FIT2101

SOFTWARE ENGINEERING PROCESS AND MANAGEMENT

RISK REGISTER

GROUP 2

Fahad Ahmad	30685265
Jonathan Wong Leong Shan	31435297
Lam Xin Le	31843549
Lim Zhen Kang	31886876
Matin Raj Sundara Raj	32124260
Yahoo Yang	31868282

Risk Register

Risk	Description	Estimated Impacts	High = 3 pts Mid = 2 pts Low = 1 pts	Likely = 3 pts Moderate = 2 pts Unlikely = 1 pts	Scale of Impact * Likelihood	Monitoring Strategy	Mitigation Strategies
			Scale of Impact	Likelihood	Severity		
Unreasonable product requirements	Some requirements require deep understandings in a specific area and most if not all developers in the team are not able to take the role.	Client might feel dissatisfied and require the team to seek an alternative solution. In worst case scenario, client might feel unacceptable and refuse to accept the final product.	Mid	Moderate	4	During each Sprint planning session, the team shall go over the PBIs one by one and collectively estimate the time, effort, skillsets expected to implement it and decide whether the PBI is beyond the team's proficiency	Negotiate with the client representative about the PBIs that are deemed to be impossible to implement by the team at the moment and discuss with them about alternative feasible ways of approaching the desired functionality.
Project may be delivered late	The project might be handed to the client late.	Late delivery might create emotional and physical strain on all team members which ultimately result in disharmony and disagreement in the team. Client might reject the project if it is handed in too late.	High	Likely	3	In each Sprint planning session, the team shall carefully estimate the total amount of time required to complete at least all the prioritised Product Backlog Items. During each Sprint retrospective, the team shall review the total time spent versus the amount of work completed in the Sprint Iteration in order to get a better understanding on the team's efficiency and to better estimate the amount of time and effort required in the next Sprint Iteration.	The estimation of time required, estimated in each Sprint planning session, shall be taken into consideration when deciding which PBIs should go into the Sprint Backlog. If the team is running out of time then Product Owner or Scrum Master has to put more items into each Sprint Backlog. And every developer will have to be allocated more tasks than usual.
Misunderstanding of or deviation from client requirements	Product not working as intended or not what client wants	Client might reject the final product or might be very unhappy. The team might have to re-engineer the product to meet the requirements in short bursts of intense development.	High	Moderate	6	Each of the product review sessions at the end of each Sprint Iteration is a good opportunity to present the developed product or featues to the client representative and seek immediate feedback from them.	Product Owner shall meet the client representative regularly to make sure the product development is on the right track. Any feedback from the client representative in each product review session (end of each Sprint Iteration) shall be taken seriously and make changes and adjustments accordingly in the next Sprint Iteration. Major decisions on the product features to be integrated require the Product Owner to seek double confirmation from the client representative.
Partial client requirements might not get fulfilled	Some client requirements might not get fulfilled; the final product might not contain all of the client requirements.	The final product might become unacceptable by the client if the missing requirements are important to them. It might not work as intended also because it is lacking some part of the requirements Client might be disappointed and refuse to accept the delivery of the product.	Mid	Likely	6	Using time and tasks tracker tools such as a burndown chart to keep track of the team's overall progression.	Tasks (Sprint Backlog items) allocated to each team member should be within the doer's capabilities and resources, otherwise assigning something that is difficult to the doer might drag the overall team's progression. Any team member who completed their own share of tasks, they should give a hand to the ones still struggling to speed up the progression.
Last minute requirements change	Product requirements might be changed in final rounds of Sprint Iteration.	Depending on the changed requirements, the team might have to redesign only partial or the entire product. In addition to that, the redesigned product might contain loopholes or bugs due to rapid redevelopment and hence unstable. Developers have to work extra hard in a short burst to cater to the new changes which can be very challenging in terms of energy and time.	Mid	Unlikely	2	Product Owner to present the developed product to date to the client regularly to avoid last minute requirements change as a result of deviation from client requirements from the beginning of the development. All team members are free to make predictions about the movements of external factors which might affect the success of the project in each Sprint planning session.	Highly uncertain, unclear requirements shall never be prioritised and shall never be put into the Sprint Backlog. Their specifications must be made certain and clear before the team can start to devise and work on solutions to them. Modularise Product Backlog Items such that the team can take baby steps towards a bigger goal. With this approach, the closer to the end goal, the more unlikely it is to be changed because baby steps taken earlier can either be reusable in other requirements or it can be easily discarded with little to no costs.

