

# Chapter 01

Product line software development depends on the reuse of existing software components to provide software engineering leverage.

- a) True
- b) False

The implementation of cloud computing requires the development of an architecture that encompasses \_\_\_\_\_ and \_\_\_\_\_ services.

- a) Good; bad
- b) Complete; incomplete
- c) Collaborative; combative
- d) Front-end; back-end

A software product line is a set of software-intensive systems that share a common, managed \_\_\_\_\_:

- a) Set of problems
- b) Set of products
- c) Set of features
- d) Set of lines

## Chapter 02

Which of the items listed below is not one of the software engineering layers?

- a) Process
- b) Manufacturing
- c) Methods
- d) Tools

The seven principles Hooker proposes for software engineering as a practice are:

- a) Communication, planning, modeling, construction, deployment, re-design, remodel
- b) The reason it all exists; KISS; maintain the vision; what you produce, others will consume; be open to the future; plan ahead for reuse; think!
- c) The reason it all exists; KISS; maintain the vision; what you produce, others will consume; analysis; design; program
- d) Analysis; planning; maintain the vision; what you produce, others will consume; be open to the future; plan ahead for reuse; think!

## Chapter 07

Which of the following is not one of the principles of good coding?

- a) Create unit tests before you begin coding
- a) Create unit tests before you begin coding
- b) Refactor the code after you complete the first coding pass
- c) Write self-documenting code, not program documentation

## Chapter 08

During project inception the intent of the of the tasks are to determine

- a) basic problem understanding
- b) nature of the solution needed
- c) people who want a solution
- d) none of the above

b) a, b, c

Which of the following is not one of the context-free questions that would be used during project inception?

- a) What will be the economic benefit from a good solution?
- b) Who is behind the request for work?
- c) Who will pay for the work?
- d) Who will use the solution?

A stakeholder is anyone who will purchase the completed software system under development.

a) True

b) False

In win-win negotiation, the customer's needs are met even though the developer's need may not be.

a) True

b) False

## Chapter 09

It is important to consider alternative actor interactions when creating a preliminary use case.

- a) True
- b) False

In many cases there is no need to create a graphical representation of a usage scenario.

- a) True
- b) False

## Chapter 10

Attributes are chosen for an object by examining the problem statement and identifying the entities that appear to be related.

- a) True
- b) False

## Chapter 11

The behavior modeling is only used in the analysis of real-time systems.

- a) True
- b) False

For purposes of behavior modeling a state is any

- a) consumer or producer of data.
- b) data object hierarchy.
- c) observable mode of behavior.
- d) well defined process.

What are the elements of a WebApp interaction model?

- a) activity diagrams, sequence diagrams, state diagrams, interface prototype
- b) activity diagrams, collaboration diagrams, sequence diagrams, state diagrams
- c) use-cases, sequence diagrams, state diagrams, interface prototype
- d) use-cases, sequence diagrams, state diagrams, sequence diagrams

## Chapter 12

Which of the following are areas of concern in the design model?

- a) Architecture
- b) Data
- c) Interfaces
- d) Project scope
- e) a, b, c

What types of abstraction are used in software design?

- a) Control
- a) Data
- b) Environmental
- c) Procedural
- d) a, b, d

Since modularity is an important design goal it is not possible to have too many modules in a proposed design.

- a) True
- b) False

Cohesion is a qualitative indication of the degree to which a module

- a) can be written more compactly.
- b) focuses on just one thing.
- c) is able to complete its function in a timely manner.
- d) is connected to other modules and the outside world.

Coupling is a qualitative indication of the degree to which a module

- a) can be written more compactly.
- b) focuses on just one thing.
- c) is able to complete its function in a timely manner.
- d) is connected to other modules and the outside world.

Which design is equivalent to the floor plan of a house?

- a) Architectural design
- b) Component-level design
- c) Data design
- d) Interface design

Which design model is equivalent to the detailed drawings of the access points and external utilities for a house?

- a) Architectural design
- b) Component-level design
- c) Data design
- d) Interface design

Which design model is equivalent to a set of detailed drawings for each room in a house?

