8-2020

A longitudinal study of a capstone course

Benjamin GAN
Singapore Management University, benjamingan@smu.edu.sg

Eng Lieh OUH
Singapore Management University, elouh@smu.edu.sg

Yin Yin Fiona LEE
Singapore Management University, fionalee@smu.edu.sg

A Longitudinal Study of a Capstone Course

Completed Research

Benjamin Kok Siew Gan
Singapore Management University
benjamingan@smu.edu.sg

Eng Lieh Ouh
Singapore Management University
elouh@smu.edu.sg

Yin Yin Fiona Lee
Singapore Management University
fionalee@smu.edu.sg

Abstract

This is a 7 years study on a capstone course completed by 1700+ students for 200+ organizations involving 300+ projects. Student teams deliver a system to solve real-world problems proposed by industry partners. We want to understand what independent variables influence student performance. We analyzed the deployment status of systems delivered, the type of organization/industry, the number of meetings and the technology used. Our results show some organization value proof of concept over fully deployed systems, student strengths are in Infocomm and Finance projects, the number of meetings is a weak correlation to performance and best performing projects are fully deployed on iOS using Microsoft technologies. Faculty can use these insights to focus on factors such as the type of industry projects and technology used. They may not be overly concern if student teams have fewer meetings or did not fully deployed their system, so long as they create value.

Keywords

Capstone course, longitudinal study, student performance, online survey, correlation, t-test, Mann Whitney U test, project-based learning, real-world problems, industry partner, deployment status, organization type, industry classification, number of meetings, application platform, database, tools, course designer.

Introduction

The objective of this study is about understanding the factors that influence student performance in a capstone course. The motivation is to help faculty understand which factors to focus on when designing their capstone course. The setting is an undergraduate capstone course where students deliver a system to solve a real-world problem provided by an industry partner. We collected data using online surveys for 7 years and conducted quantitative analysis to derive insights on the factors that we hypothesized as potential influences. For example, we look at how our student performs in different industry projects. Our analysis tells us that they performed better in Infocomm and Finance projects compare to Health projects. Using this information, a capstone course designer with similar students may feel more reassured when assigning student teams to Infocomm and Finance projects. However, the faculty may want to align students with experience in Healthcare or provide some background foundation before assigning them to Health projects.

We begin with a literature review of research on the capstone course in the next section. This will continue with our research method and results. In conclusion, we summarized the insights from our quantitative analysis and stated the limitation of this study.

Literature Review

There have been many studies on capstone courses. We categorized them into 5 categories: cross-disciplinary/cross-institution, course assessment, project management, project sourcing and technology-aided studies.

The following are cross-disciplinary/cross-institution studies for capstone courses. (Howe et al. 2017) surveyed 522 respondents at 256 institutions on implementation strategies for engineering capstone design programs across the U.S. It captured the state of capstone design education, drew comparisons across engineering disciplines and highlighted changes over 20 years. (Lee and Loton 2015) synthesized capstone curriculum across wider disciplines, outside of ICT and engineering. (Ieta and Manseur 2013) compared three different higher education institutions with regard to design, supervision and implementation of capstone courses. The next category is course assessment studies for capstone courses. (Boesch and Gan 2014) addressed the tradeoffs in allowing teams the flexibility of choosing different capstone projects and keeping assessment consistent. (Ochs and Getzler-Lin 2012) proposed that industry sponsors have direct impact on how students are assessed and suggested a formative assessment during “gradable moments”. (McKenzie et al. 2004) reported on how capstone design courses in engineering institution conduct assessment practices and recommended changes.

The third category is project management studies for capstone course. (Baird and Riggins 2012) proposed a hybrid project management method that combines waterfall and agile project management techniques. (Perez et al. 2012) identified six styles of supervising capstone projects: student alone, execution-focused, global supervision, management focused, technological mentoring, and process focused. (Aller et al. 2008) focused on team formation using a mingling method to increase performance and motivation. The fourth category is project sourcing studies for capstone courses. (Gan and Shankararaman 2015) studied two project sourcing model: internal and external to the university. Challenges for internal project sourcing is getting university started on automation and for external is clear communication with external stakeholders including Non-Disclosure Agreement and Memorandum of Understanding. (Moshkovich 2012) compared research vs development projects. (Grant et al. 2010) emphasized the need to source projects from external agencies so that students gain “real world’ exposure. (Patten and Keane 2009) studied experiential service-learning using community partners in a capstone IT course.

