10-2020

Experience report on the use of technology to manage capstone course projects

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

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

Citation

GAN, Benjamin and OUH, Eng Lieh. Experience report on the use of technology to manage capstone course projects. (2020). *2020 IEEE Frontiers in Education Conference (FIE): Uppsala, Sweden, October 21-24: Proceedings*. 1-8.
Available at: https://ink.library.smu.edu.sg/sis_research/5527

Experience Report on the Use of Technology to Manage Capstone Course Projects

Benjamin Kok Siew Gan
School of Information Systems
Singapore Management University
Singapore
benjamingan@smu.edu.sg

Eng Lieh Ouh
School of Information Systems
Singapore Management University
Singapore
elouh@smu.edu.sg

***Abstract*—This full paper presents an experience report describing lessons learnt from innovative practice use of technologies in an undergraduate computing capstone course. At our school, around fifty-five teams comprising of around 300 students take this course every year. With fifty-five teams, we needed a system to schedule presentations; improve communications; collaborate between stakeholders; share knowledge; monitor progress; team up students; match students to projects; improve grading process; showcase posters; and track improvements using analytics. The Learning Management Systems (LMS) is great to manage course content and grade submission. On the other hand, students are required to conduct agile sprint reviews with university internal and external stakeholders. The LMS forum requires external stakeholders to be registered and adhere to the university terms of use. Instead of using the LMS forum, we used a wiki. Wiki is a flexible platform used by various stakeholders to collaborate and contribute, we used it as a delivery platform for assessment as well. However, wiki lacks finer user access control and complex workflow features. In our capstone course we need workflows to schedule milestone presentations based on stakeholder's changing availabilities; to match students to teams using a content-based recommendation system; and to showcase project posters. We build custom systems to meet these workflow needs by proposing them as capstone course projects for our students. We share the lessons learnt from building custom systems. By sharing our experience, we hope to support more teachers willing to innovate in their use of technology in teaching.**

Keywords—*capstone course, learning management systems, wiki, workflow, teaching tools.*

I. Introduction

In an interview, Dr. Kenneth Green, Founding Director of The Campus Computing Project, stated that 75% of Chief Academic Officers (CAOs) agree or strongly agree that teachers support the role of technology in teaching and learning [14]. However, the CAOs estimated that only 14% of general education classes use courseware. The actual effectiveness of campus IT investments were mixed. Dr. Green [9] pointed to two main challenges: research on actual impact on learning; and meaningful recognition to teachers interested in innovating their instructional methods. He said that in many academic departments, there was a fear of trying, fostered by the absence of support for innovation.

Capstone courses had become increasingly diverse in approach. Some approaches included providing a workplace project experience; a significant paper; and a series of culminating activities in a major or an interdisciplinary area. The design of a capstone course can involve complex interactions between students and external stakeholder; institutional and course structures; as well as logistical and budgetary constraints. All this can place a significant burden on the staff designing and delivering a capstone course [12]. Sourcing and aligning external projects were challenging, leaving little time to come up with innovative instructional methods using technology [4, 8]. Howe et al. [11] surveyed 522 respondents at 256 institutions on implementation strategies for capstone design programs across the U.S. The qualitative responses reported that personal success and interaction opportunities were the most enjoyable aspect of capstone design, while many respondents struggled with heavy workload, limited time, and project numbers/funding.

We believe that innovating the learning environment using information technology is essential to manage the learning workplace to remove tedious and time-consuming tasks. This belief led us to explore ways to reduce the mundane workload. This paper shares the experience of a capstone course designer using technology to improve the capstone course. We begin by describing the IT needs of the capstone course and a literature review of similar studies [5, 7, 13, 17, 18, 19, 20, 21, 22, 25]. It is followed by the IT services provided by our university with focus on the wiki collaborative platform. We share our experience of three custom build systems for the capstone course. In the final section, we summarize the technology tool comparisons and lessons learnt. The main contribution of this paper is the sharing of our experience from exploring the IT services available to and adopted by a capstone course.

II. Capstone Course

The learning outcome for our capstone course is to apply what they learn in programming, project management, and advance technology courses into building an IT system for an industry partner. Students are set to solve a real-world problem over one term comprising fifteen weeks. Students work together in teams using the agile approach to define the problem, develop a solution plan, produce and demonstrate an artefact that solve the problem, and present their work. Agile approaches value customer collaboration and responding to change [3]. Most students adopted a modified SCRUM [24] agile approach. Around 55 teams comprising of around 300 students take this course every year. To coordinate and manage the large cohort, we needed a system to schedule presentations; improve communications; collaborate between stakeholders; share knowledge; monitor progress; team up students; match students to projects; improve grading process; showcase posters; and track improvements using analytics. The system handles the mandate tasks listed above and allow us to focus on the teaching outcome, such as standardizing assessment across different projects [4] and sourcing for quality projects so students can apply the skills they acquired [8].

A. Literature Review

Neyem et al. [17, 18] at Pontifical University Catholic of Chile faced the same problem for their computer science capstone course. They created the Educational Software Tool

(EST) to provide a college-compatible integral project tracking solution. The EST is a cloud-based mobile system that doubles as a virtual shared workspace where information can be shared and exchanged. It allowed cooperative authoring, commenting and annotating shared documents as a group activity. Other features included software development specific features for reporting, requirements gathering, knowledge managing, version controlling, messaging, quality assurance, file hosting and calendar. The event logs produced by EST were used to analyze cooperative team behaviors for feedback on the development process and quality of deliverable produced by student teams.

