**Plan of dissertation.**

Investigation of Approaches In Safety – Critical Systems

**Abstract**

Supply brief overview on what the Project is about, talk about Spark, Spec#, CheckedC, SCJ ect..

**Declaration**

I declare that this doc rep my own work except where otherwise stated.

**Acknowledgments**

Thank Steve for his help in the project.

**Chapter 1 Introduction**

* 1. **Introduction**
  2. **Definition of the problem –Review of features needed for SCD**
  3. **Hypothesis**
  4. **Project Aims**
     1. Aim
     2. Objectives
  5. **Plan**
  6. **Summary**

**Chapter 2 Background research**

**2.1 Introduction**

**2.2 Problems with Development Methods**

2.2.1 Agile

2.2.2 Waterfall

2.2.3 Formal Development

**2.3 Programming Languages for Safety Development**

2.3.1 Common Unique Features

2.3.2 Usage in Industry

2.3.3 What are they used for?

**2.4 Overview of Safety Critical System**

**2.5 Summary**

**Chapter 3 - System Design**

**3.1 Requirements**

**3.2 System Overview**

3.2.1 Description of system

3.2.2 UML Diagram

**3.3 Design of Animation**

3.3.1 Sketch of Design

3.3.2 Failures in Animation

3.3.3 Success in Animation

3.3.4 Communication between Java and Ada

**3.4 User Interface Design**

**5 Implementation**

5.1 Implementation of Railway Crossing

5.1.1 Basic Structures of types in ada

5.1.2 Modelling Failures

5.1.3 Modelling Success

**5.2 Animation**

5.2.1 Design Aim

5.2.2 Design Implementation

**6 Testing**

6.1 Railway Implementation Unit Test.

6.1.1 Testing Non Proofed parts

6.1.2 Testing animation

**6.2 Spark Tools**

**6.3 Proofs by spark.**

**7 Evaluation and Results**

# Introduction

A Safety-Critical System is a System in which a failure can result in Losses of millions, it can cost lives and injure people. Such Systems require very careful planning when developing them, some clients of programming languages may stick with such languages as C in developing embedded software.

However, languages such as C are very hard to prove correctness of a C program. Such as accessing a pointer, or trying to allocate Heap Space. What if there is no memory? What should the system do? Such faults as even divide by zero errors can lead to the compiler errors, Overflow Errors Such as *the Boeing 787 Dreamliner*[1] <https://www.engadget.com/2015/05/01/boeing-787-dreamliner-software-bug/>

<https://s3.amazonaws.com/public-inspection.federalregister.gov/2015-10066.pdf>

In which an Integer Overflow bug will eventually occur within “*248 days of continuous power”* AOL (2017). This means that it resulted in a loss of all AC electrical power.

These Types of errors are 100% not allowed to happen in Safety-Critical Systems.

Such problems may only be solvable at runtime and which excessive testing, trying to throw a range of values that the subprogram may take to find these problems.

However, there are some Languages that can be used to Tackle these problems before they arise. One example is Spark which is a subset of Ada, Checked C, Spec Sharp, Safety Critical Java. These languages offer Tools for proofing that a program is free of errors, by generating proofs to ensure no matter what happens that the programs are never going to be executed by a divide by zero for example or an overflow.

# Definition of the problem

In this paper, I will be using Spark to develop two case study’s. One being a light bulb example to get an easier of idea of what pre-conditions post-conditions and other features of spark and then develop a bigger study of a rail way crossing simulation. These will then be compared to Java in which I have 2+ years’ experience in using and comparing them to other similar languages such as Spec Sharp in background research.

I want to use these findings to then find what types of features are needed to develop safety critical Systems, and current tools for Java that can be used to develop a safety critical system and features that cannot be used. I want to also compare the usability of spark and java and Learnability.

This could be used to identify features of Spark and Spec sharp to be used in future development of Safety-Critical Systems and other types of systems which might not be Safety Critical such as a bank System. This can then be considered when clients want to choose a programming language for their system.

# Project Aims

**Aim:** Investigation of approaches to safety-critical systems development.

The Aim of the project is an Investigation of approaches in which I look at a range of software development methods, draw comparisons to them in which I will choose the one, which is most beneficial to safety. I will then develop a very small case study, and figure out what features are needed for safety critical development and compare Spark to Java in terms of usability, ease of use and errors.

**Objectives:**

1. **Research into software process methods for safety-critical Systems.**

Researching into the correct software process method to use will be critical to the project for making sure areas are covered, at least four should be researched before starting anything else to make a comparison between which is best for a safety critical System.

1. **Research into languages that are used for Safety Critical Development and see what features they offer.**

Researching current languages used for safety-Critical development will involve looking at what kinds of languages are out there to date. I want to focus on the languages that companies are more likely to use to develop the systems I will do this by trying to find sources, which point to some popular languages.

1. **Develop a full evaluation Criteria which evaluates both programming languages for a comparison**

This will be based on the research I have gathered together to find a full evaluation Criteria which should be based on things such as features, code coverage, usability and correctness.

1. **Learn Language which is popular for Safety-Critical development**

I want to learn one Language that is appropriate for safety-critical development, in this case will be spark and knowledge of Spec Sharp.

1. **See what features are needed for Safety-Critical systems**

I want to be able to identify what features are needed to fully proof that a language is free of runtime errors by examining what features are most important to do this.

# Background Research

# 2.1 Introduction

In order to fully compare these languages a lot of background research had to be done to improve my knowledge of them. A lot of my learning from languages such as Spec sharp, checked C and Safety Critical Java are done here. To improve my knowledge of such languages and most important for demonstration purposes I have used full explanation of books for Spark [Spark Ref here] and for the others have used specifications which is the only resources I could find on these other languages.

I want to use this research to make comparisons on the Languages. See what features each have to offer and if these are meet safety Critical requirements.

I can also make some assumptions on usability by comparing the code to Java in which my opinion is easier language to generate good code, and easier to write in.

# 2.2 Development Methods

Development methods used in industry can affect the overall quality of a project, where agile is for very fast moving industry waterfall is more beneficial towards projects which can afford some slack time, formal methods can be used to compare spec directly to code by actually proving the code meets the specification.

Formal development is actually used often for Safety-Critical development, saying that the program is fully proved to be identical to the specification may sounds good but if the specification is worded incorrectly behaviour may be implemented which was not previously wanted. I will talk about these in this section and what the advantages are and disadvantages of using them are using background material to support my discussion.

# Agile

Agile methodology is used a lot in the industry, nearly every company which needs to adapt to new technology needs to use it. Agile development introduces that a company which needs to cope with ever changing requirements and technology requirements means that code may have to be released with patches, which is not acceptable in safety-critical development.

A popular book Somerville which is used a lot by software engineers and recommended reading at Newcastle University states that because of rapid development “*businesses are willing to trade off software quality and compromise on requirement’s to achieve faster deployment*”. [Summerville Ref here Page 57]

Summerville also sees my concerns with Agile used as process for Safety Critical Development in that by the time the software has been released it may have changed. “*By the time software is available for use original reason for its procurement may have changed so radically that the software is effectively useless*” [Summerville Ref here Page 57]. This result of radically changing requirements result in money wasted, and if the purpose changes the whole system changes effectively meaning a new project would have to be started.

Even if Agile is used Money will always be wasted by advances in technology. If a company which develops Safety-Critical Systems have to keep making other iterations it’s hard to keep up with this. If one system is developed it would be unacceptable for then requirements to change and then other technology be bought to cope with this change.

Agile is also based on iterative development which then is improved based on feedback [https://pdfs.semanticscholar.org/69b1/9ddc8a578f4c63d1dfe15252a465ee12fe5d.pdf].

This is not needed for Safety-Critical development it may be needed a little bit but nowhere near as much as agile. If needed it can be incorporated into the project of a Plan-Driven Development.

Based on these and my own thoughts, Agile should not be used to develop Safety-Critical systems. Based on:

* Too much changing of requirements, Safety-Critical system requirements should already be known before development starts.
* Agile development can often result in bad code, which cannot be the case where people’s lives are at risk.
* Too much dependence on customer involvement.
* Patches should not be part of the process
* Is not a plan driven model.

# 2.2.2 Waterfall

Waterfall is a planned type of development process, this means that all steps before starting a project are planned and that changes are not to be made during actual lifecycle of the process.

Within the development of software using this model it is necessary to be able to finish each part of the project before starting the next unit of work [Summerville Page 31].

Because the waterfall development model is so detailed before the actual project is undertaken it means that development is very secure, this is all of course assuming as with all projects that, Requirements, work distribution and time management is correctly measured.

Each unit is treated in sequence and the next sequence should not be started until the previous section is completed [Summerville Page 31]. It ensures that no other process should be started unless it has been completed ensuring each stage of the model is completed with full confidence that it is correct.

However, because of previous changes in the model it means that certain parts of development may have to be reworked because of changes [Summerville Page 31]. This could cause problems and misunderstandings because of the constant reworking of documentation, which will lead to some parts being frozen for others to start being worked on.

This can cause slower development later as documents get refined, however has the advantage of pausing uncertain changes before they can be undertaken. Afterwards these things can be reviewed and updated.

However, because of freezing it can lead to software which is essentially wrapped around to fix problems adding redundancy which is not