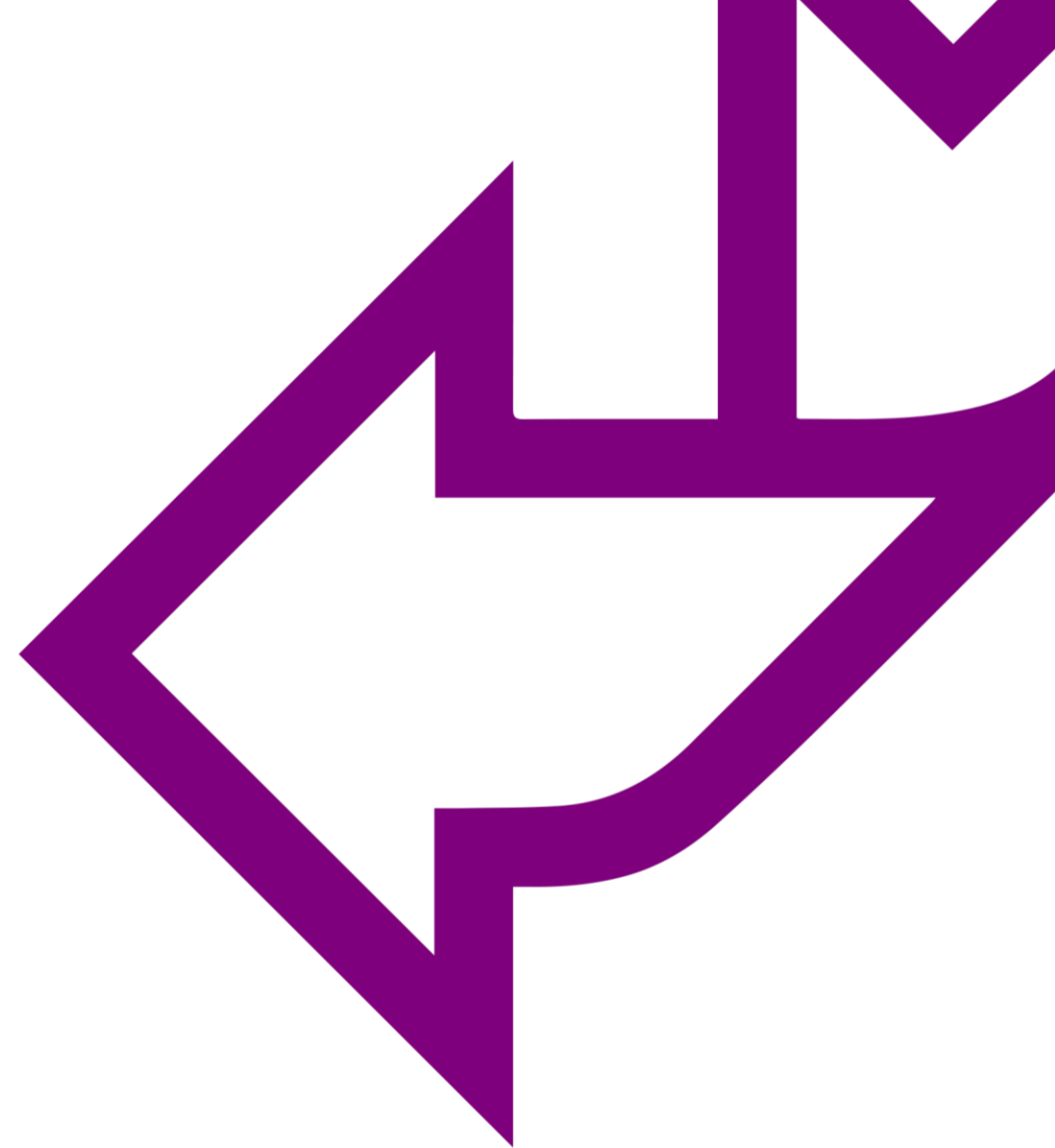




Software Developer

Agile Software Development

Software Development Methodologies





What is a software development methodology?

Early days of systems development:

Relatively unstructured: success depended on individuals.

Needed more structured and manageable approach.

Reduce random human element: build more complex systems.

Is the process of splitting of development work into distinct phases / stages:

- Each containing activities with the intent of better planning and management. Considered a subset of the systems development life cycle.
- Pre-definition of specific **deliverables** and artefacts.
- Created and completed by a project team

QA Types of system development lifecycle

Many SDLCs in use today: understand to make appropriate choice.

Sequential: linear stages / tasks

Waterfall

V Model

Evolutionary: evolves through progressive versions

Incremental: mini waterfalls

Iterative: DSDM, RAD, RUP, Rapid Prototyping, Spiral

Agile: XP, Scrum

QA **Scope of system development**

Work does not end when system released

- New / updated requirements
- Faults fixed and improvements made
- Maintenance and support

Most SDLCs just cover development and implementation but must also consider

- Ongoing support and maintenance should be considered
- Including the end-of-life and decommission

QA Sequential methodologies

Stages in a defined order

- Each completed and agreed before next started.
- Formal review and sign-off.
- Project and payment milestones.

Suitable when

- Requirements well understood and stable.
- Rigour and documentation needed – quality.
- Need to closely control cost and scope.

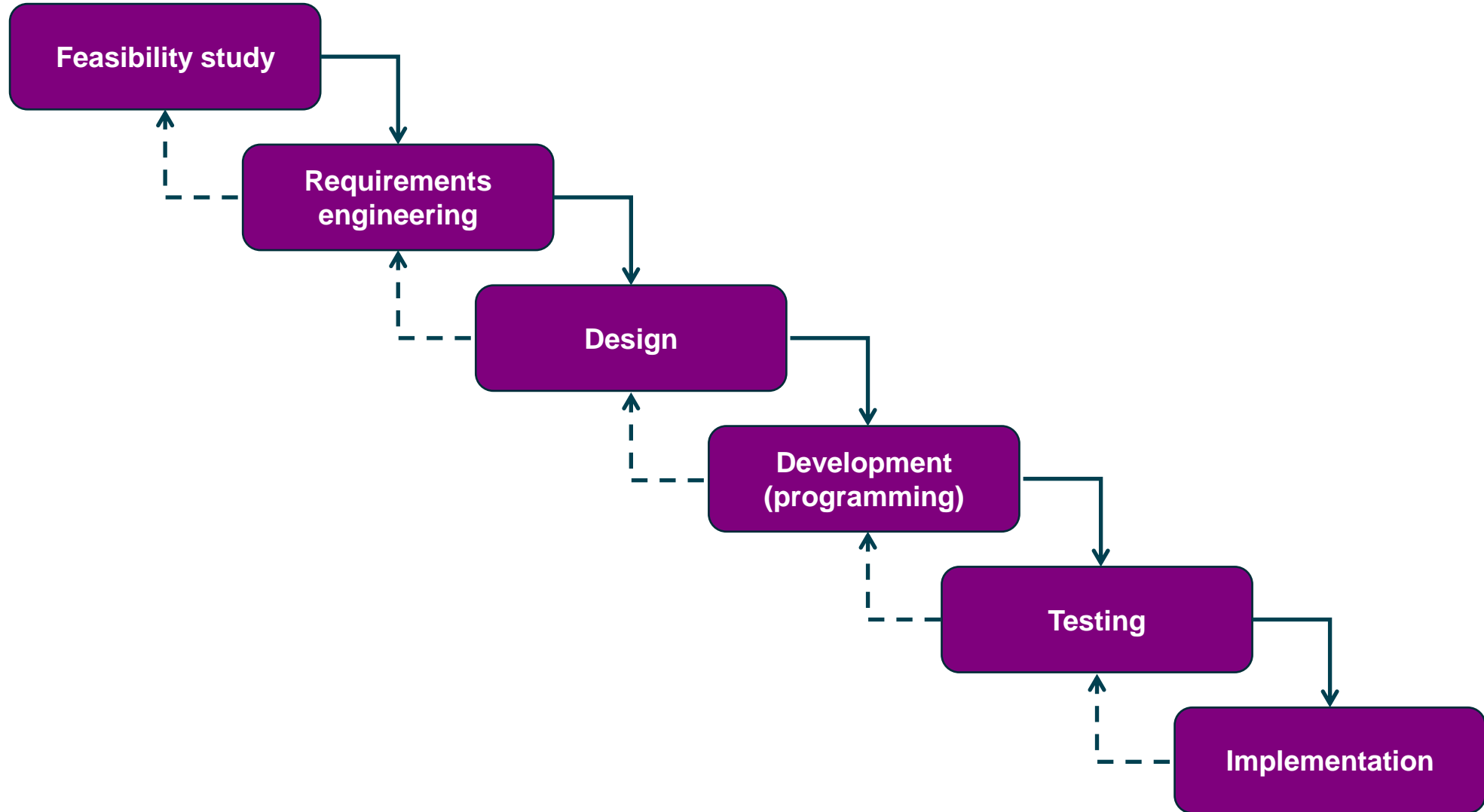
QA Sequential methodologies: Strengths and weaknesses



