Agile Approach to Enhance Student's Capstone (Industry-based) Product Delivery in Tertiary Education

BELAL CHOWDHURY[1], NASREEN SULTANA[2], SARAH CHOWDHURY[3], SUMYIA CHOWDHURY[4]

[1]School of Information Technology & Engineering,
Melbourne Institute of Technology,
288 La Trobe Street, Melbourne VIC 3000,
AUSTRALIA

[2]Monash Health Emergency Department,
Dandenong Hospital,
135 David Street, Dandenong VIC 3175,
AUSTRALIA

[3]The Royal Victorian Eye and Ear Hospital,
32 Gisborne Street, East Melbourne VIC 3002,
AUSTRALIA

[4]Department of Physiotherapy, Peninsula Campus,
Monash University,
McMahons Road, Frankston Victoria 3199,
AUSTRALIA

Abstract: - Opportunities to encounter the challenges involved in doing real-world industry projects, including project design, implementation & management, working under constraints, meeting deadlines, and uncertainties. An agile approach is applied to increase student learning in students' capstone projects involving external, real-world industry projects. In the realm of software and enterprise system development, agile methodologies have emerged as the most widely adopted approach, as they promote early and ongoing feedback that aids in aligning and adjusting the development process to meet the client's requirements. These methodologies empower students to prevent procrastination and improve learning objectives related to the design and development of capstone projects. This paper includes a case study to demonstrate an agile approach in capstone projects for both undergraduate and post-graduate students in public tertiary Institutions in Australia. Where students utilised an agile methodology, with the best practices of Scrum with three sprints in semester 1, 2024. The result has shown improvements in students' active participation, engagement, motivation, and improved understanding of agile practices within project teams and capstone product development. The results of this research can be integrated to modern engineering education, which ensures that students are well prepared for real-world challenges and not only technically proficient but also adaptable and ready to address the complex challenges of the 21st century.

Key-Words: - Capstone Product, Agile Approach, Scrum Method, Project-based Learning, Scrum, Industry Client and Modern Engineering Education.

Received: March 16, 2024. Revised: December 17, 2024. Accepted: February 13, 2025. Published: April 2, 2025.

1 Introduction

Tertiary education widely embraces project-based learning (PBL) as a favoured teaching approach, [1]. PBL students focus on acquiring essential deep learning skills like effective communication, teamwork, critical thinking, creativity, and innovation, which are crucial for future career opportunities in their industries. Capstone units concentrating on practical industry applications are integral to academic degree programs across diverse fields such as business, engineering, information technology, healthcare, and education at the tertiary level. Research has indicated that employers place a higher value on a student's ability to apply knowledge to address real-world challenges, [2]. When students enrol in a capstone unit, they collaborate with industry clients/experts to learn in-depth about a practical topic over a defined timeframe, with the goal of creating a product that will significantly enhance their career prospects. The central challenge for IT and enterprise system graduates is to effectively design, implement, and manage them quickly, [3]. However, a graduate capstone unit can demonstrate a program's educational effectiveness. Students acquire skills to research complex topics, solve problems, develop strategies, manage their time effectively, organise their tasks, and navigate obstacles. PBL requires students to collaborate in teams to tackle a real-world problem through hands on learning. Thus, tertiary education must provide students with hands-on experiences and practical learning skills that will prepare them to be competitive graduate students.

PBL produces three important outcomes: responsibility, independence, and discipline. The author investigates how PBL boosts creativity, relevance to real-world scenarios, teamwork skills, and the utilization of technology. The evolving and dynamic technological landscape in modern organizations necessitates that information technology/systems graduates acquire the ability to independently learn new technologies. PBL can help cultivate independent thinking and learning skills, [4]. The article details the specific steps for implementing PBL, emphasizing driving questions, contextual inquiry, collaboration, educational technologies, and artifacts as crucial components of this educational method. Numerous studies have been conducted regarding engineering capstone projects, [5]. It has been noted that individual projects proved to be more effective in Europe and Australia, according to earlier research from 2010. A higher number of students published papers when working on individual projects, [6]. Research from 2013 provides a comprehensive explanation of strategies to enhance final-year project and dissertation work, [7]. Research from 2006 and 2008 explores the effects of program outcomes, their implementation, and the various factors that influence these elements, [8], [9]. Research conducted in 2013 explored into the evaluation and assessment of capstone projects based on outcomes, [10]. Many academics have used agile methodologies in their class projects, with generally positive outcomes, [11]. Students familiarise themselves with agile methodologies during their studies, which include a project management class featuring concepts, case studies, and a project that employs Scrum. In the capstone unit, students get the chance to utilize their acquired knowledge by overseeing a significant project through agile methodologies. The rest of this paper is organized as follows: Section 2 covers related work that explores the basics of capstone courses, PBL, and agile methodologies. Section 3 discusses the importance and motivation of the agile approach in tertiary education. Section 4 outlines the self-managed industry-based project collaboration model and its implementation at Federation University to deliver capstone projects effectively. Section 5 discusses the practical applications in modern engineering education. Section 6 includes the conclusion and propose directions for future research.

2 Related Work

A capstone unit is positioned at the conclusion of the curriculum, enabling students to evaluate and present their development in relation to the program's objectives, [12]. Capstone units revisit program aims and assist students in a systematic reflection process that fosters self-directed learning and the ability to convey their academic accomplishments to professional colleagues, [13]. Capstone units also motivate students to take responsibility for their own learning, [14]. Students involved in PBL experience greater control over their education and the assignments they complete. These projects are typically less structured and more complex compared to traditional tasks, [15]. These projects often lack a single correct solution [16], which can lead to frustration for students yet offers a more genuine learning experience. Participants can enhance their creativity and research skills while becoming more engaged in devising project solutions. Students involved in these assignments displayed a more positive outlook toward the subject matter, [17]. In 2015, PBL was introduced in an environmental education course, where students tackled projects focused on solving environmental issues. In 2006, corporate clients collaborated with student teams to produce market research for their organizations. Students engaged in PBL typically concentrate on a final project that is often anticipated to possess high quality. Another hurdle for educators seeking to work with real-world clients is convincing the business sector of the benefits of participating in these experiences, [18]. Different educational environments have piloted PBL, such as in a mobile app development class and a computer graphics class. Although research generally yields favourable outcomes, several challenges are often encountered when utilizing PBL. Issues such as unhealthy group dynamics, ineffective time management, project-related stress, and communication difficulties are commonly reported, [19].