Fan [7] at Penn State Behrend's Computer Science and Software Engineering Department presented an orchestrated way to conduct agile sprint reviews in undergraduate capstone projects. In order to collaborate with customers and responding to change, a sprint review was conducted at the end of each sprint to demonstrate and critique the features that the development team have completed. The availability of stakeholders such as industry partners were unpredictable due to their own business commitments and schedules, making it hard to schedule meetings. To compensate for this, they used a web-based collaboration platform to collect reviews asynchronously from different stakeholders. Similarly, Chen & Teng [5] at National Central University, Taiwan uses meetings-flow system (MFS) for collaborative teamwork and organizing meetings for senior software engineering project-based learning.

Ohtsuki et al. [18] at Saga University 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. DevOps is a set of practices that automates the processes for software development, testing, deployment, release and maintenance [1, 2]. ALECSS integrated the tools: Git for version control, Jenkins for continuous integration, Ant for building code, JUnit for unit test, Checkstyle for coding style checking, and FindBugs for static code analysis. By using the data produced by these DevOps tools, it was able to provide quicker feedback with more warnings than manual checking. However, the paper did not mention integration with project management tool such as RedMind into ALECSS. It supported grading by the teachers as well. Their study focused on software engineer education support system utilizing specific DevOps Tools.

Rao [22] at Miami University's Computer Science and Software Engineering Department experienced significant growth of doubling their students in 3 years. This has led them to develop an automatic testing and grading software system called Code Assessment Extension (CODE) which is a plugin to their Canvas LMS. It automates C++ code submission, compilation, style checking, functional testing and source code annotation for feedback. He analyzed data collected from the used of CODE in a system course and found improved overall quality of learning outcomes, however instructors are concern with student's reliance on the tool. Auto-grader for CS courses is not new. Wilcox's tool [25] graded Java code. Peveler et al's tool [21] graded code from different programming languages but requires a JSON input.

B. Study Comparisons

Similar to the examples provided above, we share our experience of automating some of the tasks in teaching a capstone information systems course. Instead of a custom-built EST at Pontifical or a web-based collaboration platform at Penn State Behrend, we started with an off-the-shelf collaborative wiki platform that allowed our capstone community, including non-developers such as industry partners to access a virtual shared workspace. Teams were able to share their project scope from requirements gathering, design prototypes, technical architecture and project management deliverables. This flexible wiki platform was easy for our students to use and required less effort to maintain than a custom-built system. Supervisors and industry partners were able to do agile reviews after each sprint and contributed to the growing knowledge base on our wiki. However, when we reached the limitations of a wiki with the need for workflows, we proceeded to supplement our wiki using custom-built tools, such as a scheduling application, similar to MFT at NCU Taiwan.

In comparison to ALECSS and CODE, our capstone projects were not restricted to use specific DevOps tool or programming language. ALECSS processed code submitted using specific DevOps tools (Git, Jenkins, Ant, Junit, Checkstyle and FindBugs). CODE is an automated grading system for C++. Majority of our students coded using Java and Python. Some coded with C#, PHP, JavaScript, Objective C, Ruby, and even Visual Basic. Based on these preferences, their choices for integrated development environment (IDE) were Eclipse, Netbeans, IntelliJ, Visual Studio, Android Studio, XCode, WebStorm, PHPStorm, PyCharm, etc. Most students used Git for version control but there were a few who used BitBucket, Jira and SourceForge. Similarly, most students used Microsoft Project or just a spreadsheet for project management. Some used Trello, BaseCamp, Microsoft Team or Google Cloud. These flexibilities were partly due to the industry partner preferences to choose development tools and programming languages. We were unable to follow the approach taken by ALECSS and CODE. For this reason, we stayed away from building code analysis or automated grading tools and focused on specific tools we needed to automate scheduling, matching and showcasing.

C. IT Needs of a Capstone Course

We used the university provided Learning Management System (LMS) to deliver our courseware and for grade submission. The wiki collaborative platform is used to improve communications, collaborate between stakeholders, share knowledge, and monitor team progress. However, to schedule presentations, team up students, match students to projects, improve grading process, showcase posters and track improvements using analytics, we custom build tools. Without a budget, we relied on our student software engineers to build these tools. In the next section, we describe the IT services provided by our university and explain how we used the wiki collaborative platform.

III. University IT Services

Most university provide various IT services to aid in teaching and research. Our university provides support for teaching and learning; classroom and event; communication and collaboration; research; electronic document management system; event management system; facilities booking system; student, teacher and staff portals; integrated student information system; iTunes podcast; lecture capture support; student course evaluation system; survey tools; wiki collaboration platform, etc. [23]. The teaching and learning system is an LMS that include content management system for courseware; student feedback; grade submission; quiz and assignments; forum; etc.

A. *Learning Management System*

The LMS is able to integrate with third-party tools for course management, assessment, assignment and even customized plugin to do automated code grading as discussed by Rao [22]. The course registration system populates the LMS with registered students and courses. This makes it very convenient to use. The requirement to submit course final grades with the LMS helps standardize the grading system across the university. However, the forum provided in the LMS is insufficient for our needs. Our capstone course requires collaborations between students, teachers, industry partners, system users and testers. Students are required to conduct agile sprint reviews with university internal and external stakeholders. The LMS forum requires external stakeholders to be registered and adhere to the university terms of use. Instead of custom building a cloud-based mobile virtual shared workspace [17, 18], or a web-based collaboration platform to organize meeting [5] and to collect reviews asynchronously [7], we used a wiki platform.

B. *Wiki Collaborative Platform*

Wiki is a flexible platform that allows a community to contribute content and users to search for these contents. It allows industry partners to advertise projects for students to choose. Students were able to share their team and project information. The community can share what they learn across teams and build a knowledge base. The community was able to build their web content easily on their own. With a little bit of structure, it can be used to document a capstone project.

