Chapter 1 & 2 Questions

Dr. Paul West

Department of Computer Science
College of Charleston

February 6, 2014

- 1.1 Philip has studied “just-in-time” industrial production methods and is convinced that they should be applied to every aspect of software development. He argues that test case design should be performed just before the first opportunity to execute the new designed test cases, never earlier. What positive and negative consequences do you foresee for the just-in-time test case design approach?

- 1.1 Philip has studied “just-in-time” industrial production methods and is convinced that they should be applied to every aspect of software development. He argues that test case design should be performed just before the first opportunity to execute the new designed test cases, never earlier. What positive and negative consequences do you foresee for the just-in-time test case design approach?
- Positive: Waiting for later to do test cases can lead to less rework of test cases.
- Negative: Reduced accuracy since developers are more likely to look at what is in front of them instead of the big picture.

- 1.2 A newly hired project manager at Chipmunk questions why the quality manager is involved in the feasibility study phase of the project, rather than joining the team only when the project has been approved, as a the new manager's previous company. What argument(s) might the quality manager offer in favor of her involvement in the feasibility study?

- 1.2 A newly hired project manager at Chipmunk questions why the quality manager is involved in the feasibility study phase of the project, rather than joining the team only when the project has been approved, as a the new manager’s previous company. What argument(s) might the quality manager offer in favor of her involvement in the feasibility study?
- Risk Management: Quality manager will check how feasible the tests are to ensure the proper quality.

- 1.3 Chipmunk procedures call for peer review not only of each source code module, but also of test cases and scaffolding for testing that module. Anita argues that inspecting test suites is a waster of time; any time spent on inspecting a test case designed to detect a particular class of fault could more effectively be spent inspecting the source code to detect that class of fault. Anita's project manager, argues that inspecting test cases and scaffolding can be cost-effective when considered over the whole lifetime of a software product. What arguments(s) might Anita's manager offer in favor of this conclusion?

- 1.3 Chipmunk procedures call for peer review not only of each source code module, but also of test cases and scaffolding for testing that module. Anita argues that inspecting test suites is a waster of time; any time spent on inspecting a test case designed to detect a particular class of fault could more effectively be spent inspecting the source code to detect that class of fault. Anita's project manager, argues that inspecting test cases and scaffolding can be cost-effective when considered over the whole lifetime of a software product. What arguments(s) might Anita's manager offer in favor of this conclusion? Regression tests

- 1.4 The spiral model of software development prescribes sequencing incremental prototype phases for risk reduction, beginning with the most important project risks. Architectural design for testability involves, in addition to defining testable interface specifications for each major module, establishing a build order that support through testing after each stage of construction. How might spiral development and design for test be complementary or in conflict?

- 1.4 The spiral model of software development prescribes sequencing incremental prototype phases for risk reduction, beginning with the most important project risks. Architectural design for testability involves, in addition to defining testable interface specifications for each major module, establishing a build order that support through testing after each stage of construction. How might spiral development and design for test be complementary or in conflict?
- Complementary: Both start with highest risks.
- Conflict: Highest project risk might not be highest test risk.

- 1.5 You manage an online service that sells downloadable video recordings of classic movies. A typical download takes one hour, and an interrupted download must be restarted from the beginning. The number of customers engaged in a download at any given time ranges from about 10 to about 150 during peak hours. On average, your system goes down (dropping all connections) about two times per week, for an average of three minutes each time. If you can double availability or double mean time between failures, but not both, which will you choose? Why?

- 1.5 You manage an online service that sells downloadable video recordings of classic movies. A typical download takes one hour, and an interrupted download must be restarted from the beginning. The number of customers engaged in a download at any given time ranges from about 10 to about 150 during peak hours. On average, your system goes down (dropping all connections) about two times per week, for an average of three minutes each time. If you can double availability or double mean time between failures, but not both, which will you choose? Why?
- MTBF: A started download is less likely to be interrupted.

- 1.6 Having no a priori operational profile for reliability measurement, Chipmunk will depend on alpha and beta testing to assess the readiness of its online purchase functionality for public release. Beta testing will be carried out in retail outlets, by retail store personnel, and then by customers with retail store personnel looking on. How might this beta testing still be misleading with respect to reliability of the software as it will be used at home and work by actual customers? What might Chipmunk do to ameliorate potential problems from this reliability misestimation?