Risk Register

Risk	Description	Estimated Impacts	High = 3 pts Mid = 2 pts Low = 1 pts	Likely = 3 pts Moderate = 2 pts Unlikely = 1 pts	Scale of Impact * Likelihood	Monitoring Strategy	Mitigation Strategies
			Scale of Impact	Likelihood	Severity		
Lack of technical knowledge	The developer lacks fundamental knowledge required to complete the tasks assigned to them.	Unable to fulfill all Sprint Backlog Items and Product Backlog Items. The end product hence not meeting the client's requirements and may be delivered late. The product can become buggy as a result of lack of knowledge or inexperience.	High	Moderate	6	Use time tracking tools to keep track of the amount of time a developer spends on their share of work. Use process management tools such as Asana to record the frequency of a developer getting stuck when they are trying to complete a task.	Developer to speak out the lack of confidence on the technical aspects they have in each Sprint planning session so that task distributions can take that into consideration and assign a helper or assign to someone else. The experienced developers can recommend helpful resources, materials which other developers can follow along to educate themselves in that area.
Poor code design	End product contains huge bug; Not able to extend the product features easily	New changing client requirements will not be able to add into the product. Rate of fulfilment of client requirements as the project progresses into later weeks gets lower and lower. In other words, more unfulfilled client requirements are stacked, piled up. Poor usability of the product, bugs arises everywhere in the product causing the product to not behaving the expected way, clients and end users will be frustrated in using the end product.	Mid	Moderate	4	The team will hold short meetings (15 to 30 minutes) during the design phases in each Sprint Iteration so that team developers can review proposed software designs. Developers are free and encouraged to point out code smells (bad software design) in the implementation phase during the standup meetings.	The deisgn phases in each Sprint Iteration should go through extensive peer reviews (all developers in the team) revealing any potential bad design violating the software design principles. No code should be implemented before its design is finalised and agreed upon by all developers in the team. This minimizes the likelihood of coming up with a bad software design as the design has to be validated by all software developers in the team.
Poor design of system architecture	The decision on the type of system architecture made at the very beginning of the project might not be the intended solution.	This might cause the end product to not meeting the client's requirements. Developers might need to refactor and reimplement most of the implementation in another system architecture which will incur extra time and effort.	Mid	Unlikely	2	In each Sprint planning session, the team shall dedicate some time to review the system architecture as well as the code design to get an idea about the degree of misalignment between them.	During the Project Inception phase, all team members must carefully review the requirements and decide on the system architecture to use. During the design phase, the designers have to ensure the code designs and structures can be easily integrated into the current system architecture. If the major misalignment is found in later Sprint Iterations, code designs should be tweaked to conform to the existing system architecture and developers potentially have to look for thrid party libraries to cope with the shortcomings. The team should embrace instead of changing the current system architecture.
Incompetent developer	Developers take longer time than planned to pickup a new technical skill	The Sprint Backlog may not be fully completed at the end of each Sprint Iteration, ultimately resulting in late project delivery and missing of client's requirements.	High	Unlikely	3	During each standup meeting, all team members will have to answer questions which includes any struggle regarding their progression on picking up the new skills.	For a fairly new skill to most of or the entire team, there must be at least 2 to 3 team members learning from the same material/resource at the same time. In case of any of them failed to pickup the skillset within the expected timeframe, there are other backup candidiates.
Initial ideas not as good as expected	Usually during each Sprint planning session, developers will roughly know the solutions to tackle a problem which leads them to create spikes. But the spike test results might not work after actually going into the Sprint Iteration.	Developers will have to spend extra time to look for alternative solutions. This might cause inevitable change of project requirements and cause the project delivery to be delayed.	Low	Likely	3	The developer who wrote the spike shall test the feasibility of the idea preferrably after the Sprint planning session but before the start of Sprint Iteration and reflect the result to the other concerned developers.	Try to come up with not only one solution but multiple solutions to a problem so that if one of the spike ideas fail, the team can always have other alternatives to consider.

