

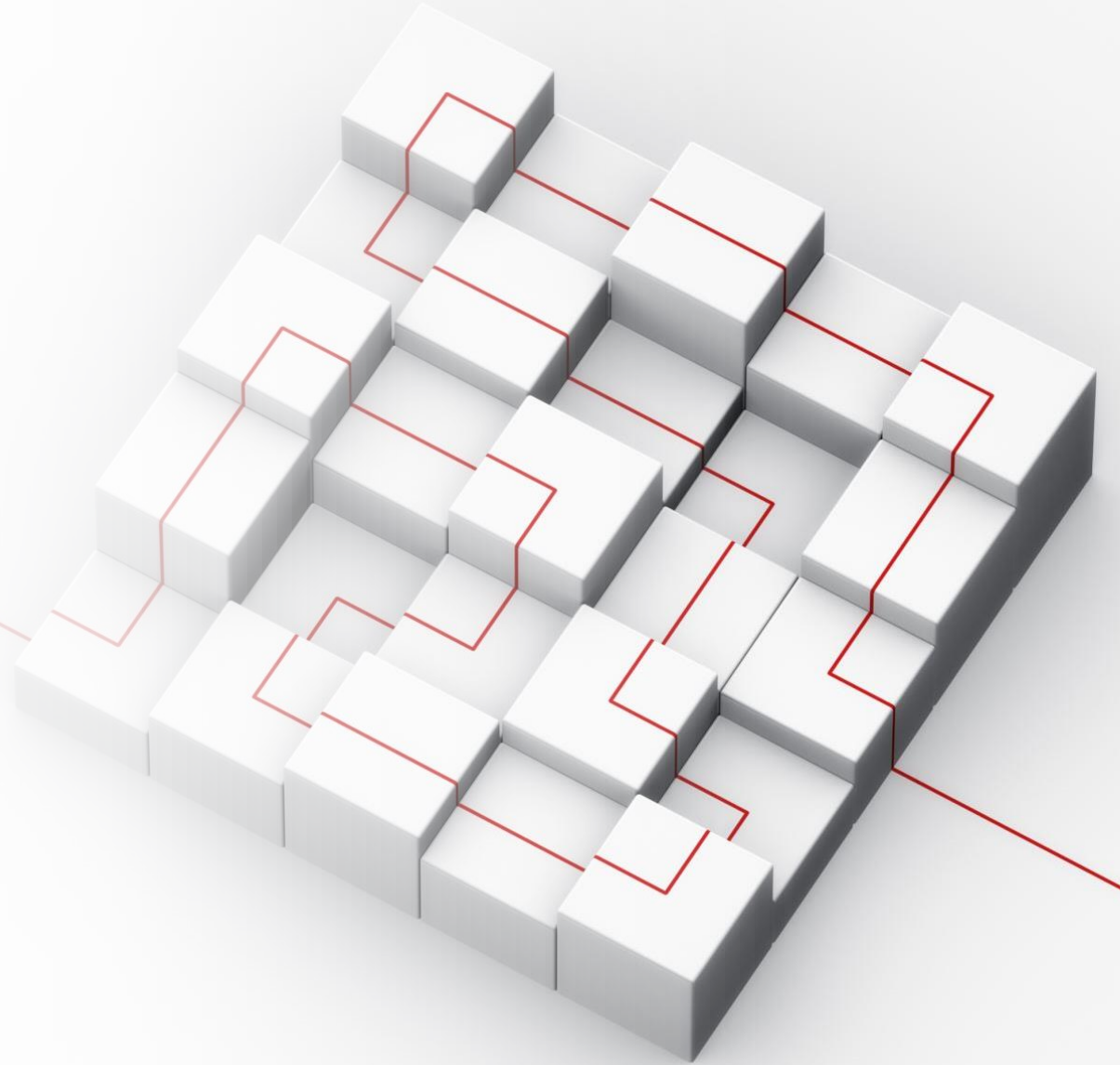


# ICTPRG534

## Deploy applications to production environments

---

TAFE 2024



# Recap

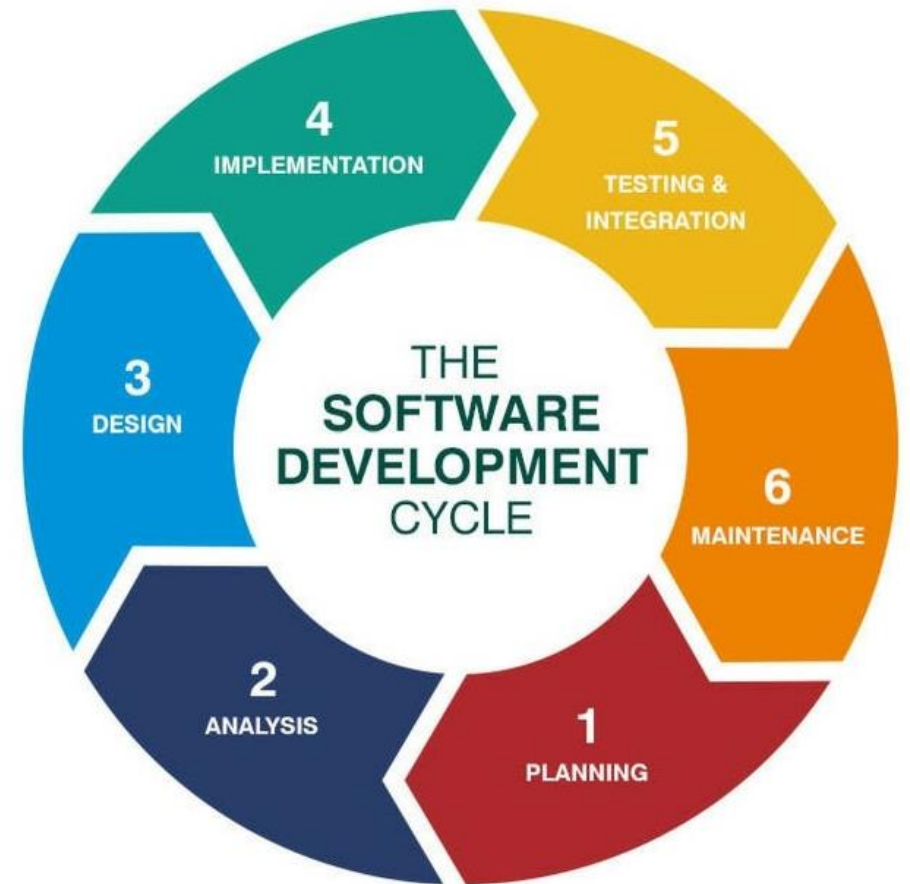
What might we need to document?

- Objectives of the deployment.
- Installation Process & Steps.
- Hardware, Software and Data requirements of the application.
- Security and Organisational requirements of the client.

# SDLC in database

The database system development lifecycle (DBMS SDLC) requires the following phases:

- database planning
- system definition
- requirement collection via fact-finding
- database design
- application design
- DBMS selection
- prototyping
- implementation
- data conversion and loading
- testing
- operational maintenance



# SDLC in database

## **Logical database design**

- Identifies the relevant objects needed for the database design
- Defines the underlying relationships between these objects

## **Physical database design**

- depicts how the logical design will be physically implemented for the database management systems, for example, data represented in a tabular format

## **Fact-finding**

- the formal process for capturing logical and physical database design via techniques such as queries and feedback forms
- uses interviews and questionnaire techniques to collect facts about interacting systems and their requirements
- must apply a fact-finding technique in the early stages of the database system development lifecycle
- when operational maintenance occurs (final phase of the lifecycle), fact-finding techniques determine if a system requires further development

# Prepare specification: software and implementation

**Before deploying a software system to design phase (in the SDLC), a software requirements specification (SRS) must be written.**

## **The goals of a well-written SRS according to the IEEE 830 standard**

- It accommodates feedback to and from the customer with a complete description of levels of functions according to its software specifications.
- It sorts the problem into more solution-based components, which would reduce the development effort.
- It provides a basis for achievable and affordable software solutions with estimating cost and schedules.
- It serves as a basis for software design specification and enhancement.
- It serves and provides a baseline for product validation and verification.