The final category is technology-aided capstone course studies. (Fan 2018) presented an orchestrated way to conduct agile sprint reviews in undergraduate capstone projects. (Neyem et al. 2014, 2017) created the Educational Software Tool (EST) to provide a college-compatible integral project tracking solution that allowed cooperative authoring, commenting and annotating shared documents as a group activity. (Ohtsuki et al. 2016) created Automated Learning and Evaluation Cycle Support System (ALECSS) to automatically check the code submitted by student teams and provide feedback on the quality of student code generated by the DevOps tools.

When designing a typical capstone course, it is hard to know how many “gradable moments”, supervising and project management is enough, what type of projects should be sourced (internal, external, research, development, or community service projects) and what technology works best. Instead of cross-disciplinary or cross-institution, our study dives deep into an Information System capstone course in an institution to give an in-depth perspective. Coupled as a longitudinal study similar to (Howe et al. 2017; Lee and Loton 2015), we have large enough sample data to analyze significant dependencies. We investigated independent variables to analyze if they influenced the team performance. This research can answer whether project deployment status influences the team performance; what type of industry partners to source for projects; how many team meetings works best; and what type of technologies are used by the best projects for agile reviews, project tracking or DevOps.

Research Method

Our course structure adopts a project-based learning pedagogy (Rob 2007; Vartiainen 2013). Students learn by implementing a system that creates value for the industry partner. Teams of 4-6 students use an agile approach (Beck et al. 2001) defined as a modified SCRUM (Sutherland and Schewaber 2013) or the Lean Startup Methodology (Ries 2011). All teams are required to do multiple iterations of user testing.

We collected data from January 2012 to May 2019, a total of 15 terms. Online surveys were sent to all teams near the end of the project during the research period, a total of 379 teams. We received 346 completed responses. Partial or incomplete responses were removed from the research. These surveys involved 346 teams, multiply by an average of 5 students in each team, which comes to an estimated 1,730 students.

Our general research question is "What factors influences student performance in our capstone course?" We measure student performance by their team grade in terms of marks from 0 to 100. The grade distribution is shown in table 1. For example, 79 gets a grade of B+ while 80 will get a grade of A-.

Marks	50	53	60	63	66	70	74	77	80	83	86	100
Grade	F	D	D+	C-	C	C+	B-	B	B+	A-	A	A+

Table 1. University Grade Distribution

We wanted to know if successful deployment and more meetings influence better student performance. We used the number of meetings to track project management, a grading criterion. The quality of the capstone course learning experience depends heavily on the quality of the project. We wanted to understand which organization and industry types provided suitable quality projects where our student performed well. Similarly, we wanted to know which technologies were used in the project and how our student performed with these technologies. The independent variables are measured against team grade, the dependent variable. The faculty who graded the projects could be a confounding variable that influenced the grade, we reduced this influence by using multiple reviewers in different stages (proposal, acceptance, midterm and finals) of the grading process and a template grading sheet with a standard grading rubric based on clarification of the project (description, scope, use cases), project plan & management (team member roles, schedule, stakeholder management, process), quality of project (quality attributes, architecture, technical complexity, user testing, deployment) and reflection (skills acquired and learning outcome). In the following subsections, we present answers to our research questions for each factor.

Deployment Status

Our capstone course requires an information system as a deliverable. The deployment status of the information system is categorized as: Aborted, To-Be Deployed, Proof of Concept, To-Be Further Developed and Fully Deployed. The success of this deployment status is ordered from 1 for Aborted (or failure), 2 for To-Be Deployed (or still in development), 3 for Proof of Concept (or have deployed enough features to demonstrate the project idea), 4 for To-Be Further Developed (or deployed with only nice to have features to be further developed) to 5 for Fully Deployed (or successfully deploy the entire project scope). Figure 1 bar chart shows the count of projects for each deployment status over the research period. We did a correlation of the deployment status with the team marks. The correlation coefficient is 0.284 which suggest a weak positive linear relationship. The regression square is 0.08 supporting the weak correlation.

