First mid Term Exam – March 2018

Class: 4 IT Subject: Software Engineering

Time: 1 Hour Marks: 10

Attend any two questions. All question carry equal marks.

1. What is software engineering? Explain the characteristics of software.

Answer: Software engineering is the application of principles used in the field of engineering, which usually deals with physical systems, to the design, development, testing, deployment and management of software systems.

Software engineering is typically used for large and intricate software systems rather than single applications or programs. Development, however, is simply one phase of the process. While a software engineer is typically responsible for the design of systems, programmers are often responsible for coding its implementation.

Software engineering involves a number of fields that cover the process of engineering software and certification including: requirements gathering, software design, software construction, software maintenance, software configuration management, software engineering mangement, software development process management and creation, software engineering models and methods, software quality, software engineering professional practices as well as foundational computing and mathematical and engineering study.

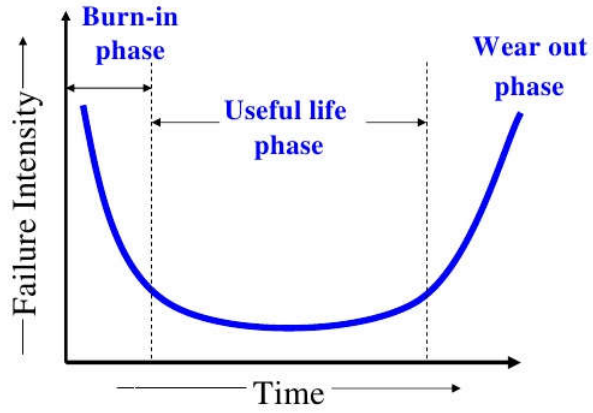
Characteristics of software:

• Software is developed or engineered, it is not manufactured in the classical sense which has quality problem.

• Software doesn't "wear out.” but it deteriorates (due to change). Hardware has bathtub curve of failure rate (high failure rate in the beginning, then drop to steady state, then cumulative effects of dust, vibration, abuse occurs).

Software Characteristics:

✓Software does not wear out.



• Although the industry is moving toward component-based construction (e.g. standard screws and off-the-shelf integrated circuits), most software continues to be custom-built. Modern reusable components encapsulate data and processing into software parts to be reused by different programs. E.g. graphical user interface, window, pull-down menus in library

2. Explain the Rapid Application Development (RAD) model of software development life cycle.

Answer: RAD model is Rapid Application Development model. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