Risk Register

Risk	Description	Estimated Impacts	High = 3 pts Mid = 2 pts Low = 1 pts	Likely = 3 pts Moderate = 2 pts Unlikely = 1 pts	Scale of Impact * Likelihood	Monitoring Strategy	Mitigation Strategies
			Scale of Impact	Likelihood	Severity		
Extra hidden, unexpected costs	Project might need external apis which incur costs. This includes third-party services, libraries and API services.	If the team decides not to pay, then some features will be missing from the product resulting in the final product not meeting client's requirements. If the team decides to pay, then this will mount extra financial burden on all team members.	Low	Unlikely	1	If the service the development team is dependent on imposes a free bandwidth limit then the developer whose account is used to sign up for the service shall monitor the quota usage closely and inform the team from time to time about the remaining quota left.	Whenever the product requires external APIs services or any third party services, the team shall do due diligence in researching the available alternatives and make a thorough comparisons between them including limits on free data usage or free API calls etc. The team shall then make decisions based on those comparisons. Should the required API service or third-party service has only one supplier, the team shall decide either to negotiate with the client representative and cut the requirements or to average out the incurred costs on all team members.
Unreliable third-party services	Third party libraries might not work as intended or not documented well enough	Mismatch between the built application and other external services will cause the end product to not function as expected and might exhibit significant bugs and even not meeting the client's requirements. Poor documentation will cause developers to spend more time than necessary to understand the usage and to adapt to the technology. In worst case scenario, this will even lead to Sprint Backlog Items not fully fulfilled at the end of each Sprint Iteration ultimately leading to late project delivery.	High	Moderate	6	The developers who are responsible for utilizing the third-party service shall report to the team the feasibility and practicability of integrating the product with the service in any of the Sprint planning sessions or standup meetings.	Avoid using third-party libraries if possible, this significantly reduces the likelihood of being overly dependent on one specific third-party library which can be dangerous in the team's chosen process model (frequent changes of requirements). Before choosing an external service, the team should do due diligence in researching all available third-party services who provide the support that the team needs. Selection criteria must include ease of use of the service, size of community support the service has (are there a lot of developers using it?) as well as the versatility of the service. Versatility and size of comminity support should be prioritised in selection criteria.
Bad software engineering process management	Adopted process management model not handling the development process well.	Internal issues including conflicts, disputes, disharmony will arise and this will eventually lead to sloppy work produced and the product not meeting the client's requirements.	Mid	Moderate	4	Use external tools such as Trello and Asana to keep track of the frequency of internal conflicts arose, frequency of Sprint Backlog not fully fulfilled and the number of items not fulfilled in each Sprint Iteration.	Divide the workload evenly and fairly throughout the entire team. Make sure everyone's opinions and issues is heard. Scrum Master conducts meetings between members to coordinate and resolve conflicts. Make adjustments to process model in case majority of team members not able to adapt to any aspect of it.
Team falling behind in a single Sprint Iteration	Some Sprint Backlog items are not completed at the end of a Sprint Iteration.	The uncompleted share of work will stack with other PBIs and this will either increase the workload of the developers in the next Sprint Iteration or it will not be considered again (not meeting the client requirements). The former strains the developers while the latter dissatisfies the client.	Mid	Likely	6	Using time and tasks tracker tools such as a burndown chart to keep track of the team's overall progression in the current Sprint Iteration.	The PBIs should be feasible from a technical point of view. Never move an impractical PBI into the Sprint Backlog. Tasks (Sprint Backlog items) allocated to each team member should be within the doer's capabilities and resources, otherwise assigning something that is difficult to the doer might drag the overall team's progression. Any team member who completed their own share of tasks, they should give a hand to the ones still struggling to speed up the progression.

Risk Register

Risk	Description	Estimated Impacts	High = 3 pts Mid = 2 pts Low = 1 pts	Likely = 3 pts Moderate = 2 pts Unlikely = 1 pts	Scale of Impact * Likelihood	Monitoring Strategy	Mitigation Strategies
			Scale of Impact	Likelihood	Severity		
Development tools malfunctioning	Computer hardware or software malfunction such as virus malware, system crash or overheating	All tasks completed so far have to be redone, resulting in overdue deadlines and possibly late submission of project	High	Unlikely	3	Every team member will have to report their computer's status to any of the team members, in 1 to 2 short sentences, from time to time. For example : Hey guys, my computer is showing signs of crashing anytime soon. Please be prepared.	Record all documents and sketches on auto-save shared spaces with back-up systems such as Google Docs and Lucidchart Assign a team member to make regular checks on these documents and sketches and to resolve any problem
		All tasks assigned to the team member facing this hazard will have to be delegated to the other team members, resulting in decreased productivity and possibly overdue deadlines	High	Moderate	6		Assign the team member facing this hazard to tasks that do not involve the usage of the malfunctioned software Tasks can also be done through other electronic devices such as phones and tablets Make use of common computers available in the library or computer labs
Lack of communication between the team and the client representative	Team members might lose interest in the project and become unwilling to communicate with client representatives or product owner and this might cause insufficient product requirements and information gathered.	Submission of a project of poor, inaccurate quality with none of the desired features met, resulting in poor marks for the project Tasks need to be redone frequently, resulting in decreased productivity and possibly overdue deadlines	Mid	Likely	6	Prepare a list of questions to ask the client representative prior to the meeting. This list shall not always be empty or contain only a few items.	Hold a meeting with the client representative each week in the Tutorial slot, all team members shall attend the meeting. Make preparations prior to meeting with the client to provide updates, changes and status of project which will ensure that the project is not off track as well as to ask questions about the project requirements.
Miscommunication or disagreements among team members	Team members might not get along well with each other as the project progresses into later weeks. Team members might have lots of disputes and disagreements and not willing to cooperate with each other.	Increased awkward meetings and difficulty to assign or check tasks, resulting in decreased motivation and productivity	Low	Moderate	2	Have a messaging group for daily discussions and small talks.	Other team members should have the two upset members to talk it out and to clarify any disagreement openly at an early stage before things worsen If persistent, the two members should go for counseling in school together or attend a scheduled meeting with the demonstrator to talk things out
		Increased workload and tasks for other team members, thus decreasing productivity	Mid	Likely	6		The Scrum Master should keep the team member in check, ensuring that tasks are completed on time and issue warnings where appropriate If persistent, the issue should be brought up to the demonstrator
		More time-consuming decision-making processes, resulting in decreased productivity	Mid	Moderate	4		Voting processes can be carried out among team members to follow the majority opinion The pros and cons of each opinion can be considered and the best decision can be decided by the team leader or the demonstrator
		Some tasks are left undone which potentially results in overdue deadlines and late submission of the project	High	Unlikely	3		All tasks with brief descriptions should be listed out clearly on a shared space such as Trello to ensure everyone is clear of their task Team members who are unclear of what their tasks entail should clarify in online group chats such as on Whatsapp so that all team members will be notified