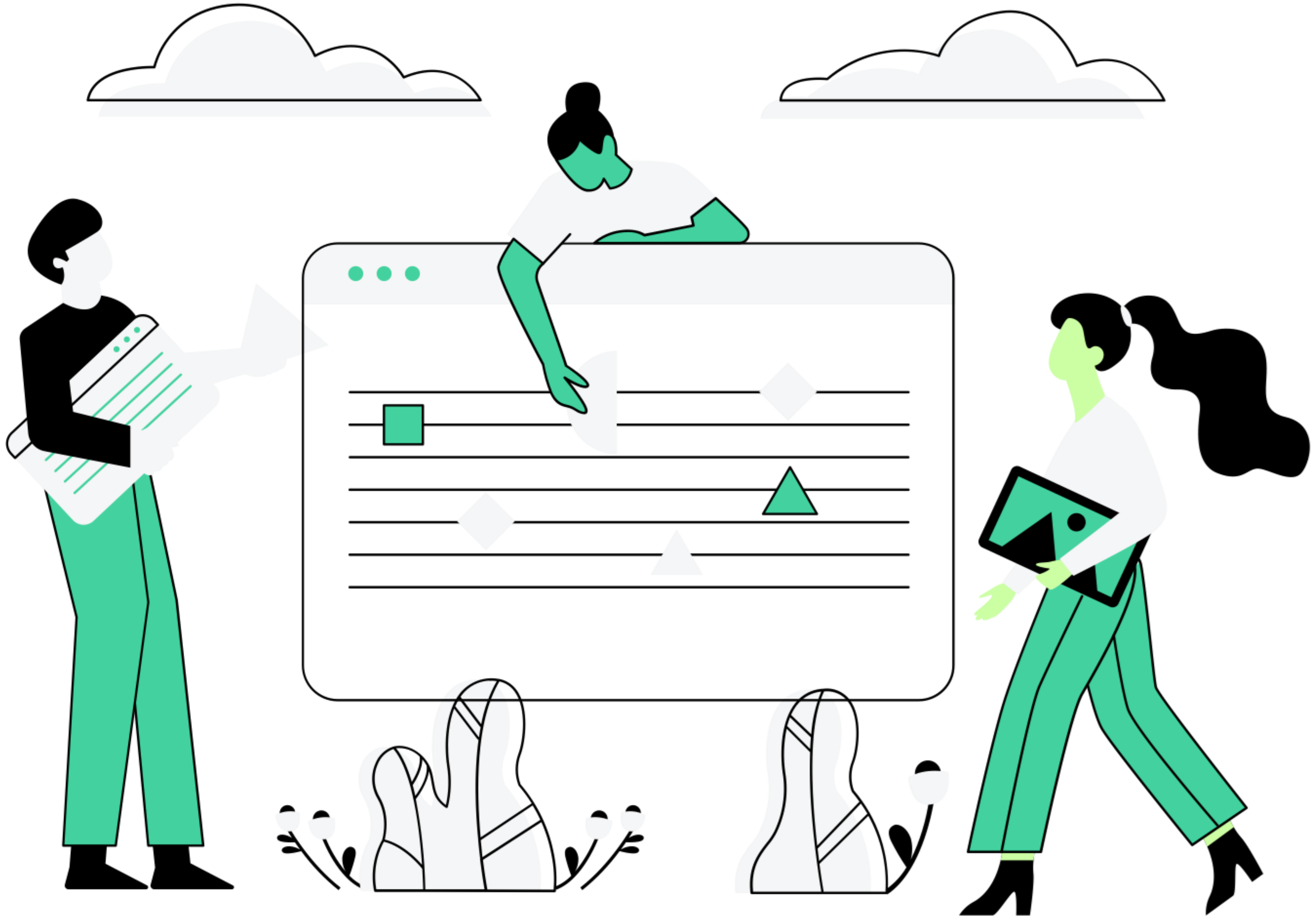
Software Methodology & Framework

Department of Computer Engineering
Sharif University of Technology
Maryam Ramezani
maryam.ramezani@sharif.edu

Table of Contents



01 CMM

02 Development Methodologies & Framework

03 Methodology vs Framework

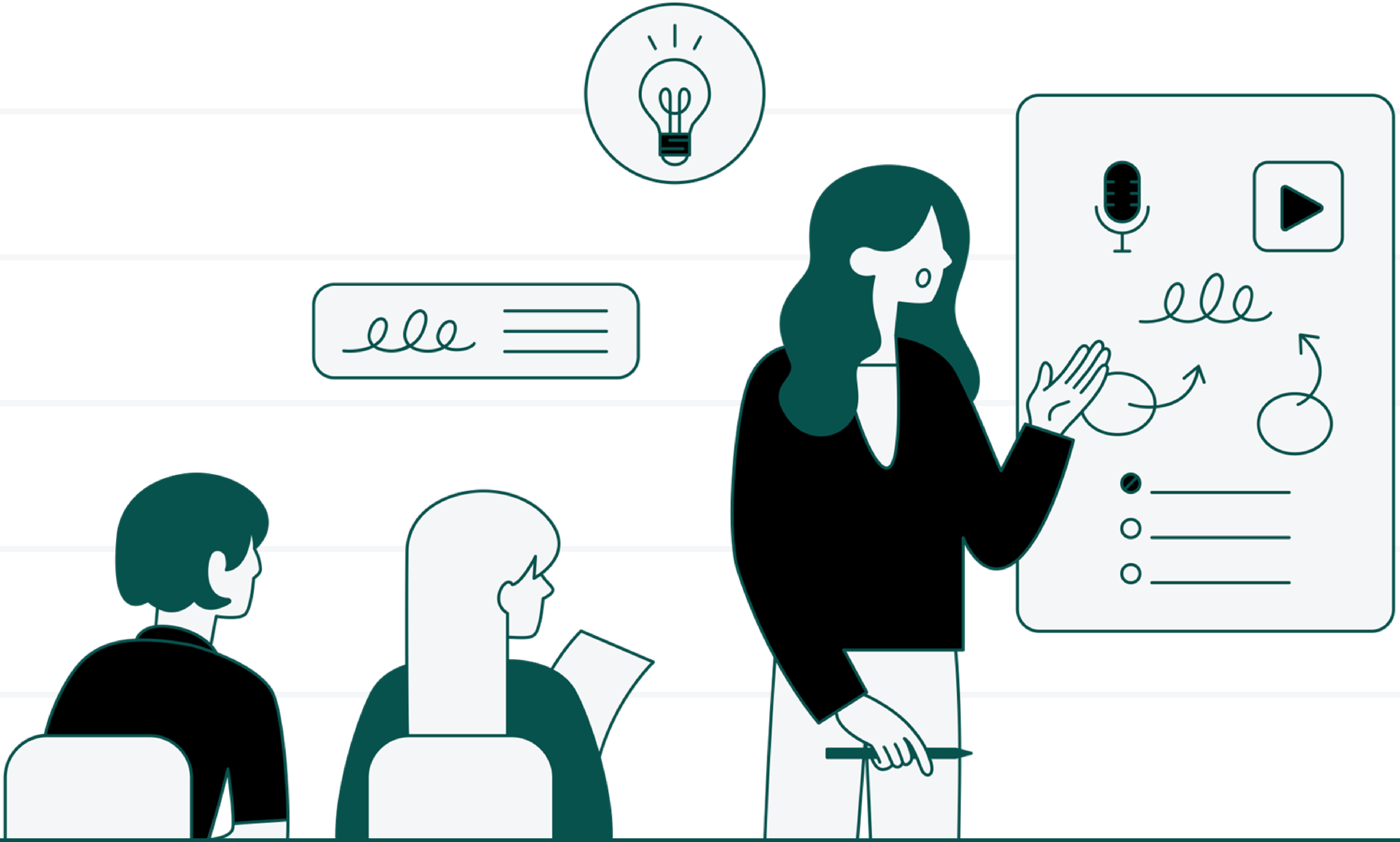
04 Structured-Waterfall

05 Structured-Parallel

- Project Charter

06 Structured-V Model

07 RAD



01

CMM

Organizational Maturity

- Organizational maturity is the measure of the quality of a company's operations. A company with a high maturity level can face challenges and seize upon opportunities. Improving organizational maturity is a gradual process that emphasizes internally driven improvement.
- In an organization, there are six criteria for measuring organizational maturity:
  - Objective
  - Process
  - Awareness
  - Responsibility
  - Tools
  - Skill.

Organizational Maturity

- **Objective:** Refers to the clarity and specificity of the organization’s goals. Are the objectives clearly defined, documented, and measurable?
  - **Importance**: Clear objectives provide direction for the organization and help employees understand what they are working towards. Without well-defined objectives, measuring success and progress becomes challenging.
- **Process:** This criterion refers to how tasks are performed and the structure of operations. Are the processes standardized, repeatable, and well-defined, or are they ad hoc and unstructured?
  - **Importance**: Well-documented and efficient processes enable the organization to operate systematically and continuously improve its performance. It reflects how well-organized and efficient the organization’s operations are.

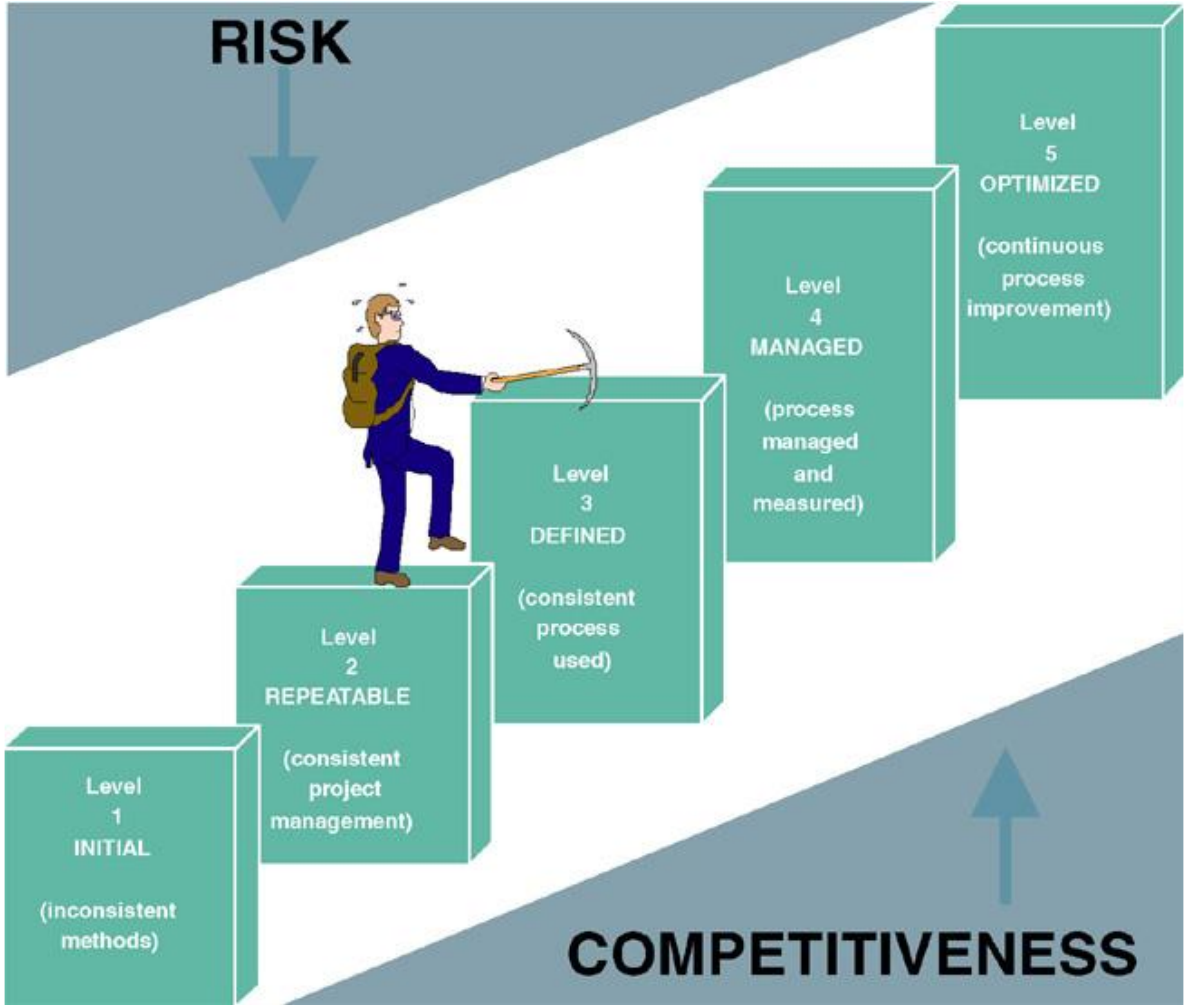
Organizational Maturity

- ❑ **Awareness:** Awareness assesses whether individuals within the organization are informed about its objectives, processes, and needs. Are they aware of their roles, responsibilities, and what is expected of them?
  - ○ **Importance**: When employees and management are aware of the organization's goals and processes, they can work more effectively towards those objectives. This fosters synergy and more effective collaboration within the organization.
- ❑ **Responsibility:** This measures how clearly roles and responsibilities are assigned within the organization. Do individuals know what tasks they are responsible for, and are they held accountable for those tasks?
  - ○ **Importance**: Clear delegation of responsibilities ensures that everyone understands their role, which increases transparency and productivity. Responsibility also fosters greater commitment and accountability in fulfilling tasks.

