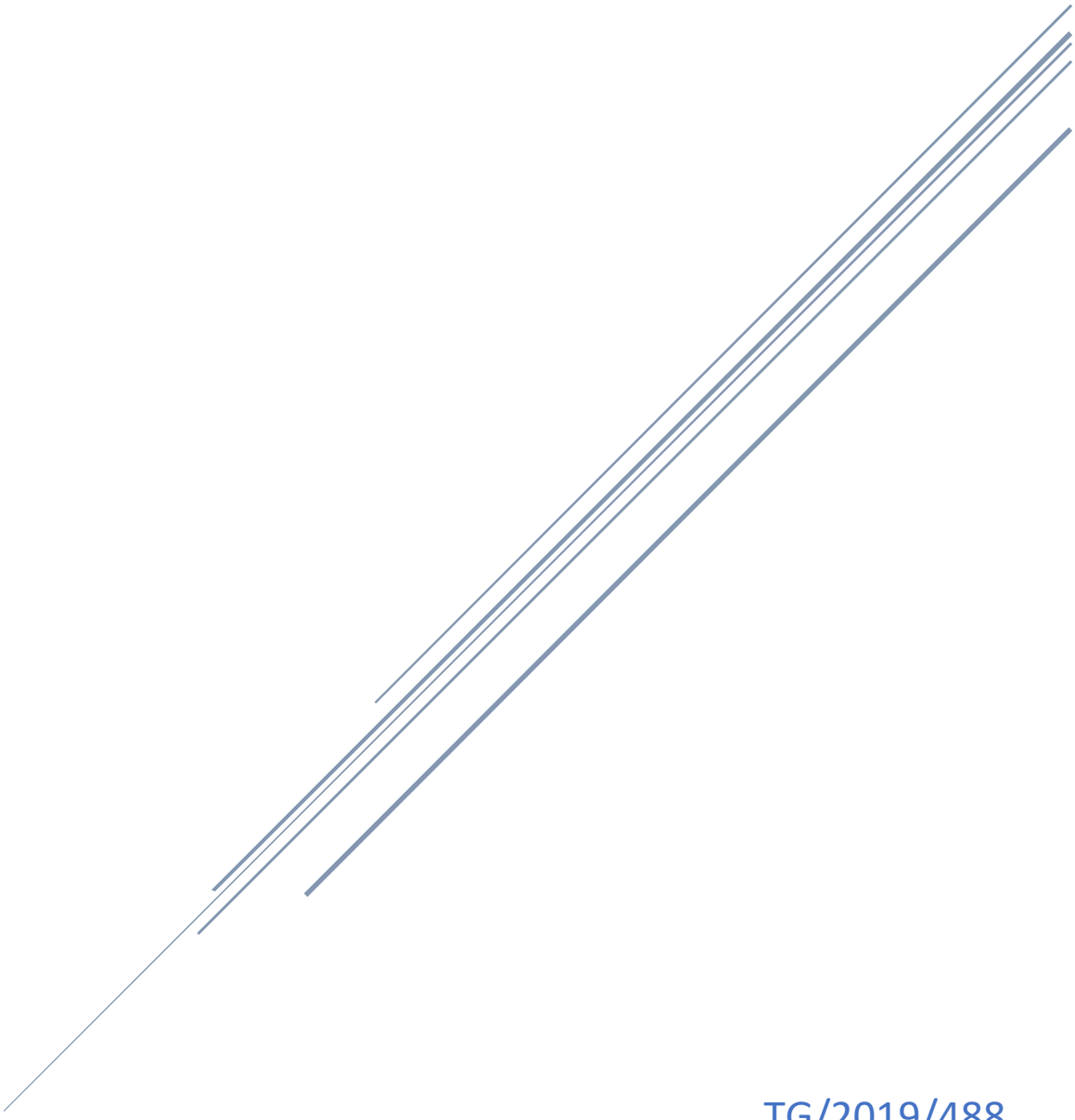
INDIVIDUAL ASSIGNMENT

Rapid and Agile Application Development | ICT3162



TG/2019/488

1) Analysis of Project Planning and Execution in SCRUM, DSDM, and Lean Software Development

SCRUM

SCRUM is an agile project management and completion framework. It focuses on segmenting work into short, 1-4 week sprints that are manageable iterations. Collaboration, constant improvement, and adaptability in response to shifting requirements are all stressed by scrum. A Product Owner, a Scrum Master, and a Development Team make up a Scrum team. The product backlog, which is a prioritized list of features or requirements, is managed by the product owner. The Scrum Master guides the Scrum process and aids in the team's continued concentration and productivity. At the conclusion of each Sprint, the Development Team is in charge of delivering the product increment.