Our decision to use the wiki was supported by the following studies. Elgort et al. [6] studied on whether wikis could facilitate collaborative learning and positively affect student attitudes to group work in the context of an assessed group project. The result was encouraging, indicating both students and teachers saw value in using wiki as a collaboration tool. For the students, it encouraged better individual participation and organizing information for group projects. For teachers, it contributed to the ease of managing and marking student work in group project. However, for students who prefer to work individually, wiki may not improve participation. Neumann et al. [15] conducted an experiment on whether wiki promote collaborative learning among students. One group jointly write a report using wiki and another group communicated in a report they wrote individually. The wiki approach produced higher engagement with other students, cognitive engagement, and class attendance than the individual approach. Similar to the findings by Elgort et al., individual preference on working in groups may not be influenced by wiki.

Our university uses MediaWiki to create a wiki for each course. Each team populated their wiki pages with project information. We suggested a dashboard home page for each team; a team page with roles and responsibilities; a project overview page with description, scope, stakeholders, X-factor, and outcome; a project management page containing schedule, risks and project metrics; and a documentation repository for meeting minutes, diagrams, wireframes and user test documentation. The wiki can store both text and images. While many teams wanted to have an ecstatically pleasing page, putting their text in images result in difficulty in searching for the text. Fig. 1 shows the wiki home page and fig. 2 shows the wiki project scope page for team Orange. Note that details in the figures are not relevant. It illustrates the ecstatically pleasing pages with text in images that cannot be searched.

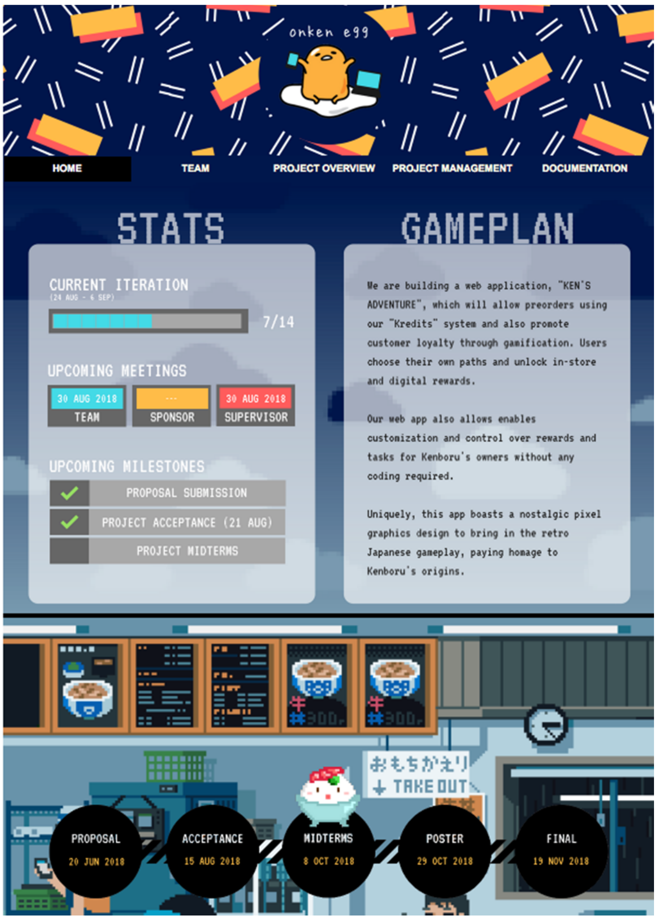

Fig. 1. Team Orange Home page.

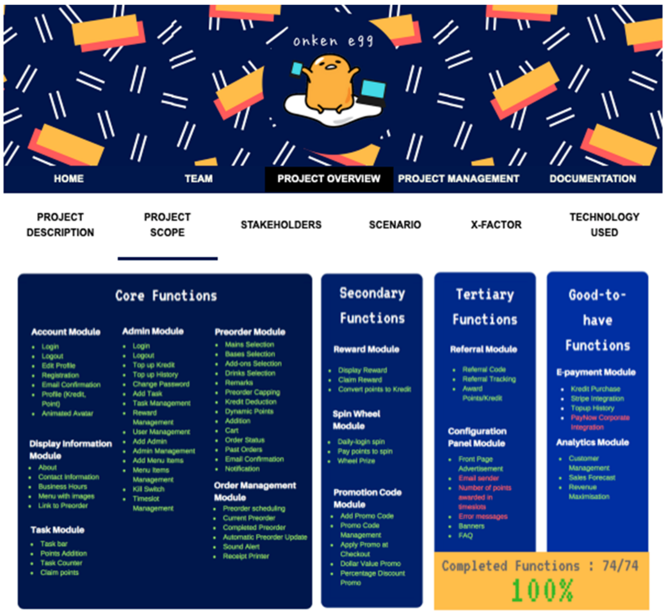

Fig. 2. Team Orange Project Scope page.

The team wiki pages were used to communicate progress as well as deliverables for grading when a milestone was reached. Some teams updated their wiki only when reaching a milestone while others updated theirs routinely. Each page included a version history log to deter students from sabotaging another team's page. The log was used to see if a team updated before each milestone or updated routinely as desired. As the wiki content grew, we began to use it for other purposes, such as scheduling presentations.

There were two drawbacks for using the wiki. The course wiki home page was only editable by the teacher, and the rest of the pages were editable by users with university accounts. Finer access control required the teacher to enter each external user and to remove edit rights for each team page. Adding and removing individual users can be tedious. This was where the LMS or a google site might be more suitable. The second drawback was that wiki was not designed to enforce a workflow. Although it was possible to build a workflow using wiki, specific customization for the capstone course was required. Wiki can create content, name objects, control versions, manage rights and combine text and images. But a wiki-based workflow would need an application-based state and workflow management [15].

