Xday, XX XXX 2009.

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) The UK Health and Safety Executive provide guidance for companies on their responsibilities under the Control of Major Accident Hazards Regulations 1999 (COMAH). This documentation includes the following statements on the decommissioning of safety related applications:

"*General measures that should be adopted for a common approach to decommissioning include:*

- *Establish communication with plant personnel to ensure surrounding plant areas are prepared for decommissioning activity;*
- *Undertake removal of hazardous substance via a cleaning procedure to ensure plant item is clean and empty with particular consideration where there may be dead-legs where material may be trapped;*
- *Consideration of the disposal of items which may be contaminated by absorption of hazardous substances and chemical change;*
- *Mechanically isolate plant item from other surrounding plant items by physical disconnection or fitting of blanks;*
- *Electrically isolate plant item from power sources by physical disconnection*".

Briefly explain why this guidance does not explicitly refer to the role that software systems might play in the decommissioning of Major Accident Hazards.

[5 marks]

[Seen/unseen problem]
The principle reason why the HSE do not focus on software within the COMAH guidance is that it focuses on Equipment Under Control as the main source of any hazard. In other words, it follows the approach embedded within standards such as IEC61508 in which software is viewed as a mitigation of risk rather than a source of hazard given that it is the EUC which directly causes injury or loss. A secondary reason may be that the COMAH regulations have their roots in the early 1990s when software was less tightly embedded within safety-critical systems. More explicit mention might be made under other sections of the COMAH guidance on decommissioning if the documentation were to be re-written today. It can also be argued that decommissioning focuses on the process components or EUC. Any decommissioning that involved only programmable systems would be seen as a more minor plant modification and so would not be covered by this HSE guidance.

1 mark for each reasonable argument with plenty of scope for changes to the examples given here.

b) The COMAH guidance encourages the use of HAZOPS as a means of assessing the risks associated with the decommissioning of safety-related applications. This technique uses process flow diagrams to identify the intentions behind different components. For example, a pump may be intended to deliver coolant to a reactor. Keywords, such as "NO OR NOT", "REVERSE", "MORE", "OTHER THAN", "LESS", "SOONER THAN", "AS WELL AS", "LATER THAN", "PART OF", are used to identify the effects of deviations in these intentions.

How might this approach be extended from the decommissioning of safety-critical applications to support the analysis of safety requirements for the decommissioning of programmable systems in COMAH processes?

[5 marks]

[Unseen problem]

Attempts have been made to apply HAZOPS techniques to software – there is an example in the sample solutions to last years' exams which the class will have seen. . An argument can be made that HAZOPS is appropriate for assessing the risks associated with software decommissioning, 'SOONER THAN', 'LATER THAN' and 'AS WELL AS' all having reasonable interpretations within the context of real time and concurrent software systems. However, these general purpose guidewords were never intended to be sufficient for all domains and so good solutions might identify a number of alternatives. These could focus on traditional software engineering issues, such as 'buffer overflows' or 'type incompatibility'. Some of the Ada predefined exceptions might be mentioned? However, this is a relatively low level approach that runs against the more generic relationships within HAZOPS. The following table shows how an alternate form of software hazard analysis (SHARD) introduces a number of more specific guidewords for a single language (in this case MASCOT). I would not expect this in the final solution to the above question but it illustrates this important point about the granularity of the keywords that are chosen in any potential solution.

		Failure Categorisation					
Flow		**Provision**		**Timing**		**Value**	
Protocol	**Type**	*Omission*	*Commission*	*Early*	*Late*	*Subtle*	*Coarse*
Pool	Boolean	No update	Unwanted Update	N/A	Old Data	Stuck at...	N/A
	Value	"	"	"	"	wrong in tolerance	out of tolerance
	Complex	"	"	"	"	Incorrect	Inconsistent
Channel	Boolean	No Data	Extra Data	Early	Late	Stuck at...	N/A
	Value	"	"	"	"	wrong in tolerance	out of tolerance
	Complex	"	"	"	"	incorrect	inconsistent

I would also welcome a more systems oriented approach where the answers considered issues such as side effects or undocumented features that could be considered during software variants of HAZOPS – especially given that these undocumented features may be a particular problem for decommissioning legacy software. The key point here is that the analysis must build on the previous exam answer that some of them will have looked at by addressing specifically the use of HAZOPS in software decommissioning/replacement. We have mentioned the problems that NASA have faced in maintaining the Shuttle software as the original processors become increasingly hard to obtain.

