Xday, XX XXX 2013.

9.30 am - 11.15am ***(check this!)***

University of Glasgow

Level M Exam

SAFETY-CRITICAL SYSTEMS DEVELOPMENT

Answer 3 of the 4 questions.

1.

a) Explain why standards, such as IEC 61508, can help to improve software safety across an industry and at the same time may create “barriers to entry” that restrict competition in the development of safety-critical systems.

[5 marks]

b) IEC 61508 requires SIL4 systems achieve a probability of failure on demand between 1 in 10,000 and 1 in 100,000. For continuous operation the probability lies in the range 10-8 to 10-9 failures per hour of operation. Briefly explain the problems that can arise in validating the risk reductions that are associated with SIL4 systems in both continuous and on-demand mode.

Hint: there are approximately 8,760 hours in a year.

[5 marks]

c) ISO 26262 is a new automotive standard closely based on 61508; it covers the safety lifecycle from management, to development, production, operation, service and decommissioning. It addresses requirements specification, design, implementation, integration, verification, validation, and configuration using automotive risk classes known as Automotive Safety Integrity Levels (ASILs).

Write a technical report, explaining to a car producer the range of technical and organizational problems that will arise when attempting to introduce 26262 for the first time into the software that they use in their vehicles.

[10 marks]

2.

a) A recent study of US Nuclear Digital Reactor Protection Systems revealed that each plant's Core Protection Calculator Systems (CPCS) used the following list of components:

- 6 computer boards
- 6 memory boards
- 4 multiplexers
- 6 watchdog timers
- 8 cold leg temperature channels
- 8 hot leg temperature channels
- 4 pressurizer pressure channels
- 4 upper core level neutron flux channels
- 4 middle core level neutron flux channels
- 4 lower core level neutron flux channels
- 4 RCP digital pump speed channels

Identify the strengths and weaknesses of both hardware and software redundancy using the CPCS as an example.

[5 marks]

b) The same study examined 141 incidents involving CPCS in Digital Reactor Protection Systems. 99 involved reactor trips triggered by the CPCS for plant conditions requiring a trip or where operational errors caused the CPCS to generate a 'fail-safe' trip. In other words, the CPCS functioned correctly to protect the plant. However, 26 events involved common cause failures which delayed a trip and jeopardised safe, acceptable design limits.

Write a brief technical report identifying techniques that might be used to identify and mitigate common cause failures in complex, safety-critical software using the CPCS as an example.

[5 marks]

c) A number of similarities were identified across CPCS failures. Many involved calibration errors; technicians selected the wrong data set of addressable constants and inserted them into all four CPCS channels. Others stemmed from calculation errors in generating the addressable constants – which were then loaded into the addressable constant registers. Only one event involved a software design error; processing failed sensor inputs.

Describe the relative importance of software bugs compared to calibration/configuration errors. Consider the changing importance of these sources of failure in future generations of complex control systems.

[10 marks]

3.

a) Briefly explain why it is more difficult to mitigate transient rather than intermittent or permanent failure modes involving complex software.

[5 marks]

b) Briefly explain the role of software fault injection in the verification of safety-critical systems. Your answer should distinguish between compile time injection and run-time injection. You should also distinguish between code insertion and code modification.

[5 marks]

c) You have been hired by a company developing a new family of medical infusion devices. These are programmable devices that can be used to deliver fluids, medication or nutrients into a patient. Write a technical report that explains the potential limitations of software fault injection and recommends potential solutions that can be used to address the concerns that you identify for this approach to software verificaiton.

[10 marks]

4. Human error is arguably the single greatest cause of system failure.

   Write a report to senior management explaining whether it is possible to entirely eliminate human error from the design, operation and management of safety-critical software. Illustrate your answer with a range of techniques that can be used to mitigate human error and comment on the long term effectiveness of those techniques.

   [20 marks]