What is Software?

developed or engineered, it is not manufactured *in the classical sense and doesn't "wear out." and most software continues to be custom-built.*

: Software Engineering *التعريف الأساسي لل*

***[Software engineering is]* establishment and use of sound engineering principles in order to obtain economically that is reliable and works efficiently on real machines.**

Software المكونات الأساسيه الموجوده في أي

(1)instructions . (2) data structures . (3) documentation

: Software Engineering أسباب أهمية ال

1-understand the problem before a software solution is developed.

2-software should exhibit high quality and maintainable.

Why must it change Legacy Software*?*

1-adapted 2-Enhanced 3-extended to make it interoperable 4-re-architected

Wear vs. Deterioration :

من أسباب تدهور النظام هو عدم التطوير المستمر والتحديث ليواكب تطورات التكنولوجيا

:Software Applications أمثله على ال

Embedded software , Product-line software , ...الخ

Product Line Software:

set of software-intensive systems that share set of features needs of a particular market , These software developed using the same application and data architectures , and allow development of many products , and share set of assets : (*requirements, architecture, design patterns, reusable components, test cases.)*

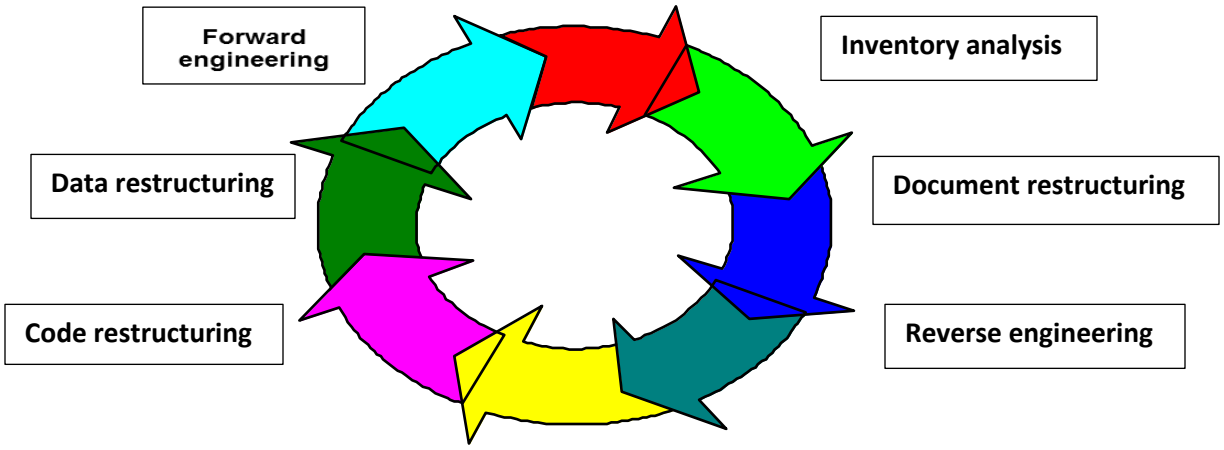
Software Maintenance : Software is released to end-users :

- ***within days, bug reports filter back***
- ***within weeks must be changed so that it can accommodate the special needs***
- ***within months They'll need a few enhancements to make it work in their world.***

Maintainable Software : It makes use of design patterns that allow ease of understanding , (design and implementation of the software must "assist" the person who is making the change)

***Software Supportability** : the capability of supporting a software system over its whole product life and Support personnel should have access to a database*

Software Reengineering :



Reengineering :

- *Business processes*
- *Software applications*
- *IT systems*

Business Process Reengineering :

- *Business definition Business goals within four key drivers : (cost reduction, time reduction, quality improvement, and personnel development and empowerment)*
- *Process identification Processes that are critical to achieving the goals*
- *Process evaluation analyzed and measured.*
- *Process specification and design. Based on information obtained during the first three BPR activities*
- *Prototyping. A redesigned business process*
- *Refinement and instantiation. Based on feedback from the prototype*

***Economics of Reengineering :** A cost/benefit analysis model for reengineering has been proposed by Sneed Nine parameters are defined:*

- P_1 *= maintenance cost for an application.*
- P_2 *= operation cost for an application.*
- P_3 *= business value of an application.*
- P_4 *= maintenance cost after reengineering.*

- P_5 = *operations cost after reengineering.*
- P_6 = *business value after reengineering.*
- P_7 = *estimated reengineering costs.*
- P_8 = *estimated reengineering calendar time.*
- P_9 = *reengineering risk factor*
- L = *life of the system*

cost associated with continuing defined as

$$C_{maint} = [P_3 - (P_1 + P_2)] \times L$$

costs associated with reengineering defined as

$$C_{reeng} = [P_6 - (P_4 + P_5) \times (L - P_8) - (P_7 \times P_9)]$$

costs presented in equations computed as

$$\text{cost benefit} = C_{reeng} - C_{maint}$$

Hooker's General Principles : بعض المبادى التي تطبق على ال Software engineering لتجاوز الصعوبات التي تواجهنا وقت عمل النظام

1-*The Reason It All Exists* 2-*KISS (Keep It Simple, Stupid!)* 3-*Maintain the Vision*

4-*What You Produce, Others Will Consume* 5-*Be Open to the Future*

6-*Plan Ahead for Reuse* 7-*Think!*

Essence of Practice in *Software engineering* :

1. *Understand the problem* (communication and analysis).
2. *Plan a solution* (modeling and software design).
3. *Carry out the plan* (code generation).
4. *Examine the result for accuracy* (testing and quality assurance).

factors must be considered when selecting a software project team structure :

- *difficulty of the problem to be solved*
- *size of the resultant program(s)*
- *(team lifetime)*
- *quality of the system to be built*
- *hardness of the delivery date*
- *degree (communication) required for the project*

Effective Software Team Attributes :

- **Sense of purpose and involvement and trust and improvement , Varity of team member skill sets .**

Avoid Team "Toxicity" :

- **poorly coordinated procedures.**
- **Unclear definition of roles.**
- **Continuous and repeated failure.**
- **which team members waste energy and lose focus on the objectives of the work .**
- **High chill caused by personal, business, or technological.**

Traits of Successful Software Engineers : من ناحية الفريق

1- **Know the needs of team members and stakeholders**
2- **Sense of individual responsibility**
3- **High sense of fairness**
4- **Pragmatic**
5- **Flexible under pressure**
6- **Attention to detail**

Organizational Paradigms :

- **closed paradigm : team traditional hierarchy**
- **random paradigm : team loosely and depends on individual initiative**
- **open paradigm : team that achieves some of the controls associated with the closed paradigm**
- **synchronous paradigm : relies on the natural divion of a problem and organizes team members to work on pieces of the problem with little active communication among themselves**

الفرق بين *mobile web application* و *mobile app* :

mobile web application : **Modern WebApps are much more than hypertext , augmented with tools like XML and Java to allow Web engineers including interactive computing capability , *allows a mobile device to access to web-based.***

mobile app : **such as cell phones or tablets Contain user interfaces , *can gain direct access to the hardware found on the device.***

Characteristics of WebApps :

1- ***Data driven :*** ***primary function to use hypermedia to present text, graphics, audio, and video.***
2- ***Content sensitive :*** ***important determinant of the quality of a WebApp.***
3- ***Continuous evolution. Unlike conventional application evolves over a series of planned.***
4- ***Immediacy : compelling need to get software to market quickly.***
5- ***Security.***
6- ***Aesthetics. : WebApp is its look and feel.***

Errors in a WebApp:

- ***you often see a symptom of the error, not the error itself.***
- ***it may be difficult or impossible to reproduce an error outside the environment in which the error was originally encountered.***
- ***many errors can be traced to the WebApp configuration.***
- ***errors can be difficult to trace across three architectural layers***
- ***Some errors are due to the static operating environment while others are attributable to the dynamic operating environment***

Cloud Computing :

provides distributed data storage and processing resources to networked computing devices.

Cloud computing containing both :

1- **Frontend services include the client devices and application.**
2- **Backend services include servers, data storage.**

Software Engineering using the Cloud:

- **Benefits : Removes device dependencies and available every where.**
- **Concerns : Dispersing cloud services outside the control of the software team may present reliability and security risks**

Impact of Social Media :

- **Blogs : share information**
- **Microblogs : Twitter allow posting of real-time messages**
- **Targeted on-line forums : post questions or opinions and collect answers**
- **Social networking sites : Facebook sharing information**
- **Social book marking : Delicious allow to keep track of and share web-based resources**

Principles that Guide Process : *Focus on quality at every step*

Principles that Guide Practice : *Focus on the transfer of information*

Agile Modeling Principles : Try to build useful models

Design Modeling Principles : Design models should be easily understandable.

Design Principles : Design is not coding, coding is not design.

Living Modeling Principles : common system view should be created.

Construction Principles : set of coding and testing ready for delivery to the customer or end-user.

Preparation Principles : Understand of the problem you're trying to solve.

Testing Principles : Static testing can yield high results.

Deployment Principles : Buggy software should be fixed first, delivered later.

Communication Principles :

1-Face-to-face communication is best ***2-Someone should facilitate the activity***

3- ***Listen focus on the speaker's words*** ***4-Prepare before you communicate***

Planning Principles :

1- *Understand the scope of the project*

2- *Be realistic*

3- *Involve the customer in the planning activity.*

4- *Recognize that planning is iterative*

Requirements Modeling Principles :

1- *information domain of a problem must be represented and understood.*
2- *behavior of the software must be represented*
3- *models information, function that uncovers detail in a layered (or hierarchical) fashion.*
4- *analysis task should move from essential information toward implementation detail*

Coding Principles :

1- *use of pair programming*

2- *Select data structures needs of the design*

3- *Write code that is self-documenting*

4-*Create nested loops makes them easily testable*

Validation Principles :

1- *Refactor the code*

2- *Perform unit tests and correct errors*

BPR Principles :

- *Organize around outcomes, not tasks.*
- *Link parallel activities instead of integrated their results. When different*

Requirements Engineering :

1- **Inception : understanding of the problem.**
2- **Elicitation : requirements from all stakeholders.**
3- **Elaboration : create an analysis model that identifies data.**
4- **Negotiation : agree on a deliverable system that is realistic for developers and customers.**
5- **Specification : set of models**
6- **Validation : looks for errors and missing information**

Requirements Monitoring :

1- *Run-time validation* 2- *Run-time verification*

Eliciting Requirements :

1- meetings by both software engineers and customers
2- definition mechanism (can be work sheets , or electronic board, chat room or virtual)

Layered Technology: بالترتيب

1-a "quality" focus *2-process model* *3-methods* *4-tools*

Process Model:

1-Framework : ***(Framework activities - Umbrella Activities)***

Framework activities :

