Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 09/03/2012 | V1.01 | Formatting edited to fit quality documentation | Mark Robinson |
| 13/03/2012 | V1.02 | Switched to SegFault Software branding + formatting edited | Mark Robinson |
| 13/03/2012 | V1.03 | Reordered and abstracted spec | Mark Robinson |
| 14/03/2012 | V1.04 | Modified and reformatted IO permissions | Mark Robinson |
| 14/03/2012 | V1.05 | Rewrote quality and risk assessment | Mark Robinson |
| 15/03/2012 | V1.06 | Refined formatting + amalgamated some specific requirements | Mark Robinson |
| 15/03/2012 | V1.07 | Striped down document | Mark Robinson |
| 19/03/2012 | V1.08 | Title page added, revision history finalised | Mark Robinson |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |



K.A.E Experience Centre Management System

Phase 1 Documentation

V1.08

19/03/2012

Contents

[1 Requirements Specification 1](#_Toc319932013)

[1.1 Inputs/Outputs 1](#_Toc319932014)

[1.1.1 Human 1](#_Toc319932015)

[1.1.2 System 1](#_Toc319932016)

[1.1.3 Output 1](#_Toc319932017)

[1.1.4 Validation 1](#_Toc319932018)

[1.1.5 Hardware 1](#_Toc319932019)

[1.1.6 I/O Permissions 2](#_Toc319932020)

[2 Functionality 2](#_Toc319932021)

[2.1 Processing 2](#_Toc319932022)

[2.2 Storage 2](#_Toc319932023)

[2.2.1 Products and Items 2](#_Toc319932024)

[2.2.2 Transactions 2](#_Toc319932025)

[2.2.3 Customers 3](#_Toc319932026)

[2.2.4 Staff 3](#_Toc319932027)

[2.2.5 Suppliers 3](#_Toc319932028)

[2.3 Login Requirements 3](#_Toc319932029)

[3 Performance 4](#_Toc319932030)

[3.1 Speed Requirements 4](#_Toc319932031)

[3.2 Reliability and Robustness 4](#_Toc319932032)

[3.3 Accuracy 4](#_Toc319932033)

[4 Other Requirements 4](#_Toc319932034)

[4.1 Portability Requirements 4](#_Toc319932035)

[4.2 Resource Requirements 4](#_Toc319932036)

[4.3 Ethical Requirements 4](#_Toc319932037)

[5 Software Quality Plan 6](#_Toc319932038)

[5.1 Introduction 6](#_Toc319932039)

[5.1.1 Purpose 6](#_Toc319932040)

[5.1.2 Scope 6](#_Toc319932041)

[5.1.3 Applicability 6](#_Toc319932042)

[5.1.4 Applicable Documents 6](#_Toc319932043)

[5.1.5 Dictionary of Terms 6](#_Toc319932044)

[5.2 Document Structure 6](#_Toc319932045)

[5.3 Documentation 8](#_Toc319932046)

[5.4 Development 8](#_Toc319932047)

[5.5 Implementation 9](#_Toc319932048)

[6 Software Quality Assurance Plan 13](#_Toc319932049)

[6.1 Introduction 13](#_Toc319932050)

[6.1.1 Purpose 13](#_Toc319932051)

[6.1.2 Scope 13](#_Toc319932052)

[6.2 Applicability 13](#_Toc319932053)

[6.3 Applicable Documents 13](#_Toc319932054)

[6.4 Project Management and Planning 14](#_Toc319932055)

[6.4.1 Organisation 14](#_Toc319932056)

[6.4.2 Tasks 14](#_Toc319932057)

[6.4.3 SQA Personnel 14](#_Toc319932058)

[6.5 Program Requirements 15](#_Toc319932059)

[6.5.1 Program Performance and Resource Allocation Monitoring 15](#_Toc319932060)

[6.5.2 SQA Program Audits 15](#_Toc319932061)

[6.5.3 SQA Records 15](#_Toc319932062)

[6.5.4 SQA Status Reports 15](#_Toc319932063)

[6.5.5 Software Documentation 16](#_Toc319932064)

[6.5.6 Requirements Traceability 16](#_Toc319932065)

[6.5.7 Software Development Process 16](#_Toc319932066)

[6.5.8 Project Reviews 16](#_Toc319932067)

[6.5.9 Tools and Techniques 17](#_Toc319932068)

[6.5.10 Software Configuration Management 17](#_Toc319932069)

[6.5.11 Release Procedures 17](#_Toc319932070)

[6.5.12 Change Control 17](#_Toc319932071)

[6.5.13 Problem Reporting 18](#_Toc319932072)

[6.5.14 Software Testing 18](#_Toc319932073)

[7 Risk Evaluation Matrix 19](#_Toc319932074)

[7.1 Risk Identification 19](#_Toc319932075)

[7.2 Risk Assessment 23](#_Toc319932076)

[7.3 Risk Avoidance Strategy 24](#_Toc319932077)

# Requirements Specification

This Software requirements specification (SRS) details all the requirements the Experience Centre Management System (ECMS) program should meet and the problems it must solve. It will detail all the things that the program must do and all the things it must not do.

