**risk management**

1. **Risk Identification**

There are many potential risk sources in this project. To identify the risks, the project team should carefully review the project scope, budget and time limits, technical feasibility, etc. Identifying risks can be a repetitive and iterative process. There will be more information about the project gained and new risks will be identified as the development progresses. Furthermore, historical data from similar products and interviews with stakeholders can also be used to provide helpful opinions on potential risks.

Generally, the risks for this project can be defined in 4 types:

1. Scope risk – insufficient understanding of requirements or frequent demand changes.
2. Personnel risk – personnel is sick or resign, and no replacement can be found in a while.
3. Technical risk – a key functional problem cannot be overcame timely.
4. Management risk – insufficient coordination and execution capabilities of managers, planning deviations, process changes, and poor communication, etc.
5. **Risk analysis**

The potential risks can be described quantitatively in two categories, probability and impact. There are 5 different risk levels defined to help on identifying and analysing the risks that may occur during the system development life cycle of the project. The Risk Assessment Scales of probability and impact are shown below.

*Risk Probability Scale*

|  |  |  |
| --- | --- | --- |
| Probability Level | Definition | Probability Range |
| 5 High | A risk event that will highly probably happen. | Greater than 70% chance of happening |
| 4 Likely | A risk event that will likely happen. | 50-70% chance of happening |
| 3 Possible | A risk event that will possibly happen. | 30-50% chance of happening |
| 2 Low | A risk event that will unlikely happen. | 10-20% chance of happening |
| 1 Rare | A risk event that will rarely happen. | Less than 10% chance of happening |

*Risk Impact Scale*

|  |  |  |
| --- | --- | --- |
| Impact Level | Definition | Impact range on budget and schedule |
| 5 Significant | A risk event that will cause a significant impact to the project, and will disrupt the initial schedule and budget plan severely. | Greater than 50% above planned budget and schedule |
| 4 Major | A risk event that will cause a major impact to the project, and the team will spend plenty of additional time and budget to recover from it. | 30-50% above planned budget and schedule |
| 3 Moderate | A risk event that will have a moderate impact to the project, causes the product below the minimum level of goals. | 20-30% above planned budget and schedule |
| 2 Minor | A risk event that will have a minor impact to the project, makes the product meet the minimum level of goals. | 10-20% above planned budget and schedule |
| 1 Minimal | A risk event that will have a minimal impact to the project, easily to get recovered and does not affect the project greatly overall. | Within 10% above planned budget and schedule |

For most cases, the relation between probability and impact of a risk is negatively correlated. For instance, database cash could be a risk that rarely happens but will have a catastrophic effect on the project.

Several examples of potential risks will be presented in the Risk Matrix below.

*Risk Matrix*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Probability | Impact | Minimal | Minor | Moderate | Major | Significant |
| High | | A member takes a day of sick leave. |  |  |  |  |
| Likely | |  |  |  | Estimating and scheduling errors. | Errors occur when the product is being tested. |
| Possible | |  | A function takes more effort than expected and causes 3-day delay. | Lack of communication, causing poor clarity and confusion. |  | Database reset. |
| Low | |  |  |  | Scope creep happens. |  |
| Rare | | Electricity of a member’s workplace is cut off for a night. |  |  |  | The sponsor goes bankrupt and the project is canceled. |

*\*Bold border stands for tolerance line.*

1. **Risk plan**

There are 5 common actions can be applied when planning to deal with the risks, which are risk acceptance, risk avoidance, risk reduction, risk transfer, and risk mitigation. And completing the Risk Matrix during the risk analysis process is crucial to apply these risk planning actions.

In the first place, by using the Risk Matrix presented above, the team can spot the tolerance line of the project and decide which risks are acceptable to occur. If a risk is minor and infrequent, so that it cannot have a catastrophic impact to the project, the better decision for the team is accepting it. Considering the small probabilities and impact the risk may have, the cost of avoiding or mitigating it is too great comparing to what it affects. For example, if the electricity of a member’s workplace is cut off for a night, accepting it is obviously the better choice rather than hiring a new member for one night.

Risk avoidance is to prevent and eliminate the hazards of risk entirely. Although it is unlikely to eliminate all risks in the development cycle, the risk avoidance strategy is used to prevent as many risks as possible to avoid disruptive and costly consequences. In this case, if the team is lacking effective communication, the whole developing process could be affected greatly. Therefore, the team has to ensure the communication channel is efficient from the very beginning to avoid the impacts of miscommunication.

Comparing to risk avoidance, risk reduction focuses on reducing the likelihood and severity of a possible risk. In the process of software project management, the development cycle and productivity are often the most difficult to control. Most problems in developing a project are caused by uncertain factors. For example, in order to reduce the risks caused by an intensive scheduling, we can apply the approach of Critical Chain, and the most common measure of the approach is to reserve a certain amount of buffer time.

Risks can also be controlled by transferring risks to external people or organizations. For example, if the developers’ abilities are not capable of implementing the aimed functionalities of the software, the team can choose to outsource the project to external a developing team, as a last resort.

Like risk reduction, risk mitigation also seeks to reduce the likelihood of risks. The difference is mitigating risks is used on the risks that will happen more possibly or already happen. For example, if a more innovative, well-performed functionality requires too much cost and time to be implemented, and is almost impossible to finish, the team can decide to implemented a simplified but not so ideal version of functionality instead.