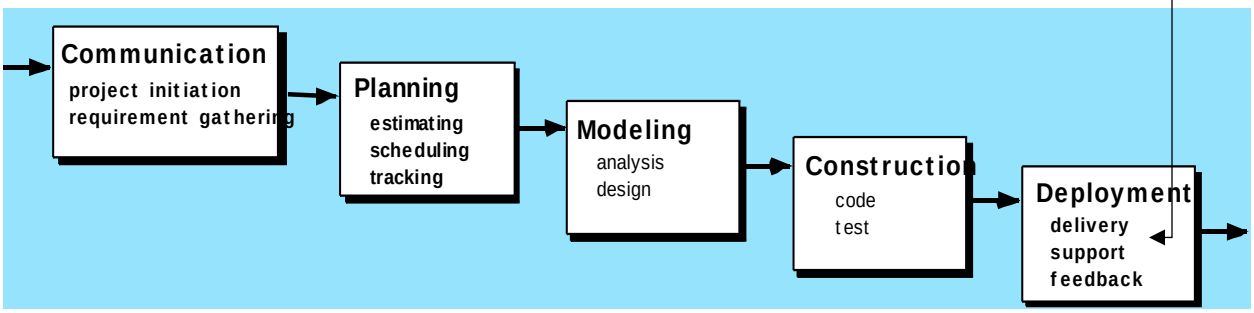
(1)Communication *(2)Planning* *(3)Modeling* *(4)Construction* *(5) Deployment*

نفس ال activityes

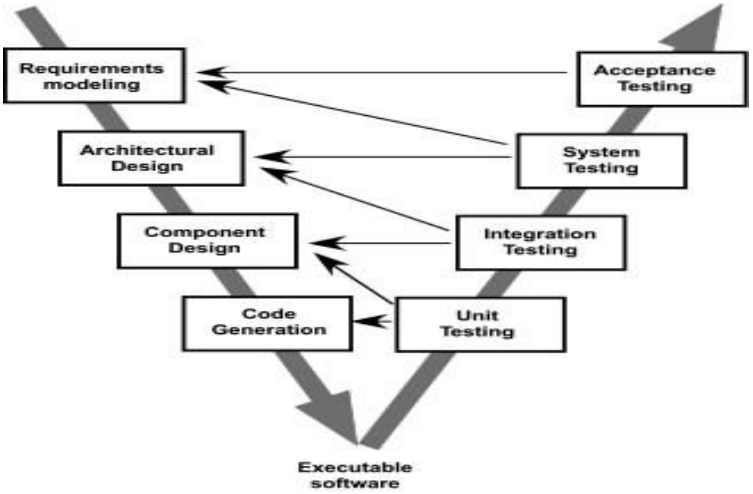
Umbrella Activities: *tracking and control and Risk management.*

.

2-The Waterfall Model :



3-The V-Model :



4-Incremental Model :

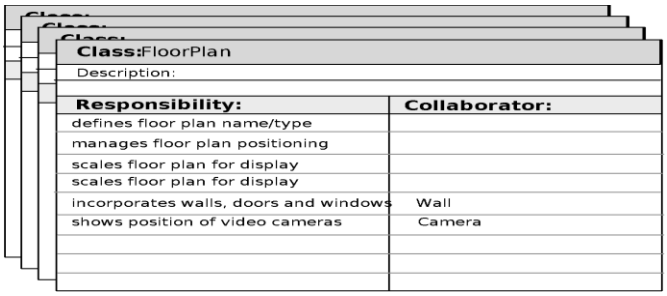
عباره عن نفس المشروع ولكن مقسم الى أجزاء ويتم العمل عليهم جميعا بنفس الوقت

5-Evolutionary Models : 3 types

- ***Prototyping :*** *سريعه وغير مرتبه*
- ***Spiral :*** *عملية التطوير تكون بشكل دائم*
- ***Concurrent :*** *مجموعه من الأنشطه التي تعمل بنفس الوقت ولكن كل نشاط له حاله معينه*

CRC Models (Class-responsibility-collaborator): ***provides a simple means for identifying and organizing the classes that are relevant to system or product requirements.***

describes CRC modeling: ***collection of standard index cards that represent classes. The cards are divided into three sections. Along the top of the card you write the name of the class. In the body of the card you list the class responsibilities on the left and the collaborators on the right.***



Functional Model : ***addresses two processing elements of the WebApp (user , operations)***

Configuration Model: ***(Server-side <operating system> , Client-side <Browser>)***

Content Model : ***are extracted from use-cases***

Interaction Model : Composed of four elements is an important UML notation :

- ***use-cases***
- ***sequence diagrams***
- ***state diagrams***
- ***a user interface prototype***

Design models : **represent characteristics of the software that help practitioners to construct it effectively .**

Requirements models also called analysis models :

1- *structured analysis :* ***considers data and the processes that transform .***
2- *object-oriented analysis :* ***focuses on definition of classes***

Patterns for Requirements Modeling : ***Software patterns are a mechanism for capturing domain by pattern .***

Discovering Analysis Patterns: ***The most basic element in the description of a requirements model is the use case.***

Domain Analysis : ***Analyze each application in the sample***

Interface Analysis: understanding :

- ***the people (end-users) who will interact with the system through the interface***
- ***the tasks that end-users must perform***

Architectural Tradeoff Analysis:

1- Collect scenarios 2- Describe the architectural styles

Inventory Analysis :

- ***build a table that contains all applications***
- ***establish a list of criteria of the application (name , year , ... الخ)***
- ***analyze to select candidates for reengineering***

Building the Analysis Model :

- **Scenario : 1-Functional 2- Use-case**
- **Class : Implied by scenarios**
- **Behavioral elements : State diagram**
- **Flow-oriented : Data flow diagram**

Behavioral Modeling : draw a state diagram or a sequence diagram

Data Modeling : focuses attention on the data domain

Class-Based Modeling : objects , operations , relationships , collaborations

Adapting a Process Model:

1-degree to which the customer and other stakeholders are involved with the project.

2-degree to which team organization.

3-degree to which work products are required

Deployment Elements : Descriptor form deployment diagrams show the computing environment but does not indicate configuration details

Component Elements : Describes the internal detail of each software component

Interface Elements : ***set of operations that describes the externally observable behavior of a class and provides access to its public operations***

Class Types :

- ***Entity classes also called model or business classes***
- ***Boundary classes are used to create the interface***
- ***Controller classes manage a "unit of work"***

Collaborations Classes : identify relationships between classes , three different generic relationships between classes :

- ***the is-part-of relationship***
- ***the has-knowledge-of relationship***
- ***the depends-upon relationship***

Associations and Dependencies :

Two analysis classes are often related to one another in some fashion : UML called associations used in data modeling.

State Representations : two different characterizations of states :

1- ***state of each class as the system performs function , state of a class takes on both passive state is simply , active state of an object***
2- ***state of the system as observed from the outside as the system performs function (state transition , event , action)***
3-

Software Design: Encompasses the set of principles, concepts, and practices that lead to the development of a high quality system or product

Software Engineering Design : (Data/Class design , Architectural design , Interface design , Component-level design)

Good software design should exhibit :

- ***Firmness:*** **program should not have any bugs .**
- ***Commodity:*** **program should be suitable for the purposes**
- ***Delight:*** **experience of using the program should be pleasurable one**

OO Design Concepts : (Inheritance , Messages , Polymorphism)

Design Classes : provide design detail that will enable analysis classes to be implemented (Boundary classes,Controller classes)

Design Class Characteristics :

- *Complete : includes all necessary attributes and methods*
- *Primitiveness : each class method focuses on providing one service*
- *High cohesion : small, focused, single-minded classes*
- *Low coupling : class collaboration kept to minimum*

Design Model Elements : (Data elements , Architectural elements , Interface elements , Component elements , Deployment elements)

Design Issues : (Response time , Help facilities , Error handling , Menu and command labeling , Application accessibility , Internationalization)

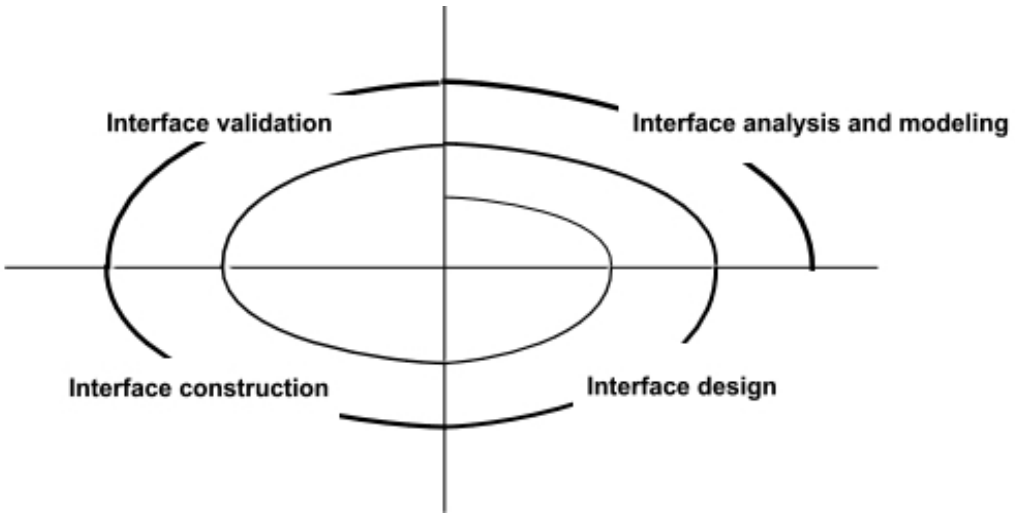
User Interface Design golden Rules :

- *Place the user in control*
- *Reduce the user's memory load*
- *Make the interface consistent*

User Interface Design Models:

- *User model : profile*
- *Design model : design realization*
- *Mental model (system perception) : user's mental image*
- *Implementation model : interface "look and feel"*

User Interface Design Process:



Interface Design Steps :

- ***define interface objects and actions***
- ***Define events***

Design and Quality : design must implement all of the : explicit requirements , understandable guide , provide a complete picture of the software.

Quality Guidelines : (using a notation , using a repeatable method)

Quality Function Deployment: *QFD* *is a quality management technique that translate the needs of the customer into technical requirements for software :*

- **Function deployment determines the "value"**
- **Information deployment identifies data**
- **Task deployment examines the behavior of the system**
- **Value analysis determines the relative priority of requirements**

Document Restructuring :

- ***Creating documentation is far too time consuming***
- ***Documentation must be updated, but we have limited resources***

Code Restructuring :

- **Source code is analyzed using a restructuring tool**
- **Poorly design code segments are redesigned**

Data Restructuring :

- **Unlike code restructuring is a full-scale reengineering activity and has a strong influence on program architecture**
- **begins with a reverse engineering activity.**

Extreme Programming (XP) ← **التركيز يكون على الكود فهو يحتوي على 2 من المبرمجين الأول يكتب الكود والثاني يصحح وبعد فتره يتبادلون الأدوار الأول يصحح والثاني يكتب الكود**

Type of agile most widely used agile process .

1- **XP Planning : cost**
2- **XP Design : CRC cards**
3- **XP Coding : pair programming**
4- **XP Testing : يكون بعد كل مرحله**

XP Team Values :

- **Communication , Simplicity , Feedback , Courage , Respect.**

What is "Agility"?

1- **Effective to change and communication among all stakeholders.**
2- **Organizing a team.**
3- **Drawing the customer onto the team.**

Yielding ... الفائده ((Rapid, incremental delivery of software))

Agile Process :

1- **driven by customer descriptions of what is required**
2- **plans are short-lived**
3- **Develops software iteratively**
4- **Delivers multiple 'software increments'**
5- **Adapts**

Generic Agile Teams :

Focus on team strength and realistic plan is governed by project requirements .

Architecture : architecture is not the operational software , enables a software engineer to:

(1) analyze the effectiveness of the design .

(2) consider architectural alternatives .

(3) reduce the risks .

Why is Architecture Important?

