Risk assessment and mitigation

Cohort 3 Team 7 (Yetti)

Bak, Bartek
Burberry, Katelyn
Collins, Lucy
Ganley, Joe
Keegan, Josh
Kirkham, Katharine
Mayall, Daniel
Sawdon, Theo

As developing a system can be a very versatile process, prone to changes and adversities, being able to predict and mitigate risks is very important. If an issue arises that could have a big enough impact, it could completely halt development of the project. We are working within a certain timeframe, with a hard set deadline, and so this happening could be detrimental, and could result in us not meeting our targets in time, which could be a great hindrance on the customer, as they may require the software to be able to function properly by a certain point. Being able to predict and plan around risks is imperative for assuring a smooth development process, and making sure that the project is complete on time.

We could start by trying to imagine every possible scenario, and thinking of what possible setbacks we might encounter (risks). From this we could then estimate the likelihood of this happening, and the impact it could have on the development. If the risk is likely to happen, or could have a big impact on the project, we should plan around this, creating plans to ensure it doesn't happen, and contingencies, just in case it does.

To do this, we need a way to effectively organise and plan for risks. As such, we have decided to do our risk register in the form of a table. With this, we can organise what might happen and what we are doing about it, in a very clear, concise and easy to understand manner. Using a table helps us to group which actions are associated with which risks, making mitigating these risks much easier. Overall, this format of tracking risks should mean they are much easier to track, and by extension easier to deal with. As a result, any issue that arises will be less likely to negatively impact the team and the development of the project.

Key: L - Low
M - Medium
H - High

Risk (ID)	Likelihood	Impact	Consequences	Mitigation	Contingency Plan	Owner
Team member stops contributing (R000)	L	H	Certain aspects of the project simply won't get done, as everyone has already been assigned certain tasks to do.	Conduct weekly check-ins to ensure everyone is coping. Also assign a shadow member to ensure the work is being done, and to help out where possible.	Redistribute the required tasks fairly across the rest of the team.	Everyone.
Remote repository goes down (R001)	L	H	No one will be able to access files related to the project i.e. code or planning documents.	Keep backups of the repository across a few locations.	Find the most up to date local copy on someone's machine, work on that in the meantime, then consolidate changes when the repository is back up.	Everyone.
Development software in use stops working (R002)	L	M	Development on the implementation would completely halt, and those coding would not be able to effectively work.	Ensure research has been done into many different options just in case we need to swap.	Try to use a different piece of software that is known to support the current code base.	Implementation team
Hardware in use stops working (R003)	L	L	All forms of contribution towards the project would stop completely, meaning we may miss the deadline for the product.	We should each have access to backup pieces of hardware that we can fall back on if any machines fail.	Make use of publicly available hardware in the meantime (i.e. library computers, university machines etc)	Everyone
Customer changes their mind	H	M	We would have to stop development on anything that	Keep regularly meeting with the customer to	Immediately stop production on those parts	Requirements, Architecture, Method

on a certain aspect of the product (R004)	🟥	🟨	doesn’t meet the new requirements, and lose any progress and time spent on these now unnecessary components.	ensure any changes to their requirements are caught quickly, so no more time is wasted.	that do not fit the new criteria, and start planning on how to work to new constraints. Once a solid plan is formed, immediately start working towards these new goals.	Selection and Planning, and Implementation teams
Feature doesn’t fit what the customer envisaged (R005)	H	M	Similarly to if the customer changed their mind (R004), we’d lose any progress made towards this component of the project. This could have an impact on our ability to meet the customer’s deadline.	Regularly meet with the customer, and double check with them to clarify exactly what they want (even if the questions may seem redundant)	We would take the customer’s feedback into account, clarify exactly what they want and how they want it, immediately start planning how to do this instead, and start production on the newly clarified requirement.	Requirements, Architecture, Method Selection and Planning, and Implementation teams
Development library in use for the development loses support and stops working (R006)	L	M	Just like if the software itself stops working (R002), development of the code would stop, and we’d lose some progress made on the coding.	Maintain a decent knowledge on different libraries, so we can swap over if need be.	As most libraries follow a similar paradigm, refactoring code would not be too much of a challenge. We’d have to spend time changing what has already been made to work with this new library. This may take a bit of time away from	Implementation team
Another company takes legal action	L	H	This could have severe financial impacts on the project and its workers if a	Meticulously find the licenses for any asset or software used to ensure we	Cease production immediately and conform with any	Everyone

against us for the project (R007)	🟩	🟥	company sues for compensation	are able to use it as we wish. If any do not fit our use case, do not use it.	requests (if justifiable) from the other company.	
The quality of code produced is poor due to certain constraints (R008)	M	M	This could lead to a poor running product for the customer, which they may complain about and ask us to fix.	Host regular code review meetings, so we can all have input on what might be best.	Allocate more people onto the implementation team, and refactor all the code to meet a certain much better standard.	Implementation
An unexpected problem arises (R009)	L	M	As this problem is not within the planned risks, the effect could be larger as we will need to establish a plan, slowing down the project	Proactively add possible risks to the register as we think of them	Develop a plan to mitigate this new risk, and execute it in a timely manner, with input from the whole team to ensure it is the correct approach.	Risk Management Team
Deployment goes poorly (R010)	L	L	The software won’t be published properly, which could lead to issues with the client if they need to hit certain deadlines.	Ensure the deployment is effectively planned.	Analyse what went wrong, and try again. Possibly using a different method.	Everyone