Harry Langford
hjel2@cam.ac.uk

Software and Security Engineering Supervision 3

There is a list of past exam questions and relevant courses on the 2021 course pages.

1 Leaks

- The question would want you to state that the requirements were convoluted.
- Accidents are the most likely way of leaking information.
- You should have gatekeepers (tests) which it is not possible to bypass.

  You would use proxy tools to gatekeep. Have multiple people set these. For example setting a time of release.
- Making people work on a specific machine in an inaccessible room would remove the chance of accidental error.
- Consider insider threats.

2 Deterrent

- Make threats tamper evident. If someone steals an exam question and you are guaranteed that you can change it then you know they won't bother.
- "Blame and train" is completely the wrong approach – if you set the rules to be stupid then people will subvert them and the only solution is to make a policy which is easier to use.
- Certain corporates will deliberately phish all of your staff to try and find out who will get phished. This is very bad and you end up blame and train and you end up getting noncompliance elsewhere.

3 Agile Development

- If you want to work out which strategy is best then you need to figure out which strategy is best. This is usually having short deadlines. However in some cases this comes across as coercive. Good project management is about figuring out which is best for different people and managing them like that.
- Agile suggests 2 week deadlines because most people will only work intensively in the last week. You want to balance the stress of short deadlines with getting things done.

  If you can't split a task up then you will "be screwed" – you have to split tasks up as much as you can – it makes things more achievable.

  Work on your worst problem first. If you can't solve the worst problem then you should scrap the project.

  Even if you're not being told to solve your worst problem then you should still do it.
- You should do PERT charts to find out the critical paths and then make GANTT charts of the critical paths. GANTT are for managing people and for management.
- If anything on the critical path gets delayed then it will delay the whole project. You use the slack time on certain paths to decide which people to use and when.
- Automated regression testing checks you don't reintroduce bugs. People are likely to make the same bugs because they're intrinsically complex and fixes are often brittle.
- Automatedd Regression Testing is a *part* of Continuous Integration! You will run automated tests (automated regression tests, unit tests and system tests).

- In nightlies (nightly builds), you run a ton of tests and build. If it fails some tests then you roll back to the last day it worked. If it passes then you release a nightly build.
- What people say in scums are usually diffs based on the previous day. What they say is very similar to the days before. The message that comes out each day is nonsense. It's motivational to know that you've got to tell people what you've been doing.

  Managers aren't allowed to talk in scrums. Systems such as agile allow you to subvert managers – it's not a great way to run a company if you've got good managers. It forces managers to listen.

  If you have a problem and say it in a scrum then you may find other people who have found the same problem and can help you.
- Git is a usability disaster although version control is necessary.

4 Protocols

- Timestamps don't need to be physical timestamps – they can be logical timestamps (which are just counters).
- Consider man-in-the-middle attacks and consider brute force attacks! Messages which are signed need to have the sender/receivers ID. You can usually interpret timestamps to be nonces.