- ***enabler for communication between all parties (stakeholders)***
- ***The architecture highlights early design decisions***
- ***Architecture "constitutes a relatively small, and how its components work together".***

Architectural description language (ADL) : provides a semantics and syntax for describing a software architecture with the ability to: (decompose architectural components , compose individual components , represent interfaces between components)

IEEE Standard defines architectural description (AD) : "a collection of products to document an architecture." , description itself using multiple view , representation of a whole system related set of [stakeholder] .

IEEE Computer Society Recommended Practice for Architectural Description of Software-Intensive System (to establish a conceptual framework , to provide detailed guidelines , to encourage sound architectural) .

Architectural Descriptions : IEEE Computer Society Recommended Practice :

- ***to establish a conceptual framework.***
- ***to provide detailed guidelines.***
- ***to encourage sound architectural.***

Architectural Genres : Genre implies a specific category within the overall software domain (For example : genre of buildings following general styleseach general style, more specific styles)

Architectural Styles : Each style describes a system category :

- ***set of components : perform a function required by a system (e.g., a database)***
- ***set of connectors : enable "communication, coordination and cooperation" among components.***
- ***Constraints : define how components can be integrated to form the system***
- ***semantic models : enable a designer to understand the overall properties of a system by analyzing constituent of parts : (Data-centered architectures , Data flow architectures , Call and return architectures , Object-oriented architectures , Layered architectures)***

Architectural Patterns :

- ***Concurrency—applications must handle multiple tasks (operating system)***
- ***Persistence—Data persistsif it survives past the execution of the process that created Two patterns are common:***
  1- a database management system
  2- application level persistence
- ***Distribution : which systems or components within systems communicate with one another in a distributed environment : (broker between the client component and a server component)***
  1- defining and refining software components that implement each archetype

Architectural Considerations:

- ***Economy – The best software***
- ***Visibility – Architectural decisions***
- ***Spacing – Separation of concerns in a design***
- ***Symmetry – implies the system and balanced in attributes***
- ***Emergence - self-organized behavior and control.***

Architectural Decision Documentation:

1. ***Determine which information items are needed for each decision.***
2. ***Define links between each decision and appropriate requirements.***

Architectural Complexity : dependencies between components within the architecture :

1-Sharing dependencies 2-Flow dependencies 3-Constrained dependencies

Agility and Architecture : To avoid rework

Use-Cases : represents the functionality of the system :

- **collection of user scenarios that describe the thread of usage of a system**
- **Each scenario is described from the point-of-view *person or device that interacts with the software***

***actors* : represent roles people or devices play as the system functions**

***users :* can play a number of different roles for a given scenario**

Scenario-Based Modeling What to Write About?

describe the things (objects) that will be manipulated by the system.

WebApp Testing Strategy :

1- *The content model for the WebApp is reviewed to uncover errors.*
2- *Selected functional components are unit tested.*

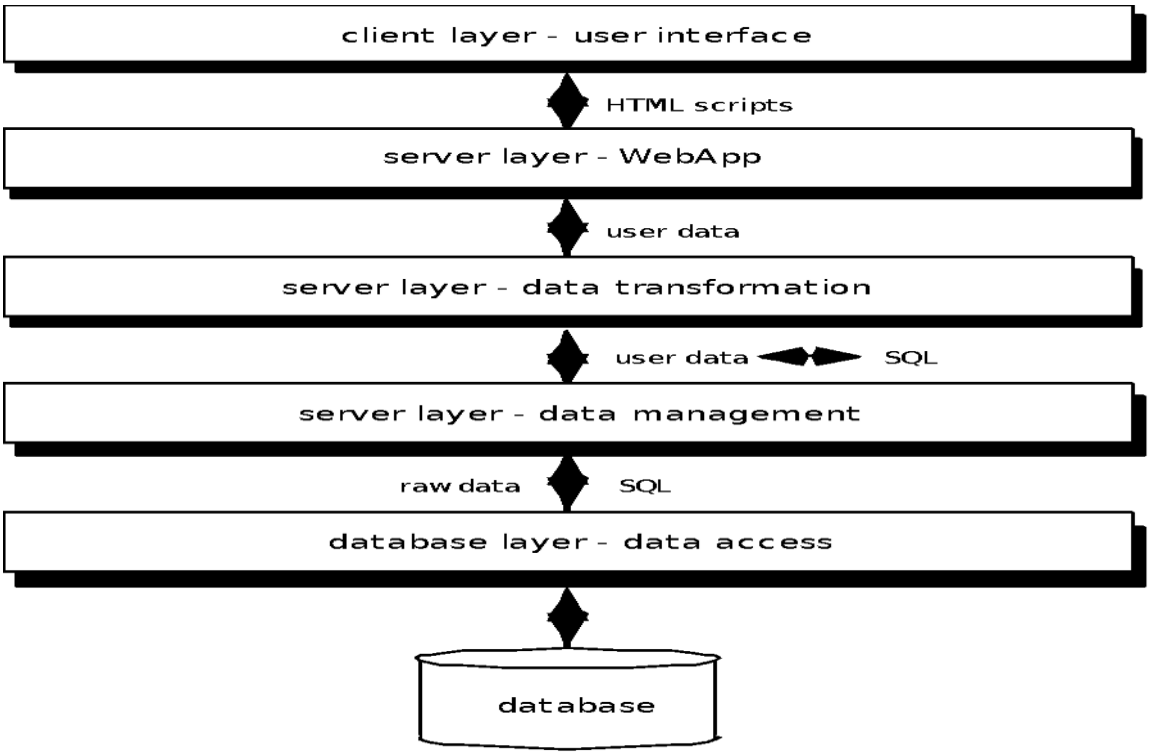
The Testing Process :

- ***Content Testing :*** *has three important objectives:*
  1- *to uncover syntactic errors*
  2- *to uncover semantic errors*
  3- *to find errors in the organization or structure of content*
- *interface testing*
- *navigation testing*
- *component testing*
- *configuration testing*
- *performance testing*
- *security testing*

Database Testing : *Tests are defined for each layer*

Testing Quality Dimensions :

1- ***Content :*** *is evaluated at both :*
   - *syntactic level—spelling,grammar*
   - *assessed for text-based documents*
2- ***Function :*** *is tested for correctness*
3- ***Structure*** *to ensure properly delivers WebApp content and function*
4- ***Usability*** *is tested to ensure supported by the interface*
5- ***Navigability*** *is tested to ensure all navigation syntax is uncover any navigation errors*
6- ***Performance*** *is tested under a variety of operating conditions*
7- ***Compatibility*** *is tested by executing the WebApp in a variety of different host configurations on both the client and server sides.*
8- ***Interoperability*** *is tested to ensure that the WebApp properly interfaces with other applications*
9- ***Security*** *is tested by assessing potential vulnerabilities*



User Interface Testing :

- ***The interface is tested within a variety of environments (e.g., browsers) to ensure that it will be compatible***
- ***Interface features are tested to ensure that design rules, aesthetics, and related visual content is available for the user without error.***

Testing Interface Mechanisms :

- ***Links : link the user to some other content object or function.***
- ***Forms : structured document containing blank fields***
- ***Client-side scripting list of programmed commands***
- ***Dynamic HTML manipulated on the client***
- ***Client-side pop-up windows small windows that pop-up without user interaction***
- ***CGI scripts allows a Web server to interact dynamically with users***
- ***Streaming content sometimes called "push" technology because the server pushes data to the client***
- ***Cookies—a block of data sent by the server and stored by a browser as a consequence of a specific user interaction***
- ***Application specific interface mechanisms include one or more "macro" interface mechanisms such as a shopping cart, credit card processing, or a shipping cost calculator.***

Usability Tests : ***Design by WebE team , executed by end-users***

Compatibility Testing : ***Compatibility testing is to define a set of "commonly encountered" client side computing configurations and their variants***

Component-Level Testing : ***Focuses on a set of tests that attempt to uncover errors in WebApp functions***

Navigation Testing : ***Navigation Testing :***

- ***Navigation links*** ***include internal links within the WebApp, external links to other WebApps***
- ***Redirects*** ***these links come into play when a user requests a non-existent URL***
- ***Bookmarks*** ***although bookmarks are a browser function***
- ***Frames and framesets*** ***tested for correct content, proper layout and sizing, download performance, and browser compatibility***
- ***Site maps*** ***to ensure that the link takes the user to the proper content or functionality.***
- ***Internal search engines*** ***Search engine testing validates the accuracy and completeness of the search***

Configuration Testing :

- ***Client-side*** ***Hardware , Operating system , Browser software , User interface components , Plug-ins , Connectivity***
- ***The number of configuration variables must be reduced to a manageable number***

Security Testing :

- ***On the client-side*****, vulnerabilities can often be traced to pre-existing bugs in browsers**
- ***On the server-side*****, vulnerabilities include denial-of-service attacks and malicious scripts**

Load Testing : ***The intent is to determine how the WebApp and its server-side environment will respond to various loading conditions :***

- ***N*****, the number of concurrent users**
- ***T,*** **the number of on-line transactions per unit of time**
- ***D,*** **the data load processed by the server per transaction**

Establishing the Groundwork :

- **Inception : Work toward collaboration**
- **Non-Functional Requirements :** ***quality attribute, performance attribute, security attribute, or general system constraint , two phase process is used to determine which NFR's are compatible:***
  1- ***The first phase is to create a matrix using each NFR as a column heading and the system SE guidelines a row labels .***
  2- **The second phase each NFR using a set of decision rules by classifying each NFR and guideline pair as or independent.**

What is a Data Object?

data object contains a set of attributes that act as an aspect, quality, characteristic, or descriptor of the object that representation of almost any composite information that must be understood by software (external entity , thing , occurrence , role , organizational unit , place , structure)

Rules of Thumb : Minimize coupling and Keep the model as simple

Defining Attributes : describe a class that has been selected for inclusion in the analysis model

Defining Operations : Operations can be divided into four categories: operations that manipulate data and about the state of object and perform a computation and monitor an object.

Exceptions : Describe situations (failures or user choices) that cause the system to exhibit unusual behavior.

semantic analysis pattern (SAP) "is a pattern that describes a small set of coherent use cases that together describe a basic generic application."

What is a Relationship?

Objects can be related in many different ways.

Separation of concerns : any complex problem can be more easily handled if it is subdivided into pieces

Modularity : ***modularity is the single attribute of software that allows a program to be intellectually manageable***

Hiding : ***controlled interfaces (Why Information Hiding? emphasizes communication through controlled interfaces , results in higher quality software)***

Functional independence : ***single-minded function and low coupling (Cohesion is an indication of relative functional strength of module performs a single task, requiring little interaction with other components in other parts of a program. Coupling is indication of the relative interdependence among modules depends on the interface complexity between modules)***

Aspects : ***a mechanism for understanding how global requirements affect design (aspect representation of a cross-cutting concern)***

Refactoring : ***a reorganization technique that simplifies the design***

****** Long answer Chapter 17******

"There are essentially two basic approaches to design: the artistic ideal of expressing yourself and the engineering ideal of solving a problem for a customer."

When should we emphasize WebApp design?