- 1.6 Having no a priori operational profile for reliability measurement, Chipmunk will depend on alpha and beta testing to assess the readiness of its online purchase functionality for public release. Beta testing will be carried out in retail outlets, by retail store personnel, and then by customers with retail store personnel looking on. How might this beta testing still be misleading with respect to reliability of the software as it will be used at home and work by actual customers? What might Chipmunk do to ameliorate potential problems from this reliability misestimation?
    - Alpha and Beta are going to use users which implies validation not verification. Hence, people may think the software is ready to do purchases, which may lead to disastrous results.
    - Chipmunk could could dummy credit cards, or have a sale representative check each transaction.

- The junior test designers of Chipmunk Computers are annoyed by the procedures for storing test cases together with scaffolding, test results, and related documentation. They blame the extra effort needed to produce and store such data for delays in test design and execution. They argue for reducing the data to test documentation and limiting test results to the information needed for generating oracle. What arguments(s) might the quality manger use to convince the junior test designer of the usefulness of storing all this information?

- The junior test designers of Chipmunk Computers are annoyed by the procedures for storing test cases together with scaffolding, test results, and related documentation. They blame the extra effort needed to produce and store such data for delays in test design and execution. They argue for reducing the data to test documentation and limiting test results to the information needed for generating oracle. What arguments(s) might the quality manger use to convince the junior test designer of the usefulness of storing all this information?
- Different oracles could test different paths of the system.
- Edge cases
- Feedback: ensure current process prevent past mistakes.

- 2.1 The Chipmunk marketing division is worried about the start-up time of the new version of the RodentOS operating system (an (imaginary) operating system of Chipmunk). The marketing division representative suggests a software requirement stating that the start-up time shall not be annoying to users.
  Explain why this simple requirement is not verifiable and try to reformulate the requirement to make it verifiable.

- 2.1 The Chipmunk marketing division is worried about the start-up time of the new version of the RodentOS operating system (an (imaginary) operating system of Chipmunk).
  The marketing division representative suggests a software requirement stating that the start-up time shall not be annoying to users.
  Explain why this simple requirement is not verifiable and try to reformulate the requirement to make it verifiable.
- “annoying” is subjective, we need a quantitative requirement (e.g. 20 seconds).

- 2.3 A calendar program should provide timely reminders; for example, it should remind the user of an upcoming event early enough for the user to take action, but not too early. Unfortunately, “early enough” and “too early” are qualities that can only be validated with actual users. How might you derive verifiable dependability properties from the timeliness requirement?

- 2.3 A calendar program should provide timely reminders; for example, it should remind the user of an upcoming event early enough for the user to take action, but not too early. Unfortunately, “early enough” and “too early” are qualities that can only be validated with actual users. How might you derive verifiable dependability properties from the timeliness requirement?
- Solicit opinions for customer.
- Use reviews of past calendar programs.
- Actually test with users/customers.

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
  - course-grain locking (one big lock)

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
  - course-grain locking (one big lock)
    pessimistic inaccuracy

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
  - automated static analysis verifies serializability with finer-grain locking, when some methods don't even use locks:
    * as this mechanism can still reject valid sets of methods that would ensure serializability

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
    - automated static analysis verifies serializability with finer-grain locking, when some methods don't even use locks:
      * as this mechanism can still reject valid sets of methods that would ensure serializability
      pessimistic inaccuracy

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
    - programmer required to use a particular concurrency control protocol and static analysis checking conformance to protocol

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
  - programmer required to use a particular concurrency control protocol and static analysis checking conformance to protocol
    simplified property

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
    - augmenting the data accesses to build a serializability graph structure representing the ’happens before’ relation among transactions in testing. It can be shown that the transactions executed in serializable manner if and only if the serializability graph is acyclic.

- 2.4: Compare and contrast the following to ensure serializability in a multi-threaded application on the three axes (pessimistic inaccuracy, optimistic inaccuracy, and simplified properties.
  - augmenting the data accesses to build a serializability graph structure representing the ’happens before’ relation among transactions in testing. It can be shown that the transactions executed in serializable manner if and only if the serializability graph is acyclic.
    Simplified property & optimistic inaccuracy