- | | |
|--|--|
| <ul style="list-style-type: none">● Distinct stages, clear stages● Agree / review each stage
No need to revisit● Details designs and specifications● Controls cost and scope creep● Simple and intuitive | <ul style="list-style-type: none">● Depends on each stage done properly● Long timescales for complex projects● Changing requirements● No value until implemented● If stopped early, no value |
|--|--|



Waterfall





Waterfall: Strengths and weaknesses



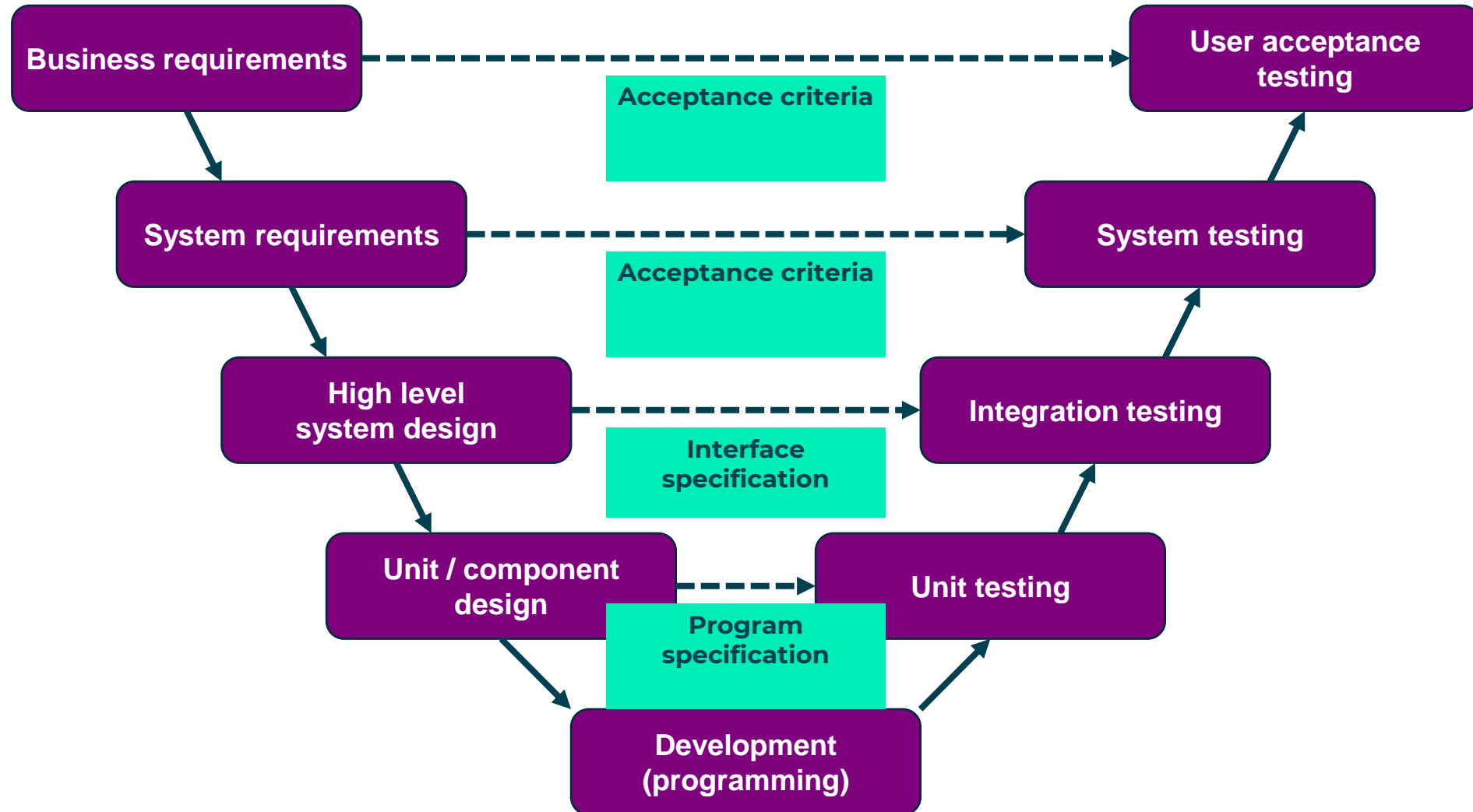
- Safe and signed off
- Thorough
- Quality built in
- Easy to manage
- Design based on all requirements
- Good documentation



- Difficult to go back
- Lack of user involvement
- No early vision of system
- Difficult to add missing requirements
- Business change has high impact
- Fosters silo mentality



V model





EVOLUTIONARY METHODOLOGIES

Solution developed through iterative releases:

- Basis of Agile development.