Businesses are increasingly incorporating agile methodologies into their development and planning processes. Most systems analysis and design, as well as software engineering departments, cover both traditional and agile software development practices. The Manifesto for Agile Software Development emphasizes four core values: valuing individuals and interactions more than processes and tools, prioritizing working software over comprehensive documentation, encouraging customer collaboration over contract negotiation, and embracing change rather than rigidly adhering to a plan, [20]. Scrum is a methodology tailored for agile software development, particularly for small teams. Over half of current organizations utilize Scrum, which has demonstrated an overall success rate of 62% in project development, [21]. It consists of sprints, typically lasting from 2 to 4 weeks, during which teams must fulfil specific product functionalities, [22]. Given the widespread adoption of agile, it is essential for students to incorporate these concepts into their projects. In class projects, students generally work within a project team, The academic assumes the role of Scrum Master, while the client serves as the product owner. An increasing number of faculty members are applying the Scrum framework to student projects, [23]. When serving as Scrum Master, an academic facilitates the process and contributes important ideas to PBL, [24]. Incorporating agile methodology into capstone projects is considered an appropriate approach, [25]. Experts suggest that "in many respects, Scrum embodies the learning process itself," indicating that exposure during college is beneficial, [26].

3 Importance and Motivation of Agile Approach in Tertiary Education

As the importance of group work and project management skills for PBL grows, project management and product methodologies emerge as the most well-known approaches to effective product delivery in educational settings.

Project development methodology encompasses all the tasks a project manager needs to undertake, whether it is related to software development, selecting an enterprise system package, or managing the relocation of a department project. Whereas product methodology defines how the details, such as product requirements, product/process analysis, architectural design, testing, and product evaluation, are handled, [27].

There are numerous project methodologies, including waterfall and agile, as illustrated in Appendix in Figure 1 and Figure 2.

The waterfall is a sequential and rigid process that emphasises thorough planning and execution in a specific order. At the same time, the agile approach encourages ongoing collaboration and flexibility, [28]. The waterfall approach is limited in its ability to change customer requirements. The limitations of time, cost, and quality should be broadened to incorporate other elements like project scope and stakeholder satisfaction. Unlike waterfall methodologies, the agile approach offers advantages in ongoing design, adaptability across projects, significant client engagement, and improved coordination within the project team.

Self-organizing teams are defined by their involvement in projects where new roles emerge that might not fit within the conventional employee hierarchy of the organization. Additionally, agile methodologies improve the iterative and incremental approaches provided by object-oriented programming. The agile development team works together towards a shared objective, offering the flexibility to meet both initial and evolving software development requirements.

Agile methods in education empower student-centred learning, by accepting students to explore, experiment, and refine their learning, based on their interests, needs, and goals. They enhance student engagement and motivation, by delivering them with autonomy, mastery, and purpose, and by making learning enjoyable and meaningful.

Agile methodologies in higher education boost student success and well-being by promoting social and emotional skills such as self-awareness, self-regulation, social awareness, interpersonal abilities, and responsible decision-making. They prepare students for the future by equipping them with vital 21st-century skills and competencies, including critical thinking, problem-solving, communication, collaboration, and leadership. Furthermore, they reduce academic workload and stress, by empowering them to work as facilitators, coaches, and mentors, rather than as lecturers, managers, and evaluators, [29].

The key distinctions between the waterfall and agile project methodologies are illustrated in the subsequent Table 1 (Appendix), [30]:

3.1 Motivation of Agile Approach

Agile development is a collaborative approach that focuses on the quick release of functional applications while emphasizing customer satisfaction. It highlights short cycles and regular launches of working software, allowing developers to receive prompt feedback on the true value of their contributions to customers. Agile development improves communication between the development team and the product owner. The agile approach is critical to creating conditions that motivate agile teams and drive performance. Setting clear goals and objectives, inspiring self-organizing teams, cultivating a civilisation of continuous improvement, and embracing transparent communication are all effective strategies for motivating organizational agile teams, [31].

A survey conducted by the Project Management Institute reported that 71% of surveyed organizations adopted agile concepts in the past decade. A recent meta-analysis of empirical studies revealed that agile leadership is significantly positively correlated (with Correlation Strength (β)) with organizational outcomes, [32].

1. Trust (β = 0.93)
2. Organisational performance and effectiveness (β = 0.90)
3. Career success (β = 0.89)
4. Innovation management (β = 0.81)
5. Overall Org. Outcomes (β = 0.49)

In 2018, a research study explored job satisfaction among agile and non-agile teams in relation to the broader IT industry, emphasizing development processes and the factors influencing job satisfaction across various teams, regions, and employees. The findings include the results from a Chi-square significance test that examines the correlation between the amount of experience in different roles (manager, worker, and consultant) with agile methodologies, along with overall job satisfaction illustrated in Table 2 (Appendix). The findings indicate a statistically significant association at the level <.0001.

A 2022 study examined how agile methodologies impact overall project satisfaction in project management. Information was gathered from the IT department of a sizable manufacturing firm in the Czech Republic, which employs over 250 staff members and generates an annual revenue exceeding 50 million euros. The findings revealed that team members utilizing agile methods experienced higher job satisfaction and reduced turnover intentions in contrast to those in non-agile teams. The findings showed a statistically significant positive effect of agile project management on overall project outcomes, as evidenced by a Chi-square test with a p-value of less than 0.001. The authors concluded that agile projects progress more smoothly than others, leading to fewer deviations, [33].

Despite the growing use of agile methods in system and software development and the quicker time to market achieved by agile teams, there are still concerns about its appropriateness for large-scale, complex projects, [34]. A key objective of agile methodologies is to enhance efficiency by alleviating the challenges associated with capturing and updating intricate system specifications. In environments with rapidly changing requirements, comprehensive specification documents would need continual revisions or might even be discarded. In such instances, agile development teams are not obligated to create exhaustive designs before commencing system construction. Instead, they focus on designing and developing small components of the system incrementally. The application of low formality requirements modelling methods within the agile framework has emerged, achieving efficiencies while still adhering to the approach, [35].

Agile transformations not only increase organizational efficiency and product delivery but also foster a collaborative, adaptable, and customer-centric culture. Spotify, ING, Salesforce, Siemens Healthineers, and General Electric's success stories serve as inspiration for other organizations considering or undergoing agile transformations, [36].

Agile methodologies in tertiary education can be employed with several frameworks and approaches, such as Scrum, Kanban, and eXtreme Programming (XP). These frameworks and methods offer tools and techniques for planning, managing, then improving learning processes and outcomes, including boards, cards, sprints, and feedback loops.

