverage and every time we made changes everything broke. We didn’t keep on top of them because of this.**

1. Integration process and peer review (checking in work and integrating it with the work done by the rest of the team)

* **This was mostly well done by everyone, especially with the final merge of the two branch, I was very happy with how well that turned out.**
* **The auto-merging on TFS is a little weird sometimes. The peer review kind of dropped off later on in development project II.**

1. Runbook and Deployment (including the application, database schema and data)

* **This document was fairly simple and was easy to go into a lot of detail about the deployment steps.**
* **The instructions for the assignment were incredibly unclear and it was frustrating to write. It was a simple task presented to us very poorly.**

1. System test

* **This was overall very helpful for finding bugs in the system and for creating a list of things that need to get fixed. I feel like almost all of our bugs were generated from our system test cases and it gave me a bit more respect for proper black-box testing.**
* **The only thing that I think went poorly with this was that when I was doing my second**

1. User Documentation

* **The user documentation felt like the only useful documentation for the system. It was directed towards the users and implemented directly on the website. It didn’t go out of date due to the changes because they were mostly back end.**
* **Planning the format of the documentation seemed kind of pointless. It would’ve made more sense to come up with a format as a team instead of all on our own, comparing, and then combining items from each of them. The steps to get there seemed kind of redundant.**

1. Developer Documentation

* **This whole document went out of date quite literally within hours of writing it. The only point of this assignment was due to the curriculum requiring us to know how to do documentation, but the entire thing felt incredibly useless.**

**Part B – Lessons Learned Conclusion**

1. What were the highlights of the project?

**The highlights of the project were being able to get experience working in agile development for the 7 development sprints that we had throughout the course of the project. Agile development is really important in today’s jobs and it was a really good experience getting to work in an agile project for that long.**

1. What are the most significant overall improvements which could have been made in the project (the process of developing the system, not the system itself)?

**I think that we should’ve been given a little more of the college’s requirements for the system upfront. Because of the fact that we were developing with our own external users, our system was kind of unique. Every time we asked questions regarding the security of the system the answers were very vague and it took a very long time for us to be told that we did need to use Microsoft Membership framework. We were also never told that we had to be storing our users in the CSAdmin database, which was something we could’ve never assumed on our own, so it was very frustrating having these two things thrown at us like we were expected to know it and it had to be done for production.**

1. What could have been done differently to improve the quality of the system?

**If we’d known the requirements for our users upfront we could’ve spent significantly less time on migrating frameworks and spent our time actually writing code. It slowed us down a lot the amount of times we had our technical requirements change. Requirements from the user I understand, but technical requirements I would expect to have given to us upfront like Entity Framework and MVC were. We were also really lacking the unit tests, we developed them a lot at the beginning and then when they became obsolete, we didn’t keep trying.**

1. List the major enhancements that should be made to the system as part of maintenance.

**The RAC Advisor currently cannot evaluate a self-evaluation and the Content Specialist flow is completely unimplemented. If think if this system goes to production, those are the major two components that I would think should be implemented.**

1. The three main criteria for measuring the success of a project are: Quality of work, Schedule, and, Budget. Overall, how successful would you rate the project based on this criteria? Please include any additional comments you have on the project.

**The quality of work was okay. Given that we had a lot of change in technical requirements, I think that we really did the best that we could to make sure that everything was as tested as possible. We also stayed well within schedule for the system and made sure that all of the agreed upon deliverables were included in the final product. We stayed in budget for the system since we weren’t getting paid, but our motivation certainly got taxed.**

**Part C – Program Integration**

1. Briefly assess how well the project integrated the knowledge from the different courses in the program.

**There was very little that we learned that wasn’t use throughout the development of this system. Obviously Android development, WPF applications, and some of the other languages we learned weren’t used, but the theory from these courses was applied. We used client-server paradigms that we learned in PHP and node.js, we used knowledge of C# programming that was introduced in with WPF applications and the entire foundation of our programming knowledge that we learned in Java. We also used a lot of the knowledge acquired in all of our systems courses. Certain things like Gantt charts were never touched, but the basics of project management were. Overall, almost all of the courses in the program became relevant in the development of this system.**

1. Briefly describe how the project could be improved to address other aspects of the program.

**There’s very little in the program that wasn’t touched on by the development project. The only things that I feel you could include are some of the other technologies that we learned, but that would require cutting other things out, which would completely defeat the purpose. What I would do is take things out of the development project because there’s a lot of documentation that needs to be done. We didn’t have access to really any, so it feels kind of cheap to say it’s for maintenance teams. I would focus more on the development, use of agile, and proper testing.**

1. From your perspective, briefly describe the strengths of the Computer Science Program.

**The Computer Science program prepares students very well for the workforce. It teaches so many practical skills that we can use in a variety of different types of programming jobs, from front end web-development to databases to system analysis. This program prepares us for many different roles that we could want to work in.**

1. Briefly describe how the Computer Science Program could be improved.

**The program focuses a lot on the technologies we learn. While that’s great because we have a lot of resume material, I think it would be worth cutting a language off the list to broaden our thinking and learn to develop using new paradigms of programming languages. I’ve said it many times now throughout various courses and I’ll say it again, I think we should learn functional programming or a language that is drastically different like a dialect of Lisp. These would introduce us to things that would open us up to things that are incredibly different from what we learn in this program. To most of us, s-expressions would be entirely unreadable, or a language without loops and strictly recursion would be terrifying. But I think people should be exposed to these kinds of things to broaden their skills beyond technologies.**

**Part D - Teammates Evaluation**

1. Complete the final Team mates assessment for the course.

**Part E - Course Evaluation**

1. Complete the course evaluation for the course.

**Marking Scheme**

|  |  |
| --- | --- |
| Part A – Lessons Learned Overall Project | 4 |
| Part A – Lessons Learned Specific Aspects | 32 |
| Part B – Lessons Learned Conclusion | 20 |
| Part C – Program Integration | 12 |
| Proper use of English | 5 |
| Correct Submission of Files | 1 |

**To submit**

When you have completed the assignment, upload the documents to Moodle.