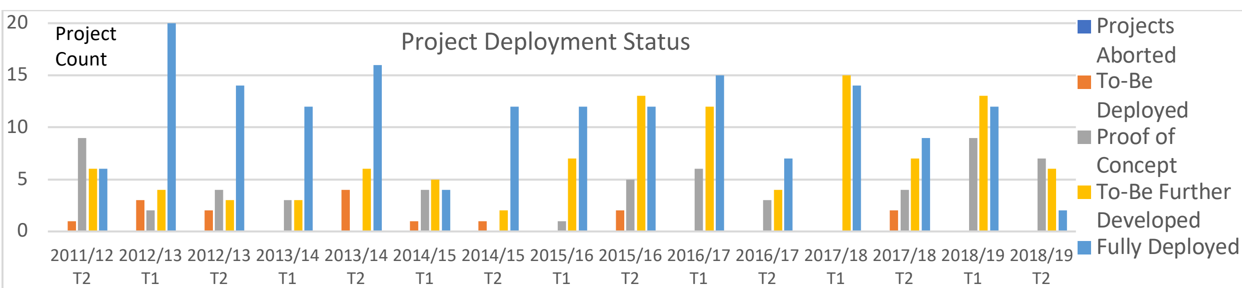

Figure 1. Project Deployment Status.

Figure 2 scatter plot shows the distribution of marks across the deployment score. The vertical axis is the deployment score from 1 Aborted to 5 Fully Deployed. The horizontal axis is the team marks. Most team marks fall between 65 to 90 (or a grade of C to A+). Majority (167) projects were Fully Deployed with an average of 81.5% marks. This is followed by 106 To-Be Further Developed projects with an average of 78.5% marks. Next, 57 projects were Proof of Concept with an average of 79.8% marks. Finally, 16 To-Be Deployed projects had an average of 76.8% marks. None were aborted. Some projects proposed by our research partners find value from the insights teams derived from their prototype test. These research partners did not require a fully deployed system. They were from University, MNC, SME, or even startup organizations. Projects from our research partners drove up the average marks for Proof of Concept projects, 79.8% which is higher than the average marks for To-Be Further Developed, 78.5%. This is evident when we comb the data and found 3 teams with A+ in Proof of Concept deployment status compare to 1 team with A+ in the

To-Be Further Deployed status. The resulting insight is while deployment status influences the team's grade, it is a weak influence. The correlation could be stronger if we removed research partner projects. The main contribution to faculty is not to be an overly concern if teams have not progress on deployment status if their industry partner provided a research project.

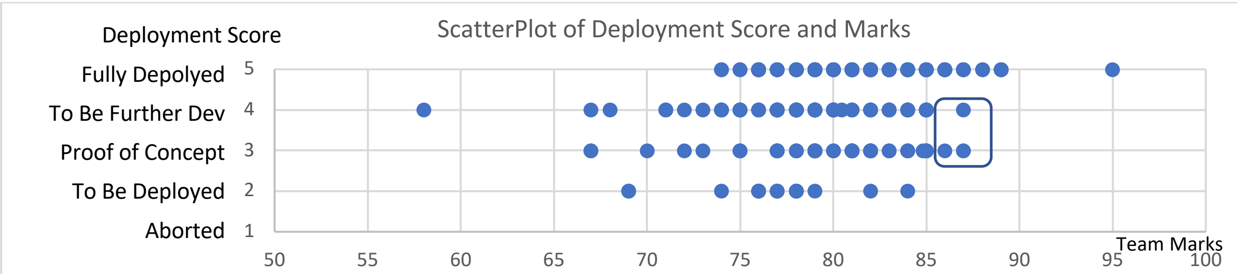

Figure 2. Scatter Plot of Deployment Score and Team Marks.

Organization and Industry Type

The quality of the learning experience in our capstone course depends heavily on the quality of industry projects. When sourcing for capstone projects, we would like to understand what characteristics of the industry partners are important in the sourcing process. We collected information such as the Organization Type and the Industry Type.

The organization can be categorized into 7 categories: MNC, SME, Startup, Government (Gov), University, VWO/Non-Profit, and Team's own project. The businesses are categorized as Multinational Corporation (MNC), Small Medium Enterprise (SME) and Startups. The Government category is institutes of higher learning (excluding our university), government agencies or statutory boards. Many of our projects came internally from within our university. We gave these projects its own category (University) to help us study their specific influence on our student performance. These projects can be proposed by our faculty, laboratories and administrations such as the various schools and centers that provide supporting services like libraries, gym, student life, finance, etc. The VWO/Non-Profit category is a loosely termed community organizations such as Voluntary Welfare Organization, Non-Profit Organization, Non-Government Organization, etc. The last category (Team) is teams working on their own projects. They must pitch their ideas to be accepted by mentors to qualify as a capstone project. Mentors were investors, entrepreneurs, or leaders in the project field. Figure 3 pie chart shows the count of projects for each Organization Type. The legend shows the average marks for projects from that organization type sorted in descending order.