IV. Custom Build Tools

We needed a workflow system to schedule milestone presentations and to match student teams to projects. We proposed two custom build systems for our capstone course students. The scheduling system allowed teams to book presentations slots. The matching system matched projects to student teams. Team Turquoise worked on the scheduling system and team Lavender worked on the matching system. The scheduling system was successfully deployed since 2013 and still in used. This single purpose system required a simple upgrade in 2016 and worked well since without any need to recode. The course coordinator will setup the student teams every term for students and reviewers to schedule their acceptance, midterm and final presentations. A TA and instructor may need to launch the scheduling process in a rare occurrence that the system unexpectedly shuts down. The matching system was deployed in 2014 and used for about a year. Students revert back to using the wiki and we decided it needed a revamp.

Team Sapphire took over the matching system in 2015. They worked on various enhancements and added a custom grading function. Unfortunately, adding too many enhancements resulted in a system that was hard to maintain and use. Our final custom build tool design posters for showcase on a website. Team Amber worked on this project in 2017 and it we used for a couple of years. These custom-built tools meet specific needs compare to commercial project/team management tools such as Slack or Trello. The following subsections explains the scope of each custom build system and the lessons learnt.

A. Scheduling System

Scheduling presentations for 25 teams each term based on the changing availability of each student, supervisors and reviewers can be a daunting task. Our university uses Microsoft Outlook with calendar or Microsoft Teams to help book appointments. However, not every teacher or students updated or used this calendar. So, we started using the wiki with open slots for student to book. The wiki booking page was editable by all 300 students. The version log was supposed to deter students from overriding each other's bookings. However, it didn't work well and tracking the logs was difficult. We tried to encourage teacher and students to update their Outlook calendar but found that they preferred to use a single event scheduling tool instead of using a different calendar or sharing their personal calendar. Thus, we commissioned the development of a custom build scheduling system. This system included constraints specific to our capstone course, such as features to set a booking window for each milestone, set the time duration for each presentation slot, set the unavailable slots, set the required participants in each presentation and set the approval process by various stakeholders to confirm the presentation slot.

Team Turquoise worked on the scheduling system. They gathered the As-Is workflow and designed a To-Be workflow, shown in fig. 3 and 4 respectively. The As-Is workflow shows a loop of emails to get an agreed presentation slot between the project team and supervisor/reviewers. The To-Be workflow shows the approval process based on available time slots of the specific supervisor/reviewer to be selected by the teams when the booking window opened. This customized approval process by the relevant stakeholders would need to be customized as well if we had. used the recommended Outlook. The solution asked teachers to enter their available slots for teams to book. The coordinator chose the milestones, attendees, duration and the booking window. A change in schedule will automatically notify relevant participants of the change. Fig. 5 shows the teacher view of the scheduling system with the teacher availability for each slot.

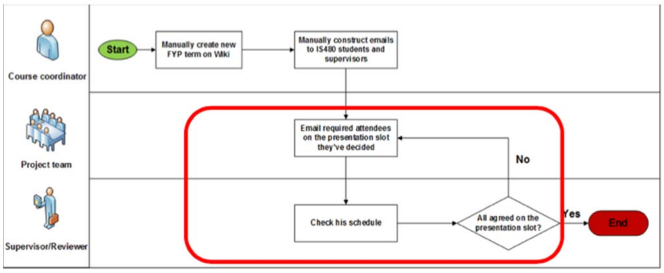

Fig. 3. Team Turquoise As-Is workflow.

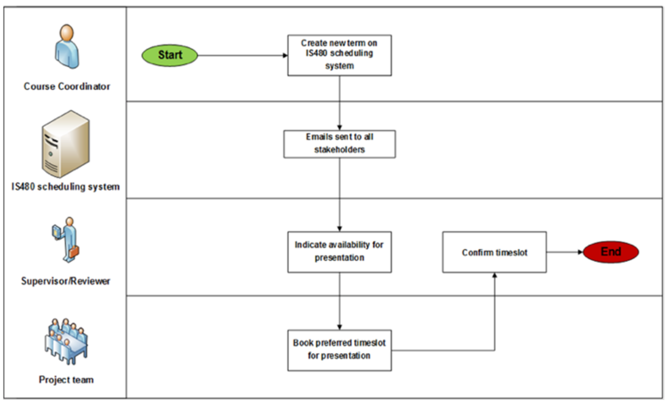

Fig. 4. Team Turquoise To-be workflow.

