ECSE-211 Winter 2015  
Final Report  
Date: 13th April, 2015   
Team 05

Contents

[1. Introduction 2](#_Toc416619963)

[2. Team organization – the start-up of the project 2](#_Toc416619964)

[3. Issues encountered in the progress of the project 3](#_Toc416619965)

[4. The budget 4](#_Toc416619966)

[5. How the process contributed to the success (or failure) of the project 5](#_Toc416619967)

[6. Conclusions 6](#_Toc416619968)

# 1. Introduction

The goal of the project is to prepare engineering students for the real world working environment by accumulating hands-on experience in the design process, demonstrated by the design of the NXT robot. Since design is one of the main aspect of engineering, it is important to be familiar with the process before entering the profession. Thus, the main reason for doing this project is for students to use it as a medium to accumulate experience and to familiarize themselves with the design process. For example, in the project, a change in the requirements was made almost a week before the competition, which reflects the client’s changing requirements. Team members were also selected randomly, much like in real life scenarios, where co-workers are assigned rather than chosen. Thus, this project encourages team management and cooperation. Not only does this project enforce the designing phase, it also pays close attention to the documentation, since documentation is crucial to the design process as a whole. In order for a product to be successful, it must be repeatable, thus documentation is needed to record the entire process to allow others to recreate the process and thus, the product.

The project was intended for students to confirm or disconfirm their career choice by using this project as a simulated simplified version of the real world working environment, and for them to acquire design experience.

# 2. Team organization – the start-up of the project

The tasks were allocated by strengths and weaknesses. The following are the main roles that needed to be assigned: project manager, documentation manager, software manager, hardware manager, and hardware/software testing. Each manager would be associated with their respective task. The capabilities document was written for a better understanding of each team member’s strengths, weaknesses, and preferences, which facilitated the separation of tasks. There were two main strategies that have been used for the tasks separation. First, in order to increase efficiency, the most proficient members would be assigned for the task, since they would be reliable and would complete the task in the shortest time in comparison to the rest of the team. The second was to have the tasks separated into multiple independent tasks. For example the building of the obstacle avoidance has no impact on the localization. Thus, one or two people would design two systems at the same time, independently, in order to enable multiple systems built, calibrated, and testing at about the same time.

From the initial specifications received about the competition, the software design was subdivided into odometer, odometer correction, localization, navigation, and obstacle avoidance. The hardware was divided into loading and shooting. Lastly, the calibration and testing for each system was added. The Gantt chart was initially designed to keep track of the progress of every main task, which were then broken into subtasks, and assigned to the available resources. After assigning resources to tasks, the work units were adjusted to account for budget overflow. Since only nine hours are allocated per week per member, one hundred units represents nine hours. Thus, every hour is approximated to eleven units.

In order to estimate the initial task breakdown, the deadlines were used as reference. The beta demo (March 27th) and the competition (April 10th) dates are constants that would never change, thus they were used as a reference and deadline. Also, each lab done prior to the final project took approximately four to six days to complete. Since most of the software will be based on what was already done, five days were allocated for the initial design of each component, which were done almost simultaneously. The calibrations, from the labs, only took one day at most to perfect, thus one day was allocated for the calibration of each system per member. Lastly, testing requires the most amount of time, since it requires time for feedback. In order to modify the system and do more tests, the team requires a lot more time, thus any extra budget was spent on testing and improvement. For the beta demo on March 27th, a working prototype needed to be ready. Thus, all systems needed to have a working prototype by that date. Afterwards, as suggested from this class’ lecture, the rest of the time (2-3weeks) was allocated to integration testing and improvements for the competition.

The first guideline used in the initial Gantt chart was to have at least two weeks allotted for the integration according to the lectures. The second guideline was that the time dedicated for every resource for every week cannot exceed one hundred percent. Although, further on in the project, this guideline is only loosely followed as things do not always go as planned, due to the budget spending and some members had to work over the weekly budget, but did not exceed the overall allowed budget of the project (63hours /member).

# 3. Issues encountered in the progress of the project

At the start of the project, many of the dependencies were not correctly identified, due to misunderstandings in the tasks and not fully comprehending the task at hand. For example, although the launcher and the software were designed independently, they still needed to be tested together in order to have an accurate representation of the robot as a whole. And every tasks designed were still dependent on each other even, because if localization process of the robot did not function properly, the robot would not navigate to the proper shooting area.

The critical path of the project is formed by the following dependencies: mechanical proposal → software proposal → initial software → code mash up → integration test → post beta modification → pre-competition integration test → final competition → final report. This makes sense, since this was the main skeleton of the entire project, taking the longest time. It is where all the other subtasks are divided, creating the critical path. The initial mechanical design had to be completed before the software architecture could be designed as the software algorithms highly depends on the hardware. As much as the team tried to make each tasks independent of each other, the code mash up for the integration test highly depended on the development of the initial software.

