ECSE 211 Final Report Format

The intention of the final report is to create a document which reviews the process you have undertaken in designing your solution to the problem presented. You already have a significant amount of design documentation including, but not limited to, the original four documents which have been updated throughout the process, the Gantt charts with their weekly versions, the budget documents and the software and hardware architecture documents. In addition, there should be full test specifications and the results of the tests. All these documents should be handed in at the end of the course on a CD, DVD or Memory Key. These documents will be reviewed for part of the mark on this course. The documents should be properly indexed through a directory structure (so that the history of the documentation can clearly be found) and ***there must be a main file describing the contents of the disk*** so that we can easily find anything we want. In addition, please ensure that the ldd file of the final robot design is included. **2 copies of the CD (or equivalent) will be handed in.**

The goal of the report is not to duplicate information which already exists in the other documents, although it may refer to that information, if necessary. The report should concentrate on reviewing **the process** itself. It is limited to six pages, single spaced in 12 point font (the same as this document). It should consist of the following sections (*each of which should be no more than 1 page long*) and answer the questions below in each section (*hint: it is often useful to write out the question and then the answer to it)*:

Red = Nayem

1. Introduction

*This should review the goals of the project – about half a page.*

*What was the main reason(s) for doing it?* The main reason for doing this project was learning by doing. As future engineers, it is important to get as soon as possible a hands-on experience on real design project. Theory on the subject are not sufficient. The primary goal of this project was therefore to gain experience on design in a real world situation, i.e. creating a robot capable of doing specific tasks. This implies different sub goals:

a) Project management: Managing a team and keeping it on schedule can be difficult. To do this job efficiently, one goal was to learn the use Gantt chart to allocate resources. Management is also about keeping a budget. This goal can be achieved by setting precise deadlines and make sure each division follows them.

b) Teamwork: Coherent communication between members was another goal of this project. In real-life situation, key members of any organization must be able to communicate their ideas and their agreement/opposition to collective decisions. In the final project, we had to make some tough decisions impacting our overall progression. To clearly understand what was at stake, we needed input from all team members.

c) Creativity: This project, even though it is not implicitly stated, has for goal developing creativity. Problem-solving never has clearly defined answers. This project forced us to find innovative ways to efficiently launch a ball. Investigating the properties of new materials like rubber bands allowed more powerful mechanisms.

*What was the project intended to achieve?* It was intended so the students can have real engineering experience early in their program. After this project, students should have developed a new set of skills essential to undertake any design project.

1. Team organization – the start up of the project

*How were tasks allocated?* Each team member filled a form with questions on their background experience and their preferences. Because of the diversity of the answers, everyone could obtain the role they chose. After getting their precise role on the project, the lead of each division (software, hardware) started dividing their main tasks into subtasks and separating them between members respectively. As an example, the documentation lead divided the hardware document into subtasks, so most of them could be answered separately.

Prior experience,

*How was the initial Gantt chart designed?*

*What information was used to estimate the initial task breakdown?*

*Were any guidelines followed in developing the first version of the chart?*

(first version given by Prof. Lowther.)

1. Issues encountered in the progress of the project

*Were all the dependencies correctly identified at the start of the project?* No, not all dependencies were identified. We knew that software was a critical component of the project. Navigation and localization were the most essential element to program. Without them, the robot could do nothing. However, because no precise deadlines were set to finish them, they were developed continuously until the very end. Because of that, we were not able to test thoroughly the final design.

*What dependencies contributed to the critical path of the project?*

*What initial ideas turned out either not to work or be based on wrong assumptions?* Numerous hardware prototypes ended up not meeting the requirements. Most of the designs were based off the raw power of the Mindstorm motors which was not strong enough to launch the ball at the expected distance. The only working prototype in the early stages was the catapult design but it the ball trajectories weren’t consistent enough to implement it on the final design.

On the software side, the main assumption which caused our team many problems during the end of the course was the one that our independent modules would flawlessly come together if they worked independently. Most of our time was spent testing individual modules, until late in the semester, instead of continuously testing our robot’s entire routine, or at least as much as we had developed. This led to making many last-minute, untested changes, which eventually led to a poor performance on demo-day.

