**ECSE 211 – Design Principles and Methods**

**Design Project: Final Report**

Author: Team 18

Date created: April 8th, 2015

Date submitted: April 13th, 2015

**0.0 TABLE OF CONTENTS**

* **1.0 Introduction**
* **2.0 Team organization**
* **3.0 Issues encountered in the progress of the project**
* **4.0 The Budget**
* **5.0 How the process contributed to success (or failure) of the project**
* **6.0 Conclusions**
* **7.0 Consent and signatures**

**1. Introduction**

The overall goal of the project was twofold: to develop an appreciation of the design process (hence the title of this course), and to create a full project, with 6 team members, that fosters engineering project skills in a workplace environment. The latter contains a more broad and variable part of the goal. In particular, our project concerned the designing, implementation, and testing of an autonomous robot capable of navigating a field with obstacles, launching balls at a target, and then returning to its original position. The more important aspect of the project was the method and system used to achieve the technical goals, and not the actual product of all the labour. From establishing a potential blueprint to testing the capabilities of the final product, the main idea was to document everything and assess progress as it happened.

Being involved in such a project is the kind of experience that not only offers insight to the participants into how real engineering projects work, but also how they shouldn’t work. The many pitfalls of the design process and project coordination are made evident, thus serving as a valuable experience that can be used in the future to better estimate in advance how, why, and when things should be done in a particular engineering project.

The project is intended to provide students with a basic scheme on how to manage this long process in an iterative and organized fashion. At many points, it is easy to lose track of progress and veer off into catastrophe, which is why the project needs to be experienced first-hand and not simply discussed in the classroom. This project intends to arm students with the ability to fuse the design process with efficient and structured project management, thus preparing them for future endeavours in the engineering industry and professional practice.

**2. Team organization**

As specified in the lectures, a Gantt chart was used to allocate tasks and resources to a particular task. The initial Gantt chart was designed using the template one provided. The initial task breakdown was split up by first allocating at least one role for each team member. The exact tasks that the project was divided into were determined based off the time given and the information provided in the lecture on how each phase should proceed. The template given was used as a guideline, and the final Gantt chart ended up looking very similar, only with some parts of the timeline shifted, and with many notes added in for more specific instructions when necessary. On occasion, an additional ‘weekly task’ was added for more clarity on what each member should be doing that particular week.

**3. Issues encountered in the progress of the project**

Many dependencies were correctly identified, but some could have been connected differently and made the Gantt chart flow a little better. The critical path dependencies basically included 3 phases: initial, intermediate, and final versions of mechanical design, documentation, and software. Other items such as proper testing and building models of certain key components were also added as dependencies. In addition, milestones such as client meetings and the beta demo were inserted in between dependencies, which spread them out a little. Some issues with the Gantt chart were that some tasks could have worked better without dependencies, since they were running concurrently in the same week with other dependent tasks. For instance, leaving out certain dependencies on documentation and some of the testing may have made the management of the chart more efficient, thus allowing for the actual project to proceed better. The assumption that all of the dependencies would work nicely and make the project more organized was definitely wrong.

In general, balancing time properly and completing dependent tasks properly proved to be an issue. If one task was not completed and another depended on it, for example software before testing, then the next task had to be delayed or squeezed in with other tasks in the following week, leaving less time for other parts. In general, this was reflected in the Gantt chart, but at times was not shown if the work was completed on the weekend. Overall, there was much lateness with many of the tasks, particularly those of software, which many of the others depended on. The project could have run much more smoothly, and although the initial plan was mostly followed, there was much delaying and problems with completing the tasks on time. In particular, the end of the project was delayed a lot, with basically no integrative testing done.

Another issue impeding the progress of the project, although not as much as the aforementioned items, was the occasional lack of clear guidelines or missing information. Indeed, it is necessary to take variable conditions into account, and the project should ideally be managed so that changes are expected later on and they can be implemented without harming the previous progress. However, having the final conditions further in advance, such as map information and specifics on what kind of balls can be used, would likely have helped the project run better. This caused some slight delays in that we had to seek the advice of a TA or professor for more information.

Other issues include members of the team not being informed prior to meetings and not keeping up with the status of the project. In general, responsibility of each team member is expected, but this was not always ideal. Proper communication is important during the course of such a project, but team members were not always informing each other and collaborating in the proper manner. The management was also not adjusting and fine-tuning the course of the project enough, something essential in the design process and in general project management.