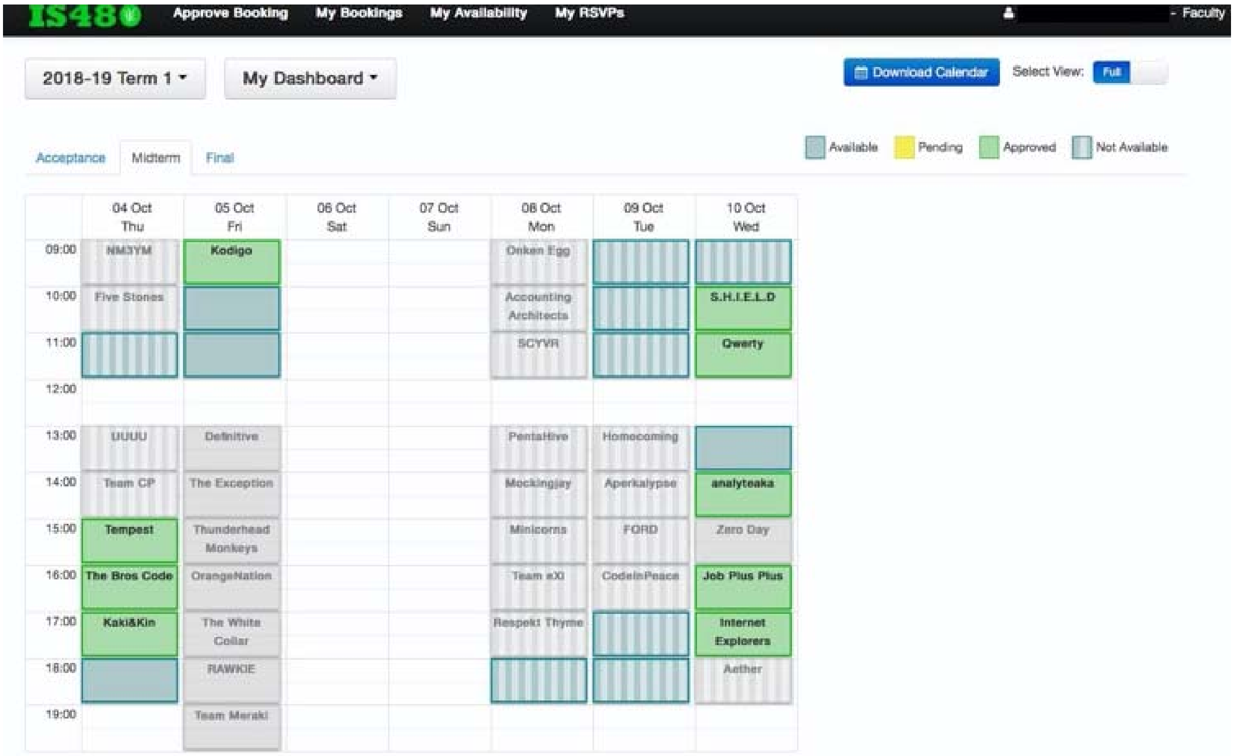

Fig. 5. Team Turquoise Scheduling System from teacher's perspective.

Scheduling presentations used to take up to months with cancellations and inconsistent status that resulted in missing participants. This project was the most successful system that we custom built and is still in use now. This success was mainly due to the simple and useful functionality that created high value for the course. We avoided asking for more enhancements. A possible enhancement was to incorporate a scheduling optimization algorithm to list possible presentation slots based on the availability constraints and allow stakeholders to vote on. Instead of adding to the scope, we conducted security, quality and user tests on the system before production. In compliance with personal data protection, a login system for each user role were carefully managed to avoid revealing schedules for unrelated teams. A code walkthrough was conducted to avoid trojan code and venerable code injection when processing input data.

B. *Matching System*

The matching system was initially proposed to replace part of the wiki. The system matched students to teams, teams to projects, and teachers to teams. Other functions included invitations, notification and a feedback system. Team Lavender worked on the matching system. The large scope delayed the deployment to after the midterm. We tested using teaching assistants and other teams. Although students tried to use the deployed matching system, we learned that they were more comfortable forming teams on their own. After one year, few students continued to use the matching system.

Team Sapphire took over the matching system to try to revitalize its usage. The system allowed students to update their profile with proficiency of skills. This was used to form teams as well as match projects with the required skills. The algorithm for matching used a content-based recommendation system for jobs [10]. While this was an interesting problem, the algorithm was good only if it had good data. Student teams preferred to scan the list of projects to choose from instead of entering their skills with proficiencies. Similarly, not all industry partners were knowledgeable in IT and some were not be able to list the required skills. Based on these findings, team Sapphire realized that matching teams to projects was better done manually and removed this from the scope. Instead of a skill matching algorithm, team Sapphire built the system so industry partners can comment on the project to avoid repeated questions from multiple teams.

While empathizing with the students, team Sapphire received feedback that students would like immediate feedback from teachers when grading their presentations. Our university LMS has a grading system where teachers can enter component grades such as presentation milestones. However, teachers did not like to use the LMS to submit comments on component grades. So, team Sapphire developed a teacher grading system using our customized capstone course grading rubric with fields for comments. Students wanted to have immediate feedback, but some teacher were not comfortable entering grades into any system. They preferred to use Word or even hardcopy to write their comments. This conflict was further studied. The teachers concerns were more with the type of comments we collected and released to students. Although we tested the system with teachers in a live grading milestone, teachers decided not to use the system.

C. *Poster Showcase System*

The school portal was used to communicate about the school, its events, research, programs and curriculum. We employed a web administrator to manage the portal content to showcase research and course projects. When students prepared their posters, it can lead to two issues. Students used a variety of tools such as Photoshop, PowerPoint, CorelDRAW, etc. It was hard to keep consistency in fonts, images, layout and colors. The second issue was that not all students were poster designers. They complained that it took up a lot of effort to come up with good posters. We proposed a Poster showcase system to manage consistency and make it easy to design a poster.

Team Amber worked on the poster showcase system which included a poster design canvas that allowed flexible drag and drop of images with text in multiple fonts and colors. The administrator created a template poster with specific fonts and layout. Students chose from multiple templates with a consistent layout. They cut and pasted the text; and dragged and dropped images to customize the template. Fig. 6 shows the preview of a template. This system collected project information such as team names, project description, team photos, industry type, company name, technology used, and X factor. This information was automatically added to the poster based on the template fields. This reduced the effort and time students needed to create a poster. The project information can be displayed for analysis. Fig. 7 shows the poster dashboard with statistics in terms of industry type and sponsor type. The completed poster is stored for each team and display for the public to view with search features. Fig. 8 shows the poster showcase page.