*What other issues/factors had an impact on the project?* One of our team members was not available for the last four days of the project due to an unexpected event. This put more pressure on the software team, because he was an important contributor for programming the robot.

*How did these affect the project progress?* Hardware affected the project for a small amount of time, because some software adjustments were needed to activate the crossbow motors. Nevertheless, because of our modular design, the software team could use the chassis to work on navigation. Furthermore, for most of the time dedicated to software development, the software team believed the project was moving along well, while not realizing that compatibility errors were creeping in without being detected. One example of something which really caused problems was that the way our navigation module would reset our odometer later caused many errors in our navigation because it used relative, rather than absolute coordinates to find the position it should navigate to.

*In particular, did the project run to the plan you had initially created?* No, because of the major hardware changes and software delay. We expected to run the robot by reusing most of the code done during the labs, but a lot of it had to be rewritten from scratch.

1. The budget

*What constraints did the budget place on your team?*

*How did initial planning for available resources and budget spending affect the development of the timeline?*

*Did you allocate resources to all the project tasks, i.e. all the way to 15 April, at the start of the project and use this to estimate the budget. If not, explain why not.*

*What would you have spent if there had been no limits on the budget and when in the process would extra budget have been useful?*

*Where were you weak in resources and what would you have done to resolve this issue if you had fewer budgetary constraints? At what point in the project could these extra resources have been brought in?*

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

*Was the process useful in achieving the goals?* The design process applied in the hardware division was very successful in achieving the goals. The robot was designed in different modules: chassis, launching mechanism, reload mechanism, defense and sensors. Each of these modules were prototyped and tested separately for a fast development phase. After meeting the requirements, the modules were implemented and tested directly on the robot. If a module failed, it would be taken out for fine adjustments if needed or replaced with a new design. At the end, the final robot had lots of complex parts but overall was robust.

*How would you modify the process to increase your probability of success?* We would put more resource on software, and set up precise deadlines to follow. The plan for testing our code modules as it was written should have followed, as well as writing the code itself, should have followed the order in which our robot would perform them, so as to always be able to test our full routine up to the point we had developed. An example of this was that a decent portion of time was spent developing and testing our shooting module, when our navigation and odometry correction were still sub-par. This created a gap of errors between our localization, the first module executed, and our shooting, which was the last element of our routine.

*Which parts of the process were the most difficult to implement and why?* The final design (the crossbow) was easy to design on paper. However, when it was time to build it, we ran into practical problems. Therefore, final part of hardware was difficult to implement. Material like rubber bands can stretch only to a certain point and can break after repetitive use. Often, some elastic would slip off the chassis or put too much stress on hit and break it. On the software side, our hardest challenge, which came late, was synchronizing our modules and making sure they synergized correctly, which was only partially achieved in the end.

*How much time was devoted to testing?*

*Was this at the subcomponent level or did you leave it all to the end?* Because of our modular approach, most of the testing was done for each module. However, we left most of the complete robot testing for the end.

*Were the tests you designed sufficient?* No, because we neglected to test the modules assembled together. In terms of time assigned to testing, we could probably have made our robot’s performance much better, with the same amount of testing, if we had continuously tested our new modules on top of the ones already built. This would have allowed us to fix issues as they came up instead of finding many incoherences late in the design process.

*How much time did you estimate full prototype (i.e. integration) testing would take?* At the start of the project, the full prototype testing was naively estimated at a week prior to the competition, more precisely 4 weeks in the project. This estimation would allowed use the remaining time to tweak and fix problems the final version of the robot.

*How much time did it actually take? If there was a difference, why?* However, with time management issues and problems encountered during the development phase, a fully working prototype was only delivered a day before the competition. This was due to a software implementation complication. If the software had always been fully integrated as they were developed, then testing the full prototype would have probably been much easier and faster. The last few days before our demo were mostly dedicated to this full prototype testing, while constantly having to trace back problems and retest our solution.

*How would you change your test design process to make it more effective?* More deadlines pyramidal process with basic tasks. Continuously integrating our modules would have brought up issues during the development phase, incrementally, and would have made our testing process much more efficient as well as representative of our robot’s real performance.

*What was the impact of the beta demo on your design process?* The beta demo clearly showed us something was very wrong with the robot. Nothing worked as expected. This is why in the hours following the demo, we worked extensively to solve each problem. After several hours of work, we manage to get the robot do what it was supposed to do during the demo. This showed that our design process didn’t sufficiently allow for testing. In the following weeks, we tried to test more extensively each module. Those efforts greatly improved the mechanical design of the robot. Because of that, mechanisms like the trigger and the ball reloading were fully functional before the competition.

1. The success of the Design (Robot) in meeting the original specifications and the performance requirements

*What is your impression of how the robot performed?* Restricting our impression on the competition, we think the robot performed poorly. Because the software was not precise enough and because no obstacle avoidance was implemented, the robot hit obstacles, therefore losing the odometer. It did not collect any ball. During the third round

*Did the robot perform as you expected – i.e. if you wrote down what you thought it would do before the demonstration, did it meet or exceed these expectations?* Yes, it did perform as expected. During the week preceding the competition, the tests we conducted all show significant errors in localization, navigation, and odometry correction.

*If the robot failed (i.e. did not meet all the performance requirements), why did it fail? Can you point to the sections of the documents that describe the decisions that led to the failure (provide the references to those decisions)?*

The main cause of failure of our robot was undeniably the non-functional software. The source of that issue can be traced back to our testing plan, which divided the testing of each module. Retrospectively, and this stands as the biggest lesson learned from our software team, it is that continuous integration should always be prioritized over individual testing and that, even in a modular design like ours, the development of modules should follow a strict order, namely one that makes continuous integration testing as natural as possible.**YO NAYEM DO YOU SEE THIS????**

1. Conclusions

*What did you learn from this course?* To conclude, we learn from this course basic engineering skills useful for any design project. We learned how the design process works//////. Furthermore, we learnt how to work as a team with a common objective. Communication is key to any successful and healthy project. Since any project has constraints, time management and prioritization of tasks compliment the communication skill to ensure optimal progress.

*Explain why a clear, effective and controlled process is necessary when working in a team and what it helped you achieve.* Everyone in a team has its own idea on how the design process should run. A clear, effective and controlled process is needed to provide a structure that each team member must follow.

*Is any of it applicable to other courses you might take?* The lessons learnt throughout this course are applicable for other courses in the ECSE curriculum but also for any future group projects.

*If so, what and why?(name the courses)* Courses like Intro to Software Engineering (ECSE 321), Digital System Design (ECSE 323) or Design Project 1 & 2 (ECSE 456 & 457) have heavy class projects with various teammates. With limited time and resources, strong time management skills are required to be able to showcase a presentable version of the product. However, Computer Engineering (ECSE 322) has an extensive written project with 5 members in a group. Therefore, in this scenario, the team has to have good communication competence to accomplish a coherent and comprehensive document.

*What would you change in what you did if you were doing it over? (important!)* Like mentioned in section ///, we would set precise deadlines that must be respected. This would create a list of priorities and permit better time management. The modular approach we used was great for working in parallel on different design, but was also a trap in the sense that it was difficult to set priorities. We were trying to do to much at the same time. Next time we do a project of this size, to prevent getting lost /////

**Two printed copies of this report should be handed in at the lecture on April 10 along with a soft copy on the CDs, DVDs or Memory keys you are handing in. Please note that 2 copies of the CD, DVD, or Memory key MUST be provided on April 10.**

Please note that the printed copies **MUST BE SIGNED BY ALL MEMBERS OF THE TEAM TO INDICATE THAT THEY ALL AGREE BOTH WITH THE CONTENTS OF THE REPORT AND THE INFORMATION HANDED IN ON THE CD, DVD OR MEMORY KEY.**

The report MUST contain the following paragraph above the signatures:

“The undersigned members of team *xx* 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.” (please replace “xx” with your team number)

DAL/DG

29 March 2017