University of Glasgow

Wednesday 11 May 2016
9:30 am – 11:30 am
(Duration: 2 hours)

DEGREES OF MSc, MSci, MEng, BEng, BSc,MA and MA (Social Sciences)

Safety Critical Systems Development (M)

(Answer 3 out of 4 questions)

This examination paper is worth a total of 60 marks

The use of a calculator is not permitted in this examination

INSTRUCTIONS TO INVIGILATORS

Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

1. **(a)** Briefly describe the main benefits to be derived from accident investigations for the development of safety-critical software.

[5 Marks]

(b) What factors make the investigation of software failure more difficult than other aspects of conventional accident investigation – for instance, compared to hardware or human factors?

[5 Marks]

(c) The UK Health and Safety Executive argue that:

> "It is the potential consequences and the likelihood of the adverse event recurring that should determine the level of investigation, not simply the injury or ill health suffered on this occasion. For example: Is the harm likely to be serious? Is this likely to happen often? Similarly, the causes of a near miss can have great potential for causing injury and ill health. When making your decision, you must also consider the potential for learning lessons. For example if you have had a number of similar adverse events, it may be worth investigating, even if each single event is not worth investigating in isolation. It is best practice to investigate all adverse events which may affect the public".

Write a brief technical report describing how these more general arguments might be used to improve the development of software within safety-critical systems.

[10 Marks]

2. **(a)** NASA Technical Standard 8719.13C stresses the importance of traceability in safety-critical systems:

> "Traceability is a link or definable relationship between two or more entities. Requirements are linked from their more general form (e.g., the system specification) to their more explicit form (e.g., subsystem specifications). They are also linked forward to the design, source code, and test cases. Because many software safety requirements are derived from the system safety analysis, risk assessments, or organizational, facility, vehicle, system specific generic hazards, these requirements; these will also be linked (traced) from those specific safety data products (examples include safety analysis, reliability analysis, etc.)".

Identify two problems with ensuring traceability in the development of complex, safety-critical systems and describe how to mitigate each of these problems.

[4 Marks]

(b) The NASA Technical Standard does not follow the approach adopted in IEC 61508, instead NASA-STD-8719.13C states that the overall system safety analysis is used to guide the assessment of software risk and that the following particular factors need to be considered:

- Level of autonomy,
- system complexity,
- software size,
- hardware/software trade-offs.

Briefly explain why each of these factors might contribute to the likelihood and consequence of software related failures. Why can it be hard to identify appropriate metrics to measure these software risk factors?

[6 Marks]

(c) NASA-STD-8719.13C identifies the importance of links between risk analysis and verification and validation:

> "…software safety testing will include verification and validation that the implemented fault and failure modes detection and recovery works as derived from the safety analyses, such as PHAs, sub-system hazard analyses, failure-modes-effects-analysis, fault-tree-analysis. This can include both software and hardware failures, interface failures, or multiple concurrent hardware failures. Fault detection, isolation and recovery (FDIR) is often used in place of failure detection when using software as software can detect and react to faults before they become failures".

Explain the problems that can arise when trying to maintain the links between risk analysis and verification and validation.

(Hint: you can use your answers to the previous parts of this question to support your solution)

[10 Marks]

3. **(a)** The Boeing 777 airplane required some 2.5 million lines of newly developed software. If we include Commercial Off The Shelf (COTS) and optional software this rises to more than 4 million lines of code, across 79 sub-systems from suppliers in many different countries. Identify factors that complicate the safety-critical software engineering of such systems.

[4 Marks]

(b) Boeing conducted a post-development review of the 777:

> "We found no correlation between the language used and the number of problems found in the system. We found instances where Ada was used effectively and the developers felt it substantially reduced software integration problems. In other cases development was hampered by problems with compilers and with other support tools. Many suppliers chose to use a restricted subset of Ada which led to fewer problems but lesser benefits".

Use this quote to justify the properties that you would look for when selecting an appropriate compiler for use in the development of safety-critical software.

[6 Marks]

(c) The FAA recently issued an airworthiness directive that required a software update for the 777's three autopilot flight director computers. Pilots noticed unexpected resistance when interacting with the control column, causing them to abort the take-off at high speed. Boeing identified two possible causes. Pilots could mistakenly press the autopilot switch on the aircraft's mode control panel (MCP) when attempting to engage the auto-throttle. Alternatively, pilots may mistakenly press the left autopilot switch on the MCP due to previous training in Boeing 757, 767, 747-400 models. These were rare events; in 15 years and more than 4.8 million flights there were nine reported instances of a rejected take-off because of inadvertent engagement of the autopilot without any runway overruns or injuries.

What lessons can we draw for the future development and operation of safety-critical software from these 777 autopilot concerns?

[10 Marks]

4. Write a technical report for the senior management of a safety-critical software company explaining the main causes of regulatory lag in safety-critical industries. Use this analysis to identify ways in which the company might continue to develop and market safety-critical software in areas that suffer from regulatory lag.

[20 Marks]