Xday, XX XXX 2015.

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

University of Glasgow

DEGREES OF BEng, BSc, MA, MA (SOCIAL SCIENCES).

COMPUTING SCIENCE - SINGLE AND COMBINED HONOURS
ELECTRONIC AND SOFTWARE ENGINEERING - HONOURS
SOFTWARE ENGINEERING - HONOURS

SAFETY-CRITICAL SYSTEMS DEVELOPMENT

Answer 3 of the 4 questions.

1.

a) When the autonomous mode of a self-driving car disengages, the driver must assume direct control to ensure the safety of the vehicle. Google report that the number of disengagements of autonomous control in their fleet of test vehicles has dropped from 785 miles per disengagement in the fourth quarter of 2014 to 5318 miles per disengagement in the fourth quarter of 2015. What are the implications of this increase in reliability for the human factors of interaction with autonomous software, controlled systems?

[4 marks]

[Seen problem]
Please note that many different answers are possible here – I am looking for the quality of the argument not an enumerated list of pre-specified solutions. Hence, for some questions I have given more solutions that there are marks available. The sample solutions represent my best effort to answer the question; I would not expect such full answers under exam pressure. Students will not be penalized for incorrect answers hence I do not want to write "give four reasons…" – if they give 5 and 1 is incorrect then they may still obtain full marks. This will be explained in the course and in revision classes.

This addresses the paradox of automation – software such as the control infrastructure for an autonomous car can reduce the potential for human error as a cause of an accident or incident (1 mark). However, this does not eliminate the potential for human error as a cause of the bugs or requirements problems stemming from the software development process (1 mark). At the heart of this question is the related issue of a 'driver' having to take manual control in an emergency (1 mark) when the relatively low frequency of adverse events means that they may not be expecting it or monitoring effectively (1 mark) and the impact of automation may also be to erode driving skills over time (1 mark).

b) Google's review of their tests on the roads of California has identified 69 reportable safe operation events. These included 13 where the test driver prevented the vehicle from making contact with another object. The remaining 56 were safety-significant because some aspect of the vehicle's behavior could have caused contact in other environments or situations including "proper perception of traffic lights, yielding properly to pedestrians and cyclists, and violations of traffic laws". Briefly describe the strengths and weaknesses of testing safety-critical systems in their final context of operation.

[6 marks]

[Seen/unseen problem]

Testing in the final environment of use is important because it is hard to validate the assumptions that may have been embedded within artificial verification environments (1 mark). A particular example in the course would be the deployment of RPAS that were developed at sea level and then struggled to maintain power when operating at altitude around Kabul – or the Abrams tank power plant problems suffered in desert environments that were not anticipated pre-deployment to Iraq (an example is not requested in the question but I will advise students to consider doing this to clarify their thinking, 1 mark). It can also be hard to anticipate the range of influences and contextual factors that might arise in a real working environment (1 mark). Balanced against these strengths are considerable concerns – for example, there are ethical issues if a *safety-critical system causes injury or death during a field trial (1 mark). It can be difficult to repeat tests to diagnose the causes of a problem (1 mark) – in a lab it is possible to replicate independent variables in a more focused way (1 mark).*

c) Companies including Audi, Mercedes, Tesla, Bosche are all developing similar autonomous vehicles. It is possible to identify common ideas across many of these vehicles, integrating GPS, laser illuminating detection and ranging (LIDAR), radar, high-powered cameras and learning algorithms to 'sense and avoid' other objects in their environment.

   Traditionally, artificial intelligence has been explicitly forbidden from higher Software Assurance Levels. Write a brief technical report for the senior management of a car company, explaining why regulators are reluctant to permit the use of AI in safety-critical applications and then summarizing the arguments that might be used to support the use of learning algorithms in self-driving cars.

[10 marks]

[Unseen problem]

AI and learning algorithms are not, typically, permitted at higher levels of assurance within safety-critical systems because regulators cannot be sure that any verification (1 mark) and validation (1 mark) activities that were conducted prior to deployment will provide assurance against the code that might be running in the eventual context of use (1 mark). Or put another way, the learning algorithm might learn inappropriate or unsafe behaviors after it has been deployed (1 mark). It becomes hard to reason about all possible future behaviors of the system (1 mark) – tests must accept state incompleteness or have some means of reasoning over the adaptive mechanisms that are embedded in the software (1 mark).