- a) Architectural design
- b) Component-level design
- c) Data design
- d) Interface design



List the four design models required for a complete specification of a software design and the role of each.

1. Data/Class design – transforms analysis classes into implementation classes and data structures
2. Architectural design – defines relationships among the major software structural elements
3. Interface design – defines how software elements, hardware elements, and end-users communicate
4. Component-level design – transforms structural elements into procedural descriptions of software components

## Chapter 13

The best representation of system architecture is an operational software prototype.

- a) True
- b) False

The architectural representations can be an enabler for communication among project stakeholders.

- a) True
- b) False

An architectural description is often documented using an architecture template.

- a) True
- b) False

An architectural genre will often dictate the architectural approach that may be used for the structure to be built.

- a) True
- b) False

Before an architectural pattern can be chosen for use in a specific system it must have a code implementation to facilitate its reuse.

- a) True
- b) False

Once selected, archetypes always need to be refined further as architectural design proceeds.

- a) True
- b) False

Architectural design has no role in agile software process models.

- a) True
- b) False

An architectural style encompasses which of the following elements?

- a) Constraints
- b) Set of components
- c) Semantic models
- d) Syntactic models
- e) a,b,c

To determine the architectural style or combination of styles that best fits the proposed system, requirements engineering is used to uncover

- a) Algorithmic complexity
- b) Characteristics and constraints
- c) Control and data
- d) Design patterns

The criteria used to assess the quality of an architectural design should be based on system

- a) Accessibility
- b) Control
- c) Data
- d) Implementation
- e) b and c

What are the elements that make up a software architectural style?

Answer (Section 13.3):

- Set of components that perform required system functions.
- Set of connectors allowing communications among the components.
- Constraints describing how the components maybe integrated to form a system.
- Semantic models that enable the designer to understand the overall system properties by

analyzing the known properties of its components.

From Slide :

Each style describes a system category that encompasses: (1) a **set of components** (e.g., a database, computational modules) that perform a function required by a system, (2) a **set of connectors** that enable “communication, coordination and cooperation” among components, (3) **constraints** that define how components can be integrated to form the system, and (4) **semantic models** that enable a designer to understand the overall properties of a system by analyzing the known properties of its constituent parts.

What is an archetype?

Answer (Section 13.6.2):

An archetype is a class or pattern that represents a core abstraction that is critical to the

design of an architecture for the target system.

From Slide:

An *archetype* is an abstraction (similar to a class) that represents one element of system behavior

## Chapter 15

Which of the following interface design principles does not allow the user to remain in control of the interaction with a computer?

- a) allow interaction to be interruptible
- b) allow interaction to be undoable
- c) hide technical internals from casual users
- d) only provide one rigidly defined method for accomplishing a task

Which of the following interface design principles reduce the user's memory load

- a) define intuitive shortcuts
- e) disclose information in a progressive fashion
- f) establish meaningful defaults
- g) provide an on-line tutorial
- h) a, b, c

Interface consistency implies that

- a) each application should have its own distinctive look and feel
- b) input mechanisms remain the same throughout the application
- c) navigational methods are context sensitive
- d) visual information is organized according to a design standard
- e) b and d

Which model depicts the profile of the end users of a computer system

- a) design model
- b) implementation model
- c) user model
- d) system perception

Which model depicts the image of a system that an end user creates in his or her Head?

- a) design model
- b) user model

- c) system model
- d) system perception

Which model depicts the look and feel of the user interface along with all supporting information

- a) implementation model
- b) user model
- c) system model
- d) system perception

Which of these framework activities is not normally associated with the user interface design processes

- a) cost estimation
- b) interface construction
- c) interface validation
- d) user and task analysis

Which approach(es) to user task analysis can be useful in user interface design?

- a) have users indicate their preferences on questionnaires
- b) rely on the judgement of experienced programmers
- c) study existing computer-based solutions
- d) observe users performing tasks manually
- e) c and d

Object-oriented analysis techniques can be used to identify and refine user task objects and actions without any need to refer to the user voice

- a) True
- b) False

The computer's display capabilities are the primary determinant of the order in which user interface design activities are completed.

