technoteach

technocamps

Llywodraeth Cymru
Welsh Government

Prifysgol Abertawe
Swansea University

Cardiff University
Prifysgol Caerdydd

Prifysgol Bangor University

Cardiff Metropolitan University
Prifysgol Metropolitan Caerdydd

University of South Wales
Prifysgol De Cymru

Cyngor Cyllido Addysg Uwch Cymru
Higher Education Funding Council for Wales
hefcw

Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Prifysgol Aberystwyth University

Prifysgol Glyndŵr Wrecsam
Wrexham Glyndŵr University

institute of CODING in wales
technocamps

technoteach
technocamps

Software Development Life Cycle (SDLC)

Software Development Life Cycle

SDLC defines the 'life' stages of software from beginning to end

Your learners are required to know:

- Principles of project management
- Stages of software development
- Legal, ethical and social issues surrounding software development

NOTE: Not to be confused with Systems Development Life Cycle which is similar and *very helpfully* also abbreviated to SDLC within the WJEC spec. We will learn about Systems Development Life Cycle later in the unit

technoteach
technocamps

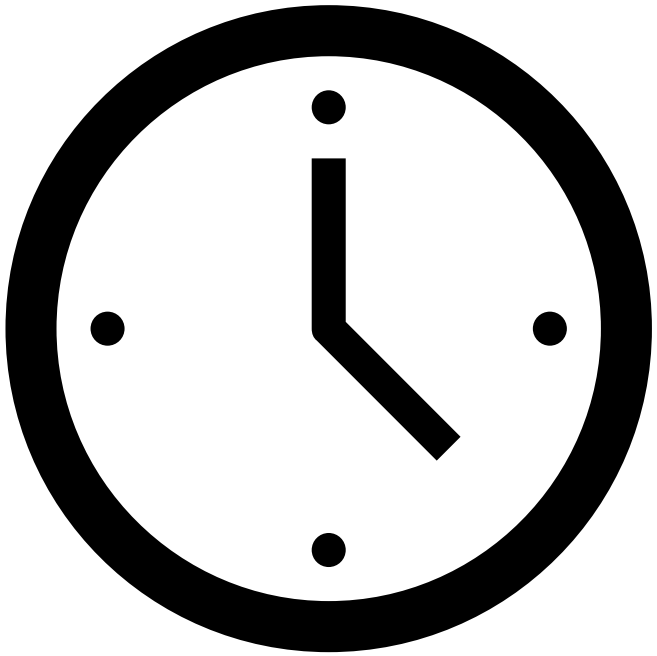
Principles of Project Management

Principles of Project Management

Learners should know the following principles:

- Project initiation & planning
- Project execution & monitoring
- Project control and evaluation

Initiation & Planning



Project Initiation:

- Defining project objectives and scope
- Identifying potential risks and constraints
- Determining stakeholders

Project Planning:

- Planning time, cost and resources
- Identifying tasks and a logical order for completion
- Developing a schedule & budget

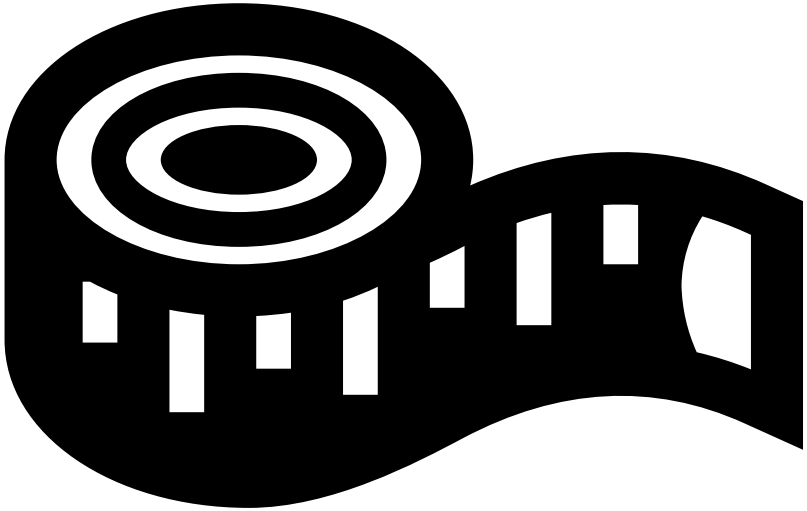
Execution & Monitoring

Project Execution:

- Project tasks are completed as set out in the planning phase

Project Monitoring:

- Measurement of ongoing project tasks
- How close to completion are they?
- *"Where are we?"*



Evaluation & Correction

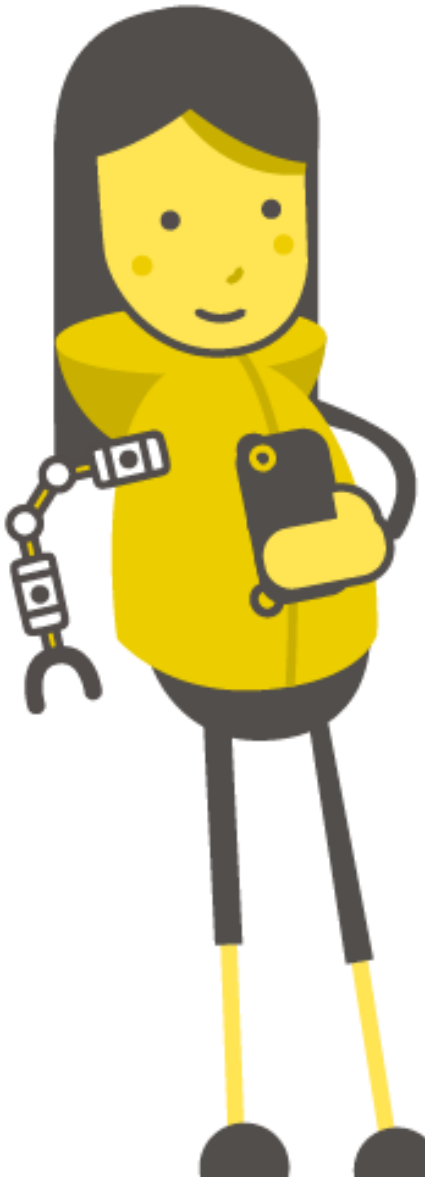
Project Evaluation:

- Comparing current state of the project against the plan
- Considers cost, scope, completion etc
- "*Where should we be?*"

Project Correction:

- Identifying actions that will address project issues
- "*How can we get back on track?*"

Project measurement, evaluation and correction create a cycle until the eventual project closure



Stages of Software Development

Stages of Software Development

Your learners are required to understand the stages of software development:

- Planning & analysis
- Design
- Development & implementation
- Testing
- Deployment
- Maintenance, support & refinement

Planning & Analysis

AKA "Investigation"

The goals & scope of the software development project are established

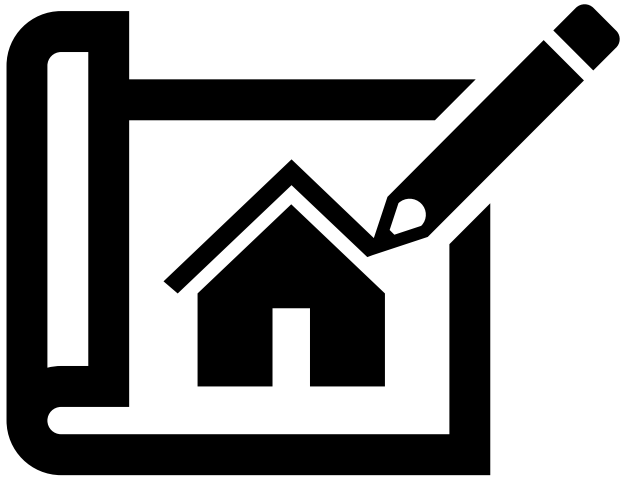
Market research and input from stakeholders are used to gather software requirements

Functions, resources, risks and a project timeline are refined into a software requirement specification

Design

Various aspects of the software's operation are outlined

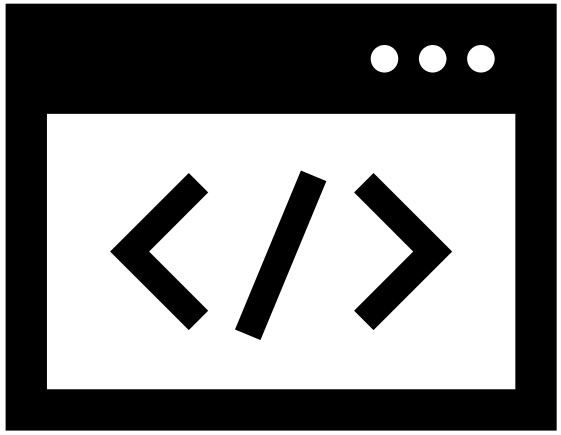
This includes user-facing features such as the user interface (UI) and user experience (UX), but also 'under-the-hood' features such as database design



Rapid prototyping, used to solicit feedback from stakeholders

Ends with the creation of a roadmap

Development & Implementation



The software development team begins writing code and building the software

They are guided by the specification and roadmap generated in the previous stages

The creation of additional systems and interfaces, such as web pages or APIs, that were not clearly defined previously may also be required,

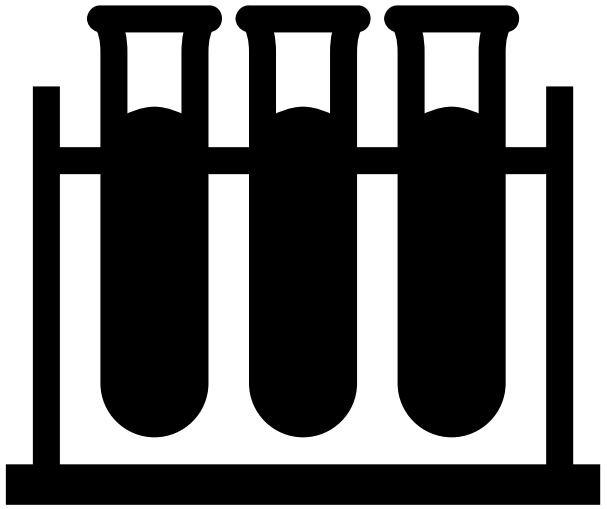
Testing

On the completion of a functional piece of software, it is rigorously tested

Various tests are used to ensure that all components work as intended and meet the software requirements

Further testing is used to identify bugs and security issues before sending back to the developers

Developers implement fixes before sending back for further testing



Deployment

A finely-tuned software product is released to users

Also ensures users understand how to use the software

Software may be deployed in stages – a beta release before the final release

May also involve the creation of manuals, conducting training sessions, and offering support

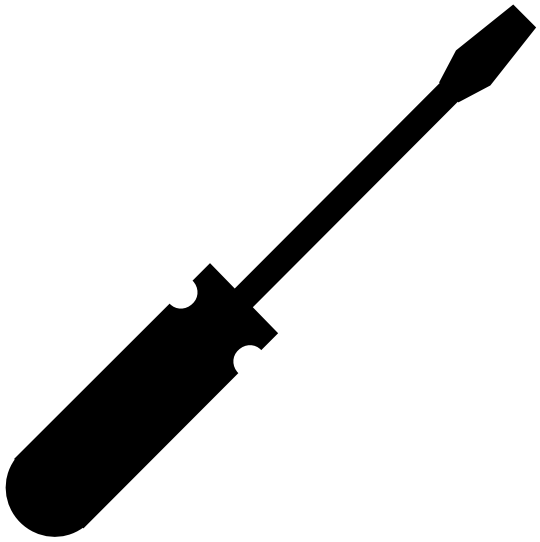
Maintenance, Support & Refinement

Software is continuously updated post deployment

Testing can only do so much and users often find new bugs that require fixing

Small and large software optimisations may be left until post deployment to ensure a timely release

New use cases may arise that require proper implementation



Legal, Ethical & Social Issues

Legal, Ethical & Social Issues

Your leaners should be aware of a variety of legal, ethical and social issues that surround software development:

- Privacy
- Security
- Fairness
- Accessibility
- Transparency
- Intellectual Property
- Environmental Impact
- Social Impact
- Informed Consent
- Accountability

Legal, Ethical & Social Issues

Privacy

- Respect user privacy, minimise data collection, and ensure that any data collected is securely stored and protected from unauthorised access.

Security

- Develop software with robust security measures, including encryption and regular security audits, to protect against data breaches and cyber-attacks.

Fairness

- Avoid bias and discrimination in algorithms and implement measures to ensure that all users are treated fairly, regardless of their background.

Legal, Ethical & Social Issues

Accessibility

• Make software accessible to users with disabilities by following accessibility standards and guidelines and providing alternative input and output options.

Transparency

• Be transparent about software functionality, limitations, and risks, providing clear and comprehensive information to users about how the software operates and what data it collects.

Intellectual Property

• Respect copyrights and licenses, attribute sources correctly, and ensure that all third-party content used in the software is properly licensed.

Legal, Ethical & Social Issues

Environmental Impact

- Minimise energy consumption and electronic waste by optimising software for energy efficiency and supporting the recycling and proper disposal of electronic devices.

Social Impact

- Consider broader societal implications of the software, promoting positive outcomes such as social inclusion, education, and well-being, while mitigating any negative impacts.

Informed Consent

- Obtain explicit and informed consent from users for data collection and processing, ensuring that they are fully aware of what data is being collected and how it will be used.

Legal, Ethical & Social Issues

Accountability

- Take responsibility for the ethical implications of the software, promptly addressing and rectifying any issues that arise, and ensuring that there are clear processes for users to report concerns.