



SRI KRISHNA ARTS & SCIENCE COLLEGE

Coimbatore - 641 008

Department of Information Technology & Cognitive Systems



22ITU10A - Software Engineering and UML

Class	II B.Sc.IT A
Google Classroom code	ylrlutw
Topics Covered	The Nature of Software

K.Sutha

Department of Information Technology & Cognitive Systems

Online Course – Programming Basics

Course content

4 sections • 13 lectures • 1h 59m total length

▼ Introduction

▼ Selenium Automation Tool - Basics

▼ Locators and Identifiers | Working with Web Elements | Selenium WebDriver

▼ Waits in Selenium WebDriver and Mouse Operations

- ❖ In this course, you will learn the basics of Selenium automation tool - setup, running code on chrome browser
- ❖ Identifying web elements on a web page using locators and identifiers in Selenium WebDriver
- ❖ Synchronizing your application under test with Selenium code using waits in Selenium WebDriver
- [Click this link to take up the course.](#)



Course Outcome

- **CLO 1-**Describes about software engineering, process models, analysis and design modelling, agility and SCRUM, frameworks and Unified Modeling Language (UML)
- **CLO 2-** Apply the framework for agility
- **CLO 3-** Adopt the concept of UML diagrams to a sample project
- **CLO 4-** Practice the concepts of selenium packages in Java to automate web browser.

ATTENDANCE



SNAP TALK



Topics to be Covered

- Software Engineering-Introduction
- Nature of Software
- Defining Software
- Software Application Domains

Software Engineering- Introduction

- Software Engineering is defined as systematic, disciplined and quantifiable approach for the development, operation and maintenance of software
- It is a systematic and cost-effective techniques to software development.
- It can be viewed as the engineering approach to develop software.

The Nature of Software Engineering

Dual role of Software

- Software is a product.
- Software is a vehicle for delivering a product.
- Software is an information transformer—producing, managing, acquiring, modifying, displaying, or transmitting information that can be as simple as a single bit or as complex as a multimedia presentation derived from data acquired from dozens of independent sources

Nature of Software (Cont'd)

As the vehicle used to deliver the product, software acts as

- the basis for the control of the computer (operating systems),
- the communication of information (networks),
- The creation and control of other programs (software tools and environments).

Software **delivers the most important product of our time**—information.

- It **transforms personal data** (e.g., an individual's financial transactions) so that the data can be more useful in a local context;
- It **manages business information** to enhance competitiveness;
- It **provides a gateway to worldwide information networks** (e.g., the Internet)
- And **provides the means for acquiring information** in all of its forms.

Defining Software

Software is:

- Instructions (computer programs) that when executed provide desired features, function, and performance;
- Data structures that enable the programs to adequately manipulate information, and
- Descriptive information in both hard copy and virtual forms that describes the operation and use of the programs.

Software is a logical rather than a physical system element.

Defining Software

Software has characteristics that are considerably different than those of hardware:

- Software is developed or engineered; it is not manufactured in the classical sense.
- Software doesn't wear out, but it deteriorates (due to change).
- Although the industry is moving toward component-based construction, most software continues to be custom built.

Software products

Generic products

- Stand-alone systems that are marketed and sold to any customer who wishes to buy them.

Examples – PC software such as editing, graphics programs, project management tools; CAD software;

Customized products

- Software that is commissioned by a specific customer to meet their own needs.

Examples – embedded control systems, air traffic control software, traffic monitoring systems

Hardware vs Software

Hardware

- Manufactured
- Wears out
- Built using components
- Relatively simple

Software

- Developed / engineered
- Deteriorates
- Custom built
- Complex

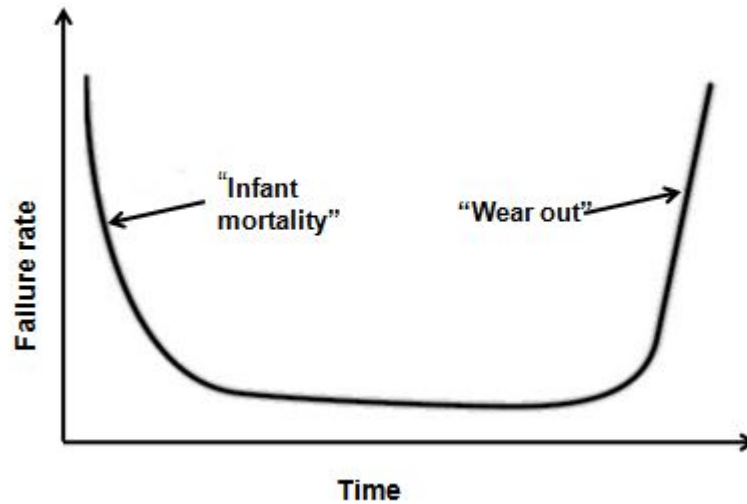
Characteristics of software

- Software is developed or engineered, it is not manufactured in the classical sense.
- Software doesn't "wear out." but it deteriorates (due to change).
- Software is custom built rather than assembling existing components.

Manufacturing vs. Development

- Once a hardware product has been manufactured, it is difficult or impossible to modify.
- In contrast, software products are routinely modified and upgraded.
- In hardware, hiring more people allows to accomplish more work, but the same does not necessarily hold true in software engineering.
- Unlike hardware, software costs are concentrated in design rather than production.

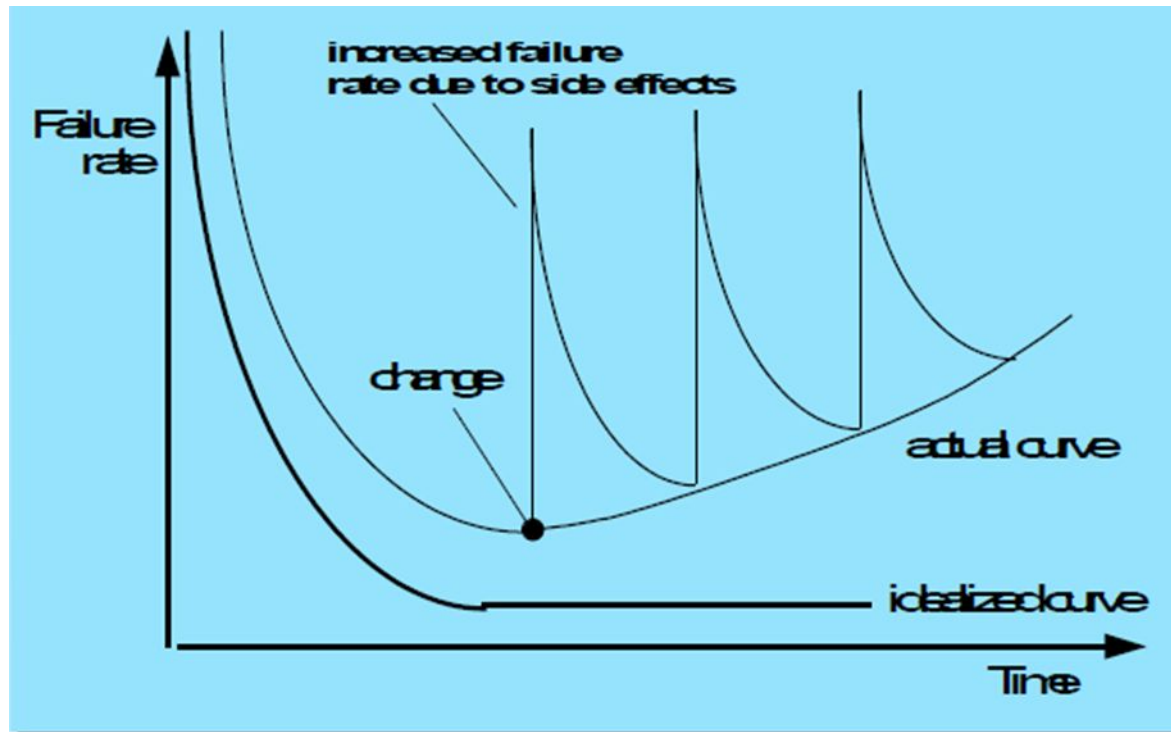
Wear out vs. Deterioration



Hardware wears out over time

Software Failure Curve

- 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 deteriorates out over time : Failure Curves for Software

Component Based vs. Custom Built

- Hardware products typically employ many standardized design components.
- Most software continues to be custom built.
- The software industry seems to be moving toward component-based construction.
- A software component should be designed and implemented so that it can be reused in many different programs.
- Modern reusable components encapsulate both data and the processing that is applied to the data, enabling the software engineer to create new applications from reusable parts.

Example: The data structures and processing detail required to build the interface are contained within a library of reusable components for interface construction.