Answers to the earlier parts of this question provide one mitigation to the potential risks of AI – it can be argued that a human operator retains overall (1 mark) control and hence has the possibility of intervening to prevent an accident should the system 'learn' an inappropriate behavior (1 mark). An alternative approach is to limit the extent to which AI techniques might modify the behavior of the software (1 mark). In other words, the system places constraints that are intended to increase the predictability of system behavior and maintain safety (1 mark). Other arguments might be based around the flexibility of AI and self-learning systems (1 mark). For example, if lab based tests made inappropriate assumptions about the context of use then self-adaptation might help to maintain safety in a manner that is more resilient than conventional programming techniques (1 mark).

2.

a) Briefly explain the key concepts behind resilience engineering.

[6 marks]

[seen/unseen problem]
Resilience engineering focuses on those properties that enable systems to respond and recover from adverse situations (1 mark). One of the key ideas is that human operators provide the flexibility that is required to adapt over time (1 mark) (see link to question one – draw the link to AI would give 1 mark) and which cannot easily be built into complex, automated applications (1 mark). It follows that we should identify and strengthen these adaptive behaviors that occur every day (1 mark) rather than focusing on the rare and unusual responses to accidents (1 mark), which may provide poor insights into the properties and behaviours that help to maintain safety for 99.999999999% of the time. Accidents when they do occur are the result of similar mechanisms – Hollnagel argues that these adaptations are constrained by finite resources (time, skill, money) hence there are limits to the flexibility described above (1 mark).

b) Identify the main challenges that can arise when introducing incident reporting into safety management systems for large-scale software engineering projects. Briefly explain how you would address these challenges.

[7 marks]

[seen/unseen problem]

Safety management systems typically begin with risk assessment and hazard analysis to identify the potential concerns in a complex development project. Answers may follow the ideas in 61508 and similar standards – the focus of the hazard analysis is to identify the system (Equipment Under Control) risks. This can be used to identify safety objectives through the allocation of a Safety Integrity Level that in turn helps identify appropriate software development practices (the second stage of the SMS). Once the system has been deployed, incident reporting systems provide feedback necessary to refine the risk assessments and driving further cycles in the SMS through redevelopment (max 2 marks for the description of an SMS but no penalty if they go straight into the question).

Some of the challenges include under-reporting, some units or types of employee will not report – for example sub-contractors (1 mark). Other challenges include causal analysis (see part c) (1 mark). Even if incident data can be identified it is hard to identify appropriate recommendations (1 mark) or then to ensure that they are implemented (1 mark). Potential solutions include the use of Sentinel techniques (focus resources to identify incidents in one unit then extrapolate) (1 mark). Automated and log analysis techniques might also be used to identify concerns without people having to report (1 mark). There are many different root cause analysis techniques (eg Eindhoven method) (1 mark). The implementation of recommendations can be tracked – for instance by having senior management review open/closed findings (1 mark). Other answers might mention the NTSB top ten recommendations list (1 mark).

c) Write a brief technical report for software engineers, explaining why root cause analysis raises non-trivial problems during the analysis of degraded modes of operation or after major incidents involving complex, software systems.

[7 marks]

[seen/unseen problem]
As with the previous question, there are many answers. One approach would be to look at the counterfactual foundations of root cause analysis. Most techniques that support RCA are based around an assumption that a root cause is one that if it were prevented then the accident would not have occurred (1 mark). This is non-truth functional. In other words, we know that the accident did take place hence we have no way of proving that the accident would not have happened – by going back in time and making the proposed change in blocking the root cause (1 mark). In particular, we might be looking at one of several ways in which an accident might have occurred – if we stop people from smoking because this led to a fire, we would not prevent further fires with other ignition sources (1 mark). Given the lack of any absolute proof that eliminating a root cause could have prevented an accident, there are considerable concerns over bias (1 mark) – for example blaming the operator rather than looking at underlying design concerns. This might lead to the cheap option of changing procedures rather than investing in redesign – with no guarantee that operators will follow the new rules (1 mark).
Software causes particular problems because so many aspects of engineering are not based on empirical evidence. This includes the benefits of standards – which can only be inferred and cannot be directly demonstrated for instance through lab studies (1 mark). Hence we might argue that an accident would have been avoided if the developers had followed MIL-STD882C but it is hard to provide objective proof for such statements (1 mark).
The role of software as a root cause is often ignored or underplayed in accident reports because many investigators lack the necessary skills in forensic analysis (1 mark). It is also very hard to replicate the exact input values that might have led to an accident – hence in many cases, we cannot accurately reproduce the problems that led to failure (1 mark). Root cause analysis of software is hard because we cannot simply stop with a bug or incorrect requirement (1 mark). We should trace it back through development to understand the reasons why the bug existed into the delivery version of the code (1 mark). Very few investigators have the time or other resources necessary to trace the causes of the failure back in this way (1 mark).