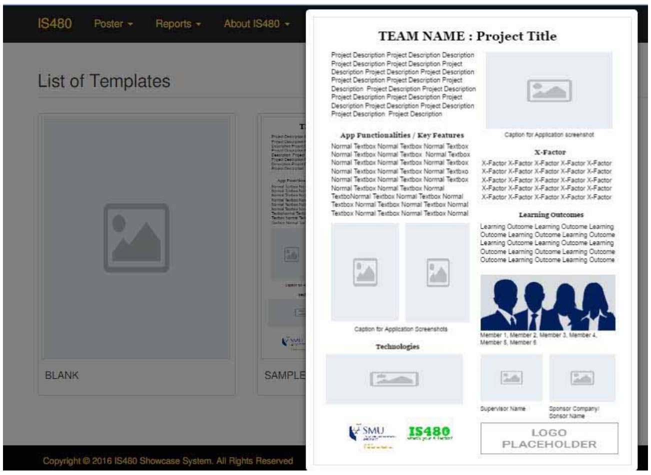

Fig. 6. Team Amber Showcase System Poster template.

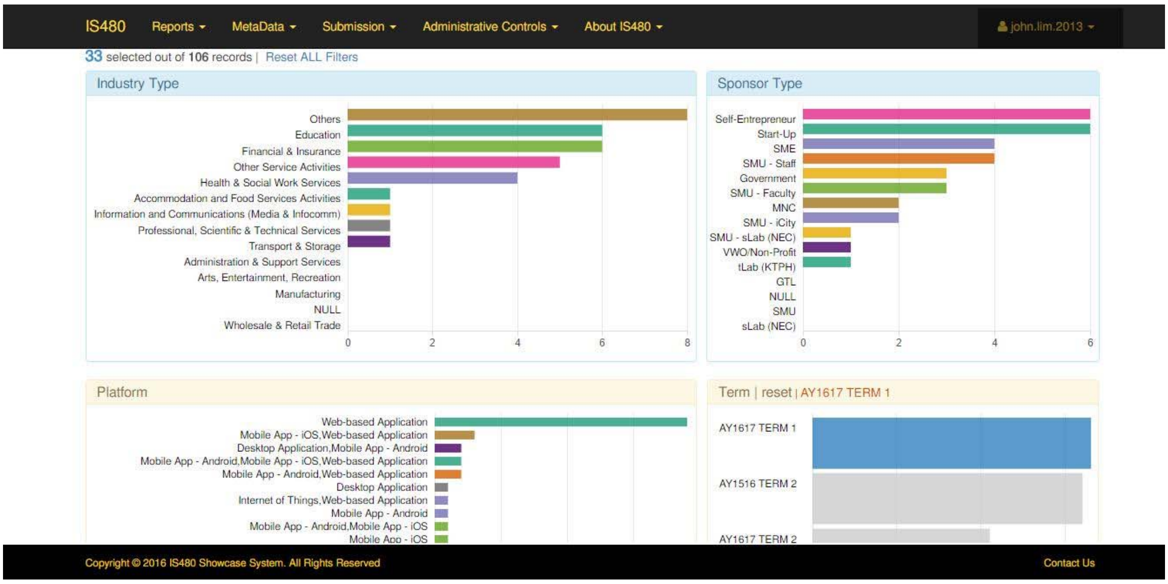

Fig. 7. Team Amber Showcase System Industry Dashboard.

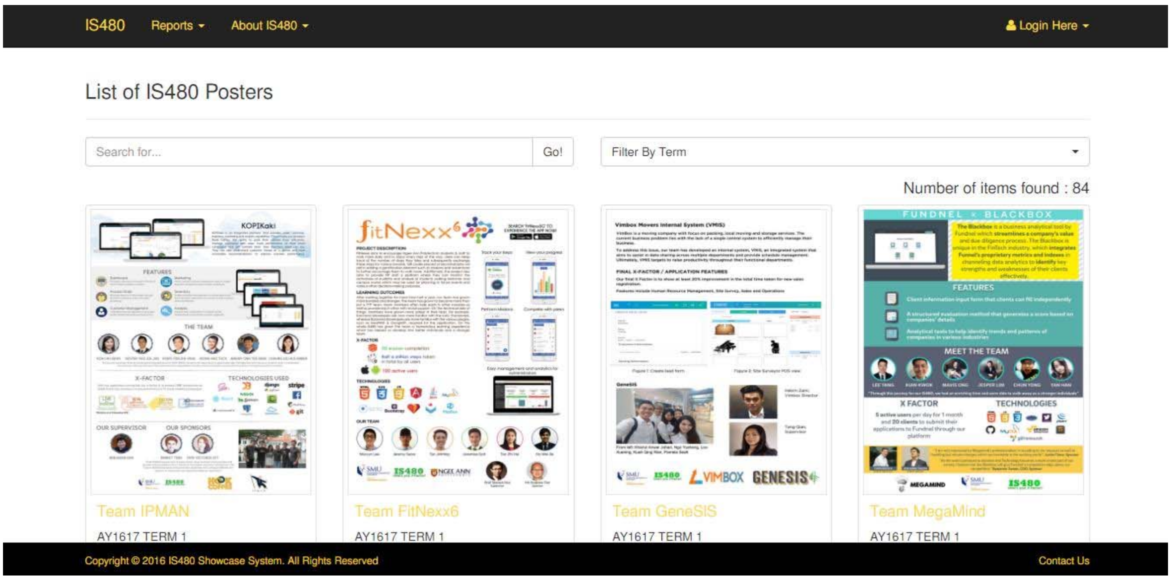

Fig. 8. Team Amber Showcase System Posters.

It was hard to strike a balance between providing a flexible poster designer and a consistent look and feel. While we wanted to restrict the design to the template layout and reduce student effort, some students still preferred to use their favorite design tools. The system was flexible enough for teams to choose from a template or to upload their own designed poster. This may still cause some inconsistent layout. An enhancement is to include an approval workflow for uploaded posters.

V. LESSONS LEARNT

With a growing in-take of IT students, automating the project management tasks for a capstone course is increasingly important. We explored the use of off-the-shelf LMS and wiki collaborative platform to track projects and aid project reviews. We explained the pros and cons of LMS and wiki, then we share our lesson learnt from building three custom systems to manage projects and teaching tasks.