- a) True
- b) False

It is sometimes possible that the interface designer is constrained by environmental factors that mitigate against ease of use for many users.

- a) True
- b) False

The reason for reducing the user's memory load is make his or her interaction with the computer quicker to complete

- a) True
- b) False

If past interactive models have created certain user expectations it is not generally good to make changes to the model.

- a) True
- b) False

## Chapter 17

Which of the following not part of the design pyramid for WebE design?

- a) Architectural design
- b) Business case design**
- c) Content design
- d) Navigation design

With WebApps content is everything, a poorly defined user interface will be quickly overlooked by frequent users.

- a) True
- b) False**

Content architecture and WebApp architecture are pretty much the same thing for many WebApps?

- a) True
- b) False**

Web navigational design involves creating a semantic navigational unit for each goal associated with each defined user role.

- a) True**
- b) False

To allow the user to feel in control of a WebApp, it is a good idea to mix both horizontal and vertical navigation mechanisms on the same page.

- a) True
- b) False**



## Chapter 19

Quality of conformance focuses on the degree to which the implementation of a design meets its requirements and performance goals.

a) True

b) False

Many software metrics can only be measured indirectly.

a) True

b) False

Software metrics represent direct measures of some manifestation of quality.

a) True

b) False

Good enough software delivers high quality software functions along with specialized functions that contain known bugs.

a) True

b) False

Which of the following is likely to be the most expensive cost of quality?

a) Appraisal costs

b) External failure costs

c) Internal failure costs

d) Prevention costs

When a system fails to deliver required functions it is because the customer changes requirements?

a) True

b) False

## Chapter 22

In software quality assurance work there is no difference between software verification and software validation.

- a) True
- b) False

Testing MobileApps is not different than testing WebApps.

- a) True
- b) False

Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that software is able to continue execution without interruption.

- a) True
- b) False

Security testing attempts to verify that protection mechanisms built into a system protect it from improper penetration.

- a) True
- b) False

Stress testing examines the pressures placed on the user during system use in extreme environments.

- a) True
- b) False

Performance testing is only important for real-time or embedded systems.

- a) True
- b) False

What is the normal order of activities in which traditional software testing is organized?

- a) integration testing, system testing, unit testing, validation testing.
- b) unit testing, validation testing, system testing, integration testing
- c) unit testing, integration testing, validation testing, system testing
- d) validation testing, system testing, integration testing, unit testing

The OO testing integration strategy involves testing

- a) groups of classes that collaborate or communicate in some way
- b) single operations as they are added to the evolving class implementation
- c) single operations as they are added to the evolving class implementation
- d) none of the above

List four types of systems tests.

Answer (Section 22.8):

- Recovery testing
- Security testing
- Stress testing
- Performance testing

Describe object-oriented unit testing.

Class testing for OO software is equivalent to unit testing for conventional software. The

focus throughout is on designing and testing appropriate sequences of operations to exercise

all class states. It makes little sense to test operations or algorithms individually for classes.

## Chapter 25

Which of the following is not one of the dimensions of quality used to assess a WebApp?

- a) Content
- b) Maintainability
- c) Navigability
- d) Usability

Which of the following is not one of the objectives of WebApp content testing?

- a) Find organizational or structure errors
- b) Identify linking errors
- c) Uncover semantic errors
- d) Uncover syntactic errors

Which of the following is not a WebApp interface mechanism?

- a) Browser
- b) Cookies
- c) Forms
- d) Links

Which of the following is not a testable WebApp security element?

- a) Authentication
- b) Encryption
- c) Firewalls
- d) Penetration

The purpose of WebApp navigation syntactic testing is to ensure the correct appearance of each navigation mechanism.

- a) True
- b) False

WebApp stress testing is a continuation load testing.

- a) True
- b) False

Summarize the steps used in a WebApp testing strategy..