A huge telecommunications company's software development team is employing Scrum to create a new customer site, for instance. Daily stand-up meetings are held by the team to discuss advancement and pinpoint any challenges. In order to evaluate their work and pinpoint areas for enhancement, they also conduct retrospectives and sprint reviews. In order to prioritize the backlog and make sure that the team is providing value to the business, the Scrum Master collaborates with the Product Owner.

DSDM

The Agile framework DSDM (Dynamic Systems Development Method) is used to deliver software projects fast and effectively. It places a strong emphasis on working together, delivering functional software frequently, and paying attention to business needs. The project is kept on track through DSDM's use of timeboxing, which establishes hard deadlines for project phases. Teams using the DSDM methodology are self-organizing, cross-functional, and they collaborate closely with stakeholders to make sure the project satisfies business requirements.

An illustration is the use of DSDM by a financial services organization to create a new trading platform. Developers, testers, business analysts, and end users make up the project team. The team closely collaborates with stakeholders to ensure that the platform satisfies business objectives and uses timeboxing to establish hard timelines for each project step. Every two weeks, the team provides functional software, and it performs regular reviews with stakeholders to obtain input and make changes.

Lean Software Development

Lean Manufacturing is a framework for Agile development that is built on the ideas of lean manufacturing. It places a strong emphasis on providing value to the client by reducing waste, streamlining the entire process, and continuously improving. Teams are encouraged by lean software development to concentrate on producing functional software as soon as feasible and to use user feedback to direct development. The framework also emphasizes the importance of an empowered and cooperative team culture.
Example: A startup is creating a new mobile app utilizing lean software development. The development team consists of programmers, designers, and product managers who collaborate closely to provide value to the client. The team focuses on providing functional software as soon as possible and uses a Kanban board to track progress. They continuously improve the app depending on user demands and behavior, using customer input as a development guide. The team also upholds a culture of continuous development, holding regular retrospectives to pinpoint problem areas and modify their procedure.

2) Highlight three benefits and three trade-offs for each of the following frameworks:

a) SCRUM

Benefits:

Continuous delivery of functional software: Scrum places a strong emphasis on providing functional software at the conclusion of each Sprint, allowing stakeholders to track development and offer comments often.

Flexibility and adaptability: Because Scrum teams work in brief iterations, they are able to modify the product backlog in response to shifting requirements and priorities.

Improvements in team collaboration and communication are made possible by Scrum's encouragement of daily stand-up meetings, Sprint Reviews, and Retrospectives.

Trade-offs:

Strong team commitment is necessary to implement Scrum, which can be difficult to maintain over the course of a lengthy project.

It might not be appropriate for all projects: Scrum works best for large, uncertain projects; it might not be useful for smaller, simpler projects.

Experienced Scrum Master and Product Owner are required because Scrum largely relies on them to manage the workflow and product backlog, respectively. Team members in these positions who lack experience or expertise can have a negative effect on the project.

b) DSDM

Benefits:

Focus on business requirements: DSDM places a strong emphasis on fulfilling business requirements and providing value to customers, which helps to guarantee that the project is in line with business objectives.

Clear project deadlines are established via timeboxing, which helps to guarantee that the project stays on course and is finished on schedule.

Early and regular stakeholder interaction is encouraged by DSDM, which helps guarantee that the project satisfies corporate needs and client requirements.

Trade-offs:

Projects with less clearly defined or more flexible requirements may not be appropriate for all projects because of the DSDM's emphasis on deadlines and tight project phases.

Requires experienced and talented team members: To be successful, DSDM's self-organizing, cross-functional team needs experienced and skilled team members.

The focus on stakeholder involvement in DSDM may entail a large investment of time and resources from the business, which could be a trade-off for some projects.

c) Development of lean software

Benefits:

By removing waste and concentrating on what is necessary, Lean Software Development places a strong emphasis on providing value to the client.

Continuous improvement is encouraged by lean software development, which aids the team in identifying and getting rid of inefficiencies over time.

Increased team autonomy and empowerment: Lean Software Development places a strong emphasis on team autonomy and empowerment, which can increase team members' engagement and work happiness.

Trade-offs:

Lean Software Development needs extensive advance planning, which can be time-consuming and resource-intensive, in order to identify and reduce waste.

Lean software development may not be effective for more complicated or uncertain projects. It is best suited for projects with clearly defined needs and a well-understood customer base.

Lean Software Development requires good team communication and collaboration, which can be difficult to maintain if team members lack the requisite knowledge or experience.

