

MET CS 682 ASSSIGNMENT 2

Michael Nguyen & Judith Glikberg

The purpose of this exercise is to give you practice exploring development processes and identifying risks when embarking on a project.

1. USE THIS TEMPLATE: Please respond by using this Word template, leaving the headings and the gray text unchanged—but exclude the *hints* section (at the end) from your solution.
2. OBSERVE LIMITS: Observe the page and paragraph limitations; however, you may include as many appendices—additional to the required Appendix 1—as you wish. All appendices should be referred to in the main text.
3. USE chatGPT: We would like you to answer this assignment using chatGPT where possible because we feel that this is the way such writing will be created going forward. We want you to practice good use of AI generation, including your verification, re-prompting, editing, and improvement.
4. Document your chatGPT interaction as in Appendix 1 (use Document Map to facilitate navigation). You are also encouraged to do outside research to support your response but use the *References* section as indicated and observe plagiarism rules carefully, including the citation of sources, the use of quotes, and acknowledgement of modifications of external sources.
5. NAME YOUR FILE: Include your last name in the file name of the assignment. (Example: SmithMichael_CS682Assignment1.docx)

Hints are provided at the end of this document. We will provide feedback comments and will use our best judgement in terms of the evaluation criteria listed.

The Scenario for this assignment is as follows: *HealthyEats* specializes in providing healthy nutrition and food delivery for corporations, specifically targeting small businesses and startups.

The following project characteristics apply—every one of them should influence your response:

1. The CEO believes that *HealthyEats* must prioritize user experience and integrate with social media, but investors want health and safety features to be the top priority, partially to avoid liability exposure—at least for the initial development.
2. Your initial team consists of one experienced systems analyst, a subject area expert, and one developer experienced with programming of this type of system. Five additional developers and two QA personnel are usually needed for this type of project to be delivered in the timeframe of the initial release based on your previous experience of working on similar projects.
3. The initial team reside in time zones with substantial difference. The experienced developer is very talented; however, they have a history of avoiding overtime, even at critical times.
4. To keep costs low, the CEO would like to hire as many offshore developers and QA staff as possible, even though they would probably be in very different time zones, potentially with deep cultural differences. To some degree, allow for pandemics, war, sanctions, and unrest to impact the offshore development team.
5. You are to assume the timeframe of the initial release and or iterations depending on your process selection

1. Process

Using the format below, compare the following development processes for this project.

1. Waterfall Process; 2. An iterative (Rapid Application Development) process with 3-4 iterations; 3. Agile Process

1.1 Comparison of Processes

For each of these three development processes, provide one strength and one and weakness relative to this particular project. Focus on what you consider the most important strength and weakness of the development process specific to the characteristics of the given scenario. (up to 2 pages of 12-point text)

1.1.1 Waterfall Strength

The Waterfall methodology, created in 1970, is a process which emphasizes defined requirements, consistent documentation, and sequential development. At its core the Waterfall methodology is thorough and puts emphasis on a hierarchical structure to ensure each step of the process is completed before moving on to the following. For this project at *HealthyEats* in particular, the structure that the Waterfall methodology enforces would most benefit our upcoming team of offshore developers. Specifically, because the project team is dispersed across different regions/time zones and considering that additional hires will be offshored for cost-effectiveness, Waterfall's main strength could significantly contribute to its success. To elaborate, this mandated thoroughness in requirements, documentation, and completion at each step of the project, will allow for ease of communication across the dispersed teams and individual employees. Within the dedicated timeframe, offshored or inexperienced developers can follow documentation established in earlier steps at their respective work hours as a guideline (Jafari, 2021). This would allow for *HealthyEats* to have more options when hiring developers and QA analysts for this project without compromising on the project's goals or success metrics.

1.1.2 Waterfall Weakness

Notably, though Waterfall's emphasis on comprehensive documentation at each step can be helpful in regards to *HealthyEats*' dispersed team for this project, this process also has prominent weaknesses that stem from its core value on thoroughness. In specific, the need for established requirements to dictate documentation and the functionality of this project would require our initial assessment of how to implement this integration with social media to be etched in stone. Moreover, lack of flexibility may cause conflict with the CEO's priorities in social media integration. Investors focusing on health and safety, which may hinder the ability to test and modify the systems based on end users' experience. Shifting in priority during this project may be a costly attempt to revisit earlier stages, which cause confusion among team members. Inflexibility may not only impact evolving requirements, however, limits the risk requirements in management and causes crucial users' feedback (Kissflow, 2023). In conclusion, Waterfall may support *HealthyEats*' offshore teams, however, it's necessary to document potential compromising accounts of adaptability and responsiveness.

1.1.3 RAD Strength

Rapid Application Development (RAD) is an iterative process that provides flexibility by incorporating analysis, design, and implementation phases, departing from traditional waterfall processes. It allows *HealthyEats* to follow the structural waterfall steps and create editable versions of social media integration software with prioritized requirements. This approach facilitates the integration of executive feedback, reducing system time for the CEO and investors. Given *HealthyEats'* focus on dietary preferences and healthy eating trends, the integration with current social media trends can provide valuable market data the sooner it is released. Working on parallel versions enables early issue detection while the system is already available to the public, benefiting users and investors (Kissflow, 2023). In addition to these advantages, RAD maintains a step-by-step process with comprehensive documentation, supporting dispersed teams and allowing room for feedback and changes.

1.1.4 RAD Weakness

RAD Phased Development leverages strengths from the Waterfall methodology by incorporating iterations for feedback and implementing necessary changes. However, it introduces certain weaknesses that merit consideration. Specifically, RAD Phased Development may compromise *HealthyEats'* CEO's key requirement of prioritizing user experience in the social media integration. While version iterations allow for user feedback, subsequent versions have limited flexibility to incorporate this input, focusing primarily on less prioritized requirements. Additionally, releasing a version 1.0 product may lead to user dissatisfaction, especially when higher-priority features are not addressed. Furthermore, the model's emphasis on time-to-market could sideline essential health and safety features, crucial for a business like *HealthyEats* (Kissflow, 2023). In summary, although RAD Phased Development enhances Waterfall by maintaining thoroughness and flexibility, it deviates from user-focused development, emphasizing time-to-market over *HealthyEats'* core tenets of health and safety. While RAD Phased Development does improve upon Waterfall and maintain levels of thoroughness and flexibility it strays from user focused development and dictates focus on time-to-market which undermines the central tenancies of health/safety titular to *HealthyEats'* long-term goals.

