hillel
компьютерная
школа

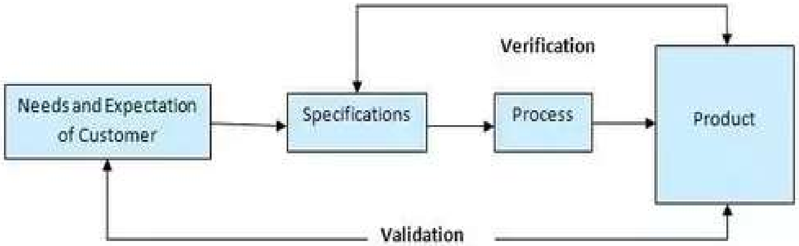
ISTQB®

Preparation to ISTQB Foundation Level Certification Exam

By Vladimir Arutin

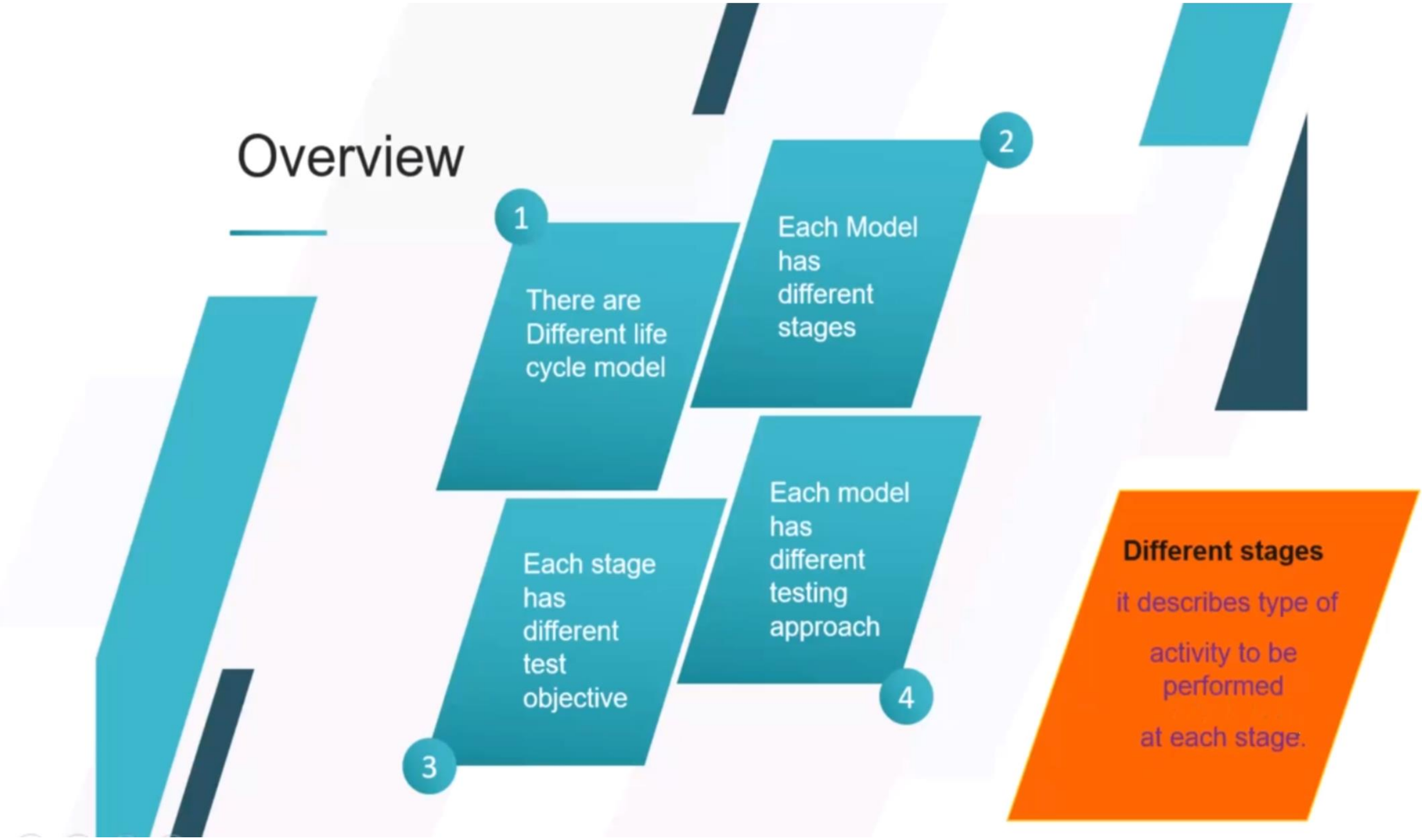
TESTING THROUGHOUT THE SOFTWARE LIFE CYCLE

Verification: was the product built right?
Validation: was the right product built?



Validation: Confirmation by examination and through provision of objective evidence that **the requirements for a specific intended use or application** have been fulfilled.

Verification: Confirmation by examination and through provision of objective evidence **that specified requirements** have been fulfilled.



Overview

1 There are Different life cycle model

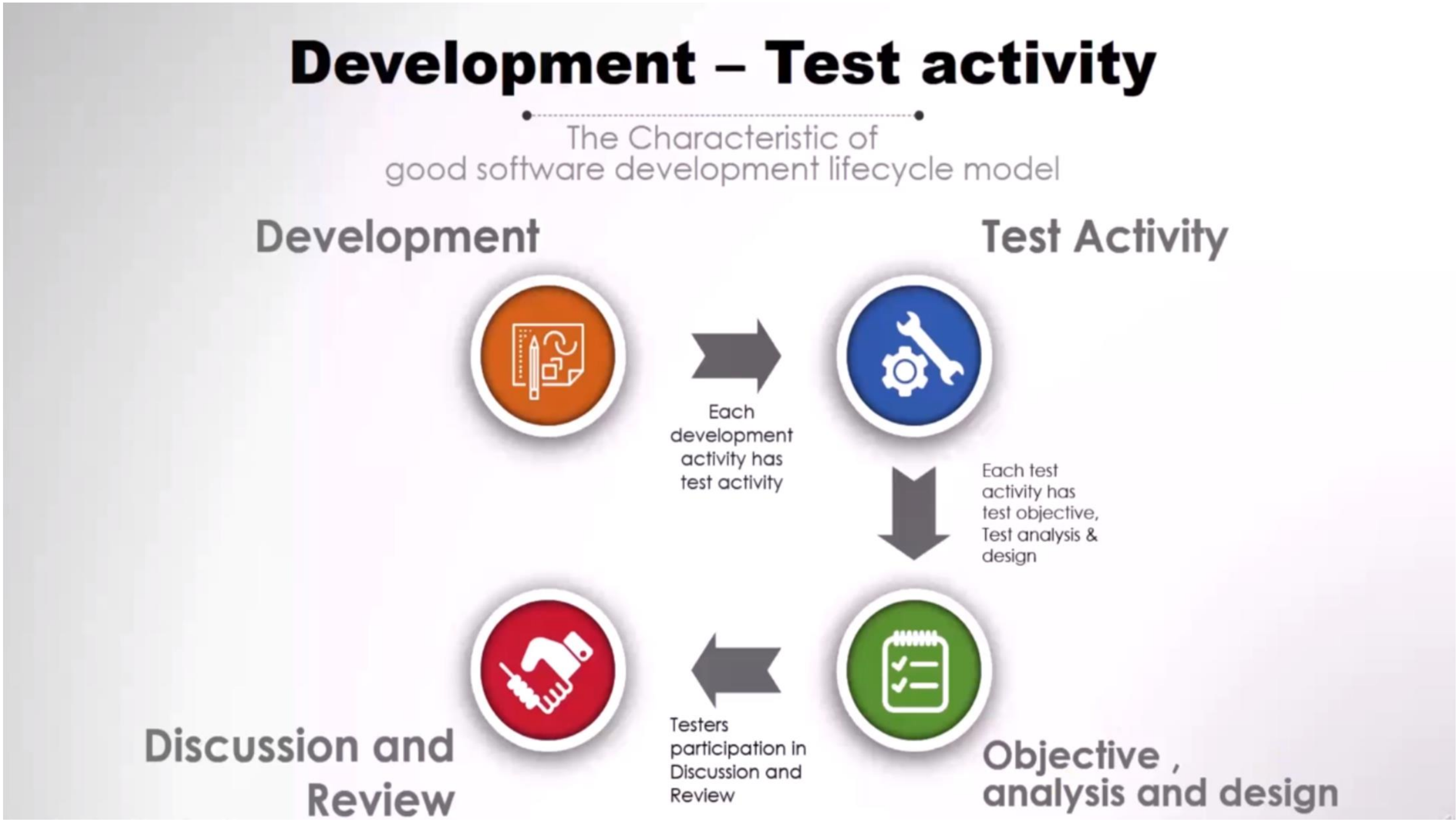
2 Each Model has different stages

3 Each stage has different test objective

4 Each model has different testing approach

Different stages

it describes type of activity to be performed at each stage.