- **when content and function are complex**
- **when the size of the WebApp encompasses hundreds of content objects, functions, and analysis classes**
- **when the success of the WebApp will have a direct impact on the success of the business**

Design & WebApp Quality : (Security , Availability , Scalability , Time to Market)

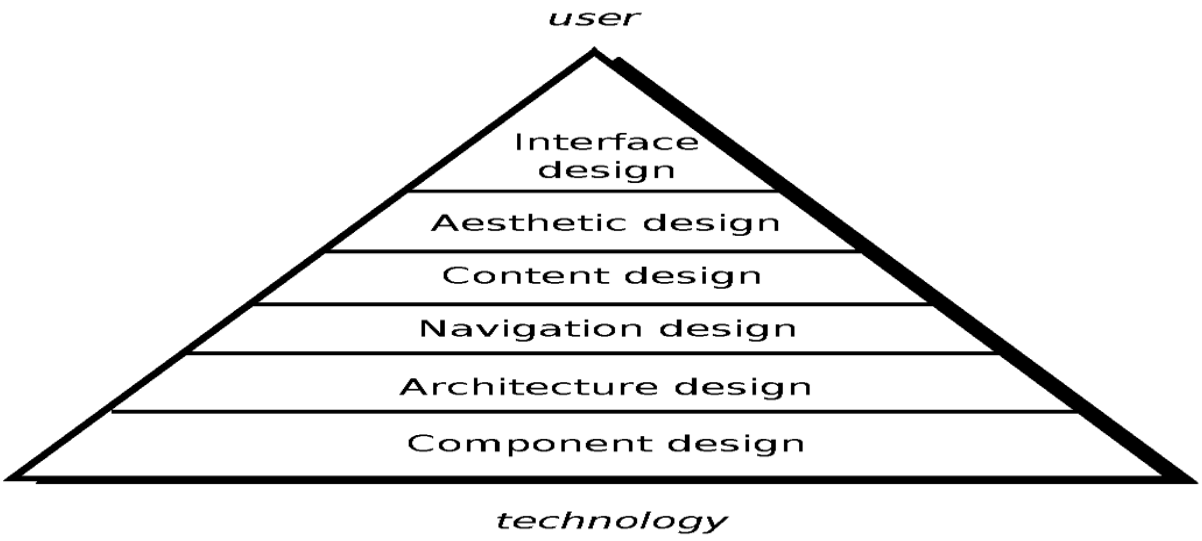
Quality Dimensions for End-Users : (*Time , Structural , Content , Accuracy and Consistency , Response Time and Latency , Performance)*

WebApp Design Goals :

1-Consistency : (Content , Graphic design (aesthetics) , Architectural design , Interface design , Navigation mechanisms)

- **2-Identity 3- Robustness 4- Navigability 5- Visual appeal 6- Compatibility**

WebE Design Pyramid:



Effective WebApp Interfaces :

- **Effective interfaces are visually apparent and forgiving**
- **Effective interfaces do not concern the user with the inner workings of the system**
- **Effective applications and services perform a maximum of work**

****** Long answer chapter 17 ******

Interface Design Principles: (Anticipation , Communication , Consistency , Controlled autonomy , Efficiency , Focus , Fitt's Law , Human interface objects , Latency reduction , Learnability , Maintain work product integrity , Readability , Track state , Visible navigation)

Aesthetic Design :

- ***Don't be afraid of white space.***
- ***Emphasize content.***
- ***Organize layout elements from top-left to bottom right.***
- ***Group navigation, content, and function geographically within the page.***
- ***Don't extend your real estate with the scrolling bar.***
- ***Consider resolution and browser window size when designing layout.***

Content Design:

- ***Develops a design representation for content objects***
- ***Represents the mechanisms required to instantiate their relationships to one another.***
- ***A content object has attributes that include content-specific information and implementation-specific attributes that are specified as part of design***

Architectural Design :

1- ***Content architecture : focuses on which objects are structured for presentation***
2- ***software must be placed into context***
3- ***set of architectural archetypes should be identified (archetype : is an abstraction " similar to a class ") .***
4- ***designer specifies the structure of the system by defining and refining software components that implement each archetype.***
5- ***WebApp architecture addresses in which the application is structured to manage user interaction***
6- ***Architecture design is conducted in parallel with interface design***

MVC Architecture:

- ***The model contains all application specific content and processing logic, including (all content objects , access to external data , all processing functionality that are application specific)***
- ***The view contains all interface specific functions and enables (processing logic , access to external data , all processing functionality required by the end-user)***

**** Long answer chapter 17 ****

- *The controller manages access to the model and the view and coordinates the flow of data between them*

Navigation semantic units (NSUs) : *set of information and related navigation structures that collaborate in the fulfillment of a subset of related user requirements*

1- *Ways of navigation (WoN)—represents the best navigation way or path for users with certain profiles to achieve their desired goal or sub-goal*
2- *Navigation nodes (NN) connected by Navigation links*

Navigation Design :

- *Begins with a consideration of the user hierarchy and related use-cases*
- *each user interacts with the WebApp, she encounters a series of navigation semantic units (NSUs)*

Navigation Syntax :

- *Individual navigation link : text-based links, icons, buttons*
- *Horizontal navigation bar : lists major content or functional categories in a bar containing appropriate links. In general, between 4 and 7 categories are listed.*
- *Vertical navigation column : lists major content categories , lists virtually all major content objects within the WebApp.*
- *Tabs : a metaphor that is nothing more than a variation of the navigation bar or column, representing content or functional categories as tab sheets that are selected when a link is required.*
- *Site maps : provide an all-inclusive tab of contents for navigation to all content objects and functionality contained within the WebApp.*

Component-Level Design :

- *perform localized processing to generate content and navigation capability in a dynamic fashion*
- *provide computation or data processing capability that are appropriate for the WebApp's business domain*
- *provide sophisticated database query and access*
- *establish data interfaces with external corporate systems.*

****Short answer chapter 19-22****

In 2005, *ComputerWorld* lamented that "bad software plagues nearly every organization "

A year later, *InfoWorld* wrote about the the " sorry state of software quality "

Today, software quality remains an issue, but who is to blame?

1- Customers blame developers , sloppy practices lead to low-quality software.
2- Developers blame customers , irrational delivery dates and a continuing stream of changes force them to deliver software

The *American Heritage Dictionary* defines *quality* as : "a characteristic or attribute of something."

For software, two kinds of quality may be encountered ??

1- Quality of design : encompasses requirements, specifications, and the design of the system.
2- Quality of conformance : is an issue focused primarily on implementation

$$User\ satisfaction = compliant\ product + good\ quality + delivery\ within\ budget\ and\ schedule$$

Quality—A Pragmatic View :

1- *transcendental view* argues (like Persig) that quality is something that you immediately recognize
2- *user view* sees quality in terms of an end-user's specific goals
3- *manufacturer's view* defines quality in terms of the original specification of the product
4- *product view* suggests that quality can be tied to inherent characteristics
5- *value-based view* measures quality based on how much a customer is willing to pay for a product

Software Quality : " *effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it* " .

Effective Software Process : establishes the infrastructure that supports any effort at building a high quality software product.

Useful Product : delivers the content, functions, and features that the end-user desires

Adding Value : *for both the producer and user* of a software product, high quality software provides benefits for the software organization and the end-user community : (greater software product revenue , better profitability , improved availability of information) .

Quality Dimensions : (Performance Quality , Feature quality , Reliability , Conformance , Durability , Serviceability , Aesthetics , Perception)

*******Short answer chapter 19-22******

Measuring Quality :

- General quality dimensions and factors are not adequate for assessing the quality
- Project teams need to develop a set of targeted questions
- Subjective measures of software quality may be viewed as little more than personal opinion
- Software metrics represent indirect measures of some manifestation of quality

"Good Enough" Software : " Good enough software delivers high quality functions and features that end-users desire, but at the same time it delivers other more obscure or specialized functions and features that contain known bugs " .

Cost of Quality :

- *Prevention costs* include : (quality planning , formal technical reviews , test equipment , Training)
- *Internal failure costs* include : (rework , repair , failure mode analysis)
- *External failure costs* are : (complaint resolution , product return and replacement , help line support , warranty work)

Cost : " The relative costs to find and repair an error or defect increase dramatically as we go from prevention to detection to internal failure to external failure costs" .

Low Quality Software : increases risks for both developers and end-users and liable to contain architectural flaws as well as implementation problems (bugs).

Impact of Management Decisions : (Estimation decisions , Scheduling decisions , Risk-oriented decisions)

Achieving Software Quality : Software quality is the result of good project management :

- Project management
- Quality control
- Quality assurance

Software Testing : " Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user " .

What Testing Shows ?

1- Errors
2- requirements conformance
3- performance
4- an indication of quality

****Short answer chapter 19-22****

Strategic Approach : **"Testing and debugging are different activities, but debugging must be accommodated in any testing strategy ".**

V & V :

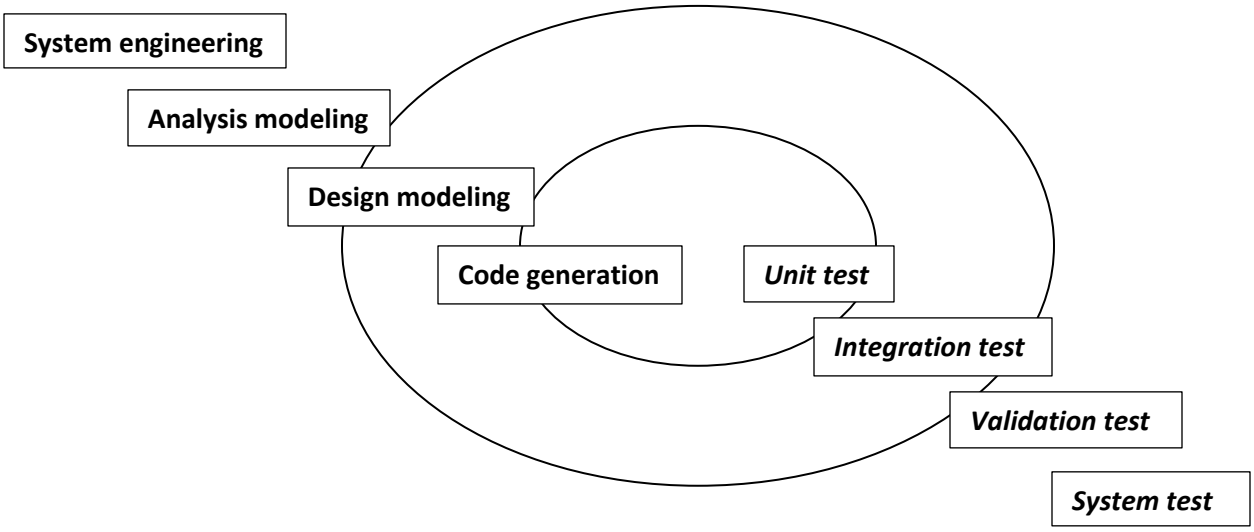
- ***Verification*** **refers to the set of tasks that ensure that software correctly implements a specific function. "Are we building the product right?"**
- ***Validation*** **refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. *Validation:* "Are we building the right product?"**

Who Tests the Software?

