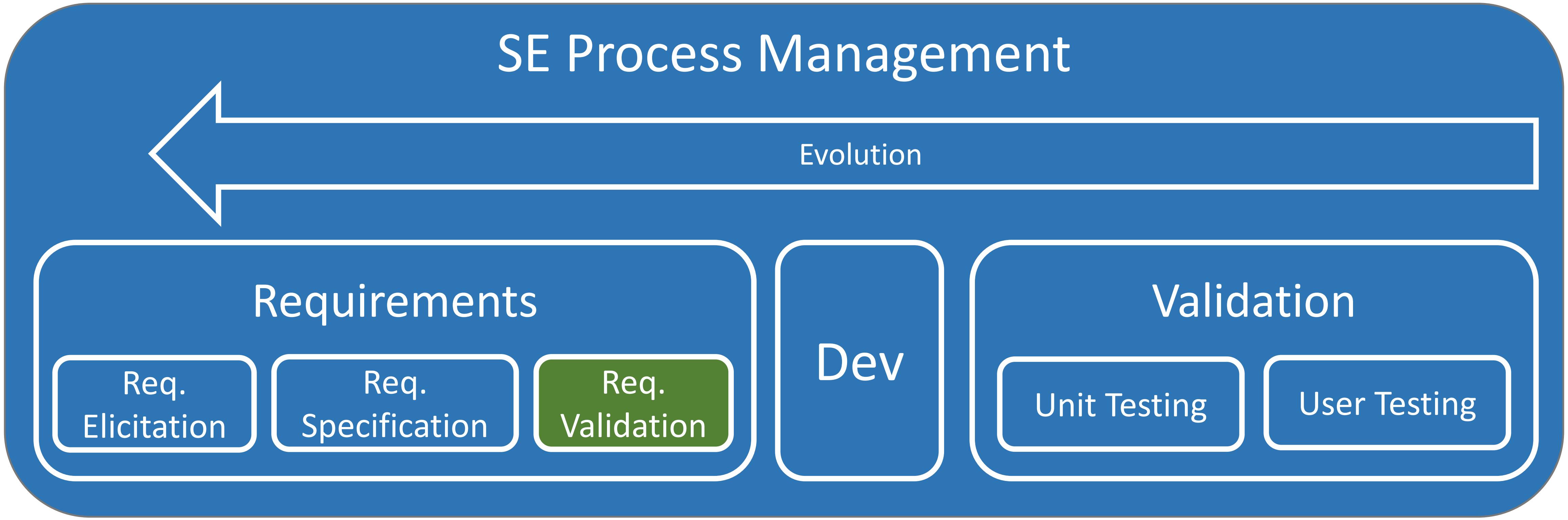
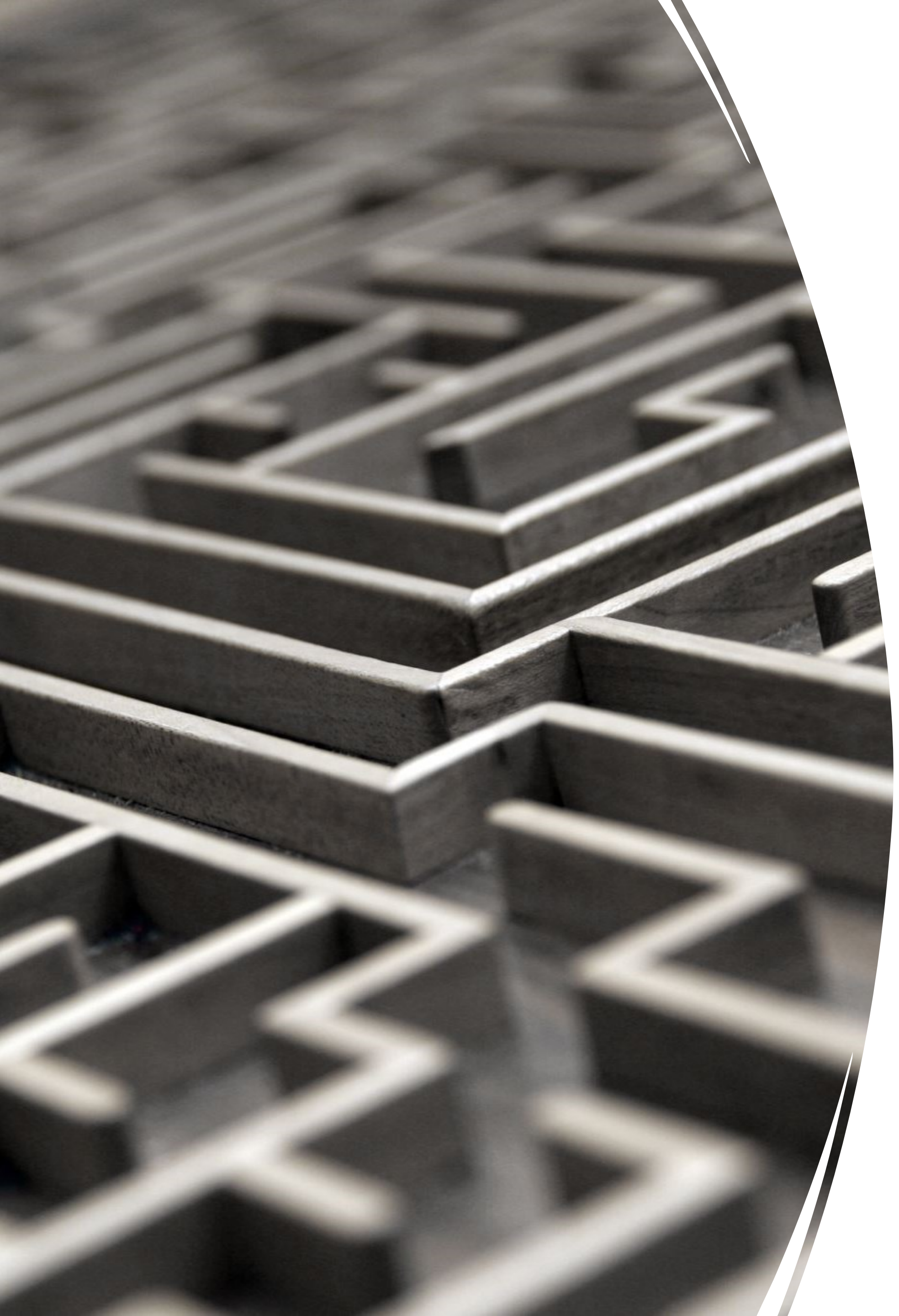
Software Engineering COMP1035