Development – Test activity

The Characteristic of
good software development lifecycle model

Development

Each development activity has test activity

Test Activity

Each test activity has test objective, Test analysis & design

Objective , analysis and design

Testers participation in Discussion and Review

Discussion and Review

CHARACTERISTICS OF GOOD TESTING

- ❑ For every development activity, there is a corresponding test activity

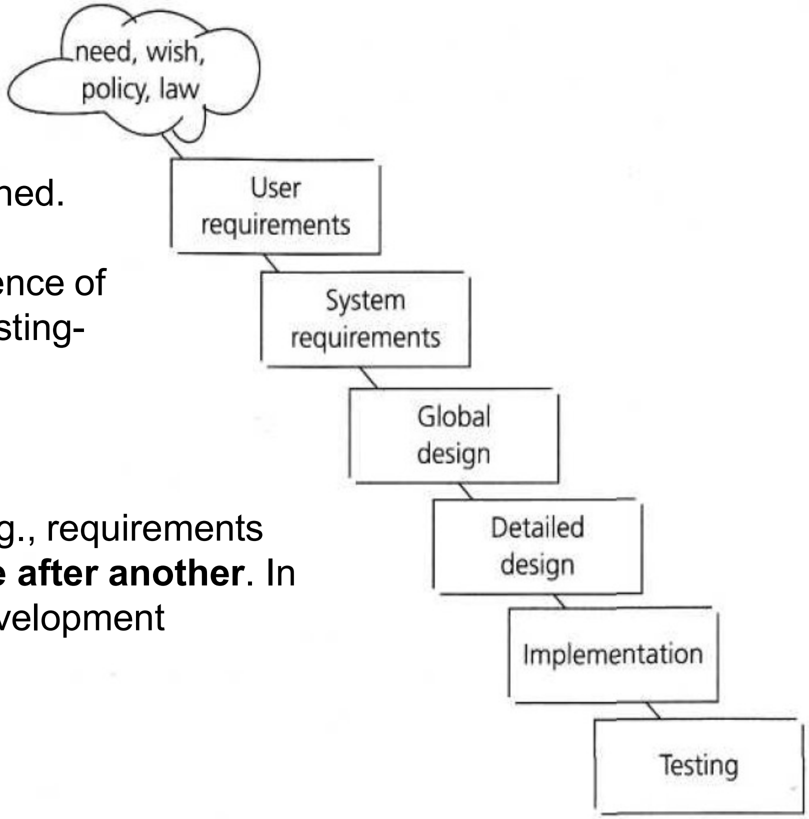
- ❑ Each test level has test objectives specific to that level

- ❑ Test analysis and design for a given test level begin during the corresponding development Activity

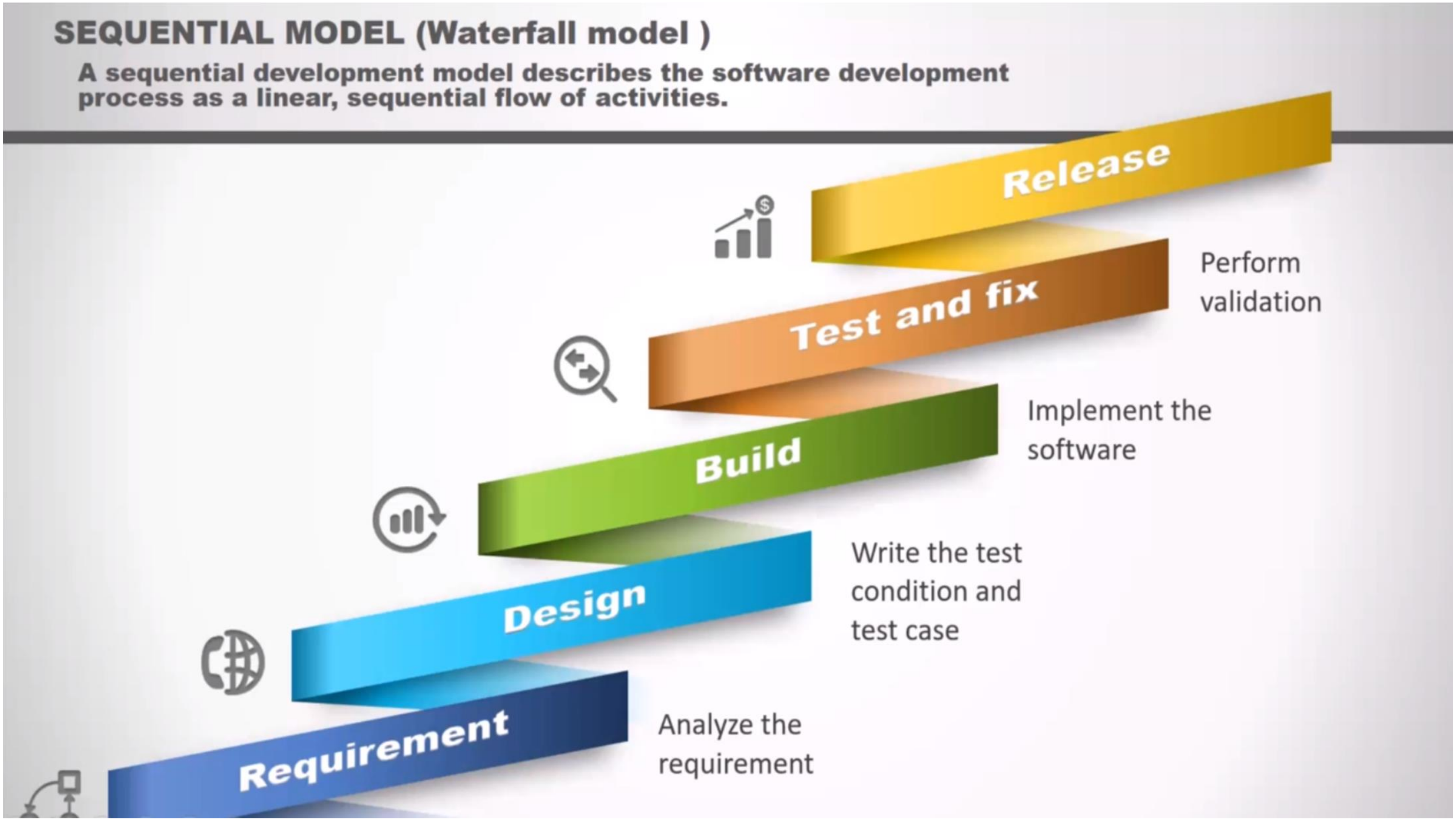
- ❑ Testers participate in discussions to define and refine requirements and design, and are involved in reviewing work products (e.g., requirements, design, user stories, etc.) as soon as drafts are available

WATERFALL MODEL

Waterfall model is one of the earliest models to be designed.
 It is a **sequential** software development process.
The progress of software development is seen as a sequence of phases like Requirements->Design->Implementation->Testing->Deployment->Maintenance.



In the Waterfall model, the development activities (e.g., requirements analysis, design, coding, testing) **are completed one after another**. In this model, test activities only occur after all other development activities have been completed.



SEQUENTIAL MODEL (Waterfall model)

A sequential development model describes the software development process as a linear, sequential flow of activities.