Organizational Maturity

- **Tools:** refer to the technologies, software, and equipment available to carry out tasks and achieve organizational goals. Does the organization have access to the right tools to work effectively?
  - **Importance**: Using the right tools enhances the speed, accuracy, and efficiency of operations. Organizations that leverage modern tools and technology tend to be more efficient and successful.
- **Skill:** refers to the knowledge and capabilities of the people in the organization. Do employees and managers possess the necessary skills to perform their tasks effectively?
  - **Importance**: A skilled workforce is crucial for the success of any organization. Having the right skills in place improves the quality and efficiency of processes and increases the organization’s competitiveness.

Capability Maturity Model (CMM)


RISK
Level 5 OPTIMIZED (continuous process improvement)
Level 4 MANAGED (process managed and measured)
Level 3 DEFINED (consistent process used)
Level 2 REPEATABLE (consistent project management)
Level 1 INITIAL (inconsistent methods)
COMPETITIVENESS

CMM Process Management Model

Capability Maturity Model (CMM) – a standardized framework for assessing the maturity level of an organization's information system development and management processes and products. It consists of five levels of maturity:

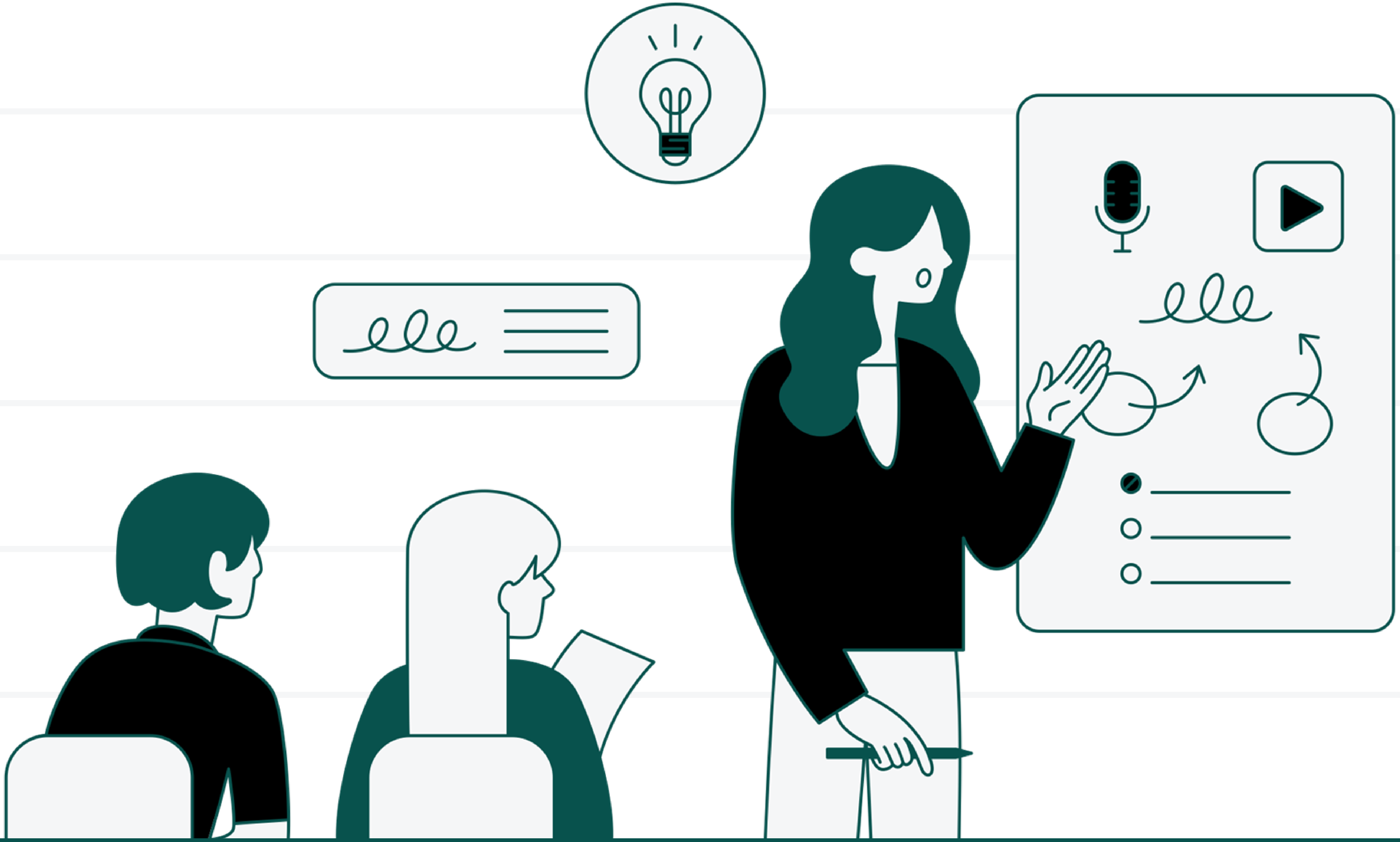
- ❑ Level 1—Initial: System development projects follow no prescribed process. Processes unpredictable. Poor controlled and reactive.

- ❑ Level 2—Repeatable: Project management processes and practices established to track project costs, schedules, and functionality. Processes characterized for projects and is often reactive.
  - ○ No documentation and no view on future.
  - ○ At this stage, the goals of the processes are limited to a department or business unit, and there is usually very little or no executive support.

CMM Process Management Model

Capability Maturity Model (CMM) – a standardized framework for assessing the maturity level of an organization’s information system development and management processes and products. It consists of five levels of maturity:

- ❑ Level 3—Defined: Standard system development process (methodology) is purchased or developed. All projects use a version of this process.
- ❑ Level 4—Managed: Measurable goals for quality and productivity are established. KPI: Key Performance Indicator.
- ❑ Level 5—Optimizing: The standardized system development process is continuously monitored and improved based on measures and data analysis established in Level 4.

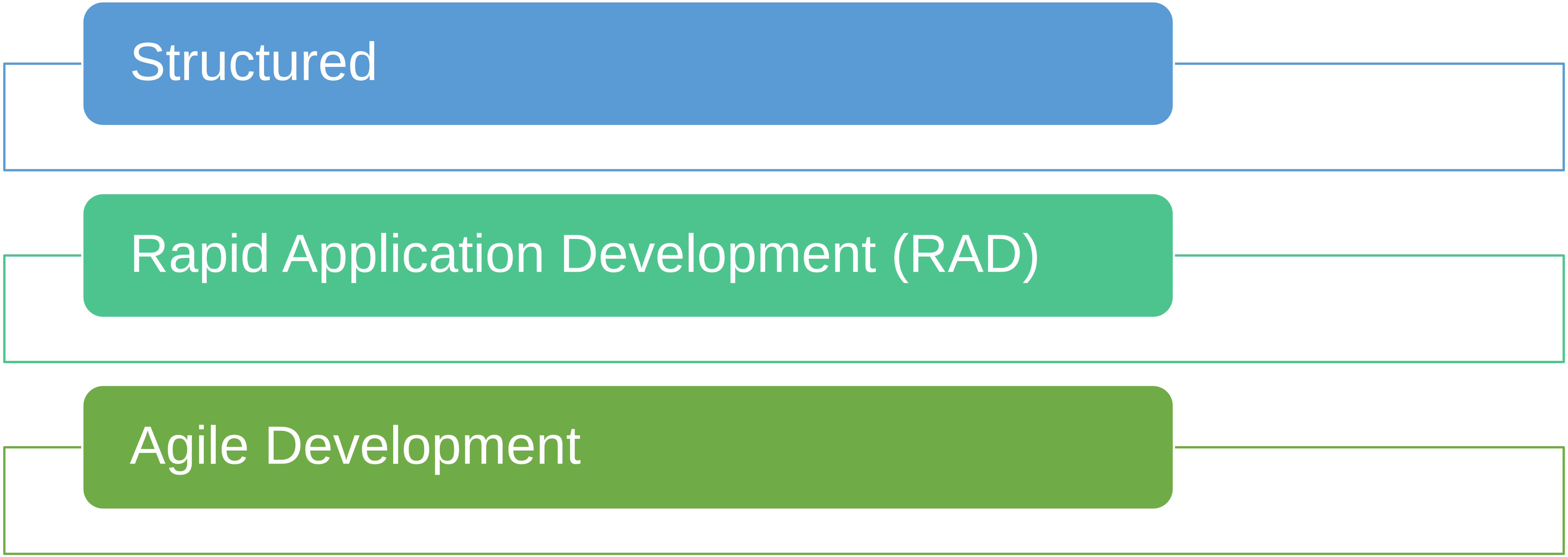


02 Development Methodologies & Framework

Development Methodology

- ❑ A Formalized approach to the systems development process;
- ❑ A standardized development process that defines (as in CMM Level 3) a set of
  - → activities,
  - → methods,
  - → best practices,
  - → deliverables, and
  - → automated tools
- ❑ System developers and *project managers* are to use to *develop* and continuously *improve* information systems and software.
- ❑ SDLC Model (Software Development Life Cycle): A framework that describes the activities performed at each stage of a software development project.

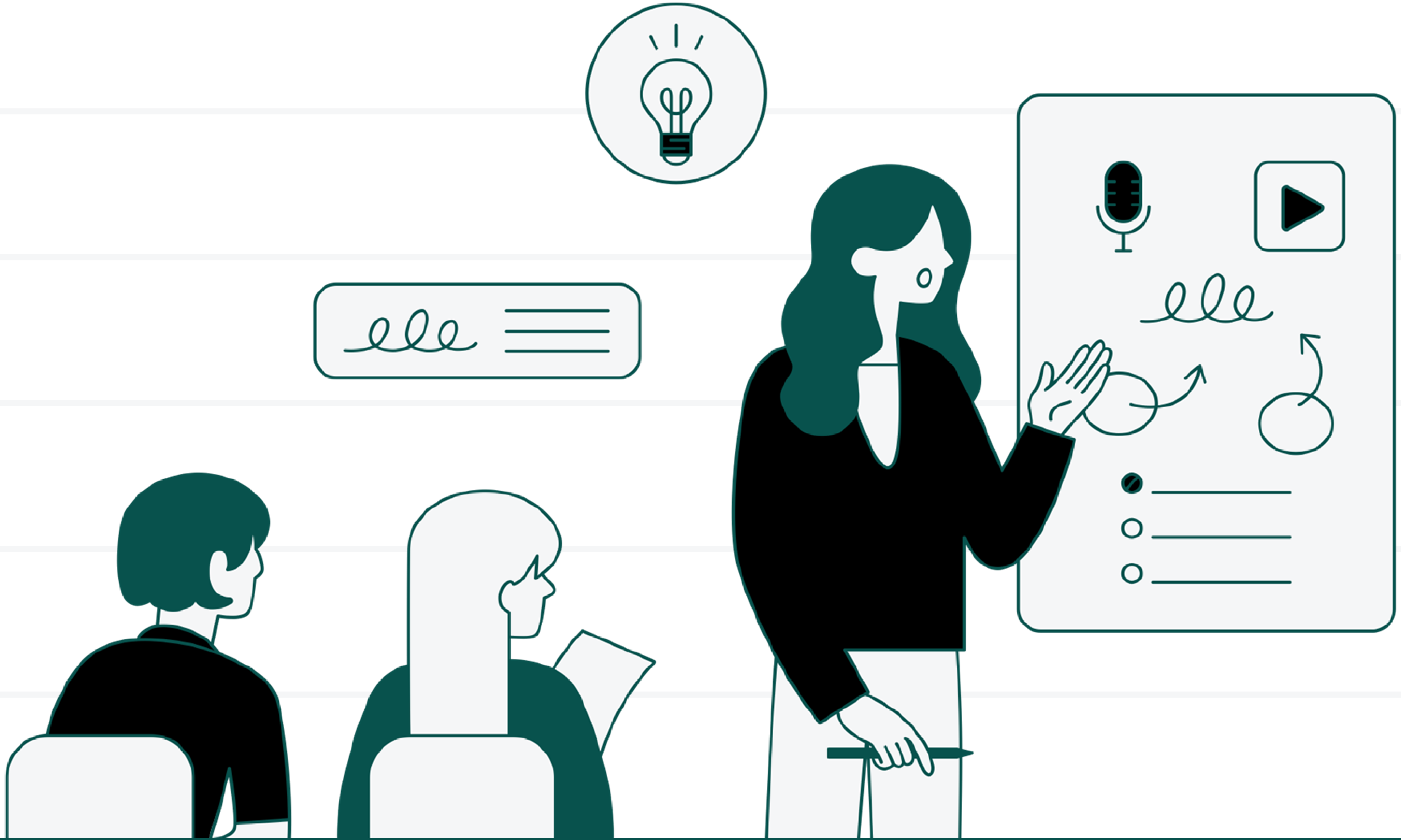
Development Methodologies & Framework



Structured

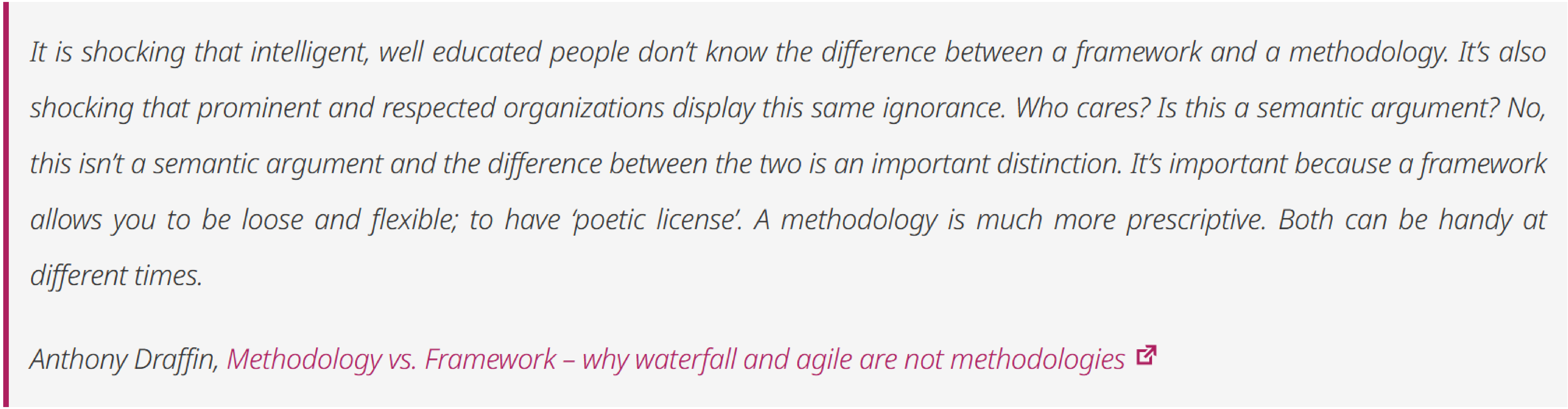
Rapid Application Development (RAD)

