

SDLC Methodologies

CE418: Systems Analysis and Design
Maryam Ramezani
Sharif University of Technology
maryam.ramezani@sharif.edu



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](#) ↗

Framework

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.

Methodology

- 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.

Let's Practice

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



SDLC Methodologies

Structured

Rapid Application Development (RAD)

Agile Development

Software Development Methodology (SDM)

- Applying software engineering practices with the specific aim of providing the necessary means for developing **software-intensive systems**
- **Methodology consist of two main parts:**
 - A set of modeling conventions comprising a **(1) Modeling Language** (**syntax** and **semantics**)
 - A **(2) Process**, which
 - **Process:** Provides guidance as to the order of the **activities**,
 - **Product:** Specifies what **artifacts** should be developed using the *Modeling Language*,
 - **Producer:** Directs the tasks of individual developers and the **team** as a whole, and
 - Offers criteria for **monitoring and measuring** a project's products and activities.

Concept	Description
Syntax	"How do we write?" (Structure and order)
Semantics	"What does it mean?" (Meaning and effects)

Software Development Methodology (SDM)

- Methodology can be:
 - **Process Center: Necessary**
 - **Product Center**
 - **People (Producer/Role) Center.**
- Agile methodology is process center and people center

Structured Method

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

Structured Method Waterfall

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



Software Requirements Specification (SRS)

Requirements



Design

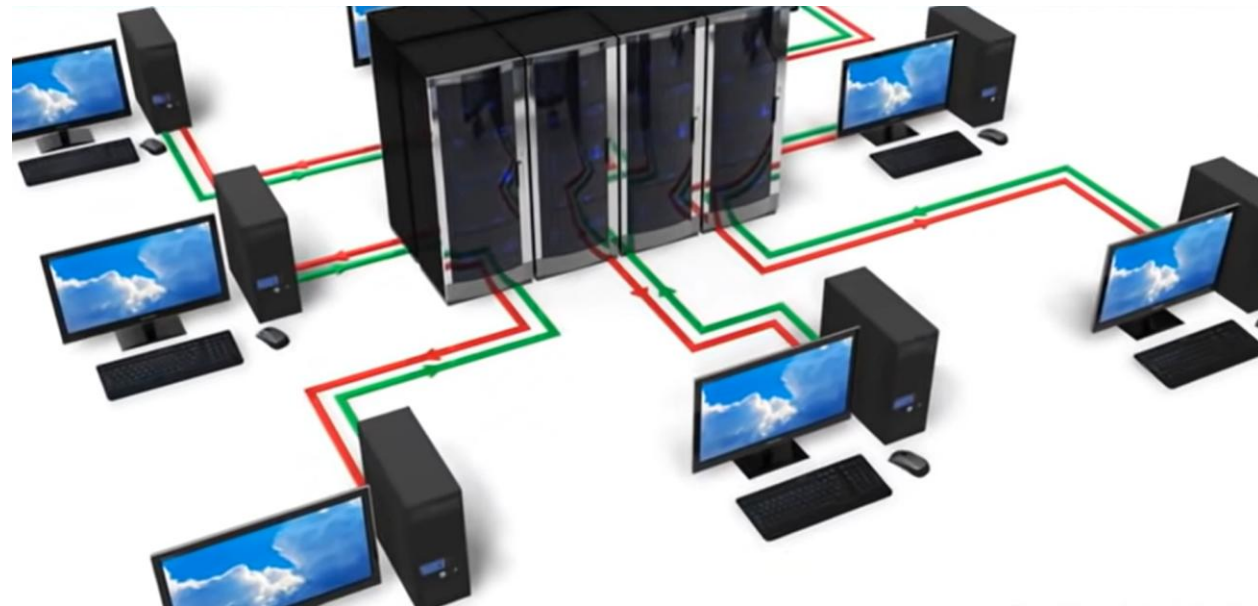
- Includes the **logical design** and the **physical design**
- **Logical Design:**
 - The abstract representation of how the software data flows the inputs and the outputs. It is often shown graphically as a diagram showing data flow.

- 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



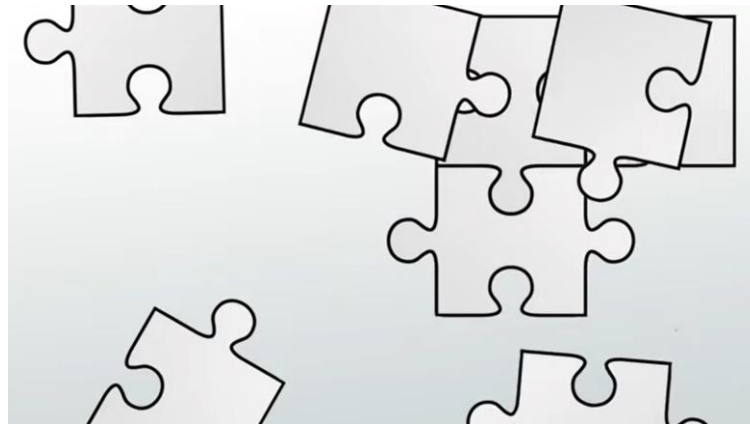
Design Document

Let the implementation begin!!



Development

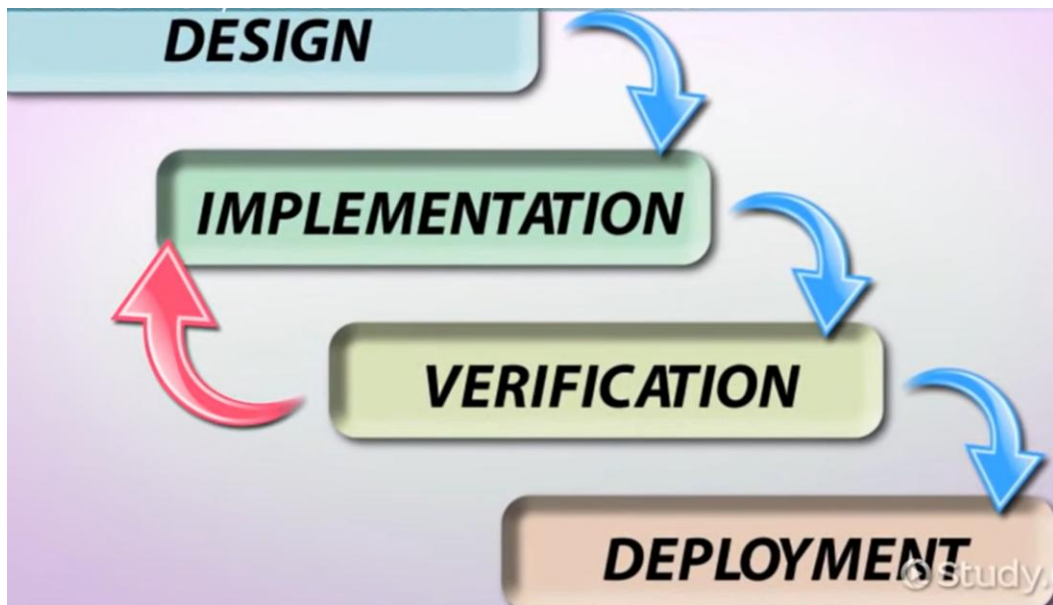
- Building the design into actual software
- Often, software is build in units and integrated into a whole at a later point.
- As one software unit implementation is complete, it is send to verification (Testing) and another software unit implementation begins.



Test (Verification)

- Testing the software against the requirements.

Fail



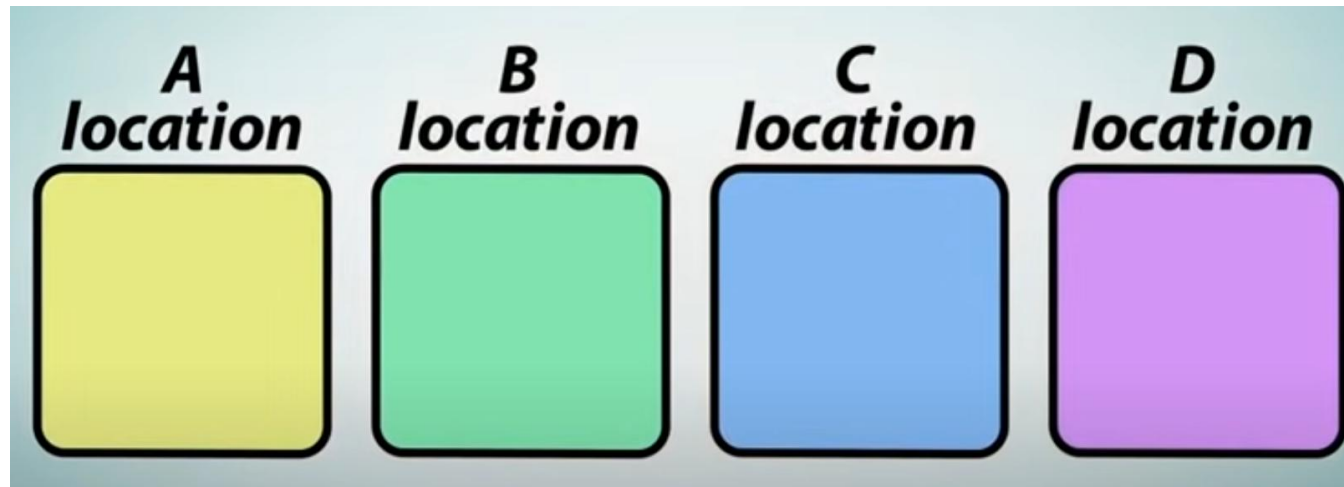
Pass

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

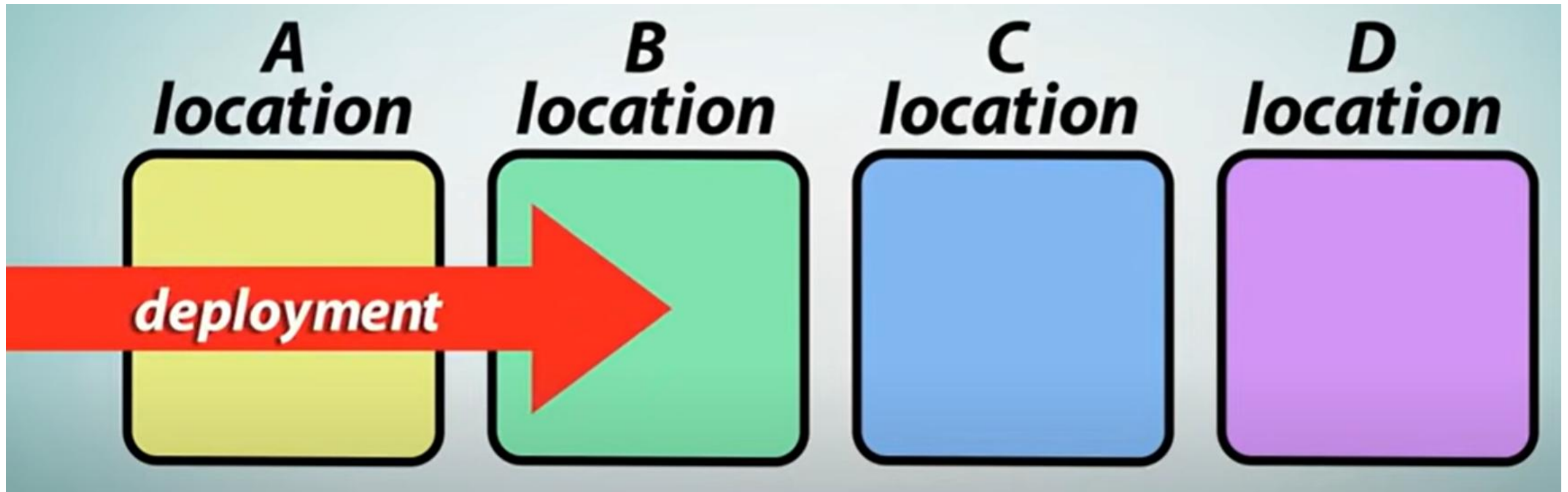


Deployment

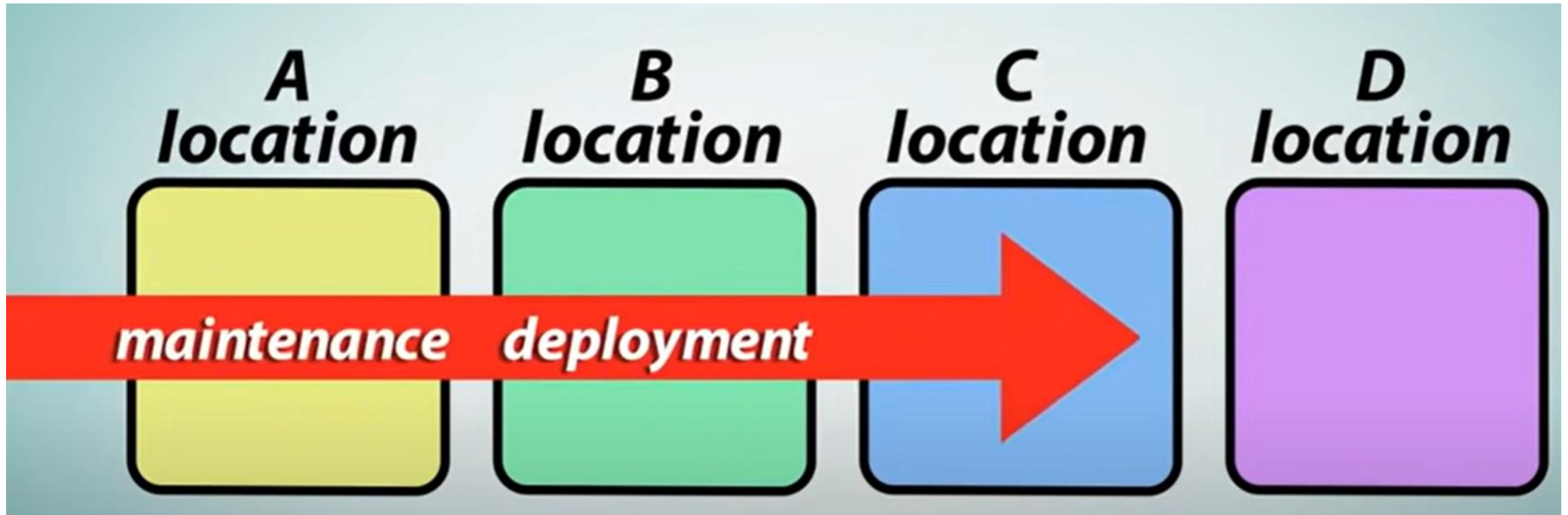
- The actual release of the new software into the information technology environment.
- This phase may be broken into units to allow for ongoing deployment and maintenance.
- Example:



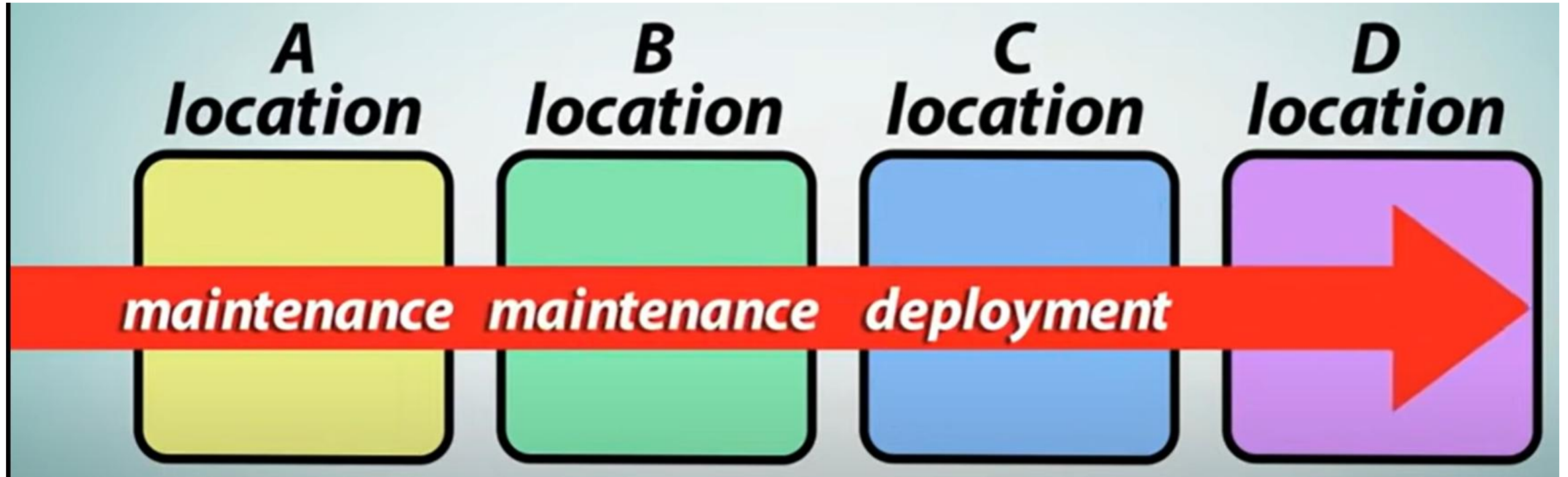
Deployment



Deployment



Deployment



Maintenance

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



Conclusion

***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.

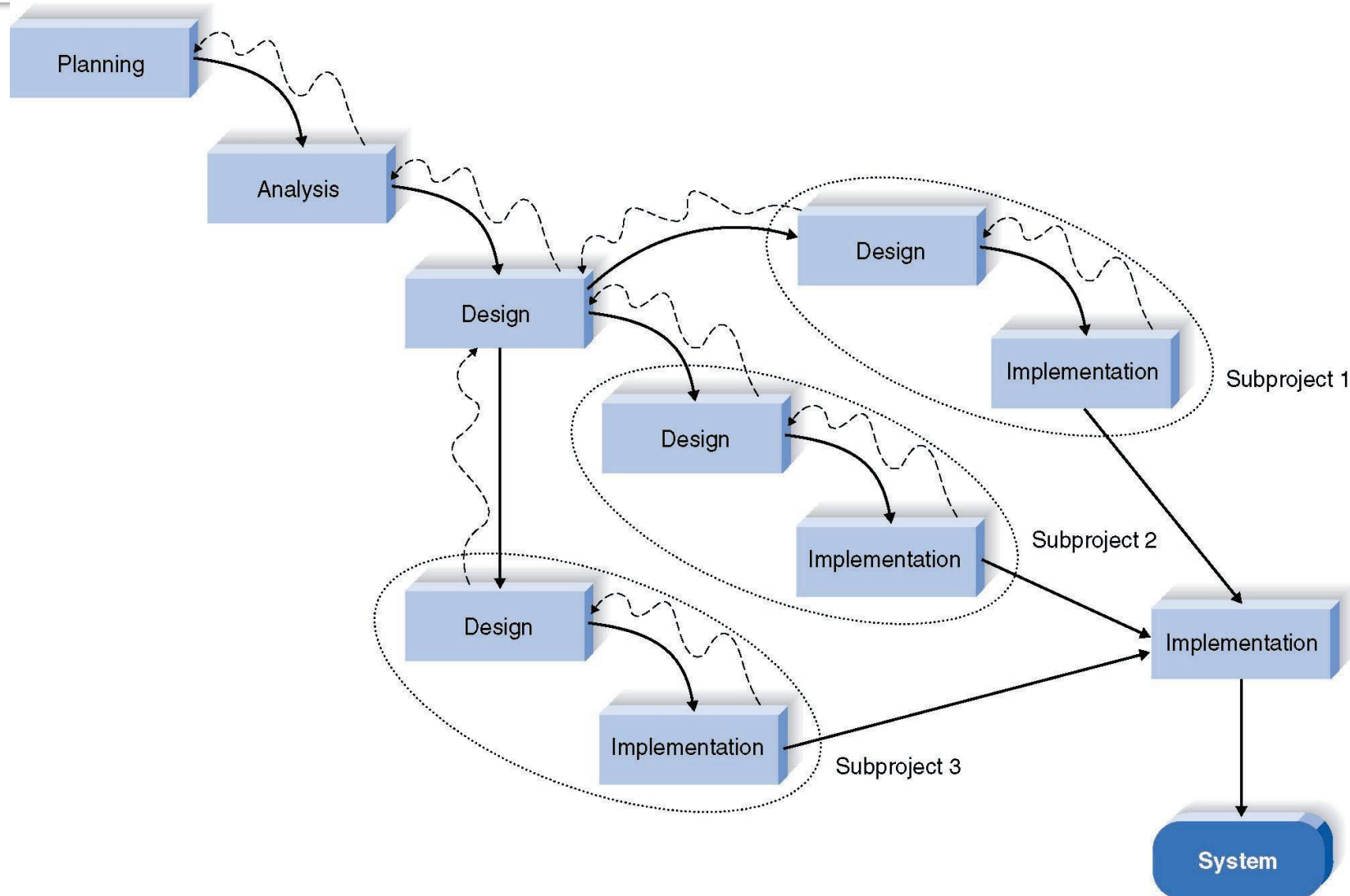
Structured Method

Paraller

Parallel Methodology

- This methodology involves developing different parts of the software simultaneously instead of sequentially. It reduces the overall time of development by working on multiple components at once.

Parallel 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

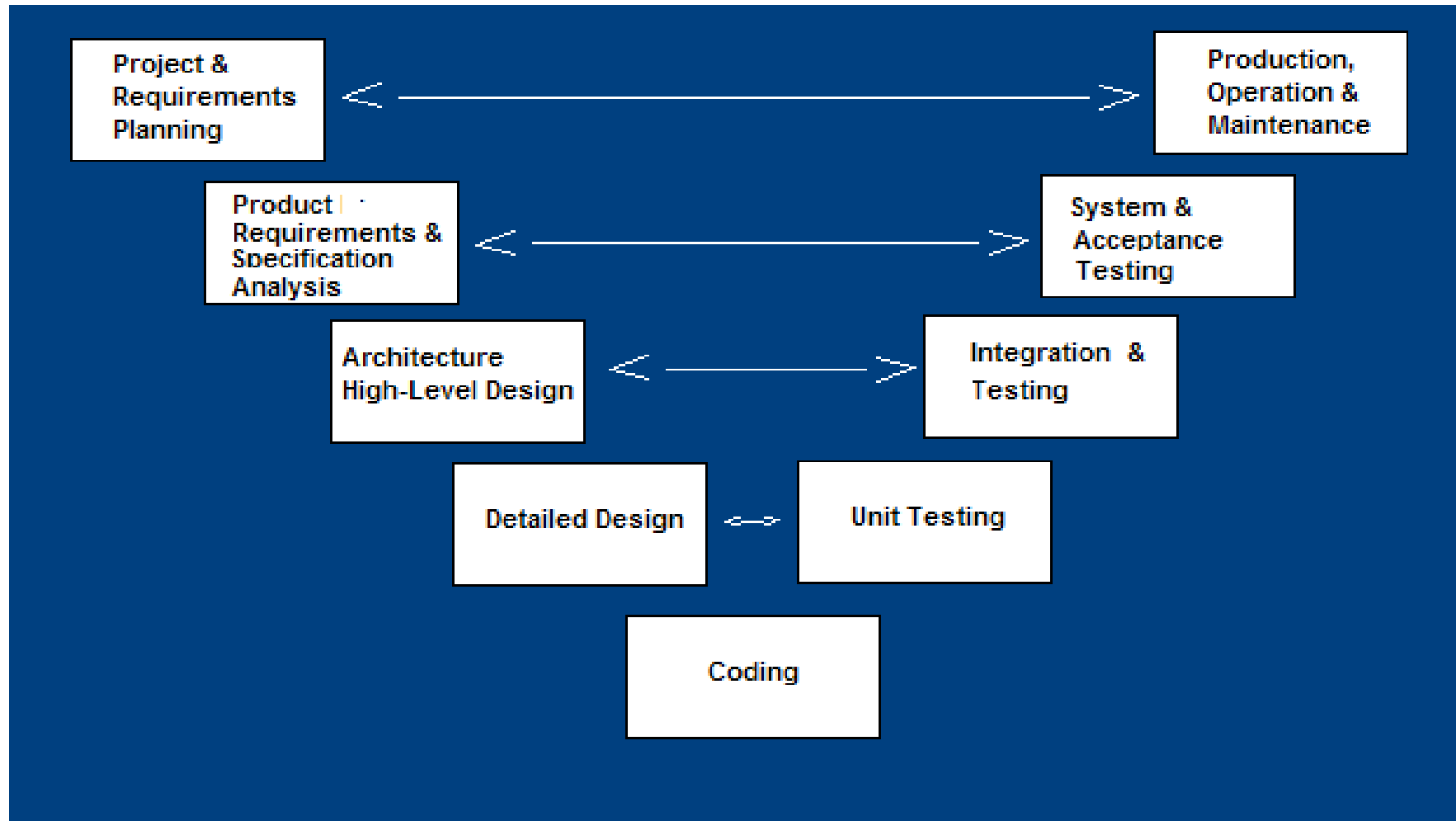
Structured Method

V Model

V Model

- The V-Model (Verification and Validation Model) is an extension of the traditional waterfall model.
- Development and testing phases are executed in parallel, with corresponding stages for each development phase to ensure quality at every step.

V-Shaped SDLC Model



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 (difficult 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**

Rapid Application Development (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

1) Requirements Planning

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

2) 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

3) 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
- 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

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

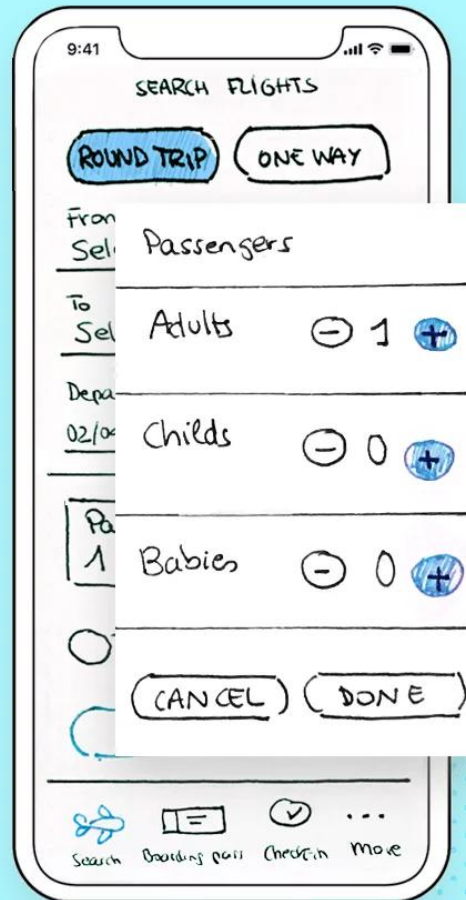
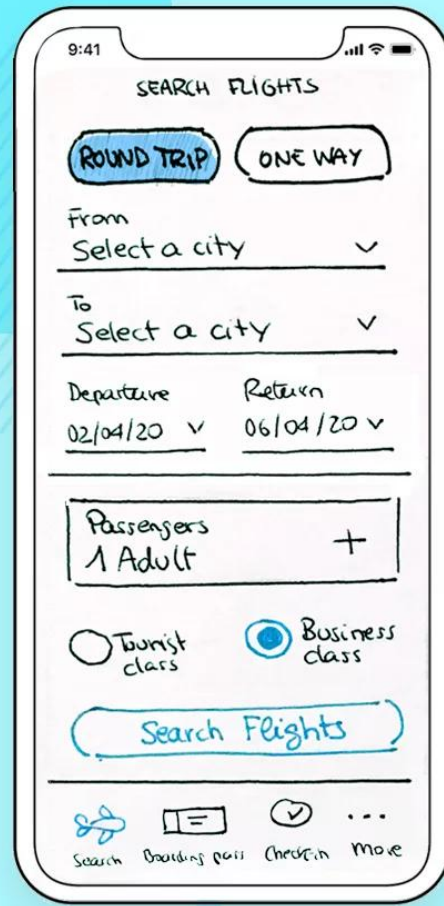
Disadvantages of RAD

- 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

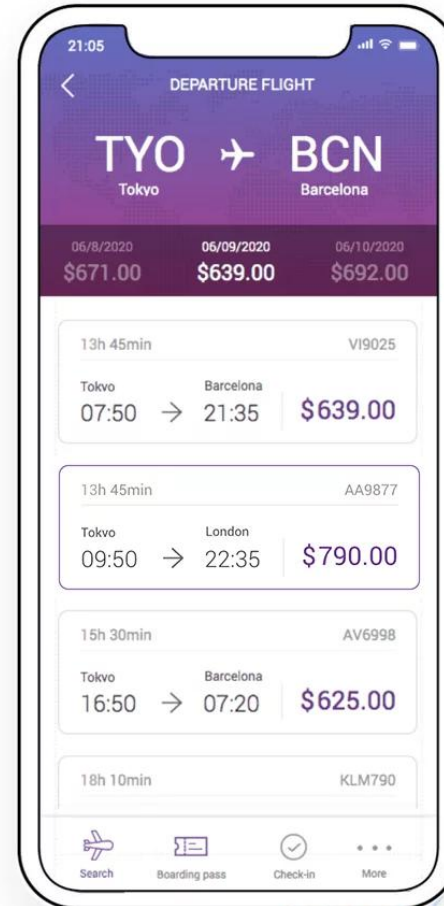
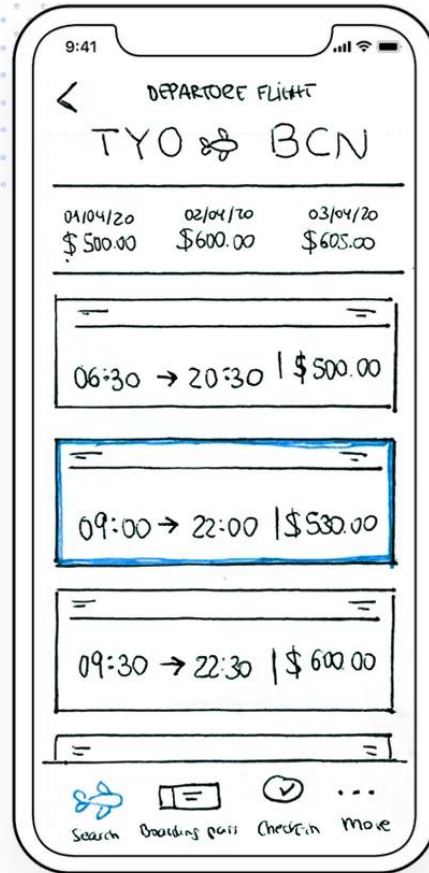
Prototyping

- 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

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 <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • Simplicity (less number of parts used) • Ease of operation (assembly and disassembly) • Easy overall maintenance and storage 	3.0		
3	[CO3:PO3] {P4}	Maintainability <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • Ergonomics - safe for operators / humans • Sustainability (environmental or eco-friendly) • Low / no contamination 	0.5		
5	[CO4:PO8] {P3}	Innovation & Commercialization <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • Appealed-to-customer value. • Good first impression, pleasing appearance, etc. 	0.5		

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

Agile Development

Next Slide!

Any Question?