



DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF MANAGEMENT STUDIES AND COMMERCE
UNIVERSITY OF SRI JAYEWARDENEPURA

ITC 1370

Information Technology for Business

Year I – Semester I

Chapter 9

Information Systems Development



Learning Objectives

Upon successful completion of this chapter, you will be able to:

- Explain the overall process of developing a new software application
- Explain the differences between software development methodologies
- Identify different types of implementation methodologies

What is an Information System?

“An information system (IS) can be defined technically as a set of interrelated components that collect, process, store, and distribute information to support decision making and control in an organization”

(Laudon and Laudon, 2016)

Components of an Information System

Hardware

Software

Data

People

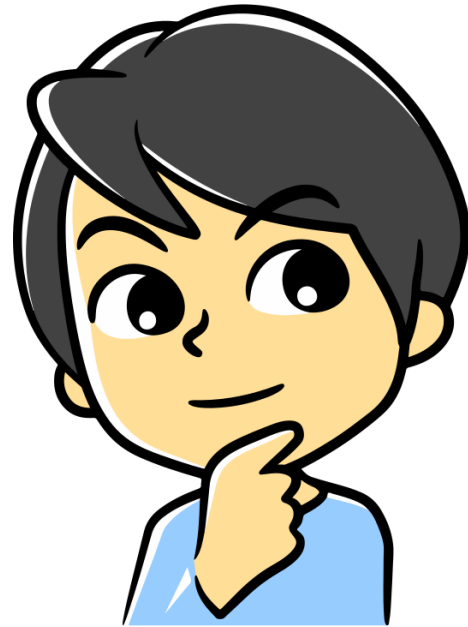
Processes





Purchase

Build





Acquisition of new software

Bespoke Development

Off-the-shelf software

End-user Development



Software Development Methodologies

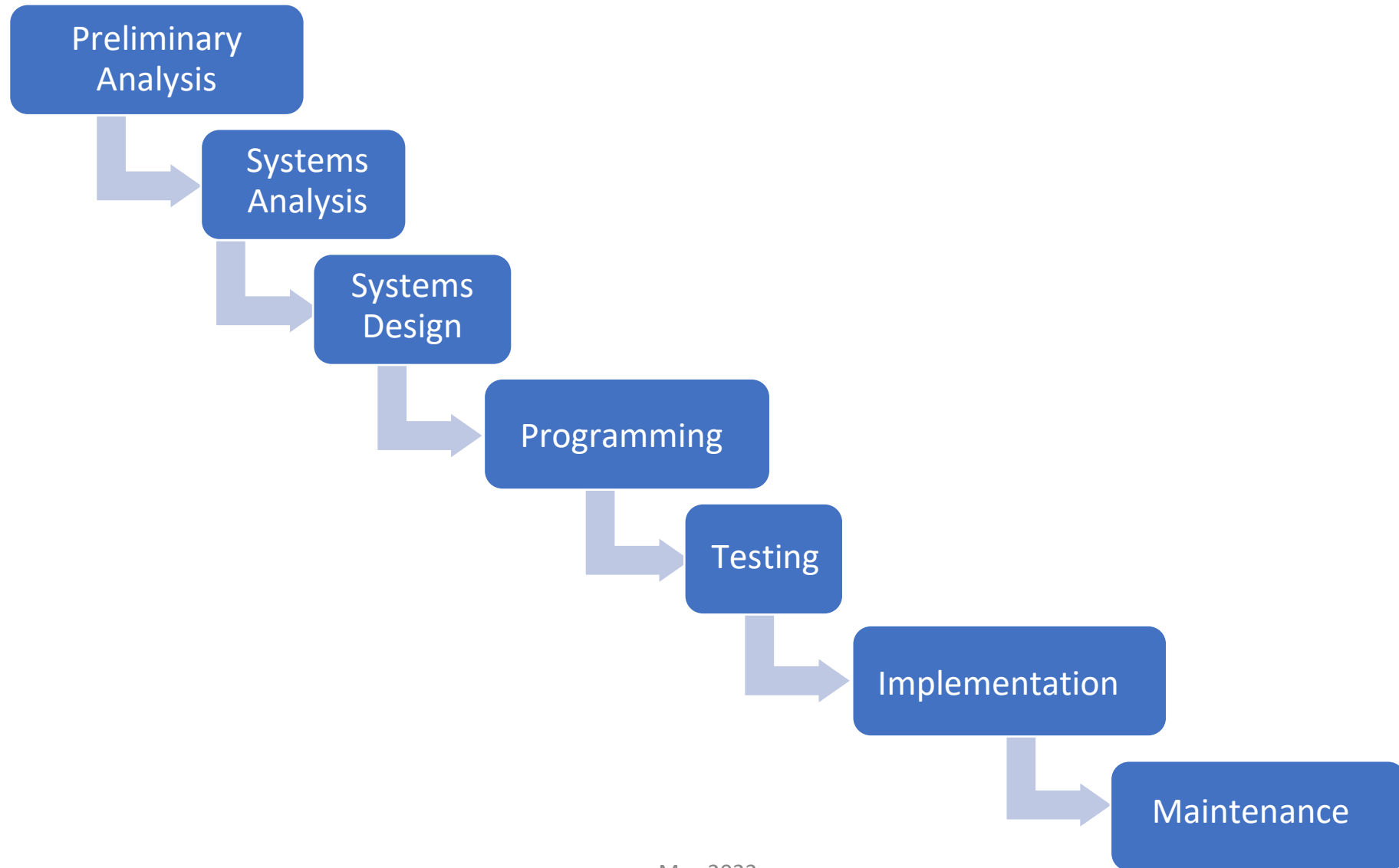
- Systems Development Life Cycle (SDLC) – Waterfall Methodology
- Rapid Application Development (RAD)
- Agile Methodologies
- Lean Methodology, etc ...



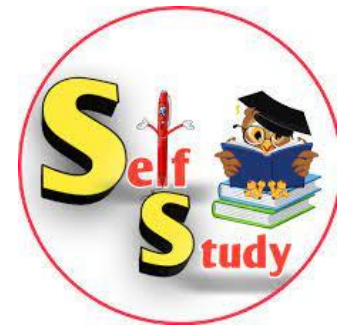
Systems Development Life Cycle (SDLC)

- Developed in 1960s to manage large software development projects
- Still in wide use today and called “**waterfall methodology**”
- Very structured
 - each phase has an end deliverable requiring approval to next phase
 - each phase must be completed before the next phase can start

Systems Development Life Cycle (SDLC)



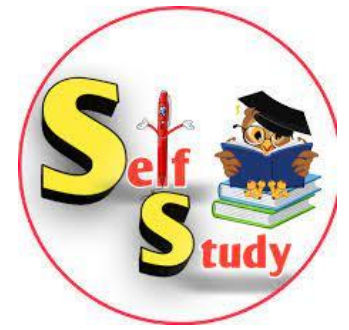
Phases of SDLC



01. Preliminary Analysis

- Reviewing the replacement request for a or new system
 - The review includes questions such as:
 - What is the problem-to-be-solved?
 - Is creating a solution possible?
 - What alternatives exist?
 - What is currently being done about it?
 - Is this project a good fit for our organization?
- Launching a feasibility study
 - Technical Feasibility
 - Economic Feasibility
 - Legal Feasibility
- The result is a feasibility analysis document
- The task is done by Requirements Analyst or Business Analyst

Phases of SDLC

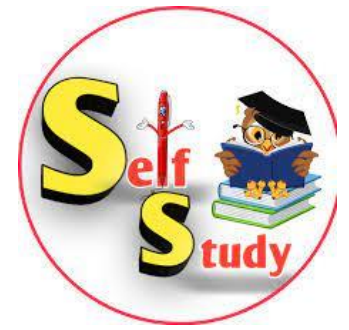


02. Systems Analysis

- Determine the specific requirements for the new system by system analysts
- No programming is done in this step.
 - Instead,
 - Document the procedures
 - Interview the key players/users
 - Develop data requirements
- The result is a System Requirements Specification (SRS)
- The task is done by Systems Analyst



