Lol this document may contain errors or be incomplete. Please change where necessary and take solutions with salt. True facts.

QUESTION 1 (Total: 24 marks)

During this course a number of different software engineering process models were discussed.

a) Evaluate the similarities between the incremental (or iterative) and agile process models and describe two of these similarities.

(4 marks)

Both software processes aim to release early and often. This to break down large projects into smaller subsections to reduce time wasted on making decisions that will be changed later, catch errors before they are too large and provide value early.

Both software processes are better suited for smaller projects. Majority of agile and lean projects are ideal for smaller teams (e.g. agile < 12 developers) due to the processes involved. This is in contrast with the waterfall.

Both software processes aim to be open to change.

b) Assess the differences between the agile and lean process models and describe one of these differences.

(2 marks)

Agile uses fixed, uniform time boxes in each iteration of the process. For example, each sprint is exactly 2 weeks. However, for lean, each iteration can be for different lengths depending on the amount of work required.

Agile is a framework outlining exact processes and roles required. However Lean is a set of principles which developers can implement at their discretion.

c) In the lectures, data from the application of different process models was used to show the percentage of projects using each process model that were successful, challenged or failed. Projects using the incremental (or iterative), agile and lean process models had very similar percentages of being successful, challenged or failed. (Lean had slightly more successful and less challenged projects than the others, but they all had very similar percentages of failed projects.)

i. Provide a justified explanation as to why there are these similarities between these three process models.

(6 marks)

ii. Describe one characteristic of each of these three process models that supports your explanation in part (i).

(6 marks)

iii. Provide a justified explanation as to why these three process models had significantly better outcomes than plan driven (or traditional) process models.

(6 marks)

i)

- Lean, agile and incremental all focus on breaking down large tasks into smaller subsections.
- All 3 methods focus on making decisions as late as possible so that no time is wasted on making decision that will be changed later.
- All 3 methods focus on building in quality as they go rather than at the end

iii)

- Agile, Lean and incremental processes are about iterative cycles of development and hence, fault rectification is possible in subsequent iterations. More feedback is involved as compared to traditional processes, and priorities can be changed quickly.
- Tasks can be co-current, teams can be multi-disciplined and there is a lot of flexibility in the process models as compared to the rigid, uni-directional flow of a traditional process model. This helps teams reprioritise, provide better values and products, and thus a significantly better outcome.

QUESTION 2 **(Total: 20 marks)**

a) For the Commonwealth Games website project, when performing release testing, describe four critical issues that the test team should be attempting to identify.

(8 marks)

NOTE: Their assignment was about building a commonwealth games website

Functionality: Test whether the system completes all specified functionality outlined in the requirements phase.

Performance: Test the stability, scalability and responsiveness under varying workloads. E.g. with a website increasing the number of clients connecting to the website.

Dependability: Test whether the system is dependable. For example test the non-functional requirements such as Resilience, Robustness, Reliability. Measures are: downtime, time to start up

Does not fail during normal use: whether the system does not fail during normal use

C)

Cocomo2: automated historical and current data easy to update
FP: Subjective estimation of complexity => LOC
Why: Both can contradict making planning hard, COCOMO2 is not subjective whereas FP is, chance of differing cost/size causing waste of time or decision making.

d) Compare release and sprint planning. How do these two activities interact with each other?

(4 marks)

Release planning focuses on deliverable features: the smallest set of stories/cases that deliver immediate business value.

Sprint planning focuses on a subsection of the release plan which can be completed in an iteration. Each sprint will contribute to part of a release.

A sprint is a potential release, but most sprints are not releases.
Sprints and Releases complement each other well. Sprints are essentially a time frame to achieve team goals, whereas Releases are usually deliverables (the product itself or part of the product).

Another difference is the scale. Sprint planning is usually on a smaller scale because features from a release plan are chosen, which can be completed within a single iterative cycle. Release planning is on a larger scale because it can cover deliverables across multiple sprints.

QUESTION 3 **(Total: 16 marks)**

The following is a short description of a software development project.

The project is to implement a web-based dashboard of that will monitor the current state of different investment funds offered by a large investment firm (e.g. a company like AMP). The dashboard will allow a user to drill down into a fund and see its performance across its different investment categories, and then to drill down further into the actual investments in each category.

The dashboard is to be used by senior executives in the company to monitor how well each investment fund is performing. They will use this to help decide investment strategy that is to be implemented by the investment teams in the company.

When answering the questions below, document any assumptions that you make about the development team, organisation or stakeholders.

a) Identify a requirements modelling technique that would be appropriate for this project. Provide a justification as to why your chosen technique would be more appropriate than other techniques.

(4 marks)

Assumption: Users are available to answer developer questions.
Assumption: This is not a critical system that is required to be completed within a limited, urgent time frame.

The development team should consider using user stories.

User stories provide intention for each story (which use cases do not). This will help provide developers to understand the reason why roles want to perform certain actions as it is an unknown domain space.

User stories, in comparison to using cases, are simpler. Therefore roles can write stories themselves.

It appears there will only be one major role in interacting with this system. Use cases are often focusing on interactions with multiple actors in the system.

OR

Use Case:

Provides a description of the user interaction with the system. Can deduce processes that users complete, actors involved and the details of these processes and actors. Also good for documentation, more professional and useful for analysis. Can collect a high range of

requirements from users, processes, typical and abnormal scenarios and the division or duplication of processes (include/extends)

b) Describe two requirements elicitation techniques that would be appropriate for this project. Provide a justification as to why you have selected these techniques.
(8 marks)

Spec by example
Spec by example is a form of living documentation which outlines the specification and requirements of a system by giving example interactions and scenarios with the system.

Spec by example provides developers with an understanding of the system due to using natural language, especially when there is a large amount of domain space knowledge required.

Furthermore, it allows for higher user acceptance as due to the natural language, clients can understand requirements outlined (e.g. UML/use case diagrams can be complex so much so that users can not understand what is outlined)

Spec by example provides acceptance criteria and the basis of testing documents which is vital for many software processes (e.g. agile, V-model, TDD) which can save developers time and resources to create these artifacts.

Spec by example is living documentation and changes as requirements change or are clarified. This nicely fits in with many forms of software processes that embrace change. <<< grasping at straws here.

Ethnography
Ethnography is the developer becomes and experiences part of culture/society/group. Assuming the developer has had little experience or understanding of finance, money or trading this is a great elicitation technique.

- Understand the values of a groups
- Understand business processes
- Learn about domain space knowledge
- May not be a lot of in-depth documentation about business' culture, practices and processes.

Focus Groups
This helps when we haven't gotten much from the users or the customers and need to initially chart their direction and explore their thoughts.

Since the question explicitly mentions that the dashboard will be used by senior executives, having a senior-executive focus group will prove to be beneficial as they allow us to dig deeper into the thought process of a small and specialised group of users.

c) Describe one significant risk that would be a potential issue for the project. Describe how this risk could be monitored, managed and mitigated.

(4 marks)