Suitable when

- requirements are not well understood.
- new products where it's difficult to define what needed.
- iterations uncover understanding.
- demo working software to gather feedback.
- early delivery more important than completeness.
- early benefits – justify funding.
- quick to market – high value features first.
- high business / technical risk.

If not feasible, stop!



Revolutionary methodologies: Strengths and weaknesses



- Early delivery of software.
 - Minimises project risk.
 - Handles complex requirements.
 - Increased collaboration with business / users.
 - Do just enough – refine later.
- Hard to manage project due to multiple teams, iterations and complex products
 - Danger of scope creep
 - Costs higher due to extra integration & testing
 - Danger of over promising early functionality



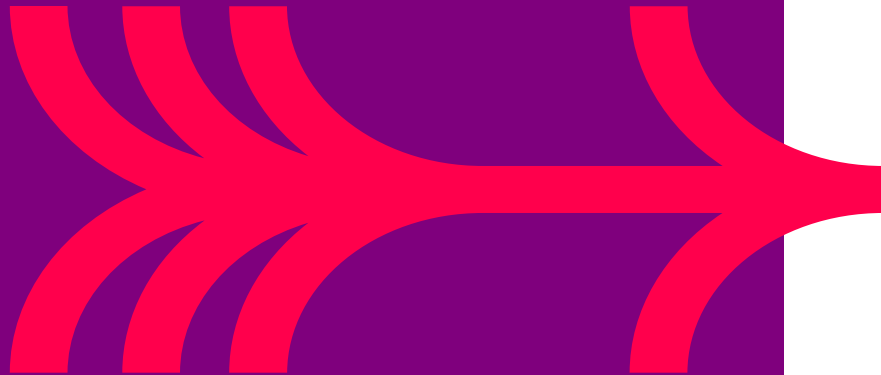
EVOLUTIONARY METHODOLOGIES

Of the 5 basic lifecycles 3 follow an evolutionary approach