2 marks for explaining the use of HAZOPS guide words when applied to software, 2 marks for relating these uses to the specific issues in decommissioning of programmable systems. 1 mark for mentioning the limitations of HAZOPS and 1 mark for identifying differences in the use of HAZOPS for decommissioning software versus application processes up to a total of 5.

c) The Health and Safety Executive have recently published the guidance that they give their inspectors on how to assess the Process Safety Management Systems and Safety Cases that companies must develop when they seek approval to operate safety-critical systems. One of the most important sections of this document is a footnote, which states that

"Note: a particular problem found by HSE's inspectors has been of HAZOP actions remaining unresolved or uncompleted".

Explain why this problem might also affect the use of HAZOPS in the decommissioning of safety-critical software. What steps might you take in order to ensure that the actions identified in a HAZOPS study are addressed in an appropriate manner?

[10 marks]

[Seen/Unseen problem]

The previous answer has argued that undocumented features are likely to be a particular problem for the decommissioning of software systems. Similarly, it is likely that legacy code may fail to address known problems that have been identified in previous risk assessments using techniques such as HAZOPS. In practice, a series of informal 'work arounds' are often easier to implement that develop revisions to complex programs. This creates a host of hazards for decommissioning – undocumented features, hazards that have not been addressed, or hazards that have been addressed in an informal manner may not easily be identified as requirements for future generations of a software system. Hence the new version of an application may suffer from the same weaknesses as previous systems. There may be further more specific risks associated with the decommissioning process as software engineers may not understand the knock-on effects of their actions in uninstalling software components.

A range of solutions or mitigations can be adopted. Some of these focus on ensuring that haards are addressed or at least documented in legacy code. Safety management systems, traceability techniques, software safety audits might all be mentioned. However, all of these techniques suffer from problems of complexity and scale. Other issues relate to communication between multi-disciplinary teams where software engineers may not be involved in the day to day operation of a complex system etc. Other answers might focus on a staged approach to decommissioning given that there can be undocumented features in a legacy application – gradually introducing new applications that are written to interfaces identified within the existing code – or the parallel operation of previous implementations with novel systems, allowing for roll-back should problems arise with new implementations etc.

2.

a) The UK Civil Aviation Authority has published CAP 722 on *Unmanned Aircraft System (UAS) Operations in UK Airspace – Guidance* this includes the requirement that "any UAS outside a UK Danger Area will not increase the risk to existing users and will not deny airspace to them". In particular, CAP 722 makes it explicit that Unmanned Aircraft Systems must provide "a level of safety and security equivalent to that of manned aviation".

Briefly explain the challenges that must be addressed before any operator or manufacturer could demonstrate that they have met an equivalent level of safety to manned operations.

[5 marks]

[Unseen problem] We have discussed the integration of UAS' into controlled airspace in the lectures as an example to other aspects of the course – eg risk assessment. We have not discussed in detail the issues surrounding equivalent risk between UAS and manned operations.

It is likely that the risks created by UAS' will be of a very different nature to those created by manned operation. There is arguably less concern over the impact of pilot fatigue but alternate problems may affect the programming of UAS platforms. Until we know more about the operational deployment of UAS in integrated airspace it may be difficult to anticipate the nature of these risks and the interactions between different types of traffic.

It can be difficult to identify the metrics that might be used to demonstrate risk equivalence. For example, the number of airprox's is extremely low – this leads to large statistical variations – apparently random causes may lead to 100% fluctuations in the number of adverse events from month to month involving manned flight.

