

W08 Prepare: Reading and Quiz

Due Jun 8 at 11:59pm Points 20 Questions 10 Time Limit None Allowed Attempts Unlimited

Instructions

- **Task:** Complete the reading and quiz
- **Purpose:** Apply CMM principles to an existing software project
- **Time:** 60 minutes

Notes

- The Prepare activity for this week will be one chapter from the Mythical Man-Month and one article.
- [To see more about this item, please click here.](https://content.byui.edu/file/a2001083-4378-4d58-a641-903d8b5bbaa6/1/Prepare/432.08.Prepare.html) [.\(https://content.byui.edu/file/a2001083-4378-4d58-a641-903d8b5bbaa6/1/Prepare/432.08.Prepare.html\)](https://content.byui.edu/file/a2001083-4378-4d58-a641-903d8b5bbaa6/1/Prepare/432.08.Prepare.html)
- Please take the quiz after doing the reading.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 3	2 minutes	18.33 out of 20
LATEST	Attempt 3	2 minutes	18.33 out of 20
	Attempt 2	6 minutes	15.33 out of 20
	Attempt 1	less than 1 minute	0.33 out of 20

🔒 Correct answers are hidden.

Score for this attempt: **18.33** out of 20

Submitted Jun 8 at 11:37pm

This attempt took 2 minutes.

Question 1

1 / 1 pts

Be very careful with the architecture of the software system. If more than one design philosophy is used, it is likely that the boundary between the sub-systems will be a problem. You can avoid this with careful interface definitions.

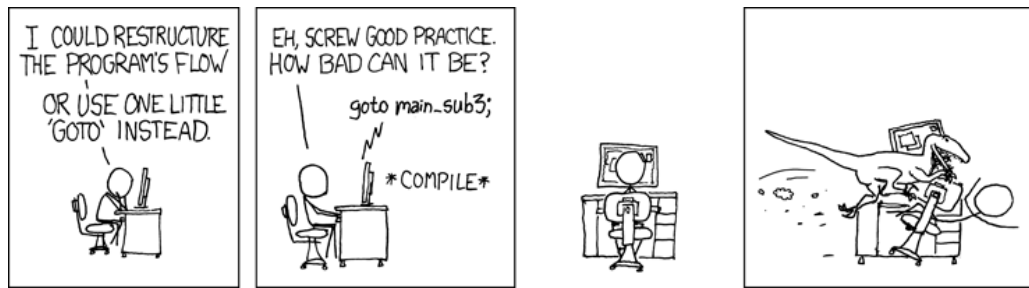
Bug-proofing the definition

Start with big task the software system is designed to do. Break this into smaller tasks. Make sure they are well defined and understood. Think cohesion (the task does one thing and one thing only) and coupling (the information interchange between the tasks are as simple as possible). Next, take each of these tasks and break them down into smaller sub-tasks. Continue this process until the task is readily understood. Then, for these small tasks, create tests that will verify them.

Top-down design

Approach the spec. with the same degree of scrutiny with regards to Quality Assurance as you do with the final product. Look for things that are missing from the spec. as well as things that are wrong.

Testing the specification



Structured programming

Answer 1:

Bug-proofing the definition

Answer 2:

Top-down design

Answer 3:

Testing the specification

Answer 4:

Structured programming

Partial

Question 2

0.33 / 2 pts

Brooks describes several component and system debugging techniques. Some are so dated that they have absolutely no relevance in today's environment (such as memory dumps and the such), but some have become so ingrained in our experience that we take them for granted. Please match the technique with the description.

Similar to the Agile notion of a Sprint: a potentially shippable and properly tested product.

Quantize updates

Bottom-up testing. Start by testing individual modules. When they are working, integrate them with other tested modules and test them together. Only as the last step is the entire system tested together.

Use debugged components

Use a tool such as GDB (on Linux), Visual Studio (on Windows computers), or X Code (on the Macintosh) to set breakpoints, view the value of variables, and set conditions.

Control changes

This is similar to the concept of Software Configuration Management (SCM) that we discuss in CS 416. The idea is that

Interactive debugging

modifications to the system are recorded and go through a formal process.

Use debug components (like asserts and debug-only tests) to exercise the code. This is similar to XP's notion of testing: "Unit tests are written by the programmers, in order to prove that programs work the way they are expected to."

Add one component at a time ▼

This is used in modern development environments. Multiple developers are able to check code into the larger repository. The repository is then locked periodically (say once a day or once a week) and the code is thoroughly tested through automation. If the automation passes, then the build is released for the other members of the development team. This ensures that a baseline of quality exists.

Build plenty of scaffolding ▼

Question 3

3 / 3 pts

The Key Practices document is huge and contains 479 pages. Somewhere in here you need to find the information you need! At the bottom right-hand corner of each page, there are page numbers. The "Level 3 Key Practices" section has 111 pages beginning at "L3-1" and ending at "L3-111". Please find the page numbers for the following sections.

Organization Process Focus:

L3-1

Organization Process Definition:

L3-11

Training Program:

L3-25

Integrated Software Management

L3-37

Software Product Engineering

L3-59

Intergroup Coordination

L3-85

Peer Reviews

L3-97

Answer 1:

L3-1

Answer 2:

L3-11

Answer 3:

L3-25

Answer 4:

L3-37

Answer 5:

L3-59

Answer 6:

L3-85

Answer 7:

L3-97

Hint: the page number of "Training Program" is "L3-25"

Question 4

2 / 2 pts

Please find the "**Organization Process Focus**" section of Level 3 key process areas. Locate the "**Activities Performed**" sub-section. Please type verbatim **Activity 2**. This is one sentence that has only one period.

The organization develops and maintains a plan for its software process development and improvement activities.

Top of page L3-7.

Question 5**2 / 2 pts**

Please find the "**Organization Process Definition**" section of Level 3 key process areas. Locate the "**Goals**" sub-section. Please type verbatim **Goal 1**. This is one sentence that has only one period.

A standard software process for the organization is developed and maintained.

The top of page L3-12.

Question 6**2 / 2 pts**

Please find the "**Training Activity**" section of Level 3 key process areas. Locate the "**Activities Performed**" sub-section. Please type verbatim **Activity 1**. This is one sentence that has only one period.

Each software project develops and maintains a training plan that specifies its training needs.

Please see page L3-29.

Question 7**2 / 2 pts**

Please find the "**Integrated Software Management**" section of Level 3 key process areas. Locate the "**Activities Performed**" sub-section. Please type verbatim **Activity 10**. This is one sentence that has only one period, three commas, and one apostrophe.

The project's software risks are identified, assessed, documented, and managed according to a documented procedure.

Please see page L3-52.

Question 8**2 / 2 pts**

Please find the "**Software Product Engineering**" section of Level 3 key process areas. Locate the "**Goals**" sub-section. Please type verbatim **Goal 1**. This is one sentence that has only one period and two commas.

The software engineering tasks are defined, integrated, and consistently performed to produce the software.

Please see page L3-60.

Question 9**2 / 2 pts**

Please find the "**Intergroup Coordination**" section of Level 3 key process areas. Locate the "**Activities Performed**" sub-section. Please type verbatim **Activity 7**. This is one sentence that has only one period.

Representatives of the project engineering groups conduct periodic technical reviews and interchanges.

Please see page L3-92.

Question 10**2 / 2 pts**

Please find the "**Peer Reviews**" section of Level 3 key process areas. Please type verbatim the **first sentence** beginning with "The purpose."

The purpose of Peer Reviews is to remove defects from the software work products early and efficiently.

Please see page L3-97.

Quiz Score: 18.33 out of 20