The details of these agile methodologies are as follows:

- Scrum - divides organizations into smaller, autonomous teams that manage themselves.
- Kanban - the workflow is represented visually: tasks are divided into small, individual components and recorded on a card that is affixed to a board.
- XP - a structure that emphasises the excellence of the software provided and prioritises engineering concerns.

The fundamental variations between Scrum, Kanban, and XP are shown in the following Table 3 (Appendix).

These and other agile approaches have significantly influenced industrial software development practices. Organizational projects are complex, and today, more than ever, they are fraught with uncertainties and unpredictable outcomes. Consequently, the modification from traditional project development to agile project development aims to address these uncertainties through agile approaches, ensuring that development teams complete projects on time and within budget. By combining flexibility, setting clear goals, breaking down into smaller, manageable tasks, and solving complex algebraic problems, the collaboration, and iterative nature of Agile with the structured approach of modern algebraic methods can create a powerful framework for tackling complex mathematical problems. It can also be a game changer for agile teams by improving recognition of accomplishments, enabling the act of helping others, and enforcing the rules that drive success. Furthermore, these approaches can help minimise the risks associated with complex projects, [37].

3.2 Agile (Scrum) Practices in Educational Settings

Agile methodologies have become increasingly popular in educational environments as a way to lower expenses and enhance adaptability to changing market conditions. This practice is based on incremental development; therefore, it makes evolution easy. The project team and stakeholders emphasize the most critical requirements through ongoing feedback and regular in-person discussions. Tertiary institutions normally require final year or semester students to take industry-based (capstone) units or subjects to complete their program, where students can undertake capstone project units with real business problems that are typically proposed by business/industry clients, [38].

There is considerable empirical evidence indicating that many tertiary institutions heavily depend on traditional methods, such as the waterfall methodology, for software or enterprise system development. This reliance may not always provide students with the essential knowledge and skills required for the job market. Hence, as academics, we must ask ourselves the following questions:

- Are there different, more hands-on methods available to teach students in software engineering and enterprise systems?
- Can we provide students with a comprehensive understanding of the methodological aspects of developing software within a larger team and correct their misconception that Software Engineering and Enterprise Systems primarily involve producing one document after another concurrently?
- What industry trends should inform our decisions on necessary modifications?

In recent years, agile development processes have become increasingly popular, [39]. These methods are designed to facilitate the rapid and early production of functional code by organizing development into small release cycles and emphasizing ongoing communication between developers and clients. By prioritizing functional code as the core of development activities, these approaches have gained popularity among those who view software engineering or enterprise systems as primarily document focused, [40]. As a result, there is increasing support from both students and the industry for integrating agile methodologies into the Software Engineering curriculum and exposing graduate software engineers to agile practices. Initially created to meet the specific demands of software development performed by small to medium-sized teams dealing with unclear and shifting requirements, agile development methodologies have gained traction in academia as an alternative to traditional software development instruction. Given this context, we have chosen to examine Scrum, one of the leading agile methodologies discussed earlier. The Scrum Product Lifecycle Model depicted in Figure 3 (Appendix) illustrates the roadmap for product release.

The best practices of scrum are as follows, [41]:

3.2.1 Promote Peer-To-Peer Collaboration

a. Product owner - communicates their requirements to the Scrum team, manages the Product Backlog, and prioritizes items according to their value to the customer.
b. Development team - multi-talented individuals responsible for the development of the product.
c. Scrum Master - advocates for Scrum principles, supports adaptive problem-solving, and guides team members to work efficiently without sacrificing quality.

3.2.2 Continually Improve Performance in Scrum Events

a. Sprint planning - involves estimating the time required to complete User Story tasks, using Story Points or time increments.
b. b. Sprint - a time-boxed period lasting between 1 and 4 weeks, and once the duration is established it cannot be altered
c. Daily standup (scrum) - a time-limited meeting held at the same time and location each day for everyone involved
d. Sprint review – review and evaluate the work with notes for future sprints.
e. Sprint retrospective - a session to reflect on mistakes and identify areas for improvement.

3.2.3 Develop clear Scrum Artifacts that can be Easily Reviewed and Adjusted as Necessary

a. Product backlog – a single source of tasks or activities that need to be done to develop the product.
b. Sprint backlog - consists of concise User Stories, each clearly broken down into tasks.
c. User story - a short, straightforward description of a product feature or function.
d. Increment - the total of all Product Backlog items finished during the Sprint, combined with the cumulative value of all previously completed increments.

Team members can make design or implementation decisions during the process, in collaboration with the product owner. Instead of larger and less frequent deliveries, work is divided into numerous rapid sprints with user feedback. Each sprint focuses on high-priority requirements and is released to customers right away, [42].

4 Case Study - Industry-based Project Collaboration Model for Effective Delivery of Capstone Project at Federation University

Federation University (Fed) is a regional public university in Australia that requires final year or semester students to take industry-based (capstone) project units in a Bachelor of Information Technology - BIT, Bachelor of Information Technology (Business Information System) – BIT(BIS), Master of Technology (Software Engineering) – MTech(SE) and Master of Technology (Enterprise Systems and Business Analytics) - MTech(EB) courses. BIT & BIT(BIS) included a two-semester and MTech(SE) & MTech(EB) one-semester capstone project units. Fed included the agile (scrum) concepts fully in the project unit(s) curriculum (unit description) for all the above-mentioned programs. Students in these programs study technical subjects such as object-oriented programming, networking, software engineering, business intelligence, project management, data analytics, professional research and communication, and more. These units equip students with the knowledge and skills needed to successfully complete their capstone project. Students utilize their technical skills for the technology aspect, management skills to oversee the project, and professional skills to effectively interact with the client.

Students in those programs have the opportunity to enroll in capstone project units with real business problems that are typically proposed by business/industry clients. As capstone projects are self-managed, students are required to form a small group and work on an IT/BIS/EB project with an industry client and a project supervisor. This teamwork helps students to develop essential communication and collaboration *to* enhance their employability skills. In the Institute of Innovation, Science and Sustainability (IISS) at Fed, both undergraduate and postgraduate IT programs are involved in capstone projects and students are required to form groups to work on those projects proposed by industry clients. However, this process starts when we invite potential industry clients for project briefs/proposals 6-7 weeks prior to the semester commencing. The collected project brief from clients is then sent to a project review panel at the Fed for approval. All proposed projects must address a real ICT requirement for an industry, whether for a commercial or non-profit client. An academic review panel, including the Course Coordinator, will assess the proposed projects to ensure they fulfil the required scope and level of technical challenge. With the Unit Coordinator's approval, students can suggest their own project and/or client, as long as the project meets the stated requirements and they are not the client.