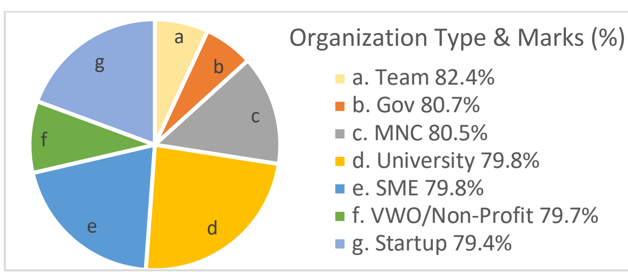

Figure 3. Project Organization Type and the Corresponding Average Marks

The University projects represented the largest organization category with 82 projects, followed by SME with 70 projects. Startups were 67, MNC 49, VWO 32, Team's own project 24 and Gov at 22. Most of our University projects came from four labs: 15, 12, 8 and 7 projects for each lab. Six faculty had multiple projects ranging from 2 to 6 projects. Two government institutions and one MNC had 5 projects. Overall, around 60% projects came from industry partners with a single project and 40% were from industry partners with multiple projects. Forty percent shows that these projects created enough value for them to propose another project.

MNC projects did slightly better than SME or Startups, with teams' average mark of 80.5% compares to 79.8% and 79.4% respectively. We plotted histograms based on the grade distribution in table 1. The project histograms are mostly very slightly skewed to the right. We calculated both T-Test Two-Sample Unequal Variances and Mann Whitney U test for MNC with SME and for MNC with Startup. The p-values are all above 0.05 which means we are unable to reject the null hypothesis. The slightly higher MNC average marks are likely due to chance. Another comparison is the team's own project has the highest average marks at 82.4% compared to university projects at 79.8%. Similarly, team marks distribution in histogram is very slightly skewed to the right. However, the T-Test and Mann Whitney U test p-values (0.01818 and 0.01315 respectively) are lower than 0.05. This is significant and we can reject the null hypothesis. It is not by chance that projects the team proposed did better than university projects. Lastly, government projects did better at 80.7% compared to VWO projects at 79.7%. However, the p-values are all above 0.05 and the sample size is small for significance.

The Industry Type is derived from the Singapore Standard Industry Classification (SSIC 2015). We used 16 classifications. Figure 4 pie chart shows the count of projects for each industry classification/type. The legend shows the average marks for that industry type sorted in descending order.

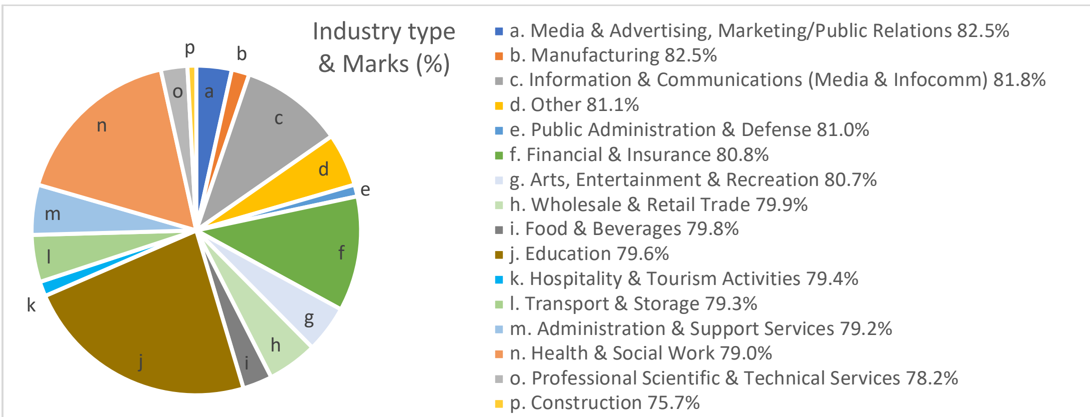

Figure 4. Project Industry Type and the Corresponding Average Marks.