Agile Development



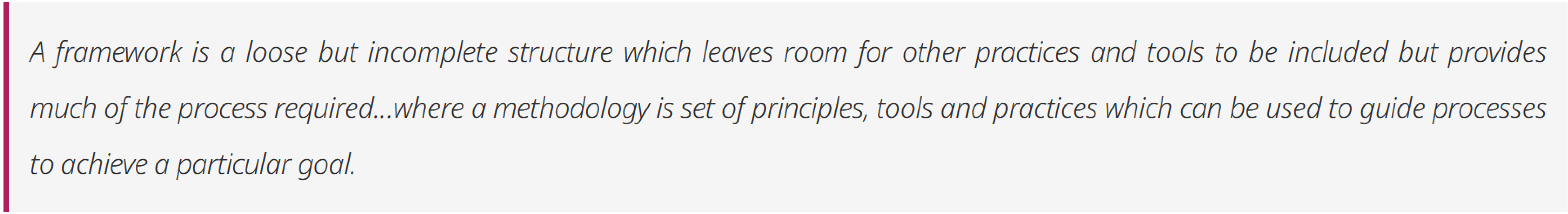
03 Methodology vs Framework

Introduction

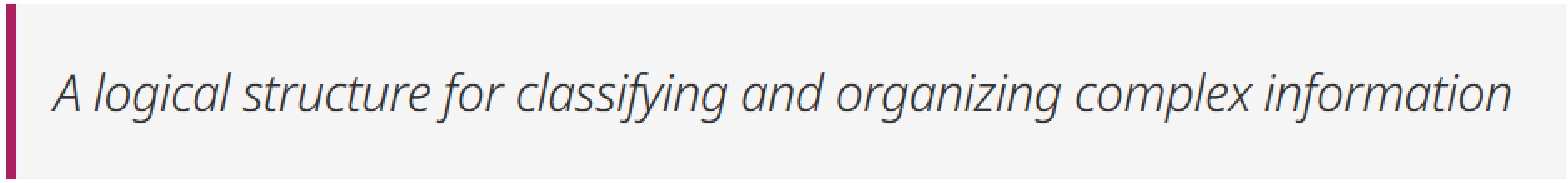


> *It is shocking that intelligent, well educated people don't know the difference between a framework and a methodology. It's also shocking that prominent and respected organizations display this same ignorance. Who cares? Is this a semantic argument? No, this isn't a semantic argument and the difference between the two is an important distinction. It's important because a framework allows you to be loose and flexible; to have 'poetic license'. A methodology is much more prescriptive. Both can be handy at different times.*
>
> *Anthony Draffin, Methodology vs. Framework – why waterfall and agile are not methodologies*

Frameworks



> *A framework is a loose but incomplete structure which leaves room for other practices and tools to be included but provides much of the process required…where a methodology is set of principles, tools and practices which can be used to guide processes to achieve a particular goal.*

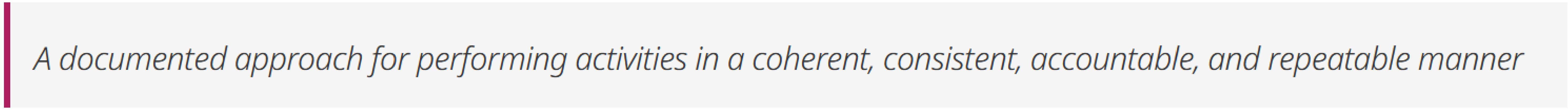


> *A logical structure for classifying and organizing complex information*

- ❑ A framework is a picture or a model that guides you to understand which artifacts you should produce when. It doesn’t tell you what to do though. In other words, a framework is a structured problem-solving approach. In many ways, it helps focus your attention on certain aspects of a problem. A framework is a general guideline that an organization can adopt. It could include many components.

Methodologies

- Methodologies, on the other hand, are stringent practices for accomplishing an outcome. While frameworks are more fluid, allowing consultants to pick and choose according to their needs, methodologies are rather stringent. The Treasury Enterprise Architecture Framework defines a methodology as:



> *A documented approach for performing activities in a coherent, consistent, accountable, and repeatable manner*

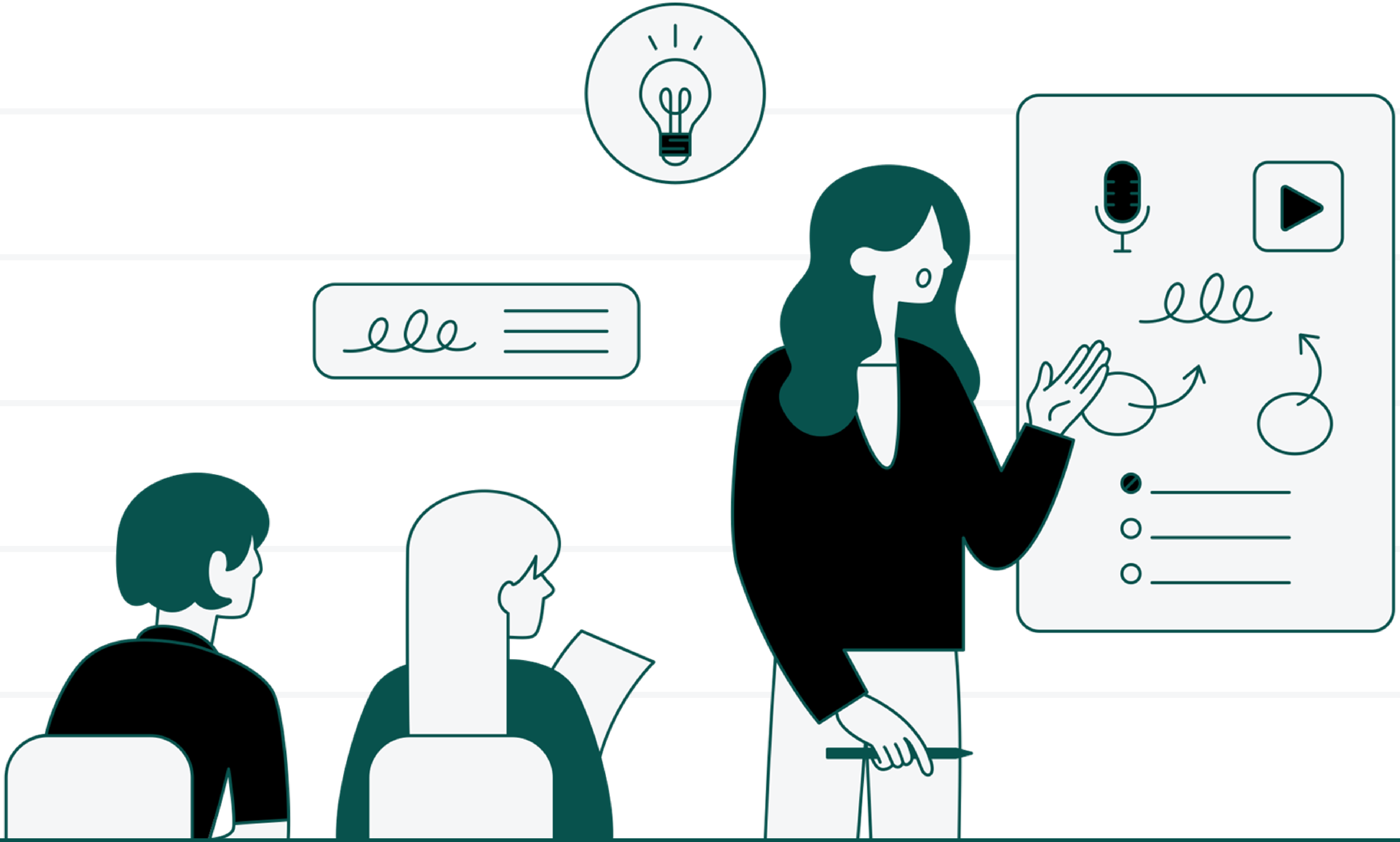
Methodology Vs Framework

- ❑ A methodology is a way to systematically solve a problem. It is a combination of two things together – the methods you've chosen to get to a desired outcome and the logic behind those methods. On the other hand, a framework is a structured approach to problem solving. Frameworks provide the structural components you need to implement a model. It is a skeletal structure around which something can be built.

- ❑ A framework is a collection of reusable components that offer a consultant shortcuts to avoid developing a structure from scratch, each time they start an engagement. Some frameworks take a more rigid approach to consulting, while others are more lenient throughout the engagement lifecycle. Such frameworks provide enough room for creativity, allowing consultants to cherry pick components according to their clients' needs or style of work. Methodology, on the other hand, has its limitations in terms of creativity because it is based on a set of pre-defined rules.

Practice

Consider managing a restaurant. Propose both a methodology and a framework for it.



04 Structured-Waterfall

Structured Method

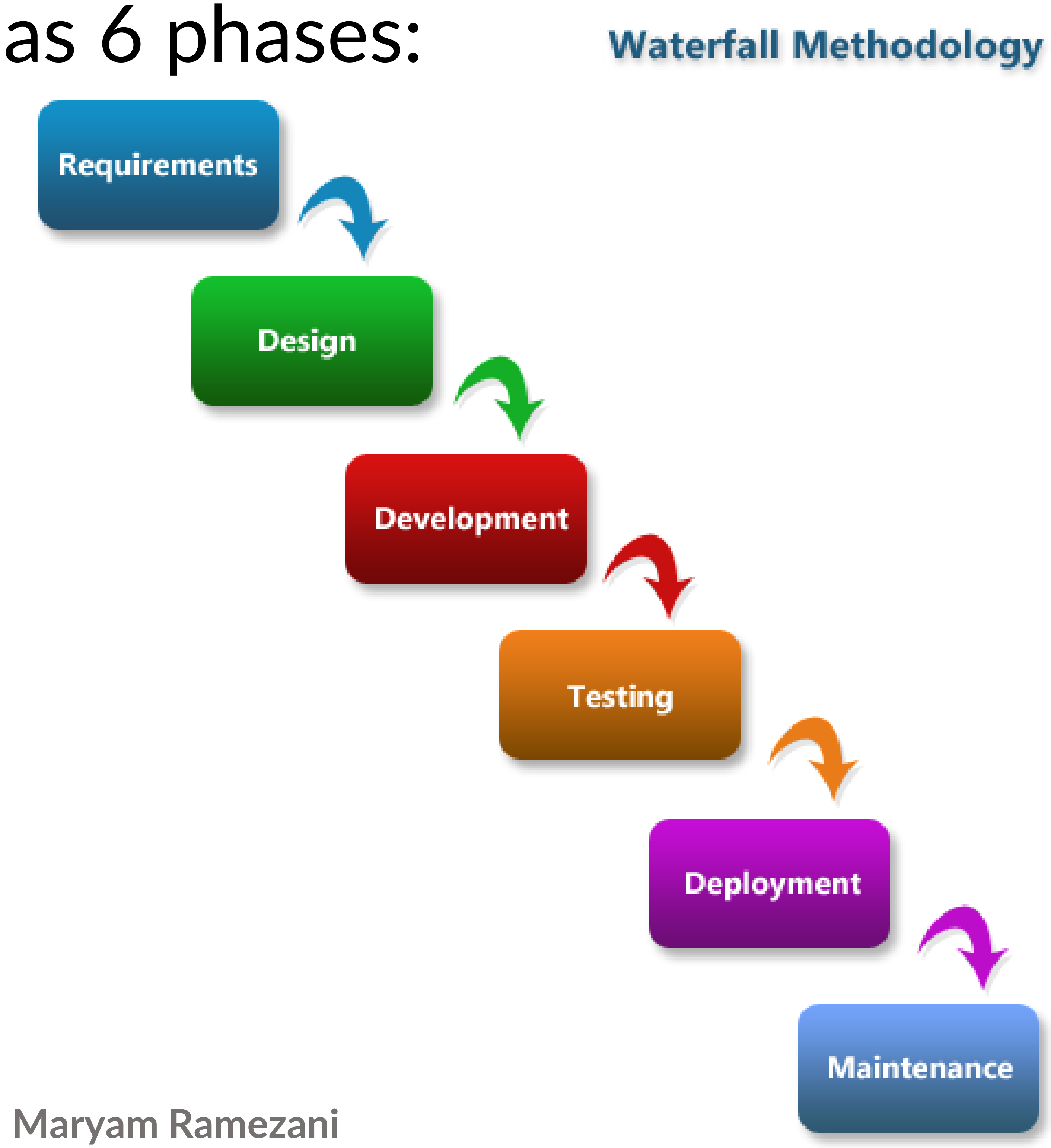
- Assume a project phase is complete before moving to the next phase
- Goal: Doing each phase thoroughly before moving forward ensures correct and high-quality outcomes.
  - Waterfall development
  - Parallel development
  - V-model

Waterfall Development

- A project management methodology based on a sequential design process.
- Good for small projects whose requirements can be definitively determined.