Once we receive approved project briefs from the review panel, we make them available for students to choose from in academic week 1 of each semester. Students self-managed industry-based project collaborative model is shown in Figure 4 (Appendix).

ITECH3208 (Project 1) and ITECH3209 (Project 2) are the two capstone (industry-based) units over two semesters within undergraduate and ITECH7415 (Masters Project) is a single-semester capstone unit (equivalent to 60 credit points) within Information Technology (IT) and Enterprise Systems and Business Analytics (EB) programs at Fed, Australia. Students are required to undertake undergraduate capstone units in the final year and Masters capstone units in the final semester of their program. In these units, students extend their experiences of real business problems with a genuine business client, where they have an opportunity to apply their academic knowledge, and professional and technical skills gained from other units during their programs. In the capstone project students from both software engineering and enterprise systems streams apply agile project and product management methods to design and implement/evaluate unique IT or enterprise solutions (e.g., product, service) as a group in a highly flexible manner, through short, iterative, faster, and interactive sessions/sprints.

Students usually enroll in ITECH3208 followed by ITECH3209 the next semester, staying with the same group of peers. Nonetheless, all industry projects must be approved by a panel of Fed academic staff, including the unit coordinator, to ensure they meet specific project goals, tasks, deliverables, and levels of technical difficulty. Students team up to work on an IT or BIS project with a real business client, under the guidance of a project lecturer and supervisor. They can suggest their own project and/or client with the unit coordinator's approval, as long as the project meets the requirements and the students aren't the clients themselves. The requirements for ITECH3208 and ITECH3209 stream projects are as follows:

- In ITECH3208, students form a team of 4-5 students and choose an approved project. They are required to plan, analyse and design the technical solutions, such as writing specifications, development methodologies, and architecture, and start building the project (a product prototype). All team members are expected to make an equal overall contribution. Each member has to submit individual statements detailing their contributions for every non-invigilated task. If equal contribution is not evident across tasks, students are graded individually, and their marks can differ based on the contributions described in their individual statements and evidence from supervisors and coordinators.

Moreover, students must stay in regular contact with their lecturer, supervisor, and project team. Each team must meet regularly to collaborate and discuss project activities, and should have at least one weekly meeting (minimum 30 minutes) with their supervisor to review progress and provide status reports. Teams should also arrange regular meetings with their client to discuss requirements, planning, and progress.

- ITECH3209 is a continuation of ITECH3208 with the same team of students, who are required to finalise (e.g., develop (coding), build, and implement (such as testing, installation, etc.)) the project and finalise documentation (e.g., test specification, developer manual, user manual, poster, brochure, etc.).
  The Lecturer/Unit Coordinator may allocate any student/s to existing teams, combine students to form new teams or disband current teams if necessary. All members of the team are expected to make an equal overall contribution. In the absence of evidence showing an equal contribution across tasks, students will be graded individually, and their marks may differ based on contributions documented in interviews, supportive evidence, and input from supervisors and coordinators.
  Additionally, teams must regularly collaborate on project activities, with a minimum of one weekly meeting (around 30 minutes) with their supervisor to discuss progress, procedure, and evaluation. Teams should also arrange regular meetings with their client to review requirements, planning, and progress. Students are encouraged to take the initiative in independently developing their skills and knowledge during their project. They should seek to enhance their Skills framework for the information age (SFIA) skills in their chosen specialization area by engaging with industry resources and conducting research to further their knowledge and experience.

ITECH7415 offers students the chance to tackle real-world IT or EB business problems and to participate in research activities. Students form teams of 3-4 to work on an IT/IS project with a business client and project supervisor. A project may involve creating new IT/IS solutions, or reviewing, assessing, or recommending improvements to existing IT/IS solutions to enhance the client's current business processes. In this unit, projects could be stream-specific or multi-stream projects. Single stream project could be specifically for Software Engineering (SE) or Enterprise Systems & Business Analytics (EB). Whereas multi-stream combines projects for both SE and EB students. Students need to research the project domain and examine how similar problems have been addressed in existing literature. However, projects require review and approval by a panel to ensure the project meets the necessary scope and level of technical difficulty. Students must employ advanced decision-making and thorough research techniques, tools, and methodologies to devise and/or implement a business solution for the client's problems. The requirements for SE and EB stream projects are as follows:

- SE projects are required to design and develop a technical solution(s) such as PC/mobile/web applications, digital dashboards, visualization or modeling tools for client's business problems or opportunities, and document processes. Developing a website with a few web pages connected to a database is NOT acceptable for this course.
- EB projects are required to design and evaluate IT/IS/ES related business problems, such as adoption/implementation of new IT/IS systems to improve business processes, Maintaining or upgrading existing systems, business intelligence and data analytics, exploration of new technology, etc. And present project outcomes to the client in the form of artifacts, like recommendations to improve their business information systems with developed products, a comprehensive research report, consultancy report, business model, change management plan, implementation plan, end-user resources, training manuals, and more.

ITECH7415 assessments done by project (six) groups using agile (Scrum) methodology in semester 1, 2024 are shown in the following Table 4 (Appendix).

Agile practice in the Fed is particularly useful, as this product methodology is gaining popularity and is widely used in information technology and enterprise systems areas. It has shown that all six groups submitted ITECH7415 assessments on time, assessed by relevant academic staff (lecturers, and supervisors) & project clients, and delivered quality products on time. Students have been assessed based on 35% individual performance and 65% group performance. Four of the six project groups did very well in functioning as effective agile groups, particularly after the second sprints. Two groups, dominated by some strong personalities, have failed to adopt the best agile practices. Student unit evaluations were mostly positive with no complaints about excessive workload or the best agile development processes. Apart from the two groups mentioned above, it appeared that the stress levels were much lower at the end than beginning of the semester.

Similar assessments and processes were done by students (eight groups) for ITECH3208 and ITECH3209 units in semester 1, 2024. Moreover, the Scrum (agile) approach has guided students' active participation, engagement, and improved understanding within project teams and product development. It also helped students effectively communicate important project factors to stakeholders and enabled them to manage IT and IS projects within the defined scope and schedule. The author is encouraged by the outcome of the initial offering and plans to repeat these units using the same format and structure in semester 1, 2025.

