ISSN: 2289-7844 | e-ISSN: 0127-9750
DOI: https://doi.org/10.37134/jictie.vol12.2.7.2025

Journal of ICT in Education (JICTIE)
Volume 12, Issue 2, Pages 99-122, 2025

Research Article

Enhancing IT Capstone Programs through Agile Practices: Insights from Filipino Community Projects

Giuseppe Ng*

Department of Information Science and Technology, Faculty, University of Asia and the Pacific, Pasig, Philippines; giuseppe.ng@uap.asia

Received: 27 May 2025; Revised: 18 August 2025; Accepted: 23 September 2025; Published: 25 October 2025

**corresponding author*

Abstract

The IT Capstone Program at the University of Asia and the Pacific (UA&P) integrates Agile methodology through the Scrum framework to provide students with practical experiences in consultancy and software development. To extend its impact, the program was later expanded through the UA&P Filipino Community Extension Program, which engages partner communities facing challenges that require IT-based solutions. This study examines the program's implementation and areas for improvement after several years of practice. Data were collected through focus group discussions with alumni, industry practitioners, capstone students, and faculty advisers. Findings highlight the need for enhancements in sprint planning, code quality, and software maintainability to strengthen alignment with industry standards. Moreover, the study reveals how contextual factors in the adopted communities—such as IT infrastructure and government funding—affect the sustainability and effectiveness of capstone-developed solutions. The insights offer implications for refining capstone programs that aim to bridge academic training, industry expectations, and community service.

Keywords: capstone program, information technology, agile, scrum framework, community extension programs.

INTRODUCTION

Technology and software have played a significant role in the community (Dam et al., 2018). Previous studies show that the utilization of software solutions can increase the productivity and efficiency of the operations of an institution (Gallera & Salvador, 2023). As such, having Information Technology (IT) students conduct software engineering capstone projects is a natural extension, providing students with real-world experience as well as providing companies with technology solutions that aid their operations (Bastarrica et al., 2017; Ng, 2018; Herbert, 2018; Aljohania et al., 2022).

Previous works show that companies deprioritize capstone works due to skepticism arising from student inexperience and lack of practical IT and Information and Communication Technology (ICT) skills (Alzamil, 2005; Jones & Davey, 2009; Marriska, 2015; Wilbur, 2016; Daff, 2021; Aljohania et al., 2022; Strain & Marshall, 2023). Given that capstone projects can result in positive results (Parsons & Lepkowska-White, 2009; Schachter & Schwartz, 2009; Sprague & Percy, 2014; Sprague & Hu, 2015; Mandale et al., 2021; Li et al., 2023), the Department of Information Science and Technology in

University of Asia and the Pacific (UA&P) conducted a long-term study of its IT Capstone Program (Ng, 2018; Ng & Cruz, 2020). To address criticism regarding the lack of practical IT skills, the program incorporates a practical implementation of Agile development with a scrum framework widely adopted across many IT companies for its fast software delivery and high success rate (Chawla, 2016; Denning, 2016; Rigby et al., 2016; Fernandez, 2024). This program yielded several well-received software solutions (Ng, 2018; Ng & Cruz, 2020).

Community extension programs are part of the responsibilities of Higher Educational Institutions (HEIs) and can range from knowledge sharing to training workshops and resource sharing with adopted barangays, schools, and other institutions in the community (Buys & Bursnall, 2007; Quezada, 2014; Castillo, 2018; Sibal, 2019; Nimer, 2020; Al-Sholi et al., 2021; Magnaye & Ylagan, 2021; Cristobal, 2023; Vera, 2023). Considering that several Filipino companies and institutions can greatly benefit from implementing software solutions to automate and improve operations (Ng, 2019; Sunio et al., 2022; Gallera & Salvador, 2023), the UA&P IT Capstone Program was integrated into various adopted communities that required such software solutions (Buenafe et al., 2024).

As noted previously (Ng, 2000; Ng & Cruz, 2018), early results showed that student capstone projects produced highly relevant IT solutions given the iterative nature of software development. However, a related study (Ng, 2018) revealed a need to better understand the long-term use and deployment of IT solutions based on client needs. This is in preparation for transitioning the program into a complete DevOps implementation as a natural progression of this concept (Alves & Rocha, 2021).

This study utilized surveys, focus group discussions, and interviews with various stakeholders: (1) students of capstone projects deployed to community extension programs; (2) capstone faculty advisers; and (3) alumni who are also industry practitioners. Specifically, the research study aimed to answer the following research questions:

1. As Agile methodology is part of modern industry practices, what could the IT Capstone program learn from industry practitioners who completed the same program?
2. What are the relevant considerations for the deployment and long-term use of software solutions, especially considering community extension programs?
3. Through thematic content analysis of the responses of current students of capstone projects, how do the insights drawn from industry practitioners and faculty advisers compare to the actual execution of the Community Extension Capstone Projects?

From discussing the program with the aforementioned stakeholders, the program can be considered successful in providing important IT skills to students and delivering highly relevant software to community extension programs. However, there are areas that the program could improve, such as providing more emphasis on student group development capacity and software deployment and rollout.

REVIEW OF RELATED LITERATURE

The following section covers concepts surrounding Software Engineering, the integration of Software Engineering in HEI courses, and Community Extension Programs to flesh out relevant concepts behind this Capstone Program attached to Community Extension Partners.

Agile Methodology with Scrum

Agile methodology with Scrum is an iterative approach of software development in contrast to the Waterfall model (Sommerville, 2016). In this approach to software development, the specification, design, implementation, and testing are interleaved.

Through iterating this process, developers can respond to change and adjust the solution based on the feedback of clients. According to Smith (2016), there are four distinct phases to the Scrum framework: (1) Sprint Planning; (2) Sprint; (3) Sprint Review; and (4) Sprint Retrospective.

The Sprint Planning phase is where developers and clients decide the work that goes into a development cycle called a Sprint. Selection of user stories from the product backlog is based on prioritization and feasibility. During the Sprint, developers will conduct short update meetings called daily scrum to keep track of the development and make any adjustments should any changes happen. A Sprint Review is conducted to assess the development work of the Sprint, while the Sprint Retrospective serves as an evaluation of the previously concluded Sprint (Smith, 2016; Apke, 2015).

Software Maintainability

Software Maintainability is the attribute that software possesses regarding its ease of development, testing, and performing corrective maintenance activities (Kazman et al., 2020). These maintenance activities are broken down into four major types (Agarwal & Majumdar, 2013):

- Corrective maintenance – the activity involving identifying and rectifying software defects;
- Adaptive maintenance – the activity of changing the software in response to ever-changing environments and situations;
- Perfective maintenance – the activity of improving the efficiency of the software; and
- Preventive maintenance – the activity of rectifying the decaying structure of software to reduce complexity and potential for further software defects from being introduced (Agarwal & Majumdar, 2013).

The maintenance phase of the software engineering process is the most expensive part of development, as it covers updating software functionality and rectifying bugs post-deployment (Sommerville, 2016; Chua et al., 2006). Improving software maintainability can reduce the overall development costs (Spray et al., 2021). According to Agarwal and Majumdar (2013), the maintainability of software is influenced by team experience, project size, and commitment of the client to the project. In addition, program understanding impacts software maintainability (Chua et al., 2006). In fact, program code comprehension takes up 58% of a software developer's time (Hebig et al., 2020).

According to Hebig et al. (2020), students found that systematic code reading, documentation, and code comments helped in improving their understanding of software solutions. In addition, observing programs on execution, debugging, and testing the system also gave them a pragmatic and user-oriented understanding of a particular software system. Strategies employed towards software maintainability can be broken down into three main approaches: (1) managing software dependencies; (2) managing software complexity and capacity to diagnose problems; and (3) improving the capacity to make changes and fast and easy deployments (Kazman et al., 2020).

As for techniques in managing dependencies and complexity, software engineers are encouraged to employ proper encapsulation, a number of module dependency restrictions, and refactoring and reduction

of modules into smaller components (Kazman et al., 2020). Proper use of assert statements, logging to expose observable states of the software, and creating sandbox modes could be techniques to expedite the diagnosis of software defects (Kazman et al., 2020).

