CROWDLY

Reflective Report

Agile Development

Team Charliezz

MANISH || CHIRAG || SALIL ||
NISCHAL|| ADARSHA

Contents

Table of Figures

Introduction:

Crowdly, a free-lancing social media platform, is aimed for simplifying recruiting and hiring process for the parties involved. With this system, employers, and potential employees could connect directly, efficiently cutting out the middleman. Moreover, with a real-time communication system integrated into this platform, employers can be confident about their applicants. In the favor of applicants, Crowdly offers total transparency in their application process by informing them about state of their application. The process of searching jobs remains one of the most hassled tasks in one's professional life. Furthermore, the lack of transparency in the process leaves the applicants unsure about their application. This creates sort of an unprofessional atmosphere between the employer and the potential employee. To address these, Crowdly was founded. With this platform, the job searching process will be made easier and hassle free. The users can apply for any job that matches their qualifications and skill sets. Their profile will directly be sent to the employers without having to upload a CV. In the favor of Employers, the employers can commit to 100% transparency of application by providing them with status of their application.

Figure 1: Crowdly

Furthermore, the integration of social media platform will play positive role to form professional network of the parties involved. Users, both applicants and company, will be able to share their views and thoughts in the form of a post. This post will be accessible by the users who have followed the users in the Crowdly social media. The post, in respect to social media, have properties such as likes and comments which will allow other users to express their views or opinions. With regard to employer, the system allows employers or company to post job vacancies which contains information about the job such as job title,

location, description and so on. The system, however, focuses on freelancing jobs or projects rather than on traditional jobs. Moreover, the system, like a freelancing platform, has a payment option to release payments to the employee/ candidate once the job is completed. Talking about the minor features, Crowdly allows users to save jobs and posts into user's personal accounts which can be latter visited by the user, if needed. In regard to improving communication, Crowdly offers real-time communication with the users. The employers can communicate with the candidates to clear any confusion or to make an offer. Similarly, the candidates can also message employers to explain their views regarding the job posting and explain their proposal directly. This system encourages communication from within our application and discourages irrelevant communication through third party applications such as Gmail or Outlook.

Why Agile?

Agile is an iterative methodology for project management and software development that enables teams to produce client value more quickly and with fewer complications. The interesting fact of the agile is it supports for the delivery of 80% of the product in only 20% of the time.

Figure 2: Scrum Framework

In addition, agile consists of different methodologies among which Scrum, Kanban, Extreme Programming (XP), Lean Development e crystal, etc. are popular. However, Scrum is the most popular version of the various frameworks that support the Agile development approach. For the development of Crowdly, the scrum approach is followed. The scrum approach is a sprint-based project management technique designed to provide the most value to stakeholders. This was used to complete this project in efficient manner by dividing the

entire project into manageable segments called "sprints". The sprint was completed in the duration of eight weeks. During each sprint, the team worked to complete the most crucial features and released a usable product. Additional product features were developed in following sprints, with iterations based on input from stakeholders and product owner. Every day began with a short, 15-minute meeting called the daily Scrum, which served to synchronize operations and discover the best method to schedule out the functioning day, permitting for a review on sprint "health" and product development. Overall, the Scrum Framework comprised of a variety of components, including accountabilities (people on scrum team), events (sprint, planning, review, retrospective), and artifacts (product backlog, sprint backlog, increments), in addition to additional aspects that provide support to Scrum.

Now, the question is why the agile framework was preferred. Considering the weight of the product, Agile scrum was used which helped to complete the project in effective and efficient manner. With scrum, an adaptive, rapid, flexible, and efficient framework, any changes in requirements met with a rapid response. The daily scrum meeting not only enhanced communication, but also identified bottlenecks, and solved issues, and helped in fostering the creation of new ideas. Since scrum is focused on sprints, learning from prior sprints contributed to the development of a high-quality standard. Moreover, due to the frequent meetings and feedback received on each, the development team received clear idea of requirements and helped in development of quality product. However, in the case of traditional project management methods, there is a visible communication gap which causes valuable time is be lost while adjusting the product midway if the requirement changes. For instance, even if the product has been developed, the team has to begin over from scratch if the requirement is changed or the result is unsatisfactory.

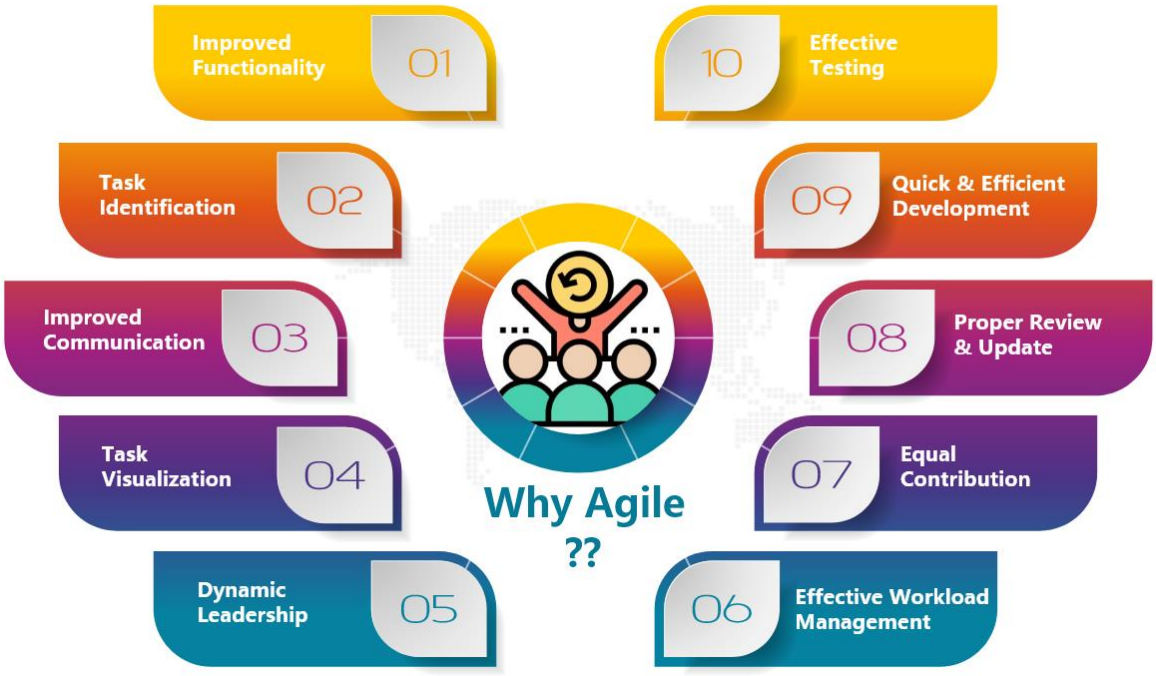
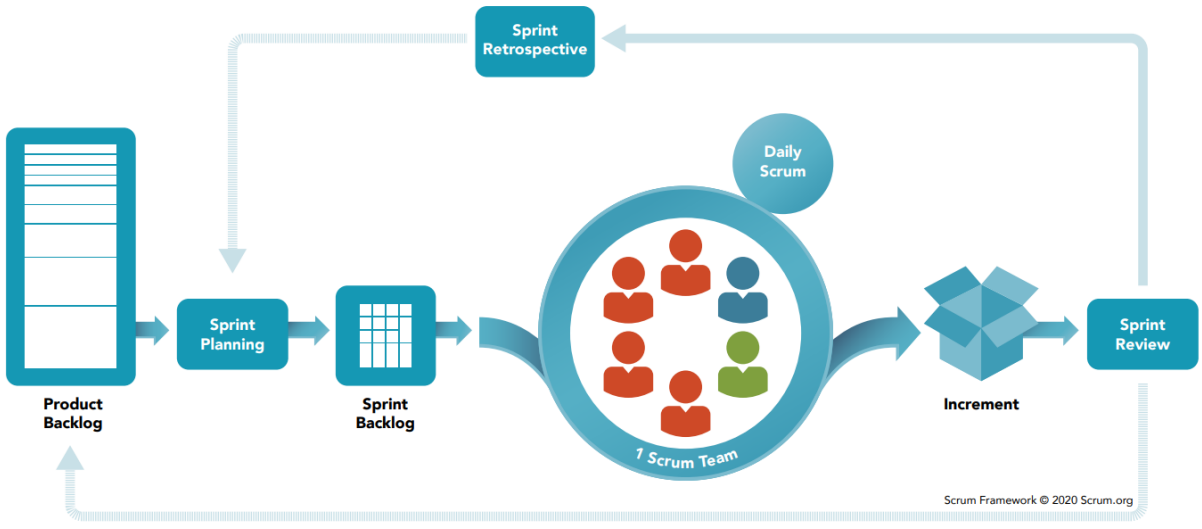


Fig: why to choose agile?

Furthermore, since the scrum continues with sprint basis works, tasks are divided between scrum teams and some portions of product is released every week achieving certain **functionality** which is reviewed between product owner and development team. As the **progress report is visible** simultaneously along with the development process, project delivery is **quick** and **efficient**. Moreover, burndown chart used in the scrum **visualize** completion of allocated tasks within the time that has been allocated to respective sprints. This helps in **identifying the tasks** remained to complete within the deadline. After analyzing the workflow through the burndown chart, team can adjust or maximize the time they have available in each sprint by **communicating** with each other for help if needed. As a result, the time allocation process in scrum enhances the development process faster. Similarly, as a part of scrum development process, **testing** has been done along with the coding during each sprint review to find out the bugs and remove the possible risks during the software development. Every sprint task/code has been tested in Crowdly development phase and helped to find some bugs and human errors resulting the reduction of future risk in the project. Likely, daily standup meeting is held to **review** and **update** daily progress which help to find out the conflict arises during the development. Not only the conflict management, but also the effort of team member is visualized and recorded in the daily meeting. Overall, agile scrum framework helps in the **communication, improves collaboration, visualize individual contribution** along with **develop leadership** skills for which it is preferred more in the Crowdly software development.

Agile Process

Agile is a project management methodology that places a strong emphasis on collaboration, speed, and adaptability. The Agile Manifesto, a collection of ideals and ideas that direct the creation of software and other products, serves as its foundation. The agile process involves iterative development, where requirements and solutions are developed in concert by self-organizing and cross-functional teams. The emphasis is on frequently offering tiny bits of value, roughly every week. This enables quick change adaptation and ongoing improvement.



For this final year project of the Agile Development, a group of 5 members was formed. The team members were picked for the project based on their knowledge, skills, and availability. The team members include Chirag, Nischal, Manish Adarsh and Salil. It was decided by the entire team that Chirag would be the scrum master and product owner. After discussion among the team members, it was decided that team Charliezz will create a web-based application called Crowdly, a free-lancing social media platform aimed for simplifying recruiting and hiring process.

Crowdly (Charliezz)	
Team Members	**Role**
Chirag Simkhada	**Scrum Master/ PO**
Manish Silwal	**Development Team**
Nischal Bade	
Adarsha Khadka	
Salil Timalsina	

After deciding on the application, all the features were listed, and user stories based on the features were created which are just brief, straightforward descriptions of a feature or need written from the viewpoint of the end user. It is a technique for encapsulating the essence of what a user requires or desires from the product, and it is frequently used in Agile software development as a method of prioritizing and planning work.

Feature ID	Feature Name	USER STORY ID	PRIORITY	AS A	I WANT	SO THAT I CAN
						I can present my details to the employer.
			1.2	Applicant	A Login system	I can apply for job.
F1	User Account		1.3	Employer	A Registration system	I can reach out to the potential candidates.
			1.4	Applicant	A Registration system	I can use additional features from crowdly.
			2.1		newsfeed section	I can get good source of up to date info
			2.2	Employer	newsfeed section	I can promote and advertize my company.
F2	Newsfeed section		2.3	User	newsfeed section	I can reach out to the larger audience.
			2.4		newsfeed section	I can scroll to the news that I have missed.
			2.5		newsfeed section	I can portray my views about specific things.
			3.1	Applicant	Bidding System	I can present my offers to the employer.
F3	Bidding System		3.2	Employer	Bidding System	I can select a candidate which better suits to my interest.
			3.3	Employer	Bidding System	I can select the most qualified from the pool of applied candidates.
			3.4	Employer	Bidding System	I can select a candidate which better suits to my interest.
			3.5	Employer	Bidding System	I can be assured that my application is considered fairly.
F4	OTP Authentication		4.1	User	OTP Authentication	My account remains secured even in the case of data breach.
			4.2	User	OTP Authentication	the risk of my accout being compromised is drastically reduced.
			5.1	Applicant	User Profile	I can show my potential skills to the employer.
F5	User Profile		5.2	Employer	User Profile	I can evaluate the potential of applicant.
			5.3	Employer	User Profile	I can view applicant previous works and reviews of those works.
				Applicant	A user profile	I can edit my personal information.
F6	payment gateway		6.1	employer		I safely pay an applicant once the project is completed.
			6.2	applicant		I safely receive payment after the completion of project.
F7	Advanced Search		7.1	Applicant	Search	I can find projects using a variety of criteria.
			7.2	Employer	Search	I can go through the feeds of my personal interest.
F8	Review and Rating		8.1	Employer	Review and rate	Review and rate applicant according to their work.
F9	Social media platform		9.1	Applicant	Social Media	I can connect with people and companies.
			9.2	Employer	Social Media	I could have a place for noble causes and practices
			9.3	Employer	Social Media	I can revisit him/her later.
F10	Real Time Chat System		10.1	User	Chat System	I can discuss ideas with others who share my interests.
			10.2	Employer	Chat System	I can ask project updates in real time.
			10.3	Employer	Chat System	I can communicate with the applicant through a real-time chat system.
			10.4	Applicant	Chat System	I can provide project updates in real-time.
F11	Bookmark Feature		11.1		Bookmark Feature	I can review that job for later .

According to Agile, the ideal way to accomplish the aim is to build a ranked list of priorities, therefore after listing all the user stories, team members had a discussion on user stories and based on their significance and value to the project, user stories were ranked in order of priority. To do so, a website called Trello was utilized. All the user stories were listed on this website and prioritized according to color, with red denoting high priority tasks, yellow denoting tasks of medium priority, and green denoting tasks of lesser priority.

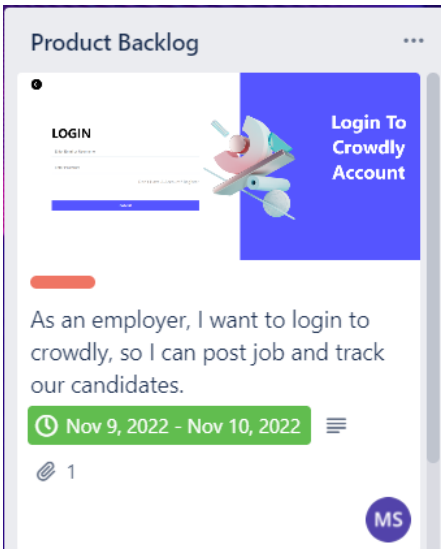


Figure 3: Product Backlog

After prioritizing all the user stories a map board was created which is agile process to visualize the progress of a project. It was created in Excel sheet and contains details on the work that must be done within a specific sprint. Here a sprint means a set period in Agile product development during which specific tasks must be completed to generate a map board also epic was separated, which is a significant quantity of work with a single common goal. In this process five different epic were listed which are user account, message, newsfeed, job system and payment system. There were eight sprints, and the tasks from each epic were divided into the appropriate sprints on a map board based on the importance of the task to the project and task weightage.

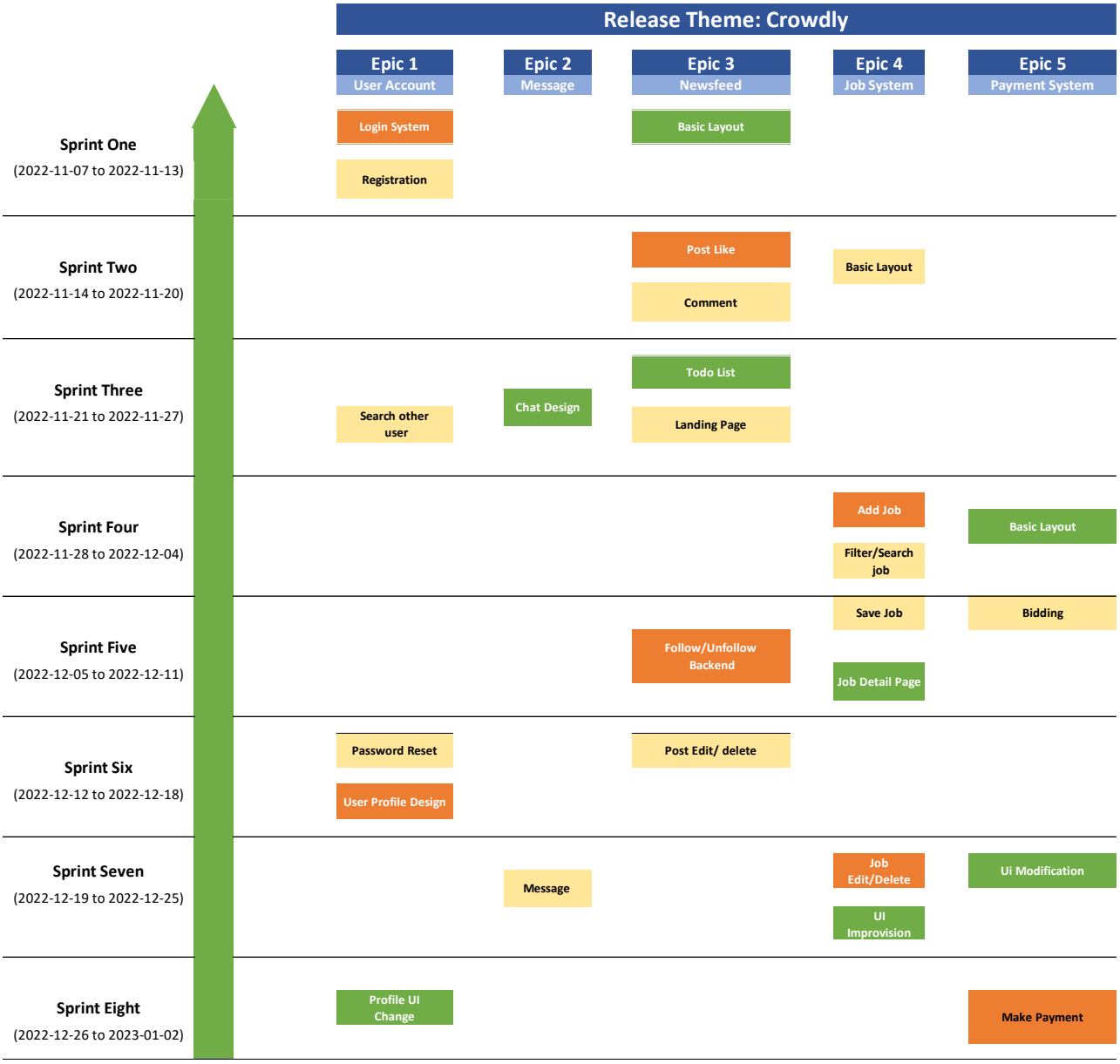


Figure 4: Map Board

After creating a map board in which tasks for each sprint were divided with team discussion a release plan was created by prioritizing product needs, development, and release. Simply,

release plan is a method of describing the specific features that will be included in a product's future sprint. It usually serves as a roadmap for the development team. In the release plan, the tasks that were broken down into individual sprints on the map board then tasks were selected, and a specific date was picked for their release and completion. The task's durations and story points were also assigned for each task based on their completion times and weightage of the task.

SPRINT	USER STORY NAME	FEATURE TYPE	START	FINISH	DURATION(IN HOURS)	STORY POINTS	RELEASE DATE
1	User Story 1- Sign Up		11/7/2022	11/9/2022	15	2.5	11/14/2022
1	User Story 2 - Sign In		11/10/2022	11/12/2022	12	2	11/14/2022
1	User Story 3 - Newsfeed Basic Layout		11/13/2022	11/14/2022	8	1.5	11/14/2022
2	User Story 4- Post Feed		11/15/2022	11/16/2022	12	2	11/22/2022
2	User Story 5- Like Feed		11/17/2022	11/17/2022	3	0.5	11/22/2022
2	User Story 6- Comment Feed		11/18/2022	11/19/2022	18	3	11/22/2022
2	User Story 7 - Job System Basic Layout		11/20/2022	11/22/2022	9	1.5	11/22/2022
3	User Story 8- User Profile Design		11/23/2022	11/25/2022	12	2	11/30/2022
3	User Story 9- Search Other User		11/26/2022	11/28/2022	16	3	11/30/2022
3	User Story 10- Chat System Design		11/29/2022	11/30/2022	9	1.5	11/30/2022
4	User Story 11- Search Job		12/1/2022	12/3/2023	9	1.5	12/8/2022
4	User Story 12- Filter/Sort Job		12/4/2022	12/6/2022	16	3	12/8/2022
4	User Story 13- Payment System Basic Layout		12/7/2022	12/8/2022	9	1.5	12/8/2022
5	User Story 14- Explore User's Posts		12/9/2022	12/10/2022	6	1	12/16/2022
5	User Story 15- Save Job Feature		12/10/2022	12/13/2022	18	3	12/16/2022
5	User Story 16- Job Details Page		12/14/2022	12/16/2022	6	1	12/16/2022
6	User Story 17- CV Profile		12/17/2022	12/20/2022	12	2	12/24/2022
6	User Story 18- Password Reset Feature		12/21/2022	12/24/2022	12	2	12/24/2022
7	User Story 19- User Profile Search		12/25/2022	12/25/2022	3	0.5	12/31/2022
7	User Story 20- Messaging Feature		12/26/2022	12/27/2022	6	2	12/31/2022
7	User Story 21- Bidding		12/28/2022	12/29/2022	9	1.5	12/31/2022
7	User Story 22- Apply Job Feature		12/30/2022	12/31/2022	9	1.5	12/31/2022
8	User Story 23- Payment Feature		1/1/2023	1/8/2023	24	7	1/8/2023

Figure 5: User Story

All the product backlog was listed in Trello after all the planning in release plan. During the sprint planning meeting, the product backlog was moved to the sprint backlog in accordance with the release plan in Trello. A sprint backlog is simply a list of the tasks that a development team must carry out during a sprint. The product backlog is a list of all the work that must be done in a project, and the sprint backlog is a subset of that list.

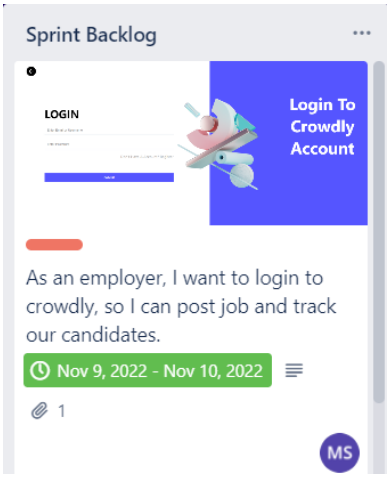


Figure 6: Sprint Backlog

To monitor the sprint progress after the start of each sprint, a burndown chart was created. It is a graphical representation of the amount of work that must be completed in a project over time. It is a simple tool in Agile software development for monitoring a sprint's progress.

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #1- Post Upload	3									
Backend Development		Chirag Simkhada	Done	2	1.5	0	0	0	0	1.5
Database Design		Salil Timalsina	Done	3	2.5	0	0	0	0	2.5
BDD Scripting		Adarsha Khadka	Done	3	0	0	1.5	1	0	2.5
Development (API & End-points)		Nischal Bade	Done	3	0	3	0	0	0	3
TDD Testing		Manish Silwal	Done	2.5	0	0	2	0.5	0	2.5
User Story #2- Post Like and Comment	3									
Backend Development		Nischal Bade	Done	2	2	0	0.5	0	0	2.5
BDD Scripting		Salil Timalsina	Done	3	0	0	1	1.5	0	2.5
Development (API & End-points)		Adarsha Khadka	Done	1.5	0	2.5	0	0	0	2.5
Database Design		Manish Silwal	Done	1.5	0	1	0	0	0	1
TDD Testing		Chirag Simkhada	Done	3	0	0	2	0.5	0	2.5
User Story #3- Job System Basic Layout	1									
Job Card		Nischal Bade	Done	1	0	0	0	0.5	0.5	1
Search Box		Salil Timalsina	Done	1	0	0	0	0.5	0.5	1
Filter		Manish Silwal	Done	2	0	0	0	1.5	0.5	2
Job Description		Chirag Simkhada	Done	2.5	0	0	0	0.5	1.5	2
Bidding Layout		Adarsha Khadka	Done	2	0	0	0	0	1	1
				33						**30**

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #1- Landing Page	0.5									
Frontend Development		Nishal Bade	Done	4	1	3	1	0	0	5
User Story #2- Search Other User	3									
Frontend Devlopment		Chirag Simkhada	Done	2	0.5	2	0.5	0	0	3
Backend Development		Adarsh Khadka	Done	2.5	0	2	0	0.5	0	2.5
BDD Scripting		Nischal Bade	Done	2.5	0	0	0	1	0	1
Development (API & End-points)		Manish Silwal	Done	1	0	1.5	0	0	0	1.5
TDD Testing		Salil Timalsina	Done	2	0	1	2	0	0	3
User Story #3- Chat Design	1									
Frontend Development		Adarash Khadka	Done	5	0.5	0	1.5	2	0.5	4.5
User Story #4- To Do List	1									
Backend Development		Salil Timalsina	Done	2.5	0	0	0	2	0	2
BDD Scripting		Manish Silwal	Done	2.5	0	0	0	0	3	3
Development (API & End-points)		Chirag Simkhada	Done	1	0	0	0	0.5	1.5	2
Database Design		Nischal Bade	Done	1	0	0.5	0.5	0	0.5	1.5
TDD Testing		Manish Silwal	Done	2.5	0	0	0	0	2.5	2.5
				30.5						**31.5**

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #5- Add Job	8									
Frontend Devlopment		Salil Timalsina	Done	3.5	1	2	0.5	0.5	0	4
Backend Development		Chirag Simkhada	Done	4	0	2	1.5	1	0.5	5
BDD Scripting		Manish Silwal	Done	4	0	0.5	0.5	2.5	0.5	4
Development (API & End-points)		Adarsha Khadka	Done	2.5	0	0	0	1.5	0	1.5
TDD Testing		Nischal Bade	Done	2.5	0	0	0	0	3	3
User Story #26 - Filter & Search Job	3									
Backend Development		Adarsha Khadka	Done	2.5	1	0	1	1.5	0	3.5
BDD Scripting		Chirag Simkhada	Done	3	0	0	1	1	2	4
Development (API & End-points)		Salil Timalsina	Done	3	0	0	0	0	3	3
TDD Testing		Manish Silwal	Done	3	0	0	0.5	0	2.5	3
				11.5						
User Story #32 - Payment Basic Layout	2									
Frontend Devlopment		Nischal Bade	Done	7	1.5	1	1	2	1.5	7
				46.5						**38**

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #1- Sign Up	3									
Backend Development (Database)		Adarsha Khadka	Done	3	3.5	0	0.5	0	0	4
Frontend Development		Nischal Bade	Done	4	2.5	0.5	0.5	0	0	3.5
BDD Scripting		Chirag Simkhada	Done	2	0	0	2.5	0.5	0	3
Development (API & End-points)		Manish Silwal	Done	3	1	0	0	0	0	1
Development (Functionalities)		Salil Timalsina	Done	3	2	0	0	0	0	2
TDD Testing		Nischal Bade	Done	1.5	0	0	0	0	2	2
User Story #2- Sign In	1									
Backend Development (Database & API)		Nischal Bade	Done	1	1.5	1	0	0	0	2.5
Frontend Development		Chirag Simkhada	Done	1	1	0.5	0	0	0	1.5
BDD Scripting		Manish Silwal	Done	2	0	0	1.5	1	0	2.5
Development (API & End-points)		Salil Timalsina	Done	2	0	0	2.5	0	0	2.5
Development (Functionalities)		Adarsha Khadka	Done	2	1.5	0	0	0	0	1.5
TDD Testing		Nischal Bade	Done	1.5	0	0	0	0	1	1
User Story #3- Newsfeed Basic Layout	1									
Post Design		Nischal Bade	Done	2	o	0	1.5	2	0	3.5
Comment Box Design		Salil Timalsina	Done	1	0	0	0	1	0	1
Post Upload Section		Manish Silwal	Done	1	0	0	0	1	0	1
Navbar		Chirag Simkhada	Done	2	0	0	0	1.5	0	1.5
User Recommendation + TO DO Layout		Adarsha Khadka	Done	2	0	0	2	1	0	3
				34						**37**

Figure 7: Sprint Burndown Chart - I

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #6- Bidding	3									
Frontend Devlopment		Chirag Simkhada	Done	3	1.5	0.5	0.5	0	0	2.5
Backend Development		Chirag Simkhada	Done	6	0	2	0.5	2.5	0.5	5.5
BDD Scripting		Salil Timalsina	Done	2.5	0	0	0.5	1	1	2.5
Development (API & End-points)		Nischal Bade	Done	5	0	0	0	0.5	1.5	2
TDD Testing		Adarsha Khadka	Done	2	0	0	0	0.5	1.5	2
User Story #44 - Bookmark Job	6									
Frontend Devlopment		Salil Timalsina	Done	3	1	1	1	0	0	3
Backend Development		Chirag Simkhada	Done	5	1	0	0.5	2.5	1	5
BDD Scripting		Manish Silwal	Done	2	0.5	0.5	0.5	0	0.5	2
Development (API & End-points)		Adarsha Khadka	Done	3	0	1	1	0.5	0.5	3
TDD Testing		Nischal Bade	Done	1	0	0	1.5	0.5	1.5	3.5
User Story #46 -Job Detail Page	3									
Backend Development		Manish Silwal	Done	3	0.5	1	1	1	0	3.5
BDD Scripting		Adarsha Khadka	Done	2.5	0	0	0	0	2	2
Development (API & End-points)		Salil Timalsina	Done	3	0	0	0	2	2	4
TDD Testing		Manish Silwal	Done	2	0	0	0.5	0	1	1.5
User Story #40 - Follow/Unfollow	3									
Backend Development		Nischal Bade	Done	2	1	2	1	0	0	4
BDD Scripting		Salil Timalsina	Done	2.5	0	0	0	0	2.5	2.5
Development (API & End-points)		Manish Silwal	Done	3	0	0	0	3	0	3
TDD Testing		Adarsha Khadka	Done	2	0	0	0	1.5	0.5	2
				52.5						**53.5**

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #6- Password Reset	6									
Frontend Devlopment		Adarsh Khadka	Done	2.5	1.5	0.5	0	0	0	2
Backend Development		Chirag Simkhada	Done	6	0	2	4	0	0	6
BDD Scripting		Salil Timalsina	Done	3	0	0	0	1	2	3
Development (API & End-points)		Manish Silwal	Done	3	0	0.5	2	0	0	2.5
TDD Testing		Chirag Simkhada	Done	3	0	0	0	2	1	3
User Story #5 - User Profile	9									
Frontend Devlopment		Nischal Bade	Done	8	0	3	2	0.5	2	7.5
Backend Development		Salil Timalsina	Done	6	0	3	3	0	0	6
BDD Scripting		Manish Silwal	Done	4	0	0.5	0.5	2	0.5	3.5
Development (API & End-points)		Chirag Simkhada	Done	3	0	1.5	0.5	1	0	3
TDD Testing		Adarsh Khadka	Done	2	0	0.5	1	0.5	0.5	2.5
User Story #49 - Post Edit/ Delete	6									
Frontend Devlopment		Nischal Bade	Done	6	5	0	1	0.5	0	6.5
Backend Development		Chirag Simkhada	Done	4	3.5	1	0	0	0	4.5
BDD Scripting		Adarsh Khadka	Done	2	0	0	0	1	1	2
Development (API & End-points)		Salil Timalsina	Done	3	0	2	1	0.5	0	3.5
TDD Testing		Manish Silwal	Done	3	1	0	0.5	1.5	0	3
				58.5						**58.5**

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #28- Message	12									
Frontend Devlopment		Chirag Simkhada	Done	12	4	4.5	3	0.5	0	12
Backend Development		Chirag Simkhada	Done	8	1.5	2	4	1	0	8.5
BDD Scripting		Salil Timalsina	Done	4	0	0	0	3	1.5	4.5
Development (API & End-points)		Manish Silwal	Done	6	0	0	4	1	0	5
TDD Testing		Salil Timalsina	Done	5	0	0	1	3	1	5
User Story #50 - Job Edit/ Delete	8									
Frontend Devlopment		Adarsh Khadka	Done	5	2	1	0.5	1	1	5.5
Backend Development		Adarsh Khadka	Done	3.5	0.5	1.5	1	1	0	4
BDD Scripting		Salil Timalsina	Done	5	0	0.5	2	2	0.5	5
Development (API & End-points)		Manish Silwal	Done	3	0	0	2	0.5	0	2.5
TDD Testing		Chirag Simkhada	Done	2	0	0	0	1	1	2
User Story - UI improvision Job Detail	2									
Improvisation		Nischal Bade	Done	7	1	4	2	0	0	7
User Story - UI Modification Payment	1									
Modification		Nischal Bade	Done	3.5	0	0	1	2	0.5	3.5
				64						**64.5**

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE(in hours)	DAY1	DAY2	DAY3	DAY4	DAY5	SPRINT REVIEW
User Story #32- Make Payment	9									
Backend Development		Chirag Simkhada	Done	10	5	2.5	2	0.5	0	10
BDD Scripting		Manish Silwal	Done	5	0	2	2.5	0.5	0	5
Development (API & End-points)		Adarsh Khadka	Done	5	1	1.5	1	1	0.5	5
TDD Testing		Salil Timalsina	Done	4.5	0	2	2	0	0.5	4.5
User Story #50 - User Profile Changes	3									
UI Changes		Nischal Bade	Done	7	2	4	0.5	0.5	0.5	7.5
User Story #50 - Password Change	6									
Backend Development		Nischal Bade	Done	4	2.5	1.5	0	0	0	4
Frontend Development		Manish Silwal	Done	3	2	0.5	0.5	0	0	3
BDD Scripting		Chirag Simkhada	Done	3	0	0	1	1.5	0.5	3
Development (API & End-points)		Salil Timalsina	Done	3	0	1	1	1	0	3
TDD Testing		Adarsh Khadka	Done	2.5	0	0.5	1	0.5	1	3
				47						**48**

Figure 8: Sprint Burndown Chart - II

After tasks from the sprint backlog were completed, they were moved to verify, and after the tasks had been verified by the supervisor, they were placed to the done section in Trello. A sprint retrospective meeting was held after the end of each sprint. The goal of the meeting was to review the previous sprint and determine how the team's practices and processes could be enhanced going forward. It is an effective way to identify areas for improvement, produce significant changes to your workflows and procedures, accelerate upcoming projects, and prevent your team from making the same mistakes. These agile meetings have the potential to turn into a severe case of the blame game if handled improperly, with everyone pointing fingers and yelling. So, a time was fixed for the meeting to prevent that from happening and run over what worked, what could have been improved, and next steps was discussed. After the sprint retrospective, what discussed was written and accordingly product backlog was updated.

Team Psychology

A team is a collection of individuals or animals with a same goal. Teams are ideally suited for activities with a high level of complexity and several interconnected subtasks. A group does not automatically constitute a team. A group, in general, is typically generated by a concerted effort that allows each individual to maximize his or her abilities and minimize their weakness. In the late 20th century, business theorists promoted the concept of forming teams. Divergent viewpoints exist regarding the effectiveness of this new management trend. Some view "team" as an overused and underutilized four-letter term. Others view it as a panacea that satisfies the human relations movement's ambition to merge what that movement considers to be best for employees and best for managers. Still others agree in the efficacy of teams but view them as potentially exploitative due to the fact that team effectiveness might rely on peer pressure and peer surveillance.

With regard to team Charliezz, the team was formed considering the **unique talent** that each members had. To be precise, the group formation process began with a thorough review of **individual's skillset**. The skillset included mastery over one these fields: Communication, Team Management, Problem Solving, Listening, Critical Thinking, Collaboration and Leadership. Communication is the bedrock of productive **teamwork**. It is crucial to communicate freely and honestly with your group members about expectations, timelines, and duties, whether you're preparing a presentation with your classmates or leading a new project at work. Establishing open communication channels fosters trust and creates a pleasant team environment. With this in mind, it was ensured that the team members had ability to **communicate** with one another before being included in the group. Next, the process included an evaluation of **user's behavior** and their **overall compatibility** with the team. Working in a team might be difficult at times, but it is more often a wonderful opportunity to discover fresh ideas, share diverse **viewpoints** and **experiences**, and improve one's own abilities. If you approach each group project as a learning opportunity, you can promote a more productive team atmosphere. However, if either of a team member is incompatible with team, it may affect the opportunity. Hence, the next step was to **ensure the compatibility**. In similar fashion, a member's adaptability, responsiveness, dedication was considered during team formation.

With regard to team behavior, it was observed that all team members were eager to take initiative. During the task division in sprint planning meeting, members themselves picked the task from a given list of assignments. Moreover, the members themselves evaluated the task at hand, estimated the time required for completion, determined the next step, drew a workflow and so on. This behavior of the team helped to effectively divide workload, identify problems and obstacles, and propose viable solutions. Also, the eagerness and enthusiastic nature of members helped in the early completion of the project. Almost all members were positive about the project and were self-motivated to complete their task. Furthermore, while doing their task, the group passed through an extensive research phase. It was during this phase that the curious nature of the members was revealed. It was observed that most of the time, the members went an "extra mile" to complete their task. For instance, the members tasked with frontend design themselves added validation and removed logical flaws from the user side. Correspondingly, the members were coordinating, and collaborative in nature. Although the team experienced some minor communication issues, it never became a reason for a sprint delay. The members constantly collaborated and coordinated with one another to complete their tasks. The presence of dependent task i.e., one task dependent on another, required for this behavior from the members.

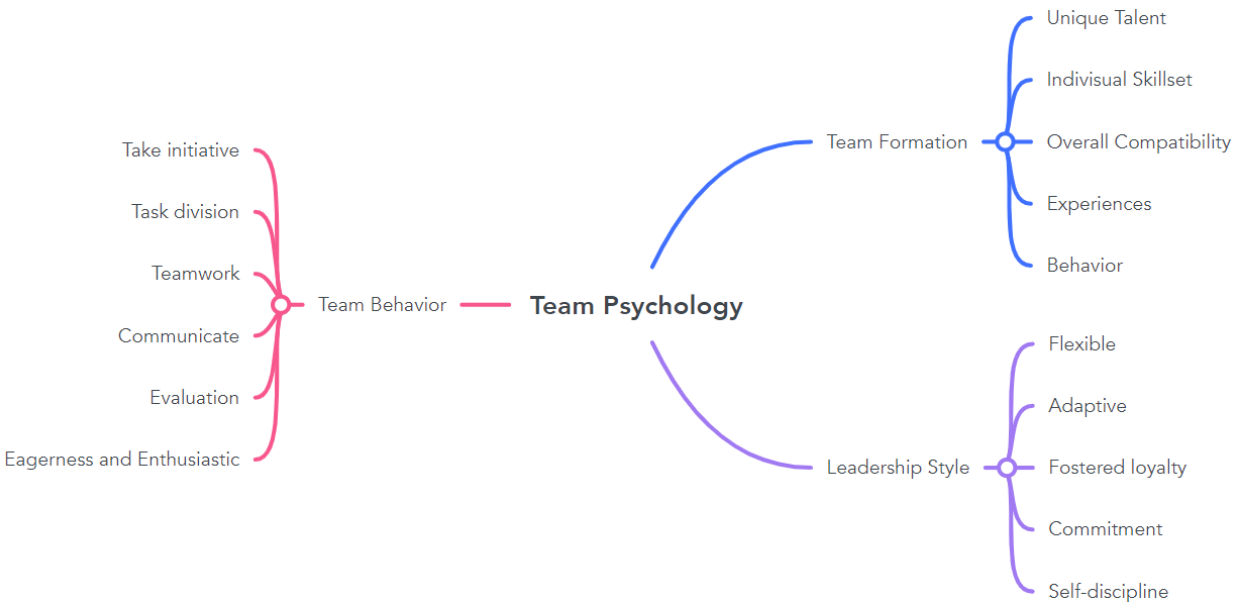


Figure 9: Team Psychology Overview

Regarding leadership style, the scrum master adapted a flexible and adaptive approach. By prioritizing developer engagement and granting greater authority to members, scrum master fostered loyalty and provided more effective solutions. The members were offered flexibility in terms of the task division and development. Any changes in requirements were handled through proper discussion with the members. The feedbacks from the members were properly evaluated and integrated. To summarize, the leadership style relied on three C's: Communication, Collaboration, and Commitment. Similarly, looking at the member's styles of contribution, it closely relates to McGregor's Participation Theory. This theory suggests that People pursue organizational goals with self-discipline and self-direction, absent external control or the prospect of penalty.

Team Communication

Agile is a framework which solely relies on communication. In case of Crowdly, communication with team members played a vital role to develop the project. Team Charliezz consisted of five individuals. The entire team was able to put forth effort by working together through effective communication, which ultimately contributed to the product's success. Every team member was invited to the meeting in case there was an interruption while the product was being developed, and they all shared their opinions and suggestions on how to fix the problem. However, not every person was skilled in every area. They all possessed unique talents in their respective fields. For instance, some of them excelled at communication, document writing, coding, and other skills. As a result, the person who was skilled in a particular skill assisted the member who lacked it. One of our team members was given the title of Product Owner (Scrum Master) after the team's core competencies were examined in order to assess his ability to lead the entire team. As a result, the scrum master was essential to the successful completion of our complete product because of his ability to oversee everyone on the team.

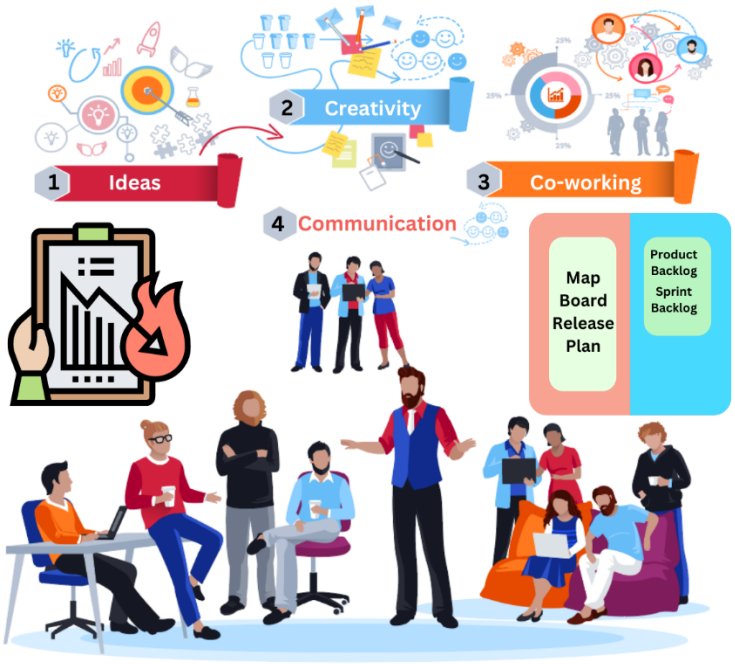


Figure 10: Team Communication

In addition to this, since the product owner and the developers communicated often, the product owner received regular updates regarding the product generated by the developer, allowing the product owner to request changes or edits as needed. Likewise, managing time for the frequent completion of specific tasks and the need for adjustments makes working in an agile team a burden, our team members chose to hold face-to-face meetings rather than virtual ones to better understand the problems that each member was experiencing. The product must be presented to the product owner every week as part of the agile methodology's weekly sprint reviews. Since documentation is not given priority, it emphasizes practicality over theory. Similarly, every member of our team was kind and modest toward one another, everyone agreed if someone recommended adding or removing

product-related functions. The above words make it clear that every member of our team was totally dedicated to the project's aim and that everyone was aware of their own responsibilities. Additionally, the team worked together continuously to advance the project, and everyone contributed equally to finding solutions to the issues that sprang up. Since these are all parts of the GRIP communication model, it is simple to declare that our team adhered to this model.

Each team member had no prior knowledge of the agile approach. We all formed an agile group and worked with an agile workflow for the first time during this time. As a result, we first encountered some issues; nevertheless, we were able to fix them. In the first place, we had no idea how to start the project. As a result, we built a sprint backlog on an excel sheet where tasks were separated and assigned to each group member in accordance with their weight. Prior to the task division, our scrum master scheduled an hour-long meeting. The gathering, often referred to as a sprint planning meeting, was held to manage an Excel spreadsheet. In addition, we handled the Trello board, which was broken up into 5 pieces, for further polishing the sprint works. First, all the user stories that needed to be implemented in the project were included in the product backlog. Second, user stories from the section of the sprint backlog that need to be completed during that specific sprint were selected. Thirdly, the sprint backlog item was moved from the starting section to the doing section and from the doing section to the verify section, respectively, after it had been completed. After being evaluated by our supervisor, every user story that was a part of the sprint backlog was ultimately moved to the verify area. Likewise, although, tiny pieces of projects were provided to the product owner each week as a part of the sprint review, which made it simpler for our development team to work on improvements immediately. The sprint review was a part of the iterative process known as "sprinting."

Finally, in regard to conflict management, if any one of the members had health conditions or personal issues, then the workload was recalculated and divided among the members. Moreover, problems confrontation was another technique employed by team Charliezz. As an example, when discussing the frontend portion, everyone had their own thoughts and ideas, and we were unsure of which to apply. Therefore, the front-end design of the project was selected in accordance with the preferences of the majority of members. Despite disagreements among members, occasionally, everyone ultimately compromised on what was best for the team. Each team member was committed to finding a solution, and frequent collaboration and work division made it easier to manage the project because no one person was under pressure to find a solution. To summarize, this is how our team resolved the dispute that emerged during the project: Problem confrontation, open communication, compromise, and equal workload division.

Conclusion

In conclusion, the product, Crowdly, was developed following agile methodology, scrum to be particular. Development of project through scrum framework enabled for a flexible development process, promoted communication, enabled efficient time management, and helped in effective development. The team was self-motivated and were in constant communication. Throughout the sprint, the team communicated through different scrum artifacts such as sprint backlog, product backlogs, Trello, and so on. This level of communication helped to build trust, and sense of membership into the team and contributed to overall product development.

Bibliography

Peek, S. (n.d.) *What Is Agile Scrum Methodology?* [online]available from https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html [8 January 2023]

Diogo, S.E. (2022) *Top 5 Main Agile Methodologies: Advantages and Disadvantages* [online]available from https://www.xpand-it.com/blog/top-5-agile-methodologies/ [8 January 2023]

Anon. (n.d.) *What Is Scrum?* [online]available from https://www.scrum.org/resources/what-is-scrum [8 January 2023]

Hampson, T.E. (2020) *Why Choose Scrum? (10 Great Benefits)* [online]available from https://scrummastercertification.ie/why-choose-scrum/ [8 January 2023]

Agilemania, E.P. (2022) *How Can We Resolves Conflicts in Agile?* [online]available from https://agilemania.com/conflict-resolution-techniques/#:~:text=The%20most%20prominent%20way%20to,conflict%20in%20the%20right%20manner. [8 January 2023]

Johnson, E.P. (2022) *15 Positive Team Behaviours That Boost Performance | Indeed.com UK* [online]available from https://uk.indeed.com/career-advice/career-development/team-behaviours [8 January 2023]

Bozickovic, A. (2022) *Modern Leadership Styles - What Does It Mean to Be a Truly "Agile" Leader?* [online]available from https://www.servicedeskinstitute.com/modern-leadership-styles-what-does-it-mean-to-be-a-truly-agile-leader/#:~:text=The%20agile%20leadership%20style%20works,Supports%20customer%20satisfaction. [8 January 2023]

Business, A. (n.d.) *The Nine Principles of Agile Leadership* [online]available from https://www.agilebusiness.org/resource/the-nine-principles-of-agile-leadership.html [8 January 2023]