5 Practical Applications in the Modern Engineering Education

The research findings mentioned above can be integrated into modern engineering education, which is mainly focused on practical applications to prepare students for real-world challenges through the following approaches:

- *Hands-on learning* through computer labs, workshops, and project-based learning to help students apply theoretical knowledge to practical problems.
- *Interdisciplinary Approach* integrates knowledge from fields like management, technology, and social sciences. This helps students develop a diverse skill set and address complex issues from multiple perspectives, [43].
- *Industry collaboration* through partnerships with industries that allow students to work on real-world projects to provide valuable insights into current industry practices and technologies, [44].
- *Emerging technologies* like AI, robotics, and renewable energy are integrated with Engineering units to equip graduates with the latest skills and knowledge needed for the dynamic job market, [45].
- *Global perspective* encourages international collaborations and exchanges helping students

understand global engineering challenges and their solutions.

- *Student-centric learning* focuses on the needs and interests of students by providing personalised learning paths and the use of digital tools to enhance learning experiences, [46].

6 Conclusions and Future Work

In this paper, we have explored the impact of agile practices on student projects in software engineering and enterprise systems, responding to the growing industry demand for graduates skilled in agile development methodologies. A case study was conducted at **Fed units** where an agile teaching approach was implemented to educate capstone project students. The goal of the agile approach is to close the gap between academic education and industry practices by involving students in agile development methods. This study assessed Scrum as a method for agile software and enterprise system development. The findings suggest that students working together with effective communication and interaction can achieve higher levels of performance compared to working independently. It has also led to positive outcomes, including better student performance, increased engagement and motivation, and a reduced academic workload and stress. However, the underlying educational goals must be carefully considered before incorporating agile methodologies into an engineering (tertiary) educational setting.

It is concluded that incorporating agile practices in engineering education would positively impact learning and teaching processes. Moreover, implementing the agile approach in undergraduate and postgraduate engineering education could enhance communication and relationships among students, foster active team participation, and increase motivation for current and future learning. It is also likely to expedite software development, enhance application quality, and manage changing requirements in engineering education.

The study's findings can be incorporated into modern engineering education to ensure students are ready for real-world challenges through student empowerment, industry collaboration, hands-on learning, emerging technologies specifically agile approaches and global perspectives. These trends could cultivate a new generation of engineers who are not only technically skilled but also adaptable and equipped to tackle the complex challenges of the 21st century.

Despite our positive experiences with agile student projects at Fed, it is crucial to remember that students are the most significant factor in determining success. In a tertiary educational setting, we can argue that students who are less knowledgeable and disciplined may find it more challenging to grasp agile principles compared to those exposed to more traditional methodologies, such as the waterfall model. Therefore, we must consider whether students would gain the same value from agile practices if they had not previously encountered traditional methodologies.

This is certainly a field that warrants further research, particularly if we consider moving from traditional methodologies to agile practices in future engineering education. Additionally, further research is essential to establish clear principles and guidelines for academics when implementing agile practices in tertiary engineering education.

References:

[1] Guo, P., Saab, N., Post, L., and Admiraal, W. (2020), A review of project-based learning in higher education: Student outcomes and measures, *International Journal of Educational Research*, Vol. 102, 101586. 2020, https://doi.org/10.1016/j.ijer.2020.101586.

[2] Schwering, R. E. (2015). Optimizing learning in project-based capstone courses. *Academy of Educational Leadership Journal*, 19(1), 90-104. Corpus ID: 142792450.

[3] Carlsson, S. A., Hedman, J., & Steen, O. (2010). Integrated curriculum for a bachelor of science in business information systems design (BISD 2010). *Communications of the Association for Information Systems*, 26(24), 525-546. http://dx.doi.org/10.17705/1CAIS.02624.

[4] Bell, S. (2010). Project-Based Learning for the 21st Century: Skills for the Future. *Clearing House*, 83(2), 39-43. https://doi.org/10.1080/00098650903505415.

[5] Krajcik, J., & Blumenfeld, P. (2006). 19. Project-Based Learning. In *The Cambridge Handbook of the Learning Sciences* (pp. 317-333). New York: Cambridge University Press. https://doi.org/10.1017/CBO9780511816833.020.

[6] Ku, H. & Goh, S., (2010), Final year engineering projects in Australia and Europe, *European Journal of Engineering Education*, Vol. 35, No. 2, May 2010, pp. 161-173,

http://dx.doi.org/10.1080/03043790903497336.

[7] Healey, M., Lannin, L., Stibbe, A., & Derounian, J. (2013). Developing and enhancing undergraduate final year projects and dissertations. York, UK: HE Academy, [Online]. https://www.advance-he.ac.uk/knowledge-hub/developing-and-enhancing-undergraduate-final-year-projects-and-dissertations (Accessed Date: October 20, 2024).

[8] Durlak, J. A., & DuPre, E. P. (2008). Implementation Matters: A Review of Research on the Influence of Implementation on Program Outcomes and the Factors Affecting Implementation. *American Journal of Community Psychology*, 41, 237-350. http://dx.doi.org/10.1007/s10464-008-9165-0.

[9] Darling-Hammond, L. (2006). *Powerful Teacher Education: Lessons from Exemplary Programs.* San Francisco: John Wiley and Sons, Inc. ISBN: 9780787972738 (hbk.). ISBN: 978-0787972738.

[10] Vijayalakshmi, M.; Desai, P.D.; Joshi, G.H., (2013), "Outcome based education performance evaluation of capstone project using assessment rubrics and matrix", *IEEE International Conference on MOOC Innovation and Technology in Education (MITE)*, 20-22 December 2013, Jaipur, India, pp 6 - 10, http://dx.doi.org/10.1109/MITE.2013.6756295.

[11] Magana, A. J., Seah, Y. Y., & Thomas, P. (2018). Fostering Cooperative Learning with Scrum in a Semi-Capstone Systems Analysis and Design Course. *Journal of Information Systems Education*, 29(2), 75-91, [Online]. http://jise.org/Volume29/n2/JISEv29n2p75.html (Accessed Date: October 20, 2024).

[12] Gaurab, K. (2023). Agile in Education: A New Way of Teaching and Learning for the 21st Century, [Online]. https://www.linkedin.com/pulse/agile-education-new-way-teaching-learning-21st-kumar-gaurab-bzeoc (Accessed Date: October 22, 2024).

[13] *Cuseo, J. B. (1998). "Objectives and Benefits of Senior Year Programs."* In The Senior Year Experience: Facilitating Integration, Reflection, Closure, and Transition, edited by John N. Gardner and Gretchen B. Van der Veer, 21–36. San Francisco: Jossey-Bass Publishers. ISBN: ISBN-0-7879-0927-0.