Software Application Domains

The seven broad categories of computer software are

1. System software
2. Application software
3. Engineering/scientific software
4. Embedded software
5. Product-line software
6. Web applications
7. Artificial intelligence software

System Software

- A collection of programs written to service other programs.

Eg: **compilers, editors, and file management utilities.**

- Coordinates the activities & functions of hardware and programs.
- Control operations of computer hardware

The systems software area is characterized by

- Heavy interaction with computer hardware,
- Heavy usage by multiple users,
- Concurrent operation that requires scheduling, resource sharing, and sophisticated process management,
- Complex data structures and multiple external interfaces.
- Coordinates the activities & functions of hardware and programs.
- Control operations of computer hardware.

Application Software

- Stand-alone programs that solve a specific business need.
- Applications in this area process business or technical data that facilitates business operations or management/technical decision making.
- Application software is used to control business functions in real-time.

Example: point-of-sale transaction processing, real-time manufacturing process control.

- In most cases, Application software resides on computer hardware.

Engineering/Scientific software

- Characterized by “Number crunching” algorithms.

Example: Astronomy, Volcanology, automotive stress analysis, molecular biology, orbital dynamics, software used to predict weather etc.

- Computer-aided design, system simulation, and other interactive applications have begun to take on real-time and even system software characteristics.

Number Crunching Algorithms

- Number crunching is the work (typically arithmetic) involved in bringing a numerical perspective to a situation.
- Business people often will say that someone will "crunch the numbers" or "run the numbers" as a reference to someone going through the facts (and quantifying them) in detail
- A financial analyst evaluates a new initiative in detail, works through the various assumptions that sales, marketing, manufacturing are using and creates a model to evaluate the multi-year financial payout.

Embedded Software

- Software resides within a product or system.
- It is used to implement and control features and functions for the end user and for the system itself.
- Embedded software can perform limited and esoteric functions (e.g., key pad control for a microwave oven) or provide significant function and control capability (e.g., digital functions in an automobile such as fuel control, dashboard displays, and braking systems).

Product-line Software

- To provide specific capabilities for customers.
- Software focus on a limited marketplace
- **Example:** inventory control products or address mass consumer market.
- **Example:** word processing, graphics, multimedia, database management personal & business financial applications.

Web Applications

- It is called “WebApps” this network-centric software category spans a wide array of applications.
- WebApps can be little more than a set of linked hypertext files that present information using text and limited graphics.
- Web 2.0 emerges, WebApps are evolving into sophisticated computing environments that not only provide stand-alone features, computing functions, and content to the end user, but also are integrated with corporate databases and business applications.

Software - New Categories

Open world computing

- The rapid growth of wireless networking may soon lead to true pervasive, distributed computing.
- The challenge for software engineers will be to develop systems and application software that will allow mobile devices, personal computers, and enterprise systems to communicate across vast networks.

Software - New Categories

Netsourcing

- The World Wide Web is rapidly becoming a computing engine as well as content provider.
- The challenge for software engineers is to architect simple and sophisticated applications to target end-users markets worldwide.

Software - New Categories

Open Source

- A growing trend that results in the distribution of source code for systems applications.
- The challenge for software engineers is to build source code, to develop techniques that enable both customers and developers to know what changes have been made and to be manifest themselves within the software.

Legacy Software

Why must it change?

- Software must be adapted to meet the needs of new computing environments or technology.
- Software must be enhanced to implement new business requirements.
- Software must be extended to make it interoperable with other more modern systems or databases.
- Software must be re-architected to make it viable within a network environment.

Summary

- Software is a product and it is a vehicle for delivering product
- Software is a computer programs , manipulate information and describe the use of programs.
- Software doesn't wear out
- System software collection of programs written to service other programs
- Application software stand-alone programs for specific needs
- Engineering/scientific software Characterized by “Number crunching” algorithms
- Embedded software non-numerical algorithm to solve a complex problem
- Product-line software
- Web applications
- Artificial Intelligence software
- Software - New Categories



MCQ

1.What is Software ?

- a)Set of computer programs, procedures and possibly associated document concerned with the operation of data processing.
- b)A set of compiler instructions
- c)A mathematical formula
- d)None of above

MCQ

2) Which of the following is not the characteristic of software ?

- a) Software does not wear out
- b) Software is flexible
- c) Software is not manufactured
- d) Software is always correct

MCQ

3) Software engineering aims at developing ?

- a) Reliable Software
- b) Cost Effective Software
- c) Reliable and cost effective Software
- d) None of Above

MCQ

4) _____ that enable the programs to adequately manipulate information?

- a) Documents
- b) Data structures
- c) Instructions
- d) Both a & b

MCQ

**5) The software industry seem to be moving towards
_____ construction?**

- a) Design based
- b) Component based
- c) Custom based
- d) Hierarchy based

MCQ

6)The _____ is a collection of programs written to service other programs.

- a) System software
- b) Application software
- c) Engineering/scientific software
- d) Embedded software

MCQ

7. Application software is a _____

- a) Collection of programs
- b) “number crunching” algorithms
- c) Stand-alone programs
- d) WebApps

MCQ

8. _____ includes the applications such as robotics, expert systems and pattern recognition.

- a) Application software
- b) Engineering/scientific software
- c) Artificial intelligence software
- d) Web applications

MCQ

9. _____ can focus on a limited an esoteric marketplace or address and esoteric marketplace?

- a) System software
- b) Web application
- c) Product-line software
- d) Embedded software

MCQ

10. WebApps is also known as_____

- a) Application software
- b) Web Application
- c) Application web
- d) None of the above

Answers

1. a) Set of computer programs, procedures and possibly associated document concerned with the operation of data processing.
2. d) Software is always correct
3. b) Cost Effective Software
4. b) Data structures
5. b) Component-based

Answers

- 6. d) Embedded software
- 7. c) Stand-alone programs
- 8. c) Artificial intelligence software
- 9. d) Embedded software
- 10. b) Web Application

QUERIES?

Thank You!



SRI KRISHNA ARTS & SCIENCE COLLEGE

Coimbatore - 641 008

Department of Information Technology & Cognitive Systems



22ITU10A - Software Engineering and UML

Class	II B.Sc.IT A
Google Classroom code	ylrlutw
Topics Covered	Software Process

K.Sutha

Department of Information Technology & Cognitive Systems

Online Course – Programming Basics

Course content

4 sections • 13 lectures • 1h 59m total length

▼ Introduction

▼ Selenium Automation Tool - Basics

▼ Locators and Identifiers | Working with Web Elements | Selenium WebDriver

▼ Waits in Selenium WebDriver and Mouse Operations

- ❖ In this course, you will learn the basics of Selenium automation tool - setup, running code on chrome browser
- ❖ Identifying web elements on a web page using locators and identifiers in Selenium WebDriver
- ❖ Synchronizing your application under test with Selenium code using waits in Selenium WebDriver
- [Click this link to take up the course.](#)



Course Outcome

- **CLO 1-**Describes about software engineering, process models, analysis and design modelling, agility and SCRUM, frameworks and Unified Modeling Language (UML)
- **CLO 2-** Apply the framework for agility
- **CLO 3-** Adopt the concept of UML diagrams to a sample project
- **CLO 4-** Practice the concepts of selenium packages in Java to automate web browser.

ATTENDANCE



SNAP TALK



Topics to be Covered

- SOFTWARE ENGINEERING

- A Layered Technology

- THE SOFTWARE PROCESS

- A Process Framework

- Framework Activities

- Umbrella Activities

- The Software Myths

Software Engineering

Some realities:

- A concerted effort should be made to understand the problem before a software solution is developed.
(**Understand the problem before build a solution**)
- Design becomes a **pivotal activity**.
- Software should **exhibit high quality**.
- Software should be **maintainable**.

Software Engineering (Conti...)

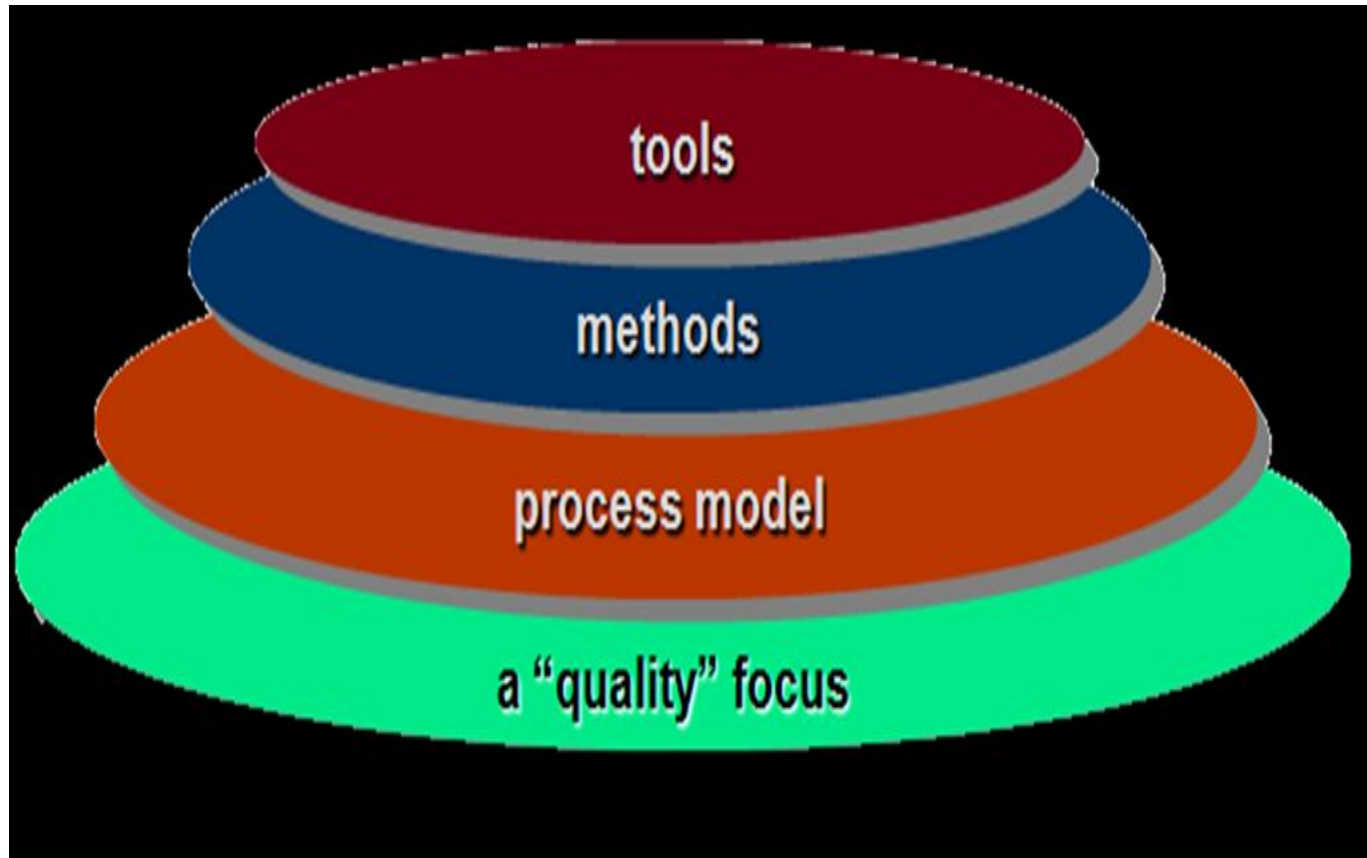
- **The seminal definition:(Fritz Bauer)**

- SE is the establishment and use of **sound engineering principles** in order to obtain **economically** software that is **reliable and works efficiently on real machines.**

- **The IEEE definition:**

- Software Engineering: The application of a **systematic, disciplined, quantifiable approach to the development, operation, and maintenance** of software; that is, the application of engineering to software.

A Layered Technology



Quality and Process

Quality

- Focus on **continuous process improvement** culture.

Process

- The **foundation** for software engineering is the process layer.
- The software engineering process is the glue that **holds the technology layers together** and **enables rational and timely development** of computer software.
- It defines a **framework with activities** for **effective delivery of software engineering technology**.
- The software process forms the **basis for management control of software projects** and establishes the context in which **technical methods** are applied, **work products** (models, documents, data, reports, forms, etc.) are produced, **milestones** are established, **quality** is ensured, and **change is properly managed**.

Methods and Tools

Method

- Provides **technical how-to's for building** software.
- It encompasses many tasks including **communication, requirement analysis, design modeling, program construction, testing and support.**

Tools

- Provide **automated or semi-automated support** for the process and methods.
- When tools are integrated so that information created by one tool can be used by another.
- A system for the support of software development, called **computer-aided software engineering** is established.

The Software Process

- A process is a **collection of activities, actions and tasks** that are performed when some work product is to be created.
- An **activity** – strives to achieve a board objective.
(**Example: Communication with stakeholders**)
- An **action** - a set of tasks that produce major work product (**Example: Architectural design**)
- A **task** - focuses on a small, but well defined objective (**Example: Conducting a unit test**)

A Process Framework

- Establishes the foundation for a **complete software process**
- **Identifies a number of framework activities** applicable to all software projects
- Also include a **set of umbrella activities** that are applicable across the entire software process.

Framework Activities

1) Communication

2) Planning

3) Modeling

- Analysis of requirements

- Design

4) Construction

- Code generation

- Testing

5) Deployment

Five Activities of a Generic Process framework

- **Communication:** communicate with customer to understand objectives and gather requirements
- **Planning:** creates a “map” defines the work by describing the tasks, risks and resources, work products and work schedule.
- **Modeling:** Create a “sketch”, what it looks like architecturally, how the constituent parts fit together and other characteristics.
- **Construction:** code generation and the testing.

Five Activities of a Generic Process framework

- **Deployment:** Delivered to the customer who evaluates the products and provides feedback based on the evaluation.
- These five framework activities can be used to all software development regardless of the application domain, size of the project, complexity of the efforts etc, though the details will be different in each case.
- For many software projects, these framework activities are applied iteratively as a project progresses.
- Each iteration produces a software increment that provides a subset of overall software features and functionality.

Umbrella activities

- Software project tracking and control:
- Risk management
- Software quality assurance
- Technical reviews
- Measurement
- Software configuration management
- Reusability management
- Work product preparation & production

Umbrella activities

Umbrella activities:

- **S/W project tracking and control**(allow s/w team to access project plan and take action to maintain schedule)
- **Risk management**(access risk that may effect outcome or quality)
- **Quality assurance**(define and conduct activities to ensure quality)
- **Technical reviews**(uncover or remove errors before moving to next activity)
- **Measurement**(collects process,project,product measures.)
- **Software configuration management**(manages the effect of changes)
- **Reusability management**(defines criteria for work product reuse)
- **Work product preparation and production**(encompasses activities to create workproduct). Example:models,documents,logs,forms and lists

Adapting a Process Model

- The process should be **agile and adaptable** to problems. Process adopted for one project might be significantly different than a process adopted from another project. (to the problem, the project, the team, organizational culture).
- Among the differences are: the overall flow of activities, actions, and tasks and the interdependencies among them!
- The degree to which actions and tasks are defined within each framework activity!

Adapting a Process Model

- The degree to which work products are identified and required!
- The manner which quality assurance activities are applied!
- The manner in which project tracking and control activities are applied!
- The overall degree of detail and rigor with which the process is described!
- The degree to which the customer and other stakeholders are involved with the project!
- The level of autonomy given to the software team!
- The degree to which team organization and roles are prescribed!



MCQ

1. Which is not Layered Technology?

- a) Process
- b) Quality
- c) Tools
- d) Activity**

2. _____ is the continuous process improvement culture?

- a) Tools
- b) Methods
- c) Quality**
- d) Process

MCQ

3. _____ is a glue that holds the technology layers together

a) software method

b) software process

c) software quality

d) software tool

4. An _____ is a set of tasks that produce major work product ?

a) activity

b) action

c) process

d) task

MCQ

5. A _____ is a collection of activities, actions and tasks that are performed when some work product is to be created?
- a) Quality
 - b) Process**
 - c) Tools
 - d) Method
6. Many causes of the software crisis can be traced to mythology based on?
- a) Management Myths
 - b) Customer Myths
 - c) Practitioner Myths
 - d) All of the above**

MCQ

7. _____ maintain budgets

a) **Management myths**

b) Customer myths

c) Practitioner's myths

d) none

8. Practitioner's are the one who develop _____

a) tools

b) **programs**

c) deployment

d) design

MCQ

9. Project requirements continually change, but change can be easily accommodated because software is

a) not flexible

b) flexible

c) complex

d) difficult

10. Efficiency in a software product does not include

a) responsiveness

b) licensing

c) memory utilization

d) processing time

Next Session

Process model: Waterfall model,
Incremental Process model

Thank You!