Given the relatively low number of Airprox's, the relative safety of manned and unmanned operations might be measured in terms of a higher number of 'near airprox' incidents. These can be more difficult to detect and validate. A near airprox may be relatively severe if not identified by an ATCO – or it may pose little concern if the vehicles are correctly advised. This in turn raises further questions about the risks created by a near airprox in which one or possibly none of the vehicles involved are under direct supervision from an ATCO.

Finally, answers might look at the ethical and practical problems of conducting trials to assess the risks created by unmanned air vehicles outside 'Danger Areas' – these issues complicate attempts to validate theoretical arguments about risk equivalence. After the first mid air collision involving a UAS it is likely that calls will be made to ban further integration even if the risk is no greater than might otherwise be realized by the same frequency of manned operations.

b) One way of achieving the integration of UAS' into controlled airspace is through the development of 'Detection, Sense and Avoid' (DSA) systems that enable the UAS to identify and then avert a potential collision with another air vehicle. The US Standards group ASTM has issued a standard for DSA technology (F2411-04 DSA Collision Avoidance), which may become the basis for future certification requirements by the Federal Aviation Administration. The ASTM standard requires that a UAV detect any airborne object within a range of + or - 15 degrees of elevation and + or -110 degrees of azimuth. The UAV must also be able to act on this information so that collision is avoided by at least 500 ft.

Explain the problems that manufacturers face in verifying that on-board software is capable of meeting the DSA requirements within ASTM standard F2411-04.

[5 marks]

[Seen/unseen problem] This question is deliberately intended to challenge the stronger students. It may seem a little intimidating but it is not very difficult. The key idea is to build on the answer to part a) – we know very little of the operational context for UAS' in integrated airspace. Hence

it may be very difficult to devise validation criteria for the tests that might be used in demonstrating that DSA software can meet these high-level requirements. This includes not simply the performance and configuration of the UAS but also the performance and operational profiles of all other airborne vehicles (and ground obstacles) that may be encountered during the operational life of the UAS. One meta-level complication is that these vehicles must also include future generations of other types of UAS' also with their own versions of the DSA – giving rise to concerns over variations on the TCAS revision logic that were raised after the Ueberlingen accident (we discussed TCAS during the course).

In the course we have covered the principle means of static and dynamic testing – so some solutions may go on to consider not just validation but also the challenges of verification. There are huge practical problems in establishing the traceability requirements that are created by the high-level constraints of ASTM F2411-04. Can we use formal or semi-formal techniques to provide static proof that these criteria will be met by an implementation – especially given the real time elements of the standard which also make assumptions about the operational performance of other vehicles closing on the UAS?

c) Describe the ways in which 'white box' and 'black box' testing might be combined during the certification of DSA software in Unmanned Aircraft Systems. How might data from previous accidents involving both manned and unmanned aircraft be used to direct the application of these techniques?

[10 marks]

[Seen/unseen problem] There are two aspects to this question. The first focuses on the relationship between white box and black box testing. White box testing assumes that teams have access to details of the internal implementation. In terms of the DSA requirements, mentioned in part b of this question, white box testing may be required to ensure that the software is likely to meet a wide range of possible operational scenarios with multiple potential targets in the range for azimuth and elevation. Verification across a range of scenarios may be far more difficult to establish using blackbox tests – which imply the need for test rigs to simulate many different conditions across given that the internal implementation details may not be visible to those conducting the verification. However, blackbox tests are more likely to encourage independent analysis given that the details of DSA software implementation are likely to be highly complex. There will also be considerable issues of commercial sensitivity given the potential market for certified DSA software.

