**Risk Mitigation(1), Monitoring(2) and Management(3)**

**Risk Description**

**Description of Risk m**

Business Impact Risk:

This is the risk where concern is that of the not being able to come up or produce the

product that has impact on the client’s business. If the software produced does not achieve its goals or if it fails to help the business of clients improve in special ways, the software development basically fails.

Customer Risks:

This is the risk where concern is client’s motivation or willingness in helping the

software development team. If the client fails to attend meeting regularly and fails to describe the real need of the business the produces software will not be one that helps the business.

Development Risks:

If client fails to provide al the necessary equipment for the development and execution of

the software this will cause the software to become a failure. So in other words customer has to be able to provide time and resources for the software development team. If all the requested resources are not provided to the software development team odds for the software development to fail rises greatly.

Employee Risk:

This risk is totally dependent on the ability, experience and willingness of the software development team members to create the working product. If the team members are not experience enough to use the application necessary to develop the software it will keep pushing the development dates until it’s too late to save the project. If one or more members of the software development team are not putting in all the effort required to finish the project it will cause the project to fail. Employee risk is one of the major risk to consider while designing the software.

Process Risks

Process risk involves risks regarding product quality. If the product developed does not meet the standards set by the customer or the development team it is a failure. This can happen because of the customer’s failure to describe the true business need or the failure of the software development team to understand the project and than to use proper equipment and employees to finish the project.

Product Size:

This risk involves misjudgment on behalf of the customer and also the software development team. If the customer fails to provide the proper size of the product that is to be developed it will cause major problems for the completion of the project. If software development team misjudges the size and scope of the project, team may be too small or large for the project thus spending too much money on project or not finishing project at all because of shortage of finances.

Technology Risk:

Technology risk involves of using technology that already is or is soon to be obsolete in development of the software. Such software will only be functional for short period of time thus taking away resources from the customer. Since the technology changes rapidly these days it is important to pay importance to this risk. If customer request use of software that soon to be obsolete software development team must argue the call and have to pursue customer to keep-up with current technology.

|  |  |
| --- | --- |
| **Category** | **Risks** |
| Employee Risks | Lack of training and experience |
| Process Risk | Low product quality |
| Product Size | Where size estimates could be wrong |
| Development Risks | Insufficient resources |
| Customer Risk | Customer may fail to participate |
| Technology Risk | Obsolete technology |
| Business Impact | Product may harm the business |

**Risk Mitigation for Risk *m(1)***

It is important to have mitigation plan to avoid risks once and for all. Goal is to attack the risk even before it comes into existence. The plan will help in identifying the possible risks and to monitor them.

**Product Size**

In this risk concern is of under or overestimation (mainly underestimation) of the number of Function Points. If we estimate too few LOC (line of code) necessary for the project we may get wrong cost figures which can prove fatal to the software development plan. To avoid this from happening we will use conservative figures to reduce the probability of the risk. This means we will overestimate the LOC a little. If we end up finishing the project earlier than that will not create any troubles. If the software cost estimations are passed with higher than actual cost required to deliver the product, the software development team can get credited for finishing the product sooner. Also for any reason the delivery dates get pushed back development team can still deliver the product on the time. In normal circumstances companies are not picky about the product size, so increasing the number will not cause any troubles in getting approval for the project.

Critique: Although this strategy may be reasonable in this context, It is not possible to avoid the risk of underestimation. Rather, your best estimates should be submitted with the proviso that (1) changes will cause a re-evaluation of estimates; (2) the estimates will be revisited at regular (short) intervals and readjusted if new information surfaces; (3) management provides an indication that it understands that the estimate are subject to change as the project progresses.

**Business Impact**

In this risk category we are concerned about the quality of the final product. As mitigation step we will spend more time with the users to understand their needs. This way we can gather all the information necessary for the project to be successful. We will try to understand business environment and can try to provide the user with help in defining software requirements. More the time team spends with the customer better the understating the team will have regarding the software. This will help in coming up with

just right product at right price for the customer. Team has to make sure that the palm size PC integration and the cost of the palm size PC is justified, meaning that it really improves inspection process.

**Customer (User) Risks**

If the users of the product fail to participate during the different phases of the software development we fail to recognize problems with the software. To avoid this in the mitigation phase we will try to meet the customer frequently and present software in phases so that customer and we can have better understanding the software being developed. More the time team spends with the customer better the understating the team will have regarding the software. This will help in coming up with just right product at right price for the customer. If customer fails to mention some special operations that have to performed with totally separate checklist and have to be stored separately, software development team does not know anything about it thus leaving big problem in the software.

Comment: The approach noted above is reasonable, but it is essential to get a written commitment from customer management that indicates that customer staff will be tasked with liaison with software developers. It's also necessary to inform customer management immediately if the liaison does not occur.

**Process Risks**

We want quality of the product to be as high as possible. To achieve this we will set up guidelines to be followed for each of the team members during all the phases of the software development cycle. The standard will be set and defined for all of the software development. This will help the team in delivering the high quality product thus increasing our reputation in the market. This will help bring in more clients in the future. It will also save customer from getting low quality product. For example, palm size PC checklists are easy to get used too. If inspector can not get used to the forms in the PC they may go back to using pen and paper which is not good for the reputation of the team.

**Technology Risks**

To avoid risk of using technology that may become obsolete in few years after the product have been developed. We will do excessive research on what technology to use for software development and will use the latest technology (programming languages etc.) to avoid this risk. Software development team has to make sure that the equipment requested (i.e. Palm-size PC) are current and will not be obsolete in near future.

**Development Risks**

If the necessary tools are not provided to all of the team members, their work will lack quantity and quality. As mitigation phase we will make sure that the budget includes cost for latest technology and tools needed to achieve the desired product. For example if the

government refused to deliver the Palm-size PC to the DEQ the PC integration part will be useless until the units are actually bought.

**Employee Risks (Team members)**

This risk concerns the knowledge and of the employees and their willingness to help make the project succeed. As mitigation step of this risk we will make sure that some one in all of the project development phases knows exactly what to do and the tools to use to achieve the goals. If the employees that have little knowledge in the main software implementation language fails to learn it, it may cause big problems when coding part begins.

**Risk Monitoring for Risk *m(2)***

Here we will identify the conditions to monitor to determine whether risk *m* is becoming more or less likely.

**Product Size**

To monitor the risk here, we will keep track of the amount of functions necessary for the program throughout the entire development cycle. This will tell us if the project may come across risk in future. We will keep tack of the Visual Basic code produced during the coding. We will also keep stack of the code required for the integration of the Palm- size PC. We will also be concerned about the size of the database. The number of table and number of tuple in each table. All of the above will help us in monitoring the size of the product.

Comment: The approach noted above is worthwhile, but more information should be provided. Specifically, where in the development cycle will a reevaluation of product size be done? What mechanism will be used to effect the reevaluation (e.g., a spreadsheet model)? What happens if product size appears to be increasing?

**Business Impact**

As monitoring step in this risk we will setup user meetings to show them the work that has been completed and to get user input on the work. We will have meetings every other week to present the work that has been done from the time of the last meeting. This will help team in staying in touch with the customers and will also be very efficient way to derive customer’s input on the progress made. It will also be a way to get customer insight on the project, which will help us determine the changes that we may have to make to the software upon customers request.

Comment: The approach noted above is worthwhile, but additional steps might be taken. A 2 week gap in evaluation may be too large. Possibly, interim work products

might be posted on a Web site for evaluation by customers on a more frequent basis. This would not require a meeting, but would require a time commitment by the customer.

**Customer (User) Risks**

To monitor this risk we will monitor the successes of the meeting by keeping track of people that have attended the meeting. If the out come of the meeting is low we can contact responsible person to help us overcome this problem. We will also have the login charts to show the customer who is attending the meetings and who needs to be reminded to start attending meetings.

**Process Risks**

To monitor the risk here we will review each other’s work to find the problems and to help each other in achieving better product quality. We will also have the general guidelines set for all of the work to be carried on for the software development. Software development team will constantly check each others work; will compare it with the set guidelines, and will inform a team member who is failing to participate in following the guidelines.

**Technology Risks**

For monitoring phase during the development of the software we will keep eye on new technology. This will help us to keep up with new technology. For example plan will be drawn to use multiple set of the Palm-size PC so team can see if the different brand PC with same operation system is capable of carrying the task needed to perform checklist operations.

**Development Risks**

For monitoring phase during the development of the software we will keep eye on tools being used and their effectiveness. This will help us to keep up with new technology. We will keep tack of the new equipment being brought into use at DEQ. We will also look for the availability of the inexpensive equipment that may provide help in completion of this project.

Critique: This discussion is vague and should be corrected. Specifically what development risks are likely to be encountered and specifically what can be monitored to determine whether they are becoming more or less likely?

**Employee Risks (Teammates)**

Monitoring and managing of this risk will include looking out for each other, that is if some team-member is having difficulties in performing some tasks or using particular tool or technique other members of that team will help him out. This is where team members may have to spend little time with each other learning or teaching what others know.

**Risk Management for Risk *m(3)***

Here we will identify several different software development risks and will try to create a plan to manage these risks if they do occur.

**Product Size**

After careful monitoring of the process, if we still end up with underestimation of the FP, we will put more man-hours into the project. This is the only way that we think we can manage the risk.

Comment: There are other options, if staffing is inflexible. The team could deliver less functionality or opt to deliver in an incremental fashion.

**Business Impact**

If a mistake has been made, user input on the completed work will provide us with information to fix or improve the software. We have done very many meeting with the clients and plan to do meeting every two weeks; this should clear any misunderstandings between the software development team and customers. This is the best way to go at since the work that is done on the project is revealed during the meetings and customer gets chance to make adjustments necessary.

**Customer (User) Risks**

If the turn out in the meetings is not encouraging we will pass out questionnaire to easily gather customers' view. We will ask them question rather than waiting for them to ask us questions. We will also talk to the manager at the DEQ to help us come up with plan that will increase the attendance during the meeting. If the outcomes of the meetings is satisfactory there should not be any major difficulties regarding customer risks.

Comment: See earlier comment about the possibility of posting interim work products on a Web site for evaluation by customers on a more frequent basis.

**Process Risks**

If the problem exists with the quality of the work, the quality assurance plan will be revised in the risk management phase. Other team member will attempt to take over or swap the work of the member whose work does not meet the quality standards.

**Technology Risks**

For monitoring phase during the development of the software we will keep eye on new technology. If we spot new techniques that can be implement without major changes in our project we will include such techniques in the development of the project. We will also keep a look out for major shifts in the technology and how it affects the software that we are working on. If there is a need change in the technology will be discussed among team members and will be presented to the client. If client agrees necessary changes will be made with the existing technology.

**Development Risks**

In the management phase if the funding for the technology and tools are not enough we will have to reschedule the phases of software development cycle to allow more time to coding phase. We will provide information on the several different Palm-size PC’s and will let the customer to choose the one that is most appropriate for the customer to buy. We will also make sure that the equipment is allowed to be purchased under government controls and contracts.

**Employee Risks (Team members)**

Monitoring and managing of this risk will include looking out for each other, that is if some team-member is having difficulties in performing some tasks or using particular tool or technique other members of that team will help him out. This is where team members may have to spend little time with each learning or teaching what others know. If team member lacks ability to use certain programming language or application, other team members will take some time off to teach the team member basics related to that application.

**Cases:**

**Risk: Computer Crash**

* **Mitigation**

The cost associated with a computer crash resulting in a loss of data is crucial. A computer crash itself is not crucial, but rather the loss of data. A loss of data will result in not being able to deliver the product to the customer. This will result in a not receiving a letter of acceptance from the customer. Without the letter of acceptance, the group will receive a failing grade for the course. As a result the organization is taking steps to make multiple backup copies of the software in development and all documentation associated with it, in multiple locations.

**Monitoring**

When working on the product or documentation, the staff member should always be aware of the stability of the computing environment they’re working in. Any changes in the stability of the environment should be recognized and taken seriously.

* **Management**

The lack of a stable-computing environment is extremely hazardous to a software development team. In the event that the computing environment is found unstable, the development team should cease work on that system until the environment is made stable again, or should move to a system that is stable and continue working there.

**Risk: Late Delivery**

* **Mitigation**

The cost associated with a late delivery is critical. A late delivery will result in a late delivery of a letter of acceptance from the customer. Without the letter of acceptance, the group will receive a failing grade for the course. Steps have been taken to ensure a timely delivery by gauging the scope of project based on the delivery deadline.

* **Monitoring**

A schedule has been established to monitor project status. Falling behind schedule would indicate a potential for late delivery. The schedule will be followed closely during all development stages.

* **Management**

Late delivery would be a catastrophic failure in the project development. If the project cannot be delivered on time the development team will not pass the course. If it becomes apparent that the project will not be completed on time, the only course of action available would be to request an extension to the deadline form the customer.

**Risk: Technology Does Not Meet Specifications**

* **Mitigation**

In order to prevent this from happening, meetings (formal and informal) will be held with the customer on a routine business. This insures that the product we are producing, and the specifications of the customer are equivalent.

* **Monitoring**

The meetings with the customer should ensure that the customer and our organization understand each other and the requirements for the product.

* **Management**

Should the development team come to the realization that their idea of the product specifications differs from those of the customer, the customer should be immediately notified and whatever steps necessary to rectify this problem should be done. Preferably a meeting should be held between the development team and the customer to discuss at length this issue.

**Risk: End Users Resist System**

* **Mitigation**

In order to prevent this from happening, the software will be developed with the end user in mind. The user-interface will be designed in a way to make use of the program convenient and pleasurable.

* **Monitoring**

The software will be developed with the end user in mind. The development team will ask the opinion of various outside sources throughout the development phases. Specifically the user-interface developer will be sure to get a thorough opinion from others.

* **Management**

Should the program be resisted by the end user, the program will be thoroughly examined to find the reasons that this is so. Specifically the user interface will be investigated and if necessary, revamped into a solution.

**Risk: Changes in Requirements**

* **Mitigation**

In order to prevent this from happening, meetings (formal and informal) will be held with the customer on a routine business. This insures that the product we are producing, and the requirements of the customer are equivalent.

* **Monitoring**

The meetings with the customer should ensure that the customer and our organization understand each other and the requirements for the product.

* **Management**

Should the development team come to the realization that their idea of the product requirements differs from those of the customer, the customer should be immediately notified and whatever steps necessary to rectify this problem should be taken. Preferably a meeting should be held between the development team and the customer to discuss at length this issue.

**Risk: Lack of Development Experience**

* **Mitigation**

In order to prevent this from happening, the development team will be required to learn the languages and techniques necessary to develop this software. The member of the team that is the most experienced in a particular facet of the development tools will need to instruct those who are not as well versed.

* **Monitoring**

Each member of the team should watch and see areas where another team member may be weak. Also if one of the members is weak in a particular area it should be brought to the attention by that member, to the other members.

* **Management**

The members who have the most experience in a particular area will be required to help those who don’t out should it come to the attention of the team that a particular member needs help.

**Risk: Database is not Stable**

* **Mitigation**

In order to prevent this from happening, developers who are in contact with the database, and/or use functions that interact with the database, should keep in mind the possible errors that could be caused due to poor programming/error checking. These issues should be brought to the attention of each of the other members that are also in contact with the database.

* **Monitoring**

Each user should be sure that the database is left in the condition it was before it was touched, to identify possible problems. The first notice of database errors should be brought to the attention of the other team members.

* **Management**

Should this occur, the organization would call a meeting and discuss the causes of the database instability, along with possible solutions.

**Risk: Poor Quality Documentation**

* **Mitigation**

In order to prevent this from happening, members who are in charge of developing the documentation will keep in contact with each developer on the team. Meetings will be held routinely to offer documentation suggestions and topics. Any topic deemed missing by a particular developer will be discussed and it will be decided whether or not to add that particular topic to the documentation. In addition, beta testers will be questioned about their opinion of the documentation.

* **Monitoring**

Throughout development or normal in and out of house testing, the development team and or beta testers will need to keep their eyes open for any possible documentation topics that have not been included.

* **Management**

Should this occur, the organization would call a meeting and discuss the addition of new topics, or removal of unnecessary topics into the documentation.

**Risk: Deviation from Software Engineering Standards**

* **Mitigation**

While it is possible to deviate from software engineering standards, it is unlikely to occur. All team members have a full understanding of the software process, and how we plan to implement them in the process.

* **Monitoring**

Technical reviews involving comparison between documentation and the actual project will help to determine if deviation will occur. All relevant documents must be as complete and accurate as possible to ensure that work will conform to expressed software engineering standards.

* **Management**

Should deviation occur, steps must be taken to guide the project back within the standards expressed in accompanying documents. Technical reviews help to determine what must be done to keep the project in line with established software engineering standards.

**Risk: Poor Comments in Code**

* **Mitigation**

Poor code commenting can be minimized if commenting standards are better expressed. While standards have been discussed informally, no formal standard yet exists. A formal written standard must be established to ensure quality of comments in all code.

* **Monitoring**

Reviews of code, with special attention given to comments will determine if they are up to standard. This must be done frequently enough to control comment quality. If they are not done comment quality could drop, resulting in code that is difficult to maintain and update.

* **Management**

Should code comment quality begin to drop, time must be made available to bring comments up to standard. Careful monitoring will minimize the impact of poor commenting. Any problems are resolved by adding and refining comments as necessary.