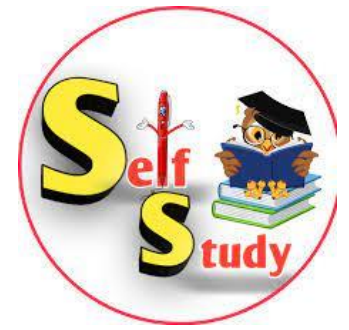
Phases of SDLC



03. Systems Design

- The designer translates the business requirements into specific technical requirements
 - The design for the user interface
 - Database
 - Data inputs and outputs
 - Reports
- The result is a system design document
- This document will have everything a programmer needs to create the system in reality
- The task is done by Systems Analyst, Developer, or Systems Architect

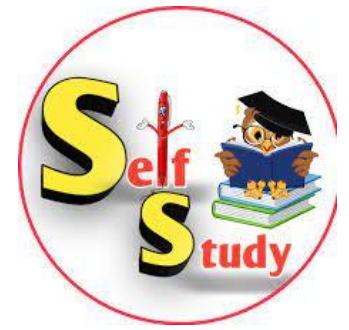
Phases of SDLC



04. Programming

- The codes are written using the system design document as a guide
- The result is an initial working program that meets the requirements specified in the system analysis phase and the design developed in the system design phase
- The tasks are done by Developer, Software Engineer, Programmer, or Coder.

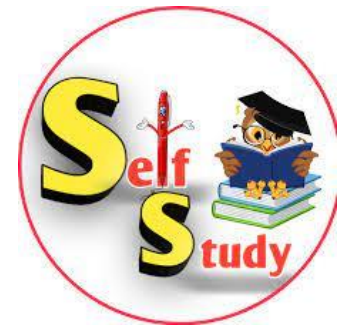
Phases of SDLC



05. Testing

- The software program developed in the programming phase is put through a series of structured tests
 - Unit test - evaluates individual parts of the code for errors or bugs.
 - System test - different components of the system are tested to ensure that they work together properly
 - User acceptance test - allows the users of the software to test the system to ensure that it meets their standards
- Any bugs, errors, or problems found during testing are resolved and then the software is tested again.
- The tasks are done by Tester, Testing Analyst, or Quality Assurance Engineer

Phases of SDLC

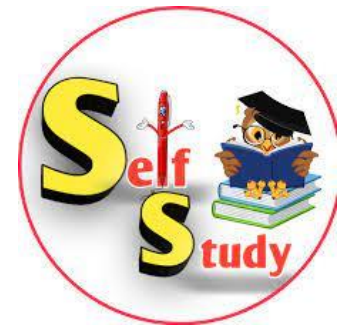


06. Implementation

- Once the new system is developed and tested, it has to be implemented in the organization (converting from old system to new system)
- In order to implement the new system within the organization following activities need to be undertaken
 - Provide user training
 - Provide documentation
 - Convert data from the previous system to the new system
- Implementation can take many forms depending on the
 - type of system
 - the number and type of users
 - how urgent it is that the system become operational