3) Determine the potential obstacles for using the following frameworks and analyze the major risks and issues associated with each of them.

a. SCRUM

Potential difficulties

1. Lack of team buy-in: The development team must be highly committed and buy into Scrum. Team members may be reluctant to follow the practices if they do not completely comprehend or support the Scrum framework.

2. Change is difficult to adjust to because of Scrum's flexibility and adaptability to shifting requirements and priorities. The level of unpredictability and change that comes with the iterative method, however, may be difficult for some team members to handle.

3. Ineffective teamwork and communication: Scrum largely depends on teamwork and effective communication. Ineffective communication among team members can affect the group's capacity to produce high-quality software.

Major dangers and problems:

1. Poorly defined or prioritized product backlog: A poorly defined or prioritized product backlog may cause delays and cause the development team to become frustrated. The quality of the given software product may also be affected.

2. Lack of Product Owner engagement can cause delays, confusion, and a lack of direction for the development team because the Product Owner is so important to the Scrum process.

3. Under- or over-commitment: Scrum depends on the development team committing to a specific set of tasks for each Sprint. Overcommitting the team might result in burnout and subpar output. Underperformance and delays can result from undercommitment, frustrating stakeholders.

b. DSDM

Potential difficulties

1. Resistance to change: For some firms, DSDM necessitates a considerable transformation in culture and thinking. The effectiveness of the framework may be impacted if the team members are resistant to change.

2. Stakeholder engagement is crucial to the success of DSDM, and there has been little of it. Delays and misconceptions may result from incomplete or delayed feedback from stakeholders.

3. DSDM stresses providing functional software quickly, but it also requires a high level of quality. This presents a challenge in balancing time and quality. Some development teams may find it difficult to strike a balance between speed and quality.

Major dangers and problems:

1. Poorly defined requirements: For DSDM to be effective, the requirements must be precise and well-defined. Misunderstandings and delays may result from improperly defined or written requirements.

2. Inefficient communication and collaboration: DSDM depends on efficient teamwork and stakeholder coordination. The quality of the supplied software product may be impacted by poor communication or team member cooperation.

3. Failure to satisfy business needs: DSDM places a strong emphasis on satisfying business needs and providing value to customers. Stakeholders may become angry and frustrated if the development team is unable to achieve these expectations.

c. Development of lean software

Potential difficulties

1. Lack of knowledge of lean principles: To effectively use lean principles in software development, one must have a thorough comprehension of them. The efficiency of the framework may suffer if team members do not completely comprehend these fundamentals.

2. Identification and elimination of waste can be challenging. This is a crucial aspect of lean software development. Inefficiencies and delays may result if the development team has trouble finding waste.

3. Resistance to change: Some firms must undergo a considerable culture and mentality transformation in order to implement lean software development. The effectiveness of the framework may be impacted if the team members are resistant to change.

Major dangers and problems:

1. Ineffective prioritizing: To provide value to the client, Lean Software Development relies on effective priority. Incorrect prioritization by the development team may cause delays and annoyance among stakeholders.

2. Ineffective processes: Lean Software Development mandates that the development team continuously evaluate and enhance its procedures in order to reduce waste. This might cause delays and inefficiencies if the team doesn't do it well.

3. Customer needs are not being met: Lean Software Development places a strong emphasis on providing value to the customer. Stakeholders may become frustrated and angry if the development team is unable to satisfy client needs.

4) Suggest key strategies from the perspective of a project manager to avoid the obstacles you have identified in Question 3. Recommend key actions that you can take in order to mitigate the risks associated with those frameworks. Provide three (3) real- world examples to support your suggestion.

a. SCRUM

Techniques for navigating challenges

1. As a project manager, it's critical to make sure that the development team is aware of and supportive of the Scrum framework. Training, coaching, and regular communication can help with this.

2. Encourage flexibility: Make sure the team understands the value of flexibility and adaptability. Encourage them to welcome change and see it as a chance to make the product better.

3. Promote good communication: Encourage team members to collaborate effectively by holding regular meetings and maintaining open lines of communication.

Steps to reduce risks include:

1. To ensure that the product backlog is clearly defined and prioritized, collaborate closely with the product owner. By doing this, the team can make sure that the most crucial tasks come first.

2. Make sure the Product Owner is actively participating in the Scrum process and giving the development team timely feedback and direction by working closely with them.

3. Manage commitments: Work with the development team to make sure they're promising a reasonable amount of work for each Sprint. By doing this, it will be possible to prevent the team from overcommitting or undercommitting.