SRI KRISHNA ARTS & SCIENCE COLLEGE

Coimbatore - 641 008

Department of Information Technology & Cognitive Systems



22ITU10A - Software Engineering and UML

Class	II B.Sc.IT A
Google Classroom code	ylrlutw
Topics Covered	Software Process Model Waterfall & Incremental Model

K.Sutha

Department of Information Technology & Cognitive Systems

Online Course – Programming Basics

Course content

4 sections • 13 lectures • 1h 59m total length

▼ Introduction

▼ Selenium Automation Tool - Basics

▼ Locators and Identifiers | Working with Web Elements | Selenium WebDriver

▼ Waits in Selenium WebDriver and Mouse Operations

- ❖ In this course, you will learn the basics of Selenium automation tool - setup, running code on chrome browser
- ❖ Identifying web elements on a web page using locators and identifiers in Selenium WebDriver
- ❖ Synchronizing your application under test with Selenium code using waits in Selenium WebDriver
- [Click this link to take up the course.](#)



Course Outcome

- **CLO 1-**Describes about software engineering, process models, analysis and design modelling, agility and SCRUM, frameworks and Unified Modeling Language (UML)
- **CLO 2-** Apply the framework for agility
- **CLO 3-** Adopt the concept of UML diagrams to a sample project
- **CLO 4-** Practice the concepts of selenium packages in Java to automate web browser.

ATTENDANCE



SNAP TALK



Topics to be Covered

- **Waterfall Model**
- **Incremental Process Model**

What is process model?

- A software process model is a simplified representation of a software process.
- Each model represents a process from a specific perspective.
- These generic models are abstractions of the process that can be used to explain different approaches to the software development.

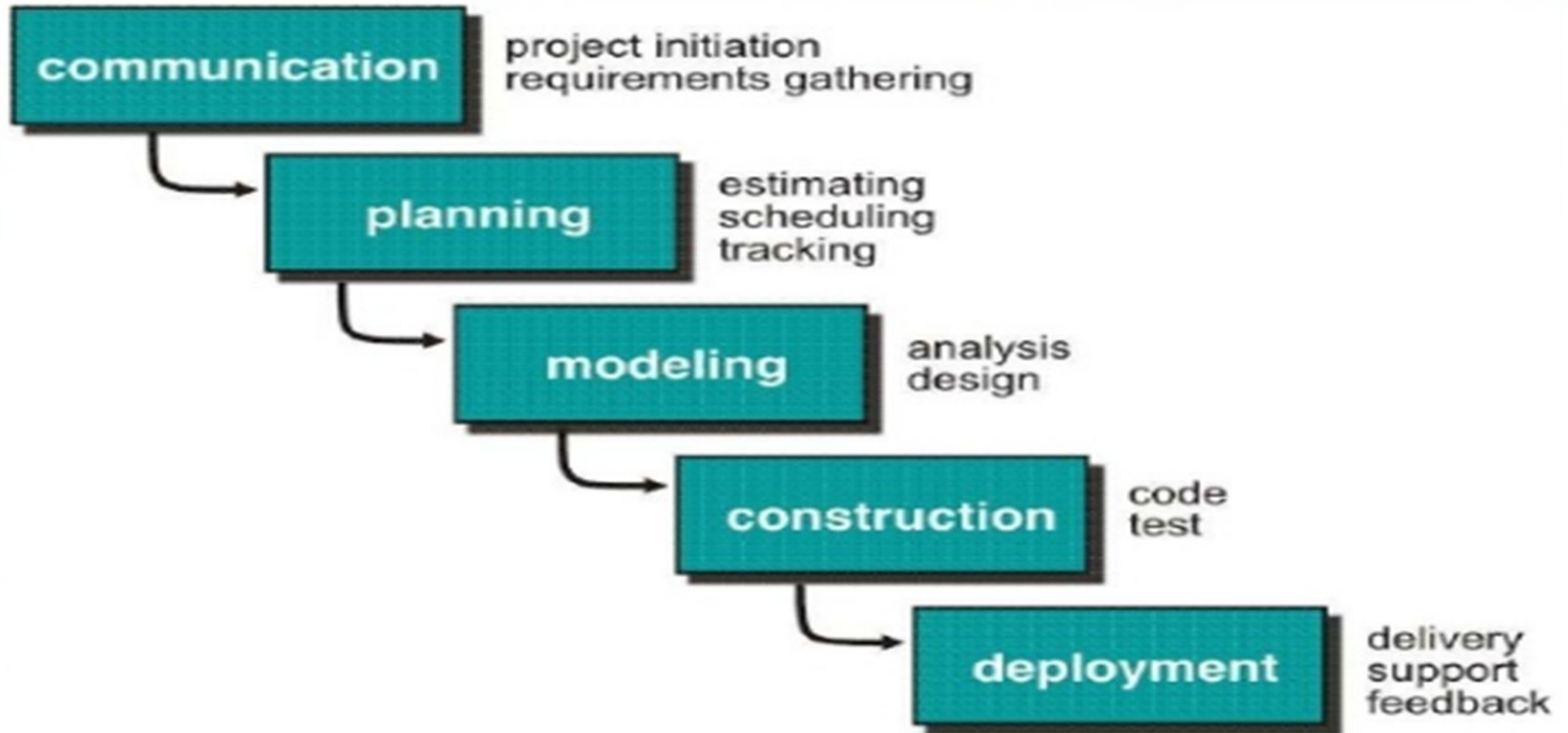
Waterfall Model

- The Waterfall Model was first Process Model to be introduced.
- It is very simple to understand and use
- Each phase must be completed fully before the next phase can begin.
- Software testing starts only after the development is complete.

Waterfall Model

- Also called as **Linear Sequential Model** and **Classic Life Cycle Model**.
- A **systematic, sequential approach** to software development.
- **Oldest Model** for software Engineering.

Waterfall Model



When to use the Waterfall model

- Requirements are **very well known, clear and fixed**.
- Product definition is **stable**.
- Technology is **understood**.
- There are no **ambiguous requirements**.
- The **Project is short**.

Advantages of waterfall Model

- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

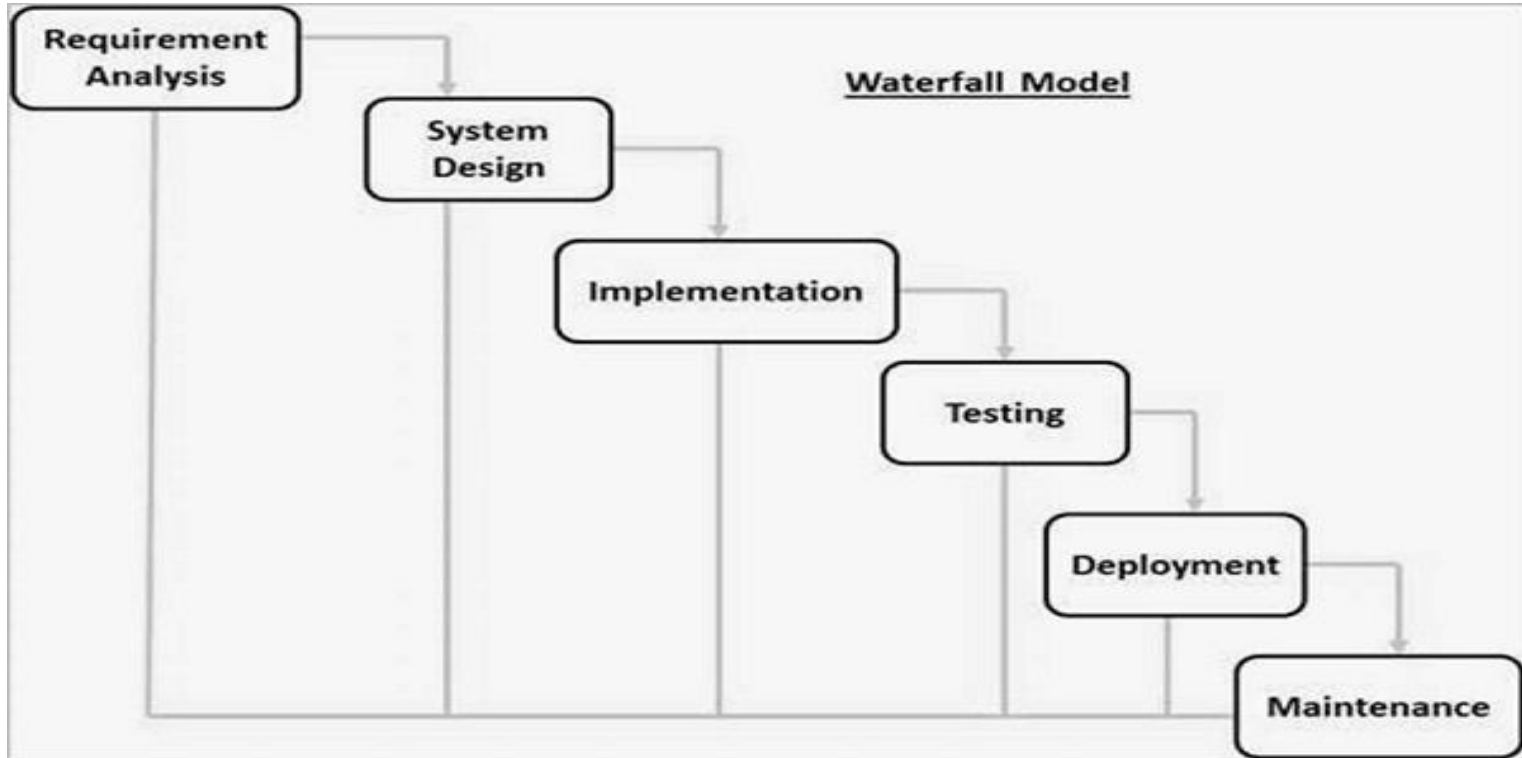
Disadvantages of Waterfall Model

- You **cannot go back a step**; if the design phase has gone wrong, things can get very **complicated** in the implementation phase.
- **High** amounts of **risk**.
- **Not a good model** for complex and object-oriented projects.
- **Poor model** for long and on-going projects.
- **Not suitable** for the projects where requirements are at a moderate to high risk of changing.

