Software Development Methodologies

Higher Computing Science

Development Methodologies

Development methodologies are used to structure and control the process of developing a software solution or information system.

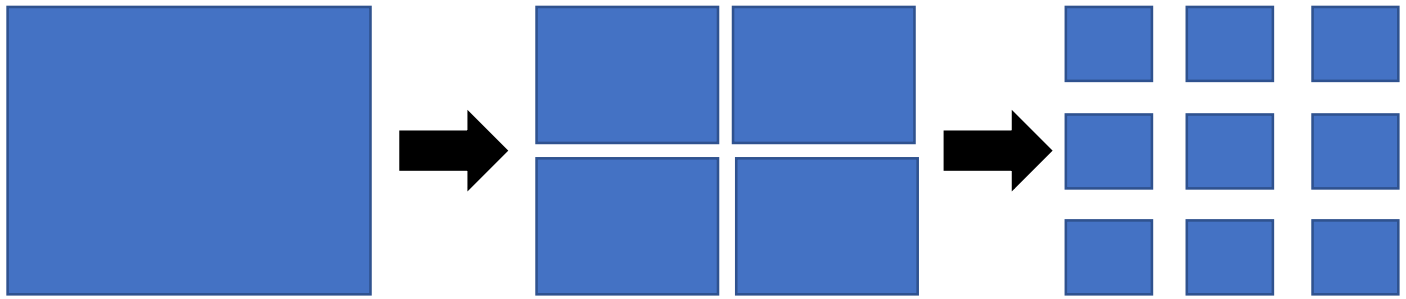
Common methodologies are:

- Top-down
- Agile
- Rapid Application Development

Top-Down

Top-down design involves identifying an overall problem and breaking it down into smaller sub-problems.

The process of **stepwise refinement** is then used to break the sub-problems down until each one is small enough that they are manageable.



Top-Down

Top-down design emphasises planning and a complete understanding of the system.

No coding should take place until a sufficient level of detail has been reached in the design.

Each sub-problem is coded as a module however this delays testing of the functional units until significant design is complete

Rapid Application Development

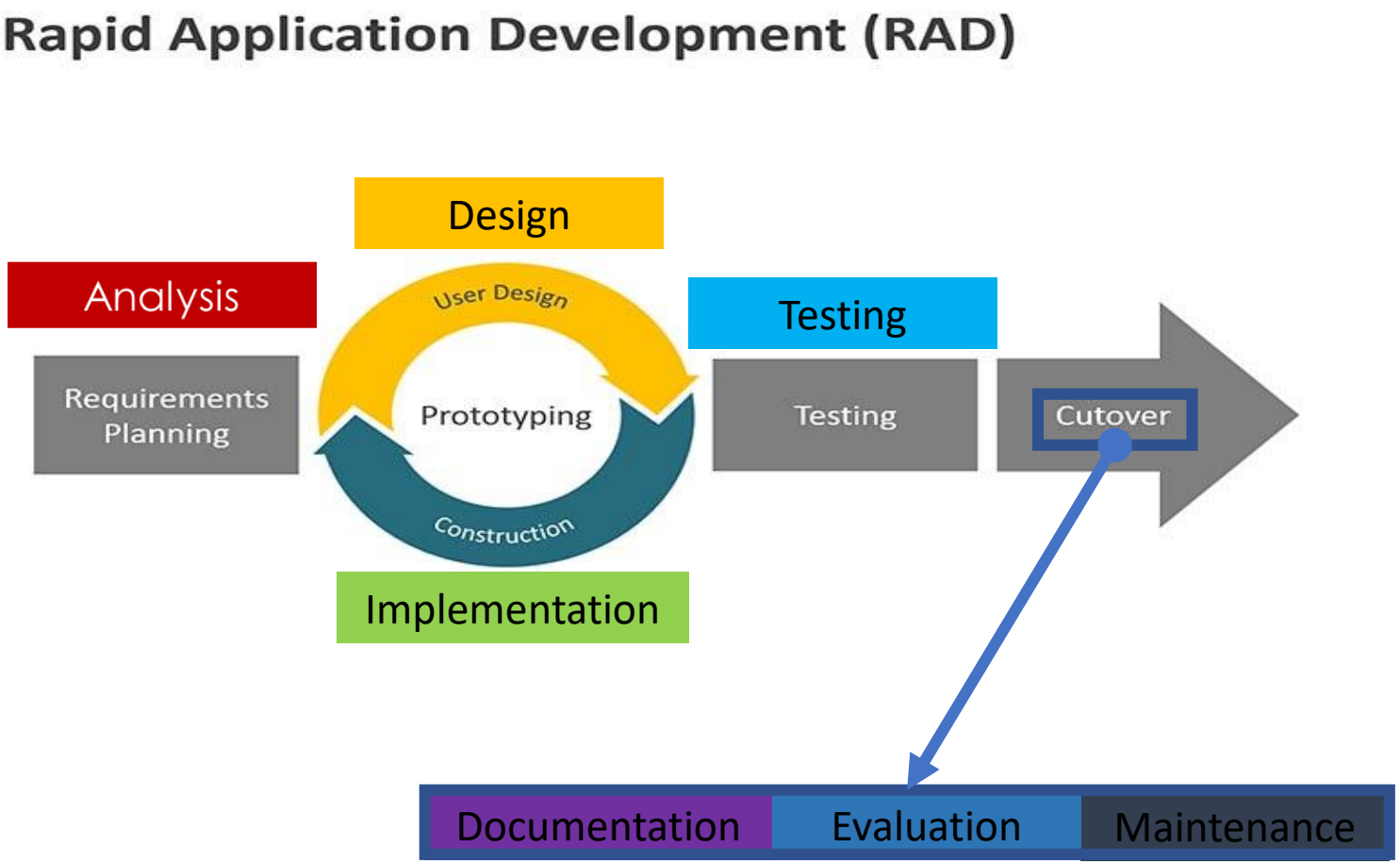
RAD involves reducing the amount of construction needed to develop a software product.

