

# Agile Practices in Teaching and Learning in Teamwork: A Conceptual Framework

Dipendra Ghimire <sup>a</sup>, Alaa Aljanaby <sup>b</sup>,

<sup>a</sup> *Faculty of IT, Aspire2 International, New Zealand*  
dipendra.ghimire@aspire2.ac.nz

<sup>b</sup> *Faculty of IT, Aspire2 International, New Zealand*  
alaa.aljanaby@aspire2.ac.nz

---

## Abstract

Agile methods are based on an iterative and incremental cycle of development where solutions and requirements are developed through collaboration and coordination between cross-functional teams and their customer. Agile method was used in software development when introduced. However, Agile has been used in other areas such as project management, education and other sectors where there is teamwork. This paper analyses the different agile methods and presents the conceptual application of some of the agile practices to address the challenges faced by students with their work in group projects.

**Keywords:** Agile practices, Teamwork, Group projects

## Introduction

The Agile concept values collaboration and responds to the change to produce quality results. Educators are constantly subjected to change led by new and emerging techniques, technologies and ways of delivering the course. Recently, many of the suggestions for improvement have come from the research community who have conducted a research on the impact of Agile on teaching and learning (Vogler et al., 2018). Such research has a huge impact on how Agile has been used in teaching and learning. However, there are many Agile practices, but little is known about how these practices can be implemented practically and what their effects are. This study evaluates and presents different Agile practices that can be used in teamwork. We believe this study will be important for students, tutors and education providers who want to bring Agile practices in their students' projects to minimise the challenges that occurs during the completion of the project. The rest of the sections of the paper present the background of Agile methods used in software development, literature review, use of Agile in education and conclusion.

## Agile software development methods

Agile software development methods consists of a set of practices for software development. Agile processes address the challenge of an unpredicted world by relying on people and their creativity

rather than processes. In 2001, seven practitioners created the Agile manifesto to define the approach now known as Agile Software Development. The Agile manifesto defines four main values and twelve principles. The values are (Fowler & Highsmith, 2001):

- Individual and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following plans

A range of software development approaches have had the Agile descriptor applied; these methods are shown in Table 1.

**Table 1:** Agile Software Development Approaches

Name	Descriptions
Extreme programming (XP) (Beck, 2000)	Focus on the best practices for development: Practices includes, the planning game, small release, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, on-site customers and coding cycles.
Scrum (Schwaber, 1997)	Focuses on the project management where it is difficult to plan ahead. Teams are <ul style="list-style-type: none"> <li>• Self-organizing</li> <li>• Cross-functional; have all the skills necessary to create product increments</li> <li>• Team members may have specialized skills and a particular area of focus</li> </ul>
Feature-driven development (FDD) (Palmer & Felsing, 2001)	A feature list is created that groups the features into sets and subject areas. After completing the grouping process a development plan is created, features are designed and then built (Boehm & Turner, 2003a).
Lean	An adoption of lean principles in production. Consists of principle: eliminate waste, amplify learning, decide as late as possible, empower the team and build integrity.
Dynamic systems development method (DSDM) (Stapleton, 1997)	Focuses on: User involvement, empowering the team, frequent delivery, addressing current business needs, iteration and incremental development and effective and efficient communication.

### Literature review

Scrum is an iterative and incremental Agile development framework used in the development of software (Schwaber & Beedle, 2002). Scrum can be observed as a lightweight management approach rather than a full process; hence, it has gained increasing popularity for use in software development projects and has a broad applicability for managing and controlling iterative and incremental projects (Kniberg, 2010; Stankovic et al., 2013a). The Scrum process is suitable for projects that have tight timelines, changing requirements, and are critical to a business because, in Scrum, the work is divided into packets with continuous testing and documentation as the project continues (Beck et al., 2001b). The packets, known as sprints, are derived from the backlog of existing requirements. A backlog is an ordered list of the work the development team must address during the next sprint (Rising & Janoff, 2000). The sprint is a time-box (time interval) of one

month or shorter, during which, a useable and, potentially releasable product increment is created (Schwaber & Sutherland, 2013). In Scrum, very short meetings, known as stand-ups, are conducted every day and the deliverable of the sprint is sent to the customer within the allocated time-box (Schwaber & Sutherland, 2013). The process followed in this approach can be viewed in Figure 1.