Disadvantages of Waterfall Model

- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Water fall model



- In "The Waterfall" approach, the whole process of software development is divided into separate phases.
- In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Waterfall model Phases

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared.
- This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

Waterfall model Phases

- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase.
- Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

Waterfall model Phases

Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

Maintenance – There are some issues which come up in the client environment.

To fix those issues, patches are released. Also to enhance the product some better versions are released.

Maintenance is done to deliver these changes in the customer environment.

Basic idea of incremental model

- Cannot use Waterfall model □ Even, when **Software requirements are well defined.**

Reason:

- Scope of the development effort precludes linear process.
- Quick need of limited set of S/W functionality.
- Such a cases, we can prefer incremental model.

Incremental Process Model

- The incremental model combines elements of **linear and parallel process flows**.
- The incremental model applies linear sequences in a **staggered fashion as calendar time progresses**. Each linear sequence produces deliverable "increments" of the software in a manner.

Example: Word processing s/w development by Incremental model.

- **Increment 1:** delivers basic file management, Editing and document production functions.(core product without supplementary features)
- **Increment 2:** more sophisticated editing and document production functions.
- **Increment 3:** Spelling and grammar checking.
- **Increment 4:** Advanced page layout capability.

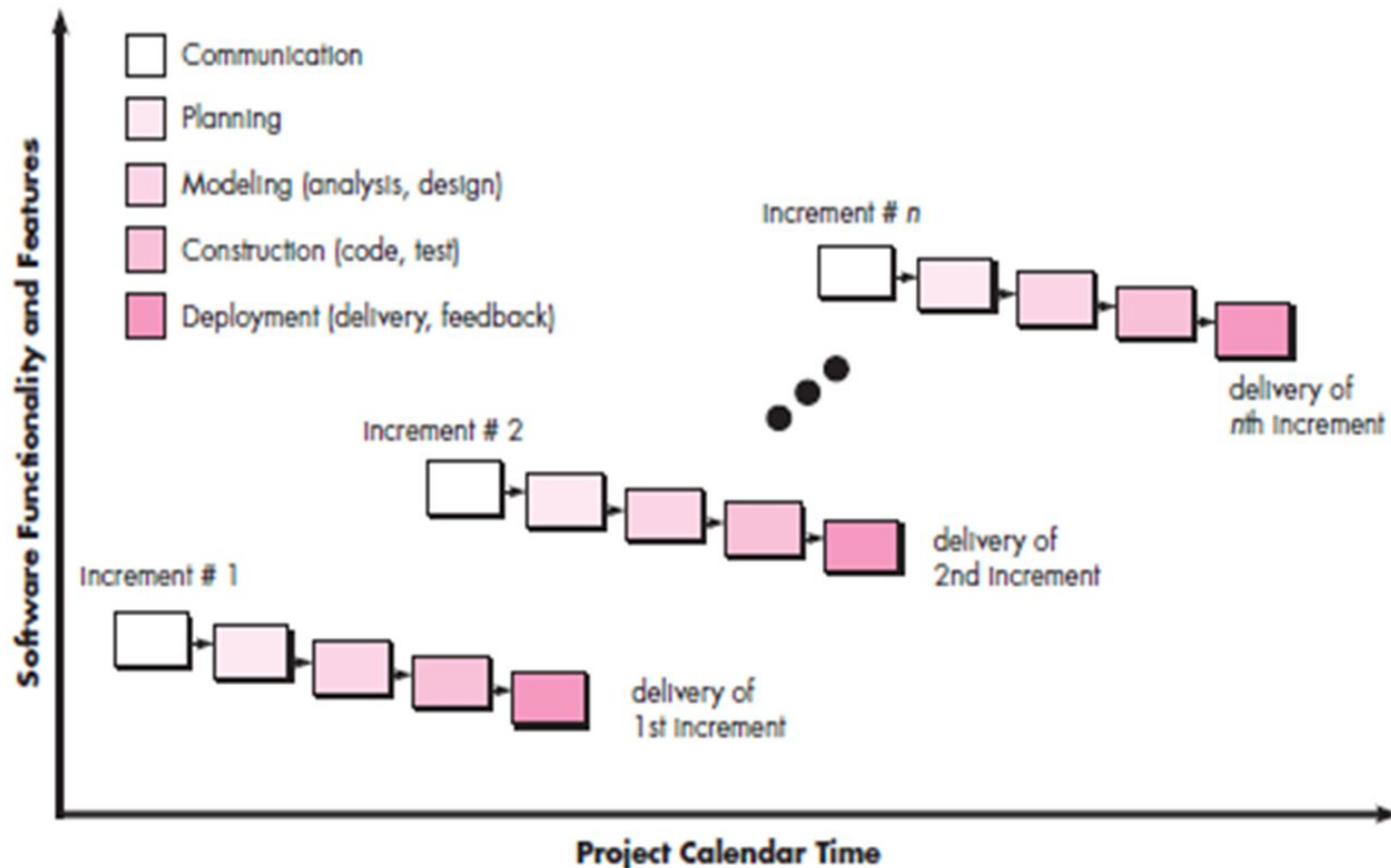
Incremental Process Model

- When an incremental model is used, the first increment is often a **core product**. That is, basic requirements are addressed but many supplementary features (some known, others unknown) remain undelivered.
- The **core product** is used by the customer (or undergoes detailed evaluation). As a result of use and/or evaluation, **a plan is developed for the next increment**. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality.
- This **process is repeated** following the delivery of each increment, **until the complete product is produced**.

Incremental Process Model

- The incremental process model focuses on the delivery of an operational product with each increment.
- Incremental development is particularly useful when staffing is unavailable for a complete implementation by the business deadline that has been established for the project.
- It is a **cyclic software development model** developed in **response to the weaknesses of the waterfall model**.
- Approach attempts to **combine the waterfall sequence** with some of the **advantages of prototyping**.
- It focus on the delivery of an **operational product with each increment**.

Incremental Process Model



Incremental Process Model

- It basically
 - Divides the overall projects into **number of increments**.
 - Then it **applies the waterfall model to each increment**.
 - The system is **put into production when first increment is delivered**.
 - As time passes **addition increments are completed and added to the working system**.
- **Note:**
 - Phase on each increment are sequential.
 - This **process is repeated**, the delivery of each increment, **until the complete product is produced**.

Incremental Process Model

Advantages

- Generates working software quickly during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk.

Incremental Process Model

Disadvantage

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally .
- Total cost is higher than waterfall.

Summary

- Waterfall model- A systematic, sequential approach to software development.
- When to use waterfall model - Requirements are very well known, clear and fixed.
- Phases of waterfall model- Requirement Gathering and analysis , System Design, Implementation, Testing, Deployment and Maintenance
- Basic idea of incremental model
- The incremental process model focuses on the delivery of an operational product with each increment
- When to use- Quick need of limited set of Software functionality



MCQ

1. Which model is also called as the classic life cycle or the Waterfall model?
 - a) Iterative Development
 - b) Linear Sequential Development**
 - c) RAD Model
 - d) Incremental Development

2. Programs, documents and data that are produced as a consequence of activities and tasks defined by process are called_____.
 - a) work product**
 - b) user product
 - c) control process
 - d) open source

MCQ

3. Which one of the following models is not suitable for accommodating any change?
- a) Build & Fix Model
 - b) Prototyping Model
 - c) RAD Model**
 - d) Waterfall Model
4. Oldest paradigm for software engineering is _____
- a) Incremental process model
 - b) RAD model
 - c) Evolutionary process model
 - d) Waterfall model**

MCQ

5. Maintenance is the final phase in waterfall model.

a) **True**

b) False

6. Each iteration in incremental model passes through the

a) communication and modeling phases

b) planning and construction phases

c) deployment and maintenance phases

d) **all phases**

MCQ

7. What is the major advantage of using Incremental Model?

- a) **Customer can respond to each increment**
- b) Easier to test and debug
- c) It is used when there is a need to get a product to the market early
- d) Easier to test and debug & It is used when there is a need to get a product to the market early.