1.1.5 Agile Strength

Agile methodology boasts several strengths that make it a popular choice in project management. One key advantage is the adaptability in changing requirements. Agile embrace flexibilities and welcomes project needs, allowing team members to easily adjust their courses as new insight emerges on priority shifting. The iterative nature of Agile promotes regular reassessment, ensuring that the product of *HealthyEats* aligns closely with the client expectations. Another strength lies in its emphasis on collaboration and communication. Agile frameworks, such as Scrum, encourage constant interaction among team members, fostering a dynamic and collaborate work environment (Peek, 2023). This leads to improvement in problem-solving, quick decision-making, and a heightened sense of collective ownership over project outcomes. In addition, Agile

incremental delivery approach allows for early and continuous release of valuable features, providing stakeholders and customers with tangible results. Similarly, users are empowered to provide feedback directly to developers who can then be reassured regarding the project's direction towards meeting requirements. The project increases overall transparencies and strengthens collectively, contributing to Agile framework of effectiveness. Moreover, delivering higher quality, customer-centric solutions within dynamic fast-paced environment (Kissflow, 2023).

1.1.6 Agile Weakness

While Agile methodologies bring many strengths, there are also challenges that may present. One of the weaknesses is the potential difficulty in handling large-scale projects with complex requirements. Agile is designed for flexibility and adaptability, however, in situations where intricate planning and predictability are crucial, certain parts of *HealthyEats'* operations may lead to Agile's dynamic nature as a drawback. To clarify, the emphasis on continuous collaboration and rapid iterations might require extensive involvement from stakeholders, which may be challenging if key individuals have limitation on availability (Sinha, 2022). Additionally, the iterative development cycles may not align seamlessly with certain long-term plan needs, potentially creating tension between Agile preferences for adaptability and the company desire for comprehensive project roadmaps (Garbar, 2023). In a company, such as food and health where regulatory compliance is a paramount, the less prescriptive nature of Agile may be necessary when efforts needs to ensure compliance to standards. Thus, while Agile is renowned for its strengths, these weaknesses should be carefully considered in the context of *HealthyEats'* specifically in requirements and operations.

A1.1 (ChatGPT re Comparison of Processes)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

Replace this with your response.

ChatGPT responses:

When considering the application of the Waterfall model to the development of HealthyEats, it's important to be aware of the potential weaknesses of this methodology. Here are some Waterfall weaknesses that might be relevant to the HealthyEats project:

Limited Flexibility to Changes:

- Waterfall is known for its sequential and rigid structure, which can make it challenging to accommodate changes in requirements during the development process. If HealthyEats project experiences is in evolving needs or shifts in priorities, adapting to these changes may be difficult.

Late Visibility and User Feedback:

- In Waterfall, the client or end-users typically see the product only after the entire development process is complete. This late visibility can lead to a situation where

issues or misalignments with user expectations become apparent only at the end of the project.

Higher Risk of Customer Dissatisfaction:

- Since the client's involvement is limited to the beginning and end of the project, there is a risk that the final product may not fully meet their expectations. This could result in dissatisfaction, especially if there are misunderstandings or changes in requirements.

Longer Time-to-Market:

- Waterfall projects often have longer time-to-market compared to more iterative methodologies. The sequential nature of Waterfall can result in delays, especially if there are dependencies between project phases.

Difficulty in Handling Complex Projects:

- Waterfall may face challenges in managing complex projects, especially those with a high degree of uncertainty or rapidly changing requirements. Breaking down a complex project into detailed phases may not always be feasible or effective.

Limited Room for Risk Management:

- The Waterfall model tends to handle risk management in a linear fashion, often addressing risks only when they become apparent during the testing phase. This can lead to challenges in proactively managing risks throughout the entire project lifecycle.

Inadequate User Involvement:

- Limited user involvement during the development phases may result in a product that does not fully align with user expectations or needs. Regular feedback and collaboration, which are promoted in Agile, are somewhat restricted in Waterfall.

Resource Utilization Challenges:

- Waterfall requires a detailed plan and allocation of resources for each phase upfront. If there are unexpected resource constraints or changes in project scope, it can be challenging to reallocate resources efficiently.

Dependencies on Initial Requirements Accuracy:

- Success in a Waterfall project heavily relies on accurately capturing and defining all requirements at the beginning. If there are gaps or inaccuracies in the initial requirements, it can lead to significant challenges later in the project.

Cumbersome for Large Projects:

- The Waterfall model may become cumbersome for large and complex projects due to its sequential nature. Managing a large-scale project with numerous dependencies between phases can be challenging.

Considering these weaknesses, it's essential to carefully evaluate whether the Waterfall model is the best fit for the HealthyEats project. Depending on the specific project characteristics, a more flexible and iterative approach, such as Agile or a hybrid methodology, may be worth exploring to address these potential challenges.

1.2 Project Methodology and Roadmap

Provide a process for this project—or, more typically, a combination of processes—that you consider most appropriate. Provide a Roadmap in the form of a Work Breakdown Structure which clearly shows the phases of the process you designate, what is to be completed, and by whom. In this section you do not need to explain your choices, this will be done in the next section. (Up to 1 page of 12-point text)

The following chart contains a roadmap and outline of the sprint iterations in developing the *HealthyEats* system. Details include the duration of each phase and the focus of each phase as well those who will take the lead on the deliverables for each. In specific, we see each iteration until release with the initiation being an establishment of the business requirements, shedding light on the improvements to the business that the system will provide, and creation of the team, breaking them down into smaller groups as necessary. Followed by specific sprint planning and estimates that determine the priority requirements and deliverables, dividing them among teams with documented standards for quality. Then, implementation of the plan by way of programming and scrum master led meetings to consistently check in allow for a finalized product to be made within a short amount of time (West, n.d.). Lastly, review of the first iteration of the product to ensure prioritized requirements were met, releasing the iteration for customer feedback, and identifying improvement points lead into the following iteration where the following prioritized requirements become the focus (Peek, 2023).