During the design process, a lot of initial ideas have been changed due to the lack of satisfaction in the output results. On the hardware side, the launching mechanism for launching Ping-Pong ball was modified due to the misunderstanding of the target locations. Originally, the shooting angle of the ball was nearly 20 degrees to the ground. Even though the launcher worked well, it was not able to hit a target placed 30 cm near the wall. The ball would either fly over the target or get blocked by the wall. Consequently, the initial shooting angle of the ball was elevated to approximately 45 degrees. As a result, a parabolic trajectory was observed and it is able to hit targets close to the wall. On the software development side, the original plan to avoid obstacle had been improved. At first, the software members assumed that only 2 Ultrasonic sensors are enough. The robot had one on the left side and one in the front. However, the downside of this design is that it can’t detect anything on its right. Hence, software and hardware team came up with a solution to install another side sensor to the right of the robot, making the angle of detection to nearly 180 degrees. This had a critical impact on the project, as the team gave up the 4th sensor port for a second light sensor to improve the robot’s odometer and relied mostly on its ability to avoid obstacles. Thus, this shifted the whole design into focusing the robot’s ability to avoid all scenarios rather than focusing on building an accurate odometer. As a result, there were complications in the software department.

One of the critical decisions that had impacted the robot significantly was the path taken. Initially, the team chose the set path as it seemed to be the easier path to take. As a result, work were done on the accuracy of odometer and odometer correction. However there were several problems such as slow speed, inability to correct heading of robot and very narrow paths for avoidance. To avoid all these problems, the team chose the random path. The speed improved by a lot since the random path was fairly straight instead of turn after turn. To solve the inability to correct heading, the team decided to not use correction as the robot traveled but rather do localization at the shooting area. Extensive testing had been done on the odometer error per tile travelled and it determined that the robot would stop somewhere inside the shooting area and was safe to do localization.

During the testing, a lot of external factor determine the performance of the robot and the team has been able to adjust and bring some modifications. One of the major factors is the battery level. It has a significant impact on the launching system since the system depends on the strength of the NXT motor. It is essential to use fully charged batteries to provide the most and consistent power to the motor. Another external factor was the level of the flooring. On the day of competition, the level of the competition floor caused an unexpected problem. Initially, the robot did not have a back wheel, but rather use a rectangular base support. In the first run, the robot was stuck by the intersection of four 4x4 square tile due one being higher and one being lower. To fix this problem, the team members quickly came up with a new design, by replacing the back support by a caster wheel.

The problems do not affect the progress of the project significantly due to the strategy of the team. When problems are found initially, they will be reported to their respective software/hardware teams. For example, a localization problem will stay within the localization team. It will have no impact on obstacle avoidance team or launching team. The team then will try to solve that problem and move on. The team has two meetings (aside from meetings with professor and TA) every week which the team members put everything they have done together, talk about their progresses and make sure everyone is on the same page. If a team fails to solve the problem before the meeting, more resources will be allocated to that team to speed up the progress. Therefore, the initial plan was followed.

# 4. The budget

The budget of this project is nine hours per week per person, totalling up to 378 hours for all seven weeks. This budget constraint the amount of time allowed to further improve the robot. Since a design can always be improved, the time constraint was necessary in order to have an end point. Without a budget, the design for both the software and the hardware could have been further developed with no end. For example, due to the poor design of the odometer correction, it has been omitted. However, if there were more resources and budget, it could have potentially been redesigned in order for it to be functional; the shooting could also have been more accurate. The obstacle avoidance could have been improved to accommodate sensory issues.

Initially, nine hours were allotted each week per member. However, the first two weeks had less budget spending than intended, so the extra hours were banked into the later weeks, where software development and testing were heavily emphasized, and needed a lot of improvement. If extra budget was available, it would have mostly been used in the software testing part of the project. By the end of the project, all remaining resources was dedicated to the software, since there was still improvements and tests that could be done to allow for a more accurate odometer, better obstacle avoidance, and a faster localization. If there were fewer budgetary constraints, more time would have been dedicated to improve the odometer, and potentially fix the odometer correction issues in order for it to be implemented to the design. Thus, the initial budget spending affected the development timeline positively, and if it was not for the banked hours, the software development would not have been as developed as it is currently.

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

The process that the team followed contributed positively to the success of the project. The initial strategy, which was to separate tasks that are as independent of each other as possible (Localization, Obstacle Avoidance, Navigation, Launching), allowed the team to quickly build an initial running software. The rest of the project was mostly dedicated to integration testing which allowed the team to have time tweaking the software and confirm its reliability through multiple tests (integration testing, speed test, etc.). Also, the fact that the team dedicated their whole design process into improving the obstacle avoidance and advanced localization, it allowed the robot to quickly avoid and localize (17 seconds and worst case of 22 seconds) and get to the shooting area. Once at the shooting area, the robot could simply localize itself again. Thus, not much time was spent on trying to improve and develop a better odometer correction. As a result, the team had sufficient amount of time on testing and refining the robot’s ability to avoid obstacles, localize, and fire projectiles precisely.