***Developer** Understands the system but, will test "gently" and, is driven by "delivery"*

***independent tester** Must learn about the system but, will attempt to break it and, is driven by quality*

Testing Strategy :



Testing Strategy :

- **begin by 'testing-in-the-small' and move toward 'testing-in-the-large'**
- **For conventional software The module (component)**
- **For OO software our focus when " testing in the small "**

Strategic Issues : **Understand the users of the software and develop a profile for each user category.**

****Short answer chapter 19-22****

What is Integration Testing Strategies ?? مهمه عباره عن خمس انواع

1- big bang approach
2- incremental construction strategy
3- Top Down Integration
4- Bottom-Up Integration
5- Sandwich Testing

Regression testing **(is the re-execution of some subset of tests , the program, its documentation, or the data that support it is changed , Regression testing helps to ensure that changes and may be conducted manually)**

Smoke Testing : A common approach for creating "daily builds" for product software the Software components that have been translated into code are integrated into a "build." The integration approach may be top down or bottom up.

General Testing Criteria:

- **Interface integrity – internal and external module interfaces**
- **Functional validity – test to uncover functional**
- **Information content – test for errors**
- **Performance**

Object-Oriented Testing:

- **begins by evaluating the correctness**
- **test case design draws on conventional methods**

Testing the CRC Model: Revisit the CRC model and Inspect the description of each CRC index

OO Testing Strategy :

integration applied three different strategies :

1- **thread-based testing**
2- **use-based testing**
3- **cluster testing**

****Short answer chapter 19-22****

MobileApp Testing :

- User experience testing
- Device compatibility testing
- Performance testing
- Connectivity testing
- Security testing
- Testing-in-the-wild
- Certification testing

High Order Testing :

- Validation testing
- System testing
- Alpha/Beta testing
- Recovery testing
- Security testing
- Stress testing
- Performance Testing

Debugging Effort :

- time required to correct the error and conduct regression tests.
- time required to diagnose the symptom and determine the cause

Symptoms & Causes :symptom and cause may be geographically separated :

- symptom may disappear when another problem is fixed or intermittent
- cause may be due to a combination of non-errors or a system or compiler error or assumptions that everyone believes

Bug Categories:

- function-related bugs
- system-related bugs
- data bugs
- coding bugs
- design bugs
- documentation bugs
- standards violations

Debugging Techniques : (brute force / testing , backtracking , induction , deduction)

Test Bank

Chapter 01

1-Which question no longer concerns the modern software engineer?

✓ **A) Why does computer hardware cost so much?**

B) Why does software take a long time to finish?

C) Why does it cost so much to develop a piece of software?

D) Why can't software errors be removed from products prior to delivery?

2-Today the increased power of the personal computer has brought about an abandonment of the practice of team development of software.

A) True

✓ **B) False**

3-Software is a product and can be manufactured using the same technologies used for other engineering artifacts.

A) True

✓ **B) False**

4-Software deteriorates rather than wears out because

A) Software suffers from exposure to hostile environments

B) Defects are more likely to arise after software has been used often

✓ **C) Multiple change requests introduce errors in component interactions**

D) Software spare parts become harder to order

5-Most software continues to be custom built because

A) Component reuse is common in the software world.

B) Reusable components are too expensive to use.

C) Software is easier to build without using someone else's components.

✓ **D) Off-the-shelf software components are unavailable in many application domains.**

6-The nature of software applications can be characterized by their information

A) complexity

B) content

C) determinacy

✓ D) both b and c

7-Modern software applications are so complex that it is hard to develop mutually exclusive category names.

✓ A) True

B) False

8-The so called "new economy" that gripped commerce and finance during the 1990s died and no longer influences decisions made by businesses and software engineers.

A) True

✓ B) False

9-The functionality of most computer systems does not need to be enhanced the lifetime of the system.

A) True

✓ B) False

10-Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind.

✓ A) True

B) False

11-Most software development projects are initiated to try to meet some business need.

✓ A) True

B) False

12-In general software only succeeds if its behavior is consistent with the objectives of its designers.

A) True

✓ B) False

Chapter 2

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

A) Process

✓ B) Manufacturing

C) Methods

D) Tools

2-Software engineering umbrella activities are only applied during the initial phases of software development projects.

A) True

✓ B) False

3-Which of these are the 5 generic software engineering framework activities?

✓ A) communication, planning, modeling, construction, deployment

B) communication, risk management, measurement, production, reviewing

C) analysis, designing, programming, debugging, maintenance

D) analysis, planning, designing, programming, testing

Feedback:

4-Process models are described as agile because they

A) eliminate the need for cumbersome documentation

✓ B) emphasize maneuverability and adaptability

C) do not waste development time on planning activities

D) make extensive use of prototype creation

5-Which of these terms are level names in the Capability Maturity Model?

A) Performed

B) Repeated

C) Reused

D) Optimized

✓ E) both a and d

6-Software processes can be constructed out of pre-existing software patterns to best meet the needs of a software project.

✓ A) True

B) False

Feedback:

7-Which of these are standards for assessing software processes?

A) SEI

B) SPICE

C) ISO 19002

D) ISO 9001

✓ E) both b and d

8-The best software process model is one that has been created by the people who will actually be doing the work.

✓ A) True

B) False

9-Which of these is not a characteristic of Personal Software Process?

A) Emphasizes personal measurement of work product

✓ B) Practitioner requires careful supervision by the project manager

C) Individual practitioner is responsible for estimating and scheduling

D) Practitioner is empowered to control quality of software work products

10-Which of these are objectives of Team Software Process?

A) Accelerate software process improvement

B) Allow better time management by highly trained professionals

C) Build self-directed software teams

D) Show managers how to reduce costs and sustain quality

✓ E) both b and c

11-Process technology tools allow software organizations to compress schedules by skipping unimportant activities.

A) True

✓ B) False

It is generally accepted that one cannot have weak software processes and create high quality end products.

✓ A) True

B) False

Chapter 4

1-Agility is nothing more than the ability of a project team to respond rapidly to change.

A) True

✓ B) False

2-Which of the following is not necessary to apply agility to a software process?

✓ A) Eliminate the use of project planning and testing

B) Only essential work products are produced

C) Process allows team to streamline tasks

D) Uses incremental product delivery strategy

3-How do you create agile processes to manage unpredictability?

A) Requirements gathering must be conducted very carefully

B) Risk analysis must be conducted before planning takes place

C) Software increments must be delivered in short time periods

D) Software processes must adapt to changes incrementally

✓ E) both c and d

4-In agile software processes the highest priority is to satisfy the customer through early and continuous delivery of valuable software.

✓ A) True

B) False

5-It is not possible to build software that meets the customers' needs today and exhibits the quality characteristics that will enable it to be extended tomorrow.

A) True

✓ B) False

6-Which of the following traits need to exist among the members of an agile software team?

A) Competence

B) Decision-making ability

C) Mutual trust and respect

✓ D) All of the above

7-All agile process models conform to a greater or lesser degree to the principles stated in the "Manifesto for Agile Software Development".

✓ A) True

B) False

8-What are the four framework activities found in the Extreme Programming (XP) process model?

A) analysis, design, coding, testing

B) planning, analysis, design, coding

C) planning, analysis, coding, testing

✓ D) planning, design, coding, testing

9-What are the three framework activities for the Adaptive Software Development (ASD) process model?

A) analysis, design, coding

B) feasibility study, functional model iteration, implementation

C) requirements gathering, adaptive cycle planning, iterative development

✓ D) speculation, collaboration, learning

10-The Dynamic Systems Development Method (DSDM) suggests a philosophy that is based on the Pareto principle (80% of the application can be delivered in 20% of the time required to build the complete application).

✓ A) True

B) False

11-Which is not one of the key questions that is answered by each team member at each daily Scrum meeting?

A) What did you do since the last meeting?

B) What obstacles are you encountering?

✓ C) What is the cause of the problems you are encountering?

12-In Feature Driven Development (FDD) a "feature" is a client-valued function that can be delivered in two months or less.

A) True

✓ B) False

13-Agile Modeling (AM) provides guidance to practitioner during which of these software tasks?

A) Analysis

B) Design

C) Coding

D) Testing

✓ E) both a and b

Chapter 5

1-The essence of software engineering practice might be described as understand the problem, plan a solution, carry out the plan, and examine the result for accuracy.

✓ A) True

B) False

2-Which of the following is not one of Hooker's core principles of software engineering practice?

A) All design should be as simple as possible, but no simpler

B) A software system exists only to provide value to its users.

✓ C) Pareto principle (20% of any product requires 80% of the effort)

D) Remember that you produce others will consume

3-Every communication activity should have a facilitator to make sure that the customer is not allowed to dominate the proceedings.

A) True

✓ B) False

4-The agile view of iterative customer communication and collaboration is applicable to all software engineering practice.

✓ A) True

B) False

5-Software engineers collaborate with customers to define which of the following?

A) Customer visible usage scenarios

B) Important software features

C) System inputs and outputs

✓ D) All of the above

6-Everyone on the software team should be involved in the planning activity so that we can

A) reduce the granularity of the plan

B) analyze requirements in depth

✓ C) get all team members to "sign up" to the plan

D) begin design

7-What role(s) do user stories play in agile planning?

A) Define useful software features and functions delivered to end-users

B) Determine a schedule used to deliver each software increment

C) Provide a substitute to performing detailed scheduling of activities

✓ D) Used to estimate the effort required build the current increment

E) both a and d

8-Which of the following activities is not one of the four things that need to be accomplished by the generic planning task set?

A) Develop overall project strategy

B) Identify the functionality to deliver in each software increment

✓ C) Create a detailed schedule for the complete software project

D) Devise a means of tracking progress on a regular basis

9-Analysis models depict software in which three representations?

A) architecture, interface, component

B) cost, risk, schedule

✓ C) information, function, behavior

10-The customer can directly observe both the difference between the internal quality of a design and its external quality?

A) True

✓ B) False

11-Teams using agile software practices never create models.

A) True

✓ B) False

12-Many of the tasks from the generic task sets for analysis modeling and design can be conducted in parallel with one another.

✓ A) True

B) False

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

A) Create unit tests before you begin coding

B) Create a visual layout that aids understanding

✓ C) Keep variable names short so that code is compact

14-A successful test is one that discovers at least one as-yet undiscovered error.

✓ A) True

B) False

15-Which of the following are tasks in the generic task set for construction?

A) Build a software component

B) Create a user interface

C) Unit test the component

D) Assess the quality of the component

✓ E) both a and c

16-Which of the following are valid reasons for collecting customer feedback concerning delivered software?

A) Allows developers to make changes to the delivered increment

B) Delivery schedule can be revised to reflect changes

C) Developers can identify changes to incorporate into next increment

✓ D) All of the above

Chapter 6

1-Software engineers do not need to consider hardware when designing a computer-based system.

A) True

✓ B) False

2-Which of the following can be elements of computer-based systems?

A) documentation

B) software

C) people

D) hardware

✓ E) all of the above

3-The system engineering process usually begins with the

A) detailed view

B) domain view

C) element view

✓ D) world view

4-To construct a system model the engineer should consider which of the following restraining factors?

A) assumptions

B) budget

C) constraints

D) schedule

✓ E) both a and c

5-By following modern system engineering practices simulation of reactive systems is no longer necessary.

A) True

✓ B) False

6-During business process engineering, three different architectures are examined.

✓ A) applications, data, technology infrastructure

B) communications, organization, financial infrastructure

C) network, database, reporting structure

D) systems, requirements, data structure

7-Which elements of business processing engineering are the responsibilities of the software engineer?

A) business area analysis

B) business system design

C) construction and integration

D) information strategy planning

✓ E) both b and c

8-The goal of product engineering is to translate the customer's desire for a set of defined capabilities into a working product.