Requirement

Analyze the requirement

Design

Write the test condition and test case

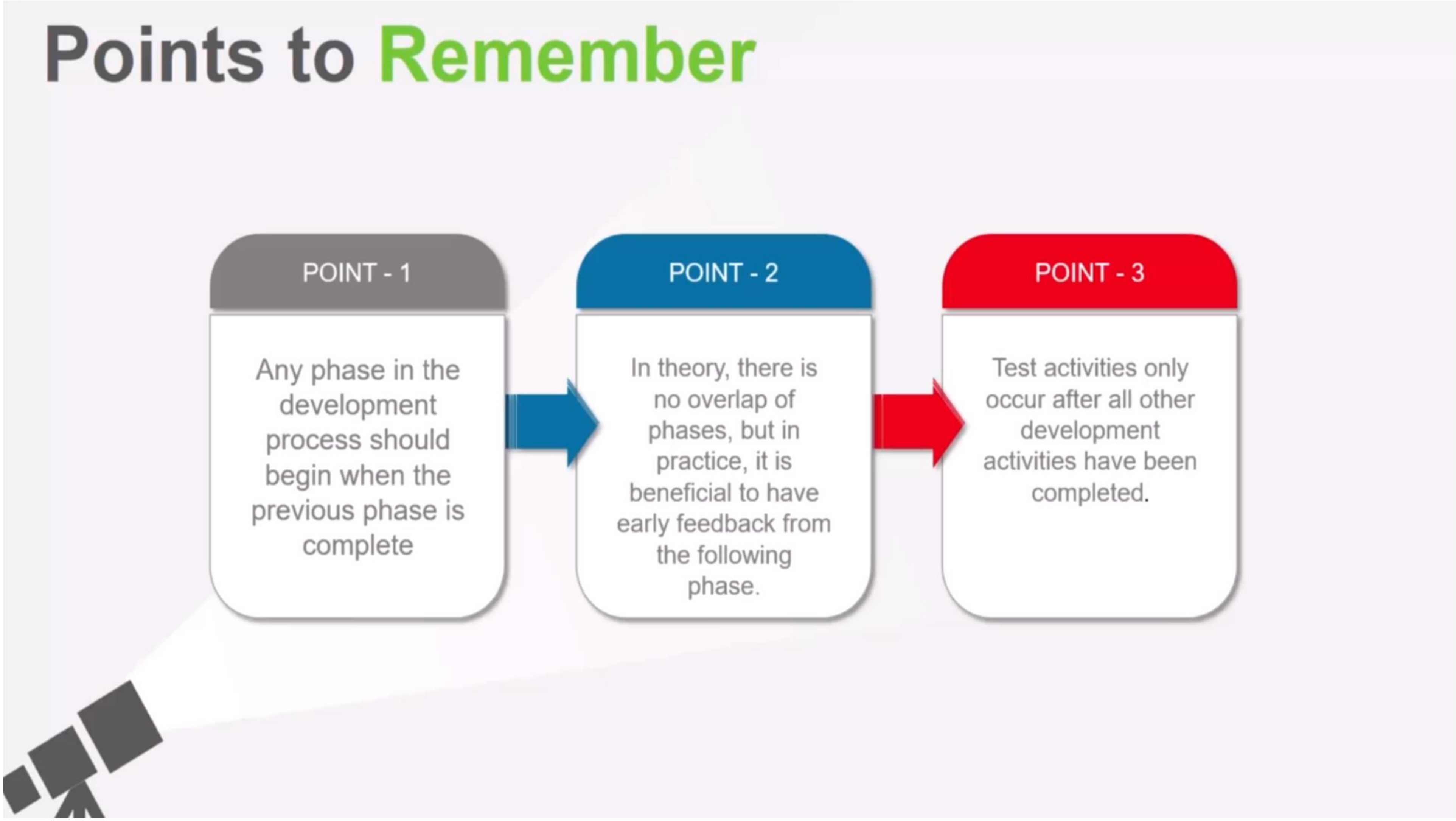
Build

Implement the software

Test and fix

Perform validation

Release



Points to Remember

POINT - 1

Any phase in the development process should begin when the previous phase is complete

POINT - 2

In theory, there is no overlap of phases, but in practice, it is beneficial to have early feedback from the following phase.

POINT - 3

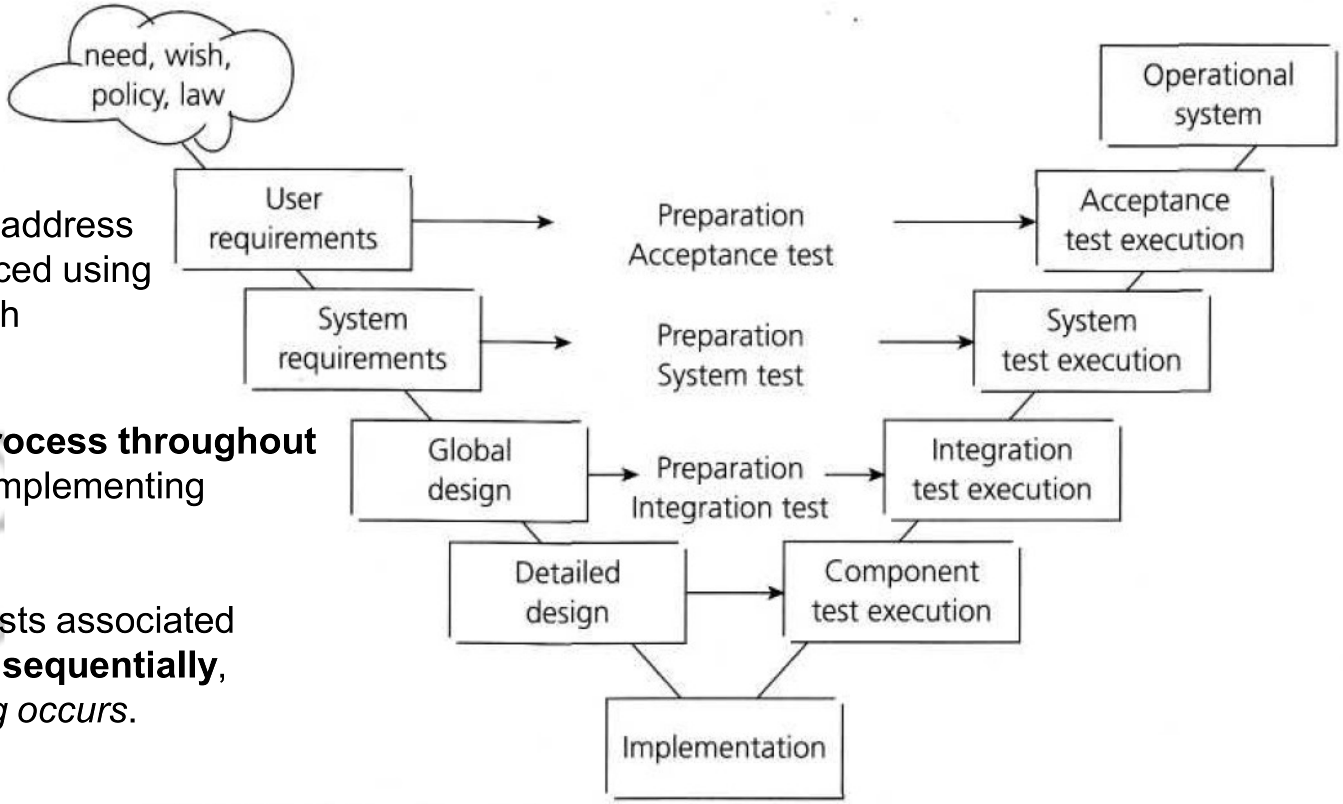
Test activities only occur after all other development activities have been completed.