The second aspect of this question relates to the use of accident information in the certification and testing of UAS applications. There are links to this question from sections a) and b), which I hope will be identified by the more able students. It is also possible to refer between questions – by mentioning links to the precepts of resilience engineering, mentioned in question 3. The intent behind this question is to challenge whether it is possible to anticipate the future risks created by the integration of DSA software into controlled airspace using previous accidents. Resilience engineering stresses the ways in which organizations and individuals will change their behaviors as new systems are introduced. Hence, it is difficult to anticipate future risks from past accidents. The previous sections have stressed the role that a diverse range of operational scenarios must play in both white or black box testing but if we cannot rely on previous accidents as a guide then it is difficult to know how best to identify the precise nature of these scenarios? With such uncertainty, I would expect first class answers to discuss the need for gradual extensions to the existing Danger Areas and to closely monitor the safety performance of UAS' across a wide range of metrics to identify any unintended consequences that might threaten safety even when the ASTM F2411-04 or comparable requirements are met.

More information about many of the issues in this question can be obtained from http://www.uavm.com/uavregulatory/collisionavoidance.html

3.

a) ‘Resilience Engineering’ has recently criticized the use probabilistic risk assessments and probabilistic safety assessments because they focus on a small number of previous failures and, therefore, neglect the many different ways in which organizations respond in a flexible way in order to maintain safe and successful operation.

Briefly explain whether or not you believe the concepts of resilience engineering can usefully be applied to support the software engineering of complex safety-critical systems.

[6 marks]

[Unseen problem] There are many different answers to this question – I am looking for a reasoned argument rather than any particular orthodoxy. My own view is that resilience engineering has provided a useful corrective to the previous fixation with SILs and other risk based metrics. However, there is a danger of criticizing techniques that have a demonstrated engineering value before we have sufficient tools and methods to put in their place. I, therefore, view the recent correctives as important to improve our thinking about the development of safety critical systems but not as a coherent replacement for PRA and PSA. Others will disagree and this is ok in terms of the marking.

Focusing on software engineering, I think there are strong parallels between the more general critical of resilience engineering and the criticisms that are often made about the development problems affecting safety-critical code. We are tempted to fixate on a small group of well known case studies including Therac-25, Ariane 5 and the London Ambulance Computer Assisted Dispatch system rather than consider the reasons why so many other applications have been successful.

Other answers may look more at some of the issues that were developed towards the end of question 2 – they may explore the difficulty of anticipating future failures based on past accidents. A particular example would be the relative absence of software as a causal factor across serious accidents within the world’s aviation industry. Can we really prepare for the challenges created by future generations of avionics applications by looking at two or three incidents that have occurred? Resilience engineering encourages us to look at the ‘other side of the coin’ and reinforce the sound engineering and operational practices that can be identified in millions of successful flights rather than very rare qccidents.

b) In order for Resilience Engineering to be successful, it has been argued that companies must be provided with:

1. The means of measuring and monitoring the resilience of organizations in particular operating environments;
2. Tools and methods to help improve the level of resilience identified through the application of measurement techniques identified in point 1;
3. Techniques to predict short and long-term effects of the changes introduced by the use of tools and methods developed in point 2.

How might these requirements for resilience engineering be applied to study the safety related processes of a company involved in the development of complex software systems?

[4 marks]

[Unseen problem] These are challenging observations made by the proponents of resilience engineering. Point one suggests that we must identify ways of measuring the adaptations that companies make to their products and processes in order to meet the risks and hazards that could not easily have been anticipated before they were introduced. The very flexibility of these

adaptations implies that it may not be possible to predict their nature before a project has commenced. This runs counter to many of the traditional precepts of software engineering where key performance indicators are identified early in the conceptual stages of development – often before requirements are fully gathered.

The second precept also creates further challenges for traditional software engineering. It has often been observed that many existing techniques have to be 'tailored' for particular companies. In safety-related industries this has led to the integration of formal and semi-formal analysis to provide common but cost-effective approaches between the safety core of an application and other less critical software components. The proponents of resilience engineering stress that we should encourage these flexible adaptations, we should measure and monitor them and create a feedback look to improve the support that they provide to the ultimately safety of an application. This can create tension when, for instance, the proponents of an original approach feel that it is being compromised or undermines by ad hoc adaptations. There is a general concern to ensure that the rigour of particular software engineering techniques is not being undermined by adaptations which might be justified by economic rather than safety concerns. Hence, it is likely that resilience engineering requires greater levels of management support and external scrutiny than other forms of traditional safety-critical engineering.