## Inputs/Outputs

### Human

* The system must allow the user to input numeric and string data.
* The user must be prompted for a username and password
* The system must allow the user to search for information about particular stock
* The system must allow the user to update stock information
* The system must allow the user to enter a transaction
* The system must allow the user to view information
* The system must allow the user to view reports
* The system must allow the user to search product information by location.
* The system must display a list of equipment for hire at a particular park

### System

* The user must be able to view a particular park’s products, staff, customer and supplier data
* The system must display a list of equipment for hire at a particular park

### Output

* The System must be set out in a user friendly fashion (enquiring with the target audience as to what would be most effective here must be completed)
* The system graphical user interface must pertain user friendly colours e.g. Green
* Data must be presented in an organised way
* Data must be accessed via a Graphical User Interface
* Data must be viewable in report form

### Validation

* Equipment must be hired out for a minimum of 1 hour and a maximum of 1 day if booking is not made in advance
* Data must be appropriately validated

### Hardware

* Output will occur via the monitor
* A standard keyboard and 1 button mouse will be required for input

### I/O Permissions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Permission | Counter Staff | Instructor | Owner | System Administrator |
| Read | ✓ | ✓ | ✓ | ✓ |
| Write | ✓ | 🗶 | 🗶 | ✓ |
| Modify | ✓ | 🗶 | 🗶 | ✓ |
| Delete | 🗶 | 🗶 | 🗶 | ✓ |

# Functionality

## Processing

* The data store must be searchable
* Data must have the functionality to be ordered aphetically and numerically
* Time dependent data must be able to be ordered chronologically.
* The system must be able to generate weekly, monthly and yearly reports detailing the most frequently and least frequently hired items at a particular park
* The system must be able to generate weekly, monthly and yearly reports detailing the most frequently and least frequently hired items at all parks
* The system must be able to generate a report detailing the least available products and product types to hire at a particular park
* The system must be able to generate a report detailing the most available products and product types to hire at a particular park
* Orders of products must decrease the numeric value of the stock field
* Delivery of products must increase the numeric value of stock field

## Storage

### Products and Items

* The System must store product names
* The System must store multiple product types
* The product types must include bikes, helmets, canoes, paddles, water proof and climbing equipment.
* It must be possible to extend the system with new: products types, products and stock items
* The System must support products of different sizes
* The System must store available stock for each product
* Each stock item must have a unique identifier

### Transactions

* The system must record transactions
* The system must able to record the time and date of a transaction to the nearest minute

### Customers

* The system must store a customer’s names (Format: First Name/Middle Name(s)/Last Name); phone number and address (Format: House Number or Name/First Line/Second Line/Town/City/County/Postcode)
* Each customer will have a unique customer identifier

### Staff

* The system must store a staff member’s: telephone number, address (Format: House Number or Name/First Line/Second Line/Town/City/County/Postcode) and name (Format: First Name/Middle Name(s)/Last Name)
* Each member of staff will have a unique identifier

### Suppliers

* The system must store supplier: names; phone number; address (Format: First Line/Second Line/Town/City/County/Postcode)
* Each supplier will have a unique identifier

## Login Requirements

* The username must be a minimum of 6 characters and a maximum of 12 characters long
* The username must not contain any special characters or white spaces
* The password must be a minimum of 6 characters long and a maximum of 16 characters long
* The password must contain at least 1 number
* The password must not contain any white spaces
* The password must be displayed as asterisks
* The system must validate an ID when a login attempt is made
* The system must not allow access to a particular username for at least 10 minutes after 5 consecutive failed login attempts