✓ A) True

B) False

9-The architecture components for product engineering are

✓ A) data, hardware, software, people

B) data, documentation, hardware, software

C) data, hardware, software, procedures

D) documentation, hardware, people, procedures

10-The top level of the hierarchical model of a system is known as the

A) AFD

B) DFD

✓ C) SCD

D) SFD

11-The system model template contains which of the following elements

A) input

B) output

C) user interface

✓ D) all of the above

12-UML notations that can be used to model the hardware and software elements of a system are

A) Activity diagrams

B) Class diagrams

C) Deployment diagrams

D) Use-case diagrams

✓ E) a, b, and c

Chapter 7

1-Requirements engineering is a generic process that does not vary from one software project to another.

✓ A) True

B) False

2-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 bbove

✓ E) a, b and c

3-Three things that make requirements elicitation difficult are problems of

A) budgeting

B) scope

C) understanding

D) volatility

✓ E) b, c and d

4-The result of the requirements engineering elaboration task is an analysis model that defines which of the following problem domain(s)?

A) information

B) functional

C) behavioral

✓ D) all of the above

5-It is relatively common for different customers to propose conflicting requirements, each arguing that his or her version is the right one.

✓ A) True

B) False

6-The system specification describes the

✓ A) Function, performance and constraints of a computer-based system

B) implementation of each allocated system

C) element software architecture

D) time required for system simulation

7-The best way to conduct a requirements validation review is to

A) examine the system model for errors

B) have the customer look over the requirements

C) send them to the design team and see if they have any concerns

✓ D) use a checklist of questions to examine each requirement

8-The use of traceability tables helps to

A) debug programs following the detection of run-time errors

B) determine the performance of algorithm implementations

✓ C) identify, control, and track requirements changes

D) none of the above

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

A) True

✓ B) False

10-The job of the requirements engineer is to categorize all stakeholder information in a way that allows decision makers to choose an internally consistent set of requirements.

✓ A) True

B) False

11-The nature of collaboration is such that all system requirements are defined by consensus of a committee of customers and developers.

A) True

✓ B) False

12-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 against this project?

C) Who will pay for the work?

D) Who will use the solution?

13-In collaborative requirements gathering, the facilitator

A) cannot be a member of the software team

B) cannot be a customer

✓ C) controls and facilitates the process

D) must be an outsider

14-Which of the following is not one of the requirement classifications used in Quality Function Deployment (QFD)?

A) exciting

B) expected

✓ C) mandatory

D) normal

15-The work products produced during requirement elicitation will vary depending on the

A) size of the budget

✓ B) size of the product being built

C) software process being used

D) stakeholders needs

16-Use-case actors are always people, never system devices.

A) True

✓ B) False

17-Which of following is not a UML diagram used creating a system analysis model?

A) activity diagram

B) class diagram

✓ C) dataflow diagram

D) state diagram

18-Analysis patterns facilitate the transformation of the analysis model into a design model by suggesting reliable solutions to common problems.

A) True

✓ B) False

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

A) True

✓ B) False

20-In requirements validation the requirements model is reviewed to ensure its technical feasibility.

A) True

✓ B) False

Chapter 8

1-Which of the following is not an objective for building an analysis model?

A) define set of software requirements that can be validated

B) describe customer requirements

✓ C) develop an abbreviated solution for the problem

D) establish basis for software design

2-Object-oriented domain analysis is concerned with the identification and specification of reusable classes within an application domain.

✓ A) True

B) False

3-The data dictionary contains descriptions of each software

A) control item

B) data object

C) diagram

D) notation

✓ E) both a and b

4-Which of these is not an element of an object-oriented analysis model?

A) Behavioral elements

B) Class-based elements

✓ C) Data elements

D) Scenario-based elements

5-In analysis models the only data objects that need representation are those that will be implemented using software classes.

A) True

✓ B) False

6-The values that are assigned to an object's attributes make that object unique.

✓ A) True

B) False

7-The relationships shown in a data model must be classified to show their

A) cardinality

B) directionality

C) modality

D) probability

✓ E) both a and c

8-The entity relationship diagram

✓ A) depicts relationships between data objects

B) depicts functions that transform the data flow

C) indicates how data are transformed by the system

D) indicates system reactions to external events

9-A generalized description of a collection of similar objects is a

✓ A) class

B) instance

C) subclass

D) super class

10-Operations are object procedures that are invoked when an object receives a message.

✓ A) True

B) False

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

✓ A) True

B) False

12-UML activity diagrams are useful in representing which analysis model elements?

A) Behavioral elements

B) Class-based elements

C) Flow-based elements

✓ D) Scenario-based elements

13-The data flow diagram

A) depicts relationships between data objects

B) depicts functions that transform the data flow

C) indicates how data are transformed by the system

D) indicates system reactions to external events

✓ E) both b and c

14-Control flow diagrams are

A) needed to model event driven systems.

B) required for all systems.

C) used in place of data flow diagrams.

D) useful for modeling real-time systems.

✓ E) both a and d

15-The data flow diagram must be augmented by descriptive text in order to describe the functional requirements for a software product.

✓ A) True

B) False

16-Which of the following should be considered as candidate objects in a problem space?

A) events

B) people

C) structures

✓ D) all of the above

17-Attributes cannot be defined for a class until design has been completed.

A) True

✓ B) False

18-Which of the following is not one of the broad categories used to classify operations?

A) computation

B) data manipulation

C) event monitors

✓ D) transformers

19-Which of the following items does not appear on a CRC card?

A) class collaborators

B) class name

✓ C) class reliability

D) class responsibilities

20-Class responsibilities are defined by

A) its attributes only

B) its collaborators

C) its operations only

✓ D) both its attributes and operations

21-An analysis package involves the categorization of analysis model elements into useful groupings.

✓ A) True

B) False

22-Events occur whenever a(n)

✓ A) actor and the OO system exchange information

B) class operation is invoked

C) messages are passed between objects

D) all of the above

23-The state diagram

A) depicts relationships between data objects

B) depicts functions that transform the data flow

C) indicates how data are transformed by the system

✓ D) indicates system reactions to external events

24-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.

Chapter 9

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

A) architecture
B) data
C) interfaces
D) project scope
✓ E) a, b and c

2-The importance of software design can be summarized in a single word

A) accuracy
B) complexity
C) efficiency
✓ D) quality

3-Which of these are characteristics of a good design?

A) exhibits strong coupling between its modules
B) implements all requirements in the analysis model
C) includes test cases for all components
D) provides a complete picture of the software
✓ E) both b and d

4-Which of the following is not a characteristic common to all design methods?

✓ A) configuration management
B) functional component
C) notation quality assessment
D) guidelines refinement heuristics

5-Software design is an iterative generic process that may be applied without modification to any software project.

A) True
✓ B) False

6-What types of abstraction are used in software design?

A) control
B) data
C) environmental
D) procedural
✓ **E) a, b and d**

7-Which of the following models can be used to represent the architectural design of a piece of software.

A) Dynamic models
B) Functional models
C) Structural models
✓ **D) All of the above**

8-Design patterns are not applicable to the design of object-oriented software?

A) True
✓ **B) False**

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

A) True
✓ **B) False**

10-Information hiding makes program maintenance easier by hiding data and procedure from unaffected parts of the program.

✓ **A) True**
B) False

11-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.

12-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.**

13-When using structured design methodologies the process of stepwise refinement is unnecessary.

A) True
✓ **B) False**

14-Software designs are refactored to allow the creation of software that is easier to integrate, easier to test, and easier to maintain.

✓ **A) True**
B) False

15-Inheritance provides a mechanism by which changes to lower level classes can be propagated to all super classes quickly.

A) True
✓ B) False

16-Polymorphism reduces the effort required to extend an object system by

A) coupling objects together more tightly.
✓ B) enabling a number of different operations to share the same name.
C) making objects more dependent on one another.
D) removing the barriers imposed by encapsulation.

17-Which of the following is not one of the five design class types

A) Business domain classes
✓ B) Entity classes
C) Process classes
D) User interface classes

18-Which design model elements are used to depict a model of information represented from the user's view?

A) Architectural design elements
B) Component-level design elements
✓ C) Data design elements
D) Interface design elements

19-Which design is analogous to the floor plan of a house?

✓ A) Architectural design
B) Component-level design
C) Data design
D) Interface design

20-Which design model is analogous 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

21-Which design model is analogous 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

22-The deployment design elements specify the build order for the software components.

A) True
✓ B) False

23-One of the key problems in software reuse is the inability to find existing reusable design patterns when hundreds of candidates exist.

✓ A) True
B) False

24-Design patterns are best thought of as coding patterns.

A) True
✓ B) False

25-Frameworks and design patterns are the same thing as far as designers are concerned.

A) True
✓ B) False

Chapter 10

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

A) True
✓ B) False

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

✓ A) True
B) False

3-Which of these characteristics are true of a data warehouse, but not a typical data base?

A) business level orientation
B) currency of information
C) integration
D) nonvolatility
✓ E) both c and d

4-Data design actually begins during the creation of the analysis model, not the architectural model.

✓ A) True
B) False

5-An architectural style encompasses which of the following elements?

A) constraints
B) set of components
C) semantic models
D) syntactic models
✓ E) a, b and c

6-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

7-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

8-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) both b and c

9-During the process of modeling the system in context, systems that interact with the target system are represented as

A) Peer-level systems

B) Subordinate systems

C) Superordinate systems

D) Working systems

✓ E) a, b and c

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

✓ A) True

B) False

11-Which of the following is not an example of infrastructure components that may need to be integrated into the software architecture?

A) Communications components

B) Database components

✓ C) Interface components

12-In the architecture trade-off analysis method the architectural style should be described using the

A) data flow view

B) module view

C) process view

D) user view

✓ E) a, b and c

13-Quantitative methods for assessing the quality of proposed architectural designs are readily available.

A) True

✓ B) False

14-A useful technique for evaluating the overall complexity of a proposed architecture is to look at the component

A) cohesion flow

B) dependencies

C) sharing dependencies

D) size

✓ E) both b and c

15-When the overall flow in a segment of a data flow diagram is largely sequential and follows straight-line paths, _____ is present.

A) low coupling

B) good modularity

C) transaction flow

✓ D) transform flow

16-When a single item that triggers other data flow along one of many paths of a data flow diagram, _____ characterizes the information flow.

A) high coupling

B) poor modularity

✓ C) transaction flow

D) transform flow

17-When you encounter both transform flow and transaction flow in the 18-same DFD the flow is partitioned and the appropriate mapping technique is used on each part of the DFD.

✓ **A) True**

B) False

19-In transaction mapping the first level factoring results in the

A) creation of a CFD

✓ **B) derivation of the control hierarchy**

C) distribution of worker modules

D) refinement of the module view

Chapter 11

1-In the most general sense a component is a modular building block for computer software.

✓ **A) True**

B) False

2-In the context of object-oriented software engineering a component contains

A) attributes and operations

B) instances of each class

C) roles for each actor (device or user)

✓ **D) a set of collaborating classes**

3-In traditional software engineering, modules must serve in which of the following roles?

A) Control component

B) Infrastructure component

C) Problem domain component

✓ **D) All of the above**

4-Software engineers always need to create components from scratch in order to meet customer expectations fully.

A) True

✓ **B) False**

5-Which of the following is not one of the four principles used to guide component-level design?

