**COIS-ADMN 3850H - 2018 FA – QUESTION SET #8**

Submitted by: \_\_\_\_\_\_\_\_\_\_Konrad Bartlett \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ # CORRECT = \_\_\_\_\_\_\_\_

Evaluated by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ # **FOR REVIEW** = \_\_\_\_\_\_\_\_

**Instructions:**

1. first, type your answers and print them out.
2. submit these completed pages at the beginning of the next class for evaluation.

Text: **MANAGING AGILE PROJECTS** | Edited by Kevin Aguanno

a) Re **Chapter 16** – **“Agile Meetings”** [pages 271-281]:

1. Describe the nature of an Agile [Scrum] Meeting … who attends and what happens?

The scrum master as well as the members of the team. The scrum master asks each team member 3 questions:

* Relative to the backlog, what have you completed since the last meeting?
* What obstacles got in the way of your completing this work?
* Relative to the backlog, what specific things do you plan to accomplish between now and the next meeting?

1. What are the three questions the Scrum Master asks each team member?

* Relative to the backlog, what have you completed since the last meeting?
* What obstacles got in the way of your completing this work?
* Relative to the backlog, what specific things do you plan to accomplish between now and the next meeting?

1. Describe five goals that an Agile [Scrum] Meeting is designed to achieve: i) Focus on backlog items

ii) Removing items from the backlog

iii) Communicate priorities of backlog items

iv) Inform members of inform progress of obstacles, and solve them

v) Track the progress towards delivering backlog functionality

1. Describe five advantages of having frequent, short meetings:

i) Keeps everyone on track

ii) I don’t know it doesn’t really say

iii)

iv)

v)

1. What are the two “warnings” detailed on page 274?

i) Stick to the meeting plan

ii) Only active participants are involved

Re **Chapter 17** –"**There Are No Silver Bullets**":

1. On pages 285-293, Kevin Aguanno details five “weaknesses” of agile development methods. What are they?

i) Teams tend to optimize for their specific project, affecting the organization

ii) May not have adequately structured design and quality review processes

iii) Safety concerns of half developed software being use in critical systems

iv) Some projects are more scalable than other

v) Do not run with a hammer thinking that everything looks like a nail

Re **Chapter 18**, "**Extreme Methods Lead To Extreme Quality Problems**" [pages 299-322]:

1. How would you describe Gerold Keefer's "Bias" about major aspects of Extreme Programming? [see page 303]

He’s biased towards engineering techniques, says that extreme programming has:

* A reliance on verbal communication
* Intuition-based decision making
* High dependency on individual skills
* Insufficient planning
* And No sound approach to system verification and validation

1. What are the benefits that he acknowledges? [see page 304]

* Very useful for testing
* Visualization aspects during meetings are good

1. What are the "Dubious Values and Practices" that Keefer cites? [beginning on page 306]iv)What are his recommendations with respect to "Large Projects"?

[beginning on page 319]

* 80% benefit with 20% work
* Change is not good in an organization
* Refactoring and reworking programs can be not good
* Simplicity means easy but also lightweight
* Pair programming sucks (trust me)
* On-site customers hardly every exist
* Having no documentation is cumbersome
* Collective code ownership
* 40-hours weeks don’t get followed
* “everything is in the code”
* Test cases are requirements
* Who system is oriented to focus on people
* Doesn’t recommend specializing people in certain skills and areas

v) What are Keefer's "Conclusions" [see pages 321-322]

Even if time is by far the most dominant factor for your project, and the environment is highly volatile, (a case where you’d employ agile), you should still plan thoroughly.