Supply the completed chart begun below.

Iteration	Phase	Objective	Duration	Notes (By Whom)
First Iteration (Sprint)	Initiation	The creation of the project's requirements, deliverables, and goals. During this step all stakeholders will be identified and team members will be assigned their roles. Formation of the product backlog which includes key features and requirements of the system for the overall team to note.	1 week	Product Owner, Project Managers, and Systems Analysts

	Sprint Planning & Estimates	In this stage, the critical forming of user stories is implemented. These meetings will guide the breakdown of the project into more manageable tasks during the sprint. The developer team creates a sprint plan to focus on certain user stories for this iteration in their scrum backlog. Determine which requirements will be the focus of this sprint.	2 -3 days	Project Manager, Systems Analyst, Scrum leaders, and Developers
	Implementation	In this stage the execution of the planned tasks commences with developers coding in teams. Daily check-ins amongst teams and planned check ins with project managers /scrum leaders occur to keep track of sprint progress.	1 week	Scrum Leaders and Developers
	Review & Retrospective Phase	This step occurs after the sprint ends allowing both stakeholder's and team members to evaluate success metrics as well as improvement points. Customer feedback is gathered and critical in ensuring steps for moving forward. Optimizing the next sprint and updating the product backlog in preparation for the second iteration all come into play.	5 – 8 hours total	Systems Analyst, Quality Analysts, Project Manager
Second Iteration (Sprint)	Sprint Planning & Estimates	In the second iteration, user stories are updated and adapted based on changing requirements. These guide updated breakdowns of the project's second sprint. Determine which requirements will be the focus of this sprint after taking into account feedback from stakeholders.	1 week	Project Manager, Systems Analyst, Scrum leaders, and Developers
	Implementation	In this stage the execution of the planned tasks commences with developers coding in teams. Daily check-ins amongst teams and planned	1 week	Scrum Leaders, Developers, and Project Managers

	check ins with project managers /scrum leaders occur to keep track of sprint progress.		
Review & Retrospective Phase	This step occurs after the sprint ends allowing both stakeholder's and team members to evaluate success metrics as well as improvement points. Customer feedback is gathered and critical in ensuring steps for moving forward. Optimizing the next sprint and updating the product backlog	5 – 8 hours	Systems Analyst, Quality Analysts,
Release	Repetition of sprints continues until a high quality, user experience focused product is deemed eligible for release. Final project deliverables are turned over to relevant stakeholders and released to the public if approved.	6 – 8 hours	Systems Analyst, Stakeholders, CEO, Project Manager
Maintenance	Continued maintenance of the product or system if required.	Post release	Systems Analyst and Developer Team

A1.2 (ChatGPT re Project Methodology and Roadmap)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

User:

Project Methodology and Roadmap.

ChatGPT:

Creating a project methodology and roadmap is essential for planning and executing a successful project. Here's a general guide on how to develop both:

Project Methodology:

- Define Objectives: Clearly outline the project's purpose, goals, and expected outcomes.
- Select Methodology: Choose a project management methodology (Waterfall, Agile, Scrum, etc.) that suits the project's nature and requirements.
- Create a Work Breakdown Structure (WBS): Break the project into smaller, manageable tasks and sub-tasks.
- Assign Responsibilities: Allocate tasks to team members based on their skills and expertise.

- **Set Milestones and Deliverables:** Establish specific points in the project timeline to assess progress and deliverables.
- **Develop a Communication Plan:** Determine how information will be shared among team members, stakeholders, and clients.
- **Risk Management:** Identify potential risks and develop strategies to mitigate them.
- **Monitor and Adjust:** Regularly track progress, evaluate performance, and make necessary adjustments to stay on track.

Project Roadmap:

- **Project Scope:** Define the boundaries, deliverables, and objectives of the project.
- **Timeline:** Create a timeline that includes major milestones, deadlines, and checkpoints.
- **Resource Allocation:** Specify the resources needed (budget, personnel, tools) at different stages of the project.
- **Dependencies:** Identify any dependencies between tasks or team members and account for them in the roadmap.
- **Communicate the Roadmap:** Share the roadmap with stakeholders to align expectations and ensure everyone understands the project's trajectory.
- **Flexibility:** Allow for flexibility in the roadmap to accommodate unexpected changes or delays.
- **Regular Updates:** Update the roadmap as the project progresses and circumstances change.

Steps to Create Both:

- **Gather Information:** Collect all relevant project details, goals, constraints, and resources available.
- **Draft Initial Plans:** Use the gathered information to create an initial draft of the methodology and roadmap.
- **Review and Refine:** Gather feedback from stakeholders, team members, or project managers and refine the plans accordingly.
- **Implement and Monitor:** Implement the plans and continuously monitor progress while making necessary adjustments along the way.

Remember, both the methodology and roadmap should be dynamic documents that can adapt to changes and evolving project needs. Regularly revisiting and updating them ensures the project stays on track and aligns with the project goals.

1.3 Explanation of Process

Explain why the process and the roadmap you selected would work best supporting the characteristics of this project. Integrate the strengths and weaknesses which you described in Section 1.1. We place more weight on this section than on the roadmap itself. (up-to 3/4 page of 12-point text)

The project methodology that would best fit in the current project requirements, the need for cost efficiency, and the dependency on offshore developer support is the Agile Scrum process for system development. Agile Scrum places an emphasis on user experience, not only offering numerous feedback opportunities, however, basing on project requirements and steps on user stories (Peek, 2023). Each iteration is then dictated by feedback from stakeholders and users to amend these user stories, as well as, the steps in the next sprint of the project. This aligns with current requirements placed by the CEO and stakeholders of the *HealthyEats* organization.