3.

a) Briefly explain why we might need to use both qualitative and quantitative risk assessments during the development of complex safety-critical systems?

[4 marks]

[seen problem]
This has links to the previous questions – for instance, in an SMS we assume that incident and accident reporting might be used to inform risk assessment, through quantitative techniques (1 mark). These approaches can be used for both consequence and for probability (1 mark). However, we must recognize limitations – problems of under-reporting lead to incorrect probabilities and differences in context can complicate any comparison of consequences (1 mark). We can also use the results of destructive testing and other lab based studies to drive quantitative risk assessments (see links to the first question with the field trials of the Google car (1 mark). The key argument here is that if we have quantitative data then we should use it.
However, in many cases quantitative data might not be available. If we have a truly innovative design we need to assess risk to identify appropriate levels of safety but how can we do this when the system has never been used before? (1 mark). In such cases, we may need to poll a range of experts to derive qualitative risk assessments ensuring that we account for subjective differences (1 mark). It may not simply be a novel design; we may also need qualitative methods because there are changes in operating procedures or in the environment (1 mark). This implies that in most cases there will be an element of judgment in risk assessment – we seldom have exactly the statistics that we might desire. I would not expect such a full answer so solutions that only address a few of these points will attract full marks.

b) Describe the role of safety integrity levels (SILs) within the IEC61508 standard and explain the differences between SILs and the Automotive Safety Integrity Levels within the ISO 26262 standard.

[6 marks]

[unseen problem]
Safety integrity levels measure the necessary risk reduction to move an assessment of the risk from Equipment Under Control to a point where the system is acceptable safe (1 mark). This risk reduction is achieved, typically, through the development of a safety system (it might also be done through placing constraints on operation or the environment of the system under consideration but this could be modeled through a reduction in the EUC risk). In 61508, there is a four-valued SIL component with SIL 4 representing the highest level of risk reduction (1 mark). Each level places successively greater constraints on the development practices (1 mark). For example, SIL 1 is assumed to represent a minor reduction in risk and hence conventional software engineering practices may be sufficient to demonstrate this has been achieved within a project. At SIL 4, there will be limits on the choice of programming language – constructs such as dynamic variables/heap use, will be prevented etc (1 mark).
26262 is derived from 61508 and there are some similarities with SILs, ASILS are measured from A to D where D is the highest (1 mark). ASIL = SEVERITY X (EXPOSURE X CONTROLLABILITY) (1 mark), a key difference here is the intervention and role of the driver. Hence an ASIL considers controllability – if a driver cannot control the vehicle then this is likely to lead to ASIL D, ceteris paribus. One way of

thinking of these metrics is that a SIL represents a degree of necessary risk reduction, while an ASIL represents the degree of risk (1 mark).

c) What is 'regulatory lag'? Write a report for a senior government official explaining how you would address the problem of regulatory lag in software development within safety-critical autonomous systems that interact with members of the public.

[10 marks]

[unseen problem].