To manage deployments better, developers should employ feature toggling mechanisms, segmented rollouts, and rollback mechanisms. These give developers control over how and when features go live. Rollback allows them to recover in case of defective releases (Kazman et al., 2020). With respect to the development of web software systems, maintainability is particularly challenging due to its hybrid nature of program code and static content, and fast development pace (Chua et al., 2006). Employing existing architectural design patterns such as MVC and layered architectural approaches may help improve the maintainability of the software system (Sommerville, 2016; Kazman et al., 2020; Spray et al., 2021).

Hasan et al. (2023) discussed that with regard to web solutions, there is great interest in migrating from monolithic architecture to microservice-driven designs. This is due to the emerging trend of utilizing cloud technologies to tap into its strengths in scalability and maintainability. Several attributes were proposed in measuring maintainability of software systems, such as: (1) cohesion; (2) coupling; (3) complexity; and (4) size.

Cohesion refers to the level at which components and microservices are logically related. Coupling refers to the degree of interconnectivity of these different components. Having loose coupling and high cohesion is mentioned as the ideal target for good module decomposition. Complexity refers to the degree of connectivity of various microservices. This also serves as the level of dependencies with respect to the number of microservices. The size of the software can be represented as the number of classes of the module and the lines of code of the microservices (Hasan et al., 2023). As capstone solutions are predominantly web technology-driven with the possibility of adopting microservices, these aspects are critical to the lifespan of these deployed solutions for community extension projects.

Previous Software Engineering Courses

A capstone program can be used to develop and demonstrate the skills of students through experiential learning. In the work by Ye and Zhao (2023), their research focused on a capstone program tapping social media and analytics in the interest of a capstone client. Their program proved to be successful as an analytics program, but ours focuses more on information system solutions.

Bragós et al. (2022) conducted a large-scale study on their capstone projects involving project types, from ideation to technical. They had found that the nature of the project had a significant impact on the performance of the team, and they had also made note that there was no significant difference in performance between the team sizes and which academic term the project was conducted (Bragós et al., 2022).

Previous works have incorporated Agile development with a capstone course (Bastarrica et al, 2017; Fan, 2018; Fagerholm & Vihavainen, 2013; de Souza et al., 2015; Matthies, 2018; Li et al., 2023; Sim et al., 2023; Kapitsaki, 2024). Fan (2018) showed a support structure for guiding student groups. These key roles were Capstone Coordinator, Capstone Instructor, Faculty Adviser, and Industry Mentor, which cover a wide range of needs. UA&P (Ng, 2018) merged the Capstone Coordinator and the Capstone Instructor to cover the administrative and Software Engineering instructor responsibilities while keeping a Faculty Adviser role separate. Industry Mentors were sought on a case-by-case basis, noting that difficulties in the availability of these resources were consistent with the results (Fan, 2018).

In terms of evaluating such programs, Fagerholm and Vihavainen (2013) proposed the use of surveys to capture key attributes of teamwork: presence, activity, eagerness, devotion, contribution, expert maturity, process, and result. UA&Ps' capstone program chose to manage such aspects internally within the group and the Faculty Adviser while utilizing the survey to assess software quality using the ISO25010 standard. The utilization of this standard is consistent with the study by Spray et al (Spray et al., 2021). The work by Matthies (2018) incorporated lectures on Scrum and Kanban concepts into their capstone program so that students remain flexible in project execution. This is an idea that UA&P adopted in its own capstone program (Ng, 2018). Utilization of supplemental lectures to cover gaps for various students was recommended (Matthies, 2018).

Students had improved understanding of the software they were developing through the iterative approach (de Souza et al., 2015), which is consistent with previous results (Ng, 2018; Ng & Cruz, 2020). Digital platforms such as virtual meetings were adopted in the work by de Souza et al. (2015) with varying degrees of success. Some groups reported project management issues, while others credited the scrum framework for higher student engagement. Through the COVID-19 situation, the utilization of digital platforms became necessary for HEI operations (Al-Sholi et al., 2021). As such, similar virtual platforms were adopted in the work by Magno et al. (2022), citing issues with regard to online connectivity while utilizing these virtual communication platforms when resolving technical challenges.

Li et al. (2023) had conducted a study on large-scale capstone projects under advisement from their partner companies. Doing so would make the capstone more realistic for large-scale software development projects that industry practitioners do. They made note that students learned the importance of trust in the success of their capstone projects. Their work also raised the importance of knowing the roles and utilization of agile roles, such as Scrum master, to perform well in the capstone project (Li et al., 2023). While their study assumed that all students had the requisite technical skills, the UA&P capstone program differs, as the irregular students have control over their study plan and may reach the capstone program stage, having missed some subjects.

The work by Kapitsaki (2024) focused on the student adaptation of a Scrum-variant project-based course. Their data from 2020 to 2021 showed that students needed time to learn user story estimates and to become productive. The work by Sim et al. (2023) suggested that having students select their own members improves their teamwork. In addition, pair programming should be adopted as well. Students underestimating the teamwork-related challenges was a key finding from Bastarrica et. al. (2017) with regard to their software engineering course. These findings mirrored study results when consulting students (Ng & Cruz, 2020).

The study conducted by Gan et al. (2020) showed that capstone programs could be successful by closely analyzing the performance of their students across multiple clients, including major enterprises and government agencies. Their observations noted that infocomm and finance-related projects tend to perform better than health projects. However, their study focused on the Singaporean setting (Gan et al., 2020).

The study conducted by Alves & Rocha (2021) integrated a DevOps course, including Agile and Lean, with students working on Open-Source projects. Their course gave students the opportunity to manage real-world software development projects and understand the intricacies and challenges surrounding software deployment and delivery. Students had expressed that the course was difficult but rewarding, and they learned the importance of product and project management. The study also highlighted the importance of team versatility, where students needed to learn multiple skills and also practice responsibility and accountability when making decisions.

UA&P Agile Methodology with Scrum Adopted Capstone Program

The Capstone Program, crafted for the UA&P IT course, mapped an Agile methodology with Scrum phases into distinct stages of a typical IT student study plan, as shown in Figure 1.

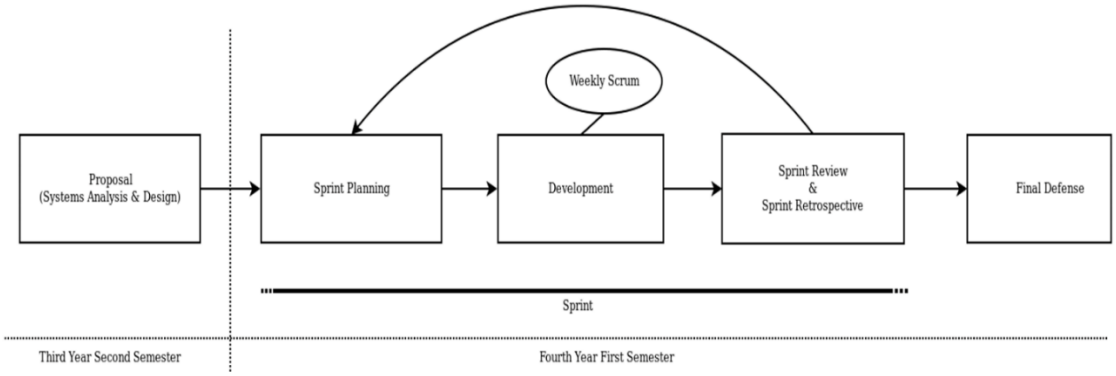

Figure 1: UA&P Agile methodology with Scrum framework adapted capstone program (Ng, 2018)

The initial stage of the capstone project is the proposal stage, where students are expected to find a client they will analyze to determine operational issues that can be alleviated with an IT solution. Once the proposal has been vetted, the capstone project will enter the Agile methodology with Scrum framework phases as discussed earlier (Ng, 2018). They are expected to have demonstrated and deployed their solutions in controlled environments, as the client is able to use and assess the solutions with respect to the problems identified.

HEIs and Community Extension Program

Various works show how HEIs executed their respective Community Extension Programs. Magnaye and Ylagan (2021) and Cristobal (2023) pursued livelihood training and additional community-related services such as computer literacy, environmental cleanup, and other resource-sharing activities for the uplifting of the adopted villages or barangays. In the case of Vera (2023), the adopted community was a child development center where resource sharing was provided, such as donations and the production of training materials. Managing the creation, archival, and destruction of sensitive government documents in a local barangay was the thrust of the research study of Requinto et al. (2019). They delivered an open-source document management system utilizing a Kanban Agile approach to address the specific needs of the government unit.