One of the key strengths of Agile Scrum is its emphasis on user experience through the use of user stories. User stories are concise, user-centric narratives that describe a specific functionality or feature from an end user's perspective. By basing project requirements on user stories, Agile Scrum ensures that development efforts are directly aligned with the needs and expectations of the end users (Kissflow, 2022). This is particularly relevant for *HealthyEats*, where the user experience is paramount, given the nature of the application.

Moreover, the iterative nature of Agile Scrum, organizes into sprint, allowing for frequent opportunities in feedback. Furthermore, the constant feedback loop is instrumental in addressing challenges associated with offshore team dependencies. Stakeholders, including offshore team members, may provide input and insights at regular intervals, allowing for quick adjustments to the developmental processes. The flexibility inherited in Agile Scrum accommodates changes in project priorities, making it easier to adapt to evolving circumstances, such as those arising from offshore dependencies (West, n.d.).

In the context of cost efficiency, Agile Scrum contributes to the optimization of resources by prioritizing the delivery of high-value features and functionalities in each sprint. Moreover, incremental development approaches allows for continuous releases of valuable updates, providing organizations with flexibility to reassess priorities and adjust project scopes based on budget considerations (West, n.d.).

Additionally, the emphasis on collaboration within Agile Scrum is well-suited to the need for outsourcing development support. Agile practices, such as daily stand-up meetings and collaborative tools, promote effective communication and coordination among team members, irrespective of geographical locations (Smelianska, 2022). This is crucial in mitigating the challenges associated with time-zone differences and ensuring that the outsource teams remains integrated into the development process.

In conclusion, the Agile Scrum process for system development offers a well-balanced and adaptive framework that aligns with the current requirements of the *HealthyEats* project. By emphasizing user experience, providing continuous opportunities for feedback, and accommodating the challenges of offshore team dependencies. Agile Scrum stands as an effective methodology that not only meets the expectations of the CEO and stakeholders, however, ensuring the efficiency and user-focus developments of the *HealthyEats* and its application.

A1.3 (ChatGPT re Project Roadmap)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

Can you provide an outline of the steps in the Agile Scrum system development process?

ChatGPT:

Certainly! The Agile Scrum framework is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer satisfaction. Here's an outline of the steps in the Agile Scrum system development process:

Project Initialization: Define the project vision, goals, and scope. Assemble the Scrum Team, including the Product Owner, Scrum Master, and Development Team.

Product Backlog: Create a prioritized list of features, enhancements, and fixes known as the Product Backlog. The Product Backlog is managed by the Product Owner and reflects the requirements from the customer or stakeholders.

Sprint Planning: The team and the Product Owner collaboratively select a subset of items from the Product Backlog to deliver in the upcoming Sprint. Define the Sprint goal and create a Sprint Backlog, which is a detailed plan for how the selected items will be implemented.

Sprint: Development Team works on the tasks from the Sprint Backlog to produce a potentially shippable product increment. Daily Scrum meetings are held for the team to synchronize their work and discuss any impediments.

Incremental Development: The product is developed in small, incremental releases. After each Sprint, a potentially shippable product increment is delivered.

Sprint Review: At the end of each Sprint, the team conducts a Sprint Review. The team demonstrates the completed work to stakeholders and receives feedback.

Sprint Retrospective: The team reflects on the Sprint's processes and performance. Identify what went well, what could be improved, and make adjustments for the next Sprint.

Repeat: Steps 3 to 7 are repeated for each Sprint. The product evolves based on feedback, and new items from the Product Backlog are selected for the next Sprint.

Release: A potentially shippable product increment is created at the end of each Sprint. When the product meets the release criteria, it can be released to customers.

Continuous Improvement: Scrum emphasizes continuous improvement. The team regularly reflects on its processes and looks for ways to become more efficient and effective. It's important to note that the Agile Scrum framework is highly adaptive, allowing teams to respond to changes and incorporate feedback throughout the development process. The iterative nature of the framework promotes flexibility and continuous improvement.

2. [Risk Analysis](#)

2.1 Five Risks

Based on the project characteristics provided, identify five risks in this project with the highest potential to affect the SDLC process which you selected in section 1.2. In addition, give each risk a title of two to three words and create a prioritization matrix for the risks. Avoid giving a generic response: focus on this particular system under development and its particular characteristics. (up to 1 page of 12-point text) Note the hints on risks below.

When considering the system development methodology for *HealthyEats* development project it is critical to understand the risks of the endeavor with the goal in mind of preparation, mitigation, and lessening impact of these risks. The following is an analysis of these risks and a prioritization created by calculation matrices in order to concisely illustrate the impact, likelihood, and cost to manage these risks.

2.1.1 Risk 1: Supply Chain Disruptions

One of the primary risks for *HealthyEats* is the disruption of its supply chain. Dependency on multiple suppliers for fresh ingredients exposes the business to risks like natural disasters, supplier insolvency, or logistical issues, which could lead to shortages and affect the timely delivery of orders (Hans, 2023).

2.1.2 Risk 2: Miscommunication Due to Cultural Differences

Similarly, a notable risk to bring to light is *HealthyEats*' dependency on offshore teams. While this is an important part of maintaining our budget goals and has become more common amongst tech companies following the recent pandemic, it would be careless to imply that this strategy lacks opportunities for risk. Offshore teams may increase chances for miscommunication due to cultural differences in work environment, expectations, and comprehension (Machado, 2023). The results may slow down the overall development process and decrease quality of the product.

2.1.3 Risk 3: Limited User Feedback & Engagement

Additionally, another risk to consider is limited user engagement and feedback. When testing for quality assurance and needs amongst users, it is important to ensure that the population we use for testing are diverse and reflect the target market for the *HealthyEats* product. The Agile Scrum methodology centers around user experience and limited engagement may lead to lack of direction and decreased understanding of the final product's requirements (West, n.d.). The benefit of consistent morale boost via feedback to developer and quality assurance teams would also be lost in this situation.