[14] Wallace, S. E. (2015). Team-based learning in a capstone course in speech-language pathology: Learning outcomes and student perceptions. *Communication Disorders Quarterly*, 37(1), 44-52. https://doi.org/10.1177/1525740114558462

[15] Taipalus, T., Seppanen, V., & Pirhonen, M. (2018). Coping with Uncertainty in an Agile Systems Development Course. *Journal of Information Systems Education*, 29(2), 117-126, [Online]. https://jise.org/volume29/n2/jisev29n2p117.html (Accessed Date: October 22, 2024).

[16] Marti, E., Gil, D., & Julia, C. (2006). A PBL experience in the teaching of computer graphics. *Computer Graphics Forum*, 25(1), 95-103. https://doi.org/10.1111/j.1467-8659.2006.00920.x.

[17] Genc, M. (2015). The project-based learning approach in environmental education. *International Research in Geographical & Environmental Education*, 24(2), 105-117. https://doi.org/10.1080/10382046.2014.993169.

[18] Danford, G. L. (2006). Project-based Learning and International Business Education. *Journal of Teaching in International Business*, 18(1), 7-25. doi: 10.1300/J066v18n01_02.

[19] Ozdamli, F., & Turan, B. (2017). Effects of a technology supported project based learning (TS-PBL) approach on the success of a mobile application development course and the students' opinions. *TEM Journal*, 6(2), 258- 264. doi: 10.18421/TEM62-10.

[20] Agile Alliance. (2018). The Agile manifesto, [Online]. https://www.agilealliance.org/wp-content/uploads/2019/09/agile-manifesto-download-2019.pdf (Accessed Date: October 22, 2024).

[21] Rising, L., & Janoff, N. S. (2000). The scrum software development process for small teams. *IEEE Software*, 17(4), 26-32. http://dx.doi.org/10.1109/52.854065.

[22] Baird, A., & Riggins, F. J. (2012). Planning and sprinting: Use of a hybrid project management methodology within a CIS capstone course. *Journal of Information Systems Education*, 23(3), 243-257, [Online]. https://aisel.aisnet.org/jise/vol23/iss3/1/ (Accessed Date: October 22, 2024).

[23] May, J., Yoir, J., & Lending, D. (2016). Play ball: Bringing scrum into the classroom. *Journal of Information Systems Education*, 27(2), 87-92, [Online].

https://aisel.aisnet.org/jise/vol27/iss2/1/ (Accessed Date: October 22, 2024).

[24] Guthrie, C. (2010). Towards greater learner control: Web supported project-based learning. *Journal of Information Systems Education,* 21(1), 121-131, [Online]. https://jise.org/Volume21/n1/JISEv21n1p121.pdf (Accessed Date: October 22, 2024).

[25] Sharp, J. H., Mitchell, A., & Lang, G. (2020). Agile Teaching and Learning in Information Systems Education: An Analysis and Categorization of Literature. *Journal of Information Systems Education,* 31(4), 269-281, [Online]. https://www.researchgate.net/publication/347034760_Agile_Teaching_and_Learning_in_Information_Systems_Education_An_Analysis_and_Categorization_of_Literature (Accessed Date: October 22, 2024).

[26] Echols, M. E. (2016). What agile/scrum Can Teach Learning. *Chief Learning Officer*, 15(8), 10-10, [Online]. https://www.chieflearningofficer.com/2016/08/08/what-agilescrum-can-teach-learning/ (Accessed Date: October 22, 2024).

[27] Litten, D. (2022), Methodologies Senior Project Managers Need, [Online]. https://www.projex.com/methodologies-senior-project-managers-need/ (Accessed Date: September 29, 2024).

[28] Bergmann, T., & Karwowski, W. (2019). Agile Project Management and Project Success: A Literature Review. In J. I. Kantola, S. Nazir, & T. Barath (Ed.), Advances in Human Factors, Business Management and Society (Roč. 783, s. 405–414). Springer International Publishing. https://doi.org/10.1007/978-3-319-94709-9_39.

[29] Schwaber, K. (1997). SCRUM Development Process. In J. Sutherland, C. Casanave, J. Miller, P. Patel, & G. Hollowell (Ed.), Business Object Design and Implementation (s. 117–134). Springer London. https://doi.org/10.1007/978-1-4471-0947-1_11.

[30] Lockhart L. (2023), Agile vs. waterfall: 10 key differences between the two methods, [Online]. https://www.float.com/resources/agile-vs-waterfall (Accessed Date: September 24, 2024).

[31] Illarionova, N (2024), Motivation Strategies Through Agile Practices: Proven Methods, [Online]. https://albimarketing.com/blog/motivation-strategies-through-agile-practices-proven-methods/ (Accessed Date: September 28, 2024).

[32] Porkodi, S. (2024), The effectiveness of agile leadership in practice: A comprehensive meta-analysis of empirical studies on organizational outcomes, *Journal of Entrepreneurship, Management and Innovation (Jemi)* 20(2):117-138, http://dx.doi.org/10.7341/20242026.

[33] Melnik, G., and Maurer, F. (2006), Comparative Analysis of Job Satisfaction in Agile and Non-Agile Software Development Teams, Department of Computer Science, 7th International Conference, XP 2006, Oulu, Finland. June 17-22, 2006, Proceedings. http://dx.doi.org/10.1007/11774129_4.

[34] Wanner R. (2024), *Why Agile Project Management is Not Suitable for Most Projects,* [Online]. https://rolandwanner.com/why-agile-project-management-is-not-suitable-for-most-projects/ (Accessed Date: September 29, 2024).

[35] AltexSoft (2023), *Agile Project Management: Best Practices and Methodologies*, [Online]. https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/ (Accessed Date: September 29, 2024).

[36] Muhammad, P. (2024), *Driving Success: Case Studies on the Positive Impact of Agile Transformation in Organizations,* [Online]. https://www.linkedin.com/pulse/driving-success-case-studies-positive-impact-agile-muhammad--eiswc (Accessed Date: September 29, 2024).

[37] Kiiskinen, A. (2023), *Rewarding motivation: How to adopt agile methods successfully*, [Online]. https://www.eficode.com/blog/rewarding-motivation-how-to-adopt-agile-methods-successfully (Accessed Date: September 25, 2024).

[38] Mustaqim, I., Raharjo, T., and Trisnawaty, N., (2023), Toward Unveiling the Execution of Large-Scale Agile Project in Global Software Development: A Systematic Literature Review, *2023 International Conference on Engineering and Emerging Technologies (ICEET)*, 27-28 October 2023, Istanbul, Turkiye. http://dx.doi.org/10.1109/ICEET60227.2023.10526017.