LMS may not fit some of our capstone course needs. However, having the official LMS which contains real-time data of student class registration and grade submission does make it convenient [23]. Wiki is an excellent collaborative platform that is used by many teachers to build a community of knowledge base [6, 15]. But it lacks workflow features for complex communication [16] and finer user access control requires tedious work to import student class registration and external user data. We implemented scheduling, matching and showcase system to meet our capstone course specific needs. However, there were risks building a system from scratch. The tools we used automate the tasks our students are required to perform, thus allowing our students to spend time focusing on the learning outcome.

The lessons learnt is keep to simple functionalities with clear value for all users. This lesson may sound simple, but it was not obvious during the project. The scheduling system reduced the effort to schedule presentation slots. Its approval workflow was implemented quickly. Instead of enhancing it with a scheduling algorithm, we did security, quality and user testing. This lesson could have been applied to our matching system. The matching system matches the student skill sets to the project required skill. It seemed like a great project to apply our student knowledge in content-based recommendation system. However, understanding user needs can be more important than taking the opportunity to apply student technical knowledge. Students preferred to search for a project instead of providing their skill sets. Some industry partners did not have technical knowledge to select the right technical skill sets required in their project. Without good data, the recommendation system did not work well. This real-life lesson is a recurring lesson our student feedback during reflection on capstone projects.

Another real-life lesson is managing conflicting user needs. In the teacher grading system, an added scope to the matching system, students wanted immediate grade feedback after a milestone presentation. However, teachers were uncomfortable giving immediate comment feedback using an online system. In the poster showcase system, the coordinator wanted a consistent layout, but some students wanted more flexibility to design their posters. We were able to come up with compromises: the teacher grading system allowed teachers to manage the release of their comments and the poster showcase system allowed those students who prefer to design using their favorite tool to upload their posters without using the template. Table 1 summarized the technology used in our capstone course and their pros and cons. Table 2 summarized the custom-built tools used in our capstone course and lessons learnt from them.

TABLE I. PROS AND CONS OF TECHNOLOGY USED

Tools	Pros	Cons
LMS	1. University standard for content management, student feedback and grade submission. 2. University populated real-time data of student class registration is convenient.	1. Features that is not customized for capstone course use. 2. Forum requires external stakeholders to be registered and adhere to the university terms of use.
Wiki	1. Flexible platform that allows a community to easily collaborate and share knowledge. 2. Users can search easily. 3. With a little structure, it can be used to document capstone projects and monitor progress. 4. Wiki log can check for desired team's routine updates instead of all updates before a milestone. 5. For students, it encourages better individual participation and organizing information for group projects. 6. For teacher, it contributes to the ease of managing and marking student work in group project.	1. While searching text on wiki is convenient, our student like to add text into their images that they upload on their wiki pages. Text in the images are not searchable. 2. Finer access control required the teacher to enter each external user and to edit rights for each team page. 3. Wiki-based workflow would need an application-based state and workflow management. 4. For students who prefer to work individually, wiki may not improve participation.
Custom build tools	1. Features are customized for specific capstone course needs. 2. Reduce administrative tasks of managing multiple projects.	1. Ensuring security, quality and usability are necessary before production. 2. Maintenance cost is more compare to off-the-shelf system.

The learning outcome for our capstone course is to apply what students learnt in their foundation and senior courses in a real problem to implement a system. In the custom-built systems, student teams learnt that technical skills to build the systems is only part of the solution. Sometimes, it is tempting to try to apply the latest technically challenging solution such as a scheduling optimization algorithm or a content-based recommendation algorithm for jobs. However, building an elegant solution that is not in used is just as bad as not building the system. The two real-life lessons: simple functionalities with clear value for all users and managing conflicting user needs can sometimes define the success of the system being built. In our capstone course, our students continuously feedback that these real-life experiences are what they learn from the project-based course. These learning reflections is part of the learning goals that will help them when they commerce on their working career.

By sharing our experiences, we hope to encourage more course designers to explore various IT services and avoid pitfalls we faced. As we try out new approaches, the never say die attitude is necessary to eventually lead to innovation. Academic departments and the teaching community through conferences allow discussions and support for these endeavors.

TABLE II. LESSONS LEARNT FROM CUSTOM BUILD TOOLS USED

Custom build Tools	Lessons Learnt
Scheduling System	1. Simple and useful functionality that created high value for the course. 2. It is important to start the project with essential functions to create a usable MVP (calendar UI) and add more features only when time permits (scheduling algorithm). 3. Test security, quality and usability before production. These tests are more important than enhancements (scheduling algorithm).
Matching System	1. Student projects require careful management of project scope (An MVP to test early, before midterm to get buy in from students) 2. It was important to understand the needs of various users (course coordinator, industry partner and student teams). 3. Balance conflicting requirements. Coordinator wanted a content-based recommendation matching system. Student preferred to choose project instead of listing their skills sets and some industry partners are unable to supply skills required for the project. 4. Focus on needs over technology beauty. Applying the content-based recommendation system allow students to apply what they learnt but if there are no good data, the recommendation system does not work.
Teacher Grading System	1. Balance conflicting requirements. Student wants immediate feedback on their grades while teachers are uncomfortable with releasing their grading comments before vetting.
Poster Showcase System	1. Compromise when faced with conflicting user needs. Students wanting a simple poster design system. Coordinator wanting consistency in the posters produced by the students. The compromise uses template for consistency but still allow students to upload their posters using their favorite tool.

ACKNOWLEDGMENT

We would like to thank student who worked our projects for their capstone course. Without their dedication and effort, teacher tasks are filled with mundane workload. Although some systems are not in use, we believe their contributions helped us understand how to improve our next project.