A) Dependency Inversion Principle

B) Interface Segregation Principle

C) Open-Closed Principle

✓ D) Parsimonious Complexity Principle

6-During component-level design it is customary to ignore organization issues like subsystem membership or packaging.

A) True

✓ B) False

7-The use of stereotypes can help identify the nature of components at the detailed design level.

✓ A) True

B) False

8-Classes and components that exhibit functional, layer, or communicational cohesion are relatively easy to implement, test, and maintain.

✓ A) True

B) False

9-Software coupling is a sign of poor architectural design and can always be avoided in every system.

A) True

✓ B) False

10-In component design, elaboration requires which of the following elements to be described in detail?

A) Source code

B) Attributes

C) Interfaces

D) Operations

✓ E) b, c and d

11-In component-level design "persistent data sources" refer to

A) Component libraries

B) Databases

C) Files

D) All of the above

✓ E) both b and c

12-The object constraint language (OCL) complements UML by allowing a software engineer to use a formal grammar to construct unambiguous statements about design model elements.

✓ A) True

B) False

13-OCL is not strong enough to be used to describe pre- or post conditions for design actions.

A) True

✓ B) False

14-Which of these constructs is used in structured programming?

A) branching

B) condition

C) repetition

D) sequence

✓ E) b, c, and d

15-Which of these is a graphical notation for depicting procedural detail?

A) process diagram

B) decision table

C) ER diagram

✓ D) flowchart

16-A decision table should be used

A) to document all conditional statements

B) to guide the development of the project management plan

C) only when building an expert system

✓ D) when a complex set of conditions and actions appears in a component

17-A program design language (PDL) is often a

✓ **A) combination of programming constructs and narrative text**

B) legitimate programming language in its own right

C) machine readable software development language

D) useful way to represent software architecture

Which of these criteria are useful in assessing the effectiveness of a particular design notation?

A) maintainability

B) modularity

C) simplicity

D) size

✓ **E) a, b, and c**

Chapter 12

1-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 interruptible

B) allow interaction to be undoable

C) hide technical internals from casual users

✓ **D) only provide one defined method for accomplishing a task**

2-Which of the following interface design principles reduces the user's memory load?

A) define intuitive shortcuts

B) disclose information in a progressive fashion

C) establish meaningful defaults

D) provide an on-line tutorial

✓ **E) answers a, b and c**

3-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**

4-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) both b and d**

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

✓ **A) True**

B) False

6-Which model depicts the profile of the end users of a computer system?

A) design model

B) implementation model

✓ **C) user model**

D) user's model

7-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**

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

✓ **A) Implementation model**

B) user model

C) user's model

D) system perception

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

✓ **A) cost estimation**

B) interface construction

C) interface validation

10-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) both c and d

11-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

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

A) True

✓ B) False

13-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

14-One means of defining user interface objects and actions is to conduct a grammatical parse of the user scenario.

✓ A) True

B) False

15-Interface design patterns typically include a complete component-level design (design classes, attributes, operations, and interfaces).

✓ A) True

B) False

16-Several common design issues surface for almost every user interface including

A) adaptive user profiles

B) error handling resolution of graphics

C) displays system

D) response time

✓ E) both b and d

17-Add-on help facilities are almost always better received by users than integrated help facilities.

A) True

✓ B) False

18-User interface development systems typically provide several mechanisms for building interface prototypes including

A) code generation

B) drawing tools

C) input validation

D) windows handlers

✓ E) both c and d

19-Usability questionnaires are most meaningful to the interface designers when completed by

A) customers

B) experienced programmers

✓ C) product users

D) project managers

20-Several usability measures can be collected while observing users interacting with a computer system including

A) down time for the application

B) number of user errors

C) software reliability

D) time spent looking at help materials

✓ E) both b and d

Chapter 13

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

A) True

✓ B) False

2-The best reason for using Independent software test teams is that

A) software developers do not need to do any testing

✓ B) a test team will test the software more thoroughly

C) testers do not get involved with the project until testing begins

D) arguments between developers and testers are reduced

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

a. integration testing
b. system testing
c. unit testing
d.validation testing

A) a, d, c, b
B) b, d, a, c
✓ **C) c, a, d, b**
D) d, b, c, a

4-Class testing of object-oriented software is equivalent to unit testing for traditional software.

✓ **A) True**
B) False

5-By collecting software metrics and making use of existing software reliability models it is possible to develop meaningful guidelines for determining when software testing is finished.

✓ **A) True**
B) False

6-Which of the following strategic issues needs to be addressed in a successful software testing process?

A) conduct formal technical reviews prior to testing
B) specify requirements in a quantifiable manner
C) use independent test teams
D) wait till code is written prior to writing the test plan
✓ **E) both a and b**

7-Which of the following need to be assessed during unit testing?

A) algorithmic performance
B) code stability
C) error handling
D) execution paths
✓ **E) both c and d**

8-Drivers and stubs are not needed for unit testing because the modules are tested independently of one another.

A) True
✓ **B) False**

9-Top-down integration testing has as it's major advantage(s) that

A) low level modules never need testing
B) major decision points are tested early
C) no drivers need to be written
D) no stubs need to be written
✓ **E) both b and c**

10-Bottom-up integration testing has as it's major advantage(s) that

A) major decision points are tested early
B) no drivers need to be written
✓ **C) no stubs need to be written**
D) regression testing is not required

11-Regression testing should be a normal part of integration testing because as a new module is added to the system new

A) control logic is invoked
B) data flow paths are established
C) drivers require testing
D) all of the above
✓ E) both a and b

12-Smoke testing might best be described as

A) bulletproofing shrink-wrapped software
✓ B) rolling integration testing
C) testing that hides implementation errors
D) unit testing for small programs

13-When testing object-oriented software it is important to test each class operation separately as part of the unit testing process.

A) True
✓ B) False

14-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) operator programs derived from use-case scenarios
D) none of the above

15-The focus of validation testing is to uncover places that a user will be able to observe failure of the software to conform to its requirements.

✓ A) True
B) False

16-Software validation is achieved through a series of tests performed by the user once the software is deployed in his or her work environment.

A) True
✓ B) False

17-Configuration reviews are not needed if regression testing has been rigorously applied during software integration.

A) True
✓ B) False

18-Acceptance tests are normally conducted by the

A) developer
✓ B) end users
C) test team
D) systems engineers

19-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

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

✓ A) True
B) False

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

A) True
✓ B) False

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

A) True
✓ B) False

23-Debugging is not testing, but always occurs as a consequence of testing.

✓ A) True
B) False

Chapter 15

1-Conformance to implicit requirements and customer expectations has no place in modern software quality assurance work.

A) True
✓ B) False

2-Which of the following is not one of three software product aspects addressed by McCall's software quality factors?

A) ability to undergo change
B) adaptability to new environments
C) operational characteristics
✓ D) production costs and scheduling

3-The ISO 9126 quality standards for computer software are useful because they lend themselves to direct measurement of software attributes.

A) True
✓ B) False

4-Most technical software metrics described in this chapter represent indirect measures of software attributes that are useful in the quantitative assessment of software quality.

✓ A) True
B) False

5-Which of these are reasons for using technical product measures during software development?

A) large body of scientific evidence supports their use
✓ B) provides software engineers with an objective mechanism for assessing software quality
C) they allow all software quality information to be expressed unambiguously as a single number
D) all of the above

6- Which measurement activity is missing from the list below?Formulation
Collection
Analysis
Interpretation

A) design
✓ B) feedback

7-The Goal/Question/Metric (GQM) paradigm was developed as a technique for assigning blame for software failures.

A) True
✓ B) False

8-One of the most important attributes for a software product metric is that it should be

✓ **A) easy to compute**
B) qualitative in nature
C) reliable over time
D) widely applicable

9-In many cases metrics for one model may be used in later software engineering activities (e.g., design metrics may be used in test planning).

✓ **A) True**
B) False

10-The function point metric is an example of metric that can be used to assist with technical decision-making based on the analysis model information, without making use of historical project data.

A) True
✓ **B) False**

11-The specification metrics proposed by Davis address which two characteristics of the software requirements?

A) functionality and performance
B) performance and completeness
✓ **C) specificity and completeness**
D) specificity and functionality

12-Architectural design metrics focus on

A) architectural structure
B) data structural relationships
C) internal module complexity
D) module effectiveness
✓ **E) both a and d**

13-Which of the following is not a measurable characteristic of an object-oriented design?

A) completeness
✓ **B) efficiency**
C) size
D) volatility

14-The depth of inheritance tree (DIT) metric can give an OO software designer a reading on the

A) attributes required for each class
✓ **B) completion time required for system implementation**
C) complexity of the class hierarchy
D) level of object reusability achieved

15-Because the class is the dominant unit in OO systems, relatively few metrics have been proposed for operations that reside within a class.

✓ **A) True**
B) False

16-Interface metrics are used to assess the complexity of the module's input and output relationships with external devices.

A) True
✓ B) False

17-Halstead's source code metrics are based on the number of

A) modules in the program
B) operands in the program
C) operators in the program
D) volume elements in the program
✓ E) both b and c

18-Most testing metrics actually focus on the process of testing rather than the technical characteristics of the tests themselves.

✓ A) True
B) False

19-Testing effort can also be estimated using metrics derived from cylcomatic complexity.

✓ A) True
B) False

20-Software testing metrics fall into two broad categories

A) metrics that focus on defect removal effectiveness
B) metrics that focus on test coverage
C) metrics that estimate the duration of the testing process
D) metrics that predict the number of test cases required
✓ E) both b and d

21-The IEEE software maturity index is used to provide a measure of the

A) maintainability of a software product based on its availability
B) relative age of a software product being considered for retirement
C) reliability of a software product following regression testing
✓ D) stability of a software product as it is modified during maintenance

Chapter 17

1-Formulation and requirements gathering are distinct and different processes during WebE.

A) True
✓ B) False

2-Which of these is not one of the formulation questions asked during Web engineering?

A) What are the objectives for the WebApp?
B) What is the business need for the WebApp?
C) Who will use the WebApp?
✓ D) Will you need to outsource development of the WebApp?

3-Which of these are goals for WebE requirements gathering?

A) Define user interaction scenarios

B) Determine performance constraints

C) Identify content requirements

D) Identify WebApp development tools

✓ E) a, b, and c

4-During requirements gathering Web engineers should attempt to define the smallest reasonable number of user classes.

✓ A) True

B) False

5-Which type of analysis is not conducted during the WebE process?

A) content analysis

B) functional analysis

C) user interaction analysis

✓ D) market analysis

6-One of the things that distinguish the development of WebApps from other software products is the need to combine the work products from both technical and non-technical tasks into a single product.

✓ A) True

B) False

7-Which of these roles is not usually assigned to members of the WebE team?

A) content developer

✓ B) marketing specialist

C) Web master

8-In building a WebE team strong team leadership is essential.

✓ A) True

B) False

9-Once formulation is complete Web engineering

A) is complete.

B) may be performed in-house.

C) may be outsourced.

✓ D) both b or c

10-Outsourcing WebApps is common practice, it is important to perform thorough analysis of the application and even create a rough design internally before selecting a vendor.

✓ A) True

B) False

11-Developing WebApps in-house is no different than developing any other piece of software.

A) True

✓ B) False

12-Which of these is not a goal for using metrics in WebE?

