CS 3321/INFO 3307 Midterm Exam Key

Fall 2021

1.) *Based on the definitions and discussion, at the beginning of the semester, for the concepts: programming, software development, and software engineering, briefly explain the relationships between these three concepts.*

- **Programming** - the work related to implementation or coding of source code
- **Software Development** - the larger process which, apart from programming, includes working with requirements, design, test, etc.
- **Software Engineering** - combines engineering techniques with software development practices. Relies more heavily upon science rather than upon craft.

These three concepts are related as follows. Programming is the means by which the end goal of software development is realized. In other words, programming is integral to software development, but software development is not wholly comprised of programming. Similarly, Software development provides the practices by which software engineering is done, but software engineering brings the science and engineering discipline to software development in order to improve the reliability and repeatability of the process.

2.) *Briefly explain how teamwork affects the dynamics of software engineering.*

Teamwork affect software engineering dynamics, as there are multiple people developing the software. Thus, artifacts such as code, requirements, tests, and designs must be understandable and comprehensible between all members of the team. This requires that each team member have higher levels of competency and the ability to create high quality artifacts.

3.) *Which of the following are not **practices** which may be adopted by a development team.*

[[Autograded]]

4.) *Briefly describe the layered architecture of essence and its components.*

The Essence architecture is made up of 4 layers: The Essence Language, The Essence Kernel, the Essentialized Practices and The Essentialized Methods. The Essence Kernel uses the Essence Language. The Essentialized Practices use the Essence Kernel. The Essentialized Methods use the Essentialized Practices.

5.) *Consider a university student registration system. With this system in mind, for each of the following groups/individuals identify if they are a stakeholder, and if so which type of stakeholder they may be (internal or external).*

[[Autograded]]

6.) *Briefly, explain the relationship between **alphas** and **alpha states**.*

- **Alphas** - An essential element of the development endeavor that is relevant to an assessment of the progress and health of the endeavor
- **Alpha States** - Provide a means to determine how far an alpha has progressed through a lifecycle.

Thus, alpha states provide a means by which we can measure the progress of an alpha. This relationship directly provides the assessment of the progress and health of the endeavor.

7.) *Which of the following is **not** a key characteristic of Essence as it relates to software engineering theory?*

[[Autograded]]

8.) *When considering a general theory of software engineering, why is the attribute of **predictability** necessary?*

Predictability in a general theory of software engineering is necessary as such a theory allows us to answer questions about what will happen next. That is, with a predictive theory we would be able to understand what the outcome of different decisions would be and would be better capable of optimizing our process.

9.) *Briefly describe the **Demming cycle**.*

The demming cycle is a method to adapt and improve a given process. This approach is a cycle of four phases: Plan, Do, Check, and Adapt.

- **Plan** - Essentially, we are attempting to determine the current and next states and how we can achieve the next state.
- **Do** - We actually work to achieve the next state and remove any obstacles as the occur.
- **Check** - We track the work (progress), and check that the work was actually done.
- **Adapt** - At this point we reflect on what happened, look for more suitable ways to work, improve the quality of the work, and attempt to reduce wasted effort.

10.) *Which of the following were methods suggested as a means by which we could visualize the progress and health of a project?*

Freebie as long as they give any three examples. Some example answers may include:

- requirements met
- artifacts developed
- defects found
- defects fixed

Note that the answers cannot be the alphas themselves (i.e., Requirements, Work, Software System, Teams), but should rather be measurable entities based around work products or properties of alphas.

11.) *Which of the following were methods suggested as a means by which we could visualize the progress and health of a project?*

[[Autograded]]

12.) *When tracking the progress/health of an endeavor, we will often notice waves of changes in alpha states. Briefly explain why this phenomenon occurs.*

Alphas tend to change state in waves due to the fact that teams will often set multiple objectives during a period of work. This fact is compounded with the fact that the alphas are interelated. Thus, as a team works to achieve their goals they are working on artifacts and items that will affect multiple alphas simultaneously, and when the next objective check occurs the team will see that multiple alphas have changed state during that period of work.

13.) *Which of the following are known to be benefits of the Scrum practice?*

[[Autograded]]

14.) In scrum, a Product Backlog Item must meet a specific set of criteria in order for the team to call it complete. List this "Definition of Done" criteria.

The criteria for Definition of Done are:

- sufficiently tested
- accepted by the Product Owner
- source code checked in
- associated documentation updated

15.) When considering the Scrum practice, which activity relates directly to the Demming cycle?

[[Review]] accept any of the following: Sprint Planning, Sprint, Sprint Review, Sprint Retrospective

16.) When considering user stories, what is the purpose of the **Card**, **Conversation**, and **Confirmation** within the process of creating the user story?

- **Card**: captures the requirement/story using a succinct headline description which details the user's requirements
- **Confirmation**: provides a bulleted list of acceptance criteria (on the back of the card) used to validate and verify the story has been achieved
- **Conversation**: a discussion between developers and stakeholders to converge on the best solution for the system (i.e., to define the story and the acceptance criteria)

17.) Briefly describe the purpose of **user stories** as they relate to essence and the software engineering endeavor.

User stories provide the following components:

- **User Story Alpha** - comprises the requirements and is describe by a Story Card work product
- **Story Card Work Product** - describes the user story alpha, and is verified by a Test Case work product
- **Test Case Work Product** - Verifies the Story Card work product

Additionally, User Stories provide three key activities all of which fall within the "Solution" domain activity Spaces

- Within the "Understand the Requirements" activity space the following actities are provided by User Stories:
  - Find User Stories
  - Prepare a User Story

- Within the "Test the System" activity space the following activity is provided by User Stories:
  - Accept a User Story

18.) Describe the both the differences and the relationships between **basic** and **alternate flows** within a Use Case.

- The Basic Flow: provides the straight forward normal use of the use case
- The Alternate Flows: provide variations of the basic flow describing ways to deal with more specific cases (i.e., enhancement, special caes, errors)

The relationship between these two is straight forward.

19.) Considering that we proposed two different approaches, **User Stories** and **Use Cases**, which solve similar problems. Please provide insight into why one might wish to use one over another. If possible use an example or scenario to clarify your thoughts.

Freebie, any reasonable answer will do

20.) How does the concept of **data hiding**, change in the context of microservices?

Microservices fully embrace the notion of components. Components provide data hiding by containing any data within themselves and away from other components, while providing only a specific interface by which they can be accessed. Microservices take this to a higher extreme by incorporating the component approach all the way from design to code to deployment. Furthermore, microservices provide for extremely loose coupling, which allows for each microservice to evolve separately.

21.) Consider the two opposing architectural ideas: **a monolithic architecture** and **a microservice architecture.** When considering these two, what purpose does the **microservice architecture** serve and what challenges does it present as opposed to a **monolithic architecture**?

Freebie, any reasonable answer will do