REFERENCES

[1] Atlassian. “DevOps: Breaking the Development-Operations barrier,” 2019. https://www.atlassian.com/devops

[2] L. Bass, I. Weber, and L. Zhu. “DevOps: A Software Architect's Perspective,” Addison-Wesley Professional, 2015.

[3] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. “Manifesto for Agile Software Development,” Agile Alliance. 2001. https://agilemanifesto.org

[4] C. Boesch, and B. Gan. “Evolving an Information Systems capstone course to align with the fast-changing Singapore marketplace,” in Capstone Design Conference Proceedings. 2014. http://www.capstoneconf.org/resources/2014%20Proceedings/Papers/0065.pdf

[5] C.Y. Chen, and K.C. Teng, “The design and development of a computerized tool support for conducting senior projects in software engineering education,” Computers & Education, 2011, 56(3), pp 802-817.

[6] I. Elgort, A.G. Smith, and J. Toland. “Is wiki an effective platform for group course work?” in Australasian Journal of Educational Technology (AJET’08). 2008, 24(2), pp. 195-210. https://ajet.org.au/index.php/AJET/article/view/1222

[7] X. Fan. “Orchestrating Agile Sprint Reviews in Undergraduate Capstone Projects,” in 2018 IEEE Frontiers in Education Conference (FIE’18), 2018, 1, pp. 1-8. https://doi.ieeecomputersociety.org/10.1109/FIE.2018.8658435

[8] B. Gan, and V. Shankararaman. “Project sourcing for capstone course experience from an undergraduate program,” in 26th International Conference of Society for Information Technology and Teacher Education (SITE’15), The Learning and Technology Library, 2015. https://www.learntechlib.org/p/150622/

[9] K. Green. “Provosts, pedagogy and digital learning. The 2017 ACAO survey of provosts and chief academic officers, Association of Chief Academic Officers,” 2017. https://www.acao.org/assets/caosurveysummary.pdf

[10] X. Guo, H. Jerbi, and M.P. O'Mahony, “An analysis framework for content-based job recommendation,” in 22nd International Conference on Case-Based Reasoning (ICCBR’14), 2014. https://link.springer.com/book/10.1007%2F978-3-319-11209-1

[11] S. Howe, L. Rosenbauer, and S. Poulos. “The 2015 Capstone Design Survey Results: Current Practices and Changes over Time,” Engineering: Faculty Publications, Smith College, Northampton, MA, 2017. https://scholarworks.smith.edu/egr_facpubs/9

[12] N. Lee, and D. Loton. “Capstone curriculum across disciplines: Synthesising theory, practice and policy to provide practical tools for curriculum design,” Office for learning and teaching, Department of Education and Training, University of Melbourne, 2015. https://minerva-access.unimelb.edu.au/handle/11343/119575

[13] V. Mahnič, & A. Časar, “A computerized support tool for conducting a Scrum-based software engineering capstone course,” The International journal of engineering education, 2016, 32(1), pp. 278-293.

[14] Mediasite. “Attitudes, realities and challenges: The role of digital technologies in higher education,” Sonic Foundry, 2018. https://mediasite.com/blog/role-digital-technologies-higher-education/

[15] D. Neumann, and M. Hood. “The effects of using a wiki on student engagement and learning of report writing skills in a university statistics course,” in Australasian Journal of Educational Technology (AJET ’09). 2009, 25(3), pp. 382-398.

[16] G. Neumann, and S. Erol. “From a social wiki to a social workflow system,” Lecture Notes in Business Information Processing, vol 17. 2009, Springer, Berlin, Heidelberg. https://ajet.org.au/index.php/AJET/article/view/1141

[17] A. Neyem, J. Diaz-Mosquera, J. Munoz-Gama, and J. Navon. “Understanding Student Interactions in Capstone Courses to Improve Learning Experiences,” in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE’17). https://dl.acm.org/doi/10.1145/3017680.3017716

[18] A. Neyem, J. Benedetto, and A. Chacon, “Improving Software Engineering Education through an Empirical Approach: Lessons Learned from Capstone Teaching Experiences” in Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE 2014). https://dl.acm.org/doi/10.1145/2538862.2538920

[19] M. Ohtsuki, K. Ohta, and T. Kakeshita. “Software engineer education support system ALECSS utilizing DevOps tools,” in Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services (iiWAS’16), 2016, pp. 209-213. https://dl.acm.org/doi/10.1145/3011141.3011200

[20] J. J. Olarte, C. Domínguez, A. Jaime, & F.J. García-Izquierdo, “A tool for capstone project management in computer science engineering,” in 2014 International Symposium on Computers in Education (SIIE), pp. 65-68. IEEE.

[21] M. Peveler, E. Maicus, and B. Cutler, “Comparing jailed sandboxes vs containers within an autograding system,” in Proceedings of the 50th ACM Technical Symposium on Computer Science Education, ser. SIGCSE ’19. New York, NY, USA: ACM, 2019, pp. 139–145. http://doi.acm.org.proxy.lib.miamioh.edu/10.1145/3287324.3287507

[22] D.M. Rao,”Experiences with auto-grading in a systems course,” in 2019 IEEE Frontiers in Education Conference (FIE’19), 2019. https://www.users.miamioh.edu/raodm/pubs/confs/fie19.pdf

[23] Singapore Management University (2020). Integrated Information Technology Serices website. https://itsupport.smu.edu.sg/

[24] Sutherland J. and Schwaber K. 2013. *The Scrum Guide*. Scrum.org.

[25] C. Wilcox, “Testing strategies for the automated grading of student programs,” in Proceedings of the 47th ACM Technical Symposium on Computing Science Education, ser. SIGCSE ’16. New York, NY, USA: ACM, 2016, pp. 437–442