Education represented the largest (80) industry type. However, please note that some university labs with industry collaboration were considered non education based on the nature of the lab. Health had 59 projects, Financial had 39 projects and Infocomm had 35 projects. We suspected some students were confused with the different industry classification and filled in Others when they were unsure. The industry that did best in terms of average marks was Media & Advertising with 82.5% with only 12 projects. The rest of the industry type with an A grade were Manufacturing, Infocomm, Other, Public Admin, Finance and Arts. Construction has the lowest marks at 75.7% but it only has 3 projects. The sample size is too small. We compared industry with >30 sample size, Infocomm (81.8% for 35 projects) did better than Health (79% for 59 projects). The histogram is slightly skewed to the right, so we use Mann Whitney U test with a p-value of 0.0022 which is lower than 0.05. We rejected the null hypothesis. This result may be obvious since our IS students would be more familiar with Infocomm domain than the Health industry. We compared less obvious industries: Finance with Education and Finance with Health. Both have non-normal distributions. We did Mann Whitney U test to get p-values of 0.0586 and 0.0178 respectively. No significance for Finance and Education but significance for Finance and Health. Our students did better in Finance than Health and this is not by chance based on our p-value.

Fortunately, we have more projects then student teams. This allowed us to be more selective to improve the quality of projects for our capstone course. The contribution from the insights in this analysis is to help us choose the projects based on how well the students did. We can align the right industry projects with our student skill sets. For example, our foundation courses may provide the necessary case studies to understand the industry type better. A recent emergent research forum (López 2019) concluded that tech startup projects finish with the end of the classes. Despite the potential of entrepreneurial class projects across multiple IS classes, there is little evidence of the continuation of these projects after the class ended. This is a pity as there are few alignments between our IS courses as well. In order to get evidence of alignment, we hope insights here can help identify skill sets that our students are compatible with. Choosing the right industry type projects that our students did well is to tap on our strength. The statistics also identify our weakness in some industries. If we want to improve the skills of our students in these industries, we can encourage more foundation courses to use case studies related to these industries.

In summary, teams who did their own projects have the highest average mark and did significantly better when compared to university projects. This may be partly due to the stringent criteria our mentor use, resulting in fewer team projects passing the acceptance stage. It may also be the passion student have when solving their own problem. Faculty supervising these teams may choose a hands-off approach for the team to excel instead of closely guiding approach necessary on a weaker team. We find no significant difference between projects of other organization types. Many of our university labs, faculty, government institution and MNC must find value in our undergraduate capstone projects to propose multiple capstone projects. Repeat projects do slightly better too with average marks of 80.4% compare to 79.8% for single projects. Faculty should encourage industry partners to continue a project or propose new projects. On industry types, our students did well in where they have good domain knowledge, particularly Infocomm and Finance. We can be selective on choosing the right projects based on the Organization and Industry Types.

Number of Meetings

We assume that the more meetings teams had, the more work they have done together leading to better grades. From August 2016 to May of 2019, a total of 6 terms, we collected the number of meetings teams conducted in each category (with Faculty Supervisor, with Industry Partner and within the Team) and their feedback ratings in each category (for Faculty Supervisor, for Industry Partner and for the Course). Figure 5 bar charts show the average number of meetings and the corresponding feedback ratings.

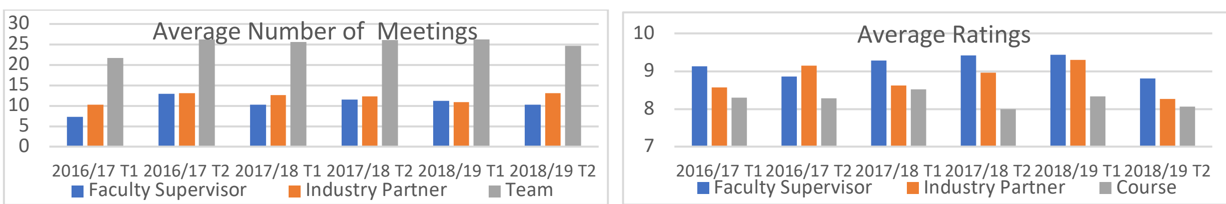

Figure 5. Average number of Meetings and the corresponding Ratings.

We calculated the correlations between the number of meetings and the ratings for each category: Faculty Supervisor (0.034), Industry Partner (0.137) and Team/Course (0.066). We found very weak positive correlation. This may be explained that sometimes, in teams with low productivity, they need to meet more often. Thus, the number of meetings may not correspond to better feedback ratings. We correlate the number of meetings with team marks: Faculty Supervisor (0.120), Industry Partner (0.099) and Team (0.118). Similarly, these are very weak positive correlations. In summary, we cannot draw any conclusion from the number of meetings teams had to improve their performance or feedback. Faculty should not just look at the number of meetings as an indicator for student performance.

Application Platform, Database and Tools Used

We are interested to understand the technology trend of our capstone project in terms of application platforms, databases and tools. Although we teach Java and Java Script mainly on the web platform using MySQL database, other courses introduce new technologies. For the capstone course, students were free to choose the technology that makes sense to the project. For example, the Industry Partner may specify a technology that they are currently using or have no budget and require free public domain technology.