Lecture 07

Requirements Validation

Keeping Track of SE Module

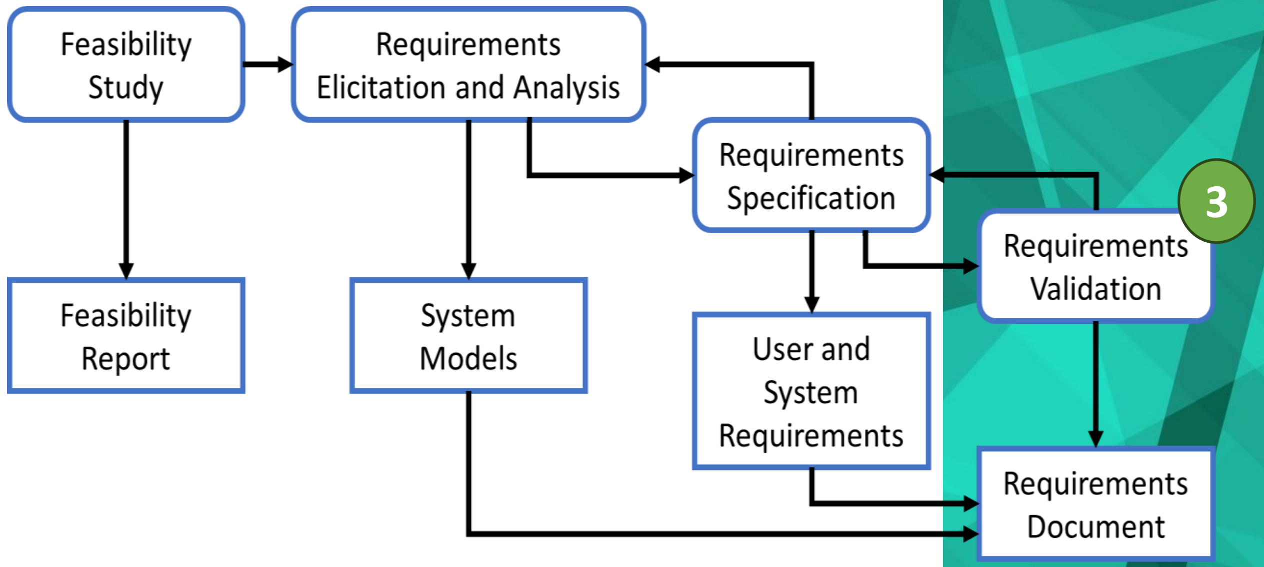




Today’s Learning:

1. What is requirements validation?
2. Who are involved in the requirements validation?
3. Requirements change process.

What is Requirements Validation?