The creation of an information system for a barangay was the project of Gallera and Salvador (2023), who showed that usability, functionality, and maintainability were crucial factors to the success of the system. The created information system was deemed to have been able to improve the local governance and efficiency of the barangay through surveys from barangay officials and community members who used the system. These previous works inspired the present study to extend the UA&P IT Capstone Program to Community Extension Programs to gain a better understanding of the various nuances and factors to consider, instead of focusing on Filipino businesses that served as project locations in the past.

METHODOLOGY

In order to maintain consistency with previous works (Ng, 2018; Ng & Cruz, 2020), the researcher adopted a post-positivist worldview perspective as the results are influenced by familiarity with Agile

methodology with scrum framework practices (Robson, 2002). In addition, a descriptive research approach was followed, utilizing data from external observers of the execution of the Capstone program and the interaction and reaction of students with realistic situations involving the relevant stakeholders in developing and delivering software solutions (Shields & Rangarajan, 2013). In assessing the success of the software implementation and the performance of the students executing their respective capstone programs, surveys based on the ISO25010 standard were utilized, as well as interviews and observation journals written by the students. Such an approach is similar to previous works (Fagerholm & Vihavainen, 2013). The average was calculated similarly to previous work (Magno et al., 2022)

To capture industry and alumni insights, several alumni industry practitioners participated in a focus group discussion. This was succeeded by follow-up interviews conducted virtually and through e-mails. Lastly, to capture the insights of faculty advisers, similar virtual calls and email correspondence were done. As was done in previous works (Ng, 2018; Ng & Cruz, 2020), thematic content analysis was performed on the data from the focus group discussions, interviews, and correspondences to draw overall themes. The comments were then organized into a table where initial themes were identified and grouped together, and the process of theme identification was repeated until no other themes could be drawn.

In total, there were sixteen (16) participants on the initial survey from alumni and industry practitioners. Nine (9) alumni and industry practitioners engaged in the focus group discussions. The researcher had follow-up e-mails with two (2) alumni. For the current capstone student groups, three (3) responded to the focus group discussion. One was interviewed through email. Three (3) faculty advisers engaged through emails, and two (2) of them through interviews.

The comments from participants in the focus group discussion were transcribed and laid out in a table by the researcher. For every comment, initial themes were identified and written by the researcher. The comments were then grouped by the identified themes and were reviewed and adjusted based on overall trends discovered in relation to other grouped comments. This process of theme identification was repeated until no other themes could be drawn. For verification, the resulting themes and the grouped comments were reviewed by another faculty member of the department for correctness.

FINDINGS AND DISCUSSIONS

For the academic year 2023 to 2024, there were five capstone program groups, of which four groups were deployed within community extension programs. They were deployed to a barangay, a mental hospital, and a school. Three groups responded to focus group discussions and follow-up conversations. Other groups participated in online conversations.

For the alumni and industry practitioners, sixteen (16) from varying graduating batches responded to the initial surveys, while nine (9) responded to the focus group discussion invitation. In addition, the research study gathered insights from three faculty advisers. Surveys, interviews, and focus group discussions were transcribed for thematic content analysis.

RQ1. *As Agile methodology is part of modern industry practices, what could the IT Capstone program learn from industry practitioners who completed the same program?*

The study is interested in assessing whether the program was relevant and determining what could be improved. Tapping the alumni industry practitioners, they were asked whether the program was useful to them. Figure 2 shows the results from the 16 survey respondents.

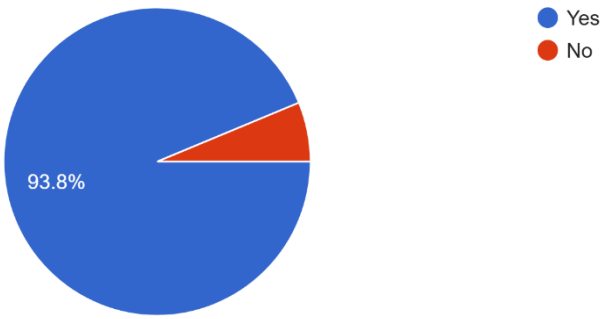

Figure 2: Responses of alumni and industry practitioners on the usefulness of the program (n=16)

While the question was simplistic in nature, it was deemed necessary to get an initial read on how good the program was for them. The researcher was pleased then that 93.8% thought the program was very useful. From discussions with alumni, there was appreciation of having familiarity with the overall process so that the learning curve at the actual companies they eventually ended up being employed with was greatly reduced. This is consistent with previous research findings (Ng, 2018; Ng & Cruz, 2020; de Souza et al., 2015). There was appreciation in understanding what the process did for the project and also learning to be flexible, adaptable, and systematic in resolving issues.

Given all that, it is not surprising that there were some who had selected No. Knowing that the program had been adapted for an education setting and deviations from actual practice had to be made, it was necessary to better understand the gaps in the program. With regard to the planning phase, several respondents discussed the differences of the Capstone Program from their experiences.

Table ***1*****:** illustrates the themes found with selected edited comments for the sake of brevity and clarity.

Table 1: Emergent themes in the planning phase derived from respondent feedback

Theme	Respondent comment
Preparation of the Product Backlog and Sprint differed between Capstone and industry practices	We had to prepare the list of tickets manually.
	We had to coordinate with the clients themselves.
	In actual work, the product owner had made the list of tickets.
	In my work, we actually took the time to understand the team capacity.
	We had a dedicated sprint for just preparing the backlog.
	For the sprint, we had properly defined tickets that could be added to the sprint if the team had extra time and capacity.
Task estimation was only fully understood at work	Only at work did I appreciate the story point estimation.
	We didn't understand why we had to estimate the work (in the capstone).
	The designation of story points was agreed upon by the team (at work).
Tasks were not thoroughly discussed in detail during the Capstone	In the capstone, we didn't revisit the ticket to refine the details on it, unlike in my team.
	While in capstone, we didn't really discuss the ticket with unit testing in mind (in comparison to work).

Three prevalent themes emerged from the collected responses from alumni: (1) Preparation of the Product Backlog and Sprint differed between Capstone and industry practices; (2) Task estimation was only fully understood at work; and (3) Tasks were not thoroughly discussed in detail during Capstone.

Preparation of the Product Backlog and Sprint differed between Capstone and industry practices

In terms of the first theme, the adaptation of the Agile methodology with the Scrum framework, there was a discussion on who could take the role of product owner. As the students were directly connected with the clients, it became natural that the responsibility of setting up the board was handled by the students themselves. This is distinctly different from industry experience as described by the alumni, where there was a product owner who acted as liaison to the client. Hence, the initial feedback like “In actual work, the product owner had made the list of tickets…” regarding the preparation of the product backlog was not surprising. This showed how assignments deviate slightly from role definitions (Fan, 2018).

As the community extension programs have actual liaison and contacts from within UA&P, the researcher thinks that having the representatives of UA&P act as a product owner becomes a natural step forward in addressing these concerns. This would mean that the management of the tickets, sprint backlog, and the product backlog would be managed by a representative of UA&P instead of by the students, alleviating some of the pressure and load from the latter.

Task estimation was only fully understood at work

In studying the alumni comments on preparing the backlog using an actual sprint, one faculty adviser mentioned that prior to the curriculum change of the capstone program that increased the number of subjects from two to three, there was an extra session in between semesters where student groups were briefed on the preparation of the product backlog and how the sprints would execute. While this was now handled in the semester, semestral load was deemed to be a contributing factor to having less prepared sprints and product backlog. The researcher views that reinstating the pre-semester briefing would be a good step to address this issue.

With regard to the refinement of tickets, currently, the faculty adviser and students would be charged with assessing the tickets, with the students discussing the estimates. One faculty adviser explained that story point estimates were used to determine how many tickets get scheduled for the sprint, but did not use any metrics to determine the capacity of the student group. Given this, the researcher determined that throughput metrics per sprint need to be reinforced by students and advisers. This will give the student group more confidence that they can deliver the tasks in the sprint.

Tasks were not thoroughly discussed in detail during the Capstone

In relation to the third identified theme, respondents explained that they did not take the time to fully discuss the tickets. According to one faculty adviser, the lack of thorough discussion could explain some development difficulties, as this could easily clear up implementation details and techniques. The researcher determined that this could be introduced to improve the team's performance within the sprint.

From the focus group discussion with the alumni, there were also themes identified when comparing the team dynamics of industry practitioners to student experiences during the capstone. The identified themes and responses were listed in

Table *2*.