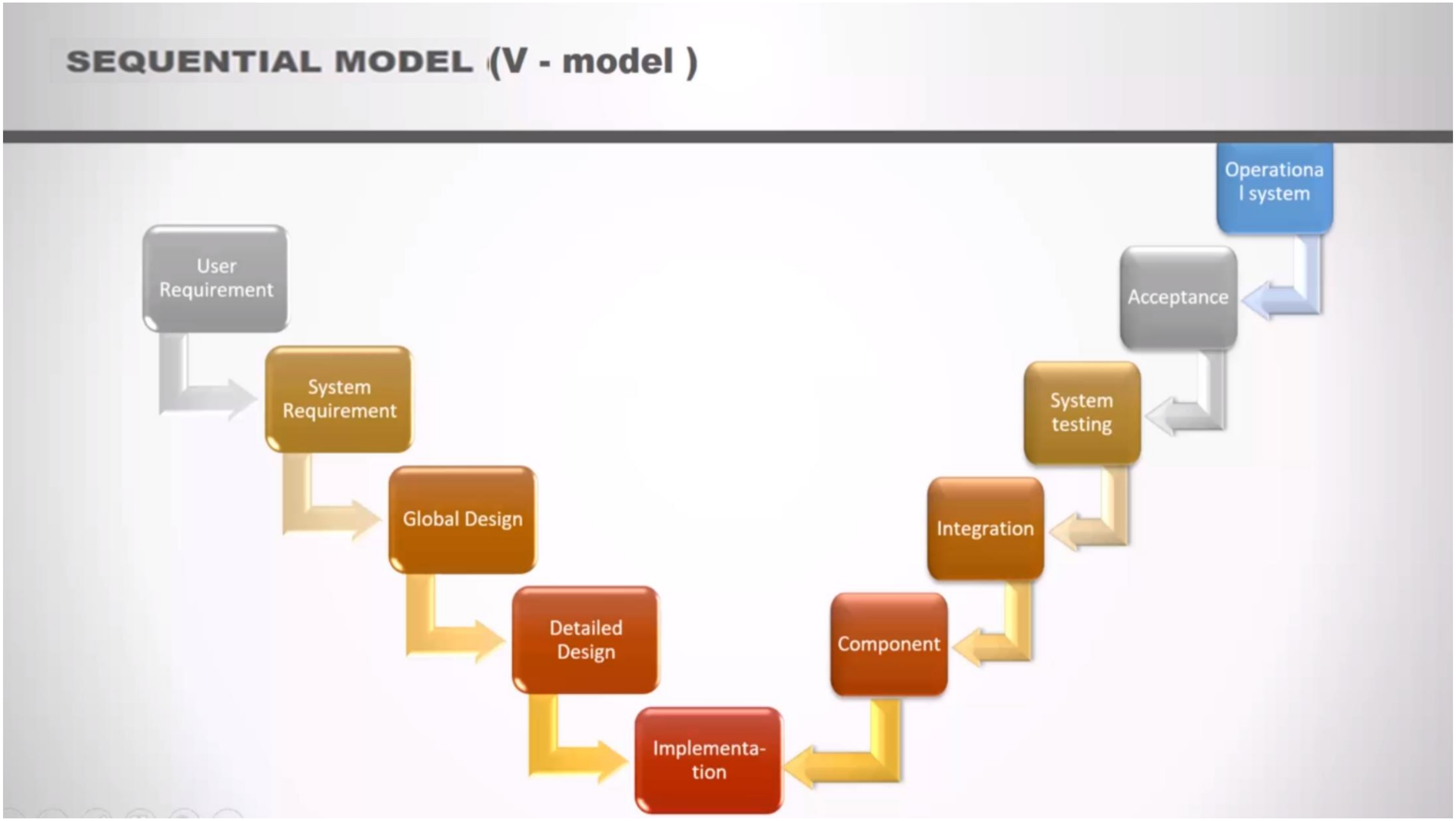
V-MODEL

The **V-model** was developed to address some of the problems experienced using the traditional waterfall approach

V-model integrates the **test process throughout the development process**, implementing the principle of early testing.

In V-model the execution of tests associated with each test level proceeds **sequentially**, but in some cases *overlapping occurs*.





SEQUENTIAL MODEL (V - model)

User Requirement

System Requirement

Global Design

Detailed Design

Implementa-tion

Component

Integration

System testing

Acceptance

Operationa l system

V-MODEL

The V-model is a model that illustrates how testing activities (verification and validation) can be integrated into each phase of the life cycle
Although variants of the V-model exist, a common type of V-model uses **four test levels.**
The four test levels used, **each with their own objectives**, are:

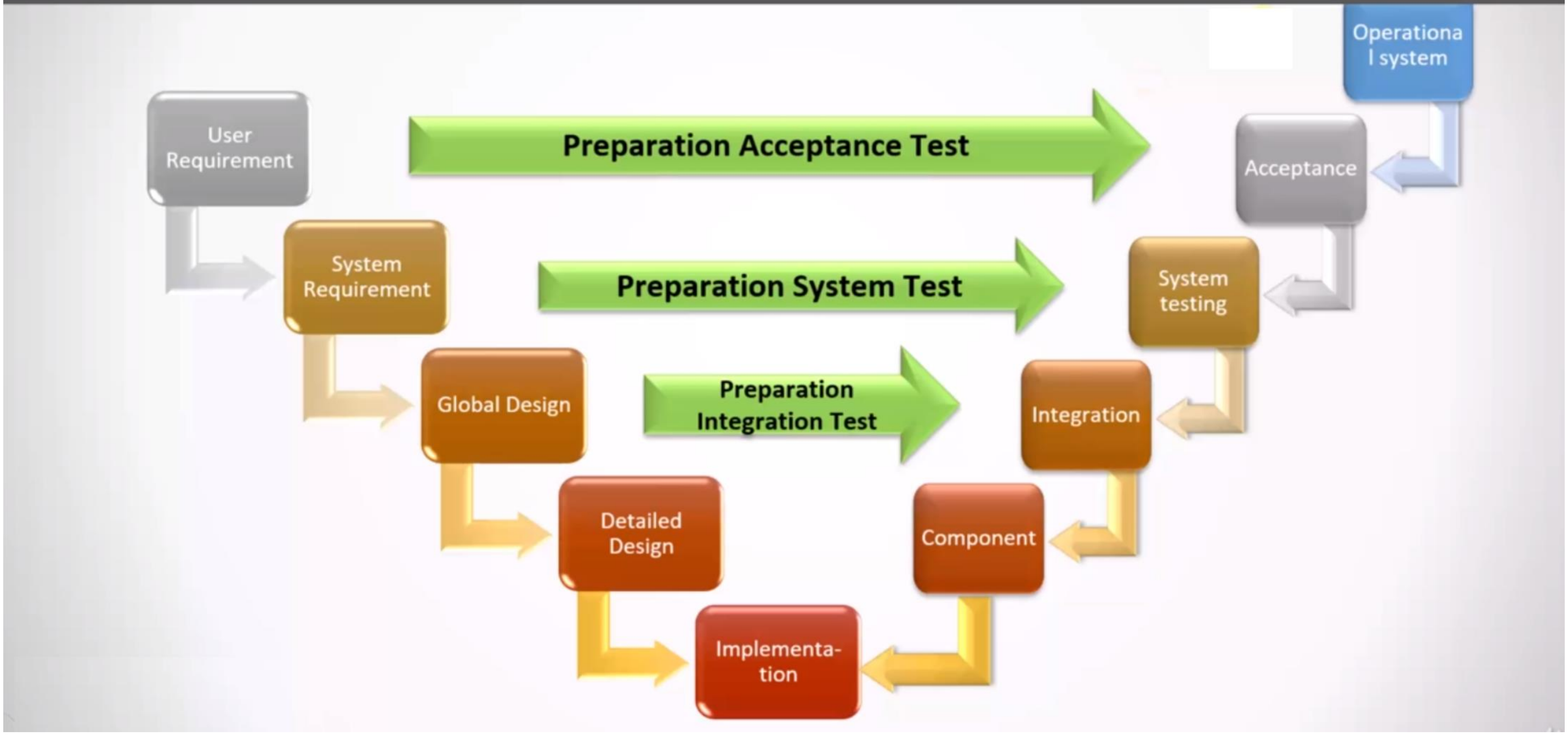
- **component testing**: searches for defects in and verifies the functioning of software components (e.g. modules, programs, objects, classes etc.) that are separately testable;

- **integration testing**: tests interfaces between components, interactions to different parts of a system such as an operating system, file system and hard ware or interfaces between systems;

- **system testing**: concerned with the behavior of the whole system/product as defined by the scope of a development project or product. The main focus of system testing is verification against specified requirements;

- **acceptance testing**: validation testing with respect to user needs, requirements, and business processes conducted to determine whether or not to accept the system.

SEQUENTIAL MODEL (V - model)