2.1.4 Risk 4: Budget Risks

Next, it is important to note that while Agile Scrum methodology can be cost effective there is also the risk of surpassing budget estimations. Budget risks may stem from the uncertainty regarding how many iterations will be needed to meet customer and the ever-changing health trend market needs. Surpassing the budget can also stem from lack of adequate research regarding financial costs to the project at the onset (Sinha, 2022). This can cause dissatisfaction in morale for teams as well as dissatisfaction at the leadership and investor level. Additionally, delays in the project or in extreme cases eventual cancellation of the initiative can occur.

2.1.5 Risk 5: Changing Health & Safety Compliance Requirements

A risk that relates specifically to *HealthyEats* that may occur is the changing of health and safety compliance requirements. As mentioned previously, investors in this project find it critical to ensure that health and safety requirements are met, which is in line with *HealthyEats*' brand and values. If research is not done to thoroughly understand these requirements at the onset of this project or predictions are not made to accommodate for possible future changes in requirements, several unfavorable outcomes may occur (Kissflow, 2023). The costs to this risk can affect the quality of the product, as well as team morale and may expand beyond this. The reputation of the company may become at risk and legal compliance issues may arise – should an audit deem these changes unaccounted for.

Risk No.	Title	Estimated Likelihood of occurrence (L: 1-10)	Estimated impact (I: 1-10)	Estimated cost of managing (M: 1-10)	Priority number (Handled in order lowest to highest) ((11 – L)*(11 – I)*M)
1	Supplier Chain Disruption	5	8	5	90
2	Miscommunication Due to Cultural Differences	7	8	6	108
3	Limited User Feedback & Engagement	4	6	5	105
4	Budget Risk	6	9	7	70
5	Changing Health & Safety Compliance Requirements	7	8	5	60

*The Priority Number is calculated using the formula: $((11 - L) * (11 - I) * M)$*

2.2 Selected Risk

Select one of the above risks and provide as below.

Risk #2 Miscommunication Due to Cultural Differences

A2.2 (ChatGPT re Selected Risk)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

User:

What are five potential risks associated with agile scrum models?

ChatGPT:

While Agile Scrum models offer many advantages, like any development approach, they come with their own set of potential risks. Here are five common risks associated with Agile Scrum models:

Incomplete Requirements:

Risk: Rapid development cycles in Agile may lead to incomplete or evolving requirements.

Impact: Unanticipated changes can affect project scope, leading to delays or rework.

Mitigation: Regular communication with stakeholders and continuous refinement of the backlog during Sprint Planning can help address this risk.

Scope Creep:

Risk: Continuous feedback may result in constant changes to the project scope.

Impact: Increased scope can extend project timelines and increase resource requirements.

Mitigation: Establish clear project goals and priorities, and have a well-defined process for handling change requests.

Dependency on Team Collaboration:

Risk: Agile heavily relies on collaboration and communication within the team.

Impact: Team issues, lack of communication, or conflicts can impede progress.

Mitigation: Foster a collaborative team culture, address conflicts promptly, and ensure effective communication channels.

Unrealistic Sprint Commitments:

Risk: Overcommitting in Sprint Planning may lead to failure in meeting deliverables.

Impact: Reduced team morale, compromised quality, and potential project failure.

Mitigation: Encourage realistic commitments, based on historical team velocity, and promote transparency in task estimation.

Lack of Stakeholder Involvement:

Risk: Limited or inconsistent involvement of stakeholders can result in misalignment.
Impact: Delayed feedback, misunderstandings, and divergent expectations.
Mitigation: Regularly involve stakeholders in Sprint Reviews, prioritize effective communication, and encourage continuous feedback.

It's essential to note that while these risks exist, Agile Scrum provides mechanisms, such as Sprint Retrospectives, for teams to reflect on their processes and improve continuously. Additionally, the Agile mindset encourages adaptability, making it easier to address and mitigate risks as they arise. Regular communication, collaboration, and a commitment to continuous improvement are key to successful risk management in Agile Scrum projects. Let's take the risk of "Supplier Shortage" as an example and delve into its concrete impacts on a project, supported by research:

ChatGPT:

Impact on Project:

A "Supplier Shortage" can significantly disrupt project timelines, cost estimates, and quality standards, leading to delayed deliveries, increased expenses, and compromised product or service quality. Research by organizations like the Institute for Supply Management (ISM) highlights the profound effects of supplier shortages on businesses across various industries. The shortage might arise due to global economic fluctuations, natural disasters, geopolitical tensions, or disruptions in the supply chain.

Research Support:

According to ISM's reports on supplier shortages in the manufacturing sector, when faced with a shortage of critical suppliers, over 80% of companies experienced disruptions in their production schedules. This led to increased lead times for materials, affecting product delivery and customer satisfaction. Moreover, about 65% of companies reported increased costs associated with finding alternative suppliers or expediting shipments to mitigate the shortage's impact.

A study by Deloitte emphasizes that supplier disruptions can have cascading effects on project timelines, causing delays in product launches or service rollouts. This delay can impact revenue projections and market competitiveness. Additionally, Deloitte's research suggests that around 45% of businesses experienced a decline in product quality due to supplier shortages, affecting customer trust and brand reputation.

In summary, a "Supplier Shortage" poses tangible risks to a project by disrupting timelines, inflating costs, compromising product quality, and potentially damaging brand reputation. Understanding and proactively mitigating these risks are critical to ensuring project success and resilience in the face of supply chain challenges.

2.2.1 Likelihood

State and explain the likelihood of occurrence of this risk. Use research to support the likelihood of this risk. (up to ½ page of 12-point text)

The likelihood of this risk occurring is moderate as this is caused by a combination of factors. At its core the likelihood of this risk centralizes on communication. However, this can be greatly dependent on the factors to be discussed.

One such significant contributing factor is the time zone differences between onshore and offshore team members. With both across varying time zones scheduling meetings and overlapping work hours for real-time collaboration can become complex tasks. This can lead to delays in overall communication, making it difficult to address issues when they arise and hindering overall progress of the project (Machado, 2023). While time zone differences are a known consideration when working with offshore teams, the extent of the difference and the adaptability of team members to these challenges will contribute to this risks' overall likelihood.