Requirements Validation

- Process of **checking that requirements define the system that the customer really wants**.
- Critically important – errors in requirements document can lead to lower rework costs.
- Cost of fixing a requirements problem after deployment is much greater than repairing design or coding errors.

Requirements Validation

"The requirements are analyzed systematically by a team of reviewers who **check for errors and inconsistencies**."

- There is a formal review process you can go through.
  - Several people in a room, **reading each requirement a loud**.
- Each person takes a ROLE – to systematically review the requirements:
  - **Validity** checks (are the areas of functionality identified as necessary?).
  - **Consistency** checks (do requirements conflict with one another?).
  - **Completeness** checks (does it specify a coherent system or only parts of it?).
  - **Realism** checks (can requirements actually be implemented?).
  - **Verifiability** checks (can requirements be tested?).
- Are the requirements correct, necessary, important?

Why We Do Requirements Validation

Checking that You are Right

Avoiding ReWorking

Contractually Agreeing

1. Checking That You Are Right

"... the process of checking that requirements actually **define the system that the customer really wants**"

- The SE Book

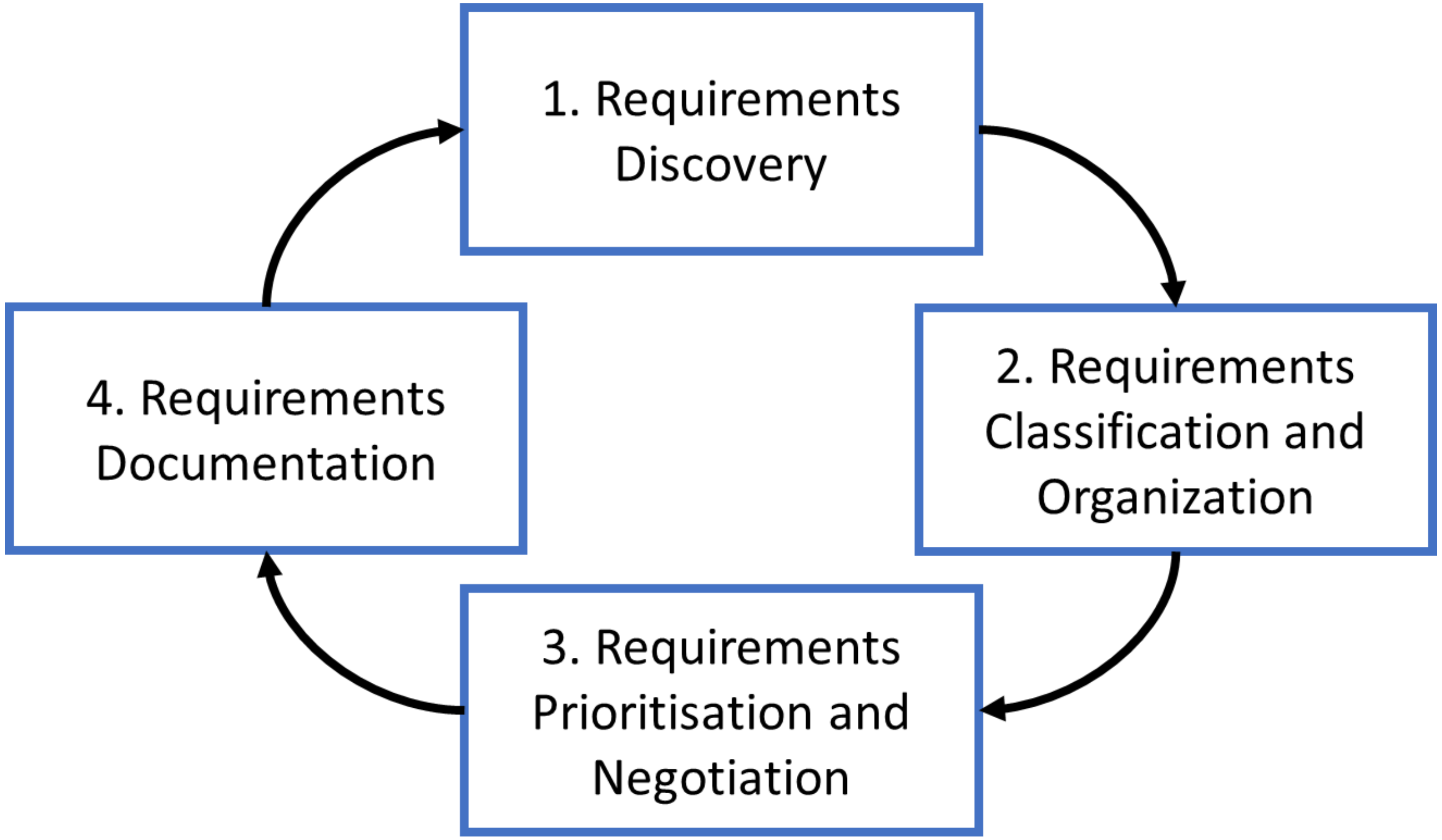
2. Avoiding ReWorking

"… errors in a requirements document can lead to extensive rework costs …
The **cost fixing a requirements problem by making a system change is usually greater than repairing design or coding errors**."

- The SE Book

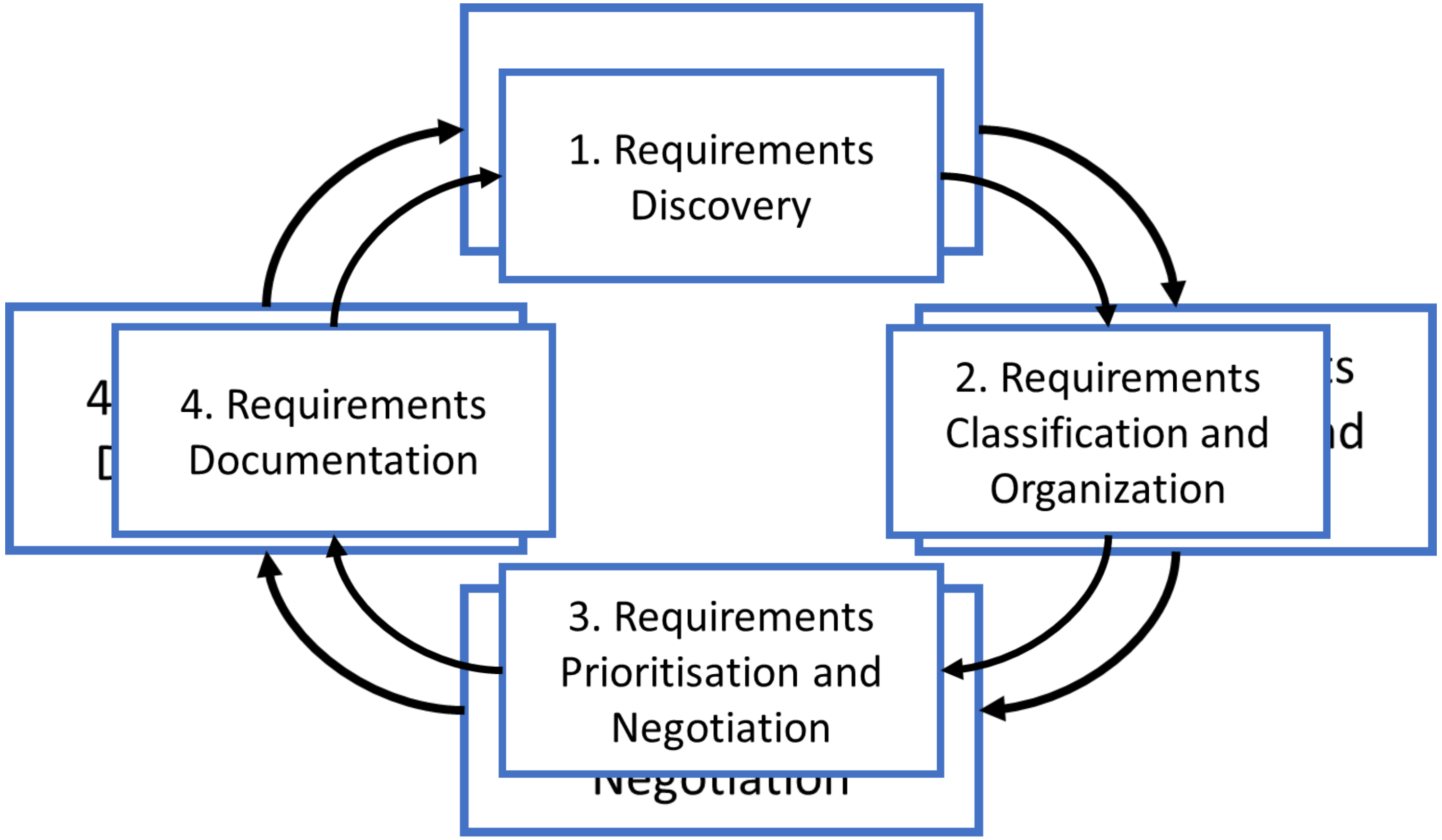
Requirements Validation Cycle

Repeated process.