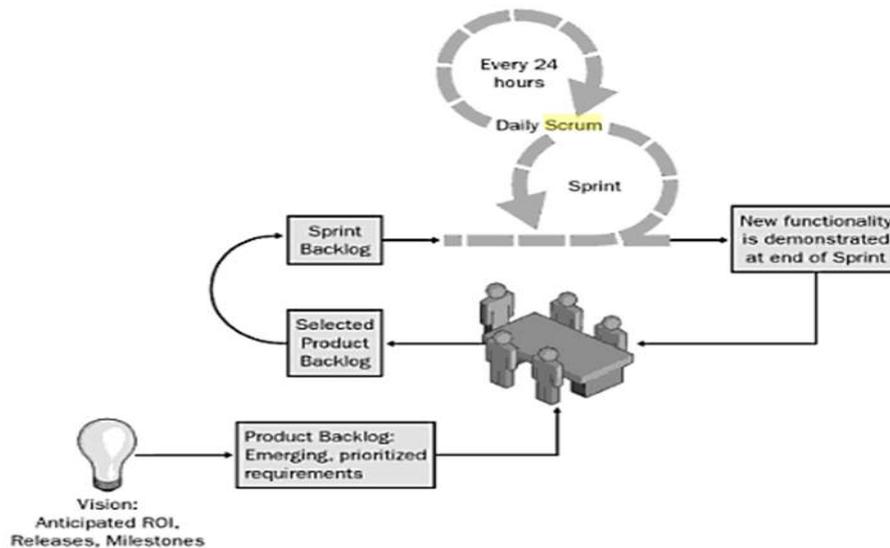


Figure 1: The Scrum process (Schwaber, 2004)

The Scrum team usually consists of the product owner, developers, testers and a Scrum master (Schwaber & Sutherland, 2013). Roles associated with a Scrum team are summarised in Table 2.

**Table 2:** Roles associated with a Scrum team

<b>Product owner</b>	<ul style="list-style-type: none"> <li>• Creates a prioritized wish list called a product backlog</li> <li>• Ensures the product backlog is visible and clear to all showing what the next work for Scrum team is (Schwaber &amp; Sutherland, 2013).</li> </ul>
<b>Scrum master</b>	<ul style="list-style-type: none"> <li>• The Scrum master is responsible for ensuring Scrum is understood and followed. This is done by ensuring that the team follows Scrum theory, practices and rules.</li> <li>• Helps in the interaction with the Scrum team to maximize the value created by the team (Schwaber &amp; Sutherland, 2013).</li> </ul>
<b>Development teams</b>	<ul style="list-style-type: none"> <li>• Self-organising</li> <li>• Cross-functional; have all the skills necessary to create product increments</li> <li>• Team members may have specialized skills and a particular area of focus, but the credit belongs to team as a whole (Schwaber &amp; Sutherland, 2013).</li> </ul>

### **Agile in education**

Group projects are a common feature of the education trend currently, although they hold their own particular challenges. Literature suggests that working in a group has an impact on the achievement of the projects. Teamwork brings together a larger range of skills and knowledge. A study done by Vogler et al (2018) mentioned that having a wide range of opinions will make sure the projects are considered from all sides and increase critical thinking skills. Such group projects will help to develop communication and collaboration skills. This helps students self-reflect on how to work in a team.

While teamwork in projects have positive impacts, literature suggest there exists some challenges. Some of the challenges are as follows (Feliciano, Storey, & Zagalsky, 2016; Jackson, 2008; Klassen & Usher, 2010; Popov et al., 2012).

- Conflict between members
- Difficulty sticking to deadlines set by the team
- Lack of planning to complete the project
- Unclear roles in the group
- Limited communication within the team
- Dominant members of the team
- Not enough discussion on topics and roles within the group
- Unbalanced work distribution with some people doing more than others
- Lack of delegation of tasks

This study explores the application of some of the Agile practices that are used in software development in other projects in the education sectors. Following practice used in Agile software development is proposed to address some of the challenges that occur in team projects.

### **Stand-up meeting**

Communication refers to the sending and receiving of information (Cockburn & Highsmith, 2001b). Communication can be achieved using different pathways such as face-to-face and email (Biehl et al., 2007). Communication between the team members occur during a project. One of the challenges student face during their group project is related to communication where team members do not share information on any difficulties with the projects. A stand-up in Agile software development is a daily meeting that involves the core team: product owners, developers, and the scrum master to answer questions such as:

What did I work on yesterday? What am I working on today? What issues are blocking me? Introducing such a practice in student group work and project will allow other team members to be aware of any issue occurring with their work. Such practices may prevent miscommunication.

