1. What do the terms verification and validation mean?

*Explain the terms verification and validation.*

Course book p. 41-42.

- Validation – “are we building the right system” – does the system really satisfy the users’ needs?
- Verification – “are we building the system right” – do the components satisfy their specification?

2. Explain the objectives of component testing.

*What are the objectives of component testing?*

Course book p. 46 – 49. The most important thing to verify the functionality for correctness and completeness. It is also important to verify the robustness of the component in case of wrong usage or special situations (borderline cases) not mentioned in the specification. Also, testing for quality is important for critical components that affect for example the integrity or efficiency of the system. The maintainability of components can be tested with static analysis methods.

3. What could be the consequences of omitting component tests and running all tests only after integration?

*What could be the consequences of omitting component tests and running all tests only after integration?*

Course book p. 55. Most of the faults in the software lie within the components. It requires a lot of effort to locate the faults based on the failed high-level test cases executed in the integration test. Also, since the focus is on sub-system level use cases, the integration level use cases do not exercise the components’ functionality with the required level of intensity and coverage, which leaves many faults uncovered.

4. Describe four typical forms of acceptance tests.

*Describe four typical forms of acceptance tests.*


5. Explain the role of testing in Continuous Integration according to Wikipedia.

*What is the role of testing in Continuous Integration?*

Explain the role of testing in Continuous Integration according to Wikipedia:

http://en.wikipedia.org/wiki/Continuous_integration
Testing has a fundamental role in CI. Only code that passes unit tests should be committed to
the common code base, from where an integrated version of the software system is build daily.
The building of the software includes running all the integration tests and collecting various
metrics for quality assurance purposes. A key issue is the automation of all the tests performed
as part of CI. There are variations on when tests are actually executed (ie. Upon every commit to
the code-line or as per an agreed fixed schedule like a “nightly build”).

6. Minkälaisia ohjeita I. M. Testy antaa testitapausten valinnasta regressiotestaukseen:

What advice does I.M. Testy give on selecting test cases for regression test:
Test the most important and most often used functionality
Retest for all functional defects that have been fixed at some point (defects tend to regress).
Organize test cases by functionality into collections and prioritize the test cases. Run the high
priority test cases in each collection as part of regression test.
Map functionality areas into code structure and run the high priority test cases for the
functionalities of which implementation has been changed in the code base (or are otherwise
affected based on the dependencies in the implementation).