Another factor of note is the cultural differences that may exist between the onshore and offshore team members. Effective communication relies not only on language proficiency but also on an understanding of cultural nuances. Differences in work approaches, communication styles, and problem-solving methodologies can lead to misunderstandings and, in some cases, impact the efficiency of collaboration. Navigating these cultural differences requires proactive efforts in team building and cultural sensitivity training, adding a layer of complexity to the dependency (Smelianska, 2022).

Furthermore, the experience and expertise of the offshore team also contribute to the likelihood of the risk. A well-experienced and skilled team is more likely to navigate challenges effectively and deliver quality work. However, if the offshore team lacks the necessary expertise or experience in the specific technology, domain, or project methodology, it increases the likelihood of issues arising during the project's lifecycle (Machado, 2023).

Lastly, the level of dependency on the offshore team plays a crucial role. If critical tasks or components of the project are heavily reliant on the offshore team's deliverables, any issues that arise on their end can have a cascading effect on the entire project. Notably, the use of Agile Scrum methodology, which relies on incorporation of user feedback and testing, puts reliance on our offshore developers and quality assurance analysts.

A2.2.1 (ChatGPT re Likelihood)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

The likelihood of the risk stemming from Offshore Team Dependency can be influenced by various factors. Here are some key considerations that contribute to the likelihood of this risk:

Communication Challenges:

Time Zone Differences: If the offshore team operates in a significantly different time zone, it can lead to communication challenges. Scheduling meetings, resolving issues in real-time, and maintaining effective collaboration become more difficult.

Cultural Differences:

Cultural Communication Styles: Differences in communication styles, work approaches, and cultural nuances can lead to misunderstandings and misinterpretations. These cultural gaps may affect the efficiency and effectiveness of collaboration.

Dependency Intensity:

Level of Dependency: The extent to which the project depends on the offshore team can impact the likelihood of risk. High dependency, especially for critical tasks, increases the risk if issues arise with the offshore team.

Experience and Expertise:

Skill and Experience of Offshore Team: The expertise and experience of the offshore team in the specific technology, domain, or project methodology can influence the likelihood of issues. Inexperienced or insufficiently skilled teams may face challenges in meeting project requirements.

Communication Tools and Infrastructure:

Technology Infrastructure: The availability and reliability of communication tools and infrastructure play a role. Issues such as poor internet connectivity or inadequate project management tools can hinder effective collaboration.

Contractual Agreements:

Clarity in Contracts: The clarity and comprehensiveness of contractual agreements with the offshore team can impact the likelihood of risk. Ambiguities in roles, responsibilities, and deliverables can lead to misunderstandings.

Project Complexity:

Complexity of the Project: The complexity of the project itself can influence the likelihood of dependency-related risks. Complex projects may require more coordination and collaboration, increasing the potential for issues.

Project Size and Scale:

Size of the Project: Larger projects with multiple dependencies on the offshore team may face a higher likelihood of challenges. Managing numerous dependencies increases the complexity of coordination.

Change Management Processes:

Change Management Practices: The presence or absence of robust change management processes can impact the likelihood of risk. Changes in project requirements or scope may pose challenges for the offshore team if not managed effectively.

Past Performance:

Historical Performance of Offshore Team: If the offshore team has a history of delivering on time and meeting quality standards, the likelihood of risk may be lower. Conversely, past performance issues may indicate a higher likelihood of future challenges.

It's essential for project managers to assess and continually monitor these factors to proactively identify and address potential issues related to offshore team dependency. Regular communication, well-defined processes, and contingency planning can help mitigate the likelihood of risks associated with dependencies on offshore teams.

In conclusion, the moderate likelihood of the risk is a result of the interplay of these factors. While challenges associated with offshore team dependencies are expected, their significance and impact depend on the specific circumstances of the project. Proactive measures such as effective communication strategies, cultural awareness programs, and thorough risk assessments can help mitigate the moderate likelihood and ensure a smoother collaboration with the offshore team in the Agile Scrum development of the *HealthyEats* project.

2.2.2 Impact

Explain the concrete impacts on the project of this risk. Use research to support the impact of this risk. (up to ½ page of 12-point text)

The impact of miscommunication due to cultural differences across teams is significantly increased in both an Agile Scrum model, as well as, in a project that centers around critical teams with offshored employment. In particular, Agile Scrum methodology depends on consistent communication of standards to focus on prioritized requirements to deliver results within a small amount of time, also known as a sprint. Not only is communication important for understanding the functionality of the system being developed but also user feedback is critical for improving the system in later iterations (Peek, 2023). Miscommunication across teams can lead, on a high level, to impacting the productivity of these teams as time will be spent by scrum masters and project managers attempting to rectify these issues. In the case of *HealthyEats* offshore teams include development and quality assurance, critical teams to the Agile Scrum process as the focus is on user feedback directly to these teams (West, n.d.).

Notably, these delays can affect sprint timelines and the quality of the end resulting project, leading to more delays in retaining the required standard before releasing each iteration for feedback. Quantifying impact that's root cause is organizational can seem difficult as the impact is quite situational. However, metrics have shown that for small businesses alone the risks of miscommunicating lead to hundreds of thousands of dollars in costs, this only scales up to tens of thousands per employee per year for larger corporations (**Aliengen, 2023**). Though these numbers scale differently for each organization, the importance of consistent communication across teams is paramount. Software development life cycles fundamentally require cross functional collaboration to see through the various steps in creating a product that can improve a business and lend to a high-quality user experience. Risking miscommunication regardless of where it stems from can come at a high cost to these goals as well as financially.

A2.2.2 (ChatGPT re Impact)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

ChatGPT: _____

Cultural differences can impact various aspects of an Agile Scrum software development team. Here are some areas where miscommunication may arise:

Communication Styles:

Verbal Communication: Different cultures may have varying preferences for directness or indirectness in communication. Some cultures may be more explicit and direct, while others may prefer a more indirect and implicit communication style.

Non-Verbal Cues: Body language, gestures, and facial expressions can carry different meanings in various cultures. Misinterpretations may occur if team members rely heavily on non-verbal cues without considering cultural nuances.

Language Barriers:

Language Proficiency: If team members have different levels of proficiency in a common language (e.g., English), misunderstandings can arise. Ambiguities or misinterpretations in requirements, user stories, or documentation may occur.