- Incremental
- Iterative
- Spiral

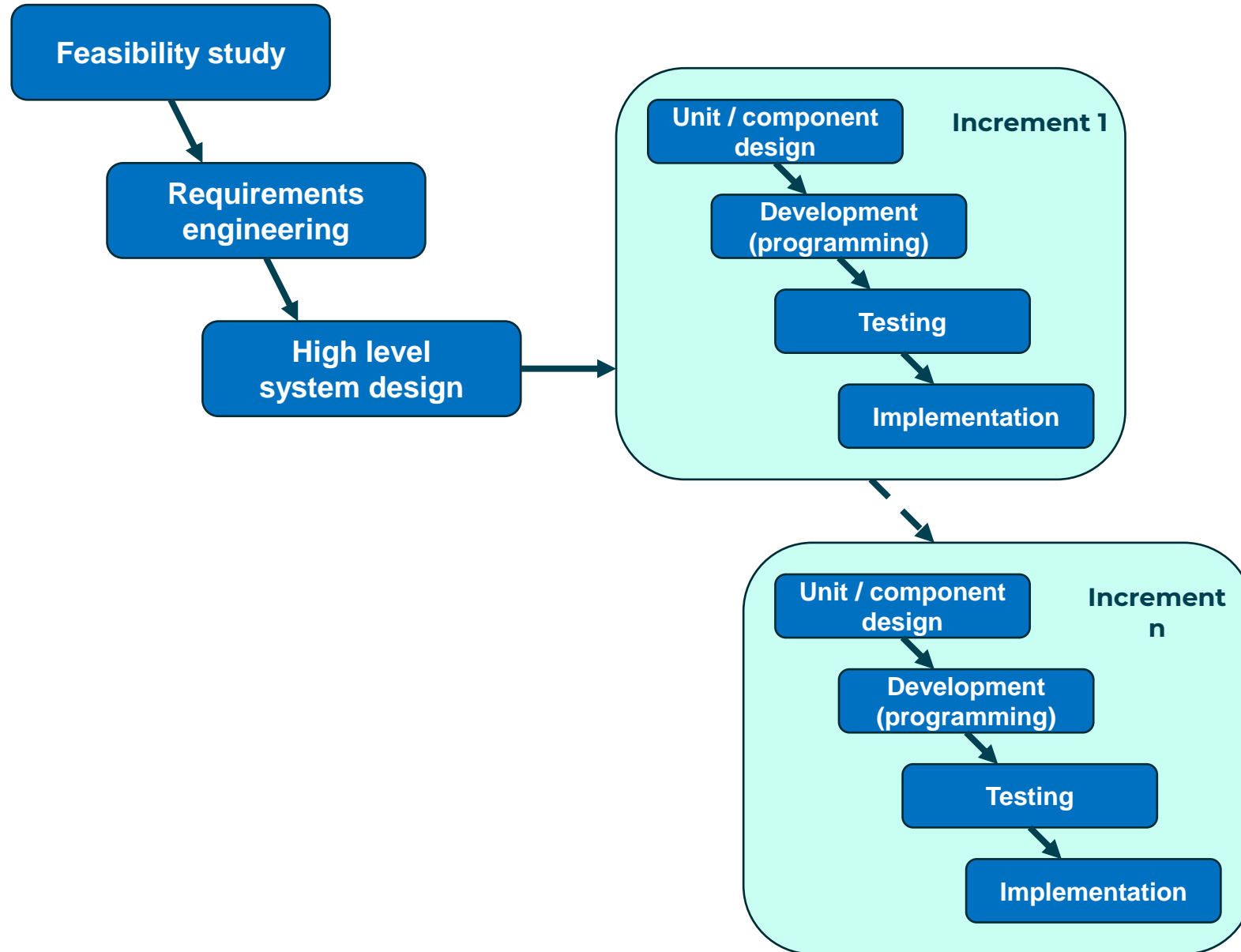
Evolutionary approaches around since 1980s.

- Underpin more popular Agile approaches today.





Incremental



QA Incremental real-life example

Involves gradually building up features.

Incremental development slices the features into fully working slices/increments.
Each new increment builds on top of the existing released.

For example, in building an e-commerce website

In the initial release increment, the team establishes fundamental purchasing functionality, encompassing search, product details, adding items to the shopping basket, and the checkout process.

This initial segment is only released upon its full completion.

Subsequently, the second release increment expands upon the foundational functionality, introducing an additional capability like favourites. This increment is released upon the completion of the favorites functionality.



