Organization Plan: Team #19

During the initial planning meeting we decided to modularize the project. This meant that each of us would be responsible for coding different parts of the project. We felt that the project design inherently encouraged modularization, since the milestones involved several different codebases, and were expected to be completed concurrently. After a long discussion we concluded that the most efficient means of developing the software for this project was to break it up as follows

Keith Pattison (Team Leader): Primary API Documentation developer, Table Class, Application assistant developer, Project Documentation, Communication with other teams

Hamdan Siddiqui: Record class, Primary application developer, Assistant developer of API documentation.

David Holdren: Database Class, Parser for WHERE expression, Test Cases

At the start David was very enthusiastic about his role in the project. During several of the early lab sessions he even explained to us in detail how he thought the project should be laid out from a software design standpoint, and he passionately urged us to use Google Test for our test cases, rather than the Boost testing framework (he volunteered to write them by himself if that was the case).

Execution of Organization Plan

To put it simply the execution of this plan slowly developed from a solid software development plan, into a complete and utter disaster. Our team successfully reached the first milestone with relative ease, and even managed to publish a high quality API, which was regarded as “very nice documentation” according to a member of Team 36 (the team we exchanged with).

After checkpoint 1, David stopped showing up to lab. We just assumed he was incredibly busy, and was working on the Test Cases that were due in a week. After all he’d been so excited to use Google Test, and write the cases. Both Hamdan and Keith sent David emails in the days leading up to the due date, and at one point he responded saying he was working on them, so nothing was ever thought of it. However, the due date of checkpoint 2 came and went without any feedback from David. We sent him a flurry of emails, texts, and Facebook messages attempting to contact him, and see if he’d turned in the code. After receiving zero responses from David for two days after the due date, Hamdan and Keith decided to write Test Cases on their own (since the team that was supposed to receive them) was consistently complaining about not having got them yet. Both Hamdan and Keith put hours of work into attempting to program the test cases, but no matter how hard they tried Google Test just would not compile with Visual Studio, furthermore the same problems ensued when they tried to quickly implement the Boost Testing Framework. With the due date for the third milestone quickly approaching, Keith and Hamdan had no choice but to abandon the test cases, and take what they thought would only be a minor 15% deduction on their project.

The problems with trying to develop a working database library then descended on Keith and Hamadan. David was supposed to have written a parser for the WHERE expression, since he spoke of having tons of experience with parsers, yet he didn’t write any of it. So Keith and Hamdan got hung up on numerous problems trying to develop the database. While we eventually succeeded in making a fully functional database system it was well past the required deadline, and we told the other team to go ahead, and use another team’s library (so that we wouldn’t hurt their grade as well).

With problems still persisting in the Database library, Hamdan and Keith focused their efforts on the application they were originally supposed to design together. Although, the project came together behind schedule both of them were able to contribute, and it resulted in a fully functional, well designed database query application. However, the team that was supposed to send us a Database Library failed to deliver a working query function. So we had to pivot from using the database library of Team 31 to the library of Team 7. After altering our code significantly to adapt to the API of team 7 we were able to complete a working database application, although a few days past the due date.

Overview of Problems Encountered

By far the biggest hurdle we faced on this project was having too much faith in the people we were working with, as well as too much faith in our own ability to deliver to the teams we were supposed to deliver too. Once we got knocked down to only two people working on the project we should have contacted either the T.A or Dr. Keyser. We had way too much faith in both of our coding abilities, and truly thought we could have delivered everything on time, and fully functional. Both of us should have accepted that we are not coding gurus, and were never going to able to make up all of the work that David didn't finish in a timely manner. We also should have been more serious about reaching out to David, and making sure he was completing the parts of the project he said he would complete. If we had been just a tad bit more responsible with maintaining contact we could have foreseen this coming, and solved all of our issues before the deadline. This was by far the worst experience either of us have had with a computer science project. We never thought that we would have to rely on so many other people other than ourselves, and in turn we never thought we would be given a programming assignment where actually finishing the project was too far out of our reach. We accept responsibility for not realizing that with our busy schedules as students, it was virtually impossible for us to hit the deadlines we needed to hit, and we understand that we should have reached out for assistance far, far sooner than we did. So even though this is the worst experience either of us has had with a programming project in college, we also feel as if this was one of the best learning experiences we have had. It allowed us to realize the value of communication in real world projects, and truly understand that you can't always write enough code to solve all of your problems in the real world of software development.