# Performance

## Speed Requirements

* The system will respond to a log in attempt within 5 seconds 97% of the time
* The system will generate a weekly equipment hire report within 3 seconds 97% of the time
* The system will generate a monthly equipment hire report within 7 seconds 97% of the time
* The system will generate a yearly equipment hire report within 11 seconds 97% of the time
* The system will generate a list of equipment for hire at a particular park within 5 seconds 97% of the time
* The system will reorder a list of equipment for hire within 3 seconds 97% of the time
* The system will change currency within 3 seconds 98% of the time
* The system will change language within 3 seconds 98% of the time

## Reliability and Robustness

* The system will have a mean time before failure no shorter than 1000 hours
* The system will have a mean time to restore no longer than 10 minutes

## Accuracy

* Information in the data store will be 99.9% accurate excluding human input errors
* Currency conversions will be rounded down

# Other Requirements

## Portability Requirements

* The system must run on a PC Linux environment from kernel 2.6
* The system must run on a PC windows environment from Windows XP onwards
* The system must run on an Apple/Mac from OS X 10.0

## Resource Requirements

* The system must be able to run on a 2.3Ghz processor
* The system must consume no more than 2GB of RAM
* The system must consume no more than 2GB of hard drive space not including data stores

## Ethical Requirements

* The system will not use open source software without crediting the source
* The system will not be released with known or intentional bugs
* The system will be developed with legal software
* The system will not use reverse engineered code from other software
* The system will not be offensive or discriminate against any age, race or gender

# Software Quality Plan

## Introduction

### Purpose

The purpose of this software quality plan is to define the standards to which the language and roles project will be developed which will enable SegFault Software to make Experience Centre Management System a high quality project.

### Scope

This document will assure that: (1) Experience Centre Management System is a fully functioning and robust application; (2) the program suits the client’s needs; (3) documentation is of a high standard; (4) the development process cycle itself is high quality with all components acting in a professional manner.

### Applicability

This document will be used throughout the: analysis, design, implementation, testing and evaluation of Experience Centre Management System. This SQP is in effect until all deliverables including the ECMS program, documentation, installation, presentation and 2 years of phone support have been delivered to the client in their entirety.

This plan applies only to software developed by SegFault Software. Products and services provided by third parties may not abide by the same quality standards.

### Applicable Documents

Microsoft UI Guidelines: <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=2695>

### Dictionary of Terms

|  |  |
| --- | --- |
| Term | Definition |
| Code | An instruction that tells a computer how to perform a computation |
| Debugging | The process of identifying and removing errors from code |
| Open source | Computer software that is available in source code form |
| Source code | Text written using the format and syntax of a programming language |
| Bug | An error in programming code |
| Operating system | Software that supports a computer’s basic functions e.g. communicating with hardware and peripherals |
| Object oriented | A type of programming philosophy based around objects |

## Document Structure



Quality can be broken down into several categories; this plan is structured according to this breakdown.

* Documentation – These apply just to the documentation used to support the program
* Development – These standards apply to the development process itself
* Implementation – These standards apply to the ECMS program
  + Operation – Refers to the way the program functions
  + Transition – Refers to the ability to move parts of the program and interface it with other programs
  + Revision – Refers to the ease of supporting and modifying the program

## Documentation

|  |  |  |  |
| --- | --- | --- | --- |
| Standard Number | Category | Definition | Time of Implementation |
| DO1 | Consistency | All documents must show consistent facts and figures | Throughout |
| DO2 | Legibility | All documents must be written in British English | Throughout |
| DO3 | Legibility | All documentation must use proper grammar and spelling | Throughout |
| DO4 | Consistency | All documents must use Segoe UI font | Throughout |
| DO5 | Consistency | The default font size for all documentation is 11 | Throughout |
| DO6 | Comprehensiveness | All documents must be include all sections used by reputable established companies | Start of each document |
| DO7 | Explicit | All documents should contain a list of definitions for any terms not in common use | Throughout |
| DO8 | Explicit | Unnecessary technical terminology should be avoided | Throughout |

## Development

|  |  |  |  |
| --- | --- | --- | --- |
| Standard Number | Category | Definition | Time of Implementation |
| DE1 | Punctuality | All team members should attend all meetings | Throughout |
| DE2 | Punctuality / Communicativeness | Emails should be read and responded to within 24 hours | Throughout |
| DE3 | Planning | A plan that covers the entire project must be made to schedule tasks | Beginning of project |
| DE4 | Planning | The plan must be adjusted throughout the development cycle so that it consistently reflects reality | Throughout |
| DE5 | Delegating | Skills matrices must be completed by all team members to allow tasks to be delegated properly | Beginning of project |

|  |  |  |  |
| --- | --- | --- | --- |
| Standard Number | Category | Definition | Time of Implementation |
| DE6 | Responsibility | Each team member is responsible for making sure their work is compliant with all quality standards | Throughout |
| DE7 | Communication | Each member must communicate either by email or in person that they have started a piece of work to prevent duplication | Throughout |

## Implementation

|  |  |  |  |
| --- | --- | --- | --- |
| Standard Number | Category | Definition | Time of Implementation |
| I1 | Correctness | The program must match the: specification that the client provided, the software requirements specification and the conceptual and technical designs | Implementation |
| I2 | Correctness | Each test in the test plan must correspond with a point in the technical design | Testing |
| I3 | Reliability | Results, tests and configuration data must be transmitted over the network accurately | Implementation |
| I4 | Reliability | Results must be accurate to at least the nearest whole number | Implementation |
| I5 | Reliability | Graph data must be accurate to at least 3 decimal places | Implementation |
| I6 | Efficiency | Distribution of new results over the network will take at most 5 seconds per client over a 100mb/s TCP/IP network | Implementation |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Standard Number | | Category | Definition | Time of Implementation |
| I7 | Efficiency | | Distribution of new questions and answers will take a maximum 1 second per client per question/answer pair over a 100mb/s TCP/IP network | Implementation |
| I8 | Efficiency | | Distribution of new configuration options will take at most 5 seconds per client over a 100mb/s TCP/IP network | Implementation |
| I9 | Integrity | | It must be impossible to use the software without logging in | Design |
| I10 | Integrity | | Only teachers and system administrators will be able to access results and configuration data | Design |
| I11 | Usability | | The UI of the pupil section of the software must not use language that does not comply with a British reading age of 4 | Design |
| I12 | Efficiency | | The program, not including the questions and answers, must not use more than 100mb of backing store space | Implementation |
| I13 | Correctness | | The program must run on the client’s systems | Implementation |
| I14 | Usability | | The GUI must meet the Windows User Interface Guidelines (29/09/2010) | Design |
| I15 | Integrity | | The program must not introduce security flaws into the school’s IT system | Design / Implementation |
| I16 | Portability | | The software must not require any configuration to install on the client’s computer | Implementation |

|  |  |  |  |
| --- | --- | --- | --- |
| Standard Number | Category | Definition | Time of Implementation |
| I17 | Portability / Integrity | It must be possible to fully remove the program including all files | Implementation |
| I18 | Maintainability | GUI objects will use a suitable standardised prefix for the object type followed by an underscore and then an appropriate variable name | Design |
| I19 | Maintainability | All code blocks will be indented by exactly one tab | Implementation |
| I20 | Maintainability | Only one statement will occur per line | Implementation |
| I21 | Maintainability | Left hand comparisons shall be used i.e. 42==a not a==42 | Implementation |
| I22 | Testability / Flexibility / Maintainability /  Interoperability | The configuration data will be stored in a separate readable file from the executable | Design |
| I23 | Testability / Flexibility / Maintainability /  Interoperability | Questions and answers will be stored in a separate file from the executable | Design |
| I24 | Flexibility / Reusability | Class names will be named with semantic names in upper camel case ie FileInputOutput | Implementation |
| I25 | Flexibility / Reusability | Class names will not use acronyms unless it is more commonly used than its unabbreviated counterpart | Implementation |
| I26 | Flexibility / Reusability | Variable names will be semantic and in lower camel case with the first word written in lower case and subsequent words beginning with a capital letter i.e. numberOfCars | Implementation |

|  |  |  |  |
| --- | --- | --- | --- |
| Standard Number | Category | Definition | Time of Implementation |
| I27 | Maintainability / Flexibility | Variables with single character names should not be used unless scope is confined to a single code block i.e. I, j, k | Implementation |
| I28 | Flexibility / Reusability | Method names will have semantic lower camel case names similar to variable names with the first word written in lower case and subsequent words beginning with a capital letter i.e. getNumberOfCars | Implementation |
| I29 | Flexibility / Reusability | All names used within the code will fully follow standard British English language including all grammar and punctuation | Implementation |
| I30 | Maintainability / Testability | Version control will use the .NET convention of: major version, minor version, revision number and build number i.e. 1.0.5.20042 | Implementation |
| I31 | Reliability | The program must be fully tested in compliance with IEEE 829-2008 |  |
| I32 | Correctness | Review of user requirements | Throughout |
| I33 | Usability | A user manual will be included with the program | Evaluation |
| I34 | Portability | The system will be usable on machines using both Windows and Linux operating systems. | Implementation |
| I35 | Reusability / Flexibility | The system will be developed using object oriented programming techniques. | Implementation |

# Software Quality Assurance Plan

## Introduction

### Purpose

The purpose of this software quality assurance plan is to define the techniques and methodologies which will enable SegFault Software to enforce the standards required to make ECMS a high quality product. The format of this document follows the requirements of ECMS’s Software Quality Plan.

### Scope

This document will assure that: (1) Experience Centre Management System is a fully functioning and robust application; (2) the program suits the client’s needs; (3) documentation is of a high standard; (4) the development process cycle itself is high quality with all components acting in a professional manner.

## Applicability

This document will be used throughout the: analysis, design, implementation, testing and evaluation of Experience Centre Management System. This SQA is in effect until all deliverables including the ECMS program, documentation, installation, presentation and 2 years of phone support have been delivered to the client in their entirety.

This plan applies only to software developed by SegFault Software. Products and services provided by third parties may not abide by the same quality standards.

## Applicable Documents

Template used: <http://acis.mit.edu/acis/sqap/>

## Project Management and Planning

### Organisation



The ECMS Project Team consists of 4 team members, 3 software developers and 1 technical author. All these members have been put under the direct management of Mark Robinson, a Java development team leader, for the development of this project. Mark acts as the point of contact for the client and reports directly to the Erin Anttila, Executive Director of European Software Development. Mark is acting software producer for this project and so is responsible for the overall project and managing contact with any other departments as required.

### Tasks

All tasks related to the development of the software including all documentation will be managed and completed by the ECMSPT. Administration tasks such as accounting and payment processing will be handled by the appropriate departments. Interfacing with these other departments will be the responsibility of Mark Robinson.

### SQA Personnel

#### SQA Training

No additional training is expected to be required as all staff members are already sufficiently trained to deliver a quality product. In the case of new staff being recruited or contractors hired it will be ensured that they have the necessary qualifications and that they become familiar with this SQA.

#### Quality Software Developer - Training Certification

Every developer in the project team has already been assessed and provided QSD Certification by SegFault Software prior to joining the project team. This assessment is repeated annually to ensure compliance.

## Program Requirements

### Program Performance and Resource Allocation Monitoring

This will be included in the usability testing as exceeding latency limits may not make the program a technical failure however will make it difficult to use in practice.

### **SQA Program** Audits

SQA will review and approve all design documents prior to development to ensure that the proposed system fits the client’s needs and SegFault Software’s quality standards. This will include ECMS but not any third part dependencies, though dependencies shall be checked for adequate compatibility with the client’s computer systems.

#### Sched***uled*** Audits

Audits will occur at the end of each development phase before delivery and at each stage of testing the software.

#### Uns***ched***uled Audits

Unscheduled SQA audits will occur both at random and when issues arise to ensure constant compliance.

#### Audi***ts*** of the SQA Organisation

Audits of those responsible for the SQA will be completed by SegFault Software’s Internal Audit department at random and on completion of each phase before delivery. This will ensure that the project team’s SQA has been effective; results shall be delivered to the Executive Director of European Software Development and archived by the Internal Audit department.

#### Audit Reports

Audit reports contain the current status of the project, its quality level and recommended corrective actions. These reports shall be delivered to the project team and the corrective actions will be brought to the attention of the member of the team responsible for that section of the project.

### SQA Records

Audit reports will be held by both the project team and the Internal Audit department. Change and meeting logs shall be kept by just the project team and will be transferred to Internal Audit on the date of delivery of the software deliverables.

### SQA Status Reports

SQA status reports will include the current quality of the project, the current progress of the project and a summary of any SQA audits accomplished since the last status report. These reports will be delivered directly to the Executive Director of European Software Development and archived by the Internal Audit department.

### Software Documentation

SQA will review all documentation including those about ECMS and those about the development process itself.

The essential documentation includes:

* Software Requirements Specification
* Costing Analysis
* Risk Analysis
* Quality Assurance Standards and Plan
* Conceptual Design
* Technical Design (UML, data storage,)
* Test Plan
* Testing
* Project Evaluation
* Client Presentation

Software documentation must be based on well-established standards or templates.

Documents will be audited to ensure they comply with the standards and templates used. Corrective recommendations will be issued if they do not comply which will be routed to the team member responsible. Documentation audits will be held by the project team and then transferred to Internal Audit for archiving.

### Requirements Traceability

SQA will audit requirements traceability via a spread sheet matrix to ensure all requirements are met at each stage of the development cycle. This will tie requirements from the SRS to lower level designs and tests of the resulting program.

### Software Development Process

SQA will audit deliverables between each phase of the software development lifecycle. This will not preclude any other audits from being carried out.

### Project Reviews

#### Formal Reviews

All deliverables must be submitted for formal review at latest the day before they are to be delivered. Any deliverable revision submitted after this time will not be formally reviewed and so will not be delivered to the client. During formal review any discrepancies will result in modification without consultation however corrective recommendations will be sent to the team member responsible post-delivery to assist in increasing the quality of future projects.

#### Informal Reviews

##### Design Walk-throughs

SQA will be invited to any and all design walkthroughs to help ensure that the design complies with: the quality standards of the project, the software requirements specification and ensures the design process is of a high quality.

##### Code Walk-throughs

SQA will be invited to any and all code walkthroughs to help ensure that the code complies with: the quality standards of the project, software requirements specification and ensure that the code is peer reviewed.

### Tools and Techniques

SQA will assure that the quality of all program critical components does not affect the quality of ECMS. This includes third party libraries, frameworks and compilers. Case tools and tools used to create documentation need not be controlled.

### Software Configuration Management

Software configuration management is the management of the progression of the software’s definition from general concept to strict technical requirements. This ensures that all stages are client focussed and that there are no discrepancies between documents from different stages.

This will take the form of change logs and the policy that when a document is altered all dependents must be checked for consistency. Consistency shall also be checked as part of the formal audit at the end of a development phase.

### Release Procedures

Internal version control will use the .NET convention of: major version, minor version, revision number and build number i.e. 1.0.5.20042. This number will be removed prior to delivery to the client

External version control are identified by the number after the title, the first version will not have a number.

### Change Control

Change control will be managed using Git which whilst it has not been audited by SQA it is well established as a quality product.

When a release is sent to someone outside of the development team for testing purposes it will be sent with a short version description which will describe the scope of the current version, any known faults and the version number of the software.

### Problem Reporting

Any problems will be reported to the lead programmer and must include the test number that corresponds with what the user was doing, a description of the problem and the version number of the software. A copy of all problems reported will also be kept by SQA and transferred to Internal Audit on delivery of software deliverables for archiving.

### Software Testing

#### Unit Test

Unit tests are necessary to ensure that each individual class functions properly these will be carried out by the lead programmer due to the fact it is a form of white box testing and therefore requires intimate knowledge of the code.

#### Suitability Testing

Suitability testing will make sure that the functionality of the program fits the initial spec received from the client and the software requirements specification. It will be the responsibility of SQA to test this and cannot involve any team member which has programmed the software.

#### Usability Testing

Usability testing will check that ECMS is usable by young children and non-IT professionals. It will be the responsibility of the project manager to arrange these tests with an entity external to the entire project.

#### Integration Testing

This will be the last round of testing to occur and will ensure that ECMS will run on the client’s systems. This will be conducted by the programming and software installation team.

# Risk Evaluation Matrix

The Risk evaluation Matrix is used to assign priority to potential risks and show possible contingency for each.

Assumptions:

* We assume we are a company eligible for prosecution
* We assume we are responsible for the authenticity of our software

## Risk Identification

|  |  |  |
| --- | --- | --- |
| Risk | Type | Description |
| Requirements Change | Project / Artefact | The customer adds or changes requirements during the project thereby invalidating the project plan and extending development time. |
| Requirements Unmet | Project / Artefact | Requirements that are not met due to poor requirements specification or development failure. |
| Undocumented Requirements | Project | Requirements the customer expects the software to meet that have not been defined during the planning phase. |
| Insufficient Research | Project | Insufficient research by the project management. Leading to issues with competitors and the customer later in the project development cycle. |
| Poorly Defined Requirements | Project / Artefact | Requirements are defined poorly. May lead to the software not meeting the customers’ expectations or wasted development time. |
| Delayed Specification | Project | The customer may be unreachable or indecisive |
| Underestimated Scope | Project / Artefact | Feature scope is underestimated leading to backtracking and plan changes. |

|  |  |  |
| --- | --- | --- |
| Risk | Type | Description |
| Feature Creep | Project / Artefact | Features are constantly added without proper planning. |
| Hardware Unavailability | Project | Hardware being non-functional or unfit for purpose. |
| Hardware Failure | Project | Hardware failure during development may cause loss of data. |
| Asset Copyright | Project/Artefact | Copyright disputes over assets such as third party libraries and graphics. |
| Staff Turnover | Project | Staff leaving for other companies. |
| Management Change | Project | Delays while the new manager becomes situated. |
| Human Error | Project / Artefact | Human Error causes bugs and wasted development time. |
| Demotivation | Project / Artefact | Demotivated developers are more likely to make mistakes and will be less productive. |
| Death | Project / Artefact / Business | If key people related to a project die it can be catastrophic. |
| Health Concerns | Project | General illness, pregnancy or anything that takes developers away from the project due to health. |
| Overestimated Ability | Project / Artefact | Delays due to confusion and time spent gathering knowledge to complete allotted tasks. |
| Unavailability of Specialists and Contractors | Project / Business | Delays while waiting for external human resources to become available e.g. Graphic Artists. |

|  |  |  |
| --- | --- | --- |
| Risk | Type | Description |
| Missed Deadlines | Project / Artefact | Missed deadlines can cause the project to be delayed or put more stress on the development team leading to an increase in errors. |
| Unforeseen Tasks | Project | Tasks that need to be completed for the project to be complete but were missed in the planning phase. |
| Disuse of Resources | Project / Business | When assets, human or otherwise, are idle and not being used on the project as effectively as they could be. |
| Artefact Unfit for Purpose | Artefact | Due to a mistake in the design and planning phases of the project the finished artefact is not fit for the customers purposes. |
| Missed Requirements | Artefact | Requirements that were missed during the project that the customer expected. |
| Final Costs Exceeding Estimates | Artefact | Final costs are far higher than anticipated. |
| Programming Errors | Artefact | Errors in the code that have an adverse effect on the final product. |
| Unexpected Usage | Artefact | When a user uses the application in a way that was not anticipated and is not supported. |
| Broken Systems | Artefact | When broken systems within the artefact cause It to not function correctly. |
| Supporting Software Failure | Artefact | Third party libraries or systems fail or behave unexpectedly causing the artefact to malfunction. |
| Change of Operating Environment. | Artefact | The customer changes their operating environment. |
| Risk | **Type** | **Description** |
| Technology is Superseded | Business | Prior to the project being finished something else is released that is superior. |
| Competitor Launches Similar Product | Business | Competitor launches a similar product. Can cause the final costing and financial return estimates to become invalid. |
| Customer Looses Interest | Business | Part way through a project the customer decides they no longer want the product. |
| Dispute over IP | Business | When either the company or an outside party claims ownership of the intellectual property rights to the product and the company or an outside party infringes upon those rights. |
| Reduction in Project Budget | Business | Project budget is reduced. May lead to delays or fewer/lower quality resources. |
| Natural Disaster | Project/Artefact/Business | May cause the loss of the entire project. |
| Bad Weather | Project | Bad weather could make transport to and from client or the SoHo office unviable. |
| Theft of Equipment | Project | Will result in a delay as replacement equipment is found. |
| Power Failure | Project | Whilst power is absent development will be impossible |
| Unexpected Staff Absence | Project | If a staff member is absent for a day on then it will delay the project by a day |
| Staff Family Illness | Project | If a staff member’s spouse or child is ill it may result in the staff member being absent |

|  |  |  |
| --- | --- | --- |
| Risk | Type | Description |
| Corporate Espionage | Business | Competitors may attempt to sabotage the project or gain insight into the systems that make up the artefact in order to give them a competitive advantage. |

## Risk Assessment

|  |  |  |
| --- | --- | --- |
| Risk | Probability | Impact |
| Requirements Change | High | Moderate |
| Requirements Unmet | Low | Severe |
| Undocumented Requirements | Low | Severe |
| Insufficient Research | Moderate | Serious |
| Poorly Defined Requirements | Moderate | Serious |
| Delayed Specification | Moderate | Tolerable |
| Underestimated Scope | Low-Moderate | Moderate/Serious |
| Hardware Unavailability | Moderate-High | Tolerable |
| Asset Copyright | Low | Serious |
| Staff Turnover | Moderate | Moderate |
| Management Change | Low | Tolerable |
| Human Error | Near Certain | Tolerable |
| Demotivation | Moderate | Serious |
| Death | Low | Catastrophic |
| Health Concerns | High | Serious |
| Overestimated Ability | Low | Serious |
| Unavailability of Specialists and Contractors | Moderate | Tolerable/Serious |
| Missed Deadlines | High | Serious |
| Unforeseen Tasks | Low | Intolerable |
| Disuse of Resources | Moderate | Tolerable/Serious |
| Artefact Unfit for Purpose | Low | Catastrophic |
| Missed Requirements | Low | Severe/Catastrophic |
| Final Cost Exceeds Estimates | High | Tolerable |
| Programming Errors | High | Tolerable |
| Unexpected Usage | High | Moderate |
| Broken Systems | Low | Serious |
| Supporting Software Failure | Low | Severe |

|  |  |  |
| --- | --- | --- |
| Risk | Type | Description |
| Change of Operating Environment | Low | Serious |
| Technology is Superseded | Low-Moderate | Catastrophic |
| Competitor Launches Similar Product | Low-Moderate | Severe-Catastrophic |
| Customer Looses Interest | Low | Catastrophic |
| Dispute over IP | Low | Severe |
| Unforeseen Costs | Moderate-High | Intolerable |
| Reduction in Project Budge | Low | Catastrophic |
| Natural Disaster | Very Low | Catastrophic |
| Bad Weather | Low | Tolerable |
| Theft of Equipment | Very Low | Serious |
| Power Failure | Low | Tolerable |
| Staff Family Illness | Moderate | Severe |
| Corporate Espionage | Moderate | Severe-Catastrophic |

## Risk Avoidance Strategy

|  |  |
| --- | --- |
| Risk | Strategy |
| Requirements Change | By being extremely thorough in the initial design we can minimize the chances that we miss out key requirements and ensure that the requirements that we do gather from the customer are concise and clear.  To help mitigate any possible delays or damage this may cause the project it is imperative that the customer is kept aware of the project state as it develops. In this way we can catch issues earlier and save man-hours. |
| Requirements Unmet |
| Undocumented Requirements |
| Insufficient Research |
| Poorly Defined Requirements |
| Feature Creep | Feature creep will not be allowed. Proper planning and development structure will be enforced thereby avoiding this issue completely. |
| Hardware Unavailability | By putting in place level of service (LOS) agreements with technical support staff and taking proper measures to ensure our hardware receives regular maintenance and servicing we can keep hardware unavailability to a minimum. |
| Hardware Failure | It is impossible to completely protect against hardware failure but by keeping regular backups we can minimise data loss. IT will hold redundant hardware to help reduce the amount of disruption this causes. |
| Staff Turnover | This is unavoidable but we can mitigate it by taking steps to ensure our employees are happy. (e.g. Keeping reasonable work hours) |

|  |  |
| --- | --- |
| Risk | Strategy |
| Management Change | If appropriate it would minimise the effect of a management change to have the new manager start working with the team before the current manager leaves. In this way the new manager could situate themselves while the current manager is still in place and then the new manager could take over seamlessly with very little disruption. |
| Human Error | Human error is completely unavoidable. But may be mitigated through providing training courses to the staff. |
| Death | Death is unlikely to be an issue in a project but it is also completely impossible to avoid. |
| Health Concerns | Impossible to avoid but may be mitigated by accounting for illness in the initial planning phase. If a staff member or their spouse or child is ill then recovery will be aided by the private medical insurance that all employees have. |
| Staff Family Illness |
| Overestimated Ability | Should not be an issue with proper management of human resources. |
| Unexpected Usage | Proper testing of the software with controlled groups of testers (who are not developers themselves) will help to mitigate this issue. |
| Supporting Software Failure | We have no control over third party software but we can ensure that we only use mature stable software in our projects. |
| Change of Operating Environment. | We can avoid this issue by making our project completely platform neutral. But often this isn’t a possibility and we will only be able to mitigate this by properly researching the customer and making sure we support their operating environment and their future plans. |
| Technology is Superseded | There is no way of avoiding this without stooping to corporate espionage. |
| Dispute over IP | By taking the proper legal action to gain complete or partial ownership of the IP prior to starting the project we can avoid this issue. |
| Natural Disaster | We can avoid loss of data from natural disasters by keeping off-site backups. |
| Bad Weather | Impossible to avoid however in the case of it affecting a client visit the monetary cost will be covered by travel insurance. |

|  |  |
| --- | --- |
| Risk | Strategy |
| Theft of Equipment | The office has been secured by unique key cards for all employees and 24 hour security guards. To reduce the effect of loss IT holds redundant workstations. |
| Power Failure | This is unavoidable without huge expense however it will cause only a minor interruption. |
| Corporate Espionage | By keeping the project work area secure and carefully selecting who gains access we can stop most attempts at corporate espionage but we can further reduce this risk by training our employees to recognise social engineering attempts. |