### Backlog user stories

The product backlog in Agile software is the list of all the work that needs to be done. It usually contains user stories, bugs, technical tasks, and knowledge acquisition (Sedano, Ralph, & Péraire, 2019). The backlog is periodically refined by the product owner and scrum team to ensure the work carried out in each sprint is always defined and prioritised. Similarly, student could use these techniques to prioritise tasks on the project. This will help students to track the task, give them an opportunity to make any changes required.

### Burndown chart

A burndown chart is a graphical representation of work left to do versus time estimated for the project. This chart is commonly used in Agile software development process (Beck et al., 2001). The backlog is often on the vertical axis, with time along the horizontal. It is useful for estimating when all of the work will be completed. Students working in a team for their group project may use such chart to track the progress of their work.

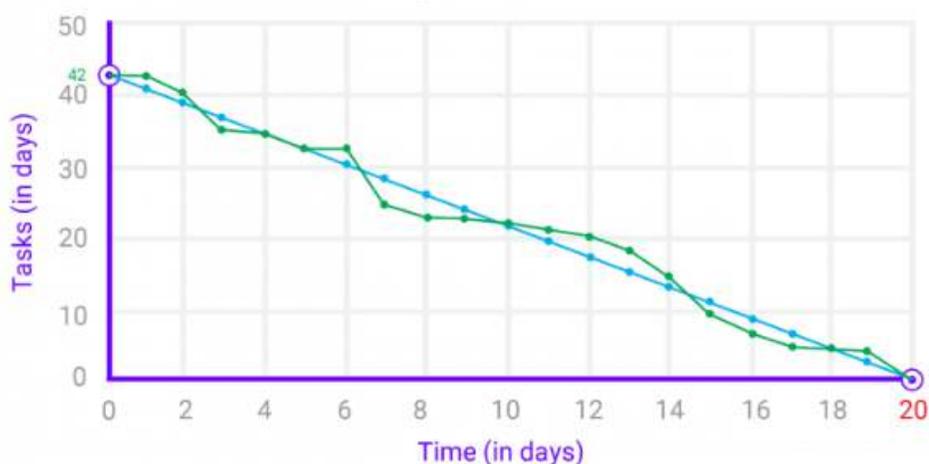


Figure 1: Burndown Chart (<https://elearningindustry.com/>)

### Scrum board

A scrum board is a tool that helps to make Sprint Backlog items visible in Agile software development process. The board is updated by the team and shows all items that need to be completed for the current Sprint (Sedano et al., 2019). Such practices could help students make their work visible to all the members. This practice could allow students get the work distributed among team members more clearly.

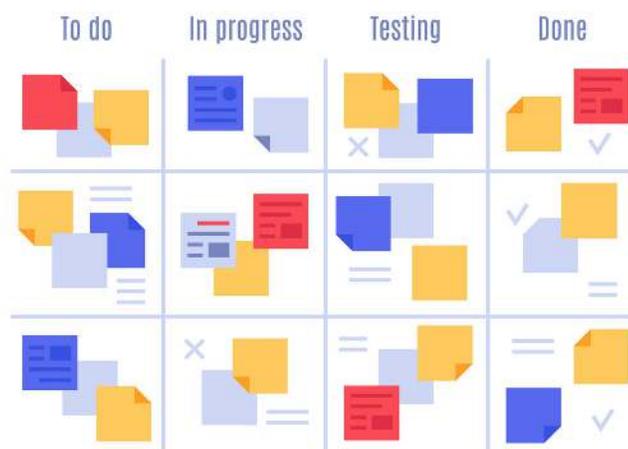


Figure 2 Scrum Board (<http://www.asianefficiency.com/>)

### Retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to reflect on their work and create a plan for improvements to be made during the next Sprint. Sprint Retrospective occurs after the Sprint Review and before the next Sprint Planning (Jovanovic, Mesquida, & Mas, 2015). The Scrum Master facilitates the meeting. This is the opportunity for the Scrum Team to improve and all members should attend. During Sprint Retrospective, the team discusses: What went well in the Sprint? What could be improved? What will we commit to improve in the next Sprint? These practices could be applied in students work where student can reflect on their work and minimise the challenge in the upcoming project work. Purposed Mapping of Agile practices with the challenges faced by the students are shown in Table 3.