**4. The budget**

In the beginning, the budget was not really an issue seeing as there were plenty of hours and there was only a limit on the final number of hours per person on each team. More specifically, each member is permitted 9 hours per week during a 7-week project, for a total of 63 hours per member and a maximum total of 378 hours, which proved to be enough. However, near the end, there was a discrepancy between certain members and the real difficulty was balancing the total number of hours so that no one’s total hours amounted to more than 10% below or above the average number of hours for the entire team.   
  
The original planning did not affect the timeline much, other than when it ended and the approximate number of hours each member should be working per week. As the timeline progressed, the main issue was with people lagging behind in hours, which did affect the resource allocation later on because of the need for more balance in hours (this is mentioned in the Gantt chart in the notes near the end in the most recent version).

Had there been fewer constraints on the budget (and assuming the budget is referring to time), more would have been allocated to software development and integrative testing of the robot. Ideally, the integrative testing phase should be much longer, since this is essentially the final optimization phase of the project. In our case not much was done since there was little time left after all the lateness from the previous tasks. In an ideal scenario, additional budget would have been useful during both the software development phase and during the integration testing phase, which are both evident in the Gantt chart.

Resources were not lacking in terms of number of resources, but in skill set. Some members were more proficient in software development than others and thus were given tasks related to this domain. As a result, sometimes the software team had other things to do and devoted less time to software, something which affected the project later on in a negative manner. Had there been fewer budgetary constraints, this issue would have been resolved by allocating both more time as well as more resources to software development. In particular, some members would have been exclusively in charge of software, with others available to help out if needed. As it happened, the other components of the project ended up being less difficult or time-consuming than the software, hence the reason for this hypothetical solution. More of this is discussed in section 6 (conclusion).

**5. How the process contributed to the success (or failure) of the project**

The overall process, from the Gantt chant to the client meetings to the documentation, by and large, was intended to maximize productivity, organize and streamline tasks, and make things easier. It was useful in achieving goals in the sense that it offered tools to record and assess progress; however, the particular elements chosen to represent this were not always ideal. Using a Gantt chart was sometimes much less productive than it should have been – much time was spent tinkering with it when a simpler schedule could have been set up with substitutes, thus saving time and possibly making changes easier. Devoting more time to actually doing tasks instead of trying to have the Gantt chart or other scheduling program represent reality exactly is likely to increase the chance of success.

The most difficult part of the process was, (ironically), following the schedule originally planned to make things easier, and staying true to it. Changes were necessary and the project was constantly pivoting from one state to the other because of all the delays that occurred. Another primary difficulty was implementing and managing the software; it proved to be much more difficult than expected. Although some members did have some software experience (more than what was given in the very basic Java course at McGill), having members only from electrical engineering majors, in U1, and none from computer or software in U2 was a disadvantage (compared to other groups). Other members who were more advanced or more experienced in software would likely have helped. As a result, delays from the software team and the subsequent changes made the project much more difficult.

In terms of testing, it developed as the plan progressed, but because of the software delays in the second part of the project, it was also significantly delayed near the end. Testing was initially supposed to occur concurrently with other phases of the project, as they evolved, and this did occur until the beta demo, when the software delays really started affecting things. Nonetheless, a fair amount of time was spent on testing, mostly on calibrating the values of certain elements of the robot such as the wheel radii. There was testing done at multiple points, at the subcomponent level, as seen in the Gantt chart. The tests designed seemed to be sufficient for the particular element, but then again a main problem was with the software overall – the odometry and navigation did not work properly, affecting many other things in the robot’s movements. Some testing of the launcher and other components were done, but these proved to be futile during the final run since the robot did not even reach that stage. Whether or not they were all sufficient is not fully clear; there were calibration problems in the software which caused many issues. In the competition, the final software was supposed to work, but didn’t, something which integrative testing may have solved, either by fixing calibration issues or making sure there was no unforeseen problem.

The full prototype testing was initially set as 2 weeks in the Gantt chart, with the expectation being 1-2 weeks for integrative testing. However, due to the previously mentioned issues with software development, basically no integrative testing was done seeing as much of the software was being fixed last-minute, and this did not provide opportunity for the testing team to actually do a full integrative run. As mentioned, it was known what didn’t work and that the robot would not get very far as a result of these issues, so integrative testing would not have been particularly useful if all that was observed was the robot bumping into a wall not long after it began the run. This was part of the chain of issues encountered in the project, and potential solutions are addressed in section 6. In short, it was not so much that the test design process was ineffective, it was that other delays caused problems in scheduling that consequently did not allow much time for testing. In order to make it more effective, components would be tested individually after completion, and there would be more collaboration between testing and software teams so as to better determine scheduling.