[39] Sharp, J. H., & Lang, G. (2018). Agile in teaching and learning: Conceptual framework and research agenda. *Journal of Information Systems Education,* 29(2), 45-52, [Online]. https://aisel.aisnet.org/jise/vol29/iss2/1/ (Accessed Date: September 25, 2024).

[40] Palopak, Y., and Huang, S. (2024), Perceived Impact of Agile Principles: Insights from a Survey-Based Study on Agile Software Development Project Success, *Information and Software Technology*, 176(5):107552. https://doi.org/10.1016/j.infsof.2024.107552.

[41] Lynn R. (2024), Scrum Best Practices for Teams, [Online]. https://www.planview.com/resources/guide/what-is-scrum/scrum-best-practices-teams/ (Accessed Date: September 28, 2024).

[42] Mahnic, V. (2012). A Capstone Course on Agile Software Development Using Scrum. *IEEE Transactions on Education,* 55(1), 99–106. http://dx.doi.org/10.1109/TE.2011.2142311.

[43] educatingengineers.com (2024), What Is Applied Engineering?, [Online]. https://educatingengineers.com/blog/what-is-applied-engineering/ (Accessed Date: September 30, 2024).

[44] Ivanov, V., Pavlenko, I., Evtuhov, A., Trojanowska, J. (2024). Major Challenges in Engineering Education. In: Augmented Reality for Engineering Graphics. Springer Tracts in Mechanical Engineering. Springer, Cham. https://doi.org/10.1007/978-3-031-44641-2_1.

[45] Yan, L., Shi, Y., Wang, J. (2023). Reform and Practice of Traditional Engineering Courses in Application Oriented Local Universities Against the Background of Emerging Engineering Education. In: Patnaik, S., Paas, F. (eds) Recent Trends in Educational Technology and Administration. EduTA 2022. Learning and Analytics in Intelligent Systems, vol 31. Springer, Cham. https://doi.org/10.1007/978-3-031-29016-9_12.

[46] MIT News, Reimagining and rethinking engineering education, [Online]. https://news.mit.edu/2018/reimagining-and-rethinking-engineering-education-0327 (Accessed Date: October 23, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Belal Chowdhury conceptualised the research framework for tertiary education, conducted literature review, identified key agile methodologies, conducted the case study, drafted most of the manuscripts and coordinated revisions.
- Nasreen Sultana assisted in the literature review, provided insight on relevant studies, reviewed and provided feedback on manuscripts drafts.
- Sarah Chowdhury facilitated connections with industry partners and coordinated the involvement of industry professionals in the research.
- Sumyia Chowdhury provided guidance on the academic rigor, assisted in refining the research objectives, contributed to the final review and editing of the manuscript.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The authors did not receive any funding for conducting this research.

Conflict of Interest

The authors have no conflicts of interest to declare.