During the design project, the software development was the most challenging part. In fact, implementing and calibrating the algorithms required lots of testing. In the case of obstacle avoidance, initially the robot used bang-bang controller to avoid obstacles. The idea was good in theory, but was proven by tests to fail to avoid obstacles in many cases due to its inability to linearly speed up/down motors with respect to distance. As a result, bang-bang was replaced by p-controller. In the two weeks after p-controller was implemented, tests were being done on the performance of it and software and hardware were adjusted accordingly after each one. In other words, after a problem arose, it returned to the designing phase, and then tested again in order to increase the probability of success. Each part of software had similar issues, where methods were constantly being improved based on test results.

The testing took a significant portion of the team budget - around 24% of the time was allocated to hardware and software testing. They were divided into two types: individual and integration tests. Individual testing were done gradually until the end of the project. Because the software is divided into three teams, each team had its own individual testing on its assigned part of the robot. For example, during the whole process, launching team did three tests specifically on the launching system and the avoidance team did four specifically on the obstacle avoidance ability. On the other hand, integration tests were used to make sure that the code from each team will have no impact on other codes. For example, a test where the robot has to travel to a point while avoiding obstacles is meant to evaluate the coordination of obstacle avoidance and the odometer. Those types of test were mostly done during the second half of the project. Initially, it was expected that the integration tests would take only the last two weeks. However, the assumption was wrong as the team started full prototype testing during the beta demo week. Thus, each subcomponent had been tested before moving onto the integration test to further improve the time it takes to complete.

The testing done for the project was not sufficient. For example, as observed in the round 3 of the competition, obstacle avoidance failed due to the lack of consideration of the extra width of the launcher (about 2mm). There should have been more simulation tests for making sure that the robot is able to run paths with complicated obstacles position.

One way to improve the testing process is to have more random simulations. For example, in the localization test, instead of having our team members placing the robot, random number generator should be used to determine the XY coordinate and heading. In a test for avoidance, a bystander should place the obstacles and more simulation is needed.

# 6. Conclusions

In this course, the team has learned how to adapt to changes, the management of time, proper communication between teammates and the procedure of a design project. The requirements for the competition changed a week before the competition date, and the team needed to make adjustments in order to meet the new requirements. The time to make this adjustment needed to be made, in order accommodate the new design. The different systems also needed to work together in order for the robot to function properly, thus communication was important. Also, the team has learned the importance of knowing each team member’s personality, capability and work habits. The latter is important because we can use each member’s strength and weaknesses to effectively bring out their potential and utilize their resources as efficient as possible.

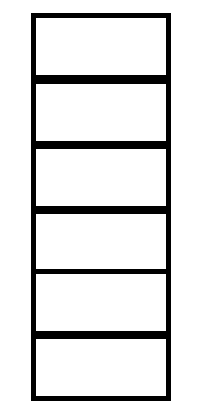
An effective and controlled process is necessary when working in a team, thus the importance of the project manager. The project manager must assign tasks to resources, and ensure that all the tasks are on schedule, otherwise to adjust the resources in order to optimize the budget. Without an effective and controlled process, tasks may potentially be late or overlap, which will then slow down the entire design process, thus reducing chance of success.

The skills learned in this project will be applicable in the future classes: ECSE Design Project 1 and 2 (ECSE 456 and ECSE 457 respectively), which are presumably the same type of project with higher level systems. It will most likely require the same management, communication, discipline, and adaptive skills that this project requires.

If the team were required to do the project over again, the main change would be to increase the amount of resources allocated to the software. By the end of the project, the main component to be improved for the competition was the software. However, only half the team had a deep enough knowledge of the software design to help, which made the job much more difficult for them. Since explaining the code’s logic would take significantly more time than simply finding the solution, the job on those software members was significantly increased, and the final performance of the robot depended solely on them. If there were more members who were aware of the changes and involved in the software design throughout the whole project, more creative solutions could have been found to the design’s problems.

The first run of the competition failed due to the caster wheel getting stuck in the crack between grids, which rendered the odometer useless. As a result, the robot navigated around the entire field, while avoiding all the obstacles. The second run of the competition was the most successful, maintaining the fastest speed of the robots that completed the course, by relying on the localization and the obstacle avoidance. The third run failed, due to one or two millimeters that lightly grazed the shooter as it was avoiding an obstacle, which once again rendered the odometer useless. Although the team was expecting better results in general due to the successful tests done prior to the competition, the team decided to sacrifice safe obstacle avoidance for speed, which resulted in the two failed runs. Overall, the team is very proud of the results, especially the second run with a time completion of only three minutes eleven seconds!

**The undersigned members of team 05 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.**

**Team Member Signature:**  
  
**Eric Vuong** (Project Manager/Software)

**Gwyneth Pang** (Documentation Manager)

**Yu Chen Xu** (Hardware Manager)

**Adrian Pan** (Software Manager)

**Shi Fao Fuh** (Software/Testing)

**Yutian Zheng** (Hardware/Testing)