User Requirement

System Requirement

Global Design

Detailed Design

Implementa-tion

Component

Integration

System testing

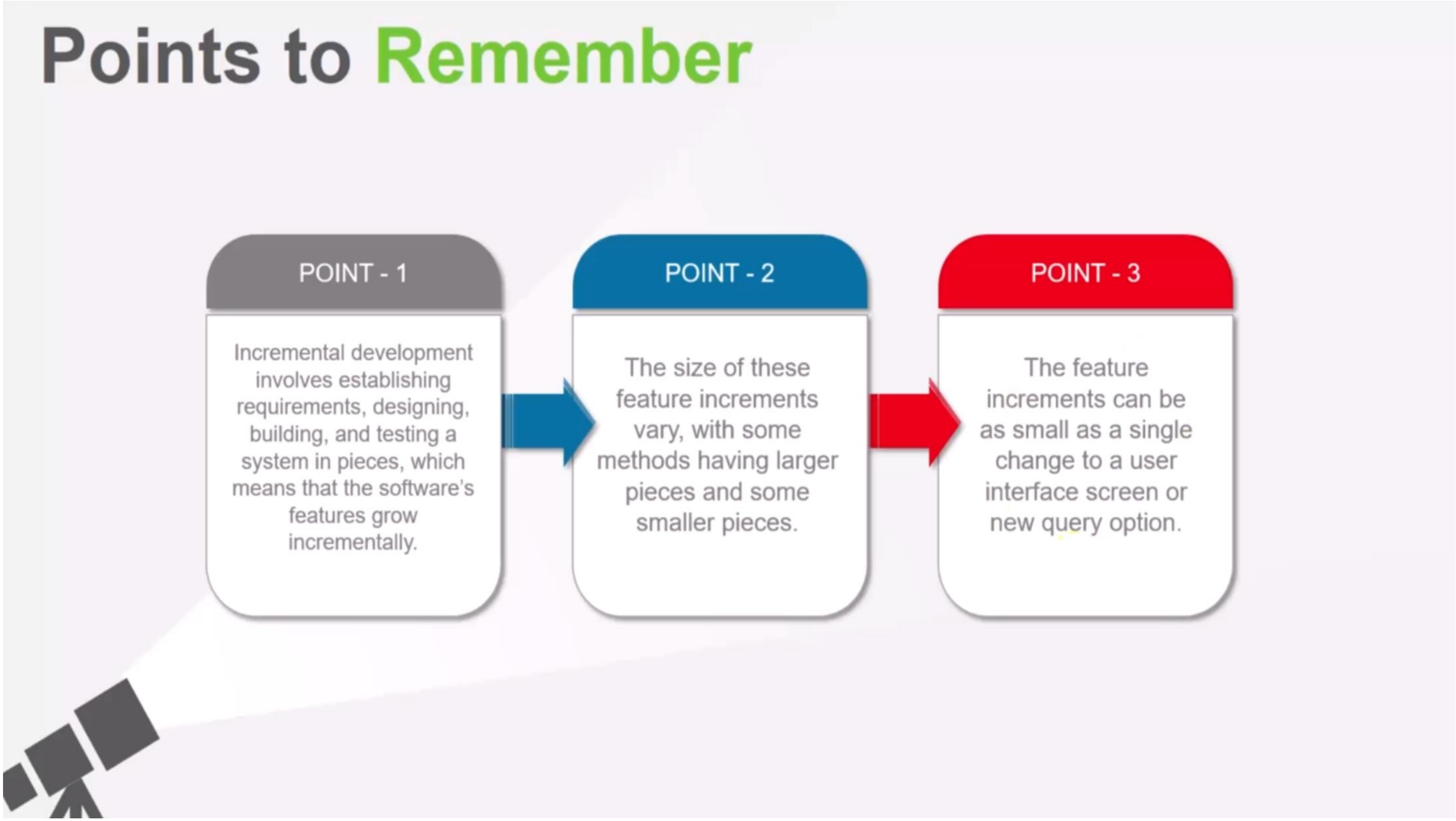
Acceptance

Operationa l system

Preparation Acceptance Test

Preparation System Test

Preparation Integration Test



Points to Remember

POINT - 1

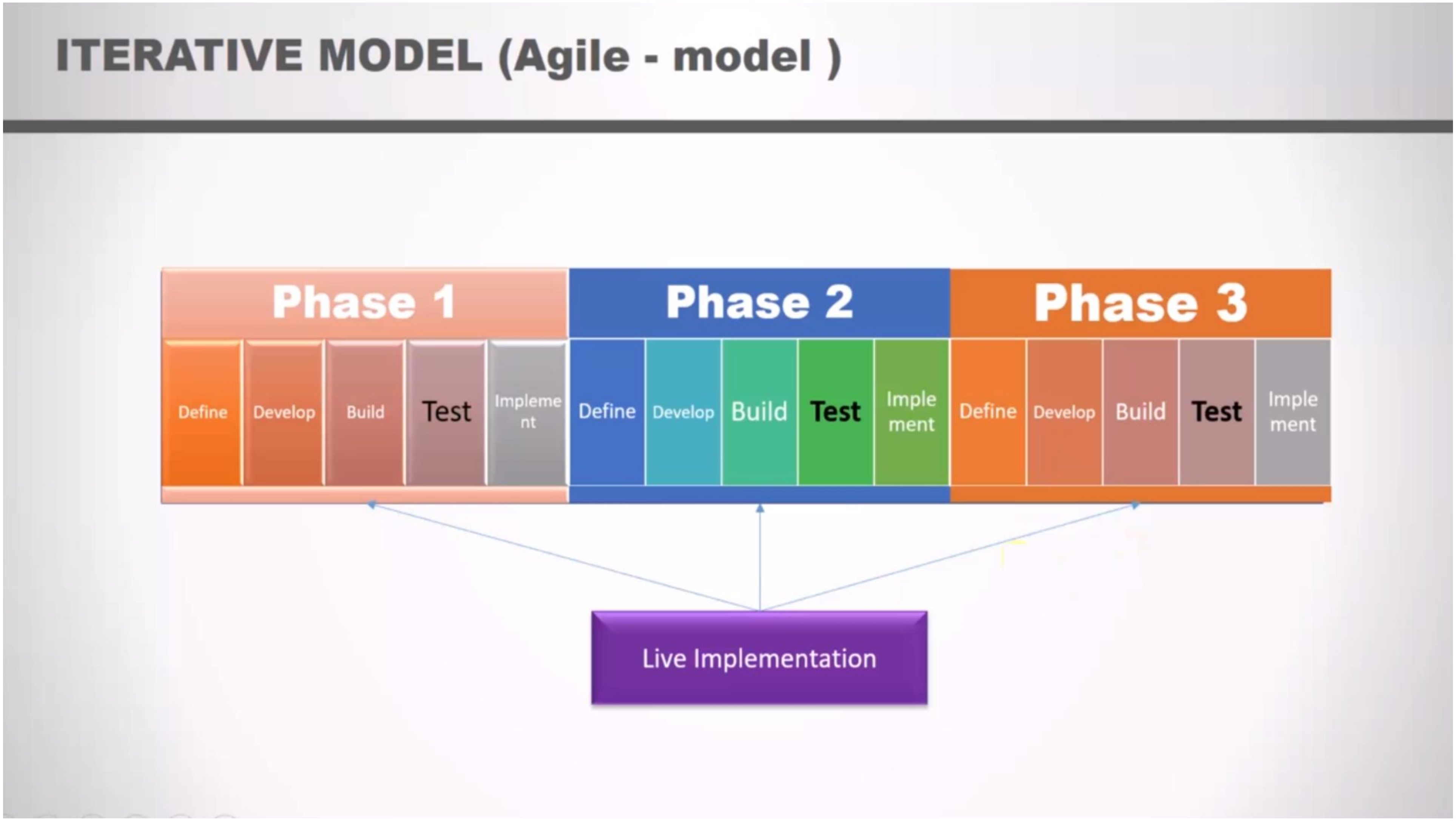
Incremental development involves establishing requirements, designing, building, and testing a system in pieces, which means that the software's features grow incrementally.

POINT - 2

The size of these feature increments vary, with some methods having larger pieces and some smaller pieces.

POINT - 3

The feature increments can be as small as a single change to a user interface screen or new query option.



ITERATIVE MODEL (Agile - model)