Decision-Making Processes:

Hierarchy and Authority: Cultures vary in terms of how decisions are made and the level of authority different team members may have. Some cultures may have a more hierarchical structure, while others may prefer a more egalitarian approach. This can impact decision-making processes within the Scrum team.

Conflict Resolution:

Approaches to Conflict: The way conflicts are addressed and resolved can differ across cultures. Some cultures may prefer a direct and confrontational approach, while others may value harmony and avoid direct confrontation. This can affect the resolution of issues within the team.

Time Management:

Punctuality and Deadlines: Cultural differences in attitudes toward time and deadlines can impact sprint planning, daily stand-ups, and overall project timelines. Some cultures may place a higher emphasis on punctuality and meeting deadlines, while others may have a more flexible approach.

Work Ethics:

Work Hours and Work-Life Balance: Different cultures may have varying expectations regarding work hours and the balance between professional and personal life. This can lead to misunderstandings if team members have divergent expectations regarding availability and responsiveness.

Feedback and Recognition:

Giving and Receiving Feedback: The way feedback is given and received can vary across cultures. Some cultures may value direct and constructive feedback, while others may find it more appropriate to provide positive reinforcement. Misunderstandings may occur if feedback is not culturally sensitive.

Cultural Sensitivity:

Awareness and Sensitivity: Team members need to be culturally sensitive and aware of potential differences. Lack of awareness or insensitivity to cultural nuances can contribute to miscommunication and misunderstandings.

To mitigate these challenges, teams can foster a culture of open communication, encourage cultural awareness training, and establish clear communication protocols. Additionally, creating an inclusive and diverse team environment can enhance collaboration and understanding among team members from different cultural backgrounds.

2.2.3 Risk Type

State whether the risk is primarily organizational or technical and explain. (single sentence)

The risk of miscommunication due to cultural differences stemming from a dependency on offshored employment is an organizational risk because its root cause can be both the management and culture of the work environment rather than having to do with the system or technical failures (Bishop, 2022).

2.2.4 Risk Management (1/2 page for draft)

Write an account of how you would concretely mitigate the risk. Explain whether you are proposing risk "conquest" or risk "avoidance". Consider cost of management vs. impact of risk occurring as a tradeoff. Support your plan with research. (Up to 1 page of 12-point text.)

In order to manage the offshore team dependency risk conquest mitigation strategies should be the focus. Embracing Agile principles, leveraging project management team members and tools, implementing a post pandemic workplace environment, and emphasizing continuous monitoring of project metrics all actively address the challenges that are likely to occur from offshore dependencies.

Firstly, Agile Scrum methodology is specifically tailored toward creating a standard of expectation and making sure to communicate that to all team members involved. These practices look to enhance coordination and productivity, increasing the quality of the overall team's experience working together. Inherently, the iterative process, if followed, will create working teams amongst the larger project members and create a sense of ownership as well as cohesion.

Developers and QA analysts will feel directly responsible for critical contributions to the project as they will be tasked with their own subsection to accomplish. Similarly, following iterations the user feedback will be integral to team members seeing the results of their efforts and feeling reassured that the project is on track to meeting the needs of users (Peek, 2023). Specifically in the instance of this project, this will have more chances of boosting morale as it also is the focus of the CEO's requirement for this systems development and implementation.

In addition, leveraging project management team members as well as the online tools enhances transparency and coordination. Tools already familiar to *HealthyEats* team members, such as Jira, that are similar to project management tools offshore developers and analysts will most likely be familiar with, can be used to facilitate visibility of progress and user stories. Project managers as well as scrum leaders can empower developers and analysts to use these tools to create subtasks or timelines for their specific project goals and monitor this progress without having to work in real time with offshored employees (Smelianska, 2022). Similarly, employees who work on different time zones can contribute, collaborate, and communicate goals to each other via these tools regardless of working in person.

It is important to note, workplace environments changed drastically due to the affect of the pandemic, limiting face to face interactions. While CDC regulations have updated allowing for users to work together face-to-face again, the lasting changes to workplace environments can be used to benefit teams that include offshore employees. Specifically for Agile Scrum processes, which includes several check-ins amongst smaller groups as well as less frequent but equally important meetings to review the product before releasing the iteration and plan out the sprint itself, updated tools for video conferencing and white boarding online assist greatly. Recordings of meetings can be made for employees in other time zones to accommodate their schedule, allowing them access to not only the data but also to the nuances of communication going on among the team members. Likewise, expectations have changed to prioritize flexibility and documentation to provide information to remote workers (Kim Parker, 2022). Overall, offshore teams greatly benefit from the changing values in workplace environment which foster better communication and team cohesion from afar.

A2.2.4 (ChatGPT re Risk Management)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

ChatGPT:

Mitigating risks associated with Offshore Team Dependency in an Agile Scrum methodology involves a combination of proactive strategies, effective communication, and continuous adaptation to ensure seamless collaboration between onshore and offshore teams.

1. Comprehensive Communication Plan: Establishing a robust communication plan is foundational to mitigating the risks associated with offshore team dependency. This involves defining clear channels for communication, regular meetings, and leveraging collaborative tools. Given the time zone differences, staggered stand-up meetings, asynchronous communication

tools, and documentation of discussions become crucial. This ensures that information is shared effectively, and potential issues are identified early.

2. Agile Principles and Adaptability: Agile Scrum emphasizes adaptability and embraces change throughout the development process. Applying Agile principles to offshore collaboration involves breaking down work into manageable iterations, or sprints, allowing for continuous feedback and adjustments. This iterative approach enables the team to adapt to changing circumstances, including those arising from offshore dependencies.

3. Cross-Cultural Training and Team Building: Addressing cultural differences is vital in a global Agile environment. Investing in cross-cultural training for team members fosters a better understanding of each other's working styles, communication preferences, and problem-solving approaches. Team-building activities, whether virtual or in-person, help build trust and camaraderie, facilitating smoother collaboration.