Table 2: Differences in team dynamics between Industry and Capstone

Theme	Respondent Comments
Communication and team dynamics	In a professional setting, it's easier to call coworkers out when there are issues.
	Because my groupmates were friends, it was harder to tell them when something was wrong.
	At work, when you get feedback, it is really for the betterment of the task.
	In the professional setting, we had set up the workflow of the team.
	Instilling ownership and a sense of responsibility at work. As students, we sometimes get lazy.
	We had difficulties adjusting to our roles at the start, but we were able to overcome these challenges by highlighting our strengths and positioning each other for success.
	In the capstone, our group had to focus on our strengths and weaknesses to progress.
Insights from experienced mentors	At work, I joined a fast-moving team, but there were many members we could rely on for guidance.
	In the capstone, I appreciated the arranged meetings with alumni who shared their stories with us.
Team composition difference	At work, you don't get to pick who you will be working with.
	For the capstone, I only picked my friends as groupmates.
	Some students in the capstone are not ready for it.

Communication and team dynamics

Although on face value, none of the themes were surprising, these showed, overall, that the dynamics of industry work simply operate differently from students working on a capstone. In some cases, the sense of professionalism had not yet developed fully for students that is expected in the industry setting. This can be seen from the responses like "Instilling ownership and sense of responsibility". From discussion with the faculty adviser, while it would be ideal for students to immediately be responsible, different students reach different levels of maturity, and the capstone experience does help in instilling those principles.

In both the industry and in capstone, the manner in which teams performed was an important point to consider to succeed. As seen in comments such as "In the professional setting, we had set up the workflow of the team.", industry practitioners would find creating the workflow to be critical. From conversations with other alumni, it became apparent that in other cases, an alumnus would be entering a team that already set the work culture and had to adapt. Although students do not have this already decided, some groups were able to determine how best to implement the capstone. This can be seen with comments like "In the capstone, our group had to focus on our strengths and weaknesses to progress". The alumnus who completed it was able to communicate with the groupmates about creating a way to proceed.

Insights from experienced mentors

Interestingly, there were comments appreciating the scheduled talks from earlier batches on how the capstone was for them. To them, this gave the alumni additional insights into how to manage their capstone projects. The students were free to ask any questions they deemed necessary regarding the capstone and

challenges they were currently facing. This sense of guidance is not dissimilar from that of industry practitioners who are able to seek guidance from supervisors in their teams.

Team composition difference

The composition of the team also influenced the communication dynamics. In a capstone, for the alumni, typically, students were able to freely select their members, while in the industry, this was rarely the case. This free selection meant that students often would pick their friends, and that also meant they were not able to communicate effectively when there were points of crisis during development due to their existing relationships. From the consultation with a faculty adviser, however, there are batches that had randomized membership selection but that did not seem to show any distinct change. Some groups performed and groups did not, and that depended mostly on their maturity levels and how quickly they grew into their roles. Alumni responses corroborated this by mentioning that they performed better when they had the maturity level to handle unfiltered feedback and criticism.

Key takeaways

Proper planning and management of the product backlog, task estimation, and task clarity are areas of improvement in the capstone program. With effective management of the product backlog, students can focus more on the delivery of the tasks with a better comprehension of the solution being delivered. Utilizing UA&P community extension staff as product owners for this purpose could be effective.

Effective utilization of estimation and throughput metrics could positively improve student execution of the capstone program and also further align the program to the principles of agile and the Scrum framework. Throughput metrics can highlight better issues that a team experienced, as discussed by de Souza et al. (2015). Adherence to the principles of the scrum framework was also crucial for improvement in the team dynamics (de Souza et al., 2015). The capstone program should be tweaked to provide more exposure to estimation, such as using story points and throughput metrics, such as velocity. This challenge on user story estimates is consistent with previous works (Kapitsaki, 2024).

The team dynamics influence the level of communication and ultimately, the success of a project. Communication is a crucial component to the success of projects, as noted by de Souza et al. (2015) implementation of the Scrum framework. It is also a noted important graduate skill according to the study by Strain and Marshall (2023). The established workflow of the team and its maturity level were key takeaways for the success of both industry and capstone projects. This is also consistent with previous works that emphasized student responsibility in delivering tasks on time (de Souza et al., 2015; Ng & Cruz, 2020). Reinforcing students on agile practices for them to experience the team workflow and communication is a strong recommendation.

The support structure and guidance of mentors were also critical factors of success. The utilization of such mentoring guidance is consistent with the defined roles and experts discussed in previous works (Fagerholm & Vihavainen, 2013; de Souza et al., 2015; Fan, 2018; Li et al., 2023). Faculty advisers and expert advisers should be briefed to be available and provide adequate guidance to student groups, especially considering that the availability of such expert advice has been found to be difficult in previous works (Fagerholm & Vihavainen, 2013; Fan, 2018). These findings and recommendations are summarized in

Table ***3***.

Table 3: Recommendations based on key takeaways from the industry

Takeaway	Recommendations
Effective management of product backlog	Utilizing UA&P community extension staff as product owners.
Effective utilization of estimation and throughput metrics	The capstone program should be tweaked to provide more exposure on estimation, such as using story points and throughput metrics, such as velocity.
Communication with respect to team dynamics and workflow	Reinforce to students on agile practices for them to experience the team workflow and communication.
Support structure and mentoring guidance	Faculty advisers and expert advisers should be briefed to be available and provide adequate guidance to student groups.

RQ2. *What are the relevant considerations for the deployment and long-term use of software solutions, especially considering community extension programs?*

Through the responses from alumni and industry practitioners, the researcher discussed their thoughts regarding the possibility of deploying capstone solutions for company clients.

Table ***4*** illustrates the themes found and the selected comments on this particular topic.

Table 4: Insights from alumni and industry practitioners on capstone solution deployment for company clients

Theme	Respondent Comments
Client environment challenges	For actual production deployment, there were differences in the UA&P environment compared to the actual environment.
	There was no available setup for deployment on the client side.
	We won't be able to tell the client what server and hardware to procure long-term.
	Code copyright was an issue for long-term use.
	We have no experience with hardware like barcode readers, but if the client wants it in the future.
Fear of support expectations	We did not want our clients to ask us to troubleshoot and fix bugs.
	Clients were not clear about support.
	We were not sure whether we should be providing support.
Lack of confidence in the code quality	We didn't have the best code quality-wise.
	Our code was made for what is the current state of the client, but was not flexible for future changes.
	Our concern with deploying was that the system managed the salaries of the client's employees.
Optimism about the project	The state of the code is not bad, and we think it addresses the concerns of the client.
	The main deployment of the solution is still pending. In theory, it would work. In terms of testing, it would work.
	In terms of code and functionalities, it covered everything.

Client environment challenges

Many of the respondents note that the lack of actual infrastructure and resources for clients was a major hurdle to deploying the capstone. Others had pointed out differing environments within the university and the client infrastructure as points to consider. In addition, the alumni felt that when it came to long-term use of capstone solutions, their level of experience was not enough to advise what hardware their clients should be procuring. In discussing this matter with the faculty advisers, one agreed that the capstone program was focused more on the software development side than the deployment and maintainability aspects. Although the topics were covered on a theoretical basis in class and other subjects in the IT curriculum, the students at the time did not get to apply these concepts significantly beyond simply setting up servers and running their application code.

There were comments regarding the copyright issues, such as “Code copyright was an issue for long-term use”; the copyright of the capstone code ultimately resided with the university. As such, long-term use of a capstone solution would need to involve the university itself. Arrangements of that matter would be difficult if there were no official relationship between the client and the university. Noting that community extension programs already have an established official partnership, the matter of long-term support would reside with finding a team that could modify the code in the future.

Fear of support expectations

Another theme from the conversation was that many of the alumni did not wish to get attached to the project outside of the capstone period. Although clients should not expect to receive support outside of Capstone, the alumni students themselves were unsure about this. Moving forward, it should be made clear to the students and the clients that there should be no such expectations.

Lack of confidence in the code quality

Lack of confidence in the solution was also another factor. This was influenced by the quality of the code, as well as not investing in the code from a maintainability perspective. Although capstone solutions had implemented architecture patterns (Kazman et al., 2020) to improve maintainability, there was the sentiment that the code was rushed. One faculty adviser mentioned that the student experience level in coding would be a big challenge to their ability to write a code base with that long-term perspective towards adaptive, preventive, and perfective maintenance activities. Even though there was a lack of confidence as novices, the alumni who responded expressed optimism that the solution implemented addressed the client concerns identified.