Phase 1

Define Develop Build Test Impleme nt

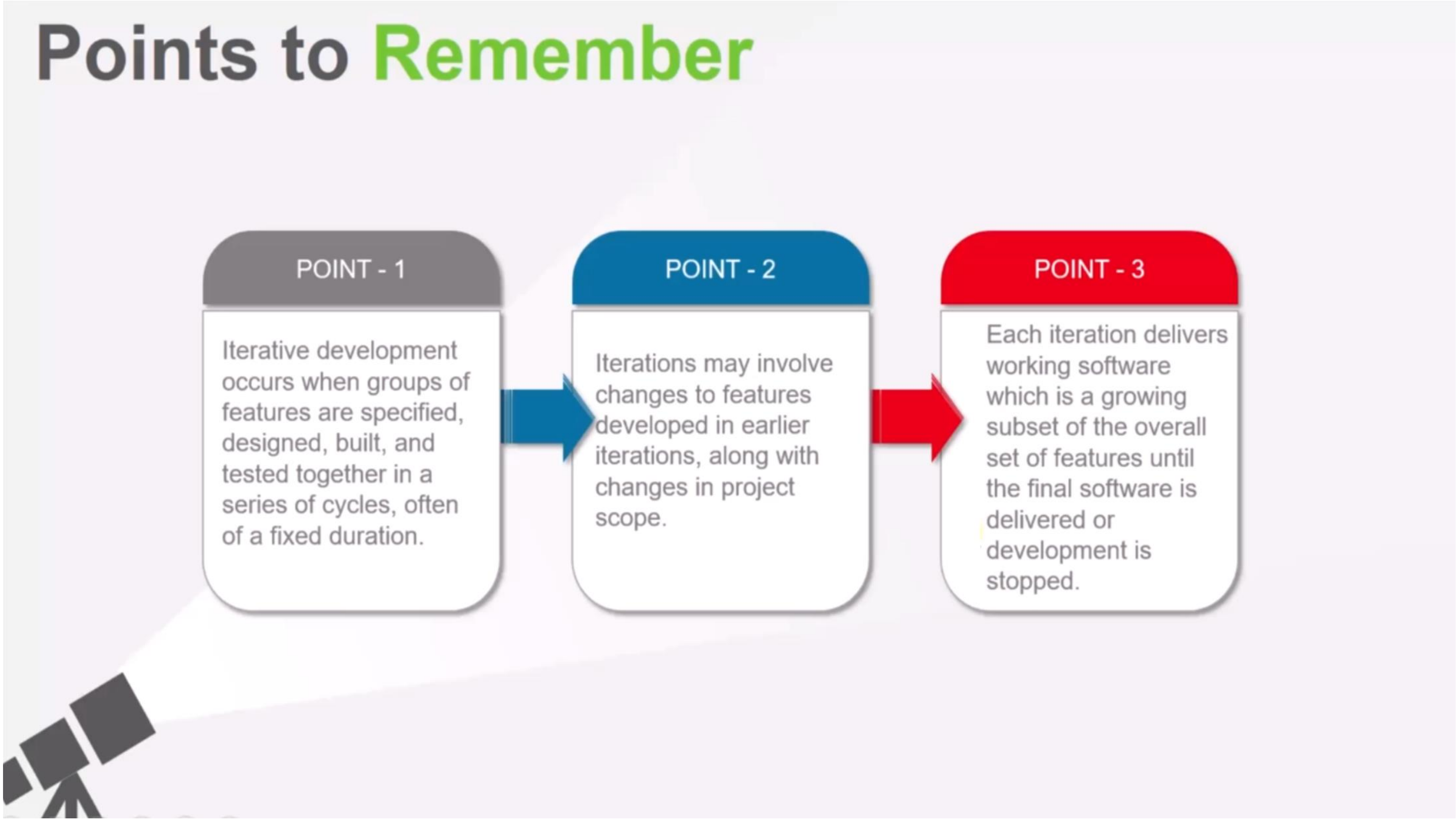
Phase 2

Define Develop Build Test Imple ment

Phase 3

Define Develop Build Test Imple ment

Live Implementation



Points to Remember

POINT - 1

Iterative development occurs when groups of features are specified, designed, built, and tested together in a series of cycles, often of a fixed duration.

POINT - 2

Iterations may involve changes to features developed in earlier iterations, along with changes in project scope.

POINT - 3

Each iteration delivers working software which is a growing subset of the overall set of features until the final software is delivered or development is stopped.

INCREMENTAL DEVELOPMENT MODELS

A common feature of iterative approaches is that the delivery **is divided into increments** or builds with each increment adding new functionality.
The increment produced by an iteration may be tested **at several levels** as part of its development. Subsequent increments will need testing for the **new functionality**, **regression testing** of the existing functionality, **and integration testing** of both new and existing parts.

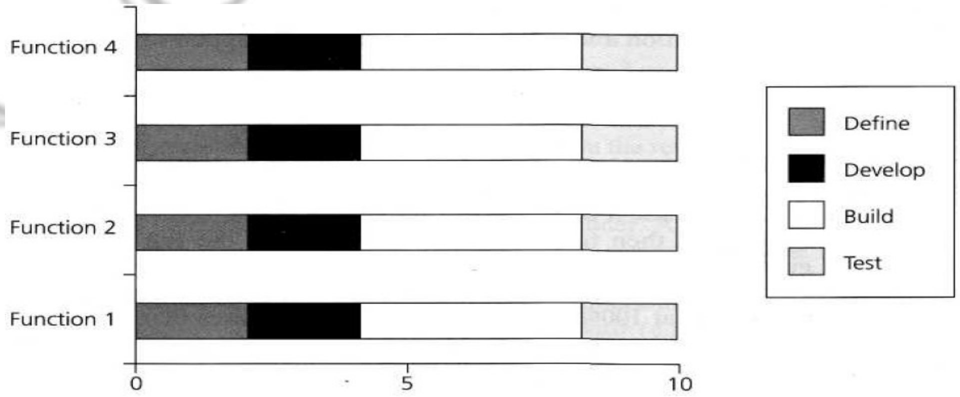
Regression testing **is increasingly important on all iterations** after the first one.

This life cycle can give early market presence with critical functionality, can be simpler to manage because the workload is divided into smaller pieces, and can reduce initial investment although it may cost more in the long run.

Also early market presence will mean validation testing is carried out at each increment, thereby giving early feedback on the business value and fitness-for-use of the product.

INCREMENTAL DEVELOPMENT MODELS EXAMPLES

Rapid Application Development (RAD) is formally a parallel development of functions and subsequent integration.
Components/functions are developed in parallel as if they were mini projects, the developments are time-boxed, delivered, and then assembled into a working prototype. Rapid change and development of the product is possible using this methodology. This methodology allows early validation of technology risks and a rapid response to changing customer requirements. Dynamic System Development Methodology [DSDM] is a refined RAD process that allows controls to be put in place in order to stop the process from getting out of control.



INCREMENTAL DEVELOPMENT MODELS EXAMPLES

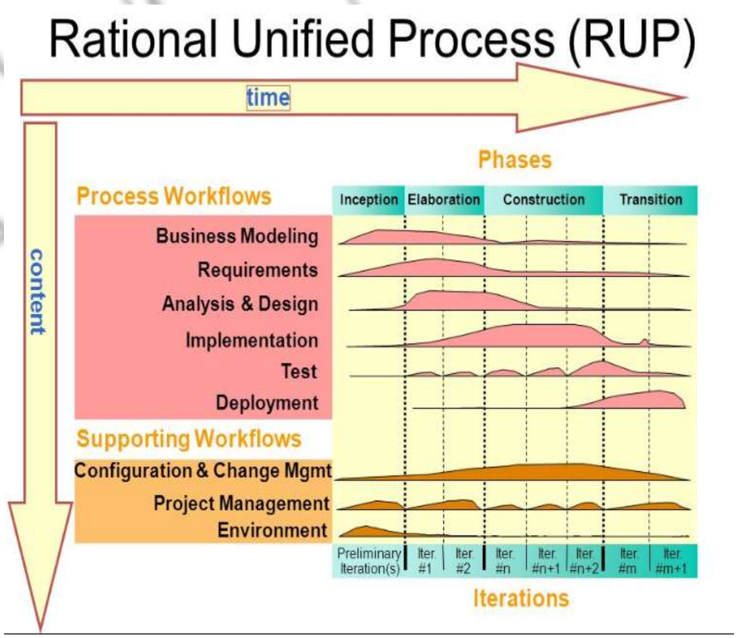
Rational unified process (RUP) is software application development technique with many tools to assist in coding the final product and tasks related to this goal. *Each iteration tends to be relatively long (2-3 months), and the feature increments are correspondingly large, such as two or three groups of related features.*

RUP has the following key characteristics:

- Architecture-centric, where architecture is a function of user needs
- Use-case driven from inception to deployment
- Iterative and incremental, where large projects are divided into smaller projects

RUP follows the following four-phase process:

- ✓ Elaboration: Use cases and architecture are designed.
- ✓ Construction: Activities from design to completed product
- ✓ Transition: Follow-up activities to ensure customer satisfaction
- ✓ Inception: The core idea is envisioned



