**DPM Fall 2017 Final Report**

Team 06

Alex Hale (260672475), Xianyi Zhan (260672960), Justin Tremblay (260743650), Joshua Inscoe (260659992), Xu Hai (260661832), Frederic Cyr (260674545)

ECSE 211 – Design Principles and Methods

December 1, 2017

Introduction

* Review the goals for the project
  + Main reasons for doing it
    - From outline: engineering process, design specifications, parameters, optimization, implementation, troubleshooting and refinement, project management (scheduling, risk analysis, project control), case studies, design examples, project
    - Learn the design process: systematic procedure that begins with the formulation of a precise specification of the project at hand and ends with the specification of a procedure and/or mechanism that meets the requirements outlined in the problem specification
      * Identify the constraints that surround the problem
      * Manipulate the design variables such that a solution is found that meets the constraints in an optimal fashion
    - Systems perspective: a problem is abstracted as a system comprised of inputs, outputs, and a process model that implements the desired input-output behaviour
    - Learn a better idea of the engineering profession and an understanding of why a particular academic background is needed
  + What was the project intended to achieve
* Phase 1: R&D
  + 5 one-week experiments covering topics essential to the design problem
    - Wall follower, odometry, navigation and obstacle avoidance, localization, zipline
* Phase 2: design phase
  + Design, implement, and test machines for the final competition
  + Fully document all activity related to the project
  + Meet on a weekly basis with the “client” to prove ongoing progress

Team Organization

* How were tasks allocated?
  + First, roles were assigned based on experience and interest
  + It was determined that software would need more of a focus than hardware, but we didn’t have enough confident software engineers
* How was the initial GANTT chart designed?
  + Used the provided template from the professor
  + Broke all tasks down into smaller tasks, max ~3 days each
  + Assigned each task to one of the team members
* What information was used to estimate the initial task breakdown?
  + Experience from hours taken to complete laboratory tasks
  + Advice from TAs, professors, and friends who had previously taken the course
  + Wild guesses
* Were any guidelines followed in developing the first version of the chart?
  + Balance all team members’ hours (roughly equivalent total number of hours for the project
  + Ensure that one team member doesn’t have too much work or too little work at a given time

Issues Encountered

* Correctly identified dependencies?
  + Assuming this means that the order of everything on the GANTT chart was correct
    - No – since we had fewer software team members than anticipated, there were some groups of tasks that were simultaneous on the GANTT chart, but subsequently completed
    - This is what caused the last-minute searching algorithm creation at the end of the project
* Which dependencies contributed to the critical path of the project
  + Go check critical path of GANTT chart
* What initial ideas turned out to be based on incorrect assumptions
  + Go through GANTT chart and recap which ones were the wrong length, wrong order, or incorrectly simultaneous with other tasks
* What other issues/factors had an impact on the project, + their effect
  + Teamwork and communication related to software development
    - Software was nearly entirely dependent on one person, the rest of the team couldn’t help even if they wanted to, since they had never even looked at any of it
  + Lack of communication in-person: online communication was great, but not always perfectly clear
    - Team members weren’t the friendliest toward each other, meaning that nobody wanted to have team meetings cause they sucked to go to, so team meetings didn’t happen, so communication wasn’t perfect
  + Arrogance
    - Lead to the same problems as #1 – nobody else could work on the code
  + The GANTT chart wasn’t followed as closely as it should’ve been
    - Sometimes people were unclear about what they should be working on
      * Need more details for this
* Did the project run to plan?
  + Surprisingly – yeah! Mostly.
    - Luckily, the GANTT chart followed the same logical procedure as we probably would’ve followed if there had been no GANTT chart, so there weren’t too many issues with this

Budget

* What constraints did the budget place?
  + Since we were under budget the entire project, not many
  + Forced people to think about what they were working on, since it would be documented
* How did initial resource/budget planning affect the timeline development?
  + Uhh…
* At the start of the project, did we allocate all project tasks and use this to estimate budget expenditure?
  + Ask Josh – I believe this did happen, yes
* If there had been no budget limit, would we have used more? Less? Would more budget have been useful?
  + Since we finished the project under budget, more budget would not have been useful
  + In fact, if there were no budget limit, our budget usage would probably have been lower
    - Some team members may have inflated hours, especially near the end of the project, to match what they thought was expected of them
    - Some team members may have worked fewer hours because:
      * Nobody would have known
      * If they finished their assigned tasks in fewer than the expected number of hours, they may have felt compelled to work more to meet the “expected” hours per week
  + The budget served more as a guideline of how many hours to work per week rather than how many hours to work for the entire project
* Where were we weak in resources?
  + We did not have enough hours spent on software, and by extension, software testing
    - One available software engineer was already spending the full budget most weeks on documentation
    - The other available software engineer worked mostly only on GANTT / budget for the first 4 weeks of the project, then spent tons of time on software right at the end: not ideal, because it didn’t give the testing team the chance to conduct any software tests on that code
  + Fewer budgetary constraints would have allowed the documentation manager and project manager to contribute more consistently on software (i.e. first four weeks)
    - However, this would cause a large unbalance between those three members and the other three members
      * The remaining three members did not have the required skills to contribute to the software development7

The Design Process: Useful, or not useful?