In terms of technical factors to consider on the capstone, the alumni and industry practitioners had discussed the value of selecting the appropriate technology stack as opposed to during the capstone, where students would often simply select the technology stack taught in the IT curriculum. These were influenced by the pressure of additional academic load and the interest in implementing rather than learning a new language or framework. This is discussed in a previous work (Ng, 2018).

The selection of the technology stack would serve as a point of contention with company clients should the capstone group choose one that differs from the technology stack of the company client. Therefore, especially if the company client has the infrastructure in place, capstone groups should align their work with the technology stack of the company client to ensure compatibility.

Key takeaways

In positioning the capstone project solution for longevity, client infrastructure is a major hurdle. There are clients with inadequate equipment, infrastructure, and technology literacy required in maintaining and supporting the solution. Although students had these concepts in the course, they had experienced difficulties in advising and supporting clients in this manner, reinforcing findings that the industry views student competency and experience with skepticism (Alzamil, 2005; Jones & Davey, 2009; Marriska, 2015; Wilbur, 2016; Daff, 2021; Aljohania et al., 2022; Strain & Marshall, 2023). The researchers recommend reinforcing the student skillset through more live exercises on deployment and having students engage clients with training activities, especially during turnover. With regards to infrastructure,

clarity on the responsibilities for equipment procurement should be done early in the capstone program phase so that clients will be ready for the solution on deployment.

Students have fears of being attached to the software developed for far longer than the capstone program period. Such fears have influenced the eagerness to deploy and use the solution for the long term. Part of the fear is influenced by the lack of confidence in the code quality. The selection of the technology stack is a major factor that can influence the confidence and ease in deploying the solution. This can also influence the robustness of the core architecture of solutions developed by students. In addition, students should be encouraged to increase their testing activities across all sprints as this improves code confidence.

With regards to the support and maintenance phase of the solution, students should be encouraged to discuss and decide with their clients on the exact length of the maintenance window. These recommendations can be seen in

Table ***5***.

Table 5: Recommendations for long-term capstone solution use

Takeaway	**Recommendations**
Inadequate client competence and infrastructure	Reinforce the student skillset through more live exercises on deployment and having students engage clients with training activities, especially during turnover.
	With regards to infrastructure, at the earliest possible point, equipment procurement should take place.
Lack of confidence due to code quality	Encourage students to adopt an industry-standard technology stack and software development frameworks to improve code quality and to be deployment-ready.
	Encourage students to increase testing-related activities.
Length of maintenance and support	Encourage students to discuss and decide with their clients on the exact length of the maintenance window.

Faculty adviser insights on community extension programs deployed capstones

The researcher consulted faculty advisers involved in capstone programs that were initiated as part of Community Extension Programs to understand their differences compared to company-sponsored capstones.

Most adopted communities lacked infrastructure and computing resources

The adopted communities needed assistance in creating the modern IT infrastructure. This made onsite deployments of capstone solutions challenging, as the client would need to procure the appropriate hardware and also the actual networking setup.

Adopted communities rely on government funds as their operating budget

This factor greatly influenced whether the client could sustain the solution if subscriptions and other recurring costs needed to be paid. Government funding would mean waiting for the local government to approve the proposed item in their funding. This has a direct impact on whether the solution will remain online for the long term.

Most adopted communities did not have an IT team that could do long-term maintenance

Although the student solutions proved to be relevant and were actively used by the client, there could not be any proper turnover of the solution from a technical standpoint. This can influence how the solution operates in the long-term future, especially in the event of outages, software, and platform failures that could occur.

Comparing the company clients with adopted communities

In discussing these two types of capstone clients with faculty advisers, the researcher noted that the resources of company clients were dictated by the size of the company and that some could have networking infrastructure, while others may not. However, for many of the adopted communities, as their resources and funding are tied to government support, their options are often limited and, henceforth, needed assistance through Community Extension Program initiatives such as the one pursued within UA&P. These options range from whether they actually had the infrastructure to realistically utilize the software developed, down to whether they could cover the costs of subscription.

Computer literacy was also a challenge for adopted communities, as while they could be trained to operate the user level side of the implemented software, performing software and hardware maintenance would be a point of concern and would require significant training. In addition, unlike company clients of previous capstone groups, stakeholders of adopted communities lack sufficient knowledge of software engineering and are keener on functioning software than the rest of the aspects of software development, such as security and infrastructure.

For the training side, this could be carried out by the HEI supporting the adopted community. In regards to the evaluation of the solution, the students and HEI could shoulder this by inviting experts to evaluate the solution. With regards to the longevity of the solution, finding dedicated personnel from the adopted community side would be necessary to handle the maintenance aspects.

In effect, much of the capstone solution undertaking and deployment from the adopted community side would be absorbed by the HEI. The researcher believes that having the students submit complete documentation and provide system demonstrations and technical discussions to the department could be a workable solution in the interim to improve program understanding (Hebig et al., 2020). These recommendations are summarized in Table 6.

Table 6: Findings on adopted communities and recommendations through the community extension program

Takeaway	Recommendations
Lack of resources and infrastructure	HEIs can play an active role in supporting adopted communities by providing and donating machines, installing, configuring computer networks, and procuring servers.
Computer literacy challenges	HEIs can provide support and training, although staffing concerns are beyond the scope of these community extension programs.
	Student submitting complete turnover documentation and training to adopted communities.

RQ3. *Through thematic content analysis of the responses of current students of capstone projects, how do the insights drawn from industry practitioners and faculty advisers compare to the actual execution of the Community Extension Capstone Projects?*

In fully understanding the nuances of the capstone program that was adapted for community extension programs, the researcher looked into the four completed capstone projects that were assigned to adopted communities. Table 7 illustrates the assignments.

Table 7: Community extension Capstone program

Capstone Groups Assigned	Technology Stack	Community Extension Client
Capstone Groups A and B	Laravel and MySQL	Hospital
Capstone Group C	Django and MySQL	Barangay
Capstone Group D	Django and MySQL	High School

Capstone Groups A and B

The groups assigned to the Hospital reported that there was an IT infrastructure in place and an IT team. As that was the case, the teams were instructed to follow the tech stack of the client and were forced to learn Laravel and MySQL. The learning curves of Groups A and B were different, as Group A managed to start implementing approximately one month after learning. Group B had taken slightly longer to learn the technology stack, but there was less confidence in their own ability. As Laravel utilized an MVC (Model-View-Controller) architecture pattern, the team had implicitly adopted the pattern. This suggestion is consistent with a previous study (Kazman et al., 2020).

Deployment of the solution was primarily handled by the IT staff of the hospital, as the technology stack is the same as theirs. Hence, the developed systems would reside in the production servers of the hospital itself. The teams used a five-point Likert scale survey tool based on the ISO25010 standard, similar to other previous work (Spray et al., 2021).

Given the scores from the evaluation survey conducted by Capstone Group A,

Table *8* presents the average scores per criterion, demonstrating the satisfaction of the client over all on the implementation, as the scores were all above 3.

Table 8: Evaluation scores for Capstone Group A

Criteria	Score
Usability	4.4
Security	4.75
Reliability	4.33

Capstone Group B conducted its own evaluation to determine the acceptability of the solution implemented. Results are shown in

Table ***9***.

Table 9: Evaluation scores for Capstone Group B

Criteria	Score
Usability	3.8

Reliability	4
Performance	4
Acceptability	3.87

Given the acceptable evaluation of the client on the two solutions, the groups demonstrated the efficacy of the capstone solutions with respect to the identified issues of the client. As the technology stack is consistent, long-term maintenance could be managed by the adopted community client itself. To complete the turnover of the systems, a non-disclosure agreement (NDA) was signed between the HEI and the adopted community client due to the copyright on the software itself.

Capstone Group C

With the local barangay, Capstone Group C noted that there was no IT infrastructure setup. The team was able to freely select Django with a MySQL database backend for the technology stack of their solution. The team adopted a client-server architecture. In addition, utilizing Django as the underlying code base meant adopting the MVC MVT (Model View Template) architecture of Django (Olanipekun & Akolade, 2023). Evaluation of the technology solution was carried out utilizing the 5-point Likert scale based on the ISO25010 standard.
Table ***10*** presents the results. Given the positive results, deployment of the solution was carried out through DigitalOcean for evaluation with key stakeholders.

