

Risk Mitigation, Monitoring and Management Plan

Introduction

Scope and intent of RMMM activities

The goal of the risk mitigation, monitoring and management plan is to identify as many potential risks as possible. To help determine what the potential risks are, GameForge will be evaluated using the checklists found in section 6.3 of Roger S. Pressman's *Software Engineering, A Practitioner's Approach* [Reference is the SEPA, 4/e, see risk checklists contained within this Web site]. These checklists help to identify potential risks in a generic sense. The project will then be analyzed to determine any project-specific risks.

When all risks have been identified, they will then be evaluated to determine their probability of occurrence, and how GameForge will be affected if they do occur. Plans will then be made to avoid each risk, to track each risk to determine if it is more or less likely to occur, and to plan for those risks should they occur.

It is the organization's responsibility to perform risk mitigation, monitoring, and management in order to produce a quality product. The quicker the risks can be identified and avoided, the smaller the chances of having to face that particular risk's consequence. The fewer consequences suffered as a result of good RMMM plan, the better the product, and the smoother the development process.

Risk management organizational role

Each member of the organization will undertake risk management. The development team will consistently be monitoring their progress and project status as to identify present and future risks as quickly and accurately as possible. With this said, the members who are not directly involved with the implementation of the product will also need to keep their eyes open for any possible risks that the development team did not spot. The responsibility of risk management falls on each member of the organization, while William Lord maintains this document.

Project Risks

Risk Identification Checklist

Product Size Risks

- **Estimated size in lines of code (LOC)**
GameForge will have an estimated 17,555 lines of code.
- **Degree of confidence in estimated size**
We are highly confident in our estimated size.
- **Estimated size in number of programs, files, and transactions**
 1. We estimate 2 programs.
 2. We estimate 10 large files for the engine, 5 large files for the user-interface.
 3. We estimate 40 or more transactions for the engine, and 20 transactions for the user-interface.
- **Percentage deviation in size from average for previous products**
We allow for a 20% deviation from average.
- **Size of database created or used**
The size of the database that we will use will be an estimated 7 tables. The number of fields will vary per table and will have an overall average of 8 fields per table. The number of records in each table will vary with the number of sprites that the user adds to the project, and the number of instances of each sprite that the user creates.
- **Number of users**
The number of users will be fairly low. There will be one user per instance of the software running, as the software is not client/server or intended for multi-user use.
- **Number of projected changes to the requirements**
We estimate 3 possible projected changes to the requirements. These will be as a result of our realization of what is required and not required as we get further into implementation, as well as a result of interaction with the customer and verification of the customer's requirements.

- **Amount of reuse of software**
Reuse will be very important to get the project started. DirectX is very simple to reuse (for the most part) and previous programs used to code for with DirectX will be reviewed and much DirectX code will be recycled.

Business Impact Risks

- **Effect on company revenue**
None, GameForge will be distributed as freeware. It will be developed using existing tools. The staff will be compensated with their term grade rather than financially. The revenue of PA Software will not be affected positively or negatively by its development and release.
- **Visibility of product to senior management**
N/A, PA software does not have an established senior management; therefore GameForge cannot be visible to them.
- **Reasonableness of delivery deadline**
Fairly reasonable. The project deadline was established before the project was undertaken. The initial planning for GameForge was executed with the deadline in mind. The scope of the project was limited to keep the project “doable” within the allowed period of time.
- **Number of customers and the consistency of their needs**
The number of customers will be fairly low. GameForge is intended as a tool to aid students in video game development. Their needs are considered consistent, as all target users will be students registered for the course CIS 587, Computer Game Design and Implementation. University of Michigan-Dearborn
- **Number of other systems/products that product must be interoperable with**
 1. Microsoft Access. More specifically the Microsoft JET Database Engine, which is included with Visual Basic and Visual C++.
 2. Microsoft Visual Basic.
 3. Microsoft Visual C++.
 4. Microsoft DirectX.
- **Sophistication of end users**
Low. The target users are novice game programmers. GameForge is designed to be easy to use, and is supplied with Wizards to guide the users through all necessary steps in video game development.

- **Amount and quality of documentation that must be produced and delivered to customer**
The customer will be supplied with a complete online help file and user's manual for GameForge. Coincidentally, the customer will have access to all development documents for GameForge, as the customer will also be grading the project.
- **Governmental constraints in the construction of the product**
None known.
- **Costs associated with late delivery**
Late delivery will prevent the customer from issuing a letter of acceptance for the product, which will result in an incomplete grade for the course for all members of the organization
- **Costs associated with a defective product**
Unknown at this time.

Customer Related Risks

- **Have you worked with the customer in the past?**
Yes, All team members have completed at least one project for the customer, though none of them have been to the magnitude of the current project.
- **Does the customer have a solid idea of what is required?**
Yes, the customer has access to both the System Requirements Specification, and the Software Requirements Specification for the GameForge project.
- **Will the customer agree to spend time in formal requirements gathering meetings to identify project scope?**
Unknown. While the customer will likely participate if asked, the inquiry has not yet been made.
- **Is the customer willing to establish rapid communication links with the developer?**
Yes, the customer is available through email, as well as in person, to all project developers.
- **Is the customer willing to participate in reviews?**
Unknown. While the customer will likely participate if asked, the inquiry has not yet been made.

- **Is the customer technically sophisticated in the product area?**
Yes. The customer trained some members of the design team in game development. He is the instructor for the course CIS 587, Computer Game Design and Implementation at the University of Michigan-Dearborn
- **Is the customer willing to let your people do their job?**
Yes. As the GameForge project is a senior design project, the customer is available if needed, but does not interfere with development operations otherwise.
- **Does the customer understand the software process?**
Yes. The customer was the instructor of CIS 375, the Software Engineering course attended by all members of the design team.

Process Risks

- **Does senior management support a written policy statement that emphasizes the importance of a standard process for software development?**
N/A. PA Software does not have a senior management. It should be noted that the structured method has been adopted for the GameForge project. At the completion of the project, it will be determined if the software method is acceptable as a standard process, or if changes need to be implemented.
- **Has your organization developed a written description of the software process to be used on this project?**
Yes. GameForge is under development using the structured method as described in part three of Roger S. Pressman's Software Engineering, A Practitioner's Approach.
- **Are staff members willing to use the software process?**
Yes. The software process was agreed upon before development work began.
- **Is the software process used for other products?**
N/A. PA Software has no other projects currently.
- **Has your organization developed or acquired a series of software engineering training courses for managers and technical staff?**
Yes. All members of the design team have attended CIS 375, Introduction to Software Engineering at the University of Michigan – Dearborn.

- **Have documented outlines and examples been developed for all deliverables defined as part of the software process?**
 Yes. The course instructor has supplied outlines for all deliverables.
- **Are formal technical reviews of the requirements specification design and code conducted regularly?**
 No. Although informal reviews are conducted.
- **Are formal technical reviews of test procedures and test cases conducted regularly?**
 No. Although informal reviews are conducted.
- **Are the results of each formal technical review documented, including errors found and resources used?**
 N/A. As formal technical reviews have not been conducted, they cannot be documented.
- **Is there some mechanism for ensuring that work conducted on a project conforms with software engineering standards?**
 No. There has been no planned method to ensure software-engineering standards will be met.
- **Is configuration management used to maintain consistency among system/software requirements, design, code and test cases?**
 Yes. The accompanying Software Configuration Management document outlines the plan for maintaining consistency among all technical documents in the GameForge project.
- **Is a mechanism used for controlling changes to customer requirements that impact software?**
 No. The customer requirements for the GameForge project are fairly flexible. The customer has allowed a great deal of freedom to the project developers. This could become a problem if the customer does change the requirements, by the likelihood of that is extremely low.
- **Is there a documented statement of work, a software requirements specification, and a software development plan for each subcontract?**
 N/A. All work is done by a single development team. No subcontracting will take place on the GameForge project.

- **Is there a procedure followed for tracking and reviewing the performance of subcontractors?**
N/A. All work is done by a single development team. No subcontracting will take place on the GameForge project.

Technical Issues

- **Are facilitated application specification techniques used to aid in communication between the customer and the developer?**
The development team will hold frequent meetings directly with the customer. No formal meetings are held (all informal). During these meetings the software is discussed and notes are taken for future review.
- **Are specific methods used for software analysis?**
Special methods will be used to analyze the software's progress and quality. These are a series of tests and reviews to ensure the software is up to speed. For more information, see the Software Quality Assurance and Software Configuration Management documents.
- **Do you use a specific method for data and architectural design?**
Data and architectural design will be mostly object oriented. This allows for a higher degree data encapsulation and modularity of code.
- **Is more than 90 percent of your code written in a high-order language?**
Yes. Code will be written in a combination of Visual Basic, C++, DirectX, with a bit of SQL.
- **Are specific conventions for code documentation defined and used?**
No. Specific conventions have not been established, but all design members have agreed to comment code as completely as possible.
- **Do you use specific methods for test case design?**
Yes. Test cases will be attempts to model existing 2-D video games such as Pong, Pacman, and possibly Super Mario Brothers.

- **Are software tools used to support planning and tracking activities?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Are configuration management software tools used to control and track change activity throughout the software process?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Are software tools used to support the software analysis and design process?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Are tools used to create software prototypes?**
 Yes. Prototypes are created using pencil and paper, as well as interface mock-ups using Microsoft Visual Basic.
- **Are software tools used to support the testing process?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Are software tools used to support the production and managing of documentation?**
 Yes. Microsoft Word will be used to support the production and management of all technical documentation.
- **Are quality metrics collected for all software projects?**
 No. No plans have been made to collect quality metrics at this time.

- **Are productivity metrics collected for all software projects?**
No. No plans have been made to collect productivity metrics at this time.

Technology Risks

- **Is the technology to be built new to your organization?**
GameForge is a software tool to aid in video game design. Development team members are familiar with game development, as well as the necessary database implementation.
- **Do the customer's requirements demand the creation of new algorithms or input or output technology?**
No. GameForge will be implemented using existing algorithms. Input and output are handled in a traditional manner.
- **Does the software interface with new or unproven hardware?**
Since GameForge is taking advantage of DirectX, the software will allow for a multitude of new or future hardware products. This is done automatically as an advantage of DX.
- **Does the software to be built interface with vendor supplied software products that are unproven?**
No. GameForge interfaces with Microsoft Access, Microsoft Visual Basic, Microsoft Visual C++, and DirectX; all are proven software products.
- **Does the software to be built interface with a database system whose function and performance have not been proven in this application area?**
No. GameForge utilizes the Microsoft JET database engine. This engine is used in Microsoft Access, and is used to develop database applications in many of the Microsoft Visual languages, including Visual Basic and Visual C++. The database system is stable and used widely.
- **Is a specialized user interface demanded by the product requirements?**
Yes. The interface is completely specialized. It is not based on anything other than every other Microsoft Windows application out. The GUI is completely our design and no other application out (to our knowledge) contains exactly what is expected of our software.

- **Do requirements for the product demand the creation of program components that are unlike any previously developed by your organization?**
 Yes. The entire GUI is composed of subsystems that our VB software engineer has never had experience with.
- **Do requirements demand the use of new analysis, design, or testing methods?**
 No. The development team will implement existing analysis, design, and testing methods for the project.
- **Do requirements demand the use of unconventional software development methods?**
 No. GameForge uses C++ code in header files, which is not unconventional. It also integrates with Visual Basic, which is not unconventional.
- **Do requirements put excessive performance constraints on the product?**
 Yes since video games push the limit of computer systems, the DirectX engine has to be efficient enough to handle large numbers of logical calculations and fast memory accessing techniques to handle the 30fps limit.
- **Is the customer uncertain that the functionality required is “doable”?**
 No. The customer has full confidence in the project as described in the System Specification Document and the Software Specification Document.

Development Environment Risks

- **Is a software project management tool available?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Is a software process management tool available?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Are tools for analysis and design available?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Do analysis and design tools deliver methods that are appropriate for the product to be built?**
 N/A. No analysis or design tools are to be used.
- **Are compilers or code generators available and appropriate for the product to be built?**
 Yes. Microsoft Visual C++ and Microsoft Visual Basic will be used to build GameForge.
- **Are testing tools available and appropriate for the product to be built?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Are software configuration management tools available?**
 No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Does the environment make use of database or a repository?**
 Yes. DirectX/C++ code is stored in header files. All other necessary files are stored in a Microsoft Access database.

- **Are software tools integrated with one another?**
N/A. No software tools are to be used therefore they cannot be Integrated.
- **Have members of the project team received training in each of the tools?**
N/A. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.
- **Are local experts available to answer questions about the tools?**
N/A. No software tools are to be used. Local experts will not be necessary.
- **Is on-line help and documentation for the tools adequate?**
N/A. No software tools are to be used. Online help will not be necessary.

Staff Size and Experience Risks

- **Are the best people available?**
Yes. PA Software has assembled a team of the most qualified software engineers to implement the GameForge project.
- **Do the people have the right combination of skills?**
Yes. The team members have experience in C++, Visual Basic, DirectX, Microsoft Access, and software development skills.
- **Are enough people available?**
Yes. Though the team is small, a larger team could take away from the productivity due to increased lines of communication.
- **Are staff committed for entire duration of the project?**
Yes. Any staff members that do not complete the project will not receive a grade for the course.
- **Will some project staff be working only part time on this project?**
No. All staff members will be working on the project for the duration.
- **Does staff have the right expectations about the job at hand?**
Yes. All team members understand what is required to complete the project, and are committed to accomplishing them.

- **Has staff received necessary training?**
Yes. Team members are familiar with game programming and software engineering techniques.
- **Will turnover among staff be low enough to allow continuity?**
Yes. Due to the nature of the staff, there will be no staff turnover.

Risk Table

Risks	Category	Probability	Impact
Computer Crash	TI	70%	1
Late Delivery	BU	30%	1
Technology will not Meet Expectations	TE	25%	1
End Users Resist System	BU	20%	1
Changes in Requirements	PS	20%	2
Lack of Development Experience	TI	20%	2
Lack of Database Stability	TI	40%	2
Poor Quality Documentation	BU	35%	2
Deviation from Software Engineering Standards	PI	10%	3
Poor Comments in Code	TI	20%	4

Impact Values:

- 1 – Catastrophic**
- 2 – Critical**
- 3 – Marginal**
- 4 – Negligible**

Risk Refinement

At various points in the checklist, lack of software tools is identified as a potential risk. Due to time constraints, the members of the design team felt that searching for and learning to use additional software tools could be detrimental to the project, as it would take time away from project development. For this reason, we have decided to forgo the use of software tools. It will not be explored as a potential risk because all planning will be done without considering their use.

Risk Mitigation, Monitoring and Management

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 from 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 basis. 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 basis. 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.

Risk Table

Risks	Category	Probability	Impact
Equipment failure	TI	70%	1
Late delivery	BU	30%	1
Technology will not meet expectations	TE	25%	1
End users resist system	BU	20%	1
Changes in requirements	PS	20%	2
Deviation from software engineering standards	PI	10%	3
Less reuse than planned	PS	60%	3
Poor comments in code	TI	20%	4