c) In a recent study of safety within NASA, David Woods identified five different ways of assessing the resilience of an organization:

1. "Preparedness/Anticipation: is the organization proactive in picking up on evidence of developing problems versus only reacting after problems become significant?
2. Opacity/Observability—does the organization monitor safety boundaries and recognize how close it is to 'the edge' in terms of degraded defences and barriers? To what extent is information about safety concerns widely distributed throughout the organization at all levels versus closely held by a few individuals?
3. Flexibility/Stiffness—how does the organization adapt to change, disruptions, and opportunities?
4. Revise/Fixated—how does the organization update its model of vulnerabilities and the effectiveness of countermeasures over time?"

Identify ways in which these dimensions characterize the causes of previous accidents and explain how they might be used to support the software engineering of future systems.

[10 marks]

[Unseen problem] This is an open-ended question with enough scope I would hope both for the stronger students to really expand on the strengths and weaknesses of resilience engineering across the syllabus of the course but also with scope for other students to bring in the examples that have been provided through the analysis of previous adverse events. For instance, the Columbia final report provides examples of each of these issues, other examples might include opacity/observability during the development of the LASCAD application, revise/fixate might be applied to the software updates at Ueberlingen and so on.

However, the previous parts of this question have raised some of the concerns that arise in applying these techniques or dimensions in a predictive manner to identify ways of improving future safety rather than tracing the causes of previous accidents. For instance, flexibility can be seen as an important strength in resilience engineering where it supports dynamic adaptation to changing circumstance. However, it might also be used to justify ad hoc changes in existing processes that serve to undermine sound management principles. Similarly, observability offers important strengths where individuals can access critical safety information. However, it can also

carry costs where teams are overwhelmed with safety information that is irrelevant to their core tasks.

First class answers should draw the links between these dimensions and the more process oriented criteria for resilience engineering that were introduced in section b) of this question. Metrics and processes need to be developed so that companies can manage performance along these different attribute scales.

4. Previous initiatives to improve safety culture have focused too much on the operator and too little on the Board of Directors. Discuss.

   (Hint: you should illustrate your answer by referring to previous accidents. Reference should also be made to the application of risk assessment techniques and to the study of human factors in safety critical systems).

   [20 marks]

   [Unseen problem/bookwork/Essay]
   There are many different approaches to this question. The intention is to give everyone something that they can work on but also to provide the space for more able students to draw links between many different areas of the course. The students have seen sample solutions from the many different previous exam papers where the final question has always been an essay option. I will also provide a revision class that mentions some of the issues surrounding safety culture, this follows a pattern established in previous years.

   There have been a number of recent initiatives to extend the development of safety culture concerns from the workplace into the board room. The INSAG documents from the nuclear industry have been mirrored by recent work in Air Traffic Management within EUROCONTROL. The aim has been to promote a 'priority for safety' and 'learning from safety' at those levels within an organization that have the greatest influence on the budgetary and policy constraints that have often undermined safety initiatives targeted more narrowly on the workforce. During the course we have referred to the importance of organizational and managerial factors in accidents – we have looked for example at the role of Safety Management Systems in Ueberlingen and the role of NASA policies in Challenger and Columbia. We have also considered how some risk assessment techniques do focus specifically on managerial and organizational factors – hence MORT could be cited as a counter example to the argument in Question 4. If this is done then it should be mentioned that MORT focuses on the operational aspects of management and may not consider many relevant aspects of board level policy.

   A number of other issues can be introduced into a wider discussion of safety culture provided that these factors are linked back to the distinction between higher levels of management and operational issues. In particular, there is huge controversy the definition of safety culture and how to measure it even amongst the better known areas of workplace initiatives without mentioning how to measure safety culture in the board room. Recent changes in the English provision for corporate manslaughter are likely to increase interest in this area, however.