Article

The Impact of Agile Development Practices on Project Outcomes

Dipendra Ghimire [1,2,*] and Stuart Charters [2]

1   Department of Business and Digital Technologies, Ara Institute of Canterbury,
     Christchurch 8011, New Zealand
2   School of Landscape Architecture, Lincoln University, Lincoln 7647, New Zealand
*   Correspondence: dipendra.ghimire@ara.ac.nz

Abstract: Agile software development methods were introduced to minimize problems faced using traditional software development approaches. There are several Agile approaches used in developing software projects, these include Scrum, Extreme programming and Kanban. An Agile approach focuses on collaboration between customers and developers and encourages development teams to be self-organizing. To achieve this there are different Agile practices teams choose to use in their projects. Some teams only use one practice whilst others use a combination of practices. The most common practices used are stand-ups, user stories, Burndown chart/Burnup chart, pair programming, Epic and User stories. This paper reports on the analysis of the data collected from people involved in Agile software development teams and identifies that the combination of practices in Agile software development have an impact on the communication in the team, project requirements and project priorities, with more practices being adopted correlating with better project outcomes.

Keywords: agile software development; teamwork; software factors

1. Introduction

Agile software development is a more flexible way to develop software compared to a plan driven approach. This flexibility can be seen with the ability to change requirements in any phase of the software development cycle [1]. An Agile approach focuses on collaboration between customers and developers and encourages development teams to be self-organizing [2]. To achieve this there are several Agile practices that can be used in projects. Agile practices improve formal and informal communication during the software development process [3]. Some practices that impact on communication are open office space, product backlog, sprint planning, iteration planning meetings, iteration reviews, daily meetings, and iteration retrospectives [4,5]. However, a study from Pikkarainen et.al [6] was mainly focused on the practices that are likely to have an impact on the communication within the team. There is little evidence on how the combination of different approaches and practices impact on the project completion time and the budget. This indicates a need for investigating different approaches and practices used in Agile projects. This study explores the different practices used in Agile software development projects and impact of their combination on project outcomes.

This paper first summarizes the current research literature on Agile approaches and practices, and then discusses the impact a combination of approaches and practices may have on the project outcomes.

The remainder of the paper is organized as follows. Section 2 presents a brief background and summarizes related work in the area. Section 3 presents the study design. Section 4 introduces the research methodology and explains the different steps used in conducting this research. The findings of this study are presented in Section 5 followed by the conclusion and suggestions for future research in Section 6.

2. Literature Review

A range of software development approaches have the agile descriptor applied to them, these methods include: Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), Adaptive Software Development (ASD), and Dynamic Systems Development Method (DSDM) [7,8]. Six factors that influence agile practices identified by Medina & Mauricio [9]: confidence, perceived self-efficacy, stress at work, integrity and availability of information and experiences learned, Quality of working life, and Media used. In examining these factors Medina & Mauricio [9] did not consider the impact of practices on project outcomes. They suggested that future studies could implement strategies to mitigate negative factors and strengthen positive factors in software project outcomes.

A study done by Hummel [10] found that daily standups impact on the communication in the team. It has a positive effective on the frequency of communication within the team. Sprint planning was used to share the information. Sprint reviews facilitate informal communication among the developers. Their findings were based on a systematic review.

Building on the work of Chao and Cao [11] and Misra et al. [12], Tam et al. [13], explored the impact of people-factors on the success of agile software development factors and found that team capability and customer involvement can explain the variance in project success.

This work looks to explore the impact of the different technical practices used by development teams on project outcomes.

3. Study Design

The research question that this research addresses is:

What impact on project outcomes do the Agile practices used within the development team and with the customer have?

Different Agile approaches use different practices [5]. After conducting a literature review a list of common agile practices was created and is shown in Table 1 [2,7–11]. This was validated with five people with Agile experience in the IT sector for at least 15 years. The list below, was given to the respondents to identify any of the practices used in their projects.

Table 1. Practices used in Agile projects.

Practice	Description
Stand-ups	A 15-min standing meeting where the team provide information about the progress of the project and if there are any issues arising.
Continuous integration	The process of bringing together the work done by the developers when changes are made.
Backlog	The list of prioritized deliverables that are to be implemented in a project.
Pair Programming	A technique where two developers team work together at one workstation.
Burndown chart/Burnup chart	A visual record of work done or to be done in a project.
Definition of Done	when all acceptance criteria that each deliverable must meet are met and ready to be released to a customer.
Refactoring	The process of improving the structure of existing code while the external behavior is preserved.
Scrum board	This board is the visual display of the status/progress of work in each sprint.
Kanban board	A visualization tool which provides a visual progress of project task, workflows, and communications.

Table 1. *Cont.*

Practice	Description
Retrospective	This is a meeting where team reflect on what worked, what did not, and why.
Epic	A body of work where work can be broken down into user stories.
Sprint/Time box	A time boxed period where a team works to complete a set amount of work.
User Stories	General explanation of software feature from the user perspective.
Planning Poker	Technique developers use to estimate the effort of a task.
Personas	A text portrait of an actual or potential user of the product.
Automated Test	Test cases are created and will run automatically when the new code is pushed to the repository.
Online Tools	Tools that can be used to visualize, track, and communicate.
Definition of Ready	Technique used to determine whether work on a task is ready to be started.
Unit test	Is a type of software testing where individual units/components of software are tested.
Continuous development	This is a practice of routinely integrating code changes to the main repository of the system.

An online questionnaire was designed to gather information about practices used in projects shown in Table 1, the challenges encountered, completion time and budget along with background information, including their role, time they have been working in software development and the size of the organization.

4. Data Collection

4.1. Method

The questionnaire was administered using the online questionnaire tool Qualtrics (Qualtrics, Provo, UT, USA). People involved in software development were recruited as participants across New Zealand, via email, through professional and participant networks, and company contact email addresses. Completing a questionnaire took on average 18 min. Seventy-three responses to the questionnaire were received in 5 months.

4.2. Population

Seventy-three responses were received representing fifty organizations. To characterize the population we categorized the size of organization as shown in Table 2 and the predominant type of development activity they undertook as detailed in Table 3. The largest group of respondents were from organizations with between 41–100 employees (34%). 12 respondents, 15.8% of the total response, represented small organizations with less than 5 employees.

The largest group of the respondents, as shown in Table 3 categorized their organization as a "product development" company; these made up 50% (38) of the respondents. Thirty-two per cent (24) categorized themselves as "in house" developers and 14% (11) as "bespoke development" companies.

Respondents were asked their role in the organization as shown in Table 4. Of the 73 respondents, the largest group of respondents were developers 42 (57%). The smallest represented role was Integrator with one (1%) respondent.

Table 2. Organization profile.

Organization Size (Employees)	Number of Respondents	Percentage (%)
More than 1000	5	6.85%
501–1000	7	9.59%
101–500	3	4.11%
41–100	25	34.25%
21–40	11	15.07%
11–20	3	4.11%
5–10	7	9.59%
Less than 5	12	16.44%

Table 3. Organizational categorization.

Company Type	Number	Percentage
Product development	38	50%
In-house	24	32%
Contract	11	14%
Other	3	3%

Table 4. Respondent roles.

Role	Number of Respondents	Percentage (%)
Developer	42	57.53
Testers	25	34.25
Scrum master	15	20.55
Business Analyst	15	20.55
Team leader	13	17.81
Product owner	11	15.07
Project manager	9	12.33
Line manager	7	9.59
Coach	6	8.22
Client representative	6	8.22
Agile mentor	6	8.22
Technical writer	5	6.85
Other	3	4.11
UI/UX	2	2.74
Trainer	2	2.74
Integrator	1	1.37

A total of 22 (30%) of the respondents have been working in the software development sector for more than 20 years whereas 7 (9.59%) of the respondent worked in IT projects for less than three years as shown in Table 5.

Table 5. Respondent experiences.

Years of Experience	Number of Respondents	Percentage (%)
20 years or more	22	30.14%
16 to 19 years	9	12.33%
13 to 15 years	6	8.22%
10 to 12 years	7	13.70%
7 to 9 years	8	10.96%
4 to 6 years	11	15.07%
1 to 3 years	7	9.59%

4.3. Agile Practice Adoption

Respondents were asked to select the practices they use in their projects from the list presented in Section 3. They also had the ability to add any practice they used that was not on the list. Respondents were able to select more than one practice. This information was used to analyse the combination of the different Agile practice used in the projects. Table 6 presents the number of practices used by the respondents in their project.
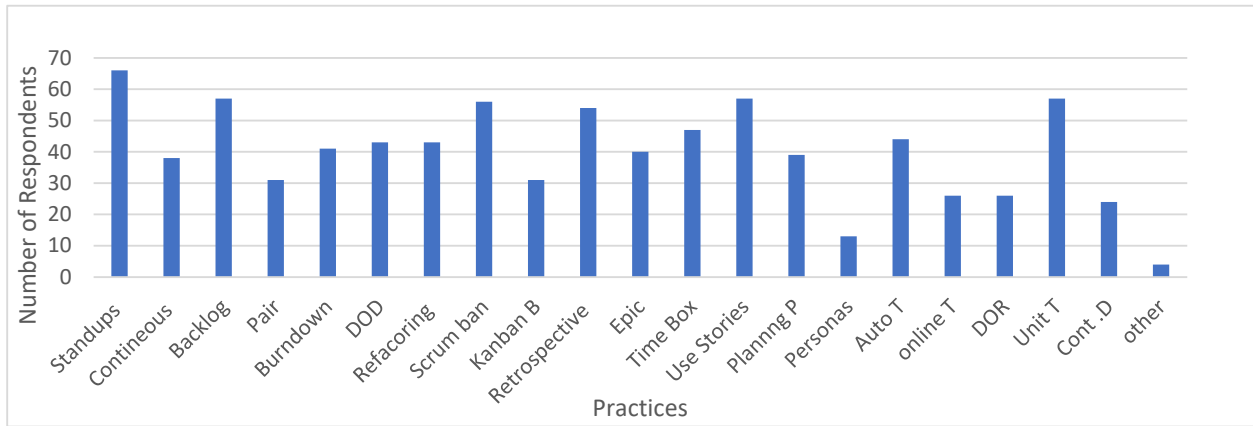
Table 6. Number of Practices used.

Number of Practices	Number of Respondents
20	2
19	2
18	1
17	5
16	5
15	9
14	6
13	2
12	3
11	9
10	6
9	3
8	3
7	3
6	5
5	2
4	3
3	1
2	2
1	1

All respondents used at least one practice, only two respondents out of the 73 reported that they use all the twenty practices. One respondent reported that they use only stand-ups.

Stand-ups was the most commonly used practice with 90% (66) of the respondents reporting they used it. The least used practice was personas used by 18% (13) of respondents. The spread of practices and the number of times each was selected is shown in Figure 1. The approaches reported in the other category were:

- At least we use something between Scrum and Kanban
- "Agile" with small a
- Aspects of Agile but not the formal Agile process
- Clipper
- Fail-fix

Figure 1. Practices used in projects.

The mean number of approaches used in a project was M = 11. Figure 1 summarizes the practices used in projects.

There were some practices which were used by more than 70% of the respondents. The following Table 7 summarizes the most used practices during software development.

Table 7. Most frequent approaches.

Practices	Number of Respondents
Standups	66
Backlog	57
Unit Test	57
User Stories	57
Scrum ban	56
Retrospective	54

Respondents for this study were asked to share their experience of difficulties in different areas during the development of their projects. The areas were challenges; in communication, interpersonal, sharing ideas within the team, distribution of work within the team.

Respondents were also asked to rate the level of disagreement with the customer on project priorities, requirements, and timeframe. These findings are discussed in the next section.

5. Findings

5.1. Challenges within the Team

Challenges within the software development team are an issue as this could impact on the outcome of the projects. Several questions were asked of respondents with answers being chosen from several listed options regarding challenges faced during the project development within the team. One question was "of the challenges your team faces how many times during a typical sprint would "Difficulties communicating within the team." This challenge occurred most as 1–3 times with a total of 38 (52%) responses. Of the 73 respondents 17 (19.17%) respondents reported that they do not face any difficulties in communicating. Table 8 presents the challenges faced within the team.

Table 8. Challenges within the Team.

Options	0 Times	1–3 Times	4–6 Times	7–9 Times	More than 9 Times	Total
Difficulties in communicating within the team	14	38	15	3	3	73
Interpersonal challenges within the team	19	41	7	4	2	73
Difficulties in sharing of ideas within the team	22	36	8	4	3	73
Problems with distribution of the work within the team	21	31	14	2	5	73

5.2. Challenges between the Team and Customer

To meet the aim of examining the impact of practices used in a project, data on the relationship between the team and the product owner was collected. Forty-three (43) respondents reported that they had one to three disagreements with the customer about project priorities in a typical sprint. Three respondents reported challenges in communication with the customer as many as seven to nine times during a sprint. A summary of responses reporting the challenges faced is presented in Table 9.

Table 9. Challenges between the Team and Customer.

Options	0 Times	1–3 Times	4–6 Times	7–9 Times	Total
Disagreement with the customer about project priorities	23	34	6	1	73
Disagreement with customer about project requirements	18	43	4	1	73
Disagreement with the customer about the timeframe of the project	31	25	7	3	73
Interpersonal challenges between the team member(s) and the customer	37	26	4	0	73
Challenge in communicating with the customer	26	30	7	3	73

5.3. Completion of the Project

Respondents were asked if projects were completed on time and on budget with responses shown in Tables 10 and 11. One (1.37%) respondent reported their project was completed early, 39 (53.42%) respondents reported that their project was completed on time. From the total of 73 respondents 18 (24.66%) reported that the project was not completed on time.

Table 10. Project completion time.

Project Completed	Number of Respondents	Percentage
Before time (Early)	1	1.37%
On time	41	56.16%
Not completed on time (Late)	21	28.77%
I Don't know	10	13.70%

Table 11. Project Budget.

Budget Performance	Number of Respondents	Percentage
Don't know	29	39.73%
Within a budget	25	34.25%
More than budget	17	23.29%
Less than budget	2	2.74%

Twenty-five (34.25%) respondents reported that their project was completed within budget. 17 (23.20%) respondents reported that the project cost more than the original budget. Most respondents 29 (39.73%) had no knowledge of project budgets.

6. Results and Discussion

To understand the impact of combinations of practices on projects the Pearson's Product Moment Correlation Coefficient was calculated with variables using SPSS: timeline of the project, budget and challenges within the team and shown in Table 12. To quantify the strength of a relationship the correlation coefficient (r) is calculated and can have a value of between +1 and −1. Correlation coefficient values greater than zero indicate there is a positive association between the two variables [14]. The closer to +1 or −1 r is, the stronger the relationship [15].

Table 12. Significant correlation between number of practices vs. challenges.

Variables	Challenges in Communication within the Team	Challenges in Priorities with Customer	Challenges in Requirements with Customer
Number of practices	Pearson Correlation-0.272 * Sig(2-tailed) = 0.020 $n = 73$	Pearson Correlation-0.236 * Sig(2-tailed) = 0.044 $n = 73$	Pearson Correlation-0.283 * Sig(2-tailed) = 0.053 $n = 73$

* Indicate that the correlation is significant.

The data suggest that there is a negative linear relationship between the number of practices used in the project and challenges in communication within the team. (R = −0.272). This suggested that if there were an increase in the number of practices used in the project there tended to be a decrease in challenges within the development team members. This is supported by Liu & Zhai [16] finding that Agile practices such as open office space, product backlog and sprint planning used in the projects have an impact on communication inside the development teams. For example, standup meetings were found to help the development team to be aware of what they should communicate, such as process and blocker issues. Retrospectives, held following a sprint, enables teams to assess the current situation and define actions to address the issues in the remainder of the project as well as in the future projects [5,16]. However, their study did not examine the impact of using more practices on the communication within the team during the software development projects.

In Agile software development, requirements evolve over time [17]. The data from this study suggests a negative correlation between the number of practices used in Agile software development and the challenges faced with customers in defining the requirements. This suggests that if there were an increase in the number of practices used in the project there could be a decrease in challenges faced between the customer and development team in defining the requirements. This may be because having more practices means more frequent interaction among stakeholders that could lead to better communication. Such an approach can make requirements clearer over time and may strengthen relationships with the customer [18–20]. For example, a daily meeting may be useful in keeping developers,

project leaders and customers aware of the project status. This helps the team members in agile teams to be clearer in the task and the status of projects.

Likewise, user stories can make it easier for teams to divide the work into tasks, where the stories bring the customer closer to the development and may help to reveal the core requirements.

Requirements prioritization is a part of each iteration in Agile software development methods. Requirements are prioritized in each iteration cycle by the customer. This practice gives value to the business to define requirements correctly. This in turn means that the development teams have a better understanding of the requirements right from the beginning of the project. Findings from this study have highlighted several practices used in Agile software development projects that have a negative linear relationship with the challenges in prioritizing the requirements. This may mean that having more practices used in the project may minimize the challenges in setting up the priorities for the projects. Practices such as standups, pair programming, and retrospectives could provide more opportunity for discussion in the team, and to get the feedback from the customer. For example, Sprint reviews could provide the opportunity to get a view of the customer and what they are thinking about in terms of needs and the services.

To understand the relationships between the number of practices and challenges in sharing ideas within the team, project completion time and project budget, the Pearson's Product Moment Correlation Coefficient was calculated. Those correlation coefficients for the variables with non-significant (n.s) linear relationship are presented in Table 13. The non-significant linear relationship means that the Pearson correlation calculated provided little or no evidence of the relationship between these variables. Using a number of practices in Agile software development were expected to have given the software development team members opportunity to share the ideas between the members. When team members get involved in more practices, they can be involved more in the project which could improve the communication and build trust between team members. However, this was not found to be significant.

Table 13. Non-significant relationship number of practices and challenges.

Variables	Challenges in Sharing the Ideas within the Team	Project Completion Time	Project Budget
Number of practices	Pearson Correlation-0.206 Sig(2-tailed) = 0.080 *n* = 73	Pearson Correlation-0.215 Sig(2-tailed) = 0.068 *n* = 73	Pearson Correlation-0.082 Sig(2-tailed) = 0.490 *n* = 73

Challenges in sharing of ideas within the team was found to have non-significant relationship (R = −0.206) with the number of practices used in the Agile projects. However, there is an indication that there could be a significant negative relationship between these two variables. It is interesting that when the development team are more involved with the team member having more practices that could help them to collaborate between them and reduce the challenges in sharing the ideas between the team members. However, the data could not confirm this.

Correlation between the number of practices and project timeline is calculated. The value of R is −0.215. This indicates that there could be a negative correlation between the number of practices used in the projects and the project timeline. This could mean that the projects where more practices are used may have impact on the timeline of the projects. However, it is interesting that there is no significant correlation between the timeline and the number of practices used in the project.

Project budget was found to have a non-significant relationship (R = 0.082) with the combination of the practices used in the projects. This result was not expected as the practices would have required more resources and time, which could increase the project

cost. However, this result showed that the combination of more practices may not impact on the budget of the project. This could also mean that increases of practices can make things more time efficient in completion of the project.

7. Conclusions

This study reports on the analysis of the data collected from people involved in Agile software development team.

This study could help the software development companies in New Zealand to be aware of the importance of Agile practices used during the software development process and their impact on the challenges faced with the timeline and the project budget.

Findings identify that the combination of practices in Agile software development have impact on the communication in the team, project requirements and the project priorities. The identification of such impacts help companies using an Agile software development approach to consider using different practices during the development of new project.

Among such practices the common practices that could have higher impact on the communication in the team, project requirements and project priorities could be Standups, Backlog, Unit Test, User Stories, Scrum ban and Retrospective.

We propose that the correlation reinforces that an "Agile-lite" approach to agile adoption does not result in good project outcomes. An increasing use of agile practices reflects a greater adoption of agile as an approach. Increasing use of agile practices, particularly standups, sprint planning and retrospectives provide greater structured opportunities for communication. This is supported by Hummel et al. [10]. Increasing use of agile practices also indicates that development teams have put thought into which practices they will adopt, and therefore are more likely to have a considered approach to software development.

The work reported is part of a larger study. In this paper the importance of Agile software development practices and their impact on the projects is explored. Information presented here may help to inform the larger project and aid in the understanding of what can be the impact of practices on the relationship between the customer and an Agile software development team. This work could help the software development community to understand the viewpoint of the development team including business analysts, product owners and project managers from their experience to develop successful projects.

There are many factors that impact the outcome of a project [11–13,16–19] but a key element of all studies is the importance of the human aspects, in particular, communication. Our work reinforces this suggesting that organizations looking to improve project success should focus on communication, both within the development team and between the development team and the product owner or customer.

8. Future Work

Future work can be done to identify the impact of the practices using a particular Agile methodology during software development projects. This study could also be replicated in other countries to compare and confirm the findings. Future work can also be done to understand the methodologies used by different teams such as integration team and project managers.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Larger sets of data can be found in the thesis. Available online: https://researcharchive.lincoln.ac.nz/handle/10182/10074 (accessed on 5 February 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dönmez, D.; Grote, G.; Brusoni, S. Routine interdependencies as a source of stability and flexibility. A study of agile software development teams. *Inf. Organ.* **2016**, *26*, 63–83. [CrossRef]
2. Ozkan, N.; Gök, M.Ş.; Köse, B.Ö. Towards a better understanding of agile mindset by using principles of agile methods. In Proceedings of the 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 6–9 September 2020; pp. 721–730.
3. Dorairaj, S.; Noble, J.; Malik, P. Effective communication in distributed agile software development Teams. In *International Conference on Agile Software Development*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 102–116.
4. Sandstø, R.; Reme-Ness, C. Agile practices and impacts on project success. *J. Eng. Proj. Prod. Manag.* **2021**, *11*, 255–262.
5. Baham, C.; Hirschheim, R. Issues, challenges, and a proposed theoretical core of agile software development research. *Inf. Syst. J.* **2021**, *32*, 103–129. [CrossRef]
6. Pikkarainen, M.; Haikara, J.; Salo, O.; Abrahamsson, P.; Still, J. The impact of agile practices on communication in software development. *Empir. Softw. Eng.* **2008**, *13*, 303–337. [CrossRef]
7. Abrahamsson, P.; Warsta, J.; Siponen, M.; Ronkainen, J. New directions on agile methods: A comparative analysis. In Proceedings of the 25th International Conference on Software Engineering, Portland, OR, USA, 3–10 May 2003; pp. 244–254.
8. Boehm, B.; Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*; Addison-Wesley Professional: Boston, MA, USA, 2003.
9. Arcos-Medina, G.; Mauricio, D. Identifying Factors Influencing on Agile Practices for Software Development. *J. Inf. Organ. Sci.* **2020**, *44*, 1–31. [CrossRef]
10. Hummel, M.; Rosenkranz, C.; Holten, R. The Role of Communication in Agile Systems Development. *Bus. Inf. Syst. Eng.* **2013**, *5*, 343–355. [CrossRef]
11. Chow, T.; Cao, D.-B. A survey study of critical success factors in agile software projects. *J. Syst. Softw.* **2008**, *81*, 961–971. [CrossRef]
12. Misra, S.C.; Kumar, V.; Kumar, U. Identifying some important success factors in adopting agile software development practices. *J. Syst. Softw.* **2009**, *82*, 1869–1890. [CrossRef]
13. Tam, C.; Moura, E.J.D.C.; Oliveira, T.; Varajão, J. The factors influencing the success of on-going agile software development projects. *Int. J. Proj. Manag.* **2020**, *38*, 165–176. [CrossRef]
14. Taylor, R. Interpretation of the Correlation Coefficient: A Basic Review. *J. Diagn. Med. Sonogr.* **1990**, *6*, 35–39. [CrossRef]
15. Jeong, H.; Mason, S.P.; Barabasi, A.; Oltvai, Z.N. Lethality and centrality in protein networks. *Nature* **2001**, *411*, 41–42. [CrossRef] [PubMed]
16. Liu, D.; Zhai, Z. An Empirical Study of Agile Planning Critical Success Factors. Master's Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, June 2017.
17. Moniruzzaman, A.; Hossain, D.S.A. Comparative Study on Agile software development methodologies. *arXiv* **2013**, arXiv:1307.3356.
18. Cao, L.; Ramesh, B. Agile Requirements Engineering Practices: An Empirical Study. *IEEE Softw.* **2008**, *25*, 60–67. [CrossRef]
19. Ghimire, D.; Charters, S.; Gibbs, S. Scaling agile software development approach in government organization in New Zealand. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, Sydney, NSW, Australia, 12–15 January 2020. [CrossRef]
20. Salameh, H. What, when, why, and how? A comparison between agile project management and traditional project management methods. *Int. J. Bus. Manag. Rev.* **2014**, *2*, 52–74.