Devops Team Project

On

Scrum Lab

Members:

Ibegbu Deborah AS (Product owner)

Adeshina O'tayo AS (Scrum master)

Akshat Katre AS (Deverloper1)

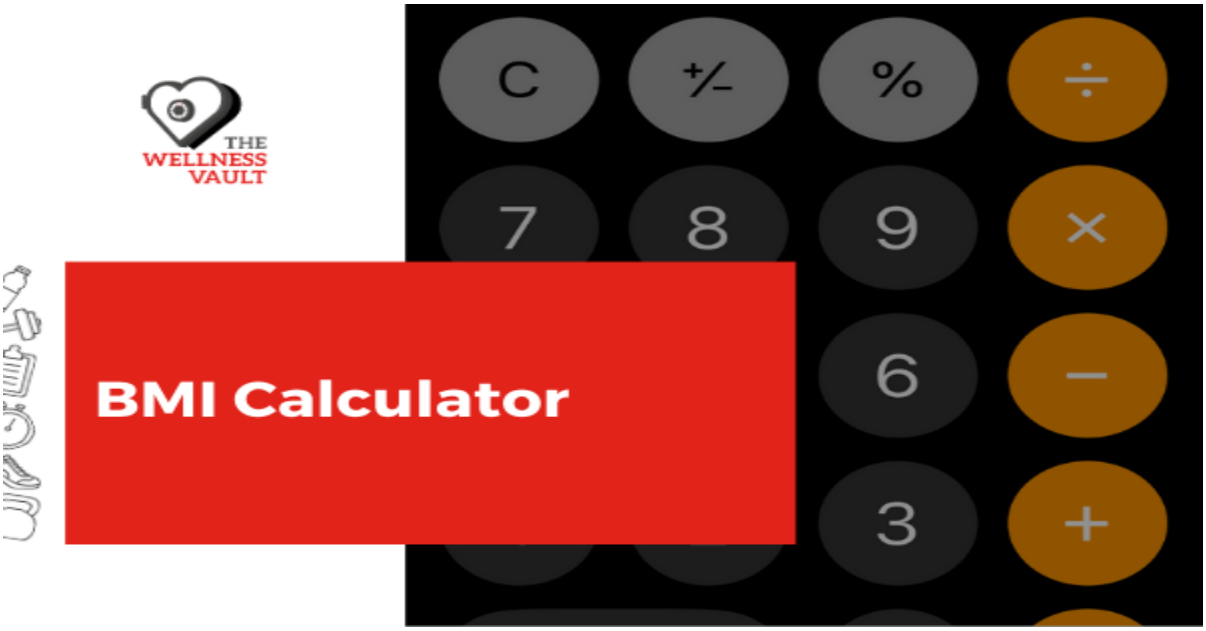
Oluwasegun Kadiri AS (Developer2)

INTRODUCTION

After going through the project carefully we had to come up with a mini project to enable us to identify our roles. During time this we create and divided ourselves into teams and here are how the division of labour was done.

We also reschedule meetings upon meetings in order for use to get to a final destination by creating roles within ourselves.

Her is the summary of the project we partook in and its final result given according to the role of this course.



This project works is based on a basic health and fitness systems (bmi) which was developed through the Android platform using also other good application such as NetBeans with java and the features will be further described in this document. The system aims to help users to keep track of their exercise progress and also allow them to measure their body condition with the health calculators provided in the application. At the end of the development of the system, there will be the delivery of an application which works as a health and fitness system.

OBJECTIVES

The main purpose of this health monitoring system from this project is to enable users to briefly keep track of their exercise progress and let them to calculate their body fitness. Hence, the main objective of the project is to help users to briefly keep track and let them know about their body fitness. Basically, the application can perform the following functions:

1. Accepting body weight

2. Accepting date of birth of user

3. Accepting height of user

4. Give results (the bmi)

5. It should tell user if they are (under weight, normal weight, overweight, obese)

TEAM NAMES WITH ROLES PLAYED

Figure 1(teams and role played)

Team IDs	Names	Role
1	Ibegbu Deborah	Product Owner
2	Adeshina O'tayo	Scrum Master
3	Akshat Katre	Developer1
4	Oluwasegun Kadiri	Developers2

Product Owner

The product owner took the lead in many areas of product development with need to access the deep well of market knowledge to strategize and present their vision to stakeholders. Another day they will need to roll up their developer sleeves to help the team meet their goals during a sprint. In this case we came up with a project called the BMI calculator,

We as team where able to put head together and make such the deliverables were done in due so it gets to hand of our customer.

Here are few responsibilities (product owner)played during the task given as a role given by the team.

1. Defining the vision

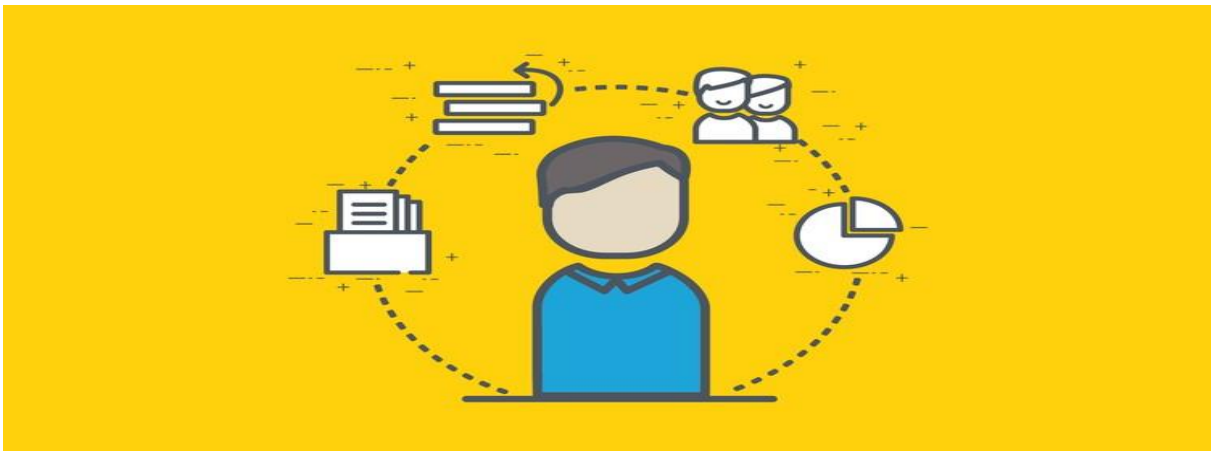
The agile product owner is the point person on the product development team, using their high-level perspective to define goals and create a vision for development projects. Hence, I was able to put it into documentation and make sure that it was easily digested before the project began, the creation of the following.

a. SYSTEM DESIGN

System design

Weight Categories	BMI (kg/m²)
Underweight	< 18.5
Healthy Weight	18.5-24.9
Overweight	25-29.9
Obese	30-34.9
Severely Obese	35-39.9
Morbidly Obese	≥40

For the BMI calculator in the application, metric measure is used, using centimeters and kilograms simply because the formula is slightly simpler than using the standard one. The formulas to calculate the BMI are as following: Metric: BMI = Masskg / (Heightm) 2 Standard: BMI = Masslb / (Heightin) 2 * 703 After calculating the BMI value, it is used to compare with a BMI table to determine the weight category that you are in. The table is as following:



2. Managing the product backlog

One of the most important responsibilities we did was the managing the product backlog. This is the development team's project to-do list.

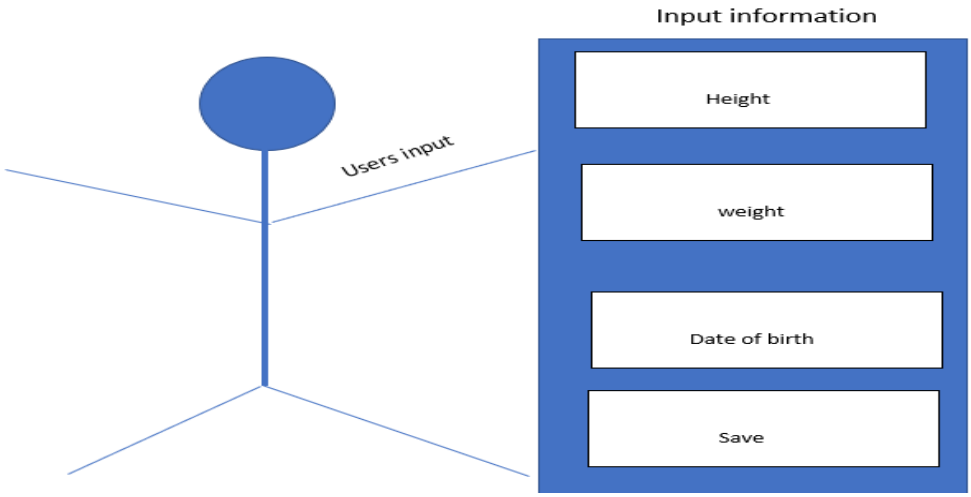
The product owner's responsibility is to create the list of backlog items and prioritize them based on the overall strategy and business objectives. Additionally, the product owner will need to map out project dependencies to inform the necessary sequence of development.

3. Prioritizing needs

Another key role we played was to prioritize needs. In other words, they must juggle the triangle of scope, budget, and time, weighing priorities according to the needs and objectives of stakeholders.

4. Overseeing development stages

With the vision, strategy, and product priorities set, we were able to spend a significant amount of time overseeing the actual development of the product. They are a key player throughout each event, including planning, refinement, review, and sprint. Such as using the UML diagram to further explain the project.



During the planning stages, the agile product owner works with stakeholders to identify and organize the steps required for the next iteration. They will then meet with their team to refine the process, identify areas for improvement, and support the sprint.

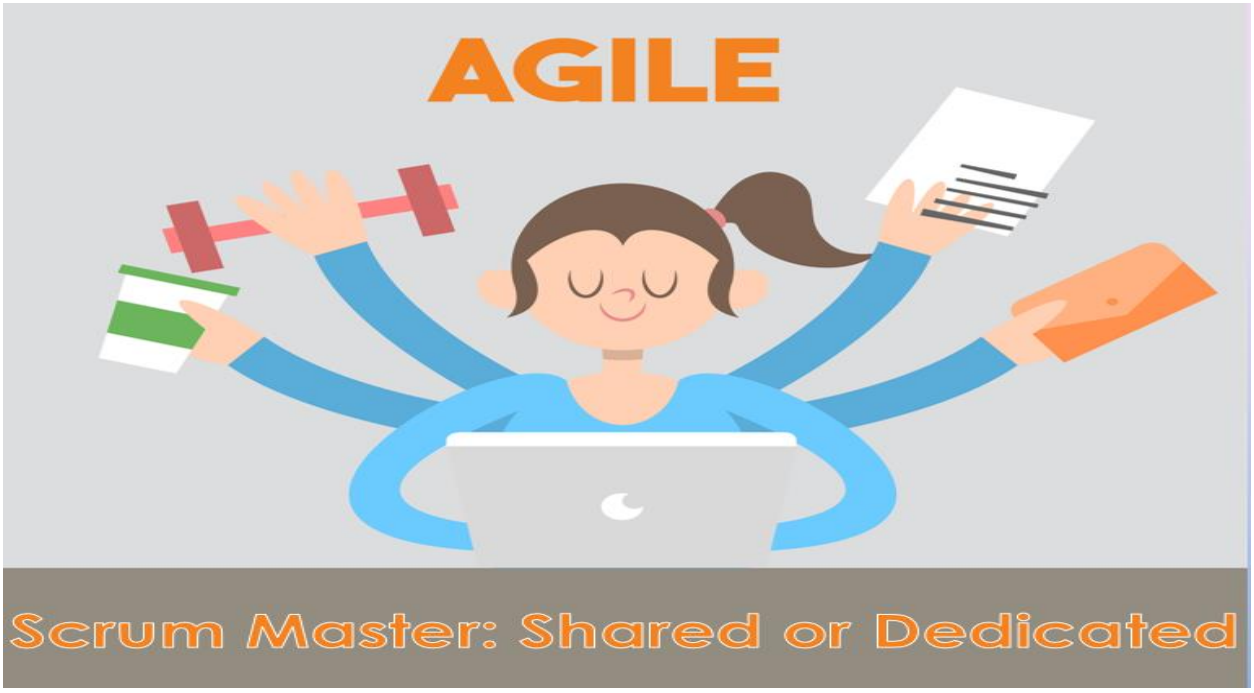
5. Anticipating client needs

I as a successful product owner was able to be an expert at understanding and anticipating the client's needs to more effectively manage the development process.

Their deep market knowledge and communication skills allow them to anticipate problems or needs and address them.

Scrum Master

The **scrum master** is the team role responsible for ensuring the team lives agile values and principles and follows the processes and practices that the team agreed they would use.



The responsibilities of this role include:

1. Clearing obstacles: being able to make sure all obstacles such planning, meetings etc. was not encountered, the course of the project as team the framework pf the project was drawn out by the scrum master, below is the framework.

Figure 1.1

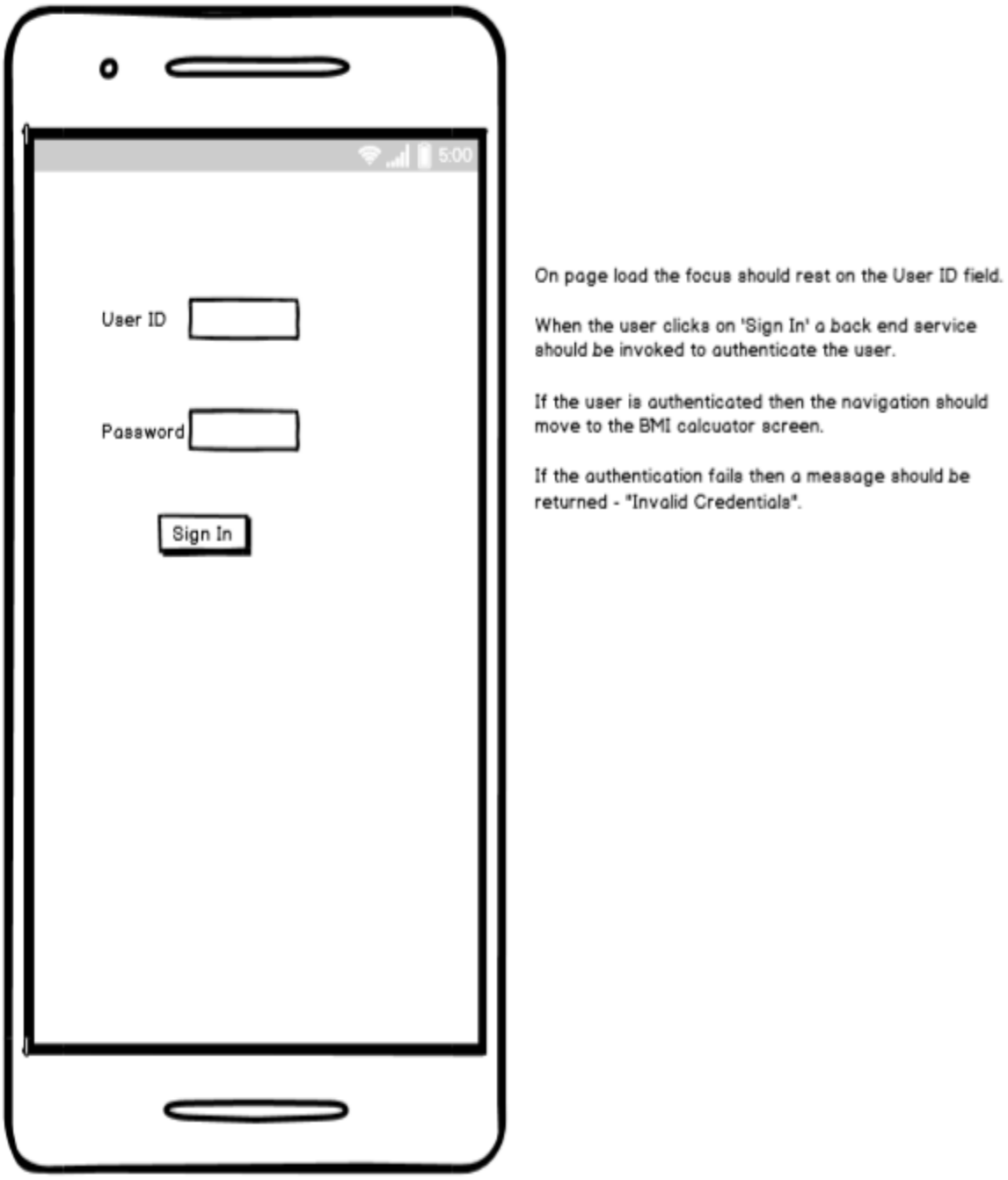
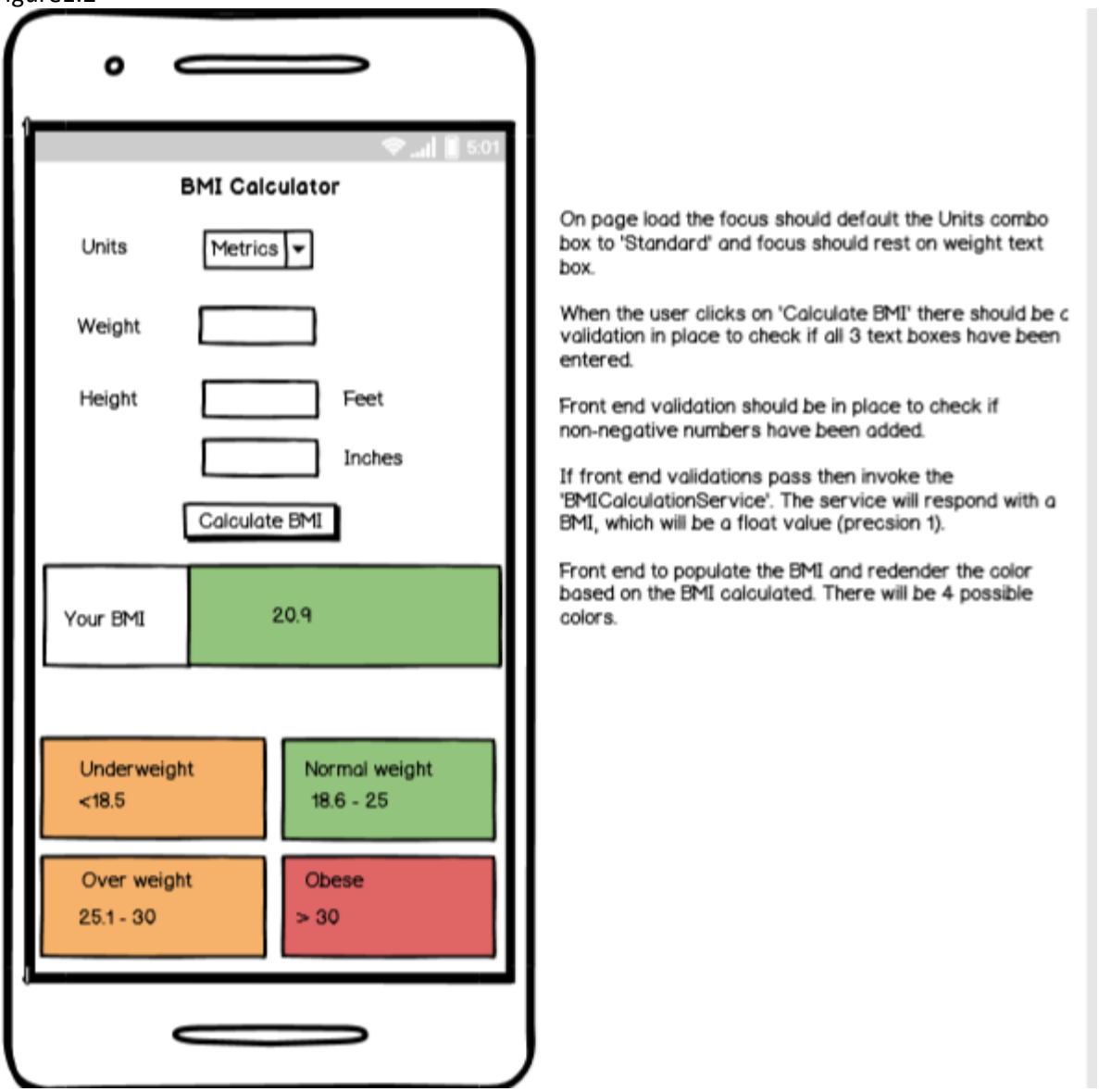


Figure1.2



2. Being able to establish an environment where the team can be effective such as
   - Knowing my **team**.
   - Tackle problems quickly with good feedback.
   - Define roles and responsibilities.
   - Break down barriers.
   - Focus on communication.
   - Pay attention.

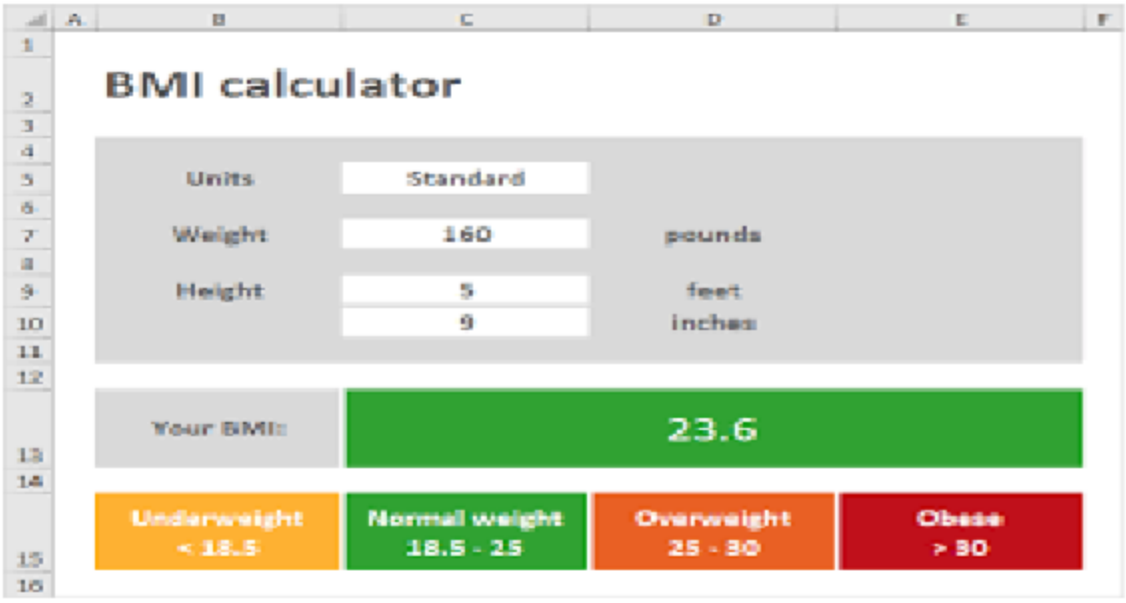
3. Ensuring a good relationship between the team and product owner as well as others outside the team. But I really enjoyed
4. Protecting the team from outside interruptions and distractions.

The scrum master role was created as part of the Scrum framework.
Finally, I really enjoyed Worked closely with the product owner in arranging the product backlog. Ensure the goal, scope and product domain are understood by the team. Also helped in removing impediments to the development team's progress.

Developers

The role of a developer in this project environment is significantly different. Developers have additional responsibilities in an Agile environment that go beyond simply writing code. recently we figured out a situation where the development team was completing tasks required for a sprint, but the overall stories were not being completed. The team allocated development tasks to individual developers and also allocated overall responsibility for each story to individual developers. However, for several sprints, the development tasks were completed with no stories.

the most seen primary responsibility as completing development tasks. Completing stories was seen as kind of an administrative coordination function. It's more than that, in our opinion – as a developer who takes responsibility for stories and project completion in a sprint needs responsible for: in our case the developer who work hand in hand with the rest of the team was able to deliver as seen in the image below



System design

This leads to the key responsibility we figured out for the developer during the cause of this project development.

I. Understanding the business purpose of the story and defining and analyzing possible alternative ways of satisfying the business purpose of the story
II. Planning and estimating development tasks with other developers that are required to fulfill the story
III. Working directly with the Product Owner to clarify and further define the details of how the story should be implemented
IV. Providing guidance to other developers as necessary who are engaged in development tasks associated with the story

All these were very challenging but yet it was fun to deal with.

In conclusion the project we picked, it deliverable which is an application for the handheld device that works as a Health and Fitness application. The main function of the application is to keep track of user's workout and allow users to send out emergency message if they are not feeling well and knowing that they are going to break down in the next moment. This could be able to save an individual's life if there is no one nearby the area, external help is useful.

It was a thing of joy to work as team members and product such project for the satisfaction of the end consumers.

Section 2 of the project (questions and answers)

Questions:

1. What are the properties of software programming that makes it particularly suitable for the Agile methodology?

> In Agile Methodology, units of delivery are accomplished via releases and iterations. A release consists of several iterations each of which is like a micro-project of its own. One of the properties of software programming especially as pertained to Object Oriented Programming that makes it particularly suitable for Agile Methodology is Encapsulation. Encapsulation is one of the fundamental concepts in object-oriented programming (OOP). It describes the idea of bundling data and methods that work on that data within one-unit e.g a class in Java. This concept is also often used to hide the internal representation, or state, of an object from the outside. With this Agile teams can work on Modular features and self-containing units of the project.

Hence Software programing can be quite abstract and in traditional IT projects we get to see the working/demo product at the end of a very long wait - usually months or years. Agile allows for breakdown into small sprints/iterations that allows for demos/product releases at the end of every few weeks. This makes for an excellent feedback loop for the business stakeholders to come back with feedback and better still if release to production happens every iteration, then new features and functionality are being released in quickly.

Software programming being quite abstract makes for lot of changes while development is in progress. Agile handles changes very well - perhaps it greatest strength. This is something traditional IT projects are not able to handle… they go down the CR loop which makes for delivering changes in-flight very tough.

Agile has a codified way of working which allow all participants to focus on their areas of work. Agile methodologies give objective ways to measure software projects.

Requirements can be managed very well. Requirements go into the product backlog, can be broken into development tasks that can be used to drive sprints. Requirements can be estimated to a good degree of

accuracy. In essence, you can pick a 'T-Shirt' size anything in the backlog to for example - Small, Medium, Large, XL effort. Using this effort estimate we can choose what to pull into the sprint backlog.

The roles of participants work well for legacy software development team structures. Project managers generally fit into the Scrum master roles, the business teams usually take up the role of Product owners and the developers fit in as the development team.

2. Why is the "definition of done" a key component of the Scrum model? How would you make sure it is well defined before starting the project?

Definition of done is a key criterion as it works as objective measure of when a Sprint is complete. In projects that I worked on Definition of done has been set as all tasks on the sprint backlog have been completed. Each development task from the sprint backlog can be marked as complete only when the unit test cases associated with the sprint task have passed. The unit test cases are often documented before the development begins and the product owner on occasion reviews the UT cases. The UT case results are documented after the test execution and can be reviewed by the product owner if necessary. Tools like confluence and JIRa have been used to make collaboration easier.

Also more definition of this is a done key component of the Scrum model because it is was serves as the acceptance criteria and the benchmark of delivery in line with the expectations of the Product owner. To ensure it is well defined before starting the project all stakeholders in the project must be in alignment of the expectation, requirements, roles, responsibilities and delivery per time on the project at the starting of the project. There should be a thorough Project Initiation Meeting were all these are defined and agreed.

3. Sprint review, daily scrum etc. what are the pros and cons of strictly following such codified events?

Pros -

- If followed correctly these codified events can allow the developers to spend most time on their actual development work rather than spending most of their time in meetings. Which I have experienced in the past in some projects.

- It helps to have an overall check on all components of the project, appraise progress and resolve possible showstoppers in time
- It helps in the possibility of meeting defined targets, goals and timelines

Cons -

- In reality although these meeting are held, the spirit of Agile is rarely followed. Stand-up meetings end up being long detailed oriented meetings. Sprint reviews end up becoming very acrimonious and contentious. Most software projects are run where participants are remote. Sprints become extremely challenging in remote setups.

- It eats into the scarce resource of time for actual work for the developers
- There is a risk of the meetings becoming monotonous and unexciting

4. Why does the development team, which can be composed of multiple developers, must be seen as a single entity (both in terms of delivery and accountability)?

It is mainly to foster a team-based thinking rather than an individual based mode of thinking. The development team is a collective that delivers the sprint together. It is essential that there is an equitable distribution of sprint backlog items within the team so that there is a sustainable pace for every sprint.

> The development team though composed of multiple developers must be a single entity to ensure that no member of the development team is working in Silos. This is very critical for the overall success of the project. Working in Silos is a situation where teams work in differing units and from different perspectives in line with the collective goal of a mission. The development team must also be a single entity to ensure cohesiveness of the team and to aid delivery in line with requirement and to avoid different accountability points.

5. According to you, is it possible for a team whose members are perfectly acquainted to the Scrum methodology to perform well without a Scrum master? Would it still be "Scrum"?

No, Scrum master role is required. It would not be Scrum if there is no 'Scrum master'.
For us, in the first place it would **not** be Scrum because for it to be Scrum, there must be the Scrum Master, the Product Owner and the Development Team. What would naturally happen in a situation where a team is acquainted with the scrum methodology is one of the team members will unwittingly assume the role of the Scrum Master alongside his/her role of development and this would affect productivity over time both for the individual and for the team. This is because asides aside the Scrum Master moderating the scrum meetings daily, the Scrum Master also resolves any impediments the team has and protects the team from the product owner and any other engagement that could distract the focus of the team from its agreed tasks.

6. How can Kanban and Scrum be used together?

Kanban offers specific measures that compliment scrum. Things such as the burn-down chart, the Kanban board offer good metrics and visuals to see how the project is faring. The philosophy of trying to keep the number of works in progress tasks to the minimum works very well in reality. Based on prior experience I have found that keeping the average of WIP tasks to low is a key indicator of how sustainable the pace of the agile team can be.

> Scrum and Kanban are different popular Agile approaches; however, both have their own practices, principles, and concepts. According to the Kanban concept, a Product Owner is responsible for gathering user stories and prioritizing them in the product backlog. The team cannot influence him/her or reprioritize stories. In modern Kanban teams, digital Kanban boards usually visualize the work. Their columns depict all stages of task performance and convenient Kanban cards include all required information about each task. The functionality of modern Kanban project management applications allows the developers to visualize their workflow in various forms, including charts, tables, and diagrams. It simplifies the work of the team and allows its members to forecast the results of their activities. The principle of work in progress limitation is also applied to modern software development. Kanban teams usually limit the number of tasks that are placed on the Kanban board at the same time.

That allows them to perform all tasks more effectively and to focus on the quality of work. Scrum methodology is the most widely spread Agile methodology that combines the best Agile practices. Scrum teams are able to achieve the best outcomes when they use this methodology properly. The method is based on iterative cycles, called sprints. Each sprint lasts 1-4 weeks and has its own backlog. The sprint backlog is the number of tasks that should be performed during a certain sprint. The testing procedure is conducted after each sprint. That allows the Scrum teams to detect all defects in the early stages of work and to remove them before the product is delivered to the customer.

So, in combining the scrum and Kanban approaches which is the Scrum ban concept, practices from the scrum approach can be aligned with practices from the Kanban approach to deliver the greatest benefit in the delivery of the project. Scrum will provide the team with a complete sprint to decide how to manage itself before the next stage while Kanban carefully observes the process through the team every time a new story is started. This has a large implication on the way that managers (masters), product owners and developers work combined. With Kanban, the team can adapt the work (its progress level) dynamically to avoid free or over-worked developers, and that is something that can be monitored by the management. Both Scrum and Kanban prosper from well-defined user stories that can be accomplished and delivered separately, and Scrum leaves the decision to the team – how much to work on the same time while Kanban focuses on limiting the number of items the team has in progress.

A key difference factor is that Scrum measures the levels of teams' productivity in terms of the team's speed, and controls how many agile points a team will score during a sprint. Kanban controls for continuous work in progress and limits the number of stories the team will accept to working on at the same time.

7. What should you do if the Product Owner does not do his job properly (ie: does not validate sprints, is not available for meetings, etc.)?

It is the scrum master's role to ensure that all members of the team are executing their role as per the defined guidelines. If the product owner is not doing what they are supposed to doing the scrum master needs to talk to the product owner and let them know clearly where they are lacking. If the Product Owner does not correct themselves then the scrum master needs to escalate to Management/Stakeholders. Escalation should happen only if the Product Owner does not correct themselves.

If the Product Owner does do his job properly, there are a few things that could be done. The first is to organize a meeting with him alongside the other stakeholders of the project and bring his negligence to his notice with the aim of correction. In that meeting emphasize on the impact of his not doing his job well can have on the project, on the success of the Sprints and on the morale of the team. In the case this persists, document this behavior and share with appropriate stakeholders. Also emphasize the non-availability of the Product owner can cause the project being prolonged unnecessarily and the resources assigned to the project can be distributed to other courses.

8. What is a stakeholder in a Scrum project?

A stakeholder is a person or persons how are directly affected by the outcome of the Scrum project. Usually this is the project sponsor and/or end users.

> A stakeholder is anyone that has a stake in a scrum project. By a stake we mean, anyone that has a reason for the project to hold and accomplish its intended cause. The Stakeholders can range from the Executive Sponsor, to Scrum Master, Product Owner, The Development Team, Quality Assurance Analysts, Control Team etc.

9. As a Product Owner, how would you deal with a missed sprint?

If there is work left at the end of sprint the Product owner can move those items to the product backlog. In the sprint review meeting try to identify the root cause of missing the sprint. Sometimes this could be due to improper estimation of the items on the backlog… there could be numerous reasons, that is why retrospective meetings can be key.

> To deal with a missed sprint as a product owner, the first thing to be done is to ascertain from the team why the sprint was missed. This is very crucial as it could be circumstances beyond the control of the team, e.g. natural disasters, prolonged sickness of team members and it could be circumstances that could have been contained or avoided. After ascertaining the cause of the Sprint, revaluate and rescope with the team how to maximise the other sprints to ensure the final estimated go live date of the project is not affected.

10. How would you convince a doubtful IT project manager that he should consider applying Scrum?

The two biggest problems with traditional IT projects are two fold - firstly, we get to see a working product/release only at the very end of the project, which at times could be months and in some cases years from when the work actually started on the project. The second issue is how to deal with changing requirements during in-flight projects. We have seen these going into CR loops that have at time derailed projects.

I would highlight how Scrum is able to deliver a releasable/demonstrable project at the end of ever sprint. This also allows for the business users to look at what has been developed so far and can get a taste of what is to come very quickly - in effect every few weeks.

Scrum deals with changes very well. If there is anything new that needs to be added it can go into the product backlog, and based on the product owners priority it can be added into the next sprint, which make for team to be very responsive to the business changing needs.
There are codified methods on when teams should meet and how they should conduct specific meetings, this allows the teams to focus on work, rather than go and waste time playing meetings.
Scrum allows for good visualizations and measure to identify how the sprint is working.

> But, the best way to convince a doubtful IT Project Manager to apply Scrum is to amplify the benefits of using scrum to him. Emphasize to the IT Project Manager that with Scrum there is REVENUE Realization. Using Scrum, new features are developed incrementally in Sprints. At the end of each Sprint a potentially releasable Increment of Software/Product is available. This enables the product to potentially be released much earlier in the Development cycle enabling benefits to

be realised earlier than otherwise may have been possible where we waited for the entire Product to be “complete” before a release.

Furthermore, there is possibility of Quality Work. Testing occurs every Sprint, enabling regular inspection of the working Product as it develops. This allows the Scrum team early visibility of any quality issues and allows them to adjust if necessary. Scrum also allows for flexibility and Agility, Business Engagement and customer satisfaction and Speed to Market.

References

- https://resources.collab.net/agile-101/agile-development-success
- https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/
- https://www.scrum.org/forum/scrum-forum
- https://www.thescrummaster.co.uk/scrum/benefits-scrum-agile/
- https://hygger.io/blog/combining-scrum-and-kanban-the-best-of-both-worlds/
- [Agile manifesto](https://agilemanifesto.org/principles.html)
- - [Kanban guide](https://scrumorg-website-prod.s3.amazonaws.com/drupal/2019-11/2019-09-Kanban-Guide-for-Scrum-Teams-English.pdf)
- - [Scrum guide](https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf)