Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

CS 709 Advanced Topics in Computer Science [Software Engineering]

Set of questions for the midterm test of November 28th, 2017

Version 2 – November 21, 2017 [final]

A. Study Material

- Chapter 2, Software Processes [Sommerville 2015] – see Lecture 04
- Chapter 3, Agile Software Development [Sommerville 2015] – Lectures 05 and 06
- Chapter 4, Requirements Engineering [Sommerville 2015] – Lecture 07
- Chapter 5, System Modeling [Sommerville 2015] – Lectures 08 and 09
- Chapter 6, Software Architecture [Sommerville 2015] – Lecture 10 and 12
- Paper presentation by Biplav – Lecture 16
- Paper presentation by Syed – Lecture 16
- Paper presentation by Hannah – Lecture 17
- Paper presentation by Jalal – Lecture 17
- Paper presentation by Anudeep – Lecture 18
- Paper presentation by Vinh – Lecture 18
- Paper by Sarah Beecham et al, 2014 – Lecture 22
- Paper by Otero and Peter, 2015 – Lecture TBD
- Additional material that might be added by November 21, 2017

B. Questions

From Chapter 2, Software Processes [Sommerville 2015]

1. Explain what a *software process* is and briefly describe the typical phases of a software process (specification, design, etc.).
2. Explain the differences between *plan-driven* and *agile* software processes.

3. Describe the *waterfall* software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

4. Describe the *incremental development* software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

5. Describe the *integration and configuration* (*reuse-oriented)* software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

6. Describe the *incremental delivery* software engineering process model. Also, indicate its advantages, disadvantages, and applicability.

7. Explain what is meant by *software specification* and describe the activities involved in the requirements engineering process.

8. Explain what is meant by *software design* and describe the activities involved in the software design.

9. Describe the *testing phases* in a plan-driven software process.

10. Explain what is meant by *software evolution* and describe the activities involved in the software/system evolution.

11. Explain what is meant by *software prototyping* and indicate its benefits. Also, explain why *throw-away prototypes* should be discarded.

From Chapter 3, Agile Software Development [Sommerville 2015]

12. Describe the *philosophy* behind agile methods as reflected in the *agile manifesto*.

13. Describe the 5 main *principles of agile methods.*

14. Give at least 5 examples of *questions* one should ask when deciding on the *balance between a plan-based and an agile approach*.

15. Describe 5 *principles or practices specific to extreme programming* (out of 10 indicated in the book).

16. What are several *problems/issues* pertaining to applying *agile methods*? (Indicate at least 5.)

17. What is meant by *refactoring*? Give at least 3 concrete examples of refactoring.

18. Describe the *Scrum* approach. Indicate what you think are its main advantages and disadvantages.

From Chapter 4, Requirements Engineering [Sommerville 2015]

19. Explain what is meant by a *user requirement* and by a *system requirement*. Give a concrete example for each.

20. Explain what is meant by a *functional requirement* and by a *non-functional requirement*. Consider your CS 709 project and give 3 examples of each. Briefly indicate what your project is about.

21. Give at least 5 examples of *different metrics* that could be used for specifying *non-functional requirements* (e.g., Gbytes for memory size).

22. Describe and compare 3 *notations* for writing a software/system requirements specification document.

23. Explain what is meant by a *software requirements document* (SRS) and outline its structure.

From Chapter 5, System Modeling [Sommerville 2015]

24. Briefly indicate and describe 4 types of *UML diagrams.*

25. Explain what a *context model* is and give an example of such model.

26. Explain what a *use case* is and give an example of *a use case diagram* with at least 6 non-trivial use cases.

27. Explain what a *sequence diagram* is and give an example of such diagram that involves at least an actor and two objects.

28. Briefly describe the three main types of *relationships between classes*. Give an example of a *class diagram* with at least 7 classes in which all three types of relationships are represented.

29. Briefly explain what a *state chart (state diagram)* is, indicate its main notation elements, and give an example of a state chart with at least 6 states.

30. Explain what *model-driven engineering* is and discuss its benefits and limitations.

From Chapter 6, Software Architecture [Sommerville 2015]

31. Indicate and comment on 5 *architectural decisions* that might need to be taken when developing a larger software product.

32. Describe the *4+1 views architecture*.

33. Describe the *model-view-controller architecture pattern*, and indicate its advantages, disadvantages, and applicability.

34. Describe the *layered architecture pattern*, and indicate its advantages, disadvantages, and applicability.

35. Describe the *repository architecture pattern*, and indicate its advantages, disadvantages, and applicability.

36. Describe the *client-server architecture pattern*, and indicate its advantages, disadvantages, and applicability.

37. Describe the *pipe-and-filter architecture pattern*, and indicate its advantages, disadvantages, and applicability.

From paper by Sarah Beecham *et al* [2014]

38. Indicate at least 5 *challenges* in Global Software Engineering (GSE). What do you think are the *reasons* practitioners do not look at related GSE research papers for finding solutions/guidance to address these challenges?

39. Describe the *Practice-Research paradox* (or divide) and discuss the *reasons* for its existence.

40. What do Beecham *et al* [2014] suggests to do in order *to make research accessible* and thus address the practice-research divide?

From paper by Otero and Peter [2015]

41. Explain what is *big data* and what is *big data software* as defined by Otero and Peter [2015].

42. Describe the three levels of *actionable intelligence* and briefly explain the *single hop* and the *multi-hop intelligence production processes* presented by Otero and Peter [2015].

43. Describe the *construction problem* identified by Otero and Peter [2015] in relation with big data analytics software engineering.

From paper presented by Biplav – Lecture 16

44. Describe the *layered approach* for Android application development proposed in the paper by Holla and Katti (2012). Indicate the *layers* of the architecture proposed, and describe each of them.

45. Present a short overview of the *Android platform* and describe at least 3 *services* that are underlying to all Android applications – see [Holla and Katti, 2012].

46. Describe three types of *security measures* for the Android platform – see [Holla and Katti, 2012].

From paper presented by Syed – Lecture 16

47. Describe in some details the *two goals* of the paper by Memon et al (2017).

48. In their paper Memon et al (2017) indicate that "we modeled the relationships between our test targets and developers, code under test, and code-change and test execution frequencies." Describe 5 of their related *findings*.

49. Chose 3 of the 5 *hypotheses* considered in [Memon et al,2017] and briefly describe them (just mention what the hypotheses were, not how they were tested). For <u>one</u> of the hypotheses chosen, describe the related results obtained and how these results were interpreted by the authors.

From paper presented by Anudeep – Lecture 17

50. Consider the paper [Muccini et al, 2012]. Describe what is meant by an application in *mobile computing* and what is meant by an application in *context-aware computing*. Also, explain what the authors consider to be, from a testing perspective, an App4Mobile application and, respectively, a MobileApplication app.

51. Consider the paper [Muccini et al, 2012]. Describe in detail 2 *peculiarities* of Apps4Mobile applications and their implications on software testing.

52. Describe 4 *challenges and research directions* on testing mobile applications. You may choose any challenges, regardless of their type, presented in Section III of the [Muccini et al, 2012] paper. Note that you do not have to describe their *Potentials and Automation*.

From the paper presented by Jalal – Lecture 17

53. Define what a *Software Product Line* (SPL) is and explain why an increasing number of organizations are building their products as product lines. Refer to paper [Bergey et al, 2010].

54. In the context of Software Product Lines (SPL) explain what is meant by *core assets* and give examples of such assets. Refer to paper [Bergey et al, 2010].

From paper presented by Hannah – Lecture 18

55. Explain why the concept of *code ownership* is important and provide a definition for it.

56. What *problem* related to *code ownership* is addressed in [Muller et al, 2015]? Briefly (in 3 to 5 lines) explain the *solution* proposed by the authors of this paper.

From paper presented by Vinh – Lecture 18

57. Consider the paper [Killalea, 2016]. Describe the following *Dividends* of implementing microservices: permissionless innovation, enable failure, disrupt trust.

58. Consider the paper [Killalea, 2016]. Describe the following *Dividends* of implementing microservices: end centralized metadata, concentrate the pain, test differently.