Table 10: Evaluation scores for Capstone Group C

Criteria	Score
Usability	3.625
Reliability	3.875
Performance	3.75
Supportability	3.75

Although the client had favorably evaluated the solution, however, given that the barangay had no funding from the national government for the online subscription, the students had to take down the solution. Further exploration was carried out on possible solutions, such as deploying the solution on a local server. Due to a lack of time, limitations on the identified server, such as its availability, and further limitations on access and usage brought about by the students being unfamiliar with configuring and managing the local server, the solution was not migrated to any local server. In addition, one of the developers mentioned that the assigned faculty adviser was not very helpful as he was more focused on the front-end.

As the capstone group had already completed the program and graduated, there was very little interest in resolving the deployment issues. According to one faculty adviser, the system was very good, but the subscription cost was too much for the barangay to shoulder.

Capstone Group D

With respect to Capstone Group D, they discovered that the client was gradually upgrading their infrastructure, but the number of computing resources was based only on the donations received from the HEI. As such, the group decided to use a cloud solution with assurances from the client that they could secure funding for the deployment. With this, the group was not restricted in the technology stack they could use and opted for Django. Initially, the team used SQLite3, but after stress testing the database, the team switched to MySQL.

Table ***11*** tabulates the software evaluation from Capstone Group D. Given the scores from the evaluation, the group deemed the solution suitable for the school they collaborated with.

Table 11: Evaluation scores for Capstone Group D

Criteria	Score
Usability	5
Reliability	5
Performance	5
Supportability	5

The deployment of the system was done through the integration of DigitalOcean and GitHub. Utilizing this automated deployment feature from DigitalOcean, the team was able to conduct rapid testing and bug fix rollouts to improve the state of the code and confidence level of the team. This finding is interesting considering that the alumni focus group discussion had mentioned the lack of confidence, but also not a large focus on QA, while they were developing their own capstone software.
Utilizing DigitalOcean required a subscription for which the school had some difficulties securing the funds and processing payments. In effect, the students were only able to support the deployed solution for a month. Unfortunately, the students had to take it down until proper funding from the client could be sorted out. Eventually, the funding was secured by the school, and the deployed solution was migrated to the school subscription.

As regards the long-term use of the system, all the key stakeholders had agreed to use the regular site visits conducted by the HEI to the school to identify new functionalities, changing requirements, and other maintenance activities that may be needed. In addition, one of the developers of the system had committed part-time to supporting the development work for as long as the system is in use. To further make the system maintainable, extensive documentation and video guides were set up by the team to support using the system. Such documentation effort mirrors similar previous study results (Hebig et al., 2020).

When discussing this level of commitment with the faculty adviser, the researcher concluded that while this benefits the adopted school greatly and the system, this is unusual, as the alumni made it clear they were not interested in committing to the capstone client long-term. In addition, other proponents of the team were not keen on helping to support the system long-term. Overall, the faculty adviser and the liaison to the capstone client had very positive feedback on the solution provided, as it was extensively used during their enrollment.

Themes drawn from the focus group discussion

For the focus group discussion, members of three of the capstone groups attended and shared their insights. As with the alumni and industry practitioners, the researcher wanted to get an understanding of the key takeaways from the students themselves on the Agile methodology with the Scrum framework adopted in the Capstone Program.

Table ***12*** illustrates the themes drawn from the focus group discussion with the students.

Table 12: Feedback on the concluded Capstone program

Theme	Student Responses
Effective communication	Easy and fast communication with everyone.
	Clarifications were immediately discussed with the client.
	Feedback from partners was very open and on time.
The program was an effective learning platform	Capstone 2, I was more confident with my skills.
	But I can say a bulk of my learning was in the capstone throughout my BSIT experience.
Effective and swift workflow	We did the bulk of the work between cap 1 and cap 2 during the summer break.
	After the first week of Cap 2, we showed the stakeholders the finished product.
	Feature complete for main crud and base code, we just needed to do bulk importing and sectioning left by cap 2
	Simply asked for revisions from stakeholders by the last semester.

Overall, student responses have been very positive regarding the program. They were able to settle on a good workflow for their team. The realization that the Agile methodology allowed for good communication and fast response in the event of a crisis exhibited the understanding of how this form of management allowed them to be flexible to any situation.

In addition, the program was able to instill practical knowledge in various areas of the software development process. The researcher is confident that the program helped address concerns raised regarding IT graduates (Alzamil, 2005). In addition, the adopted community clients clearly expressed how helpful the solutions were from evaluations that address issues raised in other previous studies (Alzamil, 2005; Jones & Davey, 2009; Marriska, 2015; Wilbur, 2016; Daff, 2021; Aljohania et al., 2022; Strain & Marshall, 2023).

Technical Aspects of the Software Development

Regarding the technical aspects of the software development, several themes were drawn from the focus group discussion. These are listed in

Table ***13***.

Table 13: Themes regarding the software development

Theme	Student Responses
Learning a good technology stack	We had to learn a new language, but the learning curve was very quick.
	Overall, relatively easy to work with.
	The framework made it easier for some of us to develop with production in mind.
	It was easy, and we could deploy the features faster.
	Development phase, it was all in our control.
Missing subjects for irregular students caused fears	Capstone was a struggle for me. I had not taken Computer Programming 2 yet.
	I was more anxious at the beginning because I didn't have the important courses.
	When I was in Capstone, I was not sure if my skills were up to the task because the fundamentals of the team were lacking in some aspects.
	I actually didn't know about virtual environments early on.
Skill reinforcement	Don't know how to deploy the system locally. I think it's a skill issue.
	Security scares me because we don't know if we did enough.
	I think we had lots of memorization in class on security.
Freedom to select the technology stack	Sir, they did not have servers, so we could pick the language we wanted.
	Their IT team told us to use Laravel. No choice.
	There was no infrastructure, so we followed the advice from our info man teacher.
	We asked, but for deployment, there were no restrictions imposed by the client.
	Originally, we were going with Java, but our adviser told us to use Python instead.

Code maintainability challenges	I used to use code comments. During Capstone 2, code legibility went out of the window.
	Looking back, the logic in the code is very inefficient.
	Knowing what I know now, I could have written less brittle code.
	We heavily commented on code that needed to be explained, knowing that the stakeholders don't have an IT background. But looking back, even though we don't know how this code works, even with the comments.

Depending on whether the adopted community client had a mature IT division, capstone groups could decide on the technology stack they wanted to use. What would influence their choice would be down to the adviser and other resource persons, and industry experts.

The selection of a mature technology stack was crucial in the development pace of the team. As shown in Table 7, the groups had gone with Laravel and Django, which are mature web frameworks. The respondents mentioned that they were able to pick up the learning curve relatively quickly and develop the system. This is despite fears about development skills from students who had a different study plan from the standard curriculum. The quick development pace corroborates with themes drawn in

Table ***12***, where a swift workflow was established.

That being said, the skill set of the students was lacking in other aspects. While they were comfortable with the development side, they realized there were gaps regarding the deployment aspects and security of the overall solution. In addition, some of the students were irregular, signifying that their study plan was not the standard due to academic or behavioral issues, as well as other extraneous circumstances such as health or family issues. These can influence the performance of students and the team. This is consistent with findings from previous works (de Souza et al., 2015; Ng & Cruz, 2020).

While there is little that could be done from a course standpoint regarding personal and family matters of a student, the researcher identified the academic areas in discussions with faculty advisers, and hence, there is a need to reexamine how these aspects can be reinforced. As noted by one student, they would like to have more practical applications instead of memorization. The discovered gap corroborates the issues raised by previous works (Alzamil, 2005; Jones & Davey, 2009; Marriska, 2015; Wilbur, 2016; Daff, 2021; Aljohania et al., 2022; Strain & Marshall, 2023).

In terms of code maintainability, the students who responded discussed trying to adhere to code quality and full code documentation, as this is consistent with the principles of Agile methods (Apke, 2015). However, the code quality had been deprioritized or was not adequate, depending on the team. Some of the students felt the project was in a good state already, and quick coding was adopted for the tweaks. Others had thought the work was adequate, but, in retrospect, realized the documentation was not enough. As noted by one of the faculty advisers, this does take some level of experience to get better at. Previous conversations with other alumni confirm this as well.