8. The Incremental Model is a result of combination of elements of which two models?

- a) Build & FIX Model & Waterfall Model
- b) Linear Model & RAD Model
- c) **Linear Model & Prototyping Model**
- d) Waterfall Model & RAD Model

MCQ

9. Incremental process model focuses on delivery of an operational product with

- a) no increment
- b) each decrement
- c) each increment**
- d) None Of These

10. Phase on each increment are _____

- a) Sequential**
- b) Non sequential
- c) both a & b
- d) None of the above

Next Session

**Process model: Evolutionary Process Model,
Unified Process**

Thank You!



SRI KRISHNA ARTS & SCIENCE COLLEGE

Coimbatore - 641 008

Department of Information Technology & Cognitive Systems



22ITU10A - Software Engineering and UML

Class	II B.Sc.IT A
Google Classroom code	ylrlutw
Topics Covered	Software Process : Evolutionary & Unified Process Model

K.Sutha

Department of Information Technology & Cognitive Systems

Online Course – Programming Basics

Course content

4 sections • 13 lectures • 1h 59m total length

▼ Introduction

▼ Selenium Automation Tool - Basics

▼ Locators and Identifiers | Working with Web Elements | Selenium WebDriver

▼ Waits in Selenium WebDriver and Mouse Operations

- ❖ In this course, you will learn the basics of Selenium automation tool - setup, running code on chrome browser
- ❖ Identifying web elements on a web page using locators and identifiers in Selenium WebDriver
- ❖ Synchronizing your application under test with Selenium code using waits in Selenium WebDriver
- [Click this link to take up the course.](#)



Course Outcome

- **CLO 1-**Describes about software engineering, process models, analysis and design modelling, agility and SCRUM, frameworks and Unified Modeling Language (UML)
- **CLO 2-** Apply the framework for agility
- **CLO 3-** Adopt the concept of UML diagrams to a sample project
- **CLO 4-** Practice the concepts of selenium packages in Java to automate web browser.

ATTENDANCE



SNAP TALK



Topics to be Covered

- Evolutionary Process Model
- Unified Process

Evolutionary Process Model

- Evolutionary models are iterative.

- Two common evolutionary process models are

1. Prototyping
2. Spiral Model

Evolutionary Process Model



- In the diagram above when we work iteratively we create a rough product in one iteration, then review it and improve it in the next iteration, and so on until it's finished.
- As shown in the image above, in the first iteration the whole painting is sketched roughly, then in the second iteration colors are filled and in the third iteration, finishing is done.
- Hence, in the iterative model the whole product is developed step by step.

When - Prototyping Model

When we can approach Prototyping model,

- Customer defines set of general objective rather than detailed requirements.
- Developer may be unsure of,
 1. Efficiency of an algorithm
 2. Adaptability of an OS
 3. Human-machine interaction

Prototyping Model

- **Stand-alone** process model.
- A Prototyping model suggest that **before carrying out the development of the actual software, a working prototype of the system should be built.**
- A prototype is **working model of software with some limited functionality.**
- Prototype is used to allow the **users evaluate the developer proposals & try them out before implementation.**

Prototyping Model (Conti...)

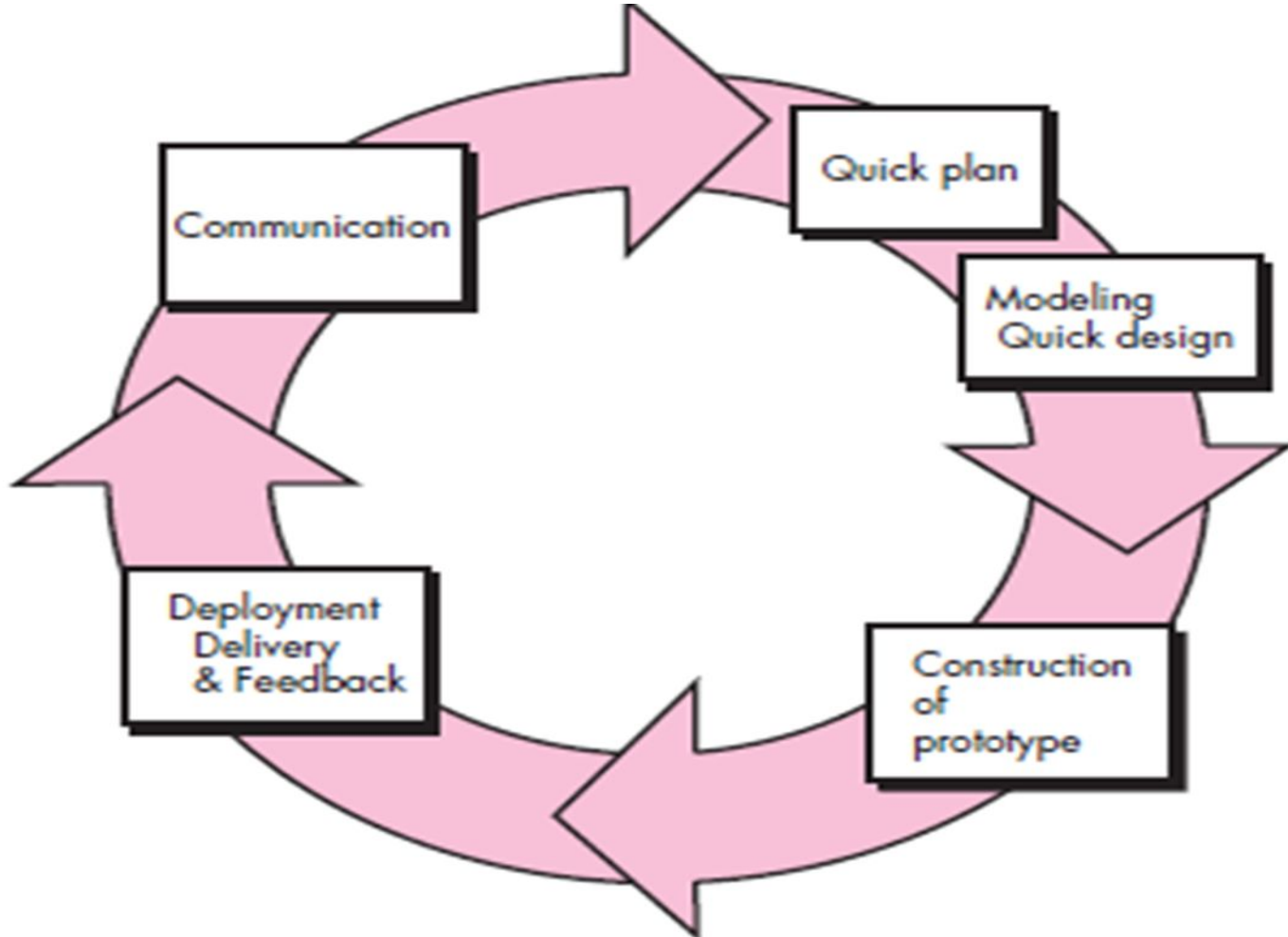


Figure: Prototyping paradigm

Prototyping Model (Conti...)

Requirements gathering & analysis (Communication):

- A prototype model begins with requirement analysis.
- The requirements are defined in detail.
- The user is interviewed in order to know the requirements of the system.

Quick design

- When requirements are known, a quick design is created.
- It's not a detailed design.
- It includes the important aspects of the system – it gives an idea of the system to the user.

Prototyping Model (Conti...)

Construction of prototype:

- Information gathering from quick design is modified to form a prototype.
- It represents a “rough design” of the required system.

Customer evaluation of prototype:

- The build prototype is presented to the customer evaluation.

Prototyping Model (Conti...)

Prototype refinement:

- Once the user evaluate the prototype, it is refined according to the requirements.
- When the user satisfied with developer prototype, the final system is developed based on final prototype. (Iterative Model)
- Test the product to find the error & maintain the system.

Prototyping Model (Conti...)

Advantages

- Users are actively involved in the development.
- Errors can be detected much earlier
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be quickly identified easily.

Prototyping Model (Conti...)