A) to provide basis for effort estimation

✓ B) to provide basis for making personnel decisions

C) to provide indication of business success

D) to provide indication of technical quality

13-Which of these is not a category for WebE effort metrics?

A) application authoring

B) media authoring

C) page authoring

✓ D) scenario authoring

14-Business people lag considerably behind Web engineers in developing, collecting, and using metrics for WebApps.

A) True

✓ B) False

15-WebApps need to be built with such urgency that planning is not possible.

A) True

✓ B) False

16-WebApps are extremely volatile, but this does not eliminate the need to understand the WebApp requirements.

✓ A) True

B) False

17-Any team of experienced software engineers can develop WebApps.

A) True

✓ B) False

18-WebApps involve so little programming that formal testing is not needed before releasing the product to the users.

A) True

✓ B) False

Chapter 19

1-Which of the following characteristics should not be used to assess the quality of a WebApp?

✓ A) aesthetics

B) reliability

C) maintainability

D) usability

2-Which of the following are design goals for every WebApp?

A) Simplicity

B) Consistency

C) Navigability

D) Visual appeal

✓ E) all of the above

3-Which of the following are not part of the design pyramid for WebE design?

A) Architectural design

✓ B) Business case design

C) Content design

D) Navigation design

4-Every WebApp user interface should be easy to use, easy to navigate, error-free and functional.

✓ A) True

B) False

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

A) True

✓ B) False

6-Which of these are WebApp interaction mechanisms?

A) Graphic icons

B) Graphic images

C) Navigation menus

✓ D) All of the above

7-UML does not have any representation schemas that are useful in building WebApp design models.

A) True

✓ B) False

8-Screen layout design has several widely accepted standards based on human factors research.

A) True

✓ B) False

9-Graphic design considers every aspect of the look and feel of a WebApp.

✓ A) True

B) False

10-Content design is conducted by

A) Copywriters and graphic designer

B) Web engineers

✓ C) both a and b

D) none of the above

11-Content objects have both information attributes defined during analysis and implementation specific attributes specified during design.

✓ A) True

B) False

12-Content objects are not normally chunked into Web pages until the implementation activities begin.

A) True

✓ B) False

13-Which of the following is not one of the browsing primitives normally found in WebApp interfaces.

A) Conditional browsing

B) Nested browsing

✓ C) Recursive browsing

D) Sequential browsing

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

A) True

✓ B) False

15-Which of the following is not one of the content architectural structures used by web engineers?

A) linear

B) grid

C) hierarchical

✓ D) parallel

16-MVC is a three layer architecture that contains a

A) machine, view, content objects

B) model, view, and content objects

✓ C) model, view, and controller

D) machine, view, controller

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

✓ A) True

B) False

18-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

19-Component level design for WebApps is very similar to component level design for other software delivery environments.

✓ A) True

B) False

20-Which of the following is a navigation pattern used during web-based design?

A) cycle

B) counterpoint

C) sieve

✓ D) all of the above

21-Which of these is not one of the design activities associated with object-oriented hypermedia design?

A) abstract interface design

B) conceptual design

✓ C) content design

D) navigational design

22-Most WebApps can be easily characterized by judicious use of widely recognized suites of software metrics?

A) True

✓ B) False

Chapter 22

1-Which of these are valid reasons for measuring software processes, products, and resources?

A) to characterize them

B) to evaluate them

C) to price them

D) to improve them

✓ E) a, b, and d

2-The terms *measure, measurement,* and *metric* all share the same definition according to the IEEE Standard Glossary of Software Engineering Terms.

A) True

✓ B) False

3-Process indicators enable a software project manager to

A) assess the status of an on-going project

B) track potential risks

C) adjust work flow or tasks

✓ D) all of the above

4-Public metrics are used يتم استخدام المقاييس العامة لـ

A) to evaluate the performance of software development teams.

B) to appraise the performance of individual team members.

C) to make strategic changes to the software process.

D) to make tactical changes during a software project لإجراء تغييرات تكتيكية خلال مشروع البرمجيات

✓ E) both c and d

5-Which of the following items are not measured by software project metrics?

A) inputs

✓ B) markets

C) outputs

D) results

6-Software quality and functionality must be measured indirectly يجب قياس جودة البرمجيات والوظائف بشكل غير مباشر..

✓ A) True

B) False

7-Which of following are advantages of using LOC (lines of code) as a size-oriented metric?

✓ A) LOC is easily computed.

B) LOC is a language dependent measure.

C) LOC is a language independent measure.

D) LOC can be computed before a design is completed.

8-Which of the following are advantages of using function points (FP) as a measure of the functionality delivered by a software application?

A) FP is easily computed.

B) FP is a language dependent measure.

C) FP is a language independent measure.

✓ E) both c and d

9-There is no need to reconcile LOC(line of code) and FP(function points) measures since each is meaningful in its own right as a project measure. ليست هناك حاجة للتوفيق بين لوك (خط التعليمات البرمجية) و فب (نقاط الدالة) التدابير لأن كل منها ذات مغزى في حد ذاتها كتدبير مشروع

A) True

✓ B) False

10-Object-oriented project measures may be combined with historical project data to provide metrics that aid in project estimation. ويمكن الجمع بين تدابير المشاريع الموجهة نحو الكائن وبيانات المشروع التاريخية لتوفير المقاييس التي تساعد في تقدير المشروع.

✓ A) True

B) False

11-Use-case oriented metrics are computed directly from UML diagrams and they are often used as normalization measures.

A) True

✓ B) False

12-Which of the following is not a measure that can be collected from a Web application project?

✓ A) Customization index فهرس التخصيص

B) Number of dynamic objects

C) Number of internal page links

D) Number of static web pages

13-Which of the following software quality factors is most likely to be affected by radical changes to computing architectures?

A) operation

B) transition

C) revision

✓ D) none of the above

14-Which of the following provide useful measures of software quality?

A) correctness, business relevance, integrity, usability

B) reliability, maintainability, integrity, sales

C) correctness, maintainability, size, satisfaction

✓ D) correctness, maintainability, integrity, usability

15-A software quality metric that can be used at both the process and project levels is defect removal efficiency كفاءة إزالة العيب (DRE).

✓ A) True

B) False

16-Why is it important to measure the process of software engineering and software it produces?

A) It is really not necessary unless the project is extremely complex.

B) To determine costs and allow a profit margin to be set.

✓ C) To determine whether a software group is improving or not.

D) To make software engineering more like other engineering processes.

17-To be an effective aid in process improvement the baseline data used must be:

A) based on reasonable guestimates from past projects

B) measured consistently across projects

C) drawn from similar projects

D) based only on successful projects

✓ E) both b and c

18-Baseline data must be collected in an on-going manner and cannot be computed by formal study of historical project data.

A) True

✓ B) False

19-Small software organizations are not likely to see any economic return from establishing software metrics program.

A) True

✓ B) False

20-The software metrics chosen by an organization are driven by the business or technical goals an organization wishes to accomplish.

✓ A) True

B) False

Chapter 25

1-Which of these are valid reasons for measuring software processes, products, and resources?

A) to characterize them

B) to evaluate them

C) to price them

D) to improve them

✓ E) a, b, and d

2-The terms *measure, measurement,* and *metric* all share the same definition according to the IEEE Standard Glossary of Software Engineering Terms.

A) True

✓ B) False

3-Process indicators enable a software project manager to

A) assess the status of an on-going project

B) track potential risks

C) adjust work flow or tasks

✓ D) all of the above

4-Public metrics are used

A) to evaluate the performance of software development teams.

B) to appraise the performance of individual team members.

C) to make strategic changes to the software process.

D) to make tactical changes during a software project

✓ E) both c and d

5-Which of the following items are not measured by software project metrics?

A) inputs

✓ B) markets

C) outputs

6-Software quality and functionality must be measured indirectly.

✓ A) True

B) False

7-Which of following are advantages of using LOC (lines of code) as a size-oriented metric?

✓ A) LOC is easily computed.

B) LOC is a language dependent measure.

C) LOC is a language independent measure.

D) LOC can be computed before a design is completed.

8-Which of the following are advantages of using function points (FP) as a measure of the functionality delivered by a software application?

A) FP is easily computed.

B) FP is a language dependent measure.

C) FP is a language independent measure.

D) FP can be computed before a design is completed.

✓ E) both c and d

9-There is no need to reconcile LOC and FP measures since each is meaningful in its own right as a project measure.

A) True

✓ B) False

10-Object-oriented project measures may be combined with historical project data to provide metrics that aid in project estimation.

✓ A) True

B) False

11-Use-case oriented metrics are computed directly from UML diagrams and they are often used as normalization measures.

A) True

✓ B) False

12-Which of the following is not a measure that can be collected from a Web application project?

✓ A) Customization index

B) Number of dynamic objects

13-Which of the following software quality factors is most likely to be affected by radical changes to computing architectures?

A) operation

B) transition

C) revision

✓ D) none of the above

14-Which of the following provide useful measures of software quality?

A) correctness, business relevance, integrity, usability

B) reliability, maintainability, integrity, sales

C) correctness, maintainability, size, satisfaction

✓ D) correctness, maintainability, integrity, usability

15-A software quality metric that can be used at both the process and project levels is defect removal efficiency (DRE).

✓ A) True

B) False

16-Why is it important to measure the process of software engineering and software it produces?

A) It is really not necessary unless the project is extremely complex.

B) To determine costs and allow a profit margin to be set.

✓ C) To determine whether a software group is improving or not.

D) To make software engineering more like other engineering processes.

17-To be an effective aid in process improvement the baseline data used must be:

A) based on reasonable guestimates from past projects

B) measured consistently across projects

C) drawn from similar projects

D) based only on successful projects

✓ E) both b and c

18-Baseline data must be collected in an on-going manner and cannot be computed by formal study of historical project data.

A) True

✓ B) False

19-Small software organizations are not likely to see any economic return from establishing software metrics program.

A) True

✓ B) False

20-The software metrics chosen by an organization are driven by the business or technical goals an organization wishes to accomplish.

✓ A) True

B) False

Chpter 36

1. How much effort is typically expended by a software organization on software maintenance?

a) 20 percent

b) 40 percent

✓ c) 60 percent

d) 80 percent

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

a) True

✓ b) False

3. Business process reengineering is often accompanied by software reengineering.

✓ a) True

b) False

4. 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

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

✓ a) True

b) False

6. 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

7. 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

8. Which of the following is not an issue to consider when reverse engineering?

a) Abstraction level

b) Completeness

✓ c) Connectivity

9. Reverse engineering of data focuses on

a) Database structures

b) Internal data structures

✓ c) Both a and b

d) None of the above

10. The first reverse engineering activity involves seeking to understand

a) Data

✓ b) Processing

c) User interfaces

d) None of the above

11. Reverse engineering should proceed the reengineering of any user interface.

✓ a) True

b) False

12. 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

13. Code restructuring is a good example of software reengineering.

a) True

✓ b) False

14. Which of these is not an example of data restructuring?

✓ a) Data analysis

b) Data name rationalization

c) Data record standardization

d) None of the above

15. Forward engineering is not necessary if an existing software product is producing the correct output.

a) True

✓ b) False

16. Reengineering client/server systems begins with a thorough analysis of the business environment that encompasses the existing computing system.

✓ a) True

b) False

17. The only time reengineering enters into work with a legacy system is when it components will be implemented as objects.

a) True

✓ b) False

18. 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

The End

أسأل الله لي ولكم التوفيق