**Table 1:** Purposed Mapping of Agile Practices with Challenges in Projects

Stand-up meeting	<ul style="list-style-type: none"> <li>• Limited communication</li> <li>• Conflict between members</li> </ul>
Backlog User Stories	<ul style="list-style-type: none"> <li>• Difficulty sticking to deadlines</li> <li>• Lack of planning</li> <li>• Unclear roles in the group</li> </ul>
Burndown chart/ Burnup	<ul style="list-style-type: none"> <li>• Difficulty sticking to deadlines</li> <li>• Lack of planning</li> </ul>
Scrum board	<ul style="list-style-type: none"> <li>• Difficulty sticking to deadlines</li> <li>• Lack of planning</li> <li>• Unclear roles in the group</li> <li>• Lack of delegation of tasks</li> </ul>
Retrospective	<ul style="list-style-type: none"> <li>• Not enough discussion on topics and roles within the group</li> </ul>

	<ul style="list-style-type: none"><li>• Unbalanced work distribution with some people doing more than others</li><li>• Lack of delegation of tasks</li></ul>
--	--

Source: Author's own Table

## Conclusion

This study has presented the application of some of the practices that could be used in Students group work. Such practises help the educational paradigm to evolve from 'traditional' education to 'Agile'. This study reviewed the current challenges in group projects and have shown how the application of Agile practices such as stand-up, backlog user stories, scrum board, retrospective to minimise the challenges students face during group projects. Incorporating the practices used in agile software development in teamwork in different disciplines may improve the teaching and learning process.

## REFERENCES

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Jeffries, R. (2001). The agile manifesto. In: Feb.
- Feliciano, J., Storey, M.-A., & Zagalsky, A. (2016). *Student experiences using GitHub in software engineering courses: a case study*. Paper presented at the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C).
- Jackson, E. (2008). The development of work discussion groups in educational settings. *Journal of Child Psychotherapy*, 34(1), 62-82.
- Jovanovic, M., Mesquida, A.-L., & Mas, A. (2015). *Process improvement with retrospective gaming in agile software development*. Paper presented at the European Conference on Software Process Improvement.
- Klassen, R. M., & Usher, E. L. (2010). Self-efficacy in educational settings: Recent research and emerging directions. *Advances in motivation and achievement*, 16(Part A), 1-33.
- Palmer, S. R., & Felsing, M. (2001). *A practical guide to feature-driven development*: Pearson Education.
- Popov, V., Brinkman, D., Biemans, H. J., Mulder, M., Kuznetsov, A., & Noroozi, O. (2012). Multicultural student group work in higher education: An explorative case study on challenges as perceived by students. *International Journal of Intercultural Relations*, 36(2), 302-317.

Schwaber, K., & Sutherland, J. (2013). The definitive guide to scrum: The rules of the game. *Improving the Profession of Software development*.

Sedano, T., Ralph, P., & Péraire, C. (2019). *The product backlog*. Paper presented at the 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE).

Stapleton, J. (1997). *DSDM, dynamic systems development method: the method in practice*: Cambridge University Press.

Vogler, J. S., Thompson, P., Davis, D. W., Mayfield, B. E., Finley, P. M., & Yasseri, D. (2018). The hard work of soft skills: augmenting the project-based learning experience with interdisciplinary teamwork. *Instructional Science*, 46(3), 457-488.

## AUTHORS

**Dipendra Ghimire** completed PhD in software engineering from Lincoln University, New Zealand in 2017. His study provided a framework to assist practitioners and researchers to understand and manage effective teamwork in Agile software development projects. Before completing his doctorate at Lincoln University, he completed his Master of Information Technology at Central Queensland University Australia. He is currently working as the IT programme leader at Aspire 2 International, Christchurch campus. He is also involved in research projects on Agile software development, Agile in teaching and learning, IoT and Human-computer interaction. Dipendra is a recipient of Best Paper Award in CITRE NZ 2016 Conference.

**Alaa Aljanaby** has a PhD in Information Technology and has served as a lecturer and assistant professor at several colleges and universities, currently working as academic manager at Aspire2 International. He has published over 20 papers in international journals and conferences. His research interests include swarm intelligence, evolutionary computation, persuasive technology, opinion mining and ICT education.