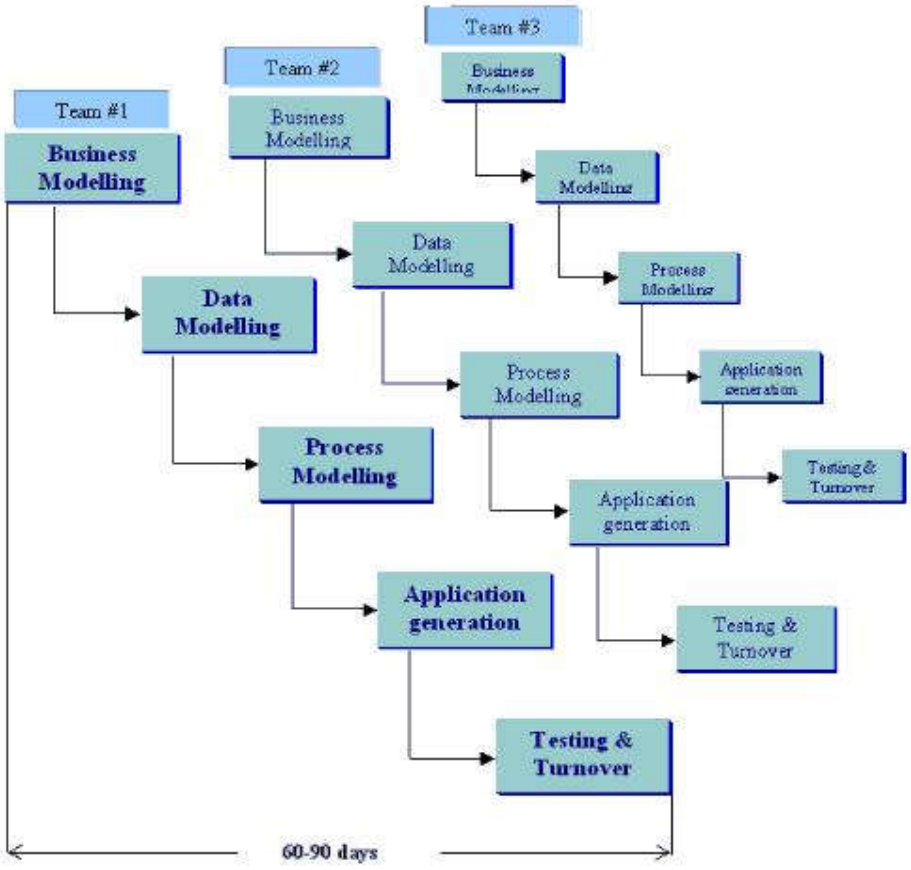


Figure 1.5 – RAD Model

The phases in the rapid application development (RAD) model are:

A. **Business modeling:** The information flow is identified between various business functions.

B. **Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.

C. **Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.

D. **Application generation:** Automated tools are used to convert process models into code and the actual system.

E. **Testing and turnover:** Test new components and all the interfaces.

Advantages of the RAD model:

- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

Disadvantages of RAD model:

- Depends on strong team and individual performances for identifying business requirements.
- Only system that can be modularized can be built using RAD
- Requires highly skilled developers/designers.
- High dependency on modeling skills
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

When to use RAD model:

- RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
- It should be used if there's high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
- RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

3. Write short notes on –

   a) Data Dictionary

   Answer: The data dictionary is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definitions so that both user and system analyst will have a common understanding of inputs, outputs, components of stores and intermediate calculations.

Today, the data dictionary is always implemented as part of a CASE "structured analysis and design tool."

Although the format of dictionaries varies from tool to tool, most contain the following information:

• Name—the primary name of the data or control item, the data store or an external entity.

• Alias—other names used for the first entry.

• Where-used/how-used—a listing of the processes that use the data or control item and how it is used (e.g., input to the process, output from the process, as a store, as an external entity.

• Content description—a notation for representing content

• Supplementary information—other information about data types, preset values (if known), restrictions or limitations, and so forth. Once a data object or control item name and its aliases are entered

The notation used to develop a content description is noted in the following table:

Data	Construct Notation	Meaning
	=	is composed of
Sequence	+	and
Selection	[|]	either-or
Repetition	{ }n	n repetitions of
	( )	optional data
	* ... *	delimits comments

For example consider the data dictionary of telephone number. The data dictionary entry begins as follows:

name: telephone number

aliases: none

where used/how used: assess against set-up (output)

dial phone (input)

description:

telephone number = [local number|long distance number]

local number = prefix + access number

long distance number = 1 + area code + local number

area code = [800 | 888 | 561]

prefix = *a three digit number that never starts with 0 or 1*

access number = * any four number string *

b) Entity – Relationship Diagram

Answer :An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

Three main components of an ERD are the entities, which are objects or concepts that can have data stored about them, the relationship between those entities, and the cardinality, which defines that relationship in terms of numbers.

For example, an ER diagram representing the information system for a company's sales department might start with graphical representations of entities such as the sales representative, the customer, the customer's address, the customer's order, the product and the warehouse. (See diagram) Then lines or other symbols can be used to represent the relationship between entities, and text can be used to label the relationships.

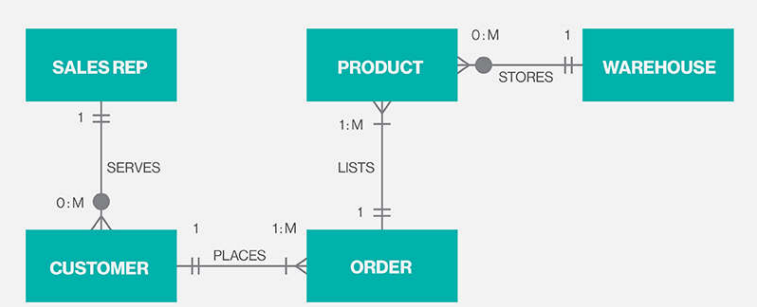


Figure 1. Entity-Relationship Diagram

*1 INSTANCE OF A SALES REP SERVES 1 TO MANY CUSTOMERS
*1 INSTANCE OF A CUSTOMER PLACES 1 TO MANY ORDERS
*1 INSTANCE OF AN ORDER LISTS 1 TO MANY PRODUCTS
*1 INSTANCE OF A WAREHOUSE STORES 0 TO MANY PRODUCTS