4. Distributed Agile Practices: Implementing practices specifically tailored for distributed Agile teams enhances coordination and productivity. This includes leveraging tools for virtual collaboration, version control, and continuous integration. Additionally, adopting pair programming and code reviews can help bridge the gap between onshore and offshore developers, ensuring a shared understanding of the project's codebase.

5. Risk Identification and Mitigation Planning: Regular risk identification sessions within the Agile framework are crucial. The team should actively assess potential risks associated with offshore dependencies, considering factors like time zone differences, language barriers, and task dependencies. Once identified, these risks should be prioritized, and mitigation plans should be developed collaboratively, involving both onshore and offshore team members.

6. Continuous Monitoring and Metrics: Utilizing metrics and key performance indicators (KPIs) helps monitor the health of the project and the effectiveness of the collaboration. Metrics such as sprint velocity, lead time, and defect rates can provide insights into the performance of both the onshore and offshore teams. Regularly reviewing these metrics allows the team to detect early signs of issues and implement corrective actions promptly.

7. Agile Project Management Tools: Leveraging Agile project management tools enhances transparency and coordination. Tools like Jira, Trello, or Asana provides a centralized platform for task tracking, backlog management, and real-time collaboration. These tools facilitate visibility into the progress of tasks and user stories, aiding in effective project management.

In conclusion, mitigating the risks associated with Offshore Team Dependency in an Agile Scrum methodology involves a holistic approach that combines effective communication, cultural awareness, and the application of Agile principles. By fostering a collaborative and adaptive environment, Agile teams can successfully navigate the challenges posed by offshore dependencies and deliver high-quality software products.

References other than chatGPT

Show that you used a wide variety of resources by listing them below and clearly indicating in the body above where you used. Make sure to use proper referencing in your paper. We suggest using APA format, but other formats are fine as long as they clearly distinguish your work from work of others in your response. In general, observe the stated plagiarism rules.

References

- Aliengen. (2023, October 26). *Unlock Success: The Hidden Costs of Miscommunication Between Sales and IT Teams*. Retrieved from LinkedIn: https://www.linkedin.com/pulse/unlock-success-hidden-costs-miscommunication-between-sales-teams-pwb4c?trk=public_post_main-feed-card_feed-article-content
- Bishop, K. (2022, May 18). *4 Types of Risk Categories in Project Risk Management*. Retrieved from The Ascent - A Motley Fool Service: <https://www.fool.com/the-ascent/small-business/project-management/articles/risk-categories/>
- Garbar, D. (2023, March 25). *Common Risks in Agile Projects and How to Prevent Them*. Retrieved from BelitSoft Software Development Company: <https://belitsoft.com/custom-application-development-services/agile-methodology>
- Hans, R. (2023, April 6th). *Supply Chain Disruptions: A Bad News for Businesses*. Retrieved from Deskera: <https://www.deskera.com/blog/supply-chain-disruption/#:~:text=This%20often%20results%20in%20lost,impact%20on%20the%20bottom%20line.>
- Jafari, L. (2021, April 1). *Agile, Waterfall, or RAD: Which Methodology Is Best For Your App Development?* Retrieved from WinaTalent: <https://winatalent.com/blog/2021/04/agile-waterfall-or-rad-which-methodology-is-best-for-your-app-development/>
- Kim Parker, J. H. (2022, December 9). *Pew Research Center*. Retrieved from how the Coronavirus Outbreak Has - and Hasn't - Changed the Way Americans Work: <https://www.pewresearch.org/social-trends/2020/12/09/how-the-coronavirus-outbreak-has-and-hasnt-changed-the-way-americans-work/>
- Kissflow. (2022, August 24). *What are User Stories and Why are They Important*. Retrieved from Kissflow: <https://kissflow.com/project/agile/creating-agile-user-stories/#:~:text=In%20the%20Agile%20framework%2C%20user,the%20development%20of%20the%20project.>
- Kissflow. (2023, October 11). *2023, What is Rapid Application Development (RAD)? An Ultimate Guide for*. Retrieved November 11, 2023, from Kissflow: <https://kissflow.com/application-development/rad/rapid-application-development/#:~:text=Rapid%20Application%20Development%20or%20RAD,a%20prototype%2C%20rather%20than%20planning.>

- Kissflow. (2023, 10 16). *Waterfall vs RAD vs Agile - Which Method Is The Best For You?*
Retrieved from kissflow: <https://kissflow.com/application-development/rad/rad-waterfall-agile-which-one-is-best-for-you/>
- KMS Solutions. (2023, March 7th). *7 Best Practices to Manage Your Dedicated Offshore Team Effectively*. Retrieved from KMS Solutions: <https://blog.kms-solutions.asia/best-practices-to-manage-your-dedicated-offshore-team-effectively#:~:text=Employ%20Agile%20Approaches,-The%20Agile%20development&text=Moreover%2C%20Agile%20methodology%20helps%20prioritise,zone%20differences%20and%20limited>
- Machado, D. (2023, January 20). *Why Offshore Development Fails, and How to Transform Your Odds of Success*. Retrieved from TowerHouse Studio:
<https://www.towerhousestudio.com/blog/why-offshore-app-development-often-fails-and-how-to-transform-your-odds-of-success>
- Pal, S. K. (2022, June 07). *Software Engineering | Comparison between Agile model and other models*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/software-engineering-comparison-between-agile-model-and-other-models/>
- Peek, S. (2023, October 23). *What Is Agile Scrum Methodology?* Retrieved from Business News Daily: <https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>
- Sinha, S. (2022, September 15). *Risks Involved in Agile Software Development and How to Manage Them*. Retrieved from Aspire Systems: <https://blog.aspiresys.com/software-product-engineering/risks-involved-agile-software-development-how-to-manage/>
- Smelianska, A. (2022, October 30). *Agile Offshore Development Teams: How to Work Effectively and Where to Find Them*. Retrieved from YouTeam:
<https://youteam.io/blog/how-to-work-effectively-with-agile-offshore-development-teams/>
- West, D. (n.d.). *Agile Scrum Roles*. Retrieved from Atlassian Software Development:
<https://www.atlassian.com/agile/scrum/roles>