Figure 6 bar chart shows the percentage of projects using a particular application platform for that term. The majority of projects (82%) used the web platform which included web mobile, access from a mobile device's web browser. Native Android or hybrid apps running on Android comes in a distant second at 26%, iOS or Hybrid iOS at 17% and Desktop applications at 9%. When we compared to web, mobile and desktop platforms, we have an average of 82%, 43% and 9% respectively. (IDC 2020) shows a market share worldwide smartphone shipments of roughly 85% for Android and less than 15% for iOS from 2017-2019, a ratio of 85:15. Our 26% for Android compared to 17% for iOS is a ratio of 26:17. This could had been due to the increasingly popular cross platform mobile frameworks such as Flutter, React Native, Ionic, Framework 7 and PhoneGap (renamed as Apache Cordova), making it easier to develop on both popular mobile platform. Some projects implemented multiple platforms as a solution. One team implemented all four platforms, supporting both mobile platforms, web for the public users and desktop for the staff. Projects on iOS platform was the best performer with an average of 81.2% team marks. Compared to web platform, using T-Test Two-Sample Unequal Variances with normal distribution, the p-value is 0.047 which is significant (<0.05).

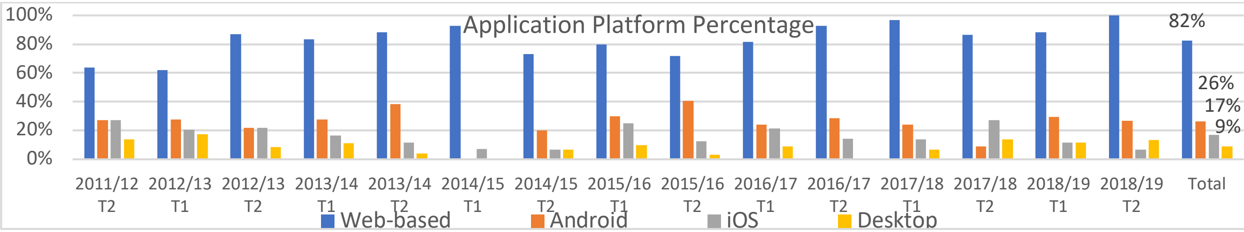

Figure 6. Percentage of projects using each Application Platform.

Figure 7 bar chart shows the number of projects with 2 platforms and with more than 2 platforms. When the cross platform mobile frameworks started with PhoneGap and Ionic, students experimented with them during 2013-2014. As they encountered problems with the first generation of cross platform frameworks, they suspended using them in 2014-2015. When more stable versions of PhoneGap, Ionic, React Native and Flutter were introduced, more students began to use them again. In 2014-2015, we had a low 7% of the projects with more than 1 platform. The rest of the terms, we have 20%-40% with multiple platforms. The average mark for projects with multiple platforms was 80.4%, which was a little higher than the overall project average mark of 80.1% This may indicate a small acknowledgement for complexity or effort in delivering multiple platforms. The T-test p-value for single vs multiple platforms is not significant.

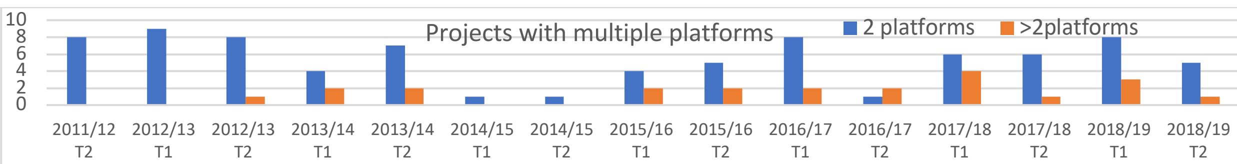

Figure 7. Projects with multiple platforms.

Figure 8 bar chart shows the percentage of projects using a particular database. The majority of projects (66%) used MySQL. The rest were Postgres at 9%, Microsoft SQL at 8% and MongoDB at 7%. Not included in the chart are databases or cloud storage frameworks used by <3% of projects. They are Google Cloud storage for App Engine, Firebase, Amazon S3 Buckets, Azure Database, Microsoft Access, etc. During our research period, all terms had higher number of MySQL projects compared to the sum of the projects using non-MySQL databases, except one. In 2015/16 T1 we had 30% projects used MySQL and 38% used non-MySQL. Of the 38%, both Postgres and MongoDB were used by 15% of the projects and Microsoft SQL were used by 10% of the projects. The average marks for projects with Postgres (81.1%), MSSQL (81.4%) and MongoDB (81%) were all higher than projects with MySQL (79.7%). Using T-Test Two-Sample Unequal Variances with normal distribution, the p-value is 0.0036 which is significant. Faculty had rewarded projects that used a different database than the one we taught except for cloud storage such as Google Cloud storage for App Engine which had the lowest average marks of 78.9%.

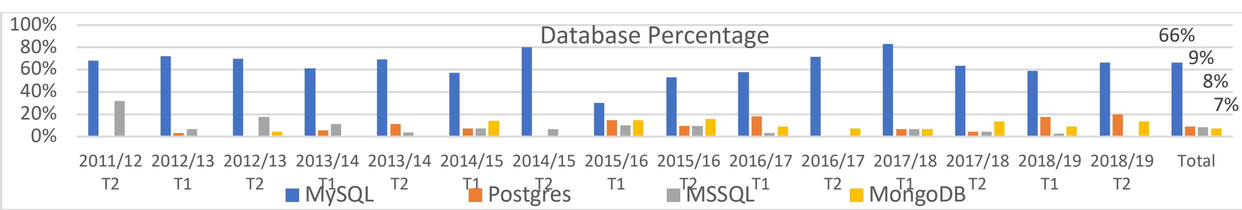

Figure 8. Percentage of projects using each Database.

Figure 9 bar chart shows the percentage of projects using a particular tool for that term. The majority of projects (36%) used NetBeans. This was followed by Eclipse (22%), Google tools (21%), Microsoft Visual Studio (VS) (18%), Facebook tools (15%) and Bootstrap (13%). NetBeans, Eclipse and VS are IDE (Integrated Development Environment) which provides features to edit, browse, build and debug code.

Additional features are code generation, version control, GUI frameworks, documentation, etc. Non-IDE tools that were popular were from Google and Facebook. The most popular framework was a web front-end framework Bootstrap with JavaScript and jQuery plugins. The rest of the tools not in the chart were Microsoft other tools, PyCharm (a Python IDE), IntelliJ IDE, Github and Dreamweaver. Two tools had declining trend lines. Eclipse IDE became less popular with a regression square R^2 at 0.456. Facebook tools decreased in popularity at a higher rate of R^2 at 0.641. Projects with VS IDE performed best (81.1%). Compared to NetBeans (79.8%), using T-Test Two-Sample Unequal Variances with normal distribution, the p-value is 0.0313 which is significant.

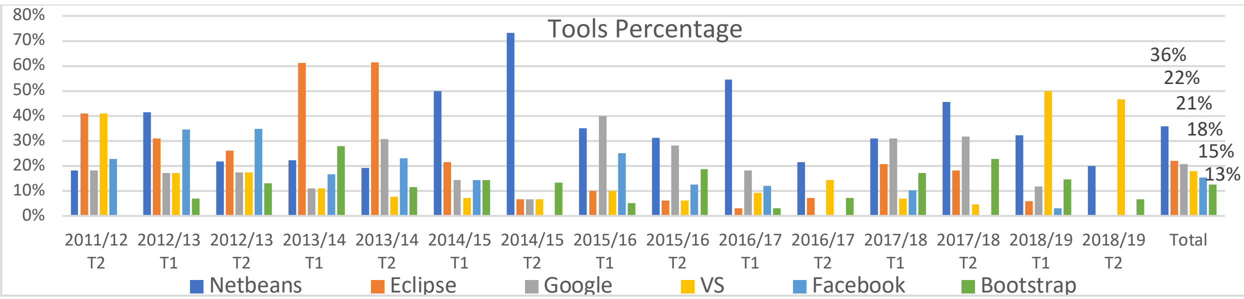

Figure 9. Percentage of projects using each Tool.