Lastly, the students were unanimous in that the government funding of the adopted community was a major factor regarding the deployment of the solutions and their long-term use. The instances where the solution was taken down were all from the point of view of the adopted community not being able to sustain the expenses of the subscription. Areas of improvement based on these technical discussions are shown in

Table ***14***.

Table 14: Areas of Improvement based on Student Feedback

Student Feedback	Recommendation
Code quality concerns	Reinforce future batches to pursue industry-standard frameworks.
	Reinforce coding quality practices and testing-related activities.
More focus on deployment-related activities	Recommend more practical applications of post-development activities and security.
Client resources and infrastructure impacting adoption	Recommend concrete discussions between students and the client regarding equipment and funding so that preparations can be made.

In terms of code quality concerns, there is a need to reinforce the future batch of students to select industry-standard coding frameworks, as these were shown to be effective based on student feedback. However, the attention to testing activities and the verifiability of the code needs to be reinforced. With students expressing concern over experiences in deployment and security, we recommend that more attention be given to these areas in the course, especially in the capstone phases.

A key impediment in the adoption of the solution resided in the client environment, whether it was infrastructure or equipment equipment-related, or funding for sustainability reasons. These are areas we recommend the future batch of students to have conversations with their clients to prepare for the eventuality of production deployment and usage of the solution.

Summary

The researcher found that the capstone program had generally been appreciated by both alumni and industry practitioners, and students who had completed the program, as noted in Figure 2 and

Table ***12***. Consistent with discussions with the faculty advisers regarding the IT infrastructure of adopted communities, the capstone group selected a technology stack, and deployment options were influenced by whether the adopted community did not have a proper IT infrastructure.

Long-term technical support and software updates on the solution remained an issue if the adopted community client did not have a dedicated team that would take over the development side. This support team is an established entity, as discussed in a previous study (Sommerville, 2016). The researcher suggests HEI that supports the adopted community to provide training, consistent with previous works (Buys & Bursnall, 2007; Buenafe et al., 2024).

Both the faculty adviser and students pointed out how crucial it is for the sustainability of capstone solutions to secure the government budget. Although some secured the funding, others were waiting on approval and could not manage the cost. Both the alumni and industry practitioners and the capstone groups had taken note of the significance of selecting the proper technology stack in creating the software solution and the pace of development. Code quality and maintainability of the capstone software were a point of concern from both the capstone groups and the alumni and industry practitioner sides. Their lack of confidence also stemmed from their lack of experience at the time.

CONCLUSION

This study drew insights from alumni and industry practitioners, faculty advisers, and capstone groups to understand Agile methodology with the Scrum Framework Capstone Program as it is implemented for community extension programs. As noted in RQ1, to more closely align the capstone program to industry standards, the following recommendations were found: (1) a more meticulous sprint planning and preparation could be done; (2) community extension staff be used as product owners on the capstone projects; (3) reinforcing agile practices to enhance communication and effective workflow; and (4) being able to carry forward mentorship from previous capstone groups mirrored the same mentorship established in the industry.

In responding to RQ2 and the deployment of solutions, themes were identified: (1) the varying client environments and resources; (2) the disinterest of students to provide support; and (3) lack of confidence in the code quality. These themes were corroborated with RQ3. In addition, the nature of the adopted community client dictated the limitations of the solution regarding deployment. In relation to RQ3, students raised code quality concerns, lack of experience in deployment and security, as well as client environment, resources, and funding affecting the adoption of the solution for long-term use.

From the findings, the recommendations in addressing these identified themes are: (1) to provide more deployment-related class activities; (2) to prepare complete documentation and training during the turnover phase; (3) establish agreements regarding equipment procurement and maintenance period; and (4) position student capstone projects to adopt good technology stacks and testing-related activities. From a capstone program standpoint, there needs to be constant communication across faculty advisers and students with regard to their suitability.

For future work, it would be interesting to revisit various aspects of the IT curriculum to reinforce the skills for security and deployment. From the community extension capstone program standpoint, it would be good to study this aspect further to determine how to address gaps regarding long-term support of the solution and equipping adopted communities with technical skills and IT infrastructure to use the solutions. In contrast to adopting cloud solutions, an on-premises one would be much more appropriate.

ACKNOWLEDGMENTS

The author acknowledges with sincere appreciation the valuable support of Dr. Henry Ligot, Mr. Anthony Saul, and the UA&P IST Department, as well as the contributions of the capstone groups, alumni, and adopted communities whose participation was integral to this study.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

AUTHOR CONTRIBUTIONS

Giuseppe Ng: Conceptualization, Original draft preparation. Data curation, Writing- Reviewing and Editing.

DECLARATION OF GENERATIVE AI

The authors declare that no generative AI was used in the writing of the manuscript.

DATA AVAILABILITY STATEMENT

Data available on request due to privacy/ethical restrictions.

REFERENCES

Agarwal, M., & Majumdar, R. (2013). Software maintainability and usability in an agile environment. *International Journal of Computer Applications (0975 – 8887) , 68*(4).

Aljohania, N., Aslama, A., Khadidosa, A. O., & Hassanb, S.-U. (2022). Bridging the skill gap between the acquired university curriculum and the requirements of the job market: A data-driven analysis of scientific literature. *Journal of Innovation & Knowledge Vol. 7(3)*.

Al-Sholi, H. Y., Shadid, O. R., Alshare, K. A., & Lane, M. (2021). An Agile educational framework: A response for the Covid-19 Pandemic. *Cogent Education, 8*(1).

Alves, I., & Rocha, C. (2021). Qualifying software engineers undergraduates in DevOps - challenges of introducing technical and non-technical concepts in a project-oriented course. *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET).* IEEE.

Alzamil, Z. (2005). Towards an effective software engineering course project. *In Proceedings of the 27th International Conference on Software Engineering 2005 (ICSE 2005)* (pp. 631-632). IEEE.

Apke, L. (2015). *Understanding the Agile Manifesto: A brief and bold guide to Agile.* Symone Huggins.

Bastarrica, M. C., Perovich, D., & Samary, M. M. (2017). What can students get from a software engineering capstone course? *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET).* IEEE/ACM.

Bragós, R., Aoun, L., Bermejo , S., Pegueroles , J., & Charosky, G. (2022). Analysis of students' performance in Capstone projects. *Proceedings of the International CDIO Conference.* Reykjavik, Iceland: CDIO Initiative.

Buenafe, C. P., Mendoza, J., Ng, G., Saul, A., & Tolentino, S. (2024). A descriptive research exploration of online modality in providing for equity, diversity, and inclusion. *International Conference on UN Sustainable Development Goals and Social Innovation.*

Buys, N., & Bursnall, S. (2007). Establishing university–community partnerships: Processes and benefits. *Journal of Higher Education Policy and Management, 29*(1).

Castillo, L. S. (2018). Call of Duty: A case study of ICT integration in Philippine provincial public schools in San Isidro Davao Oriental post K-12 implementation. *JPAIR Multidisciplinary Research, 30*, 1-16. https://eric.ed.gov/?id=ED583178

Chawla, H. (2016). *Agile methodology in large enterprises.* Retrieved from http://www.itbusinessnet.com/article/Agile Methodology-in-Large-Enterprises-4751403

Chua, C. E., Purao, S., & Storey, V. C. (2006). Developing maintainable software: The readable approach. *Decision Support Systems, 42*(1), 469-491.

Cristobal, E. B. (2023). The community extension of university as a driving factor to societal reform. *Journal for Educators, Teachers and Trainers, 14*(2).

Daff, L. (2021). Employers' perspectives of accounting graduates and their world of work: Software use and ICT competencies. *Accounting Education, 30* (5).

Dam, H. K., Tran, T., Pham, T., Ng, S. W., Grundy, J., & Ghose, A. (2018). Automatic feature learning for predicting vulnerable software components. *IEEE Transactions on Software Engineering, 47*(1).

de Souza, R. T., Zorzo, S. D., & da Silva, D. A. (2015). Evaluating capstone project through flexible and collaborative use of Scrum framework. *2015 IEEE Frontiers in Education Conference (FIE).*

Denning, S. (2016, December 9). *The age of agile: What every CEO needs to know.* Retrieved from Forbes: https://www.forbes.com/sites/steved enning/2016/12/09/the-age-of-agile-what-every-ceo-needs-to-know/

Fagerholm, F., & Vihavainen, A. (2013). Peer assessment in experiential learning, assessing tacit and explicit skills in agile software engineering capstone projects. *2013 IEEE Frontiers in Education Conference (FIE).*

Fan, X. (2018). Orchestrating agile sprint reviews in undergraduate capstone projects. *2018 IEEE Frontiers in Education Conference (FIE).*

Fernandez, S. (2024). Agile software development - recent advances and best practices: Analyzing recent advances and best practices in agile software development methodologies for improved project management. *Journal of Artificial Intelligence Research and Applications, 4*(1).