Examples from real life

1. A consumer products company's software development team was having trouble getting the rest of the team on board with Scrum. To help the team grasp the advantages of the Scrum framework, the project manager set up a training session. Collaboration and team buy-in were enhanced as a result.

2. A healthcare company's software development team was having trouble adjusting to changes in the Scrum approach. The team was urged by the project manager to see change as a chance to enhance the final result, and support and coaching were given to aid in their adjustment.

3. A financial services company's software development team was having trouble communicating with one another during the Scrum process. To enhance collaboration and communication, the project manager established frequent stand-up meetings and transparent communication channels.

b. DSDM

Techniques for navigating challenges

1. Encourage stakeholder involvement by highlighting its significance throughout the project. Encourage regular stakeholder communication and input.

2. Encourage a culture of change by stressing its significance and motivating the team to see change as a chance to enhance the final result.

3. Work with the development team to make sure that they are successfully balancing time and quality.

Steps to reduce risks include:

1. Clear requirements definition: Collaborate closely with stakeholders to establish and clearly describe requirements. This can ensure that the group is working on the appropriate tasks at the appropriate time.

2. Promote efficient communication and teamwork: Encourage efficient communication and teamwork among team members and stakeholders. Establish regular gatherings and transparent avenues of communication.

3. Deliver value to the client: Assist the development team in making sure the client is receiving value. Examine progress frequently, and change priorities as necessary.

Examples from real life

1. A retail company's software development team was having trouble getting stakeholders involved in the DSDM process. To increase collaboration and involvement, the project manager established frequent stakeholder meetings and transparent communication channels.

2. A technology company's software development team was having trouble keeping time and quality in check during the DSDM process. The project manager collaborated with the development team to pinpoint areas where it would be possible to better balance quality and timeliness.

3. A tourism company's software development team was having trouble working together during the DSDM process. The project manager promoted effective communication and teamwork and instituted routine team meetings.

c. Development of lean software

Techniques for navigating challenges

1. Promote comprehension of the Lean concepts by making sure the development team is aware of and supportive of them. As required, offer coaching and training.

2. Encourage the development team to regularly evaluate and enhance their processes and emphasize the necessity of waste removal.

3. Encourage the development team to adopt a culture of continuous improvement and to see change as an opportunity to better the product.

Steps to reduce risks include:

1. Effectively prioritize requirements by collaborating with all relevant parties. This can ensure that the development team starts by focusing on the most crucial tasks.

2. Encourage the development team to continuously evaluate and enhance their procedures in order to reduce waste and boost productivity.

3. Deliver value to the client: Assist the development team in making sure the client is receiving value. Examine progress frequently, and change priorities as necessary.

Examples from real life

1. A manufacturing company's software development team was having trouble grasping the ideas behind lean manufacturing. To help the team better grasp and promote Lean principles, the project manager offered coaching and training.

2. A software development team at a healthcare organization was having trouble finding ways to streamline their operations. The project manager collaborated with the development team to find wasteful processes and put improvements into place.

3. A financial services company's software development team was having trouble providing value to the client. In order to determine priorities and make sure they were providing value to the client, the project manager collaborated with the development team.

5) Use at least four quality resources in this assignment. Note: Wikipedia and similar Websites do not qualify as quality resources. You may use the resources your choosing.

1. Schwaber, K., & Sutherland, J. (2017). The Scrum Guide. Scrum.org. Available at: https://www.scrum.org/resources/scrum-guide [1]
2. DSDM Consortium. (2014). Agile Project Framework Handbook. DSDM Consortium. Available at: https://www.agilebusiness.org/resources/agile-project-framework-handbook [2]
3. Poppendieck, M., & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit. Addison-Wesley Professional. [3]
4. Agile Alliance. (2021). Agile Manifesto. Agile Alliance. Available at: https://agilemanifesto.org/ [4]

References:

[1] K. Schwaber and J. Sutherland, "The Scrum Guide," Scrum.org, 2017. [Online]. Available: https://www.scrum.org/resources/scrum-guide. [Accessed: 8-May-2023].

[2] DSDM Consortium, "Agile Project Framework Handbook," DSDM Consortium, 2014. [Online]. Available: https://www.agilebusiness.org/resources/agile-project-framework-handbook. [Accessed: 8-May-2023].

[3] M. Poppendieck and T. Poppendieck, Lean Software Development: An Agile Toolkit. Addison-Wesley Professional, 2003.

[4] Agile Alliance, "Agile Manifesto," Agile Alliance, 2021. [Online]. Available: https://agilemanifesto.org/. [Accessed: 8-May-2023].