

Question 1

- a. i. Spiral model development.
 - This methodology allow development teams to plan out development processes while mitigating risks that jeopardize critical operations, and are suitable to be used for complex and expensive projects.
 - This methodology allows system development to happen incrementally, therefore the project can adapt to changing requirements and allow clients to check on project status when a incremental phase is completed.
- ii. **Advantages:**
 - Confirming client expectations before actual development of a feature.
 - Obtain feedback on unforeseen system exceptions or bugs.
 - Allow clients to use a placeholder application during the development of a newer version
- Disadvantages:**
 - Spend time building prototypes and not the final product feature.
 - Constantly require client feedback.
 - Hold back other important features.
- b. (Not explaining the entire model, only design activities)
 - Architectural design is the design of system architecture, which includes the hardware or topologies required to run the system.
 - Interface design is the design of system software, including user interfaces and programming language
 - Component design is the design of the system functionalities, such as code libraries required by the system, object classes, and functions.
 - Database design is the design of the database storage used by the system, including the data structure in system database and entity relationships.
- c.
 - Use caching techniques. For example, a software can make use of cookies or local storage of a browser to store temporary user data, which enable users to resume unfinished operations after they left the system.
 - Provide preset values for inputs that have limited categories. For example, dropdown inputs should be frequently used for inputs that have limited values so that users do not provide invalid inputs.
- d. i. Software developers should protect user privacy when they are working with confidential user data.
 - ii. Software developers should possess enough technical knowledge to be able to develop reliable and working software applications.
 - iii. Software developers should not abuse their knowledge to use their computers to work on anything else that falls outside of a software development's job scope.
- e. Institute of Electrical and Electronics Engineers (IEEE)
or ACM

Question 2

a) **Ceremonies:**

Daily Scrum Meeting

Sprint Planning

Sprint Retrospective

Artifacts:

Product backlog

Sprint backlog

Burndown charts

- b)
- As a customer, I want to place order for café items online so that I can let the café deliver my items to me without going to their shop.
 - As a customer, I want to view the café's outlet locations so that I know which outlet location is nearer to go to from my location.
 - As a customer, I want to save my delivery location so that I don't need to repeatedly input my location everytime I want to place an order.
 - As a customer, I want to view the café's menu online so that I know what items are there in the store that I can order.

c) S - Specific

M - Measurable

A - Achievable

R - Relevant

T - Time boxed

d) i. Screens complexity = Medium = 2

Report complexity = Difficult = 8

Application point count = screens(4 x 2) + reports(4 x 8) = 40

ii. New application point count = 40 * 85% = 34

iii. Effort to develop = normal = 34/13 = 2.615 effort/person

Question 3

- a.
 - i) Constructing class names and also file name, folder name, variable name, function name etc
 - ii) Using multiple words for variable names ✓
 - iii) Refactoring ambiguous variable names to something meaningful ✓

b. - Method implementations can be abstracted away and developers can write declarative code. ✓

- Prevent users that do not have the appropriate access levels from accessing protected members in the class. ✓

c. - Difficulty to identify the necessary test cases to fully test a particular feature. Some bugs may go undiscovered during validation testing and appear in the client side if test cases are constructed poorly.

- Lack of time to execute all of the prepared test cases. System features that are too large may use a lot of time for validating and hold up features that have higher priority.

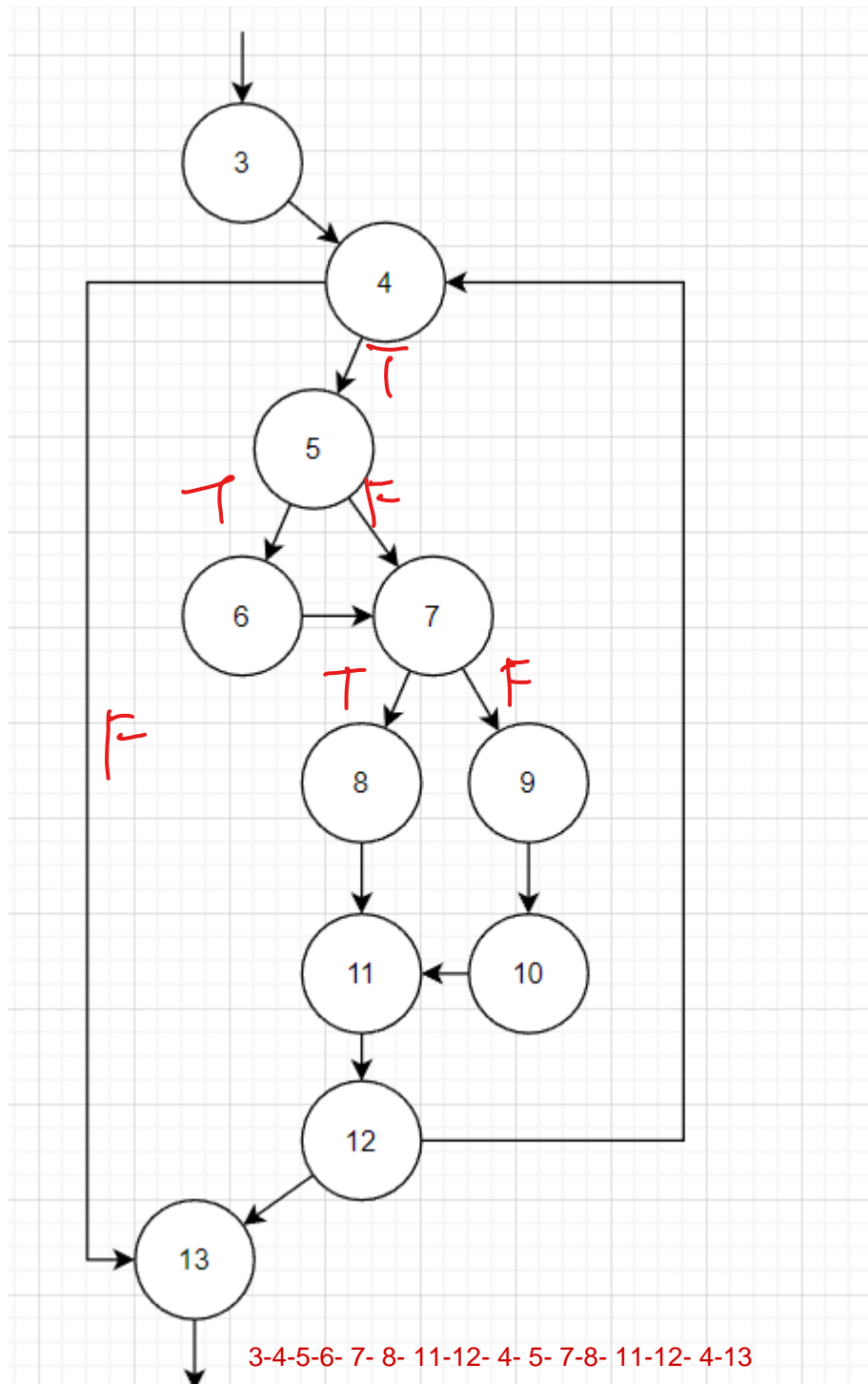
- Missing critical defects, Software crashes ✓

- Software is not in a stable state

- Could bring down the system or crash

- Tester too occupied with fundamental issues because of lack of time
- Not able to focus on critical functions.

- Too much time taken for development
- Planned test cases not executed and number of bugs not captured



d.

e. 3-4-5-6-7-8-11-12-4-5-7-9-10-11-12-13 5000 is divisible by 4

f. Fault avoidance: The capability of software programs to operate normally without throwing errors.

Fault detection: The ability of software program to detect unforeseen software errors. ✓

Fault tolerance: The ability of software program to handle unforeseen exceptions with grace during application runtime. ✓

The system is developed in such a way that human error is avoided and thus system faults are minimized. The development process is organized so that faults in the system are detected and repaired before delivery to the customer.