Waterfall Development

- A simple waterfall model has 6 phases:



Requirements

- ❑ Starts with the concept, or the idea of what the customer wants to do
- ❑ Project manager interaction with customers, experts, and other stockholders.
- ❑ Gathering requirements:
  - ○ Needs the software will address
  - ○ Problems the software will resolve
  - ○ Functionalities the customer desires

APPROVED

Software Requirements Specification (SRS)

requ

Design

- Includes the logical design and the physical design
- Logical Design:
  - The abstract representation of how the software data flows the inputs and the outputs. It is often shown graphically as a diagram showing data flow.
    - ER
    - UML
    - DFD
    - …

Design

- ❑ Includes the logical design and the physical design
- ❑ Physical Design:
  - ○ Hardware and storage which make the logical design reality once the design is fully complete.

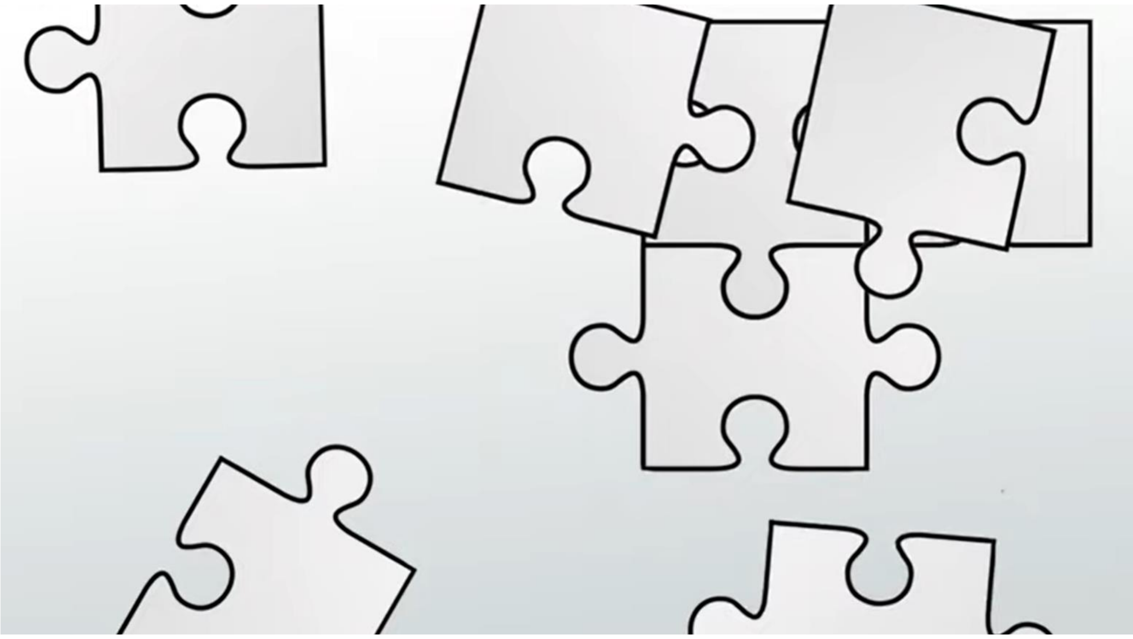
Design

Design Document

Let the implementation begin!!

Development

- ❑ Building the design into actual software
- ❑ Often, software is build in units and integrated into a whole at a later point.
- ❑ As one software unit implementation is complete, it is send to verification (Testing) and another software unit implementation begins.

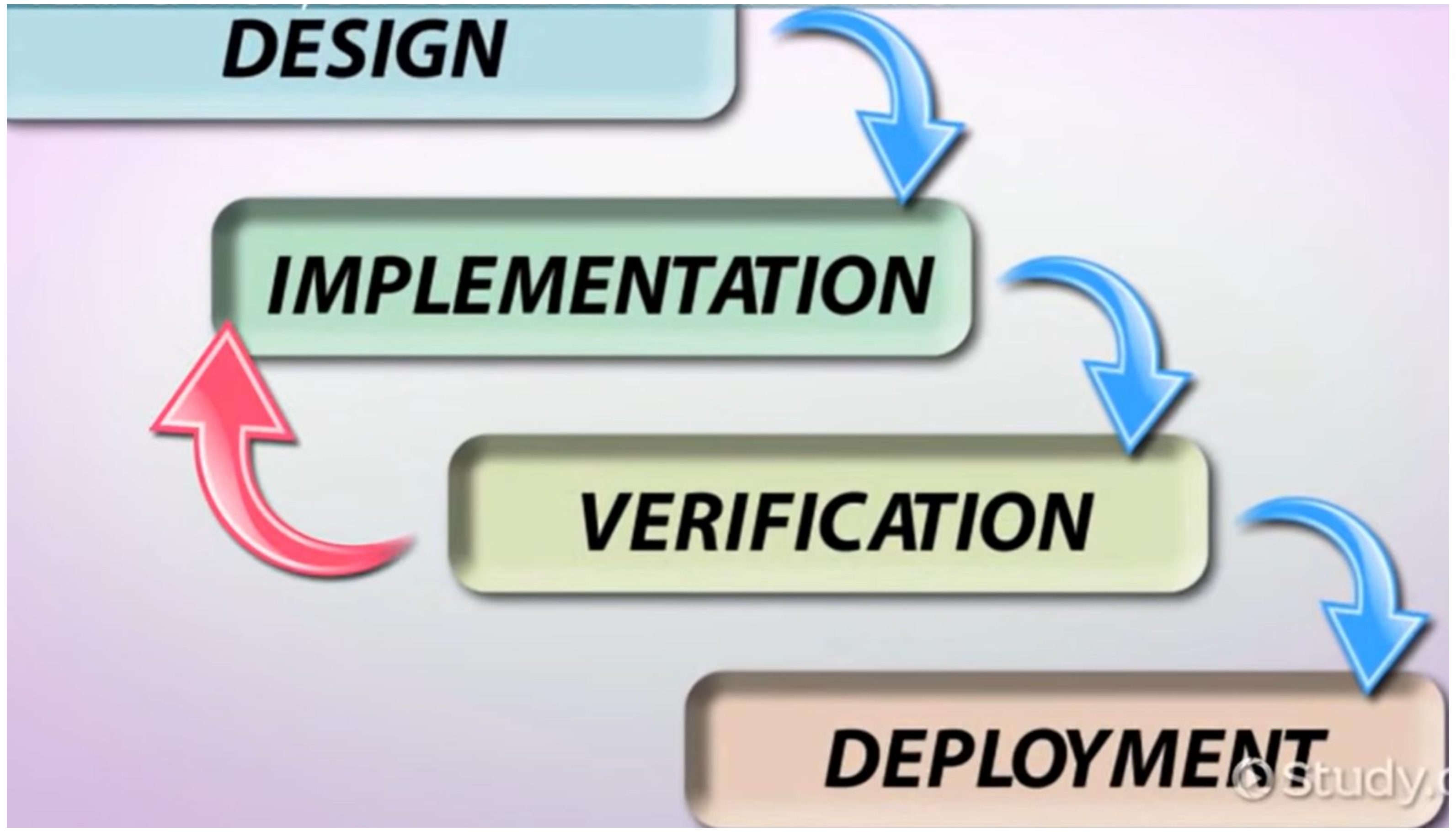


Test (Verification)

- Testing the software against the requirements.

Fail

Pass

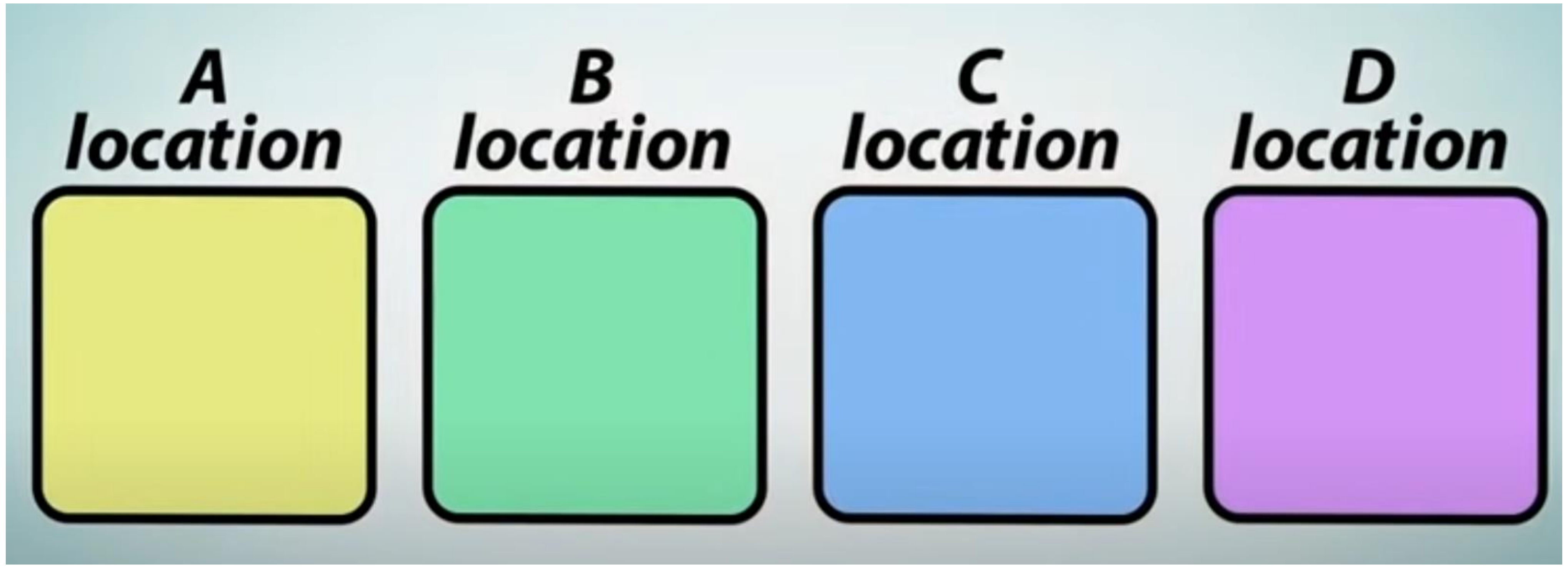


Once all verification actions are done and the phase is deemed complete, the project moves to the next phase.