* Did the process help us achieve our goals?
  + Yup
  + It gave people a structure to follow, instead of working on whatever they felt was necessary
* How would we modify the process?
  + Uhh…..
* Which parts of the process were the most difficult to implement
  + People actually following the GANTT chart exactly
    - Would require the project manager to issue constant reminders, which didn’t happen
  + Consistent effort – team members sometimes put in lots of effort in one week and not so much effort the next, which (assuming that wasn’t planned in the GANTT chart) didn’t help follow the GANTT chart
* Time devoted to testing + was it enough + was it spread out enough, or left till the end?
  + Use the percentage from the poster, or update it if necessary
  + The testing in the first 4.5 weeks was awesome – there was lots of it done about individual components of the robot, and the results got given back to the respective teams to make changes
  + The testing in the last 1.5 weeks was not great, because the components that required testing were not yet finished
    - The searching algorithm needed to be tested, but it was not complete until right before the competition, so it didn’t get to be tested
    - Full-game testing (starting from both teams) needed to happen, but since the searching algorithm wasn’t done, it didn’t happen
      * What *should’ve* happed was that the testing team ran the full-game simulations while ignoring the searching phase entirely (i.e. navigate the robot to the search zone then navigate away without searching), but due to a lack of clear communication between the software team and the testing team about the state of the robot, this didn’t happen
* Were the designed tests sufficient (i.e. did they produce the required information about the condition of the robot?)
  + Mostly, yes. As mentioned above, the preliminary tests about individual components of the robot were sufficient. However, the tests about the final competition performance of the robot were not sufficient
    - The professors warned all semester that integration testing was the most important. We had this in mind, but since the searching algorithm was completed so late and the necessary adjustments weren’t made due to a lack of communication, the necessary amount of integration testing was not completed.
* How much time did we estimate that FULL integration testing would take?
  + (check GANTT chart)
* Was it enough?
  + Who knows! We didn’t get there. Based on the length of time that the earlier testing phases took, it was probably enough
  + Since the task earlier on the GANTT chart (searching) didn’t get completed until the last minute, we never even got to the final testing phase
* How would we change the test design process to make it more effective?
  + distribute the report-writing responsibilities. The tester can take notes, and somebody else can write the report
    - especially applicable for those who struggle with writing – distribute the task to someone who is a better writer
  + only one tester is required, having multiple testers at the same time is a waste of resources unless absolutely required (rare!)
    - e.g. the testing engineer doesn’t know how to program the testing script, so the software engineer has to be there
      * even in this case, the software engineer could just perform the test alone
* impact of beta demo on design process
  + gave a deadline to meet that was in advance of the final deadline => forced things to be on track
  + caused some panic: robot completely failed the demo because of the two random coordinates that were supplied
    - kicked testing team into gear – once they completed the in-house beta demo, calm was restored

Design Success

* how do we think the robot performed?
  + Compared to the other teams, quite well!
    - our searching algorithm was too slow to be effective, so we just ignored it, but so did other teams
      * the only team seen to “succeed” in capturing a block didn’t even have their colour sensor on – the robot just happened to beep when detecting a line near the correctly coloured block, so the judges awarded the points
    - some bugs were present at the start of competition that really should’ve been ironed out beforehand
      * typo in the code that turned the robot into the wall in the first demo
      * infinite localization loop that plagued the robot after the zipline dismount in the second and third demo
    - navigation accuracy was a shade too low
      * final run: robot got stuck when mounting the zip line
      * final run for fun: after being given a little boost to mount, the robot navigated back to the home corner, but was one tile off
* exceeded expectations, or big letdown?
  + Totally exceeded expectations. Only one team member had touched the code or tested the robot as they worked on the searching algorithm, so only they knew how it was going to perform. And they informed the team that they had “very low expectations”, without giving any details (again, lack of communication!)
    - The robot localized, navigated to the bridge, crossed the bridge, navigated to the search zone, promptly said goodbye and navigated to the start of the zip line, relocalized, mounted / traversed / dismounted, relocalized, then got stuck in an infinite localization loop. Not a bad performance for a robot with “very low expectations”!
* Why did the failing components of the robot fail?
  + Search: the decision was made *not* to turn the sensor motor in favour of simplicity, which made the searching algorithm very slow. In the interest of using remaining time to perform more components, the time allotted for searching was very low, and the robot didn’t ever properly search the zone.
  + Returning home: the navigation was not quite accurate enough (in fact, there was no odometry correction during navigation, again in favour of simplicity). The only form of error correction was relocalization, and since there was no localization performed when nearing the home corner, the robot was off by one tile
  + TODO: reference sections n the documentation where the decisions were made that lead to these two failures

Conclusions

* What did we learn from this course?
  + Need a strong team manager that will get the team to work together even when they aren’t friends
  + Follow the outlined procedure (in this case, GANTT chart) more closely
  + When changes to the procedure (GANTT) are required, discuss as a team to determine options, and let the manager make the final call
* Why is a (description) process necessary when working with a team; what did it help us achieve?
  + Each of the team members need to know what they’re doing at any given time
  + Since the components fit together in a specific order, they need to be completed in order
  + Avoid disagreements about what someone should be working on
* Is anything we learned applicable to other courses? Which ones (specifically)? Why?
  + Todo
* What would we change if we were to do it again (lots of
  + Keep the final competition in mind when performing labs
    - Remind yourself that the labs are R&D for the project; they’re not just for their own marks
  + Allocate team members to a wider variety of tasks
    - Essentially, we needed more people on software, but 2/3 people available were taken by documentation/PM and couldn’t help as much as should’ve
    - If HW/testing people had been able to help more with docs/PM, the docs/PM people could’ve helped more with software
  + More in-person meetings
    - Make them happen. Even if they’re short, they produce more good things than expected. Discussing over Slack only works if everybody is *descriptive and clear*!
  + Make GANTT chart modifications official
    - Instead of just discussing a change, actually make the change right away, so everybody is perfectly clear on what their new task is
  + Pay closer attention to the GANTT chart
    - Constant (daily) reminders from the PM aren’t annoying, they’re helpful
  + If you’re unsure about how a process is going, speak up.
    - If the person isn’t clear enough, push them for a better answer
  + Ask for more clarifications from professors/TAs

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.