The ability to reduce the development time comes about due to early **prototyping**.

Prototyping involves creating incomplete versions of the software being developed.

A prototype simulates only a few aspects of the final product but enables the client to interact with the system at an early stage.

Rapid Application Development (RAD)

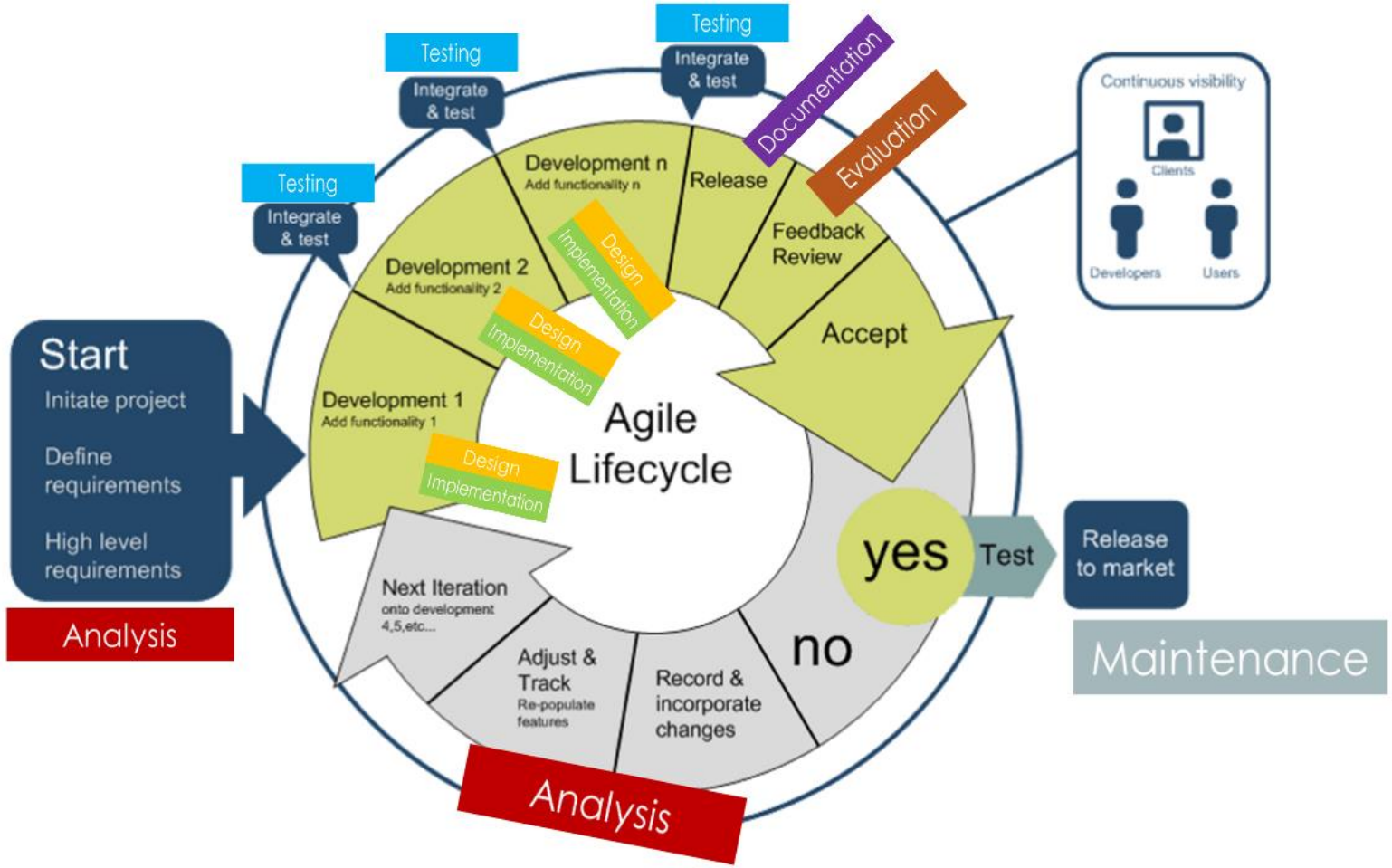


Rapid Application Development

Benefits of Prototyping

- Development team gains valuable early feedback from users.
- Client can compare if the software being developed matches the specification and meets their needs.
- Improves the quality of requirements and specifications provided to developers.
- Because changes cost more to implement the later they are detected, early determination of what the user really wants can result in faster implemented and less expensive software.

Agile Methodologies



Agile Methodologies

Agile development emphasises real time, face-to-face communication involving all the people necessary to finish the software. Very little written documentation is produced.

Software is developed in short iterations, each one like a miniature software project of its own.

The purpose of a single iteration is not to produce the final completed solution, but to add additional functionality that produces working software.

After each iteration, the project priorities are then re-evaluated.

Agile Methodologies

Top-down attempts to produce software by assuming a perfect understanding of the client's requirements from the start.

In reality top-down rarely delivers what the client wants as the client often doesn't know exactly what they want until they see it.

Agile methodologies embrace iterations where small teams work to develop quick prototypes.

During each iteration, working software is produced following which the requirements for the next iteration can be evaluated.