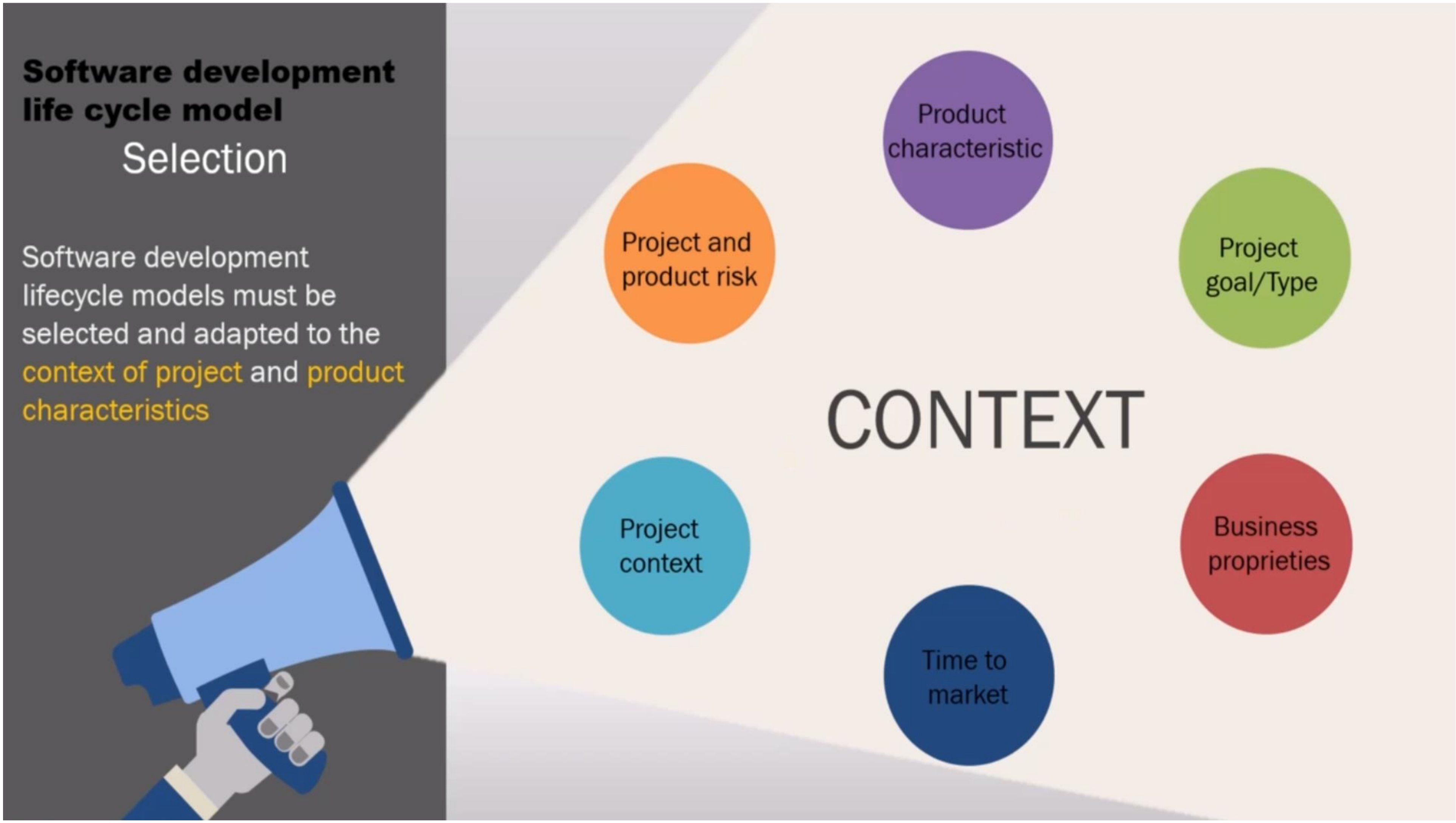
INCREMENTAL DEVELOPMENT MODELS EXAMPLES

Extreme Programming (XP): claims to be more human friendly than traditional development methods, promotes the generation of business stories to define the functionality, pair programming and shared code ownership amongst the developers, states that component test scripts shall be written before the code is written and that those tests should be automated, states that integration and testing of the code shall happen several times a day.

Scrum: Each iteration tends to be relatively short (e.g., hours, days, or a few weeks), and the feature increments are correspondingly small, such as a few enhancements and/or two or three new features

Kanban: Implemented with or without fixed-length iterations, which can deliver either a single enhancement or feature upon completion, or can group features together to release at once

Spiral (or prototyping): Involves creating experimental increments, some of which may be heavily re-worked or even abandoned in subsequent development work



Software development life cycle model Selection

Software development lifecycle models must be selected and adapted to the context of project and product characteristics

CONTEXT

Product characteristic

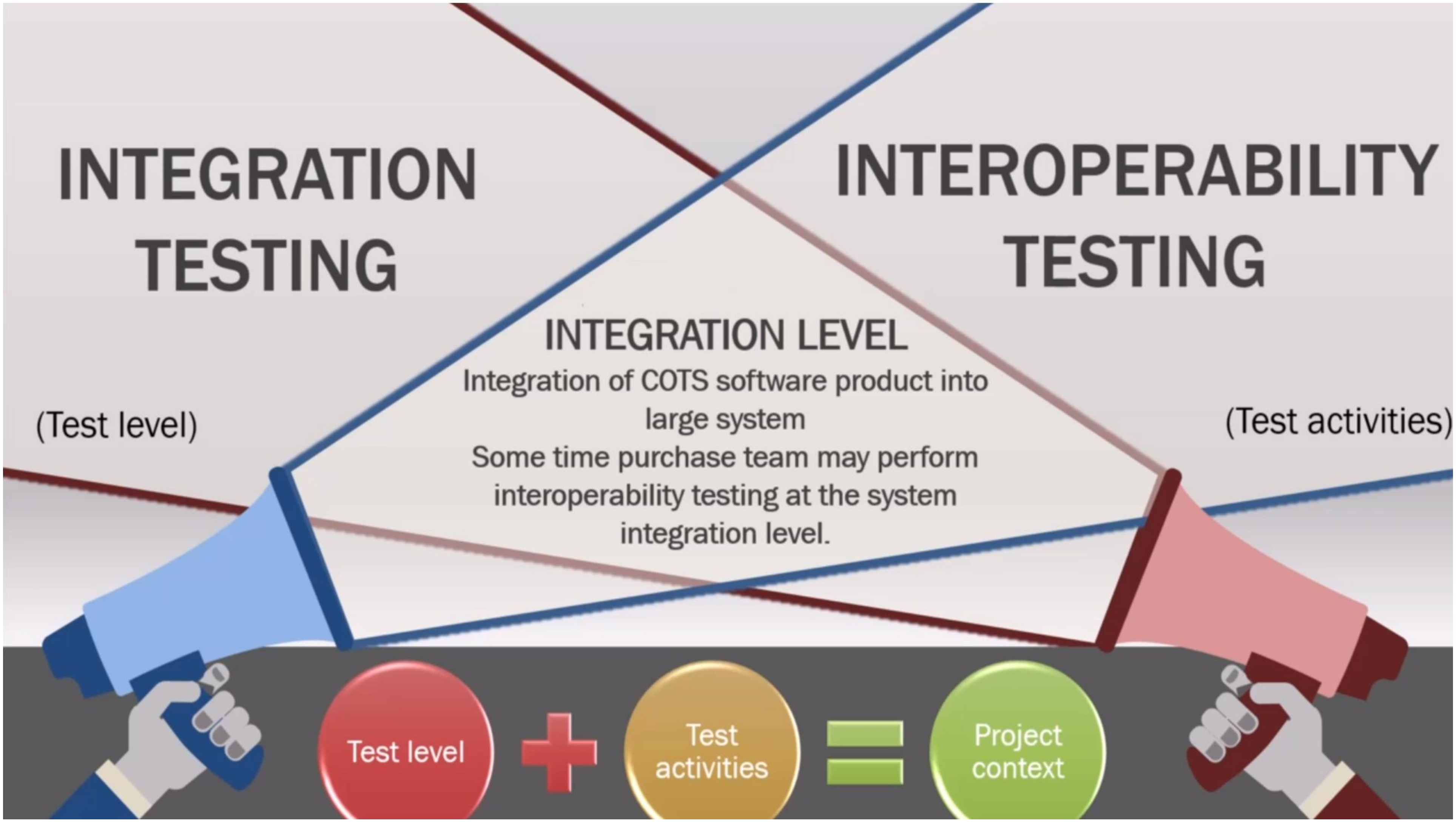
Project and product risk

Project goal/Type

Project context

Business proprieties

Time to market



INTEGRATION TESTING

INTEROPERABILITY TESTING

(Test level)

