**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. **Project Aims and Objectives**

**Chapter 2 Background research**

**2.1 Introduction**

**2.2 What is SC development?**

**2.3 Problems with Development Methods**

2.3.1 Agile

2.3.2 Waterfall

2.3.3 Formal Development

**2.4 Programming Languages for Safety Development**

2.4.1 Common Unique Features

2.4.2 Usage in Industry

2.4.3 What are they used for?

**2.5 Overview of Safety Critical System**

**2.6 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.