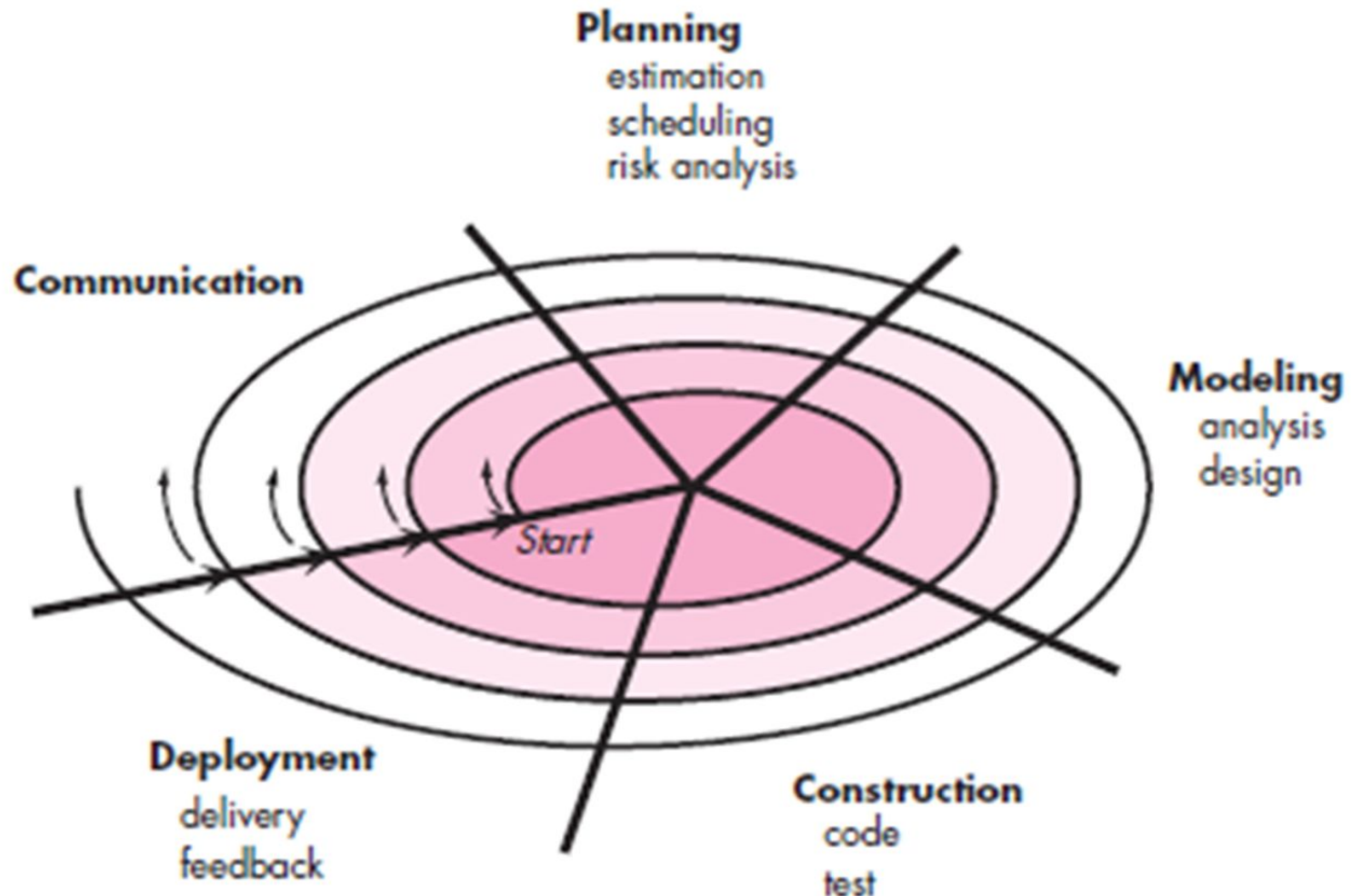
Disadvantage

- If the user is not satisfied – A new prototype is developed.
- Increase the complexity of the system beyond the original plans.
- User confusion of prototype & finished system
- Developer misunderstanding of user objective.
- Excessive development time of the prototype.

Spiral Model

- Spiral model was first described by “Barry Boehm”.
- It combines features of prototyping and the waterfall model.
(Iterative)
- It is best suitable for complex, expensive, and large projects.
- Risk-driven rather than document driven

Spiral Model (Conti...)



Spiral Model (Conti...)

- It has two distinguishing features. They are,
 1. Cyclic approach □ implementation while decreasing its degree of risk
 2. Anchor point milestones □ milestones for ensuring stakeholders' commitment □ Combination of work products & conditions are attained along the path of the spiral.
- Spiral model divided into a set of framework activities is called task regions.

When to use Spiral Model

- When costs and risk evaluation is important
- For medium to high-risk projects
- Unsure of user needs
- Requirements are complex
- Significant changes are expected

Spiral Model (Conti...)

Advantages

- Changing requirements can be accommodated.
- Allows for extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early
- Early and frequent feedback from users.

Spiral Model (Conti...)

Disadvantages

- End of the project may not be known early.
- Not suitable for small or low-risk projects and could be expensive for small projects.
- Process is complex.
- Spiral may go indefinitely.
- Risk assessment expertise is required.

UNIFIED PROCESS MODEL

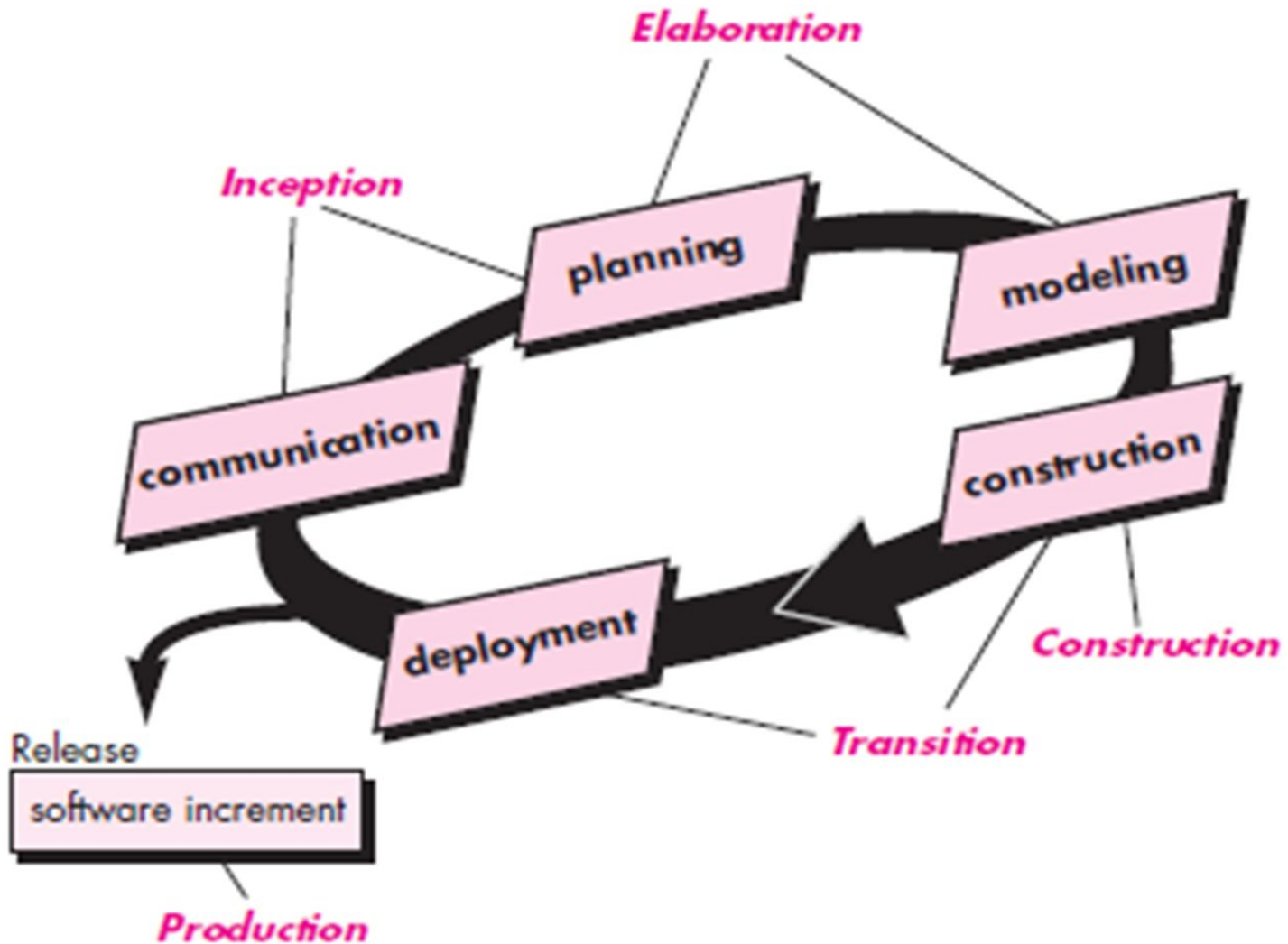
Unified Process Model

- Attempt to draw on the **best features** and characteristics of **traditional software process models**.
- Importance of **customer communication** and **streamlined methods** for describing the customer's view of a system.