Phases of SDLC

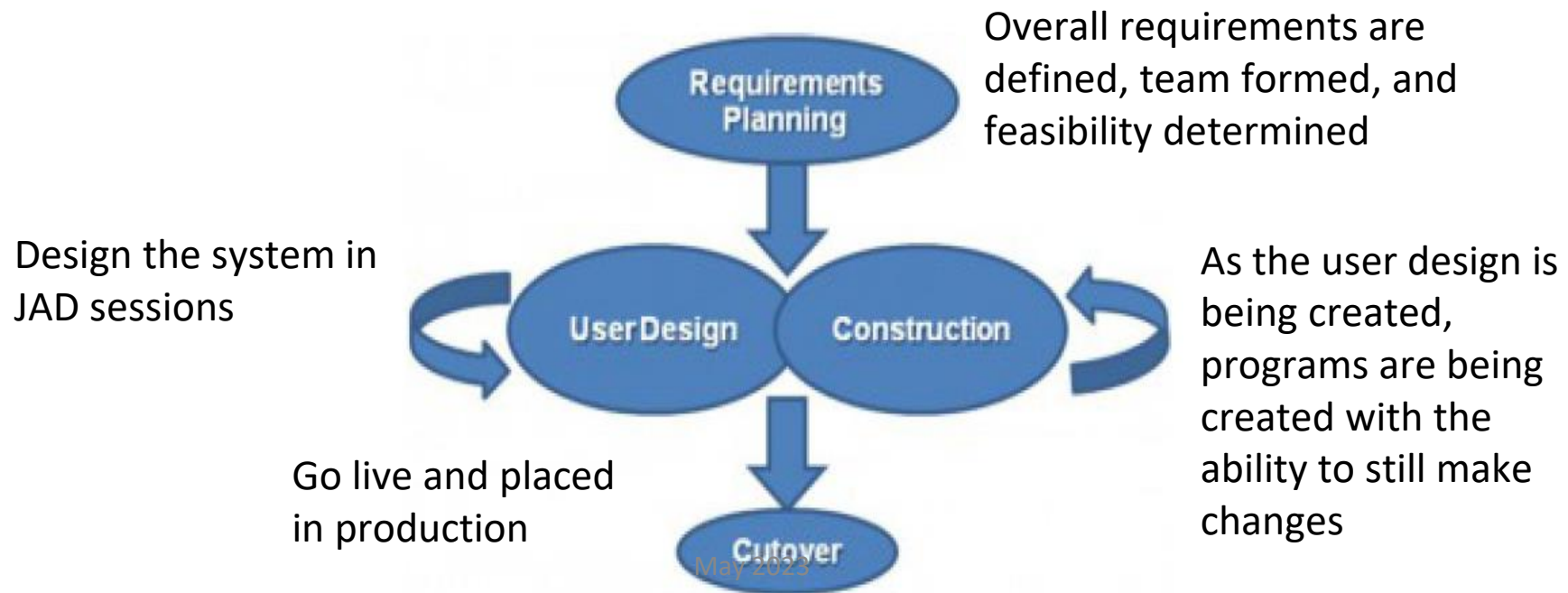


07. Maintenance

- Fix the reported bugs and implement the requests for new features after evaluating them
- Create system updates and backups of the software for each new version of the program
- This is a continuous process throughout the project life cycle

Rapid Application Development (RAD)

- Focuses on quickly building a working model and getting feedback from users to update the working model for smaller projects and giving users the ability to provide feedback during the process

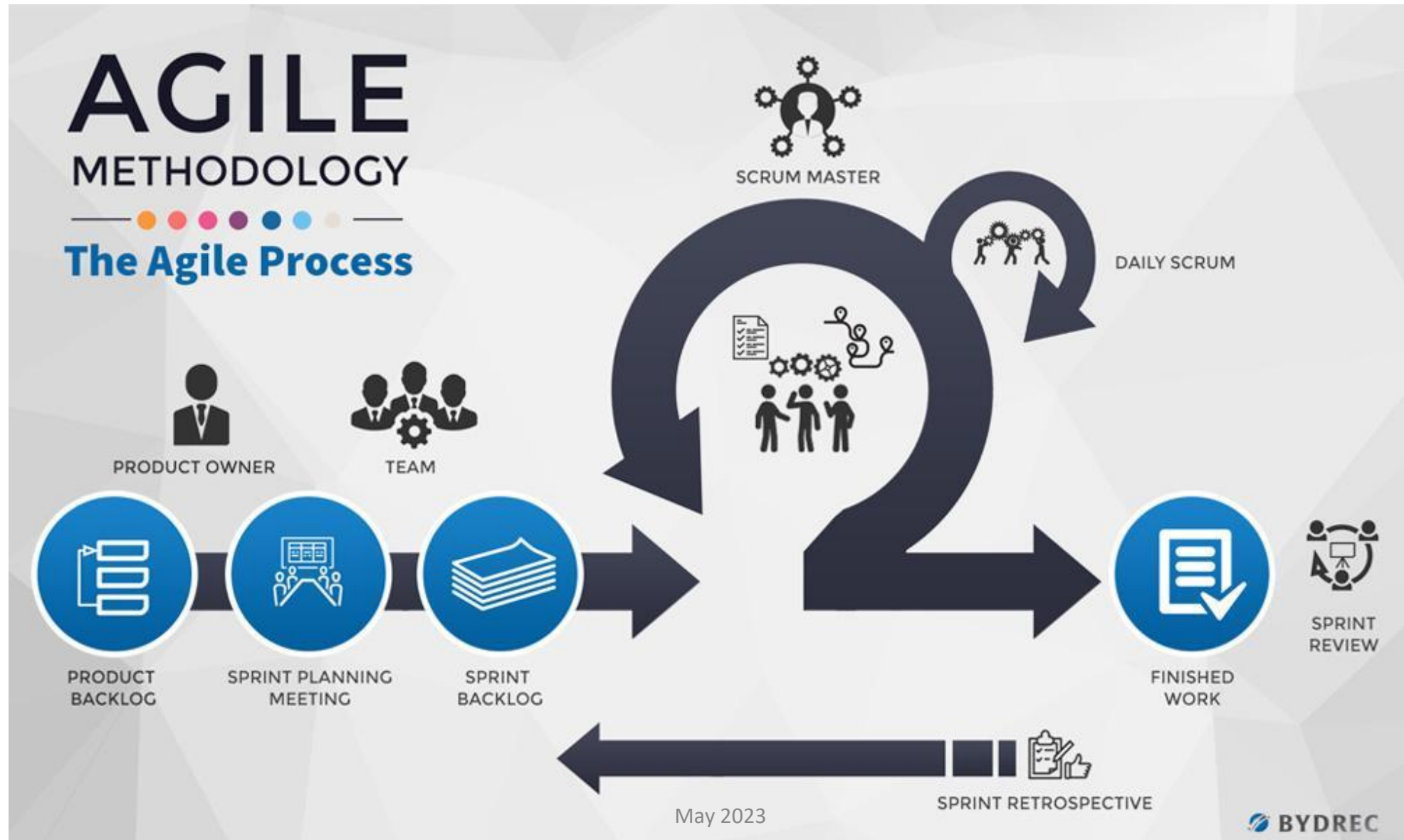




Agile Methodologies

- Group of methodologies for incremental changes focused on quality and attention to detail
- Each increment is released with very specific objectives
- The agile methodologies are based on the “Agile Manifesto”
- Characteristics:
 - Small cross-functional teams are used
 - Daily status meetings are held
 - Short timeframe increments for each change
 - A working project is completed at end of each iteration and demonstrated to stakeholders
- The goal is to provide the flexibility of an iterative approach while ensuring a quality product
- Video Link : <https://www.youtube.com/watch?v=1iccpf2eN1Q&t=9s>

Agile Methodologies

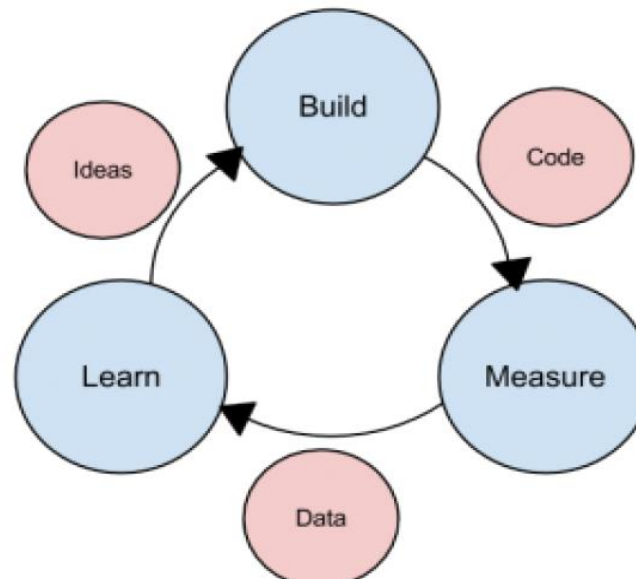


Lean Methodology

- Takes an initial idea and develops a Minimum Viable Product (MVP)
- The MVP is a working software application with just enough functionality to demonstrate the idea behind the project
- Works best in entrepreneurial environment

While moving through the phases, feedback is key and is generated in 2 forms:

1. Direct observation and discussion with users
2. Usage statistics gathered from the software itself

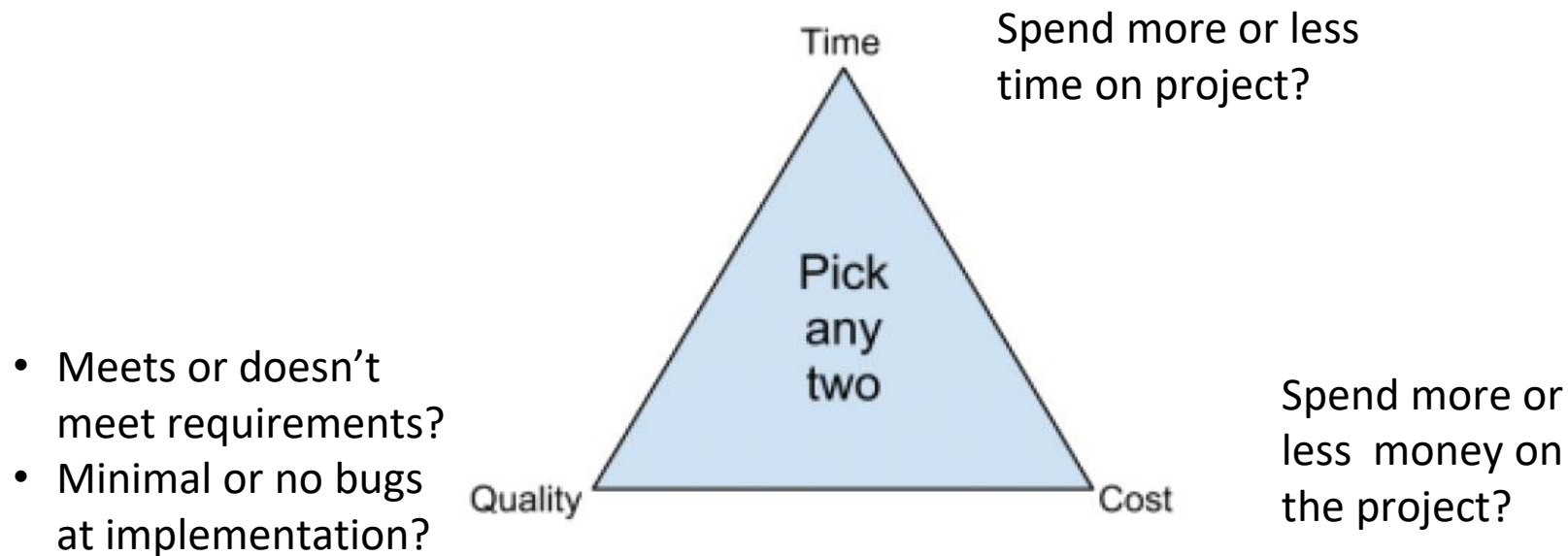


Usually requires several iterations as the team uses the feedback to determine whether to continue in same direction or pivot in a new direction or MVP



Quality Triangle

- Decisions are made during development that affect 3 factors – time, cost, and quality
- Only 2 can be addressed requiring compromise/tradeoffs





End User Computing

- Non-IT individuals develop their own solutions that are not trained in programming or software development
- Advantages:
 - Development is closer to those that will use them
 - Quick development of software
- Disadvantages:
 - Several applications may perform the same functions which may or may not have same results
 - May not be fully tested and bug-free
 - Data is not always backed up



Implementation Methodologies

Several implementation methodologies exist:

- Direct cutover – new system is turned on and old system turned off
 - Riskiest but least expensive – no need to support 2 systems
- Pilot implementation – a small group uses the new system
 - Small impact on organization is something goes wrong – still have old system running
- Parallel operation – all transactions are entered in new and old system
 - Very expensive to maintain 2 systems
 - Least risky that you can identify bugs and go back to old system if needed
- Phased implementation – new functions are implemented as parts of old system are turned off
 - Slowly move from old system to new one



Implementation Methodologies Support

Every implementation requires support in 2 key areas:

1. Change management

- All proposed changes should be communicated to all affected personnel, including IT

2. Maintenance

- Often newly implemented systems still need changes for fixing bugs
- Management needs to ensure that the system continues to run and is aligned with business priorities

Summary

- Explained the overall process of developing a new software application
- Explained the differences between software development methodologies
- Identified different implementation methodologies