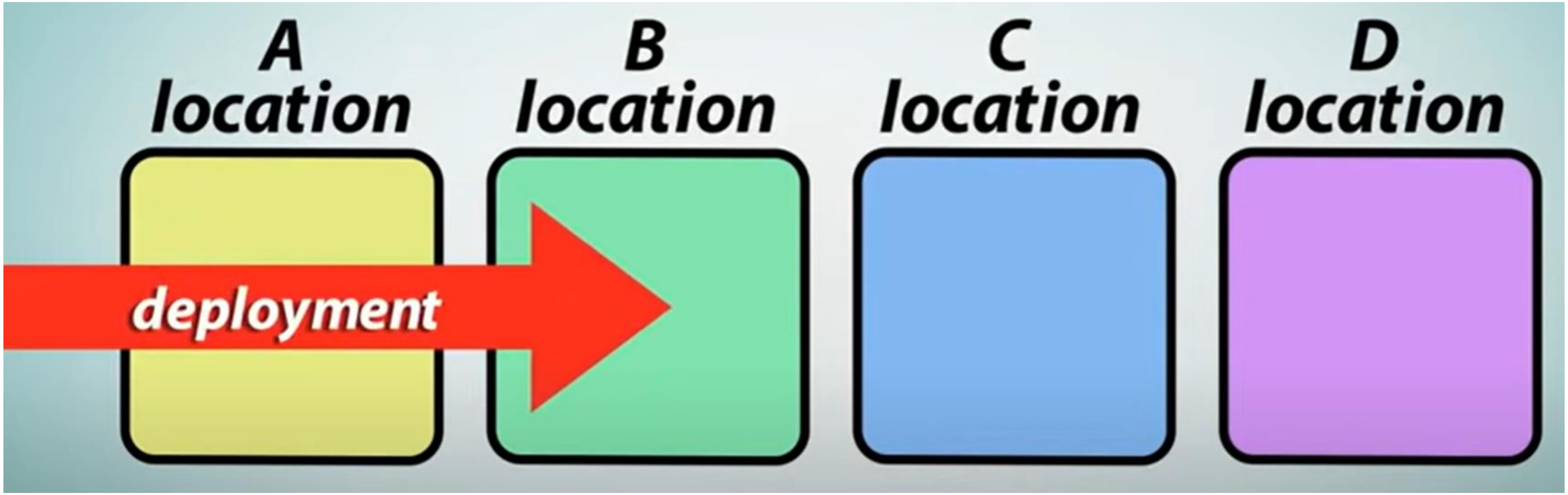
APPROVED
Study.com

Deployment

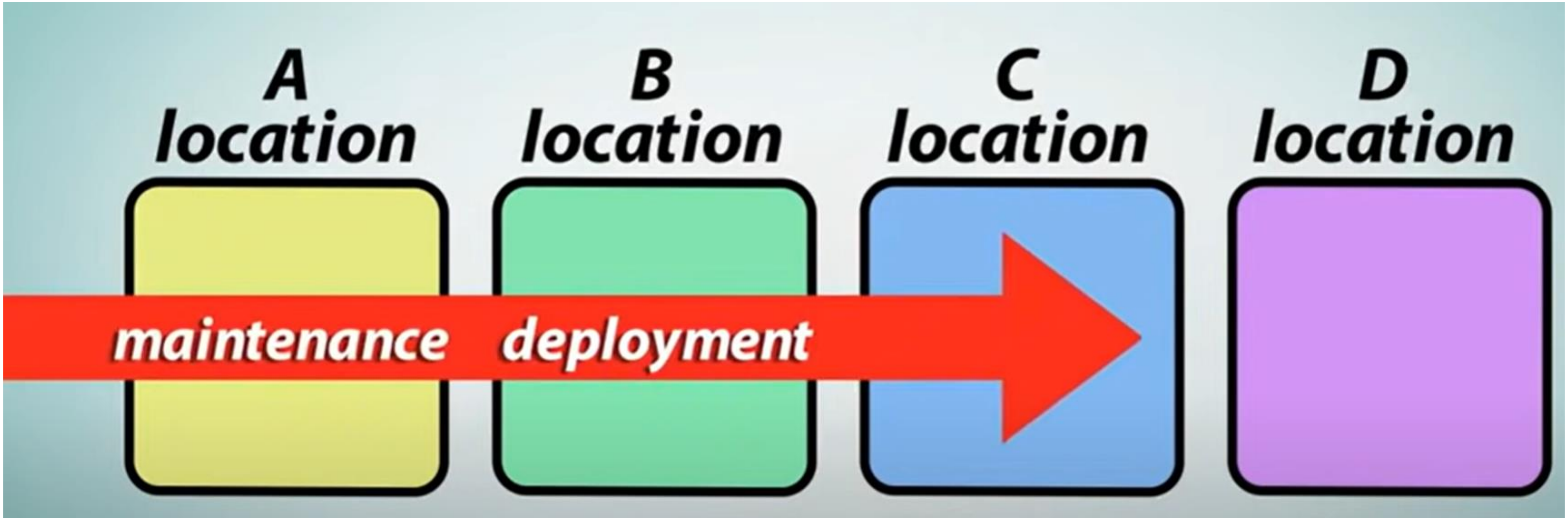
- The actual release of the new software into the information technology environment.
- This phase may be broken into units to allow for ongoing deployment and maintenance.
- Example:



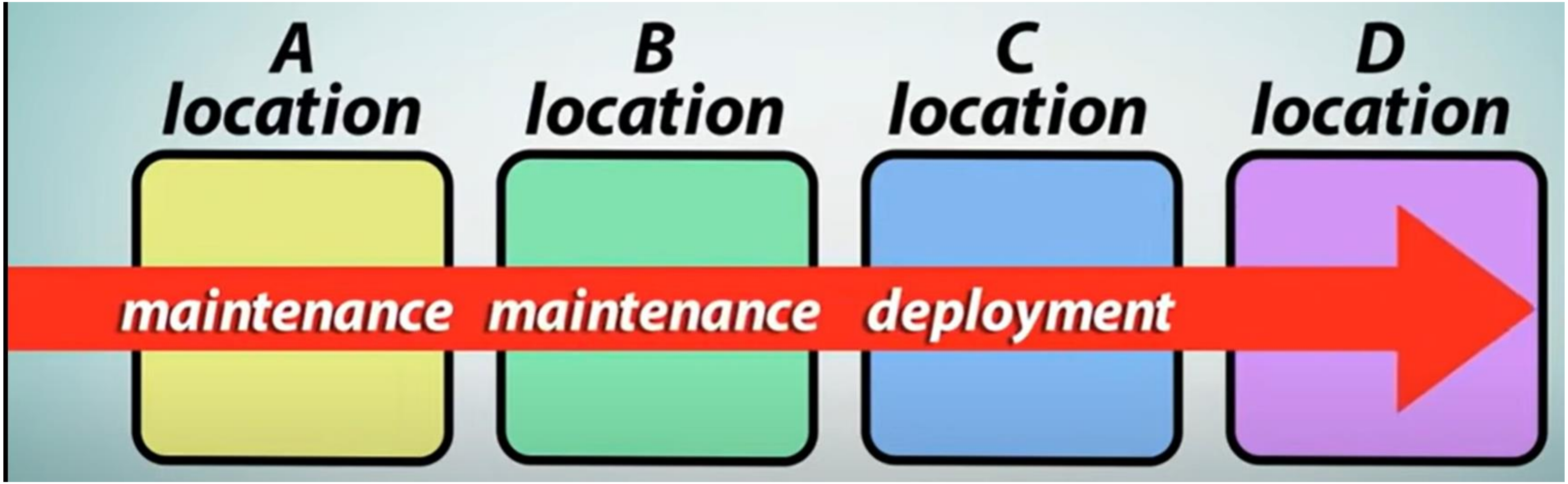
Deployment



Deployment

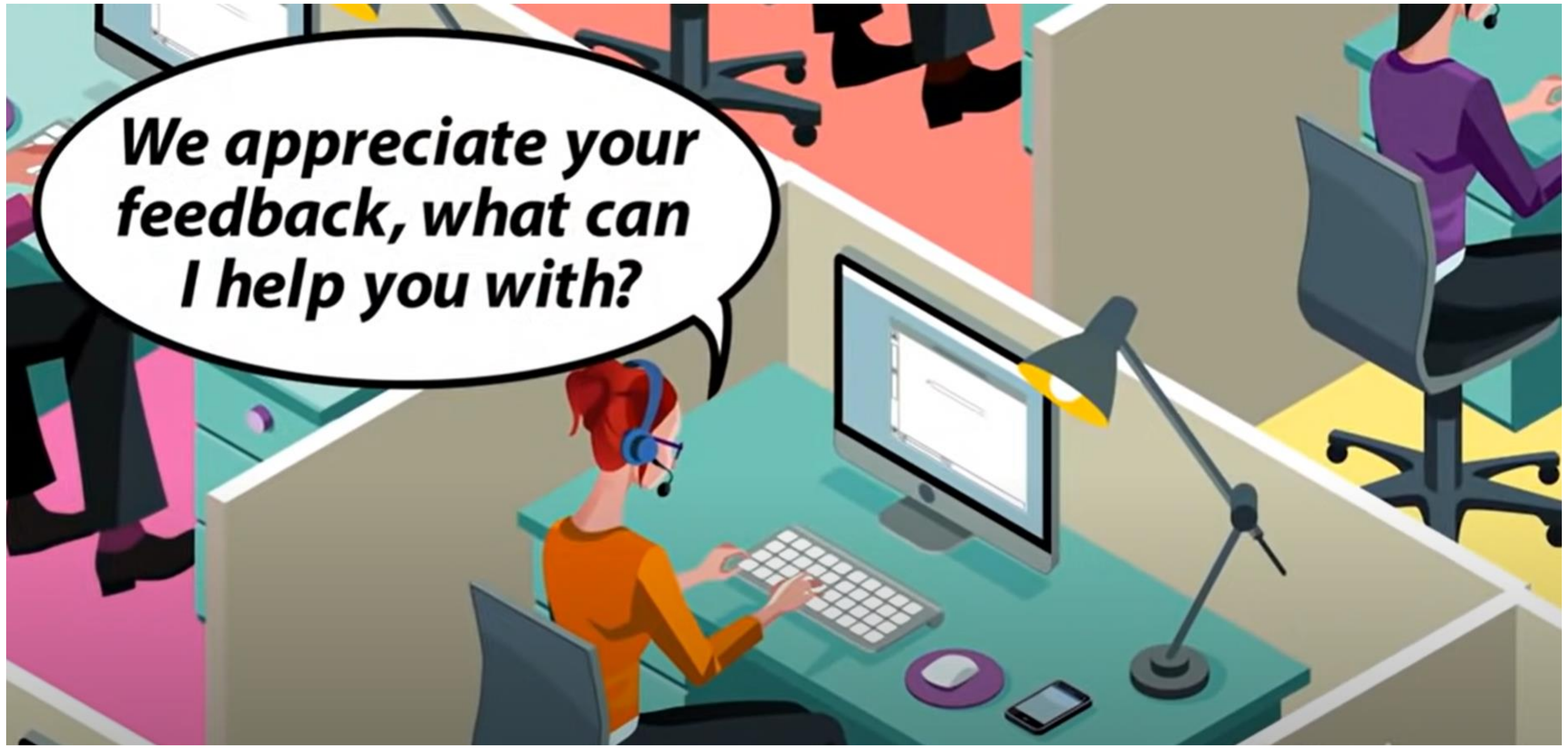

A
location
B
location
C
location
D
location
maintenance
deployment

Deployment


A
location
B
location
C
location
D
location
maintenance
maintenance
deployment

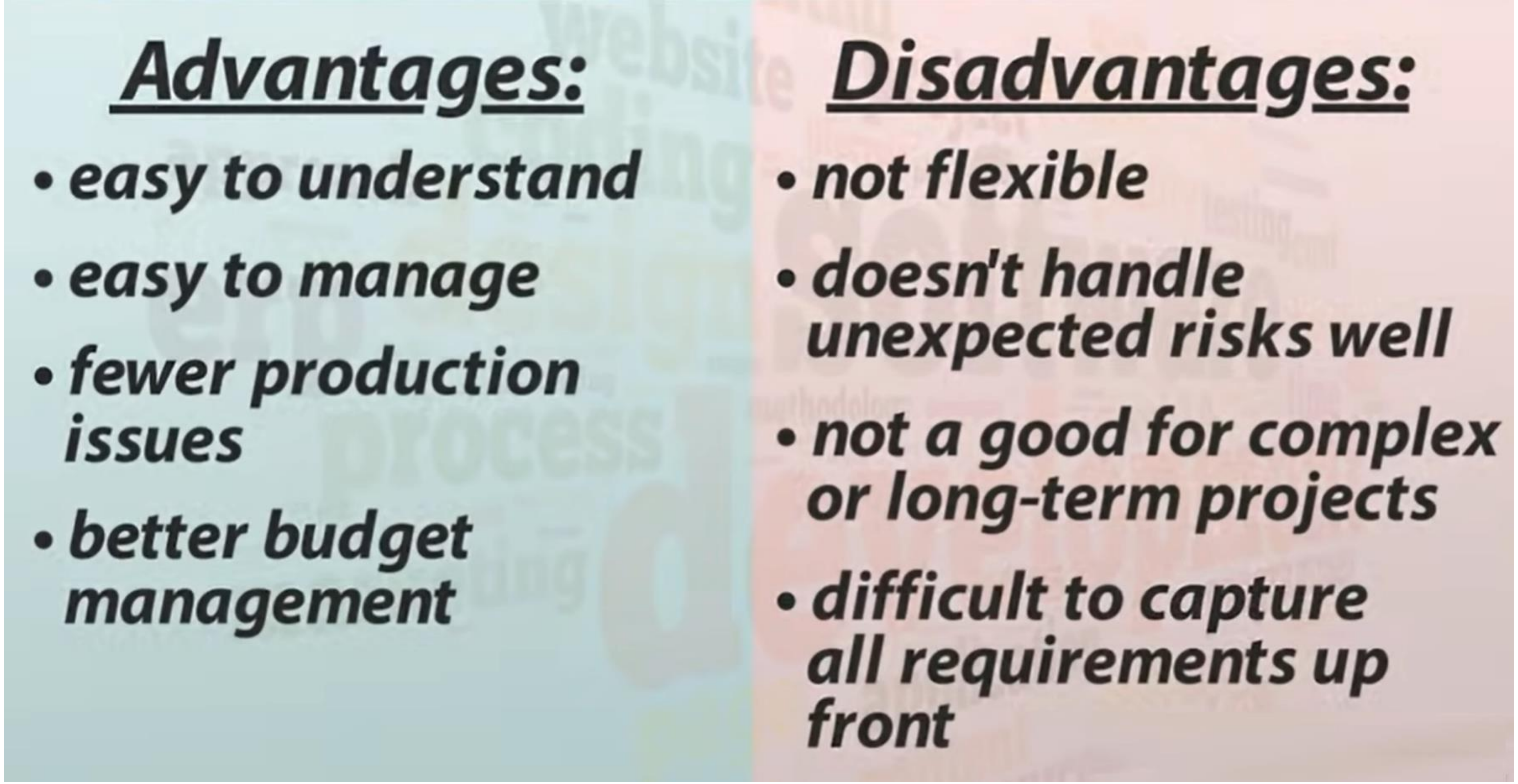
Maintenance

- ❑ The phase of the project in which initial issues are resolved.
- ❑ Live environment to verify acceptable deployment.
- ❑ Customers provide feedback



It's easy to understand and manage.
Has fewer production issues.
And is easier to keep on budget.

Conclusion



Advantages:

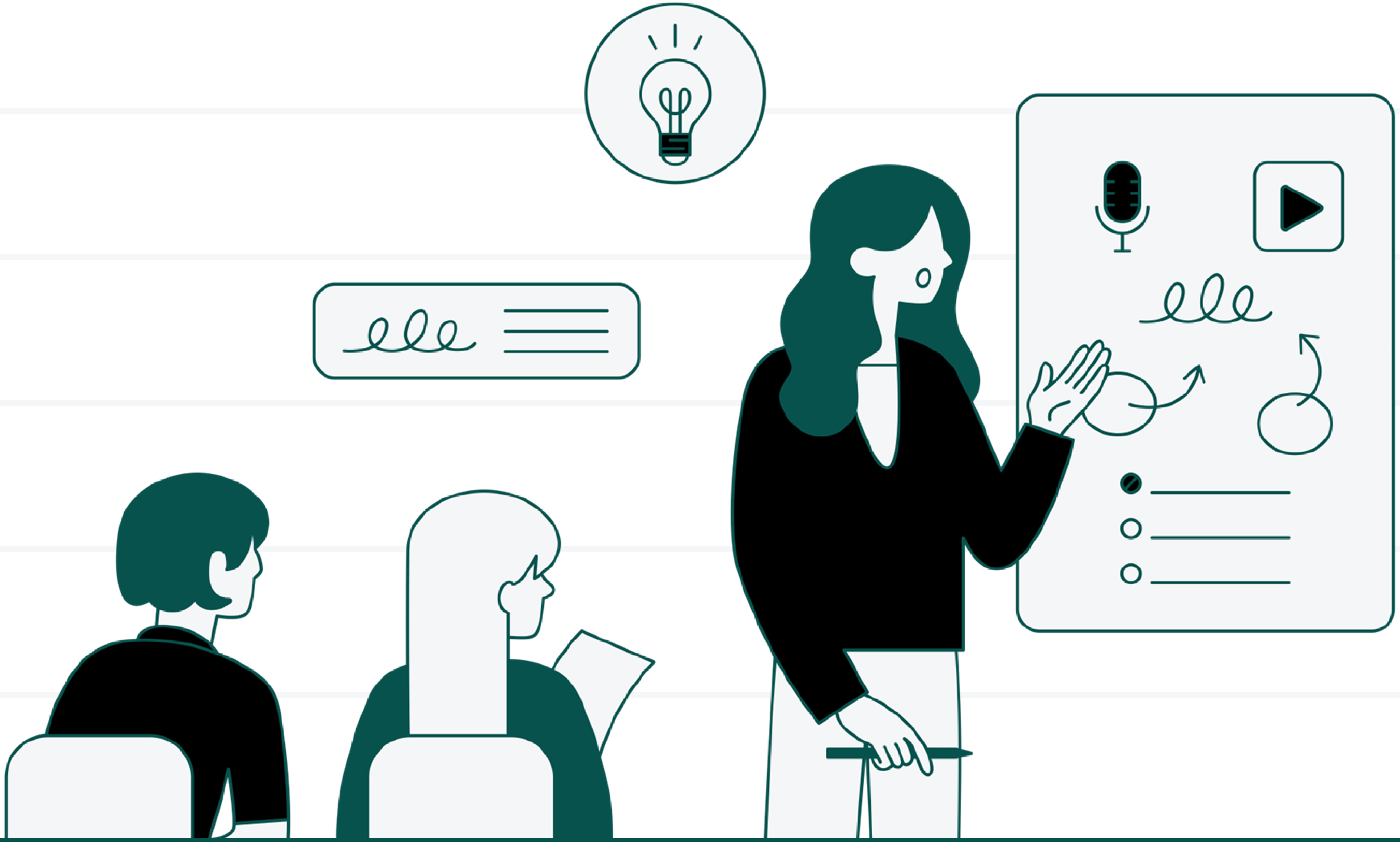
- *easy to understand*
- *easy to manage*
- *fewer production issues*
- *better budget management*

Disadvantages:

- *not flexible*
- *doesn't handle unexpected risks well*
- *not a good for complex or long-term projects*
- *difficult to capture all requirements up front*

When to use the Waterfall Model

- Requirements are very well known
- Product definition is stable
- Technology is understood
- New version of an existing product
- Porting an existing product to a new platform.

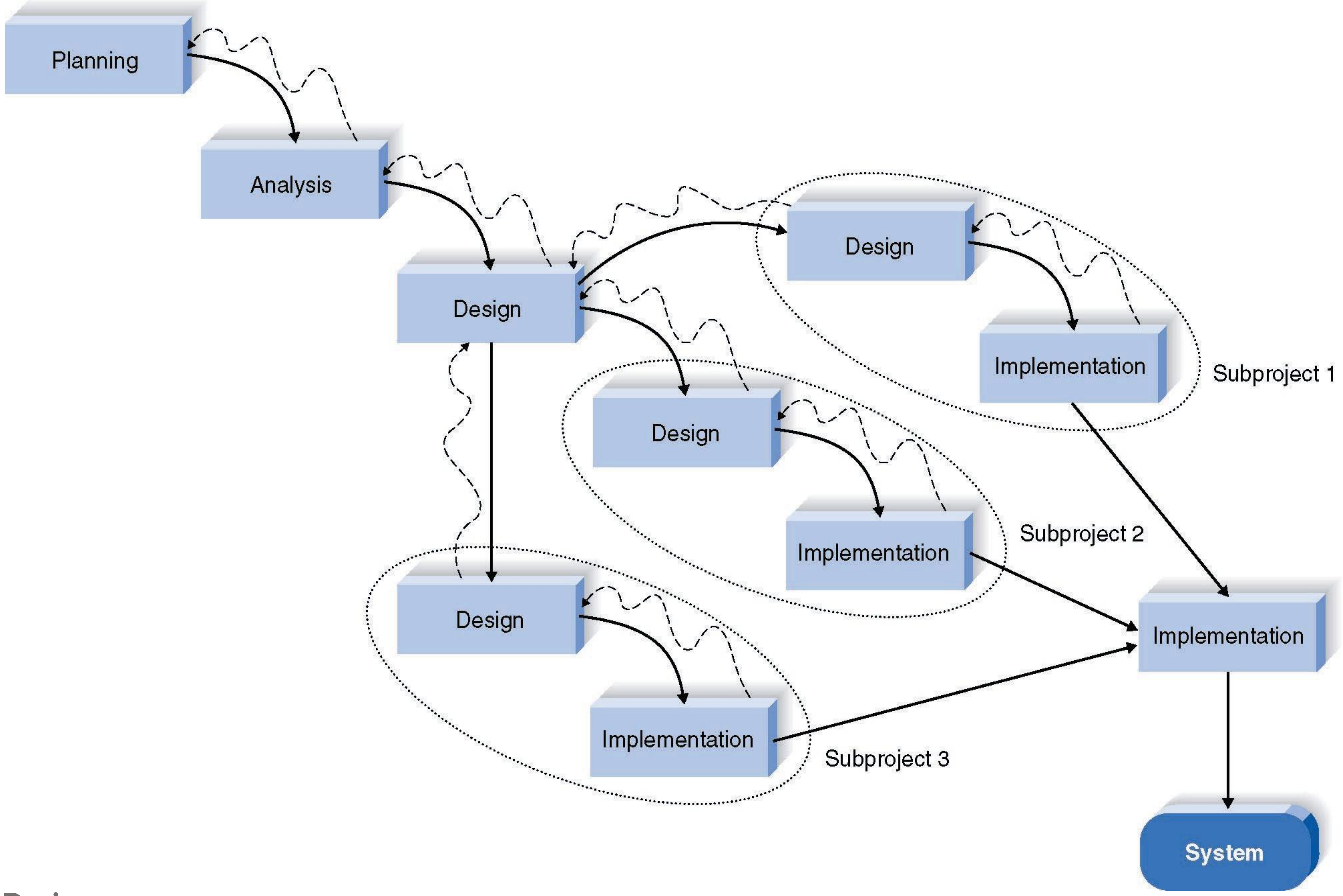


05 Structured-Parallel Methodology

Structured Method

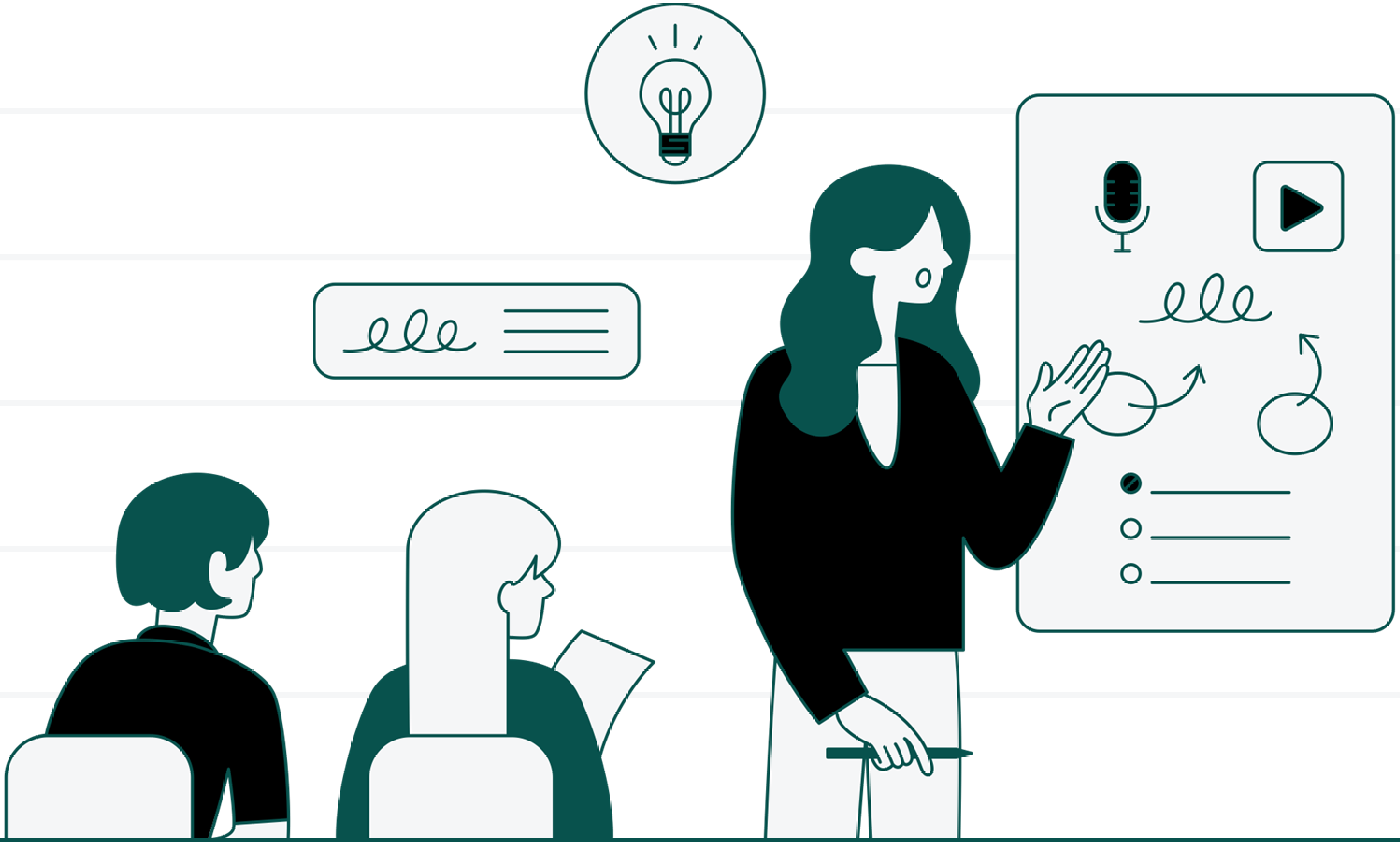
- Assume a project phase is complete before moving to the next phase
- Goal: Doing each phase thoroughly before moving forward ensures correct and high-quality outcomes.
  - Waterfall development
  - Parallel development
  - V-model

Parallel Development Methodology



Pros and Cons of Parallel Development Methodology

Pros	Cons
Reduces Schedule Time	Still Uses Paper Documents
Less Chance of Rework	Sub-projects May Be Difficult to Integrate



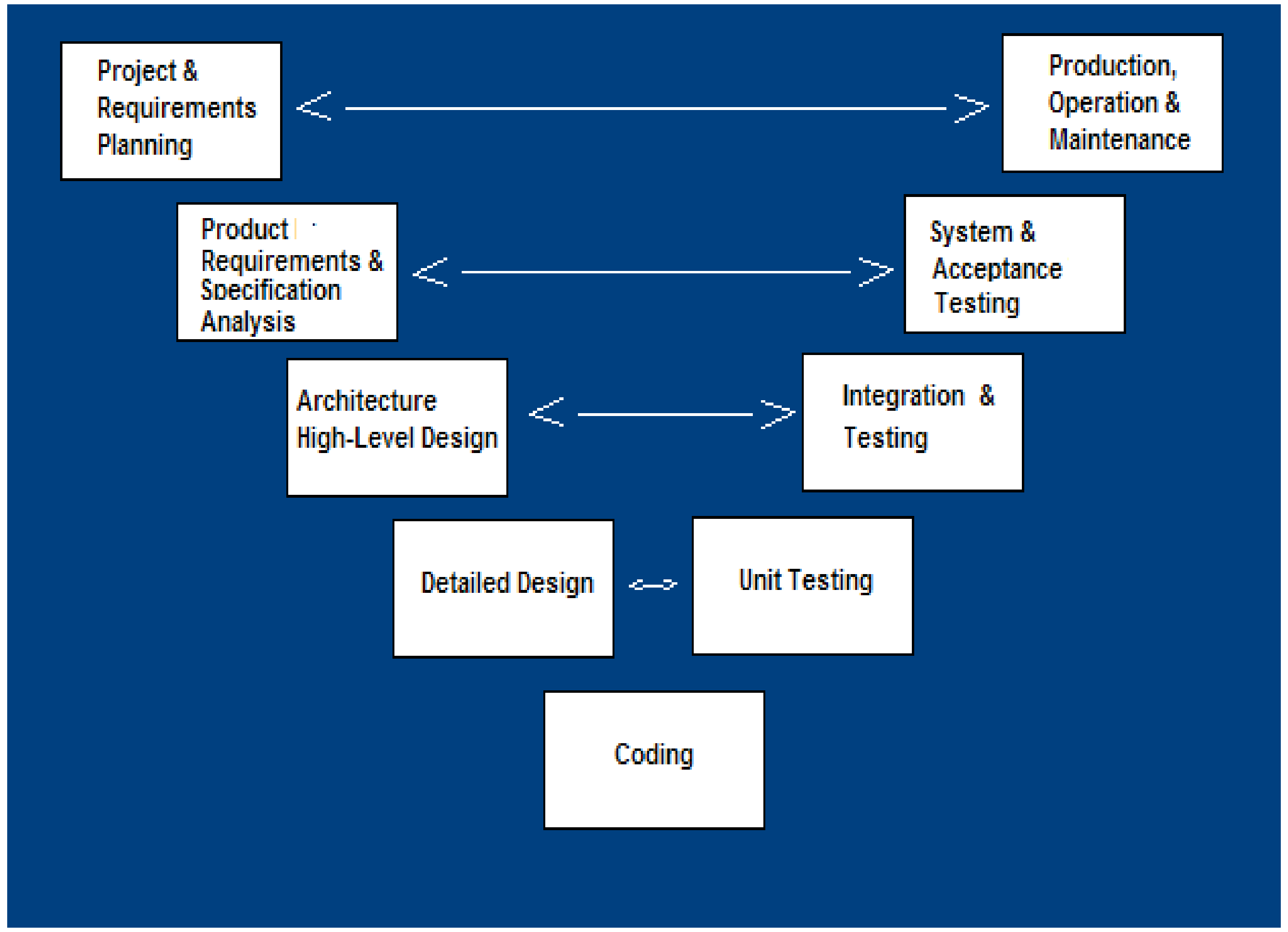
06 Structured-V Model

Structured Method

- Assume a project phase is complete before moving to the next phase
- Goal: Doing each phase thoroughly before moving forward ensures correct and high-quality outcomes.
  - Waterfall development
  - Parallel development
  - V-model

V-Shaped SDLC Model

- A variant of the Waterfall that emphasizes the verification and validation of the product.
- Testing of the product is planned in parallel with a corresponding phase of development


Project & Requirements Planning
Production, Operation & Maintenance
Product Requirements & Specification Analysis
System & Acceptance Testing
Architecture High-Level Design
Integration & Testing
Detailed Design
Unit Testing
Coding

V-Shaped Steps

Project and Requirements Planning – allocate resources

Product Requirements and Specification Analysis – complete specification of the software system

Architecture or High-Level Design – defines how software functions fulfill the design

Detailed Design – develop algorithms for each architectural component

Production, operation and maintenance – provide for enhancement and corrections

System and acceptance testing – check the entire software system in its environment (client environment)

Integration and Testing – check that modules interconnect correctly

Unit testing – check that each module acts as expected

Coding – transform algorithms into software

V-Shaped Strengths

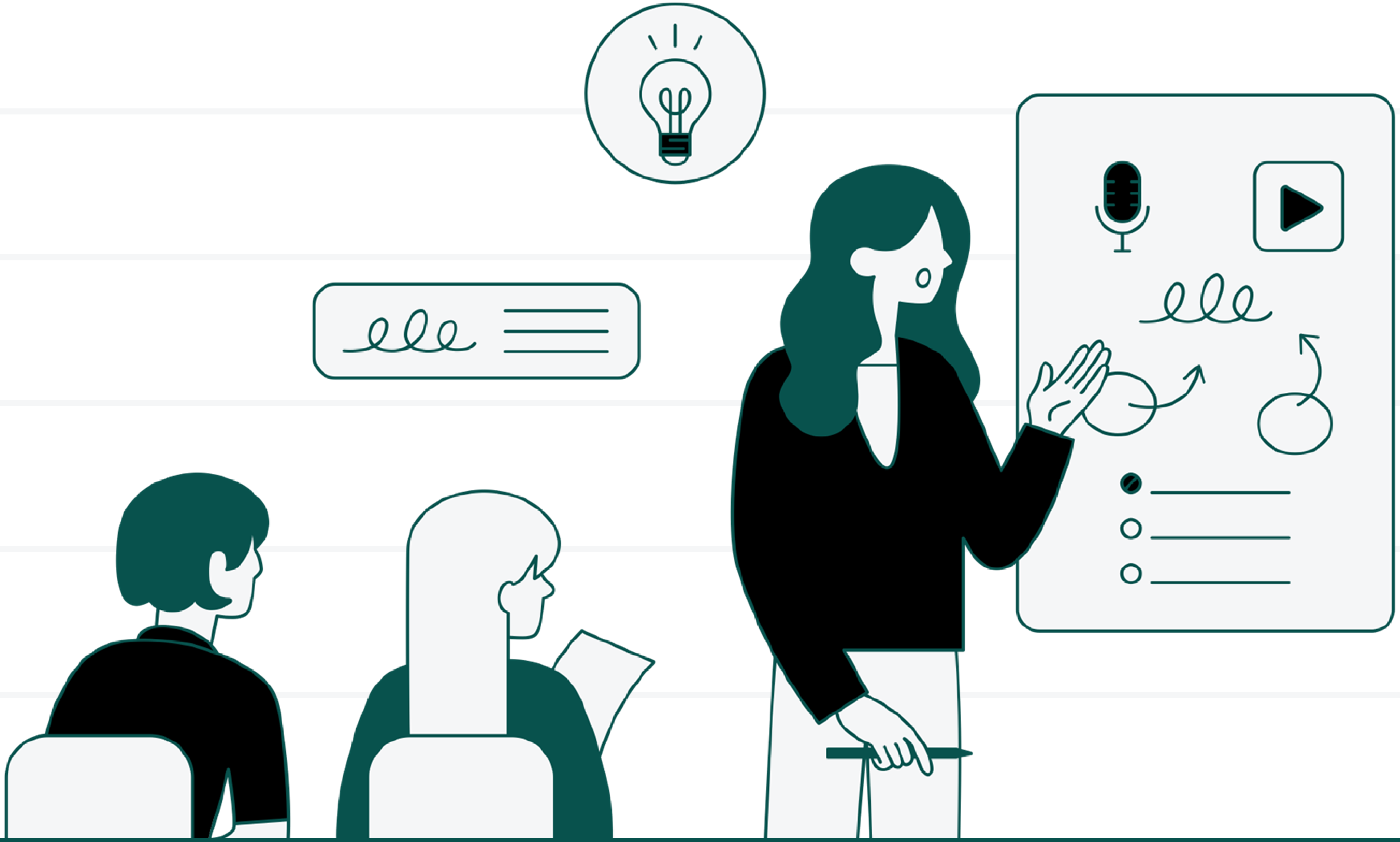
- Stress planning for **verification and validation** of the product in early stages of product development
- Why testing is important?
  - **Each deliverable must be testable**
  - Project management can **track progress by milestones…**
  - **Easy to use**

V-Shaped Weaknesses

- ❑ Does not easily handle **concurrent events/tasks, because tasks are created plan-driven…**
- ❑ Does not handle **iterations** or phases
- ❑ Does not easily handle **dynamic changes in requirements (diffucult to change plans in any time!!!)**
- ❑ Does not contain **risk analysis** activities

When to use the V-Shaped Model

- ❑ Excellent choice for systems requiring **high reliability** – e.g. hospital patient control applications
- ❑ **All requirements are known** clearly
- ❑ **If solution and technology are known well**



07 RAD

Rapid Application Development (RAD)

- RAD, or rapid application development, is an object-oriented approach to systems development that includes a method of development as well as software tools

RAD Phases

- There are three broad phases to RAD:
  - Requirements planning
  - RAD design workshop
  - Implementation

Requirements Planning Phase

- Users and analysts meet to identify objectives of the application or system
- Oriented toward solving business problems

RAD Design Workshop

- Design and refine phase
- Use group decision support systems to help users agree on designs
  - Examples of GDSS tools include online collaboration platforms like Google Workspace, Microsoft Teams, and Slack; brainstorming and idea management tools like Miro and Trello; and polling and voting platforms like SurveyMonkey and Mentimeter.
- Programmers and analysts can build and show visual representations of the designs and workflow to users
- Users respond to actual working prototypes
- Analysts refine designed modules based on user responses

Implementation Phase

- As the systems are built and refined, the new systems or partial systems are tested and introduced to the organization
- When creating new systems, there is no need to run old systems in parallel

Martin Approach to RAD

- The Martin approach to RAD includes four phases:
  - Requirements planning
  - User design
  - Construction
  - Cutover

RAD and the SDLC

- ❑ RAD tools are used to generate screens and exhibit the overall flow of the application
- ❑ Users approve the design and sign off on the visual model
- ❑ Implementation is less stressful since users helped to design the business aspects of the system

When to Use RAD

- RAD is used when
  - The team includes programmers and analysts who are experienced with it
  - There are pressing reasons for speeding up application development
  - The project involves a novel ecommerce application and needs quick results
  - Users are sophisticated and highly engaged with the goals of the company

Using RAD Within the SDLC

- RAD is very powerful when used within the SDLC
- It can be used as a tool to update, improve, or innovate selected portions of the system

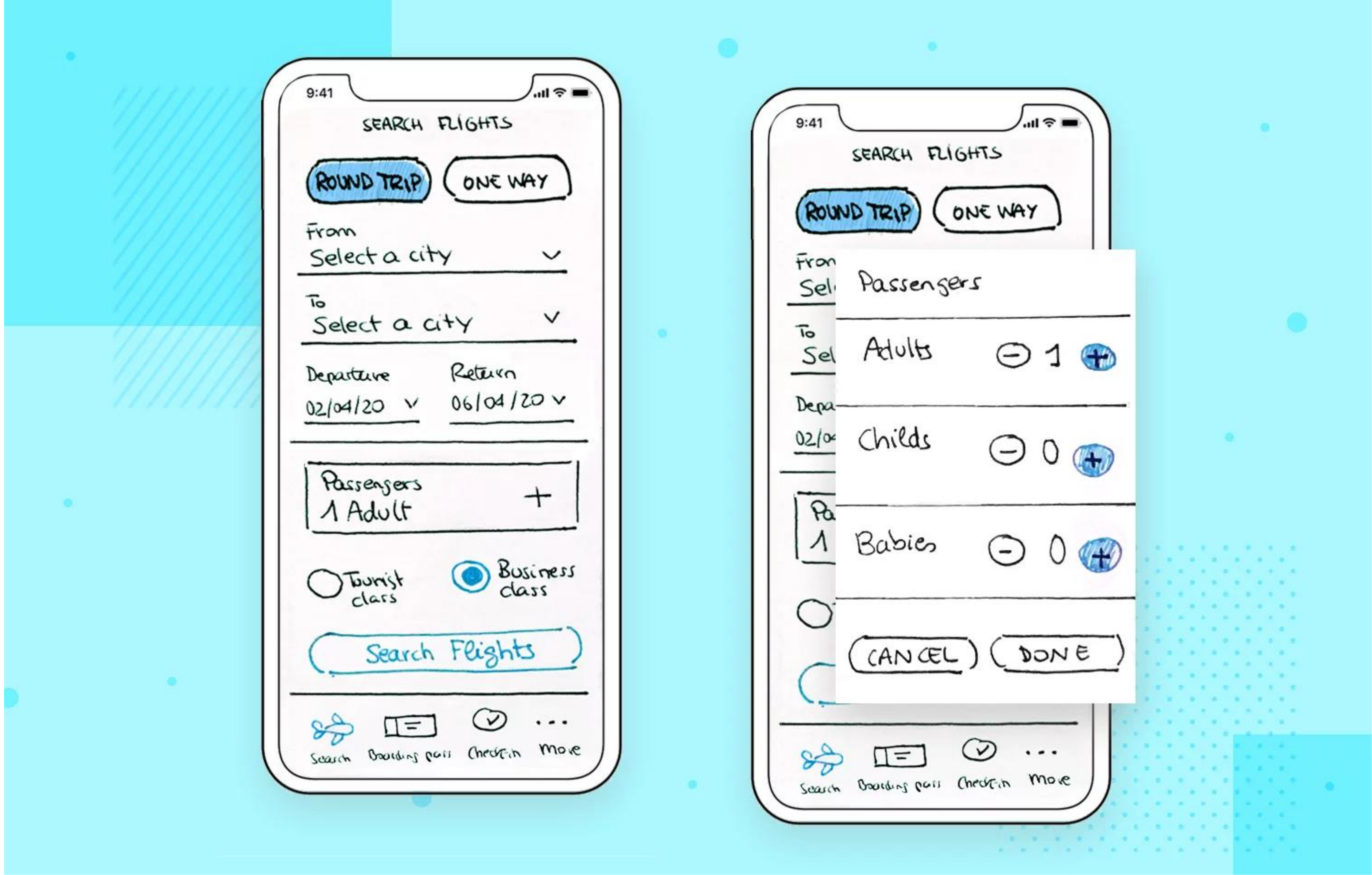
Disadvantages of RAD

- May try and hurry the project too much
- Loosely documented
- May not address pressing business problems
- Potentially steep learning curve for programmers inexperienced with RAD tools

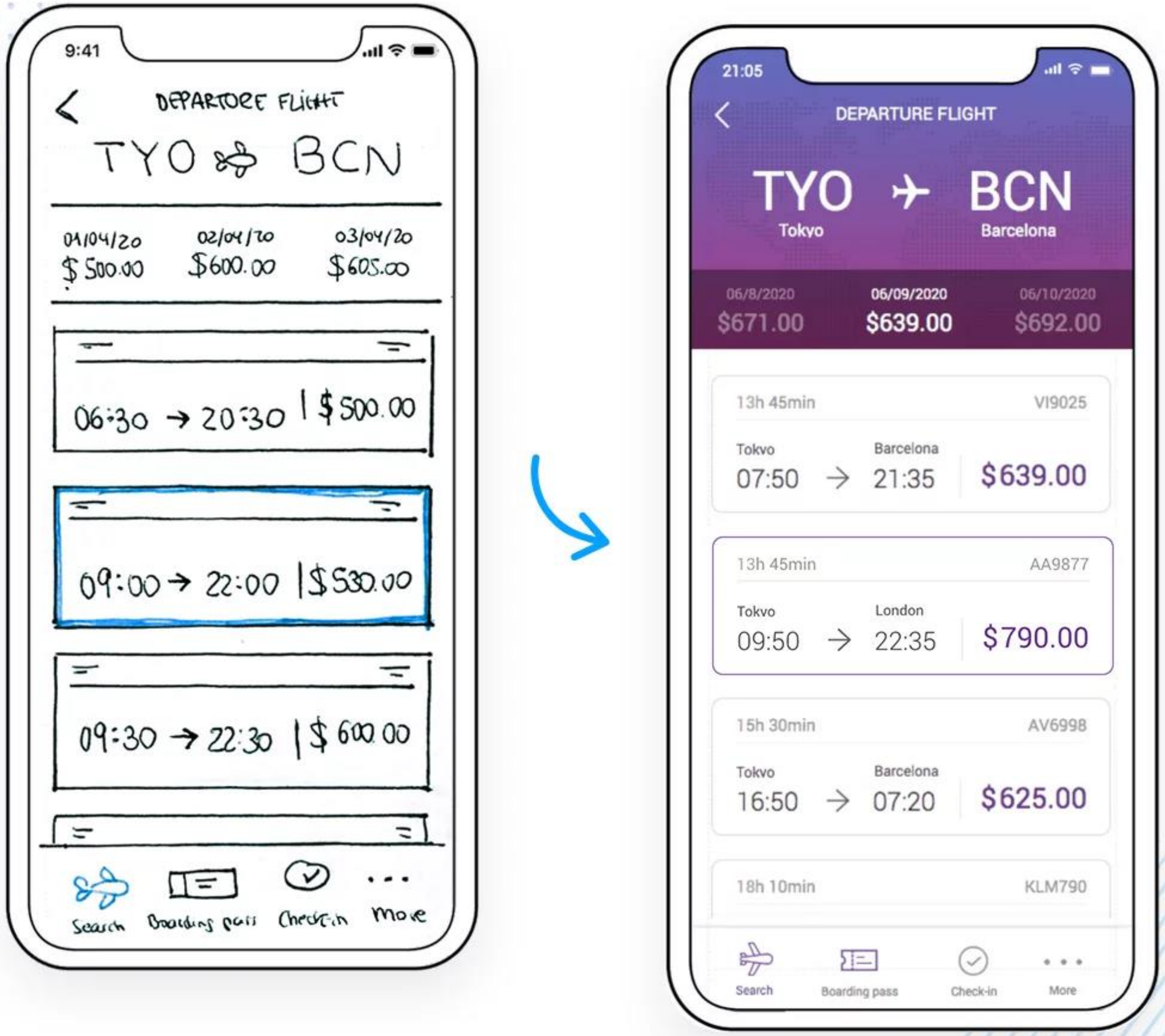
Prototyping

- Prototyping is an information-gathering technique
- Prototypes are useful in seeking user reactions, suggestions, innovations, and revision plans
- Prototyping may be used as an alternative to the systems development life cycle

Prototyping



Prototyping



Initial User Reactions

- Reactions must be gathered from users
- There are three types
  - User suggestions
  - Innovations
  - Revision plans

Four Kinds of Prototypes

- There are four conceptions of prototypes:
  - Patched-up prototype
  - Non-operational scale model
  - First full-scale model
  - Prototype which contain only some of the essential system features

Patched-up Prototype

- This is a working model with all the features but is inefficient
- Users can interact with the system
- Storage and retrieval of data may be inefficient
- Workable but inefficient
- May contain only basic features

Nonoperational Scale Models

- A nonoperational scale mode is one which is not operational, except for certain features to be tested
- Prototype input and output

First Full-Scale Models

- Create a pilot system
- An operation model
- Useful when many installations of the same information system are planned
- An example is a system to be installed in one location, tested and modified as necessary, and later implemented in other locations

Selected Features Prototype

- An operational model that includes some, but not all, of the final system features
- With the acceptance of these features, later essential features are added
- Some menu items are available
- System is built in modules
- These are part of the actual system

Prototyping As an Alternative to the Systems Life Cycle (SLC)

- Two main problems with the SDLC
  - Extended time required to go through the development life cycle
  - User requirements change over time

Prototyping may be used as an alternative

Prototype Development Guidelines

- Guidelines for developing a prototype are
  - Work in manageable modules
  - Build the prototype rapidly
  - Modify the prototype in successive iterations
  - Stress the user interface

Prototype Advantages

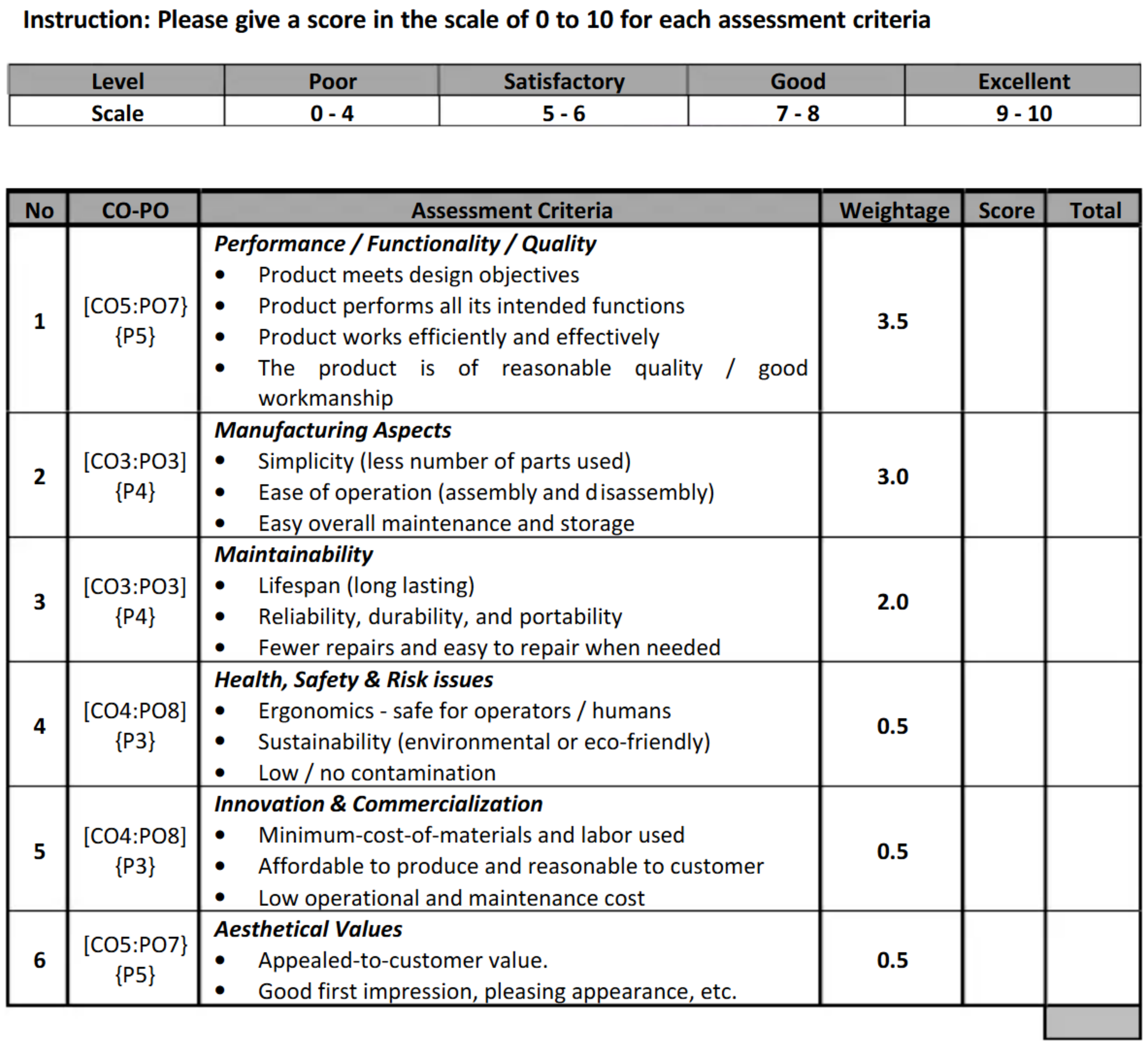
- ❑ Potential for changing the system early in its development
- ❑ Opportunity to stop development on an unworkable system
- ❑ Possibility of developing a system that closely addresses users' needs and expectations

Prototype Disadvantages

- Managing the prototyping process is difficult because of its rapid, iterative nature
- Requires feedback on the prototype
- Incomplete prototypes may be regarded as complete systems

Prototype Evaluation

- Systems analysts must work systematically to elicit and evaluate users' reactions to the prototype
- Three ways the user is involved
  - Experimenting with the prototype
  - Giving open reactions to the prototype
    - Use a prototype evaluation form
  - Suggesting additions to and/or
  - deletions from the prototype



Instruction: Please give a score in the scale of 0 to 10 for each assessment criteria

Level	Poor	Satisfactory	Good	Excellent
Scale	0 - 4	5 - 6	7 - 8	9 - 10

No	CO-PO	Assessment Criteria	Weightage	Score	Total
1	[CO5:PO7] {P5}	***Performance / Functionality / Quality*** • Product meets design objectives • Product performs all its intended functions • Product works efficiently and effectively • The product is of reasonable quality / good workmanship	3.5		
2	[CO3:PO3] {P4}	***Manufacturing Aspects*** • Simplicity (less number of parts used) • Ease of operation (assembly and disassembly) • Easy overall maintenance and storage	3.0		
3	[CO3:PO3] {P4}	***Maintainability*** • Lifespan (long lasting) • Reliability, durability, and portability • Fewer repairs and easy to repair when needed	2.0		
4	[CO4:PO8] {P3}	***Health, Safety & Risk issues*** • Ergonomics - safe for operators / humans • Sustainability (environmental or eco-friendly) • Low / no contamination	0.5		
5	[CO4:PO8] {P3}	***Innovation & Commercialization*** • Minimum-cost-of-materials and labor used • Affordable to produce and reasonable to customer • Low operational and maintenance cost	0.5		
6	[CO5:PO7] {P5}	***Aesthetical Values*** • Appealed-to-customer value. • Good first impression, pleasing appearance, etc.	0.5		

Prototyping on the Web

- Prototyping on the Web can help to facilitate the prototyping process by
  - Allowing users at a distance review the prototype and send comments
  - Allowing users to review the prototype when they have time, and on any machine that has Internet capabilities
  - The analyst does not have to install the software on the user’s computer