A Brief History

- The unified software development process is a software development process that is “**use-case driven, architecture-centric and iterative and incremental**”.(Jacobson, Booch, Rumbaugh)
- The unified process is **component-based**.
- The unified process uses the **Unified Modeling Language** for documentation and design.

Unified Process Model



Phase of Unified Process

- UP consist the following phases,
 1. Inception Phase
 2. Elaboration Phase
 3. Construction Phase
 4. Transition Phase
 5. Production Phase

Inception Phase

- It encompasses both communication with customer and planning activity.
- Requirements are described through a set of preliminary use cases.
- Use cases are used to capture the functional requirements and to define the content of the iterations.
- Rough architecture is proposed.
- Planning identifies resources, assesses major risks, defines a schedule & establish a basis for the phases are to be applied.

Elaboration Phase

- It encompasses the communication and modeling activities of the generic process model.
- It expands the preliminary use cases, and architectural representation to include 5 different views of Software,
 1. Use-case
 2. Requirements
 3. Design
 4. Implementation
 5. Deployment

Elaboration Phase

- Creates an executable architecture baseline to represent “First cut” executable system.
- Demonstrates the viability of the architecture but does not provide all features & functions of the Software.
- Major task – ensure the scope, risk and delivery dates(modification of plan).

Construction Phase

- It is identical to the construction activity defined for the generic software process.
- It uses an architectural model as input and creates software components (Use-case operational)
- Final version of the software increment is then implemented in the source code.
- Unit tests are designed and executed for each.
- Integration activities (component assembly and integration testing) are conducted.
- Use cases are used to derive a suite of acceptance tests that are executed prior to the initiation of the next UP phase.

Transition Phase

- It encompasses construction & deployment activity.
- Given to end user for beta testing.
- User feedback reports (defects & necessary changes).
- Creates the necessary support information.

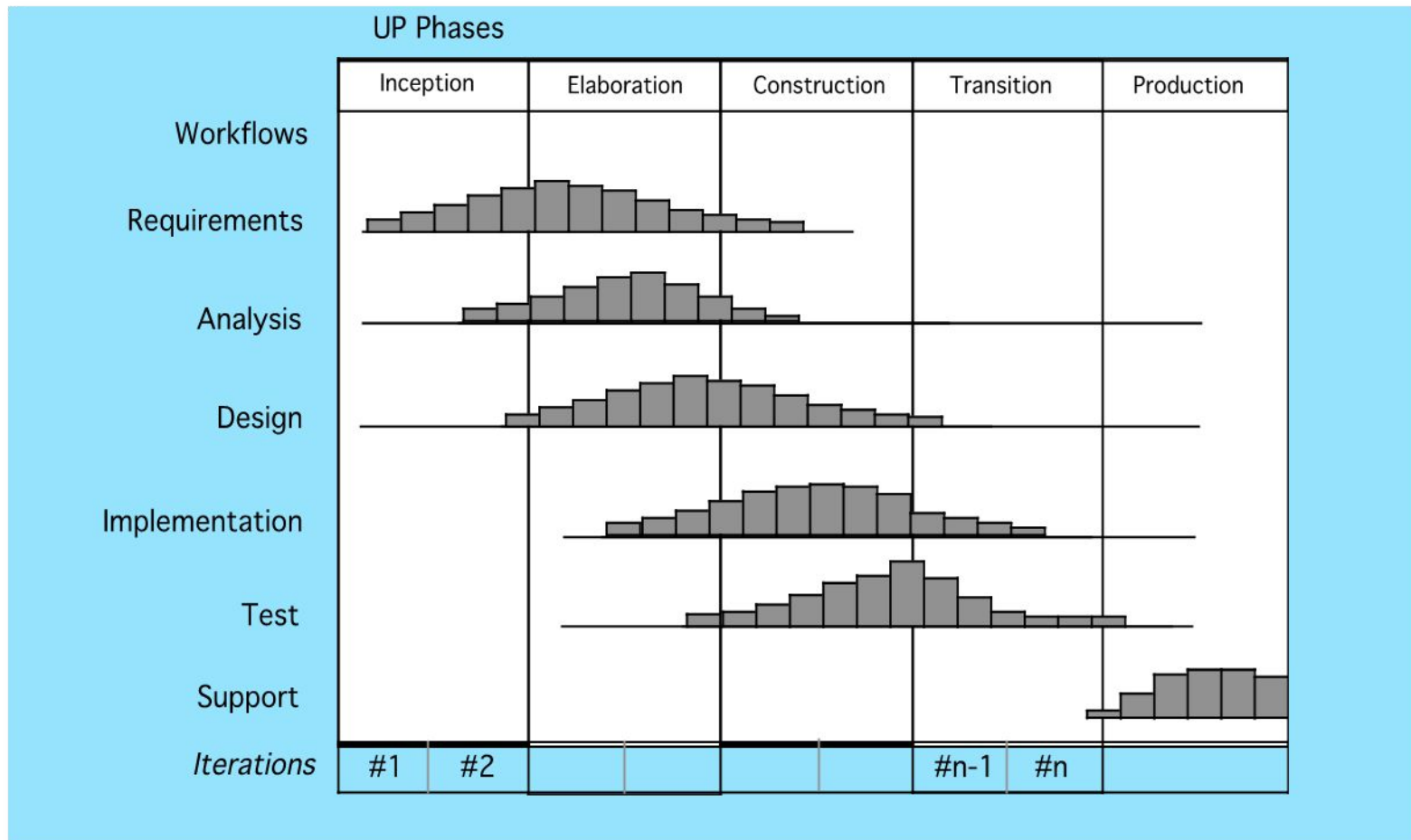
(Example: User manuals, troubleshooting guide, installation procedure)

- Finally, it becomes a usable software release.

Production Phase

- It coincides with the deployment activity.
- During this phase,
 - Ongoing use of Software is monitored
 - Support for the operating environment is provided.
 - Defect report & changes requirements are submitted and evaluated.
- A software engineering workflow is distributed across all UP phases.
- Workflow is analogous to a task set.
- A workflow identifies the tasks required to accomplish an important software engineering action and the work products that are produced as a consequence of successfully completing the tasks.
- The team adapts the process (actions, tasks, subtasks, and work products) to meet its needs.

Unified Process Phases



Summary

- Evolutionary models are iterative.
- A prototype is a working model of software with some limited functionality
- A prototype model begins with requirement analysis.
- When requirements are known, a quick design is created
- Information gathering from quick design is modified to form a prototype.
- The built prototype is presented to the customer evaluation.
- Once the user evaluates the prototype, it is refined according to the requirements.



MCQ

1. The spiral model has two dimensions namely _____
and _____
- a) diagonal, angular
 - b) radial, perpendicular
 - c) **radial, angular**
 - d) diagonal, perpendicular
2. The spiral model was originally proposed by
- a) IBM
 - b) **Barry Boehm**
 - c) Pressman
 - d) Royce

MCQ

3. Identify the disadvantage of Spiral Model.
- a) **Doesn't work well for smaller projects**
 - b) High amount of risk analysis
 - c) Strong approval and documentation control
 - d) Additional Functionality can be added at a later date
4. To refine requirements for the software, prototype model use?
- a) **feedback**
 - b) quick plan
 - c) Construction
 - d) quick design

MCQ

5. Prototype is evaluated by customer or end user in

- a) **deployment**
- b) quick plan
- c) quick design
- d) communication

6. The unified process is _____ based.

- a) **Component**
- b) Custom
- c) Process
- d) Activity

MCQ

7. The _____ phase encompasses both communication with customer and planning activity.

- a) **Inception Phase**
- b) Elaboration Phase
- c) Construction Phase
- d) Transition Phase

8. The phase that expands the preliminary use case is _____.

- e) **Elaboration Phase**
- f) Construction Phase
- g) Transition Phase
- h) Production Phase

MCQ

9. Final version of the software increment are then implemented by source code in _____.

- a) **Elaboration Phase**
- b) Construction Phase
- c) Transition Phase
- d) Production Phase

10. The phase that coincides with the deployment activity is _____.

- a) Elaboration Phase
- b) Construction Phase
- c) Transition Phase
- d) **Production Phase**

Next Session

System Engineering

Thank You!