Answer (Section 25.1.3):

- WebApp content model is reviewed to uncover errors.
- Interface model is reviewed to ensure all use-cases are accommodated.
- Design model for WebApp is reviewed to uncover navigation errors.
- User interface is tested to uncover presentation errors and/or navigation mechanics problems.
- Selected functional components are unit tested.
- Navigation throughout the architecture is tested.
- WebApp is implemented in a variety of different environmental configurations and the compatibility of WebApp with each is assessed.
- Security tests are conducted.
- Performance tests are conducted.
- WebApp is tested by a controlled and monitored group of end-users (looking for content errors, navigation errors, usability concerns, compatibility issues, reliability, and performance).

What are the objectives for content testing?

Answer (Section 25.3.1):

- Uncover syntactic errors in all media (e.g. typos)
- Uncover semantic errors (e.g. errors in completeness or accuracy)
- Find errors in organization or structure of content presented to end-user

Describe the WebApp interface testing strategy.

Answer (Section 25.4.1):

- Interface features are tested to ensure that design rules, aesthetics, and related visual content is available for user without error.
- Individual interface mechanisms are tested using unit testing strategies.
- Each interface mechanism is tested in the context of a use-case of navigation semantic unit (e.g. threafor a specific user category)
- Complete interface is tested against selected use-cases and navigation semantic unit to uncover interface semantic errors
- Interface is tested in a variety of environments to ensure compatibility

## Chapter 36

Software supportability is not concerned with either the provision of hardware or infrastructure.

- a) True
- b) False

Business process reengineering is often accompanied by software reengineering.

- a) True
- b) False

Business process reengineering does not have a start or end, it is an evolutionary process.

- a) True
- b) False

Code restructuring is a good example of software reengineering.

- a) True
- b) False

The cost benefits derived from reengineering are realized largely due to decreased maintenance and support costs for the new software product.

- a) True
- b) False



Which of the following is not an example of a business process?

- a) Designing a new product
- b) Hiring an employee
- c) Purchasing services
- d) Testing software

Which of the following activities is not part of the software reengineering process model?

- a) Forward engineering
- b) Inventory analysis
- c) Prototyping
- d) Reverse engineering

Software reengineering process model includes restructuring activities for which of the following work items?

- a) Code
- b) Documentation
- c) Data
- d) All of the above

Which of these benefits can be achieved when software is restructured?

- a) Higher quality programs
- b) Reduced maintenance effort
- c) Software easier to test
- d) All of the above

What are the benefits of software restructuring?

Answer (Section 36.7):

- Improved program and documentation quality
- Makes software easier to learn, reduces frustration, and improves productivity
- Reduces effort required to maintain software
- Software is easier to test and debug

Final Exam Question

How does business process reengineering (BPR) differ from software reengineering?

Answer (Section 36.4):

BPR defines business goals, identifies and evaluates existing business processes,

and creates revised business processes that better meet current goals. The software

reengineering process activities have as their intent the creation of new versions of

existing programs that exhibit higher quality and higher maintainability.

## Chapter 37

An effective software process improvement effort relies on the same framework for each project.

- a) True
- b) False

SPI is only justified for large software organizations.

- a) True
- b) False

The most difficult part of SPI is establishing a consensus for starting the process.

- a) True
- b) False

It is often difficult to achieve consensus among different constituencies during the SPI selection and justification activity.

- a) True
- b) False

Evaluation only occurs during the SPI post mortem activity

- a) True
- b) False

SPI often fails because risks were not properly considered and no contingency planning occurred.

- a) True
- b) False

To be effective in modern software development SPI frameworks must become significantly more agile.

- a) True
- b) False

As process assessment is conducted which of the following issues should be focused on?

- a) Acceptance
- b) Commitment
- c) Consistency
- d) All of the above

Which of these individuals are not involved in the SPI education and training activities?

- a) Customers
- b) Manager
- c) Practitioners
- d) Stakeholders

Which is not one of the processes that need to be considered during process installation and migration?

- a) As-is
- b) Here-to-there
- c) Just-in-time
- d) To-be
- e) a, b, d

What are the elements of a software process (SPI) framework?

Answer (Section 37.1.1):

- Set of characteristics that must be present if an effective software process is to be achieved
- Method for assessing whether those characteristics are present
- Mechanism for summarizing the results of any assessment
- Strategy for assisting a software organization in implementing those process characteristics that have been found to be weak or missing

From Slide:

