**A Prototype Local Positioning Based Content Interaction Peripheral System   
    
by Luke Daniel Emery**

BSc (Hons) Computer Science

Staffordshire University

A project submitted in partial fulfilment of the award of the degree of BSc (Hons) Computer Science from Staffordshire University

Supervised by Bob Hobbs

May 2018

# Acknowledgements

# Summary

**Table of Contents**

[Acknowledgements 2](#_Toc504710580)

[Summary 3](#_Toc504710581)

[1.0. Introduction 8](#_Toc504710582)

[1.1. Problem Area Identification 8](#_Toc504710583)

[1.2. Motivations 8](#_Toc504710584)

[1.3. Proposed Solution 8](#_Toc504710585)

[1.4. Report Structure 8](#_Toc504710586)

[2.0. Project Management and Development 9](#_Toc504710587)

[2.1. Development Methodology 9](#_Toc504710588)

[2.1.1. Selecting an Appropriate Methodology 10](#_Toc504710589)

[2.1.2. Selected Methodology 13](#_Toc504710590)

[2.1.3. Methodology Role in Project Planning 17](#_Toc504710591)

[2.2. System Development Life Cycle 19](#_Toc504710592)

[2.2.1. Feasibility Study 19](#_Toc504710593)

[2.2.2. Business Study 19](#_Toc504710594)

[2.2.3. Functional Model Iteration 20](#_Toc504710595)

[2.2.4. System Design and Build Iteration 21](#_Toc504710596)

[2.2.5. Implementation 21](#_Toc504710597)

[2.3. Project Plan 22](#_Toc504710598)

[2.3.1. Resource Identification 22](#_Toc504710599)

[2.3.2. Time Scale Identification 23](#_Toc504710600)

[2.3.3. Deliverable Identification 24](#_Toc504710601)

[2.3.4. Procurement of Resources 25](#_Toc504710602)

[2.3.5. Measure of Success Identification 25](#_Toc504710603)

[2.3.6. Project Life Cycle 26](#_Toc504710604)

[2.4. Ethical Considerations 26](#_Toc504710605)

[2.5. Health and Safety Considerations 27](#_Toc504710606)

[2.6. University Department Interaction 27](#_Toc504710607)

[2.7. Version Control 27](#_Toc504710608)

[2.8. Research and Analysis Methods 27](#_Toc504710609)

[2.9. Design Methods Supported by Development Methodology 28](#_Toc504710610)

[2.10. Testing Methods 28](#_Toc504710611)

[2.11. Reflection of Project Proposal and Development 29](#_Toc504710612)

[3.0. Project Overview 30](#_Toc504710613)

[3.1. Top Down Overview Objectives 30](#_Toc504710614)

[3.2. Prioritised Objectives 30](#_Toc504710615)

[3.3. Project Scope 30](#_Toc504710616)

[3.4. Project Domain 30](#_Toc504710617)

[3.5. Success Criteria 30](#_Toc504710618)

[4.0. Preliminary Literature Review 31](#_Toc504710619)

[4.1. Research Method 31](#_Toc504710620)

[4.2. Review of Available Literature 31](#_Toc504710621)

[4.3. Reflection 31](#_Toc504710622)

[4.4. Influence on Project Plan 31](#_Toc504710623)

[5.0. Phase One 32](#_Toc504710624)

[5.1. Overview of Phase One 32](#_Toc504710625)

[5.2. Research into Problem Area 32](#_Toc504710626)

[5.3. Analysis of Problem Area 32](#_Toc504710627)

[5.3.1. Analysis Method 32](#_Toc504710628)

[5.3.2. Application of Analysis 32](#_Toc504710629)

[5.4. Design of Phase One 32](#_Toc504710630)

[5.4.1. Design Method 32](#_Toc504710631)

[5.4.2. Application of Design 32](#_Toc504710632)

[5.5. Creation and Implementation 32](#_Toc504710633)

[5.6. Phase One Modular Testing 32](#_Toc504710634)

[5.6.1. Testing Method and Strategy 32](#_Toc504710635)

[5.6.2. Testing Plan 32](#_Toc504710636)

[5.6.3. Testing Conclusions 32](#_Toc504710637)

[5.7. Reflection of Phase One 32](#_Toc504710638)

[5.8. Influence on Project Plan 32](#_Toc504710639)

[6.0. Overview of All Components 33](#_Toc504710640)

[6.1. Implementation of All Phases 33](#_Toc504710641)

[6.2. Testing of Complete Implementation 33](#_Toc504710642)

[6.2.1. Testing Method and Strategy 33](#_Toc504710643)

[6.2.2. Testing Plan 33](#_Toc504710644)

[6.2.3. Testing Conclusions 33](#_Toc504710645)

[6.3. Reflection of System 33](#_Toc504710646)

[6.4. Influence on Project Plan 33](#_Toc504710647)

[7.0. Critical Review and Further Work 34](#_Toc504710648)

[7.1. Problem Area Post-hoc Analysis and Reflection 34](#_Toc504710649)

[7.2. Project Delivery Success Identification 34](#_Toc504710650)

[7.3. Project Delivery Methodology Reflection 34](#_Toc504710651)

[7.4. Project Artefact Success Identification 34](#_Toc504710652)

[7.5. Feature Change or Addition 34](#_Toc504710653)

[7.6. Future Development Opportunities 34](#_Toc504710654)

[8.0. Evaluation and Further Work 35](#_Toc504710655)

[8.1. Professional Skill Value 35](#_Toc504710656)

[8.2. Academic Skill Value 35](#_Toc504710657)

[8.3. Development and Learning 35](#_Toc504710658)

[8.4. Alternative Project Paths 35](#_Toc504710659)

[8.5. Reflection and Evaluation of Project Success 35](#_Toc504710660)

[9.0. Graduate Exhibition (GradEx) Catalogue Entry 36](#_Toc504710661)

[10.0. Conclusion 37](#_Toc504710662)

[References 38](#_Toc504710663)

[Appendices 39](#_Toc504710664)

# Introduction

Digital content delivery is increasingly becoming a part of daily life where users of technology are receiving information on an unprecedented scale when observed against previous decades. Much of the digital information being delivered to users is based on a form of context; whether that be advertising derived from their internet usage habits, simple two or more way conversations with peers or through direct interaction with a system. As technology expands into numerous industries, the receipt of digital content in those environments becomes inevitable.

Much research and indeed product development has been conducted into tailoring information for a user based on their habits in both the virtual and physical world, utilising key points in their digital footsteps to evaluate trends and interests ensuring that any material for consumption is considered relevant based on an activity or requirement that the user may be believed to have at a given moment. The widespread adoption of contextual information has led to scenarios such as a restaurant being recommended based on a user’s physical position when navigating a city or a device recognising that a user has parked their vehicle and prompting the location of the parking space so that the user does not forget where the vehicle has been left.

The concept of positional context has also been extended into the domain of gaming, with Virtual Reality (VR) providing new opportunities to cooperate with a virtual world. The user’s position is constantly tracked and any motion is relayed as an interaction with the perceived environment in which the player is immersed. This could range from a simple interaction such as a user’s hand being tracked as it opens a virtual door to witness a new scene in the gameplay through to something more complex where the information being delivered is provided in perhaps a less direct way than the location of their vehicle being notified on a mobile device but nonetheless the information provided only occurred due to the context of the users position.

Both of the aforementioned implementations utilise common technological concepts to achieve a common goal but in vastly different environments. Through the use of sensors to determine the position of a consumer and a system to react to the parameters derived from their location, information can be provided which makes sense given the last interaction or current setting that the user may find themselves in. The scale of the two implementations differs considerably and therefore the specific technologies being utilised also differ to adapt to the given situation, for the user traversing a street it is possible to use sensors receiving a Global Positioning System (GPS) signal whereas for the user immersed in VR gameplay the limitations associated with GPS demand an alternative tracking method be applied, often determining position through a form of non-visible light radiation sensor within the local environment.

Themed entertainment is an exemplar field where technology is rapidly evolving the way that its consumers, or guests, interact with its product. The controlled environment of many themed entertainment facilities provides a unique and interesting area to explore the application of contextual based content delivery platforms. A peripheral system is proposed specifically for the local positioning of a guest navigating an attraction providing a way for them to interact and receive content based on their locational context within a themed environment; combining the immersion that is expected in a themed attraction, such as a theme park or a museum and experience in VR gameplay, with positional derived advertising as explored by the above-mentioned world scale system. Such a system would permit the interaction with a physical local environment much in the way that a user interacts in a virtual world in VR gameplay as the unique circumstances enabled by an attraction where themed exhibits consisting of screen based or physically constructed animated characters would respond to an interaction stimulus providing content to the user.

The development of such a proposed system needs to deliberate the range of available technologies for the specific and potentially complex environment. Within the specific criteria of computing, by which this project will be assessed, detailed consideration should be given to the range of sensory hardware available; the architecture of a distributed system to respond to, monitor and permit interactions; the protocols associated with networking between components; anticipated standards for Human Computer Interaction (HCI) and the user experience (UX) on specific platforms; the use of appropriate devices and methods to introduce the level of immersion expected from visual content within a themed environment. As this project is experimental in nature, there will be a period of the research phase wherein it will be necessary to identify additional gaps in knowledge that are relevant to the successful design and implementation of the desired concept. The research, design, implementation, testing and evaluation demands challenging the skillsets accrued throughout the course of undergraduate degree and the practical and professional skills developed during the undertaken placement period.

## Problem Area Identification

Within the themed entertainment industry a lucrative opportunity exists to utilise the concepts of context driven content delivery to both provide a highly immersive experience for guests and afford the attraction operator an opportunity to exploit this context to provide location derived advertisements.

In recent years the progression of technology has enabled the adoption of more computing focused solutions to be implemented in a number of museums and theme parks, these range from simple attraction map applications through to complicated distributed systems providing a virtual reality experience to a high volume of guests simultaneously. With an increasing trend in technology use in these settings guest perception shifts towards the use of systems to provide experiences that they may not be able to experience in the home.

Guests and attraction managers would benefit from a system that provides content relevant to the physical world that the visitor is experiencing at any given moment. In a museum scenario, the guests could be provided in depth content about an exhibit that would be automatically updated as they approach a display, this could be coupled with any form of content delivery that the operator choses such augmented reality to provide an interactive visual insight into the history of an object or an article that could be delivered to their device permitting them to read more about an item with the additional benefit of the digital content being rapidly deployable in the event that it should need to be updated; by updating content based on the position it would be possible for a metrics system to be included which offers the attraction management an awareness of popular areas of an attraction and could be coupled to another secondary system to provide a heat map or show in real time the location and distribution of guests. In a theme park scenario, the concept of using the position could be further extended permitting two way interaction with content. The content may react to an interaction provided by the guest, for example the press of a button to trigger an event, this would be transmittable to a variety of content providing systems that could range from a simple visual display through to an animated character that would respond to the guest when they are within its proximity, allowing for a story to be told throughout the attraction that would not be able to be delivered as seamlessly without the context of the guests position being made available to these systems, additional personal data could be utilised to deliver a very individual experience. The attraction management in this scenario would be afforded the same comprehension of guest movements but this could be coupled with an automated advertisement system recommending a nearby commercial unit within the attraction space to increase revenue opportunities. Such a content delivery system would permit the attraction operator to communicate with consumers through their devices in the event of an emergency or to inform a specific user or group of users of any information that the operator feels necessary.

As attractions age and begin to become stale the attendance may experience a decline, a solution would be require to provide an easily configurable and updateable experience that could be changed with ease to suit a change in guest habits.

Through a digital implementation, the expensive costs of updating content, providing numerous staff to man an exhibit to communicate effectively its story and attempting to predict consumer’s locations within the attraction area could all be mitigated and appeal to the guest who is seeking a new and inventive experience.

## Motivations

The primary goal of this report is to comprehensively document the creation of a system to counter and solve the problem area identified in 1.1. The solution should be completed in a fashion that clearly demonstrates an aptitude for academic work and integrates those skills, including research and communication, with an evident repository of technical and professional ability.

The project itself is intended to serve as a demonstration of the application of the above abilities in a format that is representative of a real world scenario, documenting the reasoning behind design decisions, explaining the process of interaction with features, recording the world completed and analysing the world completed whilst adhering to a project management strategy.

Specifically this artefact to be constructed as a result of the project competition is envisioned to address the use of local positioning context to generate an interaction platform for guests in the leisure industry, an extensive and exciting industry consisting of theme parks, themed environments, exhibits and museums.

It is hoped to achieve a professional, robust solution through the use of a combination of sensory hardware and a system architecture to permit a modular, extensible, configurable and customisable set of applications that would prove desirable to both an attraction operator and visitor whilst working cohesively within the installed setting.

## Proposed Solution

In addressing the problem area, several technologies have been identified that both cover the requirements of the award and suitably tackle the identified conceptual issue comprising the project domain. Whilst the issue is identified as conceptual it very much has firm roots in the real world and an eventual realisation of the solution would aim to be a market ready and valid product. This section provides a high level overview of the proposed solution.

The proposed solution would comprise of a number of systems working together to provide a coherent experience to the guest and a simple control system for the attraction operator. It is envisaged that this system would therefore embrace a distributed system architecture with soft real time elements such as the indication of a guest position and the reaction to an interaction. Such a system should also be designed in a way that the experience component can be easily exchanged allowing for the applications to be configurable to a given scenario in a rapid timeframe. As a result of this desire, the system would comprise of a number of core applications and then secondary applications would provide the customisable content delivery adhering to a known standard and accepting a certain list of commands to ensure compatibility with the core system. The content and interactions could be tailored to meet the individual accessible requirements for a range of guests.

With the system being required to work in an attraction-like setting, it would be necessary for there to be a mechanism to support multiple guests simultaneously each operating a digital device that would provide the peripheral control to interact with content systems. These guest devices would be tracked within the local environment, if they enter a specific area that is considered part of a region defined to allow interaction with a given content application the device commands should be passed to that application, as the guest navigates throughout the attraction this could result in their interaction with a number of separate applications.

In an ideal scenario devices that are specific to the attraction would be provided, this would add to the immersion into the experience by being both thematically appropriate and consistent in specification eliminating any compatibility concerns; alternatively it could be possible to provide a mobile application as many devices contain an array of sensors that could be utilised in tracking the guest but this would introduce potential exclusions of lower specification equipment. Either case presents the requirement for processing on-device sensor values to establish various elements of the guest’s location in 3D space and the need for some mobile computing program to be developed which calculates user actions in real time and relays them to appropriate peer systems whilst also receiving commands and messages from the control system and providing these messages in an understandable way to the guest.

Sensory hardware would be located throughout the attraction setting, this would be used to determine the position of the guest to a system that could make use of the information to ensure that content is being delivered appropriately and to update the attraction operator of the position of a guest. Additional sensors would be housed on the guest device to determine rotations, the fusion of this data would provide the context from which the guests experience would be derived. Some form of calibration may be required and should be incorporated into the attraction.

To allow communication between all the systems, numerous approaches could be considered. It is suggested that a central server would be responsible for ensuring that messages are transmitted to a guest device, rather than the guest device being required to store an entire attraction configuration to be aware of what other system to communicate with. Additionally the server would allow for a location that positional information is passed to and could be recorded or rerouted to analyse metrics for commonly visited locations.

An attraction operator interface would be provided, allowing the operator to visually observe guests moving through the attraction and the state of other software components. The interface would allow for the definition of areas that an interaction would take place with another system.

The content delivery systems would adhere to a designed standard and be able to process specific high priority events that could be transmitted from the control interface or routed to the system based on an interaction within its proximity. These systems would otherwise be definable and customisable by the end client as the actions performed during an interaction would be highly specific to an attraction configuration. The device hardware could range from a simple screen allowing the guest to play some form of coconut shy type game displaying a reticule on screen, determined by guest position and rotation, or a more complicated animated character that would respond to a guest’s interaction within its proximity.

## Project Scope

The project intends to produce a prototype system addressing the identified problem area, this problem and the solution specified in section 1.3. Such a system is anticipated to challenge a range of technical concepts that are considered members of the Computer Science award. This will require further, individual study of technologies not fully covered throughout the already completed stages of study.

The nature of the solution is, to a degree, experimental and challenges ability to combine academic methods with professional skills in designing a system with little prior knowledge. This will require considerable effort in prototyping, testing and researching components as they are encountered in addition to the core literature review which will be conducted to establish a firm foundation from which to start. As a result of its experimental nature and the field which the project is rooted in, this document is intended for an audience comfortable with computing terms, mathematical terms and an understanding of the technologies covered.

Areas of exploration for this project are intended to consist of sensory hardware, soft real time applications of the processed results from hardware, distributed system architecture to permit communication between several applications, mobile application development, and to a lesser degree the manipulation of graphical objects.

It is acknowledged that due to a limited personally provided budget that the scope of sensor procurement, and by extension deployment, will be limited which may or may not have demonstrable effects on the results. Resources required will be scheduled within the project plan to assess the best use of the available finances.

With time limitations, the artefact completed is intended as a prototype and not considered a ready for market product. This means that several components may not be completed to a standard considered marketable but should be completed in a manner that is indicative of professional grade software and validates the core concepts of the proposed solution as a technical demonstration. Time may also factor into the testing of alternative sensory hardware in addition the previously specified financial limitations.

Above all, the project must be conducted in a manner deemed ethical to all stakeholders; including the author, assessor, supervisor, ethics committee and British Computer Society (BCS) conforming to the requirements and code of conduct of the latter two. The full ethical statement has been submitted as required by the examination board and is made available in this document.

As, ultimately, this project is conducted under the umbrella of Computer Science; certain components of a fully immersive environment would not be expected to be reproduced in a laboratory setting. The project aims specifically to provide a prototype system that could be deployed in a setting such as a theme park or museum, the mechanics of interaction such as game play theory and immersive visual content are not intended to be implemented in the final artefact produced as they are not covered by the specific branch of the Computer Science discipline for which this study is being produced.

Areas described in the proposed solution that do not directly affect the proof of concept may be relegated to lower priorities when analysing the project. Key areas that provide a better understanding of the systems core abilities will be given precedent in the design and implementation stages.

## Report Structure

Throughout this report, the entire project will be documented from the initial identification of a proposed solution and its background through to completion and reflection. All components of the report will thoroughly explore their domain and provide both an initial expectation and final reflection on the individual body of work completed. Additional documentation pertaining to the project management, supervision arrangements, briefing and progress meetings and specific guides and maintenance information will be made available in the appendices of the report providing a holistic view of the project.

The general format will follow a chronological overview of the project, this begins within this chapter wherein the problem and motivations are described. Following chapters will be dictated in their format by the methodology, which is identified in succeeding chapters, however it is anticipated that the structure will provide a narrative of the development from the identification of gaps in knowledge, procurement of required background research and any tools, analysis of acquired information and from the analysis an interpreted design will be developed in accordance with the specifications of the applied project methodology; from this design an artefact will be created, due to the nature of the award for which this project is being completed it will likely be a software article, which then should be evaluated and critically analysed with regard to the original outset.

Acknowledging that the final structure will be defined by the applied methodology it is anticipated that the general structure will be as follows:

* Project Management
  + Wherein the methodology best suited for the nature of the project will be identified, described and an overview of its application will be provided.
* Project Overview
  + This chapter will describe how the project will be completed in relation to the proposed solution and desired methodology and framework.
* Preliminary Research
  + Within this component, areas pertaining to the identified gaps in knowledge will be thoroughly investigated and reviewed providing an understanding of how to perform the implementation stage.
* Analysis and Design
  + From the information gathered in the previous chapter, knowledge will be extrapolated to produce a model and series of designs in accordance with the methodology to produce the artefact.
* Implementation
  + Combining all previous components of the report, the implementation will document (and provide insight to) the produced software items defined by the solution and design.
* Testing
  + Professional level testing will be performed and documented to ensure that the project adheres to its specification, success criteria, design and expected professional standards.
* Critical Evaluation
  + Analysis of the work completed and observation of the performance within each area.
* Evaluation
  + Assessment of the success of the project both as a whole and in terms of each individual module.
* Further Work
  + Opportunities to develop the project further to produce a potentially marketable product.
* Conclusion
  + A comprehensive and complete narrative of the project.

# Project Management and Development

Within the motivations for this report, it was stated that the venture is intended both as a demonstration of academic principles and professional skills where it will likely be required to formulate a strategic and formulated approach to complete a project based task.

Such a professional approach would require the adherence to a recognised and appropriate method, providing a structure from which the work load can be planned and completed with relevant documentation and validation being completed and presented.

This chapter investigates the feasibility of undertaking a number of development methodologies and specifically how these methodologies would coalesce with the project management requirements of the proposed solution. It is recognised that many methodologies can be applied with greater success for certain system types than others and that the production of a system should engage a suitable framework for the specific task at hand.

The following subsections will identify what is meant by a methodology, criteria by which a method will be selected, the chosen framework, a discussion of the system lifecycle and how this method will influence the project plan.

## Development Methodology

System development is a core component within the field of computer science and by extension is a primary skill that an individual studying any branch of computing should be able to undertake to a high standard. Whilst being a demonstrable academic skill, it is also a common part of most careers relevant to the award thus having a strong and proven understanding of the concepts relating to the creation of a computer-based information system will likely prove a valuable trait of any graduate.

A variety of approaches may be undertaken in the development of a system for an assortment of reasons however it is identifiable that the benefits of a formalised methodology with its specific set of tools and techniques enhance the development process and the end product (Avison and Fitzgerald, 2006, p.570). The desire for development methodologies stemmed initially from the requirement of organisations to better manage the manufacture of computer applications in contrast to the earlier ethos of development being driven by programming and later maintenance (Avison and Fitzgerald, 2006, pp.576-577) these efforts resulted in the first identifiable system life cycle model. It is the same determination for management of the development process that is the reason for utilising, and adhering to, a methodology in the production of the Final Year Project artefact and the associated body of work.

Methodologies encompass a number of elements in the development of software and provide a prescriptive guide to progression; although different methodologies are appropriate for different application requirements and can vary in scope covering different parts of the process generally the areas that a methodology defines are as follows:

* The identification of phases
* Procedures
* Tasks
* Rules
* Techniques (models to conceptualise systems, tables to describe decision making, pseudocode)
* Tools (modelling software, task management software, database management)
* Documentation
* Organisation of the approach
* Validity of design
* Stakeholders (all those with any level of involvement with the system from conception through to operation and maintenance)

A division of methodologies, encountered with the advent of Agile style development, frameworks provide a less comprehensive alternative development strategy and are summarised by Avison and Fitzgerald (2006) as providing guidance to the developer when choosing applicable methods, techniques and tools.

Explaining why to use a methodology varies depending on the situation; the benefits can be viewed from the perspective of the client, managers for the project, financial managers within an organisation and developers. The methodology can be viewed as a factor in complexity management, providing a disciplined approach to the analysis and design procedures (Williams, 2006) and provides a method of coordination when teams are involved in the development process (Britton and Doake, 2006). From a managerial perspective methodologies provide the ability to view the development process as a series of small tasks aiding the planning and monitoring of development progress (Britton and Doake, 2006), this level of transparency in the development is beneficial when considering budgetary constraints and client demands.

As a result of the above ideas, the methodology provides a platform from which many stakeholders in the system may gain information therefore it should be ensured that the methods are understood by the entirety of the team (Williams, 2006).

The objective when using a methodology is to provide a detailed plan for the production of a system, this is achieved by following the steps identified by the methodology and making use of the set solutions to given stages, the stages being derived from a life cycle model of the complete system in accordance to the specific priorities of a given method or framework. The solutions for each step may describe specific tasks to be undertaken, the order in which tasks should be approached and documentation requirements (Britton and Doake, 2006, p.17).

It has been noted that certain application classifications, such as real-time systems or distributed systems, tend to be developed under a specific set of methodologies and indeed different methodologies may lend their structure better to a given set of requirements (Britton and Doake, 2006, p.18); although arguments exist both for and against this philosophy with some examples demonstrated later in this section.

In the previous chapter of this report, it was specified that the proposed solution comprised of several systems working in unison. Additionally, the core technology is time dependant in the delivery of its process. These two factors are indicative of a system that could be classified as being real-time (firm real-time due to there being no critical issue if there is a time related failure but it would degrade the overall performance of the system) and distributed as the multiple applications all contribute to the overall performance of the system. Notably a significant time constraint exists in the production of the project and as a result this indicates that a methodology should be approached that does not treat the time constraint as a variable. Much of the proposed system embraces a number of technologies in which the author is not well versed, in this case whilst the high-level requirements and functionality of the system can be easily identified, in the early stages of the project development the more specific attributes associated with functionality will be subject to background research and literature review.

### Selecting an Appropriate Methodology

As outlined previously in this chapter, different methodologies are intended for different use cases. The features of each methodology or framework aid in defining how suitable it may be when mapped against the objectives of a certain type of program. The focus of a method can generally be identified by the breakdown of the task structure, the specific models required and what elements are considered variable in the development process, this results in a list of methods that may be considered appropriate for a given system proposal (Britton and Doake, 2006, p.18).

Conventional development methods, such as the Systems Development Life Cycle and later the Waterfall Model, specific stages in the process of creating a software solution (Britton and Doake, 2006, pp.10-11):

* Problem Definition
* Feasibility Study
* Requirements Engineering
* Design
* Implementation
* Maintenance

Each of the identified steps within the cycle contain defined requirements and tasks to be undertaken, the entire process is undertaken sequentially meaning that each step has to be completed before the next stage is permitted to be undertaken (Avison and Fitzgerald, 2006, p.577). Due to these types of methodologies having matured significantly, being conceived in the 1970’s and 1980’s, there is plenty of documentation and instruction in conducting a project to these plans however they have drawbacks associated with the linear flow of the process.

The phased approach to the system development has proven to be a successful approach and is undertaken by numerous methodologies; the early models identified several steps that should be completed, by understanding the tasks to be completed in each phase this significantly improves the knowledge of time scale and resource requirements. Following the completion of each stage the team may review the progress at that point and manage the expectations and requirements moving forward.

When a project must be conducted in a linear fashion it does not provide the ability for any changes to be made either by the client or developers as lessons are learned throughout the development process. This results in minimal scope for reflection based enhancements to be made as the requirements of the system are formally established at the earliest stages of the development. Whilst this could be argued to be a positive in that changes are not made that could require dramatically reengineering the proposed solution, it does not permit for improvements to be properly applied throughout the life cycle. Developers may be required to spend a significant portion of the time establishing a strong design based on the problem area, although project management solutions may be able to account for this, where the application consists of technologies and techniques that are unknown to the creators. The single pass through of the steps also presents significant issues where the problem area may not operate in a fixed sense; when the initial analysis is being conducted to determine the requirements and model these into a system it may not be possible to capture every possible dynamic that the problem area experiences, this can later result in a system that cannot cope with the actual processes that may have changed significantly since the early stages of the method were conducted. Methods to handle the above issues may include extending the time frame in which the project may be delivered or alternatively producing a system that is considered incomplete. As the aforementioned models only permit a single point of implementation this could result in being the only opportunity where a client views the product, it is possible that several concerns may arise at this point presenting a challenging work load for the maintenance stage where many changes would have to be instigated (Avison and Fitzgerald, 2006,pp.38-43).

Whilst it is acknowledged that the benefits of a staged process improve the system, from the evidence presented throughout this chapter, the fixed structure of some methodologies do not necessarily lend themselves to projects which consist of numerous unknown attributes and are subject to static time limits. Alternative methods and frameworks exist which embrace the dynamic elements of certain projects and are designed to make time for reflection based learning to be implemented into the project life cycle. The focus of these types of methodologies is in the creation of parts of the system and then to incrementally and iteratively enhance the product until it can be considered a success against the identified criteria.

Due to the real-time and distributed nature of the proposal, research into the methods and frameworks targeted towards this type of development was conducted. It was noted early in the research stage into this area that there are not a significant amount of methodologies designed specifically with the production of this system classification in mind with examples being: Structured Analysis/ Structured Design, Concurrent Design Approach for Real-time Systems, Finite State Methods and Object Oriented Design (OOD) which focus most heavily on the architecture of the application (Williams, 2006, p.3) and may be applied within a documented development methodology, potentially proving to be important later in the design process. Critical systems, under which many fall into the real-time category, are required to conform to certain standards applicable to the industry for which the software is being developed and although the standards often indicate a framework under which the development should take place it often does not provide a prescriptive method to follow.

Primarily it was identified that an iterative and incremental development method would be beneficial to address the requirement to learn certain technologies throughout construction whilst retaining a formalised structure with an emphasis on the quality of the product and covering all stages of the development, rather than specifically relating to the modelling or implementation phases. As time scales are non-negotiable, an approach would have to be chosen that would ensure that a demonstrable artefact came into existence at the earliest possible stage and is accompanied by comprehensive documentation.

The above requirements imply that a form of Rapid Application Development (RAD) would be beneficial. The term rapid application development is an umbrella containing a variety of different methods and frameworks, including a methodology named Rapid Application Development which was created in 1991 (Avison and Fitzgerald, 2006,p.128), other methods include the Agile approach. Main philosophies incorporated into RAD methodologies include the acknowledgement that not all requirements may be known in the early stages of system development and may only become apparent after a working system has been created and recognises that iterative and time specified development stages may handle disparities in the initial and final requirement definitions. This permits the requirements and functionality of the system to be tailored throughout the life cycle to better reflect the desire of the end user and abilities of the developer within the constraints provided (Avison and Fitzgerald, 2006,p.129).

Initial investigations found that there was an element of conflict when applying RAD methods to the development of systems with a real-time element, with some sources noting that a low percentage of real-time systems are developed under an Agile framework (Vijayasarathy and Turk, 2008) and others identifying RAD to be inappropriate for real-time applications (Centers for Medicare & Medicaid Services, 2008). Other sources have demonstrated well established companies completing a real-time type system whilst utilising a subset of the RAD approach to software development, Extreme Programming (XP), and demonstrating positive results in the utilisation of an object oriented language in conjunction with the strategy (Bowers *et al.*, 2002); a further example goes on to make use of another subset of RAD, Dynamic System Development Method with different features (specifically designed to overcome known issues with RAD) and again demonstrates high satisfaction results in the finished product combined with delivery within the specified time scale but acknowledges this was at the expense of higher client costing (Barritt, 2002, p.7).

Owing to the real-time nature components of the proposed solution in conjunction with the fixed time frame for development and potentially changeable nature of the system functionality as more is learnt in relation to the technologies to be utilised it was decided that a subset of the RAD approach would be preferential rather than a traditional and non-iterative alternative.

### Selected Methodology

The Dynamic System Development Method (DSDM) was identified, amongst a variety of RAD derivatives, as a contending methodology due to the iterative and incremental approach acknowledging the progressive understanding of the functionality to be developed.

***FIGURE GOES HERE AS A DECISION TABLE RAD VS TRADITIONAL***

***FIGURE GOES HERE AS A DECISION TABLE DSDM VS OTHER RAD***

Under the process of DSDM the development life cycle is compartmentalised into five distinct stages (Stapleton and Consortium, 1997, p.3). These stages comprise of:

* Feasibility Study
* Business Study
* Functional Model Iteration
* System Design and Build Iteration
* Implementation

Throughout the system development, there are recognised points at which it is possible to return to an earlier phase to further evolve the application; the initial order through the method is described above but it is possible from the Design and Build Iteration to move back to the Functional Model Iteration and from the Implementation stage it is possible to return to any of the earlier phases. The system life cycle is more comprehensively explained in Section 2.2. Within each phase testing is performed to ensure that the project is delivered to industrial quality, the aim of DSDM as a method was to eliminate the perception of “quick and dirty” which is associated with traditional RAD (Stapleton and Consortium, 1997, p.55).

DSDM has, through a number of case studies, been identified as a suitable method by several large companies including: Orange, Sema Group and British Telecoms (Stapleton and Consortium, 1997).

The method is described in nine principles which form the foundation of DSDM and provides a framework from which the development process can be worked to. The framework provides the ability to be tailored to the requirements of the development practices and of the project:

* Active User Involvement
  + A proactive approach to evolving the requirements of the system throughout development
  + Review and reflection on progress of the application as it is created
  + Frequent review of the work leads to accurately meeting the requirements
  + By consulting stakeholders often assumptions can be eliminated
  + The consumer being consulted should be senior enough to have an overall view of the entire project aim whilst maintaining a knowledge of the required processes
* Development Team Empowerment
  + Developers should be permitted to make decisions in accordance to the design
  + Prevent costly use of time by consulting numerous levels of management or supervision
  + Guidelines should be set to ensure that the decisions do not reengineer the proposed solution
  + Decisions regarding priority of implementation (following the prescribed high level priorities) and technical solutions should be carried out by developers
* Focus on frequently achieving a deliverable
  + Produce visible and tangible elements of the system for review and reflection
  + Elements may be software, models, designs or documentation
* Fitness for purpose for acceptance of deliverables
  + Rapidly produce elements of the product that demonstrate requirements
  + Other activities that may hard time scales should be reconsidered and delivered at a later stage
* Iterative and Incremental development
  + Allow for dynamic changes to embrace the evolution of the system
  + Perform course corrections to meet new requirements as the system is better understood
* All changes are reversible
  + It should be acknowledged that an iteration may be the incorrect branch of a concept and as a result it may be necessary to back track
  + Due to frequent reviews and reflections this should not result in the loss of large amounts of work
* Requirements are described as a high level
  + Requirements that are identified during the business study describe the scope of the project
  + Detailed requirements can be elicited throughout iterations as more knowledge is gained
* Testing throughout the life cycle
  + As components of the system are potentially developed at different times individual, incremental testing of each component should be performed at the end of phases throughout the project
  + Integration testing should be performed as the system reaches a point at which components can be integrated
* Collaboration and cooperation between all stakeholders
  + All individuals who are associated with the system should work together, sharing information and reviewing phases
  + Interdepartmental communication to ensure that all requirements are understood and met
  + Compromising deliverables if constraints affect work load

DSDM is prescribed for use under specific set of circumstances; namely scenarios where the end product may be interactive with functionality visible via some sort of interface, there is a clearly defined target audience, is not entirely constructed from complex components, can be broken into smaller functioning elements, are time constrained and have high level objectives (Stapleton and Consortium, 1997, pp.19-25). The proposed solution for this project can be directly plotted to the circumstances identified by the DSDM Consortium as the project consists of an interactive content delivery platform allowing the end user to perform actions through their movement to affect the interface, the application is intended for use within the themed entertainment industry, whilst components may contain computational technology with which the developer is not well versed there are also significant components that are not complex and lend themselves to rapid development, the system is intended to be distributed and thus is separated into multiple components with specific functionality and responsibility, the project must be completed by the deadline enforced by the module and the general functionality is known but greater detail is intended to be developed alongside the research and development of the project respectively. For these reasons DSDM was ultimately chosen as the method of development.

***TABULAR EXPLAINATION OF THE ABOVE.***

The framework of DSDM, in addition to identifying phases of progress, makes use of several development concepts for the creation of the application. The method embraces the use of timeboxing (a strategy seen in other RAD methodologies where individual development iterations consist of Analysis, Design and Implementation) to develop core functionality first and achieve the highest level of benefit to the entire system (Avison and Fitzgerald, 2006, p.129) this is in contrast to conventional development where all system analysis must be completed at the early stages of the process. Additionally, the objectives should be prioritised under the rules of MoSCoW ensuring that the development is not hampered by a wish list of features and instead ensures that the most important functionality can be implemented early on. Joint Application Development (JAD) workshops involve stakeholders in the system meeting to facilitate information gathering and requirement agreements and the use of prototyping within the phased structure provides a base from which the reflection and review process can begin (Avison and Fitzgerald, 2006, p.477).

MoSCoW identifies the priorities of the system under development and represents the following levels:

* Must Have
  + The minimum functionality
* Should Have
  + Important requirements but the system is still usable without their implementation
* Could Have
  + Features that may be beneficial but can be more easily excluded from the incremental level of development
* Want to implement but will not be completed in this iteration
  + Valuable requirements but not critical and can be postponed until further development

This rule set is used both for the entire basis of the development and within each timebox in the project and aid the decision-making process.

Timeboxes are individual steps within a given phase in which a requirement should be achieved either fully or in part. It is stated that generally the shorter period a timebox lasts the better as it provides a more manageable way to visualise what may be achieved in this time (Stapleton and Consortium, 1997, p.28). The primary goal of a timebox is to ensure that the iterative phases do not become indefinite cycles of development, each timebox will undergo three phases:

* Investigation
  + A first pass ensuring that activity is correctly targeted
* Refinement
  + Improvements from the investigation are made to ensure that the deliverables will be achieved
* Consolidation
  + Ensure that the deliverable is complete and is in keeping with the project

These phases plot to documented meetings or checkpoints that comprise of establishing relative priorities through MoSCoW for the timebox based on reflection from previously undertaken timeboxes, progress is checked at the refinement phase and deliverables are checked to be on track and finally the consolidation phase checks against the quality criteria established at the beginning of the timebox. Each phase encountered enforces a check point at the beginning and end of its segment.

***DIAGRAMATIC EXPLAINATION OF THE ABOVE.***

No specific tools and techniques are enforced until the DSDM method but it does recommend that models are developed to at least a minimum standard where maintenance can take place, this includes well-structured and well documented code referencing its purpose and any changes. Additional documentation should include context diagrams, an overview diagram and document, descriptions and models of each component and how they interact with each other, data structures and documents specifying design decisions. The developer may utilise either object-oriented analysis techniques or structured design and analysis techniques within the framework to best meet the requirements of the product. It is important that the analysis and design techniques are acknowledged prior to the initiation of the development process. Core models to the system development should be reviewed for content and accuracy as the system evolves throughout the iterative process (Stapleton and Consortium, 1997, pp.19-25).

The method was developed by the DSDM Consortium in 1994, originally consisting of 17 member companies with an interest in RAD this has grown to several hundreds of members including IBM and The Ministry of Defence (Avison and Fitzgerald, 2006, p.473).

### Methodology Role in Project Planning

Some of the key features of a software development methodology focus on the delivery and management of an application within the identified constraints, frequently time and cost, therefore the methodology provides a life cycle view of the application which can be utilised when working to determine the required time scales and resources that will be required throughout the software development. Software development methods and frameworks contribute to the greater picture of project planning through the shared themes of managing the progress of building a system but do not fully encompass the scope of project management.

The primary facets of project management are concerned with the estimation of resources and scheduling of the stages of the development. For these tasks, a different toolset exists to aid in the planning stages, tools such as Gantt charts and job lists help to identify the timing requirements of everyone associated with the development (Britton and Doake, 2006, pp.190-197). Project management also provides a series of contingency systems which can be used in the event of an issue arising within the development process.

Project plans can be derived, through analysis, from the information that is gained when applying a methodology to a specific problem area. The methodology provides an overview of the tasks to be tackled and the series of tools which will be used in addressing each step. Consequently, the project manager is able to determine how the time frames, that have been acknowledged when mapping the methodology to the problem area, can be best utilised and where they will fit within the strategy (Avison and Fitzgerald, 2006, p.75). As the development progresses, these time frames may be monitored against an initial plan to ensure that the project is on track to meet deadlines (Britton and Doake, 2006, p.193). As the project develops a picture is formed which shows a gradual transition between what was intended to happen and what has actually happened.

Although the specific allocations of time and other available resource, factoring in a static deadline and developers available, may have to be decided prior to the initiation of any development; when a development method is used in conjunction with a project plan it is possible to determine both a technical view of work to be done and a managerial view of resources respectively (Avison and Fitzgerald, 2006, p.74). The specific allocations of a variety of project components can be mapped as a result of the work flow layout determined by the development strategy.

Additional tasks under the umbrella of project management include the communication, administration and documentation of non-technical events throughout the development. Quality management is a core duty of any project management plan, ensuring that the deliverables are met to the agreed list of standards and where possible tested (Avison and Fitzgerald, 2006, p.76). Quality may consist of elements where a metric can be applied, such as timing performances, quantitative elements can be proven through tests and the project management should ensure that the testing strategy identified within the methodology accurately reflects the agreed points. Alternative qualitative values may not be easily applied to a metric; it is possible to determine, through the use of another project management tool Key Performance Indicators (KPIs) a measure of how satisfied an individual may be with these attributes and by conducting a number of tests with a variety of entities it is possible to establish some clarity with the more abstract areas of a project.

A variety of methodologies have been developed that are designed to better assist the project management. PRINCE is a methodology which defines project management aspects of a system under development (Avison and Fitzgerald, 2006, p.74).

## System Development Life Cycle

The conventional System Development Life Cycle model was established to address the requirements for providing a structured approach to the development of software systems. Many different methods and frameworks have been defined since each proposing a different view of the life cycle.

For the chosen development method, as stated in Section 2.1.2, the life cycle is defined as five phases. These phases are described in the following subsections, in accordance to the DSDM Method in Practice (Stapleton and Consortium, 1997, pp.3-10), describing the actions which should take place throughout each point and serves as a device from which the specifics of the project plan can be derived.

### Feasibility Study

This initial phase is designed to determine whether DSDM is the correct approach to the proposed solution development. Additional, common areas of feasibility studies are also undertaken at this point such as identifying the problem definition, the technical requirements of the proposal and the resource impact. Ultimately these attributes of the problem area should assist in the answering of the primary concern; to identify whether DSDM is appropriate, as answered in Section 2.1.2, in its framework and strategy to be applied in the development of the proposed solution.

This stage should produce a feasibility report; which for this study can be identified as section 2.0 of this document, an outlined plan for development and possibly the production of a prototype to demonstrate the validity of the proposal. The prototyping at this stage may not be undertaken if components of the system require further understanding.

### Business Study

The second phase forms the foundation from which all work will be completed throughout the development and beyond. It is intended for this section of the work flow to be short and assess the understandings of operational and technical constraints moving forward. The overall theme of the business study is to identify the processes to be automated and the requirements associated with these processes.

As, under strict DSDM, this area would make use of large group of stakeholders in a workshop style environment to determine the high-level requirements (with the lower level and more specific requirements being recognised at a later stage) it has been slightly modified for the purpose of the final year project. With the goal being to understand the processes being developed, this area has been utilised for primary research and literature review to achieve the same aim as the system being proposed more closely aligns with a research task than necessarily a business oriented application and as a result this study has been extended from the strict DSDM approach.

During the study, the list of high level requirements should be prioritised under the MoSCoW rules. This prioritisation is led by the technical and timing constraints that are imposed on the project.

As the research and understanding is developed, overview models of the proposed solution should be developed to aid in the later design and analysis stages. If utilising a structured analysis approach these diagrams would most likely consist of:

* Dataflow diagram of high level processes
* Entity-relationship diagram

Alternatively, if the system was designed with an object-oriented approach:

* Business Object Model

This phase is the point where the initial identification of the proposed system architectures should occur. These are permitted to change if, through later iterations, it is deemed necessary due to an unforeseen constraint. The architecture identification should consist of:

* Development platforms
* Target platforms
* Software architecture of major components
* Interfaces

Following the completion of the aforementioned tasks in this phase, the original plan outlined in the first phase should be refined and reflected upon. Following the refinement, the plan should accurately described the proposed plan for the next phase of prototypes and functional models and the configuration of the applications to be developed moving forward.

### Functional Model Iteration

Four key activities exist within each iteration of this phase:

* Identify the tasks to be completed
* Determine how to complete the tasks
* Perform the tasks
* Review the work completed

Within this phase, the iterations should produce analysis models and software components that will satisfy the prioritised requirements. As the cycles progress the reflection provides a base from which to improve when moving forward with documented models being updated to reflect the updated system.

Testing should be performed as elements are produced.

A variety of documentation should take place at this stage:

* Refined, prioritised functions with more specific requirements
* Functional prototype review documents
* Non-functional requirements
* Risk analysis for development moving forward

### System Design and Build Iteration

At this point the system is developed to a standard where the major functionality of the product is complete to the previously identified requirements. This should have undergone sufficient testing by this point at each of the iterations in the both this phase and the previous phase have undergone the pattern of testing as functionality is developed.

From this iterative cycle, the additional features may be added as time scale permits or if required the phase may step back into the previous stage. Both the design prototypes and functional equivalents are intended to be evolutionary and as such should be designed and built in a modular and expandable approach.

### Implementation

The final phase of the method sees the delivery of the system and transition from development to operation. At this stage in the life cycle of the development, the system will be installed on to its target infrastructure and the required training and documentation completed.

The User Manual should be completed and provide a comprehensive guide to the operation of the system, where possible it is encouraged under DSDM that the end user should assist or complete the manual as they will understand the operation of the system due to their involvement in the earlier phases and be able to communicate the concepts of operation in terminology that will be understood by the rest of the operational team.

The Project Review Document will summarise the achievements of the application with respect to the outlined functionality requirements and any further work that is required. From this point, the life cycle permits the development to move once again into an earlier phase if required. Reasons for moving into an earlier phase may be:

* Area of concern identified during build and design but development was prevented by resource constraints
* A requirement was identified in a late stage of development that could not be addressed due to resource constraints
* Lower priority requirements that had been identified throughout the life cycle were not implemented due to resource constraints.

## Project Plan

Following the understanding of the proposed solution plotted against the selected development methodology it is possible to specify estimated time scales and identify when resources will be required.

It should be noted that the time scales within the project plan tools will be adapted following the completion of a reflective phase where lessons learned can be applied in the endeavour of completing the next iterative phase. The project plan is therefore informed by the methodology and the reflection of work undertaken or any key checkpoints or issues that may arise in the development process.

Throughout the project development, the progress will be mapped and monitored using a Gantt chart and log of meetings. This chart will be updated frequently, following review points, to both be indicative of work that has been undertaken and work that is to be done with time scales being adjusted accordingly.

### Resource Identification

To produce the proposed solution various resources will be required. To manage the resource requirements and assess financial expenditure a resource schedule will be produced to work in unison with the overall project plan. The resources identified are as follows.

Hardware Requirements:

* A minimum of three Bluetooth Low Energy (BLE) beacon devices, if possible more would be beneficial up to the range of 9 (range subject to testing environment size).
* A mobile device (or devices) featuring an accelerometer or Inertial Measurement Unit (IMU), Wi-Fi and Bluetooth (preferably Android 6.0 or newer device).
* A suitable specification development PC, featuring Windows 10.
* A Windows PC for demonstration and testing, acting as a calibration station. A 1080p monitor will be required to display the calibration software interface.
* A Windows PC to run the server application.
* Two Windows PCs with a projector display for demonstration and testing to act as an interactive content exhibit.
* Associated cabling (power cables, USB micro to USB A, display cables) and consumables.

Software Requirements:

* Visual Studio IDE (including the Xamarin component for Android development). Alternatively, separate Windows and Android development IDEs.
* Android Debug Bridge (ADB).
* Device drivers for mobile device.
* Project management software (to be decided subject to research).
* Outlook email client.
* Microsoft Office suite for documentation.
* Prezi for development of interactive poster/presentation for use during viva examination and mid-point review.
* Associated software with the beacon systems.
* Modelling software.

Information Requirements:

* Supervisor advice for project management, structure and development.
* Mathematical expertise relating to positioning in 3d space.
* Research access, library access and laboratory access.

User Requirements:

* Willing testing participants.

Infrastructure Requirements:

* Mounting location for beacon devices.
* Laboratory access, use for demonstration and testing.
* Desk space, for development, demonstration and testing.
* Projector mountings at two ends of the laboratory.

Further to the aforementioned requirements a full version control suite will be utilised to ensure that, in accordance to the chosen methodology, it is possible to navigate branches of development to move forward and backward through the development process as required.

### Time Scale Identification

The time scale for the project is dictated by the module hand in date. All work must be completed and submitted by 24/04/2018. Prior to this a series of important milestones must be achieved, the entire project consisting of 148 working days.

Additional constraints include other module deadlines which take place within the above time frame. Specifically, the identified module deadlines are as follows:

|  |  |
| --- | --- |
| **Semester One** | |
| **Subject** | **Deadline** |
| Real Time and Safety Critical Systems | 10/12/2017 (updated to 07/01/2018, updated to 14/01/2018, updated to 15/01/2018) |
| Emerging Technologies and Concepts | 07/01/2018 (updated to 10/01/2018) |
| Interfacing and Distributed Computing Techniques | No assignment deadline within this period. |

**INSERT TABLE NUMBER HERE**

|  |  |
| --- | --- |
| **Semester Two** | |
| **Subject** | **Deadline** |
| Real Time and Safety Critical Systems | Assignment: 02/03/2018  Assignment: 16/03/2018  Assignment: 20/04/2018  Demonstration: 23/04/2018 |
| Emerging Technologies and Concepts | No assignment deadline within this period. |
| Interfacing and Distributed Computing Techniques | Assignment: 29/04/2018  Examination: Summer examination period |

**INSERT TABLE NUMBER HERE**

Further to the main time constraint identification, important dates exist in the schedule for the development of the project which will be mapped accordingly on the Gantt chart and its iterations which are present in the appendices of this document.

The initial study phases will lead into the iterative development phases which will be constructed of timeboxes, as per the DSDM framework, where the initial fixed length of these timeboxes will be estimated based on the work to be completed and the available time remaining in the project life cycle and then adjusted following the completion of an iteration and its reflection period.

### Deliverable Identification

Software Deliverables (artefact to be made available in requested storage format):

* Android application for user interaction within the interactive exhibit room
* Windows based server application to communicate, monitor and action triggers between content and user interaction software.
* Windows based content application to be interacted with by the user interaction application via the server application.
* Windows based calibration application to be interacted with by the user interaction application via the server application.

Documentation Deliverables:

* All documentary requirements identified in the Final Year Project Handbook.
* Evidence supporting any non-literature based research activities.
* Log book and supplementary meeting notes (in digital format and original handwritten notes to be made available).
* All design and implementation documentation produced through the development method (to be selected, subject to research).
* Fully documented and analysed test results, raw data to be made available.
* Full program listing.
* Installation documentation.
* Operational documentation.
* Technical documentation.
* Project plan, with evidence of each iterative change and justification.
* Project management evidence generated by project management software utilised.
* Presentation and presentation delivery notes.
* Project evaluation.
* Additional appendix information pertaining to research. Sensor equations, functions and comparisons.

Demonstration Deliverables:

* A recorded and observed practical demonstration of the system in use.
* Evaluated practical demonstration.

### Procurement of Resources

Within the identified resources, Section 2.3.1, several properties are not available without capital expenditure. To ensure that financial spending may be kept to a minimum, the procurement of costed resources will only occur following thorough specification and at the correct time in development based on the project plan, development method and resource schedule.

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Specification Required** | **Phase in Development Required** | **Cost (*estimate)***  **(confirmed)** |
| Beacon | Room dimensions and research into broadcast ranges and signal degradation. | Functional Model Iteration and onward | *£70 (6 beacons)* |
| Source Control Cloud Storage | Time frame for software development. | Functional Model Iteration and onward | *£15 (3 months)* |
|  |  |  | *£85* |

**INSERT TABLE NUMBER HERE**

### Measure of Success Identification

The success of the project will be measured against several identified criteria. Important factors include the quality of each produced element, which is decided in accordance with the MoSCoW rules applied at the beginning of each development phase and the overall timeliness of the system.

* A clearly defined, addressed and justified problem domain.
* A set of well researched points of interest to contribute to the solution.
* A professionally developed project plan based on proper analysis of the initial problem domain and its academic requirements.
* A well identified, defined and utilised development method or hybrid of methods, how this has been adhered to and how it benefited the creation of a well-designed artefact.
* A strong specification and requirement definition that both addresses the problem and justifies the proposed solution to the problem.
* A successfully designed and implemented prototype that solves the identified concern, is relevant to the specified domain, meets the requirements of the academic award, adheres to points of research and has been developed in accordance to the identified methodology and evaluated through comprehensive testing.
* An appropriate testing strategy designed to acknowledge any flaw in the artefact and conclude with best practices of how these should be addressed based on evidence and results that have been collated and analysed with the analysis taking place on copies of the data ensuring that the raw information remains uncorrupted.
* A Human Computer Interaction system that follows recommendations determined from research.
* A body of work that is completed to a high standard, properly addresses the requirements, produced on time and in accordance to the project plan.

### Project Life Cycle

To fully cover the requirements of project management, the development framework chosen works concurrently with the project life cycle. The project life cycle is identified under four distinct phases:

* Project Initiation
* Project Planning
* Project Execution
* Project Closure

The framework under which the solution is being developed specifies these areas from a technical perspective whilst the project life cycle manages the administrative aspects of the development. Items such as meetings, reviews, communication and non-technical documentation may be encapsulated within these four stages and as a result should be mapped in the project plan.

## Ethical Considerations

As an academic research undertaking, this project is subject to scrutiny by the university to ensure that it upholds ethical performance at all times.

The project will be conducted in accordance with the ethical codes of conduct of both the British Computer Society (BCS) and Staffordshire University. A statement of the ethical procedures to be followed is available in the appendices of this document.

## Health and Safety Considerations

To ensure that the project is undertaken in a safe and responsible manner, a comprehensive health and safety evaluation has been undertaken in the form of a risk assessment. This document is considered an active document throughout the development of the solution and as a result will be updated when any issue is encountered and solved.

The risk assessment is available in the appendices of this document.

## University Department Interaction

Whilst preparing for the procurement of beacon devices pertaining to the system development, it was noted that dimensions for testing environments would be required therefore correspondence with departments within the university was established.

The aim of the correspondence was to determine the dimensions of two laboratories housed within the Mellor building. As a result of the communication, meetings were discussing to be scheduled with the Information Services Department, who were trialling a system with a similar end goal, and Facilities Department.

Transcripts of the email correspondence and log books of completed meetings are available in the appendices of this document.

## Version Control

Version control has been identified as a key component in the development process for the project. It is hoped that through utilising a repository the risk of losing work may be mitigated and it will be possible, in accordance with the identified development framework (Stapleton and Consortium, 1997), to revert to an earlier version of the work to remove any functionality which is later deemed superfluous in review.

Indeed configuration management is specifically stated as a requirement of the framework (Stapleton and Consortium, 1997, p.73).

The philosophy of rapidly producing demonstrable work will be easily monitored by identifying versions of the product against the project plan allowed the review of changes to be made.

The concept of version control will be utilised alongside other forms of data storage and back up.

## Research and Analysis Methods

Under the DSDM approach, no specific tools or toolsets are prescribed and it is viewed that the analysis completed should be appropriate to the task in hand (Stapleton and Consortium, 1997).

Background analysis will be performed to model, from a high-level, the proposed data flow and entity relationships within the system.

As the study portion of the framework has been adapted to suit the additionally required background research, several research methods have been identified to complete the initial literature review from which a hypothesis may be developed.

Primary research may include performance testing of technologies to determine their suitability or conducting observations and interviews as deemed appropriate by the development method.

Secondary research may include the extraction of information from textual sources.

Important areas identified in the development of the system have been added to a research design which will be followed to ensure that a systematic approach is followed in producing analytical information.

**FIGURE OF RESEARCH PLAN HERE**

## Design Methods Supported by Development Methodology

DSDM does not enforce any specific design methods and encourages the design to be undertaken in a method appropriate to the task that is being undertaken. It does however specify and require a minimum level of design and modelling to be performed with the aim of assisting the future maintenance of the system. Well documented and designed code should be kept up to date and changes noted in the review sections and documentation followed the end of each iteration cycle (Stapleton and Consortium, 1997).

As such, and in accordance with the explanation in Section 2.1.3, the design methods will be specified at each phase.

## Testing Methods

As part of the quality delivery of DSDM, testing is undertaken at every point where a functional element is produced.

Testing tools may be used to capture and replay applications (Stapleton and Consortium, 1997, p.73).

Functional testing is important to ensure that the application performs against the outlined requirements at each checkpoint in the application development.

Dynamic testing should be performed to ensure that the program does not contain any logical errors. Individual component testing will be performed throughout the development at each stage and integration testing performed at the stages where integration of two or more components is possible.

Specifically undertaken testing will be documented and identified within each phase.

## Reflection of Project Proposal and Development

Within this chapter the groundwork has been laid in preparation for the development of the system. This has consisted of a development method being identified and the time constraints applied to that model. As a result of the work completed through this portion of the body of work the Gantt will be further evaluated to properly reflect the current estimations of work.

The DSDM approach appears to be well matched to the requirements of the proposed solution, although it is conceded that the initial business study component of the framework may be modified in order to better accommodate the requirement for a level of background knowledge and technology justification prior to the initiation of the development iterations.

The iterative approach and more formalised documentation and testing associated with the DSDM framework proved appealing and less in line with an unplanned system which may be considered of other RAD and Agile approaches which maintaining a facility to more dynamically adapt to evolving functionality requirements as the problem domain is better understood.

The value of following a recognised methodology will be evaluated throughout each of the reflective points within the project and whether indeed the specifically allocated framework has proven to be the correct decision for the proposed solution.

# Project Overview and Feasibility Report

Following the completion of chapter 2.0 which serves as the feasibility study for the chosen methodology within this project, the following structure for this chapter has been established.

This project overview demonstrates the feasibility report, providing an outlined plan for development and investigation into a potential prototype. As stated in “DSDM The Method In Practice”, this prototype make not be undertaken in the event that further understanding is required, responding to this requirement a fully established prototype will not be undertaken for reasons identified in the following subsections. Prototyping will be undertaken however throughout the primary research section to gain a more complete understanding of certain tools which will be employed in the production phase of the software artefact.

The feasibility report consists of the refined and elucidated problem definition, as initially established within section 1.1 of this document; a technical assessment of the plausibility of the problem definition, answering such questions as the value to the business and the impact on resources and finally the applicability of the methodology. As previously discussed, DSDM is to be applied in situations where the development of the system is considered to be required urgently (Stapleton and Consortium, 1997, p.5), when being modelled against the proposed solution and the available resources permitted by the framework of the Final Year Project module the methodology fits well in delivering the distributed components within the very small time frame that can be allotted to each component when dividing the entire module length. In a conventional business setting, a feasibility study may take up to several years to be completed due to the cautious nature of a company in the lead up to developing a new system; under DSDM it is intended for this part of the project to be completed in no more than a few weeks, as this project is not being completed in a traditional business but rather as an individual piece of work many of the common business requirements can be mitigated

Associated with the feasibility report, an outline of the development plan and a prototype can be employed (with the latter being optional depending on the availability of existing knowledge and requirements of the project) to demonstrate that the project is technically possible. Supporting the optionality of the prototype, it is stated that it may not be wise to consider building immediately where either the development of the system is well understood and tested or that there is not enough existing knowledge to produce a sound demonstration; this is contrary to other fast paced development methods such as RAD which may approach the build sooner and handle any encountered problems at a later point framework (Stapleton and Consortium, 1997, p.5).

This chapter will detail the feasibility report for the project, with the business study following in chapter 4.0; the requirements are identified and through the use of the MoSCoW system as prescribed by DSDM (Stapleton and Consortium, 1997, p.26) the objectives will be prioritised allowing for a tactical approach to developing the initial project development plan. It is anticipated that this project development plan will evolve throughout the project, it is perfectly acceptable under the method of DSDM to modify the plan as the system becomes better understood by its incremental implementation.

Within the graphical representation of the methodology this area of study is visualised as the top of the pyramid having completed the pre-project section throughout preceding chapters. At this point there is not a recognised back step however it is permissible for the project to be reconsidered thus ending this DSDM approach. Any decision to end the project at this point should be made as early as possible to allow for alternative plans to be drawn up.

## Stakeholder Identification

As an academic body of work, this project has three physical stakeholders that are identified in the briefing documents for the module. Namely the three stakeholders are:

* The Author
* The Supervisor
* The Second Assessor

Where additional resources are required or access to specific equipment or facilities are required the individuals responsible for the resource or facility would also be considered a stakeholder:

* Estates Department
* Digital Services Department

Additionally within the theoretical domain of the project, the leisure industry, a number of additional stakeholders exist. If this resulting artefact from this project were to be developed beyond a prototype, as is the specified and titular intention, these groups would be heavily referenced in the report and additional consultation would be performed to ensure that the delivered system is both appropriate for their needs and completed to a satisfactory standard as a professional product.

Stakeholders in the scenario of a marketable product would include:

* Attraction Guests
* Attraction Management
* Attraction Operating Company
* Attraction Host Staff
* Attraction Operator Staff
* Attraction Maintenance Staff
* Attraction Information Technology Staff
* Attraction Sanitation Staff
* Attraction Security Staff
* Attraction Medical Staff
* Attraction Marketing Department

### Stakeholder Justification

Each of the identified stakeholders contribute in some way to the requirements of the system or specify its technical direction.

The first three educational stakeholders are critical to the delivery of the report and its assessment as an academic body of work.

It is the responsibility of the author to produce an individual piece of work; providing a conceptual project title, problem area and proposed solution to be assessed for suitability by an academic from whom the author must request official supervision. The author must then proceed to produce the required ethical statements, health and safety documents and arrange and attend regular meetings with the supervisor. Further responsibilities of the author include the production of this report; the professional standard design, implementation and testing of the software components; conduct the project adhering to a project management strategy and prepare all deliverables to be made available for submission. Additionally, interaction with the second assessor will be required to be established for both the mid-point review and the Viva Voce examination at the conclusion of the project.

The supervisor responsibilities include performing an advisory role to aid the author with their implementation and documentation. Their expertise may be required to provide direction when completing ethical statements and any changes required to the project. The supervisor will additionally be responsible for using the university systems to ensure that the project is administered in accordance with the university requirements for assessment and attend the meetings scheduled with the author.

The second assessor contributes to the final marking of the author’s body of work in conjunction with the supervisor, they are required to perform both the mid-point review and the Viva Voce alongside similar administration of the project marking through university system in accordance with the university requirements for assessment.

Additional university stakeholders have been identified through their causational influence on the project completion. The author contacted the documented departments to acquire further insight into the availability of facility dimension documents, these secondary sources could be eliminated should the author personally measure facility layouts. The need for this information reflects a requirement that may occur in the event this project is developed further and identifies similar stakeholders in a production system.

Attraction level stakeholders cover the range of departments, groups and individuals who would come into contact with the proposed solution in some way.

The guest encapsulates an end user of one component of the proposed solution, where they would utilise the digital device being tracked around the attraction and as such would be using both a hardware and software component. It would be necessary to ensure that these components are considered safe for their use and created in such a way that they are accessible to a wide range of guest requirements and intuitively easy to use with little instruction.

Attraction management would be consulted to ensure that the project meets their vision and provides a reasonable investment. They would be responsible for the allocation of staffing resources and budgetary resources for the operation of such a system however would not be a direct user. They may also maintain an interest in the guest locations throughout the attraction for metric purposes. Furthermore the attraction management would be responsible for implementing relevant health and safety procedures and informing other departments of any particular requirements such as security or medical teams in the event of a guest being extracted from the facility.

The attraction operating company would be associated by brand and provide the capital expenditure to the management for the procurement of the system. Whilst not a direct user, they would hold financial and operational interest in its success.

Attraction host staff would be defined as any staff within the attraction who do not interact with its control software but interact with the guests moving through the attraction. As such they would be required to have a level of operational knowledge, being able to resolve guest issues efficiently and to recognise components of the system to instruct maintenance and operator teams of any noted failures. Host teams would require clear and succinct documentation and training.

Operator staff would be defined as team members who directly interact with the control software but not with any of the back end systems. They may, subject to the requirements, not have full user privileges but would be responsible for the daily operation of the system and thus would require training, documentation, consultation and a user interface that is considered acceptable to both their requirements and conforms to the requirements of any governing body for the attraction type.

Maintenance staff would be required to carry out regular checks and maintenance on the attraction, these maintenance staff are defined as team members who deal with traditional attraction hardware and not the computing hardware or systems. They would be required to have a level of understanding of the system to ensure that it would not be inadvertently misused during any maintenance period and would be consulted on its installation to ensure that it does not have a negative impact on existing procedures.

Attraction Information Technology staff would be responsible for the system software and hardware, providing the technical support for the attraction once the system has been handed over from any installation company. Their input into the system implementation would aid in the ease of maintenance in the future, additional training and more detailed maintenance documentation would be required to suitably address their level of technical expertise and potentially elevated privileges for the system control to permit diagnostic and testing abilities.

Attraction sanitation staff would likely come into contact with components of the system when cleaning the attraction, they would not require any software based instruction but should be advised of the correct methods to ensure that the hardware is not damaged or exposed to any harmful substances when being cleaned whilst also being sure that the hardware is suitably hygienic for use.

Attraction security staff are defined as both physical security responsible for guest wellbeing and securing the attraction assets, they would be made aware of safe routes through the attraction where they would not damage or be harmed by any system.

Attraction medical staff would be made aware of any particular best method to extract a guest in the event of an emergency where their efforts would not be hampered by any hardware.

Attraction marketing teams would be consulted when designing the user interface and all guest facing components of the hardware and software, ensuring that the system meets the requirements of the companies brand and identity, they would also be advised about design decisions influenced by researched best practice for user interface design to ensure functionality is maintained.

## Top Down Overview Objectives

## Prioritised Objectives

MoSCoW

## Project Technical Scope

Walk around experience, not a dark ride.

## Project Technical Domain

## Success Criteria

Refine through research, general overview (specifics applied per section)

## Deliverables

# Business Study

Leads into research to fully evaluate the business case.

# Preliminary Literature Review

## Research Method

## Review of Available Literature

## Reflection

## Influence on Project Plan

# Phase One

## Overview of Phase One

## Research into Problem Area

## Analysis of Problem Area

### Analysis Method

### Application of Analysis

## Design of Phase One

### Design Method

### Application of Design

## Creation and Implementation

## Phase One Modular Testing

### Testing Method and Strategy

### Testing Plan

### Testing Conclusions

## Reflection of Phase One

## Influence on Project Plan

# Overview of All Components

## Implementation of All Phases

## Testing of Complete Implementation

### Testing Method and Strategy

### Testing Plan

### Testing Conclusions

## Reflection of System

## Influence on Project Plan

# Critical Review and Identified Areas for Refinement and Furtherance

## Problem Area Post-hoc Analysis and Reflection

## Project Delivery Success Identification

## Project Delivery Methodology Reflection

## Project Artefact Success Identification

## 7.3. Incomplete Features and Artefact Refinement Opportunities

## Feature Change or Addition

# Evaluation

## Professional Skill Value

## Academic Skill Value

## Development and Learning

## Alternative Project Paths

## Reflection and Evaluation of Project Success

# Further Work

# Graduate Exhibition (GradEx) Catalogue Entry

# Conclusion

User Manual and Documentation for: Server, Guest App, Control Software

# References

Avison, D. and Fitzgerald, G. (2006) ‘Tools, Information Systems Development, Methodologies Techniques and Tools’. London : McGraw-Hill Education . Available at: http://staffs.summon.serialssolutions.com/2.0.0/link/0/eLvHCXMwY2AwNtIz0EUrE5KBacnIwMTQItk8KSkx0SzN3CjNNNXILM3YxDQ5EbzpNcjDNCDA0i3KyAmxHAo6\_gZsJ6WlFUM3V8E6jdn5GYngjqUuZLEldMdaDtA7kLs0DI3NLSxNmRmYjU3Al5sb-8ETmgVo9MvEANJLMwd2CYCVKPQ0HhjfDHR2KNhepPrGTZCBBbQH.

Barritt, D. (2002) ‘IEC 61131 and DSDM in real-time process control applications’, *Computing & Control Engineering Journal*, (April).

Bowers, J. *et al.* (2002) ‘Tailoring XP for Large System Mission Critical Software Development’, *Extreme Programming and Agile Methods — XP/Agile Universe 2002 SE - 10*, 2418, pp. 100–111. doi: 10.1007/3-540-45672-4\_10.

Britton, C. and Doake, J. (2006) ‘Software system development: a gentle introduction ’. London : McGraw-Hill . Available at: http://staffs.summon.serialssolutions.com/2.0.0/link/0/eLvHCXMwY2AwNtIz0EUrE4wM08ySEoE9H8MUo2RT4ySzZJO0RIMkM9DhMalJ4KU6QR6mAQGWblFGTojlUNDxN2A7KS2tGLq5CtZpzM7PSAR3LHUhiy2hO9ZygN6B3KVhCNpzasnMwGxsAr7c3NgP0iszB2ZvYLkNPX0HxrcAnRUKtgepfnETZGAB7TkQYmBKzRNmEIDd.

Centers for Medicare & Medicaid Services (2008) ‘Selecting a development approach’, *Centers for Medicare & Medicaid Services*, pp. 1–10. doi: 10.1016/j.csi.2016.06.003.

Stapleton, J. and Consortium, D. S. D. M. (1997) *DSDM: dynamic systems development method : the method in practice*. Harlow: Addison-Wesley. Available at: http://staffs.summon.serialssolutions.com/2.0.0/link/0/eLvHCXMwbV1LC8IwDA4-LorgG5\_gH5jMdpvr1cfwMhjqyYtkW4ug6GH-f2y7TkU8Noe2gbRJPpIvAJTMbevnT\_Bx4XNpOgmNPc8VyITMS9BbIpVSkij8bb9zo4gFJ7L6MAwZ\_E3GSUJkprmqSBqvjwvqxNLKiy1Nx9pNqpPP0mCuJjctU0fPNqea\_lH5O5n1MWrId4q1.

Vijayasarathy, L. R. and Turk, D. (2008) ‘Agile software development: A survey of early adopters’, *Journal of Information Technology Management*, XIX(2), pp. 1–8. doi: 10.1287/mnsc.48.8.1008.163.

Williams, R. (2006) *Real-Time Systems Development*. Elsevier Ltd.

# Appendices