HIDECS

High Integrity Distributed Embedded Control Systems

Certification of Forth

By Paul E. Bennett IEng MIET

Certification of Forth

1. Introduction
2. Pre-requisites to Certification
3. Certification Process
4. Evidence to Show

Introduction

- Software is now in control of systems that have the potential to cause harm if such controls failed.
- Newer systems will put even more software into positions of critical control.
- Regulated Industrial Sectors require evidence based certification to prove software is safe to use.

Introduction

Certification Requirements

Arg 1 - the system has been specified to be safe - for a given set of Safety Criteria, in the stated operational environment

Arg 2 - the resulting system design satisfies the agreed specification

Arg 3 - the implementation satisfies the system design

Introduction

Such demonstration is given by provision of:-

- **Direct evidence** - which provides actual measures of the attribute of the product (i.e. any artefact that represents the system), and is the most direct and tangible way of showing that a particular assurance objective has been achieved.
- **Backing evidence** –which relates to the quality of the process by which those measures of the product attributes were obtained, and provides information about the quality of the direct evidence, particularly the amount of confidence that can be placed in it.

Pre-requisites to Certification

- The existence of a Requirements Specification that meets the criteria of **Arg1**.
- The existence of fully developed documentation that meets the criteria of **Arg2**.
- The above have been brought together by a process that has taken the development of the documentation through all versions and tracked all changes.

Pre-requisites to Certification

Requirements should be

- Clear
- Concise
- Correct
- Coherent
- Complete
- Confirm-able (Testable)

Pre-requisites to Certification

- Documentation Standards exist within the development organisation and are applied.
- Coding Standards exist within the development organisation and are applied.
- Cyclomatic Complexity of the Requirements, Design and Implementation is kept as low as is reasonably practicable.

Documentation Standards

- Having a standard that is enforced ensures that presentation style is made consistent.
- Consistent Style eases the location of pertinent information to the design.
- Standards, enforced through review and audits, improve quality generally.

Coding Standards

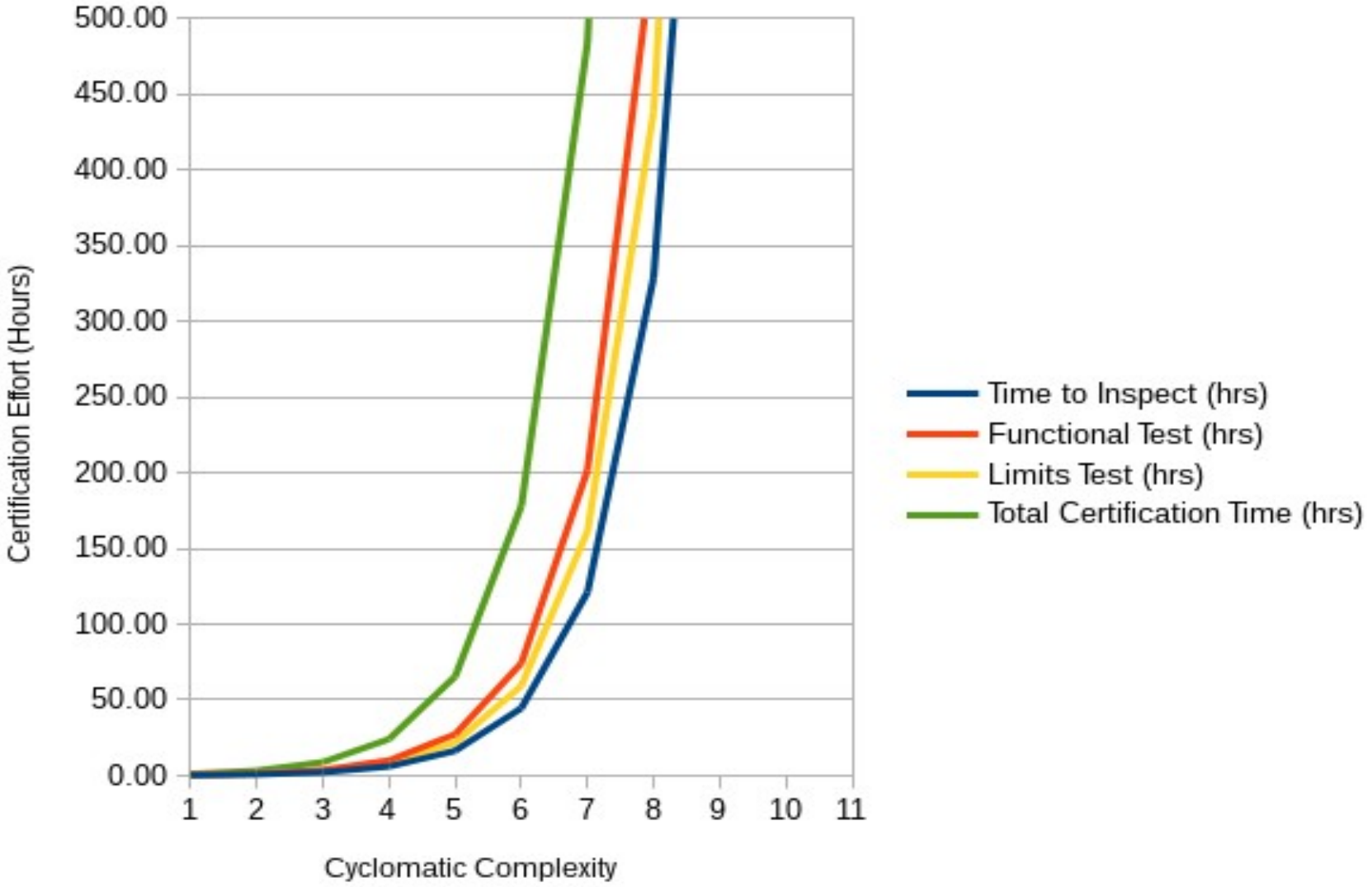
Coding Standards, when enforced:-

- Makes code easily readable and improves style consistency (even across teams).
- Prevents use of poor coding technique
- Highlights where re-factoring would be of benefit
- Provides the base-line against which Static Analysis can be conducted.

Why reduce Cyclomatic Complexity

- Minimum Cyclomatic Complexity (1) is easy to test and can be accomplished in a reasonable time-frame.
- Higher Cyclomatic Complexity requires tests to be run several times to ensure full coverage of all logical pathways.
- The higher the Cyclomatic Complexity number requires exponentially more time to accomplish the tests for full coverage.

Cyclomatic Complexity vs Effort



Components

- Have a unique identifier
- Have a data-sheet
- Interface at surfaces
- Specify considered environmental constraints
- Can be re-used many times without re-certification
- Can be inspected tested and certified individually
- Conform to standards
- Have published limits for guaranteed operation.

Summary

- It is important to have clear standards of documetation and coding styles.
- Component Oriented techniques can keep things simpler in the minds of reviewers
- Time to Inspect and Test is exponentially proportional to the Cyclomatic Complexity of the component.