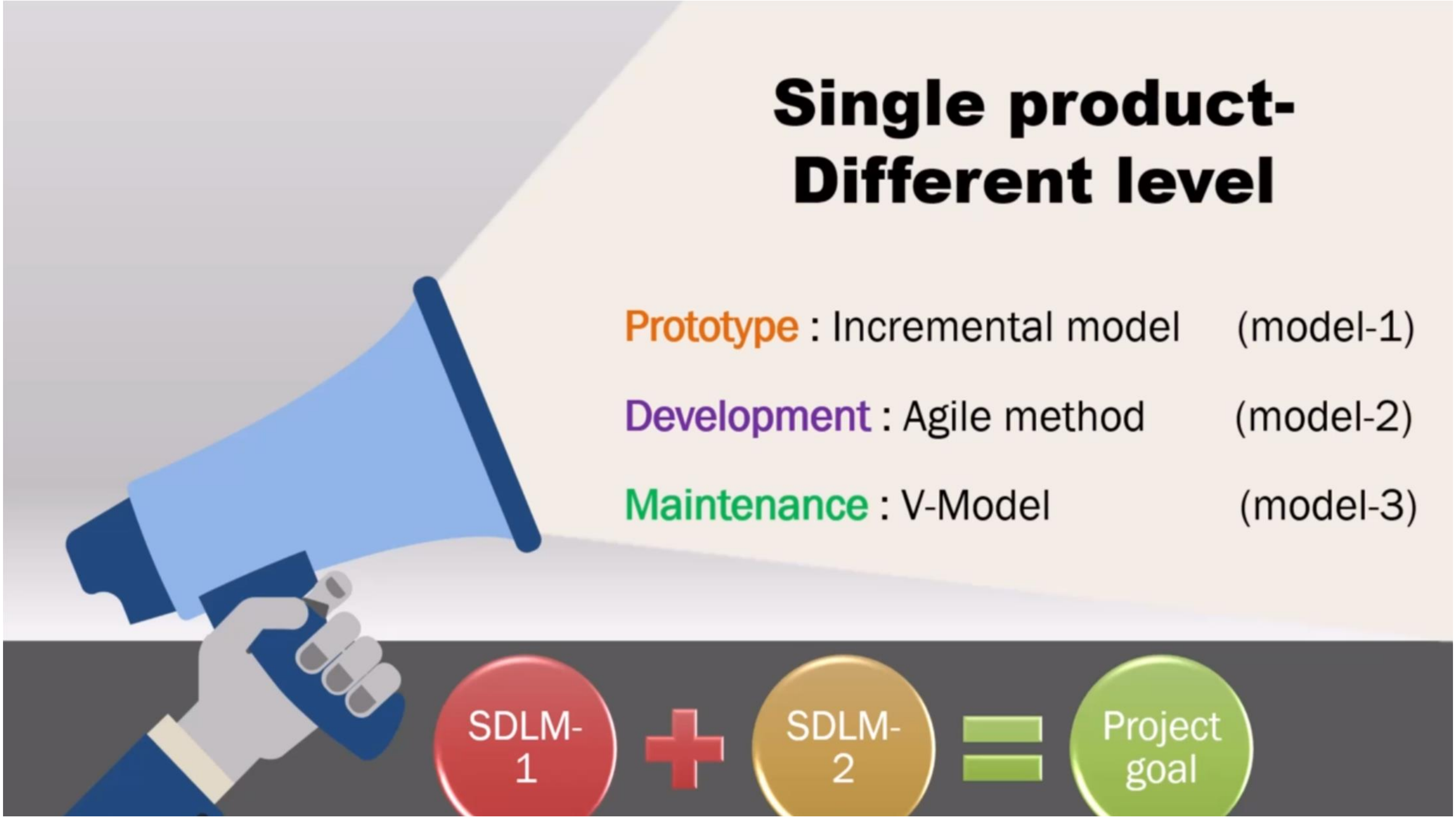
(Test activities)

INTEGRATION LEVEL

Integration of COTS software product into large system

Some time purchase team may perform interoperability testing at the system integration level.

Test level + Test activities = Project context



Single product- Different level

Prototype : Incremental model (model-1)

Development : Agile method (model-2)

Maintenance : V-Model (model-3)

SDLM-1 + SDLM-2 = Project goal

Single product - different object

Device Product Service

For each object they may have separate SDLM.

SDLM-1 + SDLM-2 = Project goal

hillel
компьютерная школа

ISTQB®

V-Model is:

A. A software development model that illustrates how testing activities integrate with software development phases
B. A software life-cycle model that is not relevant for testing
C. The official software development and testing life-cycle model of ISTQB
D. A testing life cycle model including unit, integration, system and acceptance phases

V-Model is:

A. **A software development model that illustrates how testing activities integrate with software development phases**

B. A software life-cycle model that is not relevant for testing

C. The official software development and testing life-cycle model of ISTQB

D. A testing life cycle model including unit, integration, system and acceptance phases

Which of the following is true of the V-model?

A. It states that modules are tested against user requirements.
B. It only models the testing phase.
C. It specifies the test techniques to be used.
D. It includes the verification of designs.

Which of the following is true of the V-model?

A. It states that modules are tested against user requirements.
B. It only models the testing phase.
C. It specifies the test techniques to be used.
D. It includes the verification of designs.

Which test levels are USUALLY included in the common type of V-model?

A. Integration testing, system testing, acceptance testing, and regression testing
B. Component testing, integration testing, system testing, and acceptance testing
C. Incremental testing, exhaustive testing, exploratory testing, and data driven testing
D. Alpha testing, beta testing, black-box testing, and white-box testing

Which test levels are USUALLY included in the common type of V-model?

A. Integration testing, system testing, acceptance testing, and regression testing

B. Component testing, integration testing, system testing, and acceptance testing

C. Incremental testing, exhaustive testing, exploratory testing, and data driven testing

D. Alpha testing, beta testing, black-box testing, and white-box testing

What is the MAIN benefit of designing tests early in the life cycle?

A. It is cheaper than designing tests during the test phases
B. It helps prevent defects from being introduced into the code
C. Tests designed early are more effective than tests designed later
D. It saves time during the testing phases when testers are busy

What is the MAIN benefit of designing tests early in the life cycle?

A. It is cheaper than designing tests during the test phases
B. It helps prevent defects from being introduced into the code
C. Tests designed early are more effective than tests designed later
D. It saves time during the testing phases when testers are busy

A software model that can’t be used in functional testing

A. Process flow model
B. State transaction model
C. Menu structure model
D. Plain language specification model

hillel
компьютерная
школа

ISTQB®

A software model that can’t be used in functional testing

A. Process flow model
B. State transaction model
C. Menu structure model
D. Plain language specification model

Majority of system errors occur in the __________ phase

A. Requirements Phase.
B. Analysis and Design Phase
C. Development Phase
D. Testing Phase

Majority of system errors occur in the __________ phase

A. **Requirements Phase.**
B. Analysis and Design Phase
C. Development Phase
D. Testing Phase

In any software development life cycle (SDLC) model, which of the following are characteristics of good testing?

I. Providing complete test coverage of all branches of the system code.

II. Having a corresponding testing activity for each development activity.

III. Testers should be involved in reviewing documents as soon as drafts are available.

IV. Each test level has test objectives specific to that level.

A. II, III and IV
B. I and III
C. I, III and IV
D. I and II

In any software development life cycle (SDLC) model, which of the following are characteristics of good testing?

I. Providing complete test coverage of all branches of the system code.

II. Having a corresponding testing activity for each development activity.

III. Testers should be involved in reviewing documents as soon as drafts are available.

IV. Each test level has test objectives specific to that level.

A. **II, III and IV**
B. I and III
C. I, III and IV
D. I and II

Match every stage of the software Development Life cycle with the Testing Life cycle:

i. Hi-level design a Unit tests
ii. Code b Acceptance
tests
iii. Low-level design c System tests
iv. Business requirements d Integration
tests

A. i-d , ii-a , iii-c , iv-b
B. i-c , ii-d , iii-a , iv-b
C. i-b , ii-a , iii-d , iv-c
D. i-c , ii-a , iii-d , iv-b

Match every stage of the software Development Life cycle with the Testing Life cycle:

i. Hi-level design **a Unit tests**
ii. Code **b Acceptance**
tests
iii. Low-level design **c System tests**
iv. Business requirements **d Integration**
tests

A. i-d , ii-a , iii-c , iv-b
B. i-c , ii-d , iii-a , iv-b
C. i-b , ii-a , iii-d , iv-c
D. i-c , ii-a , iii-d , iv-b

Which test may OPTIONALLY be included in the common type of the V-model?

A. Component (unit) testing
B. Acceptance testing
C. System integration testing
D. Validation and verification

Which test may OPTIONALLY be included in the common type of the V-model?

A. Component (unit) testing
B. Acceptance testing
C. System integration testing
D. Validation and verification

Which of the following is a characteristic of good testing in any life cycle model?

A. All document reviews involve the development team.
B. Some, but not all, development activities have corresponding test activities.
C. Each test level has test objectives specific to that level.
D. Analysis and design of tests begins as soon as development is complete.

hillel
компьютерная школа

ISTQB®

Which of the following is a characteristic of good testing in any life cycle model?

A. All document reviews involve the development team.
B. Some, but not all, development activities have corresponding test activities.
C. Each test level has test objectives specific to that level.
D. Analysis and design of tests begins as soon as development is complete.