# SRS specifications

A good SRS should demonstrate that it is:

- **correct** – An SRS is correct if, and only if, every requirement stated therein is one that the software shall meet. Traceability makes this procedure easier and less prone to error.
- **unambiguous** – An SRS is unambiguous if, and only if, every requirement stated has only one interpretation. If you find ambiguities, fix them.
- **consistent** – Consistent refers to internal consistency. This means it must agree with some higher-level document, such as a system requirement specification. An SRS is consistent if no subset of individual requirements is in conflict. For example, if you use a tick box in one place, do not call it a check box in another.
- **ranked for importance** – Some requirements in the new system may just be a wish list, and some may not be achievable. You must provide this information in the SRS.
- **verifiable** – Provide quantitative requirements. Do not put requirements such as "the output shall be quick to appear for the corresponding event." Instead, write, "The output shall, in all cases, respond within 30 seconds of the corresponding input event."
- **modifiable** – Refers to whether the structure and style are laid out so that any changes to the requirements can be made easily, completely, and consistently while retaining the structure and style.
- **traceable** – It is traceable if the origin of each of its requirements is clear and facilitates the referencing of each requirement in future development or enhancement documentation.

# Software installation and deployment

You can deploy Windows-based applications from Visual Studio by using either of the following technologies:

- ClickOnce - to publish the application to a centralised location. The user installs or runs the application from that location.
- Windows Installer - to create an application installer file (.msi) and distribute that file. The users run the file to install the application.

[https://learn.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/e2444w33\(v=vs.110\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/e2444w33(v=vs.110)?redirectedfrom=MSDN)

**After the deployment process is complete, the software deployment activities include the following actions:**

- Release the resources necessary for the operations to prepare the system for the client.
- Arrange and deploy the executable components of the software by creating a repository package to install/uninstall the software application several times.
- Deactivate the software system before any update/upgrade/adapt activity.
- Install the version tracking system to install the updates to the software system.
- Uninstall the system that is no longer required.

# Plan a software installation.

## Installation methods

- **Attended installation** requires a user to determine actions during the process. These processes involve acknowledging an end-user license agreement (EULA), installation location, and user passwords and providing the necessary code to activate the product.
- **Silent installation** is a process that does not present any installation windows or notifications during the procedure. An example is malware.
- **Unattended installation** does not require a user during the installation process. It utilises command line switches (also known as answer files) that can be used to fulfil the needs of the installation.
- **Headless installation** is a process that does not require a computer monitor. Instead, a local network is used. This allows a program to be installed on multiple computer machines simultaneously.
- **Automated (scheduled) installation** runs at a scheduled time. When the program is not in use, necessary updates are installed.
- **Clean installation** runs without the presence of other interfering aspects of the program, such as outdated versions or previous installation leftovers.
- **Network installation** uses a shared network resource to install a program. The process is broken up into separate parts where the main infrastructure and its relevant packages form the full program and are downloaded over a network.



# Plan a software installation.

## Installation tools. Installer packaging

- **ClickOnce** allows the user to install and run an application derived from a Windows-based smart client by clicking a link within a web page.
- **Medium scale integration (MSI)**, also known as Microsoft installer, is an installer package format used in Windows for installation, storage, and the removal of programs. Some third-party products can create MSI packages for hassle-free deployment to Windows-based systems.
- **InstallShield** aids in the creation of installers for desktops, servers and cloud platforms in Windows. It also develops installers such as MSI and executable file format (EXE), as well as various Windows server app (WSA) packages.
- **InnoSetup** is a free software packaging and installation tool that can be extended using the Pascal programming language to perform complex tasks during installation. It includes a script wizard to generate a series of installation options without any custom scripts.