APPENDIX

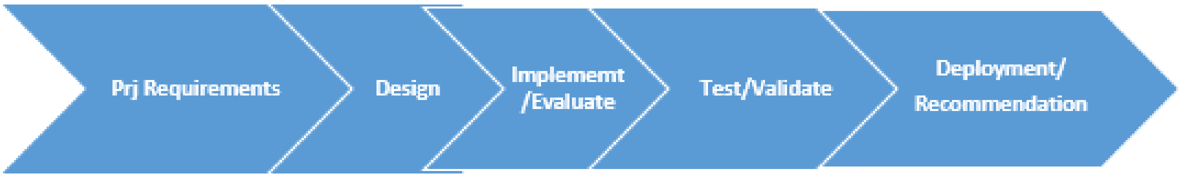

Fig. 1: Waterfall Product Lifecycle Model

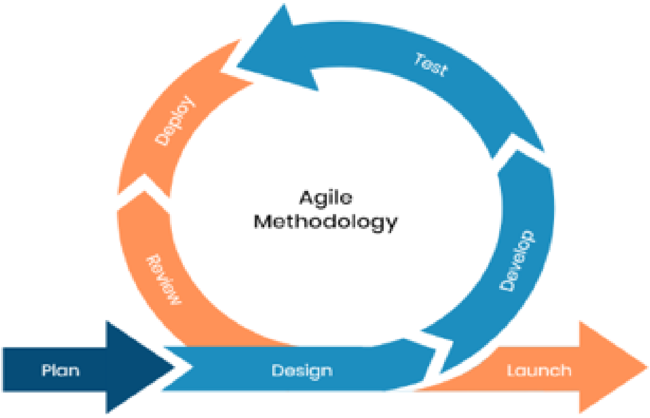

Fig. 2: Agile Product Lifecycle Model

Fig. 3: Scrum Product Lifecycle Model

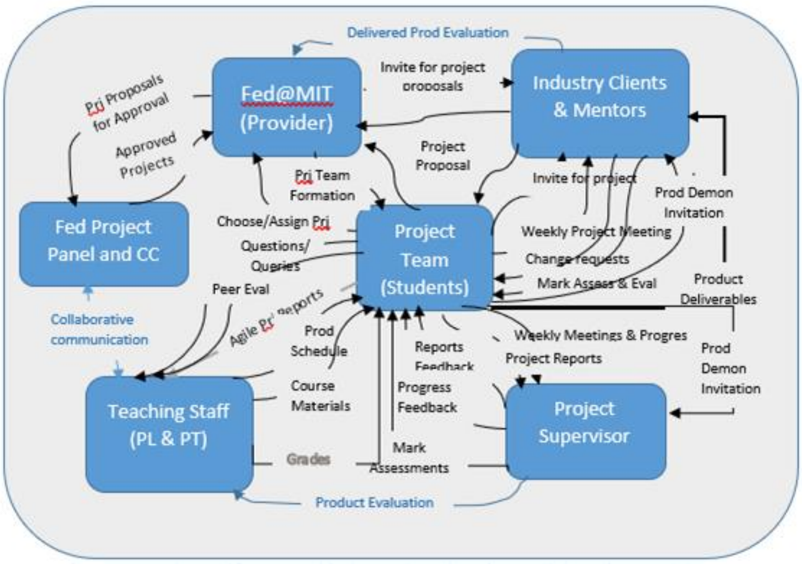

Fig. 4: Student self-managed industry-based project collaboration model

Table 1. Differences between the waterfall and agile methods

Aspect	Waterfall	Agile
Communication	More formal with detailed communication plans and progress reports shared across multiple stakeholders.	Informal communication, with frequent interactions between individuals or small groups of stakeholders.
Roles	Strictly assigns roles to project team members.	Empowers self-organising team to collaborate on different aspects of the project.
Planning	Essentially making a good plan at the beginning of the project and sticking to it as a pre-planned process.	A continuous process throughout the project's life cycle with adjustments made as the situation demands.
Scope	Generally, discourages changes to the project's scope.	More adaptable to changes in scope, as requirements change.
Speed	Take longer as all requirements must be agreed upon before beginning of the development.	Usually delivered more rapidly due to the iterative development cycles.
Time frames	Long-term projects with predetermined timelines.	Uses short iterations (sprints) to deliver value rapidly.
Delivery	Requires the completion of all tasks before any work can be released.	Quick delivery of projects with shorter lifecycles.
Flexibility	Less flexible and resistant to change once the project's scope has been defined.	More flexible, encourages teams for responding and adapting changes quickly during the development process.
Documentation	Relies heavily on documentation to ensure that all team members are on the same page.	Relies on minimal documentation, focusing on self-organizing teams and collaboration.
Testing	Essential, usually done at specific milestones, often towards the end of the project.	Essential, emphasises incremental testing to identify and resolve issues throughout the development process.

Table 2. Tests for general job satisfaction among employees in agile teams, non-agile teams, and the broader IT industry

	Non-agile	**General IT**
Agile	$\chi^2 = 80.96$ N=447 $p<0.0001$ null hypothesis rejected ✓ $rho_s = 0.39$ moderate positive association	$\chi^2 = 95.63$ N=1,252 $p<0.0001$ null hypothesis rejected ✓ $rho_s = 0.26$ moderate positive association
Non-agile	—	$\chi^2 = 17.15$ N=1,067 $p= 0.0018$ null hypothesis not rejected ✗ $rho_s = 0.05$ no association

Table 3. Differences between Scrum, Kanban, and XP

Aspect	Scrum	Kanban	XP
Goal	Use of cross-functional, self-organised, empowered teams, and divide their work into Sprints	To alleviate impediments that cause teams to take longer to deliver	To organise people to produce higher-quality software more productively.
Current state	Used for small projects. However, it can easily be scaled for large projects	Project values and goals are communicated down to developers or other contributors.	Very effective for the small development team of 5 or less people.
Speciality	Ensures transparency in communication and environment of collective accountability and continuous progress.	Workflow is visualised: work is broken down into small, discrete items and written on a card. Number of items that can be in progress at any one time is strictly limited.	Empowers development teams to confidently respond to changing customer requirements, even late in the life cycle. Fully emphasis on collaborative team and customer satisfaction.
Values	- Focus - Courage - Openness - Commitment - Respect	- Transparency - Agreement - Balance - Respect - Understanding - Leadership - Collaboration - Customer focus - Flow	- Communication - Simplicity - Feedback - Courage - Respect
Key metrics	Sprint Velocity (2 weeks)	Cycle time	Iteration time (2 weeks)
Change philosophy	Not make changes to the sprint forecast during the sprint.	Change can happen at any time.	A high degree of developer discipline along with continuous customer involvement for the duration of the project.
Cadence	Regular fixed length sprints.	Continuous flow.	Iteration.
Release methodology	At the end of each sprint if approved by the product owner.	Continuous delivery or at the team's discretion.	At the end of iteration.

Table 4. Unit (ITECH7415) Assessments

Assessment Name and Mark	**Requirements**	**Assessed By**
Project Storyboard (Group) – 5%	Outline the project problems and objectives. Clarify Client's expectations. Outline Client's preferred way of working with the students. Propose a few key 'user stories' in conjunction with the client and outline some solution concepts for client review.	Lecturer, Project Client and Supervisor
Weekly Stand-ups (Individual) – 10%	Weekly 3 minutes de-brief during workshops to include answers to the following questions: i) What did you accomplish since the last meeting? ii) What are you working on until the next meeting? iii) What is getting in your way or keeping you from doing your job?	Lecturer/Tutor
Product Roadmap (Group) – 5%	Includes: i) The Product Vision: a quick summary, to communicate how your product supports/addresses your client's goals. ii) Product Backlog: The full list of what is in the scope for your project, ordered by priority. iii) Revised User Stories: a clear identification of the project requirements and objectives. Acceptance criteria for user stories should also be included. iv) Product Schedule: goals, milestones, and deliverables.	Lecturer, Project Client and Supervisor
Sprint Retrospectives – Individual video presentation – 25%	At the end of each sprint, each project team will conduct a Sprint Review meeting, after which every student is expected to submit a retrospective in their individual Mahara ePortfolio page created for the project. The retrospective should be a video recording (no more than 5 minutes) embedded as a media file in the portfolio. Students are welcome to attach documents to support retrospectives. Reflections are expected to cover both team and individual components of the project. Individual Component include personal tasks and learning during the Sprint and should provide answers to the following questions: i) What did I accomplish in the past sprint? ii) How did I apply stream specific knowledge to my project? iii) What did I learn from in the past sprint? iv) What could have gone better in the sprint?	Lecturer/Tutor
Product Demonstration (Group) – 15%	Include PowerPoint slides, Posters, and /or Practical Demonstrations. Each team will be given up to 20 minutes for their presentation and there will be an additional 5-10 minutes for a question/answer session. Supervisors and Clients will be invited to attend and assess presentations. Students are to make 2-minutes promo videos to demonstrate their projects in digital format. Demos will capture the following i) What product was developed, ii) How it addresses the client's problem, iii) What tools/technologies were used, iv) What type of research/analysis activities were conducted, and the benefits gained.	Lecturer, Project Client and Supervisor
Client/Supervisor Evaluation (Group) – 20%	Project Storyboard, Product Roadmap, Sprint Reviews, and Product Demonstration will be assessed by both client and supervisor.	Project Client and Supervisor
Project Write-up (Group) – 20%	This document should cover ALL aspects of the project and include solutions to the project problem. The recommended length of the report is between 7000 - 10,000 words, excluding appendices. The project Write-up must include the problem statement, project objectives, related academic and commercial research, project approach, discussion of product developed, and limitations. The Write-up is to be accompanied by a project reflection that reports individual contribution to the project. Individuals will be awarded marks for the Project Write-up based on the quality of their reflection. Reflections should include Justification of contributions as per skillsets and time allocated to projects, - Description of the largest 3 contributions with evidence, - Unique things learned/contributed, and - Ideas to improve the course.	Lecturer