Gallera, J. M., & Salvador, A. S. (2023). Assessment of digital information systems for local barangays. *International Research Journal of Advanced Engineering and Science, 8*(2), 112–15.

Gan, B., Ouh, E., & Lee, Y. (2020). A longitudinal study of a capstone course. *Proceedings of the 26th Americas Conference on Information Systems (AMCIS 2020)*, (pp. 1-9). Salt Lake City.

Hasan, M. H., Osman, M. H., Admodisastro, N. I., & Muhammad, M. S. (2023). From monolith to microservice: Measuring architecture maintainability. *International Journal of Advanced Computer Science and Applications, 14*(5).

Hebig, R., Ho-Quang, T., Jolak, R., Schröder, J., Linero, H., Ågren, M., & Maro, S. H. (2020). How do students experience and judge software comprehension techniques? *ICPC '20: Proceedings of the 28th International Conference on Program Comprehension.*

Herbert, N. (2018). Reflections on 17 years of ICT capstone project coordination. *In Proceedings of the 49th ACM Technical Symposium on Computer Science Education-SIGCSE, vol. 18* (pp. 215-220). Baltimore, Maryland, USA: Association for Computing Machinery.

Jones, G., & Davey, J. (2009). *Stakeholder perspectives on the potential for student consultancy-based projects between a business school and SMEs.* Australian and New Zealand Academy of Management (ANZAM).

Kapitsaki, G. M. (2024). Students' perceptions on adaptations and workload versus artifacts in a project-based Scrum-variant course. *IEEE Transactions on Education, 67*(5), 639-647.

Kazman, R., Bianco, P., Ivers, J., & Klein, J. (2020). *Maintainability* (Technical Report CMU/SEI-2020-TR-006). Software Engineering Institute. https://doi.org/10.1184/R1/12954908

Li, Z., Arony, N., Devathasan, K., & Damian, D. (2023). "Software is the easy part of Software Engineering"- Lessons and experiences from a large-scale, multi-team capstone course. *ICSE-SEET'23: Proceedings of the 45th International Conference on Software Engineering: Software Engineering Education and Training* (pp. 223-234). IEEE Press.

Magnaye, R. P., & Ylagan, A. P. (2021). Effectiveness and impact of community extension program of one Philippine higher education institution as basis for sustainability. *Asia Pacific Journal of Academic Research in Business Administration, 7*(2).

Magno, A. E., Belo, P. A., Miranda, C. M., & Ng, G. C. (2022). An Agile approach to a project management information system amidst the COVID-19 pandemic. *Proceedings of the Research Colloquium on the Digitization of Business Processes in the New Normal 2022.*

Mandale, R., Patil, A., & Adamuthe, A. (2021). Improved undergraduate software capstone project development with adoption of industry practices. *Journal of Engineering Education Transformations, 34* (4). https://doi.org/10.16920/jeet/2021/v34i4/153757

Marriska, A. (2015). *Implications of client involvement in student projects: A comparative study between a project without a real client and a project with a real client*. [Master's thesis, Blekinge Institute of Technology, Sweden]. DiVA Portal. https://www.diva-portal.org/smash/get/diva2:920636/FULLTEXT01.pdf

Matthies, C. (2018). Scrum2kanban. *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials - SEEM'18.*

Ng, G. (2018). A study of an Agile methodology with Scrum approach to the Filipino Company-Sponsored I.T. Capstone program. *International Journal of Computing Sciences Research, 2*(2), 68-88.

Ng, G. (2019). Emerging themes on Filipino companies' needs towards quality management. *Innovatus, 2*(1).

Ng, G., & Cruz, R. V. (2020). Student and faculty adviser insights in an agile methodology integrated Filipino company-sponsored I.T. capstone program. *International Journal of Hybrid Information Technology, 13*(2), 33-44.

Nimer, B. B. (2020). *Evaluating an extension program through the lens of the human security approach* [Doctoral dissertation, Notre Dame of Kidapawan College]. ResearchGate. https://doi.org/10.13140/RG.2.2.19692.18569

Olanipekun, & Akolade. (2023, February 13). *Django MVT: An introduction to the model-view-template architecture of a Django app*. Hashnode. https://akolade.hashnode.dev/django-model-view-template-architecture

Parsons, A., & Lepkowska-White, E. (2009). Group projects using clients versus not using clients: Do students perceive any differences? *Journal of Marketing Education, 31*(2), 154-159.

Quezada, R. G. (2014). La responsabilidad social universitaria como política pública: Un estudio de caso. *Documentos Y Aportes En Administración Pública Y Gestión Estatal, 14*(22), 103–127. Universidad Nacional del Litoral. https://www.redalyc.org/pdf/3375/337531584004.pdf

Requinto, A. M., Macapagal, R. A., Cortez, C. A., Pineda, A. A., Canlas, F. Q., & Natividad, R. (2019). Implementing Document Management System (DMS) technology in Barangay Paligui, Apalit, Pampanga. *Innovatus, 2*(1).

Rigby, D., Sutherland, J., & Takeuchi, H. (2016). Embracing Agile: How to master the process that's transforming management. *Harvard Business Review, 95*(5), 40-50.

Robson, C. (2002). *Real world research: A resource for social scientists and practitioner-researchers* (2nd ed.). Blackwell Publishers.

Schachter, D., & Schwartz, D. (2009). The value of capstone projects to participating client agencies. *Journal of Public Affairs Education, 15*(4), 445-461.

Shields, P. M., & Rangarajan, N. (2013). *A playbook for research methods: Integrating conceptual frameworks and project management.* New Forums Press.

Sibal, C. (2019). Impact assessment of extension program of the College of Public Administration, Cagayan State University, Carig Campus. *International Journal of Advanced Research in Management and Social Sciences, 8*(6).

Sim, Y. H. R., Lua, Z. Z., Kelaver, K. K., Chua, J. Q., Lim, I. Z. J., Cao, Q., Keoh, S. L., & Lim, L. H. I. (2023, August 7–9). Experiences and lessons learned from real-world projects in software engineering subject. In *Proceedings of the 35th IEEE International Conference on Software Engineering Education and Training (CSEE&T)* (pp. 162–166). IEEE. https://doi.org/10.1109/CSEET58097.2023.00036

Smith, R. (2016). *Scrum guide: Agile project management guide for scrum master and software development team.* CreateSpace Independent Publishing Platform.

Sommerville, I. (2016). *Software engineering (10th ed).* Pearson Education.

Sprague, M., & Hu, O. (2015). Assessing the value to client organizations of student practicum projects. *Journal of Public Affairs Education, 21*(2), 263-280.

Sprague, M., & Percy, R. (2014). The immediate and long-term impact of practicum experiences on students. *Journal of Public Affairs Education, 20*(1), 91-111.

Spray, J., Sinha, R., Sen, A., & Cheng, X. (2021). Building maintainable software using abstraction layering. *IEEE Transactions on Software Engineering, 48*(11).

Strain, J., & Marshall, J. (2023). Mismatch in academia and industry: An exploration of perceived graduate competencies and industry expectations in information technology. In *Proceedings of the ISCAP Conference* (Vol. 9, Article 5962). Information Systems and Computing Academic Professionals. https://iscap.us/proceedings/2023/pdf/5962.pdf

Sunio, V., Gaspay, S. M., Rivera, J., Pontawe, J., Billones, R., Mora, R., & Afundar, K. (2022). Development of intelligent transportation system from the perspective of local government units, national government agencies, and transportation cooperatives. *Innovatus: A Journal on Computing Technology Innovations, 5*(2).

Vera, J. S. (2023). Awareness and effects of extension services on the adopt-a-day care school program. *East Asian Journal of Multidisciplinary Research, 2*(2).

Wilbur, J. (2016). *Meeting key challenges in delivering client-based learning programs within MBC programs.* EduSourced White Paper.

Ye, C., & Zhao, L. (2023). Teaching tip: A social media analytics capstone research project with community engagement. *Journal of Information Systems Education 34(4)*, 406-417.