In summary, the most popular technologies were web-based platform using MySQL database and NetBeans IDE. Although implementing multiple platforms required additional work and complexity with no significant evidence of getting better grades, there were still 29% of projects implemented on multiple platforms. This could be in part contributed to the popularity of cross platform mobile frameworks. It is not clear if the multi-platform implementation is due to students wanting to learn a new tool or a requirement from the industry partners. If the industry partner does not require additional platform, students can take this insight of no significantly better grade to avoid implementing the multi-platform. The dominance of MySQL database in our university was clear from the statistics. Except for 1 term, the sum of projects using non-MySQL database was always lower. However, faculty consistently reward students with better grade when using non-MySQL. Faculty may want to reconsider rewarding effort over the suitability of the database. The three most popular IDEs were NetBeans, Eclipse and Microsoft Visual Studio. The two most popular non-IDE tools were from Google and Facebook. The single most popular framework was Bootstrap. Eclipse and Facebook saw declining interest. Our survey asked students to list the 3rd party tools and frameworks. We suspect some teams may not fill in tools they used. It might be better to list some popular tools as selection in our survey. Best performance projects were projects running on iOS built using Microsoft tools, such as MSSQL and VS. Worst performance projects were projects running on Desktop built using Google App Engine Storage and Eclipse.

Conclusion

We hope the quantitative analysis presented can give faculty a better understanding of what factors to consider when improving the design of the capstone course or when supervising students. The contributions to faculty are as follows: Do not to rely too much on deployment status, number of meetings and multi-platform implementations, they have weak correlation and not significant to student performance. Faculty should focus more on research projects, organization type, industry type, repeat projects, iOS platform, non-MySQL databases and Microsoft Technologies. These factors do influence the student performance. When sourcing for industry projects, Infocomm, Finance, repeat and technologies described earlier plays significant part on student performance. If there are opportunities to choose industry partners and projects, consider the insights presented here. However, if our students decide to propose their own projects, by all means, they usually perform better.

Table 2 is the summary of insights we gather from our statistics.

Most of our projects came from the following organizations: University, SME and Startup. Together they comprised of 63.3% of all our surveyed projects

Most of our projects were classified under the following industry: Education, Health, Finance and Infocomm. Together they comprised of 61.6% of all our surveyed projects.
University labs, faculty, government and MNC must find value in our capstone course to propose multiple capstone projects. Two out of every five projects were from a repeat industry partner.
Our students were most skilled and performed best in Infocomm projects. They did well in Finance projects too. These are the industries our students are skilled in. Our curriculum could improve in providing more use cases in Health and Education to address weakness to perform in these industries.
The most dominant technology is web-based platform using MySQL, in line with what we taught. One out of three projects implemented on multiple platforms. The three most popular IDE were NetBeans, Eclipse and Visual Studio. The two most popular non-IDE tools were from Google and Facebook. The single most popular framework is Bootstrap. Eclipse and Facebook saw declining interest.
Factors that made significant influences on project performance: teams choosing their own projects did better than university projects; Infocomm projects did better than Health projects; Finance projects did better than Health projects; iOS projects did better than web projects; Postgres, MSSQL and MongoDB projects did better than MySQL projects; and Visual Studio projects did better than NetBeans projects.
Factors that did not made significance influences on project performance: the number of meetings within the team, with their faculty supervisor and with their industry partners; which organization projects they chose: between MNC, SME or Startup business projects, between government or VWO projects, and between finance or education projects; and whether they implemented a single or multi-platform system.
Best performance projects were fully deployed system on iOS using Microsoft technologies like, MSSQL and Visual Studio proposed by the teams working on an Infocomm or finance project.
Worst performance projects were projects on Desktop with Google App Engine Storage using Eclipse.

Table 2. Summary of insights

This research is based on 346 data sample collected from our student teams. We believe some data could be more accurate. For example, we have 18 data with Other as the industry classification. This could be due to the difficulty in deciding which industry classification the project should be classified as. Another improvement in our survey is to provide options for popular technologies teams used. Our grading system uses multiple peer reviewers with a standard grading rubric to assess the team performance. This design reduces internal validity issues with dependencies on a single grader, a potential confounding variable. Another validity issue is the industry partners who did influenced the team performance. Our collaboration with industry partners in various communities, university, business and society may not be present in general and as a result reduce the external validity. Our students, curriculum, capstone course structure and grading rubric do limit the external validity as well. This generalizability limitation should be considered when trying to apply the insights presented here to your capstone course design.

Acknowledgements

We would like to thank all our capstone course students, industry partners and founding Dean, Steve Miller.

REFERENCES

Aller B. M., Lyth D. M., and Mallak L. A. 2008. "*Capstone Project Team Formation: Mingling Increases Performance and Motivation*," Decision Sciences Journal of Innovative Education (6:2), pp. 503-507.

Baird A., and Riggins F. J. 2012. "*Planning and Sprinting : Use of a Hybrid Project Management Methodology within a CIS Capstone Course*," Journal of Information Systems Education (23:3), pp. 243-257.

Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R. C., Mellor S., Schwaber K., Sutherland J. and