Incremental: Strengths and weaknesses

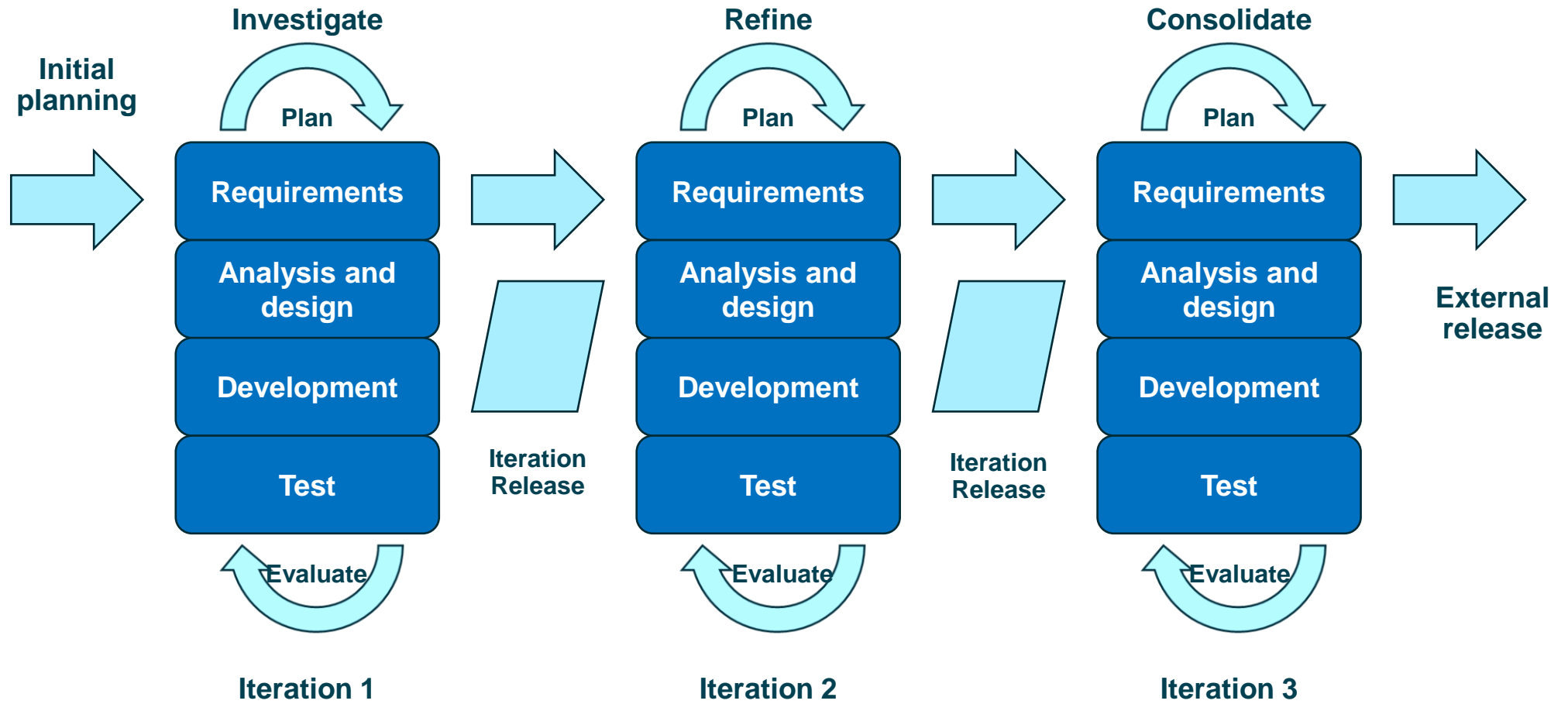


- Delivers software early.
- Easier to manage risks by identifying and handling in early increments.
- Easier to test and debug smaller increments.
- Overall delivery time can be reduced by building increments in parallel.



- Hard to manage parallel deliveries.
- Still hard to manage change as requirements are agreed before increments are decided.
- Overall costs can be higher due to the additional overheads of parallel implementation and additional regression testing.

QA Iterative



QA Iterative real-life example

Involves gradually building up features.

Teams release basic versions of features and enhance them in subsequent iterations, incorporating feedback and making improvements based on stakeholders' input and reviews

For example, in building an e-commerce website iteratively, the initial release includes stripped-down versions of features like Search, Product information, Cart management, Checkout, favourites, and reviews.

Subsequent releases improve existing functionality, add new features, and may eliminate low-value areas based on feedback and evolving requirements.



Iterative: Strengths and weaknesses



- Requirements evolve.
- Encourages collaboration.
- Change easier to manage.
- Cost can be controlled.

- Hard to manage due to multiple iterations.
- Evolving requirements result in scope creep.
- Can become like single Waterfall project.
- Higher cost due to additional integration and testing activities.

QA Agile manifesto

- **Agile** is a popular term now given to evolutionary and iterative approaches.
- Meet many of today's growing business challenges.
- Agile manifesto developed in 2001 by 17 leading IT professionals.
- **We will investigate Agile later**

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

That is, while there is value in the items on the right,
we value the items on the left more.

QA Agile: Strengths and weaknesses



- Earlier delivery of software.
- Risks identified and handled sooner
- Requirements evolve and change easier to manage.
- Encourages collaboration so more customer buy-in.
- Costs controlled.

- More complex to manage multiple iterations and teams that may be dispersed.
- Danger of scope creep.
- Not suited when there are audit, regulatory, safety-critical requirements in strict contractual commitments.
- Additional costs of testing, integration & deployment.

Methodology differences

Feature	Sequential (waterfall)	Evolutionary (Agile)
Structure	Single pass of linear steps Backtracking difficult	Repeated passes Backtracking possible
Value	At end of project	At end of each increment / iteration
User involvement	Generally, only at requirements and testing stages	Each and every stage
Time	Solution only delivered after all stages completed	Regular releases delivered for feedback and approval
Manpower	More specialist / silo approach	Cross-functional, collaborative teams
Changes	More difficult to accommodate / backtrack	Include in future increments / iterations
Risk	Not highlighted until late in the project	Identified and resolved earlier

review





QUESTIONS AND FEEDBACK