**6. Conclusions**

Many things were learned from the experience in this course: the theory behind engineering design and process, namely scheduling, documentation, testing, optimization of parameters, and project management. As a direct result of the project experience, the main lesson is that many things do not go as expected and a good engineer or team of engineers will anticipate such problems in order to refine and control their schedule and productivity.

A clearly defined and controlled process is necessary when working with multiple individuals because a system needs to be used for communication, mutual understanding, and generally avoiding confusion from different methods being used to accomplish tasks. By setting up a precise methodology, it is less likely to run into issues of compatibility, and also more likely to boost productivity. The knowledge and experience gleaned from this course will also be of use later in our McGill engineering studies, particularly from the coding in the labs, but also from the engineering project experience. More specifically, it will be useful for ECSE 456 (ECSE Design Project 1) and ECSE 457(ECSE Design Project 1), which are 2 connected courses concerning a 2-part electrical engineering design project. The experience of the design project will be useful in these courses, since this is where a professional project is undertaken by the student under the mentorship of a staff member. Another course where these skills will be useful is ECSE 323 (Digital Systems Design), where students learn the principles of control design, and sequential circuit design. ECSE 211 is a prerequisite to all 3 of the aforementioned courses and it should provide some good background due to the design and project experience it was intended to provide.

In this course, if the project were to be done again, multiple things would change. To begin with, the schedule would be structured differently, with fewer dependencies so that the Gantt chart is more malleable. For instance, as mentioned in section 3, things like the initial documentation and software class hierarchy do not need to be on the critical path. The first design meeting would involve creating the entire schedule elaborately, determining the necessary dependencies more accurately and better paving the route to success with other relevant changes (mentioned in section 3). The design and building phase would be pushed earlier and more time and resources would have been allotted to software development and testing (as mentioned at the end of the Budget section). While it is important to have a feasible and durable design, this should not take as long and is not as crucial to the project as the software turned out to be. Some of the software should be written in advance, before the robot is built. Once the design is chosen and the robot built, changes can occasionally be done to the hardware but the software development must move forward and be tested with the robot – this phase of refinement is the most important. Finally, there would be better collaboration between team members (as mentioned in section 5), so that the type and extent of testing could be determined more precisely. This might also push the software to be done earlier. More design meetings would be set up, with shorter meeting times so that no members can get sidetracked, and meetings with only part of the team would be set up as well when necessary. Specific hours, and not only tasks, would be assigned to each team member each week. Most importantly, the management would plan better to deal with the evolving status of the project: the Gantt chart updated periodically, resource allocation varying when necessary, regular status updates during the shorter meetings, and better collaboration between team members who would be more involved in scheduling and planning (as opposed to simply having different members observe the plan on the sidelines). All of this would be done in the best effort to streamline productivity and achieve success.

Assessing how we did during the project involves analyzing both our process and the final outcome during the competition. The ultimate performance of the robot of Team 18 was evidently not satisfactory, with the localization phase being the farthest checkpoint it reached. However, in terms of documentation, testing, and clarity of materials and methods used during the process, the work was satisfactory. The entire point of this course is not to have a functioning robot, but to gain an appreciation of the process and design involved in engineering. The main question we need to ask ourselves when we want to know how well our process was is “Can others reproduce it?” and the answer is yes. There is enough information, with clear references to changes and other relevant sections in the extensive documentation that occurred throughout the semester. Although many parts of the process were far from ideal, with improved solutions outlined above, the team succeeded in reproducing everything that occurred over the course of the project in documentation, which was the primary goal of the project in the first place.

The engineering profession is a challenging one that requires a precise methodology, active attention to detail, and special care for planning ahead in order to succeed and achieve the ingenuity that defines it. While it would have been nice to have a fully working robot by the end of this course, which would have represented a peak achievement, this course nevertheless taught us the valuable lesson that even if things do not go as planned (a common occurrence), we must be prepared to face the changes and variable conditions that the engineering occupation demands of its constituents. The way we prepare is by experience, and even if we make mistakes, we use them as an educational milestone and not as a failure.

**7.0 Consent and signatures**

“*The undersigned members of Team 18 agree that the contents of both this report and the information handed in on CD, DVD or memory key, provide an accurate representation of the work done on this course and the contributions of each team member*.”

Joshua Donath:

Victor Repkow:

Musa Shakib Khan:

Bobak Hamed-Baghi:

Mathieu Pigeon:

Clara Dionet: