Statement and Confirmation of Own Work

***A signed copy of this form must be submitted with every assignment.***

***If the statement is missing your work may not be marked.***

**Student Declaration**

I confirm the following details:

|  |  |
| --- | --- |
| **Candidate Name:** | Shihab Mirza |
| **Candidate ID Number:** | P00190603 |
| **Qualification:** | NCC-L5DC |
| **Unit:** | Agile Development (AD) (20 Credit) |
| **Centre:** | ZCAS University |
| I have read and understood both NCC Education’s *Academic Misconduct Policy* and the *Referencing and Bibliographies* document. To the best of my knowledge my work has been accurately referenced and all sources cited correctly.  I confirm that this is my own work and that I have not colluded or plagiarised any part of it. | |
| **Candidate Signature:** |  |
| **Date:** | 01/10/2022 |



**OPS020\_Candidate Statement of Own Work**

**TASK--1**

1. **8 principles of DSDM and ways of how each can be fulfilled.**

* Focusing on the needs of the business

The foundation of any successful project is always a solid business case that is both plausible and realistic. Your organization's strategy and your long-term objectives should be supported by the project.

The Dynamic Systems Development Method (DSDM) makes use of tools like the MOSCOW (Must have, Should have, Could have, and Will Not Have) prioritization method to ensure that your project remains relevant and serves a purpose.

* Delivering on time

In order to build trust between the client and the project team, it is essential to deliver results on time. Timeboxing methods can help you meet deadlines more often.

Timeboxing is an extremely straightforward idea: You prioritize your tasks by setting deadlines. For instance, in order to plan tasks more effectively, determine in advance how much time you will devote to a meeting and each of its topics.

* Collaboration

Collaboration and cooperation are crucial in project management. The team should work together to achieve a common objective. The project team should be able to rely on each other to make decisions quickly.

It is necessary to work together with all stakeholders: In order to gather feedback throughout the development process, particularly from your client, make sure to include all relevant individuals.

* Never compromising quality

Quality should not be viewed as a criterion for accommodating unpredictability. Upholding it ought to be a constant standard. Throughout the project, everyone should agree on and adhere to quality standards.

Up front, conduct sufficient analysis and layout (EDUF) to establish solid foundations. Reassess the ongoing project's viability informally and formally with each delivered increment.

The Dynamic Systems Development Method (DSDM) places a strong emphasis on ongoing improvement: Quality cannot be evaluated only at the conclusion of the development cycle or process.

* Incrementally building from firm foundations

Similar to Scrum and other agile methodologies, DSDM targets on providing the value sooner and building up on previous progress at each of the steps. At the outset of the process, carefully defining the task's scope and project requirements ensures that everyone is working toward the same unified goal.

By releasing new features early, you can avoid wasting so much time on details and ensure that the project is always moving in the right direction.

* Developing Iteratively

It's recommended to use iterative development: Throughout the project, new features should be published and evaluated before being modified in subsequent releases based on client and tester feedback.

Experimental research is urged, as well as value is added with each release.

* Incorporate business responses into each iteration.
* Recognize that the majority of the details will emerge later instead of sooner.
* Accept change; without it, the correct solution will not emerge.
* Encourage creativity, experimentation, and learning through iterative development.
* Clear and Continuous communication

Stakeholder communication can either make or break a project. You have control over their expectations and can make sure that nothing stands in the way.

The project team's communication is also very important: IN order to encourage informal interaction, DSDM advocates for daily stand-up conferences, straightforward documentation, and workshops.

* Demonstration of-Control

Using pertinent and precise Key Performance Indicators (KPIs), you must be able to track and follow the progress of your project in order to maintain control. Be on the lookout by keeping everyone informed about the team's progress and reviewing what went wrong to learn or discuss best practices.

* Make plans and advancement visible to everyone.
* Measure progress by focusing on product delivery rather than completed activities.
* Proactive management
* Based on the company's objectives, assess the ongoing project viability.
* Tracking and reporting should be done with the appropriate level of formality.

1. **Two Instrumental success factors and how they could be met prior to the beginning of a project using DSDM.**
2. Acceptance of the philosophy that DSDM strives to entail, before starting work.

The following is some background on the DSDM philosophy: The best business value emerges when projects are linked to clear business objectives, produce frequently, and involve the cooperation of motivated and equipped individuals.

The DSDM project approach must be accepted and comprehended by all participants and stakeholders. In addition to adhering to the DSDM philosophy that projects produce the best business value when they are associated with clear business objectives, deliver frequently, and involve the cooperation of energized and empowered individuals, the project may or may deliver less than 100% of the potential solution(s).

This factor should be met prior to the beginning of a project because;

It will help stakeholders:

• Recognize and back the company's vision and objectives.

• are given the authority to make decisions in their area of expertise

• Collaborate to provide a business solution that is appropriate for the task at hand

• Work together to meet agreed-upon deadlines while keeping the priorities of the company in mind.

• Be aware that as knowledge of the solution grows over time, change cannot be avoided.

• Anyone involved in or affected by the project, whether within or outside of it, is a stakeholder.

1. The Appropriate empowerment and encouragement of the Solution Development Team.

The Solutions Development Team as a whole should be given the authority to make decisions within the parameters that were agreed upon during the foundations, and each position on the team must be given the authority to make decisions based on their expertise.

As a result, other Solutions Development Team members must be given the authority to make comprehensive daily decisions about how the solution should be built and tested.

Within the business roles, senior executives (Business Sponsor, Business Visionary) must agree to delegate day-to-day judgment to the Business Ambassador of the Solution Development Team. They will have to join the team if this is not possible. If business decisions are made somewhere else and at a different time, team advancement will slow down.

**TASK--2**

1. **Facilitated workshop with an outline of the facilitator and scribe roles.**

A place where a specific task is completed and a product is produced is known as a facilitated workshop. a method for gathering information and making decisions that is based on teams. An independent facilitator, empowered staff, and interactive communication

As organizations and information have become increasingly complex, this tactic has in fact been widely used, particularly to get more people to buy into new ways of thinking. Employee interactions and actions frequently determine an organization's success or failure. It is now less effective and less common to understand or influence through the exercise of hierarchical power than through consultation and direct relationships. Therefore, teaching people how to interact effectively in groups yields significant benefits. It has been demonstrated that facilitation and Facilitated Workshops are a productive and efficient method for quickly achieving better buy-in to decisions, problem-solving, idea generation, and action planning. This has been the case for so many years. Below are some of the points to consider when executing a workshop:

• Definition (owner, merchandise, and participants);

• Get ready for the workshop (an agenda with goals); Start on time, respect others' points of view, silence equals agreement, only one speaker at a time, follow the workshop's direction, adhere to the five-minute rule, and lead the workshop session;

• Assess the workshop;

• The workshop report, the final document;

• Discuss with the workshop owner

The process and dynamics of the workshop are managed by the ***facilitator***, allowing participants to concentrate on the content and outcomes. The goals, deliverables (outcomes), and participants of the workshop should not affect the facilitator's impartiality. It is their responsibility to assist the group in achieving the Workshop goals. To ensure impartiality and to represent the project, the Workshop Facilitator should come from outside the organization. Workshop facilitators are assigned by some organizations internally, while others employ consultants from the outside. It is essential that the group believes that the Workshop Facilitator's actions are dependent on the outcome, even if they are part of the project.

The workshop's ***scribe*** should produce and distribute the output as soon as possible—usually within two days—to all Workshop Attendees and, if appropriate, any other interested parties whose lives will be impacted by the output. This document ought to be brief and straightforward, and it ought to contain the following data:

* Decisions
* Actions involving action owners
* Unresolved issues
* The Workshop's own output, as appropriate
* And, on occasion, the procedure used
* It does not hold minutes or verbatim records.

1. **Risk logs and how they can aid the scenario given and how the use of the PAQ might help reduce risk early**

An information radiator known as a risk log is used to make the project's risks clear to the team and stakeholders. The probabilities and effects of the identified risks are displayed using a risk log. The members of the agile team and the project manager should regularly review this log in stand-up meetings. It is an essential component of any methodology for project management. This gives you a way to keep track of the risks you've found, how bad they are, and what management steps need to be taken.

An example risk log for the scenario given will be:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk ID | RISK (Description) | Likelihood  3 - very likely  2 - likely  1 - least likely | Cost or Impact | Timescale (weeks spent to mitigate issue) | Management approach |
| R1 | The website won’t make a difference in terms of sales | 1 | The clients will suffer a loss | Approx. 8 weeks spent on adverts. | Advertisement will be done. |
| R2 | The website will use more hosting resources than planned | 2 | The team will have to spend time optimizing the website | Approx. 4 weeks spent on optimizations. | Website optimizations will have to be done. |
| R3 | The company fails to pay for the website | 3 | The team will suffer a loss | Unknown. | The team will have to try and find ways of getting the money they are owed. |
| R4 | The stakeholders are not fully appreciative of the website | 2 | Trust in the team starts to weaken | 2 weeks spent on fixing issues | The team will have to make changes to the website |
| R5 | The website has many bugs on release. | 2 | The team will have to fix bugs | 1 week spent on fixing bugs | The team will be fixing bugs. |

***Note: There could be more or lesser columns. This is just an example…***

To determine which aspects of a project or its environment are not suitable for the DSDM method, the Project Approach Questionnaire (PAQ) is used. It can be used to negotiate changes that reduce risk and boost chances of success. When changes cannot be made quickly or easily or if too many changes need to be made at once, the PAQ can be a helpful guide for customizing DSDM to meet the needs of each project. If, upon collaboratively completing the PAQ, everyone either strongly agrees or agrees with every statement, the risk of using DSDM to manage the project is low. According to the first sections of this assignment, DSDM ought to work well, so tailoring probably is not necessary.

**TASK--3**

A gamified form of project management task guessing known as "planning poker," "scrum poker," and "pointing poker" is used by development teams. These estimations are more engaging and accurate than other methods because they are based on the input and consensus of the entire group. Teams use planning poker cards, which look like poker cards, to figure out how many stories points each task requires.

Planning Poker aims to encourage teamwork and collaboration across the board. Planning Poker makes it simple to estimate significant amounts of time and effort in order for a team to produce high-quality outputs. Members of the team communicate their estimates with a deck of playing cards rather than verbally. By simultaneously drawing cards and placing them face down, bias is eliminated. During the estimation process, everyone follows this path, which encourages individual estimations and eliminates peer influence.

An agile user story is read aloud and reviewed by the product owner or customer at the beginning of a poker planning session. A user story is a general and informal description of a software feature that explains how the end user (the client) will benefit from it.

We may run a planning poker in a workshop in the following way/s:

The entire tech team is involved in poker planning. The meeting will take between two and four hours, and you will need a big table at which the team can sit with their decks of cards.

Use the method outlined below for each of the user story that needs to be estimated:

* A user story is introduced by the Product Owner.
* The user story is the topic of discussion among the team.
* In order to represent their individual estimate, each person places a card down.
* Estimates are shown and cards are turned over simultaneously.
* The members of the team that has the highest and lowest of the estimates are then asked to back up their responses.
* Now that the estimates have converged, the team plays again.
* If everyone on the team is happy with the estimate after a few rounds, you can stop.

The roles that should be involved in this process are: Product Owners or Customers, Scrum Master, Scrum team members, Estimators.

Because they are based on the input and consensus of the entire group, group estimates are more engaging and accurate than other methods. Estimating as a team enables to discuss, plan, and comprehend the upcoming features on which they will work. Estimations can also assist in breaking work down and incorporating it into the subsequent sprint. A quick measure of the team's velocity—the amount of the completed work produced by the teams each sprint—can be then obtained/gleaned from a record of this. The team can create a release plan and identify productivity peaks and troughs with the velocity information. In general, people underestimate the amount of time it will take to complete a task, but they are better at estimating the difficulty of a feature in relation to previous work the team has completed.

Prioritizing which project requirements will yield the greatest return on investment (ROI) is the goal of the four-step MOSCOW method. The letters "must have," "should have," "could have," and "will not have" were added to MOSCOW to make it easier to pronounce.

The MOSCOW method is utilized in a wide range of business fields. It lets everyone involved in a project know what needs to be done first and how it will help increase revenue, cut operational costs, increase productivity, or make customers happier. When choosing a software vendor, stakeholders can use it to help frame discussions about the significance of particular product features. The MOSCOW method helps project teams prioritize story points, which is an important part of Agile project management from an information technology (IT) perspective.

**Must Have**. All of the requirements necessary for the project's successful completion are included in this first category. The minimum usable subset (MUST) of requirements should be provided by these non-negotiable elements.

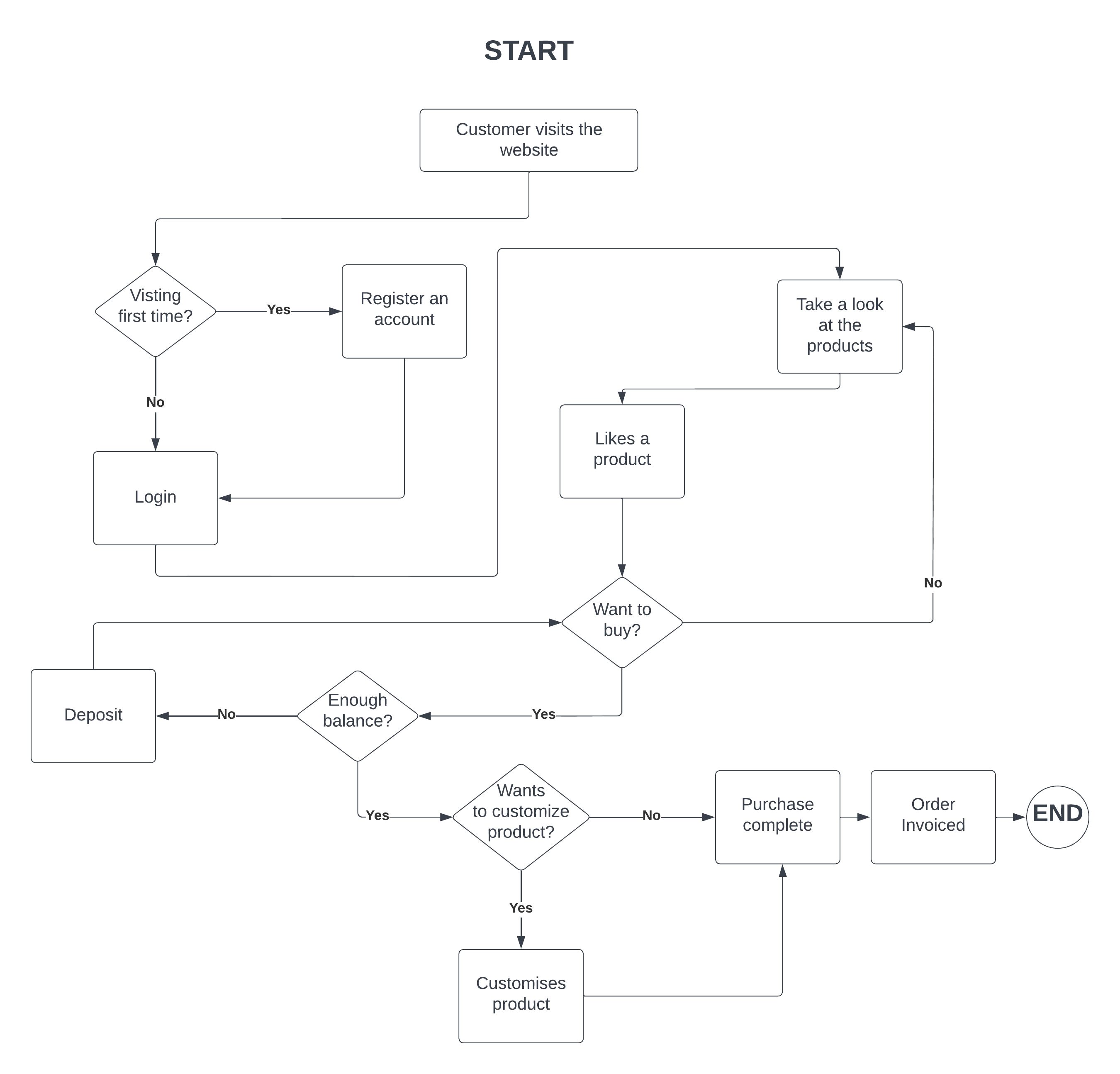
**Should Have**. A step below "must have" is this second category of requirements; It doesn't affect the current project and can be used to prepare requirements for a future release. The completion of a project depends on should-have elements, but they are not required.

**Could have**. Requirements that are not included in the project have a much smaller impact under this category. Should-have and must-have requirements will always take precedence, so could-have requirements are frequently the first to be prioritized.

**Will not have**. All of the requirements that have been identified as not being a priority for the project's timeframe are included in this final category.

**TASK--4**

Activity diagram model for a customer:



The customer starts by visiting the website on a browser. Then the website will ask the customer to log in to their account. If they do not have and account, they will now register an account and they will be redirected to the login page after successful registration. The customer will then login and will be shown a variety of products on the website, the customer if likes a product and wants to buy it, they will be able to do so, if they have enough money balance, if not, they will be prompted to deposit some funds into their account in order to buy. The customer after clicking buy will be prompted to customize the product if they want to. If they are finally happy with everything, the purchase will finally be made. After the purchase is made, the customer can still browse around the website for other products. Note that the customer can also browse around the website if they do not have an account, and if they do not wish to purchase the product because of not having enough funds, they can just stop right there and not proceed any further than that. It’s just that they cannot purchase anything. One other example of a UML diagram that could help details the new solution is the flow of how a manager or employee can post a new product on the website using a different account that has a bit more privileged access on the site itself. This will only be relevant to the employees of the company though.

**TASK--5**

1. **Three prototyping perspectives**

Iterative development is an important method for gradually moving from a high-level idea to a finished product.

A prototype is:

• A portion of the complete solution that is missing;

• Utilized to learn more about the requirements;

• Disposable or evolutionary (evolving into the final solution);

• Evolutionary prototyping is a method for developing the solution in small steps and practicing what you learn.

There are 3 perspectives when it comes to iterative development and prototyping:

**Functional –**

* focuses on its usefulness;
* It explains how the functionality of a business has been achieved.
* The developer demonstrates that the business needs are met;
* This verifies the developer's comprehension of user requirements;
* Proves that "building the right solution" is true.

Focuses on the main problem rather than the other things like design, user interface, security etc.

**Usability –**

* emphasizes the user interface;
* illustrates the solution's ease of use;
* user tests the solution's ease of use.

Focuses on things that matter from the user’s perspective like the design, the controls on the application, the user interface, etc. while still holding the technical part of it in the background of the prototype.

**Non-Functional –**

* It focuses on things that are nonfunctional (like response time, security, etc.). The non-functional requirements of a solution are examined by a solution developer.

Focuses on things that are non-essential, but still needed to make the system reliable like the security of the system, how performant it is. How much response time it is taking to load up, and if it is having bugs or not?

1. **2 development strategies, the type of project they are most suitable for and how they impact the structure or focus of timeboxes**

**The Horizontal approach –**

The horizontal approach looks at the solution one layer at a time, with each Timebox gradually adding layers of technology or business process complexity or completeness.

The horizontal approach has the advantage of allowing an early glimpse of the solution's full breadth. The disadvantage is that until the final horizontal slice is delivered, nothing functions fully. As a result, no benefit to the business can accrue prior to that point. Given the same set of functional areas, a technical individual might make the observation that there may be a common domain model that supports all three functional areas: The shipping system would need to connect to the inventory domain or at least share a model with it in order to use the order processing domain. A common (or derivable) definition of what constitutes a ship-able entity would be required by the inventory domain in turn.

The project that this approach is suitable for is a solution that can be naturally viewed and developed in layers.

Nothing that is fully functional and usable is finished until the final timebox.

**Vertical approach –**

Each Timebox provides one or more fully functional features as part of the vertical approach, which slices through the solution's multiple layers.

The vertical approach has the advantage of allowing Solution Increments to be deployed into live use more quickly and frequently by delivering prioritized features. The disadvantage is that the solution's full scope is only discovered at the end of the project.

Consider a straightforward example. Let's say a company wants to create a typical software product that handles shipping, tracks inventory, and processes orders (which we can refer to these as "functional areas"). When a product owner and a business analyst discuss the scope of the project, they might decide to start with inventory management, then move on to order processing, shipping, and so on. This would make sense, because each functional area has a really clear conceptual boundary that even a layman could draw around. Software management and those in the planning field frequently consider this approach to be optimal. This is a very common strategy.

This type of approach is more suitable for projects that have features that are to be shipped into production more frequently and quickly.

Each timebox will deliver a fully functional, usable or at least close to a fully functional feature.

**TASK--6**

**Critical Assignment reflection and review:**

Description and Analysis of what came (the what and so what?).

The assignment brought upon a scenario that forced me to research and reflect upon the learning I did for Agile development. It brought upon a plethora of new concepts, ideas and motives that were introduced to me through the use of scenario-based questions. Questions encouraged me to research on things like risk logs which I had no idea about before starting this assignment. The activity diagram model also related to logic diagrams that I learnt for prototyping programming which made me reflect back to that topic. There were many new terms and concepts that were introduced to me like the Planning poker, the PAC (Project Approach Questionnaire) etc. I learnt quite a lot from this assignment and from the way it was structured. However, one thing I do sort of hold a tinge of controversy for, is the fact that not all of these things may or will be used in the real-world development scenarios and that even executing such methodologies like the planning poker and so on, only will increase the complexity of the project at hand.

Future Plan (now what?).

I just need to research more on the efficiency and use case of each of these Agile approaches, in order to correctly understand where and when to use Agile methodologies for small. Medium and large-scale projects. I need to gain a deep understanding of how these methodologies positively or negatively affect the time and efficiency and psychology for the people using them when doing a project. I need to make use of the DSDM method when carrying out a small to medium sized project for a SME business to achieve a better understanding about DSDM. Additionally, I need to understand why DSDM is compatible with the majority of corporate environments without sacrificing agility.

**Conclusion:**

Because it takes advantage of a deeper understanding and ensures that the Deployed Solution fulfills the genuine business need, DSDM was appropriate for this project. DSDM makes change possible through iterative development and regular reviews to ensure that what is being developed meets the actual needs of the business. In addition to addressing many of the issues that are inherent in a conventional approach, DSDM addresses many of the general concerns regarding Agile development. In particular, DSDM requires early consensus on the project's fundamental foundations. Before the proposed solution is developed, businesses are able to comprehend its scope, fundamental characteristics, and production method. By clarifying and agreeing on the project's foundations from the combined perspectives of business, solution, and management, DSDM projects reduce the likelihood of unpleasant surprises. For larger corporations or organizations with intricate architecture and/or governance standards, finishing the foundations early in the project is especially crucial.

**WORD COUNT: Approx. 4100**

**REFERENCES:**

DSDM PROJECT FRAMEWORK

<https://www.agilebusiness.org/dsdm-project-framework/choosing-dsdm.html#:~:text=DSDM%20capitalises%20on%20the%20greater,what%20the%20business%20really%20needs>. *Accessed on: 04/10/2022*

Planning Poker

<https://www.atlassian.com/blog/platform/a-brief-overview-of-planning-poker>

*Accessed on: 04/10/2022*

Risk Logs

<https://sitemate.com/resources/articles/safety/risk-log/> *Accessed on: 04/10/2022*

DSDM Lifecycle

<https://agile-mercurial.com/2018/07/09/a-full-lifecycle-agile-approach-dynamic-systems-development-methodology-dsdm/#:~:text=DSDM%20is%20intended%20to%20be,entire%20project%2C%20including%20the%20releases>.

*Accessed on: 04/10/2022*

Agile development cheat sheet

<https://www.interaction-design.org/literature/topics/agile-development#:~:text=Agile%20development%20is%20an%20iterative,show%20Agile%20development%20in%20action>.

*Accessed on: 04/10/2022*

Iterative development

<https://www.zstream.io/blog/what-is-the-agile-iterative-approach-and-where-is-it-used>

*Accessed on: 04/10/2022*

Activity diagrams

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/> *Accessed on: 04/10/2022*

Instrumental success factors

<https://www.coursehero.com/file/p24d3gmt/Instrumental-Success-Factors-ISFs-Reducing-the-risk-1-Acceptance-of-the-DSDM/>

*Accessed on: 04/10/2022*