Regulatory lag describes situations in which companies seek to innovate at a rate that is not matched by the guidance on acceptable means of conformance provided by the regulatory authority (1 mark). This could be illustrated by confusion over the operation of RPAS in controlled air space (1 mark) or the degree of cyber-security protection that should be provided by the operators of national critical infrastructures (1 mark). Regulatory lag causes concern because it can prevent a new industry from growing (1 mark). Companies will be reluctant to invest because they cannot determine what they might need to do to satisfy the regulator (1 mark) – they might be prevented from operating or if they can operate the lack of definitive regulatory guidance might create scope for litigation should an accident occur (1 mark). From the regulators point of view, they often lack leading edge technical skills with fiscal constraints (1 mark). They may also be reluctant to issue inappropriate guidance before an industry is mature and the risks are better understood (1 mark). There are significant differences across Europe and North America – in the USA, there is a tendency to offer waivers to other regulations allowing industry to develop and gathering the insights sufficient to draft new regulations. In Europe, there is a tendency to extend existing regulations but after some period of time – hence creating uncertainty (1 mark). In the course we will discuss the regulations covering sub-orbital flights as an example of this (1 mark). These two approaches can be contrasted in the second part of the question. The first question in this exam, dealing with Google cars represents permission for limited field trials and can be thought of as a variant on the waiver approach (1 mark). There is no blanket approval for the introduction of autonomous cars but certain exceptions to existing regulations are allowed to enable the industry to develop (1 mark). An alternative approach would be to require that all autonomous systems have to reach the level of reliability exhibited by a human driver (meeting existing regulations) before it can be deployed. This might hold the industry back for many years (1 mark).

4. Safety is a non-functional requirement.

Write a brief technical report to a Chief Executive Officer who only understands office based software systems, explaining why non-functional requirements cause particular problems for the management of complex, software projects.

Identify at least three different measures that might be introduced into the software development lifecycle to help manage non-functional requirements from conception through to maintenance.

[20 marks]

[Essay – unseen/seen problem]
There are many different solutions here with ample space for more able students to focus on particular examples that we have met during the course.

We can think about the Equipment Under Control following the distinctions made in standards such as 61508 (1 mark). This is the raw functionality of the system under consideration. The safety functions help reduce risk to an acceptable level (1 mark). In other words, we can consider safety as a set of requirements in addition to those that are essential for the system to function. Other non-functional requirements include security (1 mark), usability (1 mark) and reliability (1 mark) (note that a system may function correctly for a short period of time and then fail in a safe way, hence reliability is not the same as safety). We can identify several common features across non-functional requirements. They are typically, relative concepts not absolutes (1 mark). We cannot be 100% safe – because there may be hazards we have not considered or changes in the environment that we cannot anticipate (1 mark). In the same way that we cannot be 100% secure – this would require complete knowledge of all potential attack vectors, which is impossible (1 mark). Nor can a system be 100% usable – by every possible person in all possible working environments (1 mark).
A further common problem across non-functional requirements is that we cannot determine easily how much to spend in achieving a particular property (1 mark). Given that we cannot be 100% safe – how much is safe enough? (1 mark) Regulators provide acceptable means of compliance and associated guidance to help with this (1 mark). In the UK, we have the concept of risk reduction being As Low As Reasonable Practicable (ALARP) (1 mark). In the course we also mention ALARA and MEM as different regulatory approaches (1 mark).
It is also difficult to manage the achievement of non-functional requirements because we must focus on the validation of verification criteria (1 mark) – in other words, how do we demonstrate that we have achieved a target level of safety? (1 mark) We have spent a long time in the course covering the distinction between product and process based techniques for regulation (1 mark). At the heart of this is the notion that we cannot test a system to demonstrate that it is safe – over hundreds of thousands of hours of operation by millions of users around the globe (drawing on the answer to question 2) (1 mark).
Examples of techniques for managing non-functional requirements might include the development of safety-cases (1 mark) – we have shown the use of GSN in various lectures. These link higher level arguments about the acceptable level of safety in design and operation of a critical system to the underlying documentation that provides evidence in support of these arguments (1 mark).
A second technique could be the use of formal or semi-formal reasoning to show that specific safety-related properties hold across an implementation (1 mark). One way of thinking about a safety-critical system is to associated the functional requirements with the axioms that lead to model development while the safety properties that are proven of those associated states are the non-functional constraints (1 mark).

A third approach might be the traceability techniques advocated within DO-178C – where we can move from the high level statement of non-functional requirements to the specific code that is intended to ensure that particular properties are met (1 mark). This is essential what both GSN and the use of formal methods also provide (1 mark). However, it is hard to use surface level traceability techniques to show that a property always holds rather than to trace that a particular line of code implements a specific safety function (1 mark) – hence there is scope for the combination of these conventional software engineering techniques with modal checking etc (1 mark).