1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritisation and Negotiation
4. Requirements Documentation

Requirements Validation Cycle

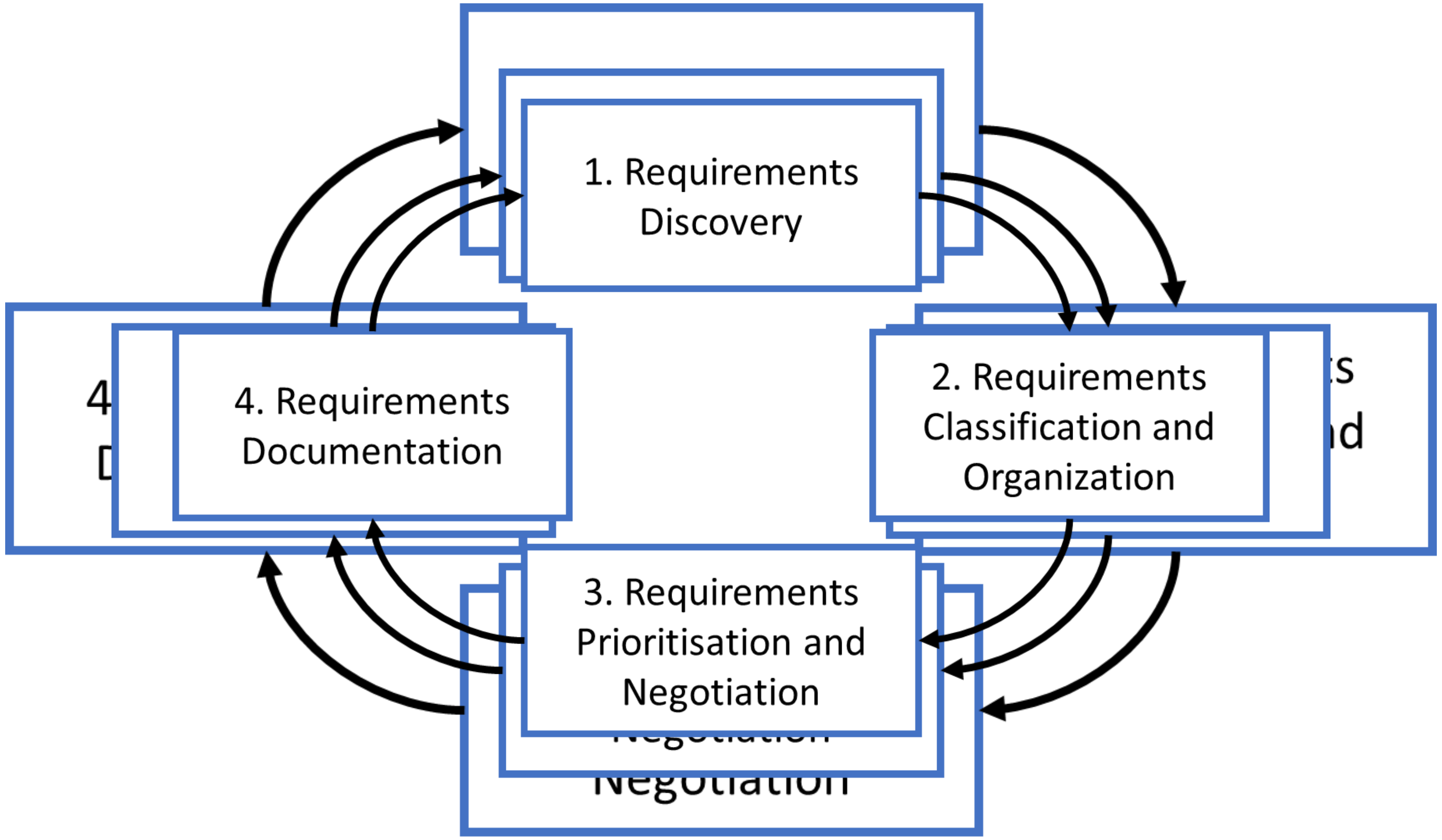
Repeated process.



1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritisation and Negotiation
4. Requirements Documentation

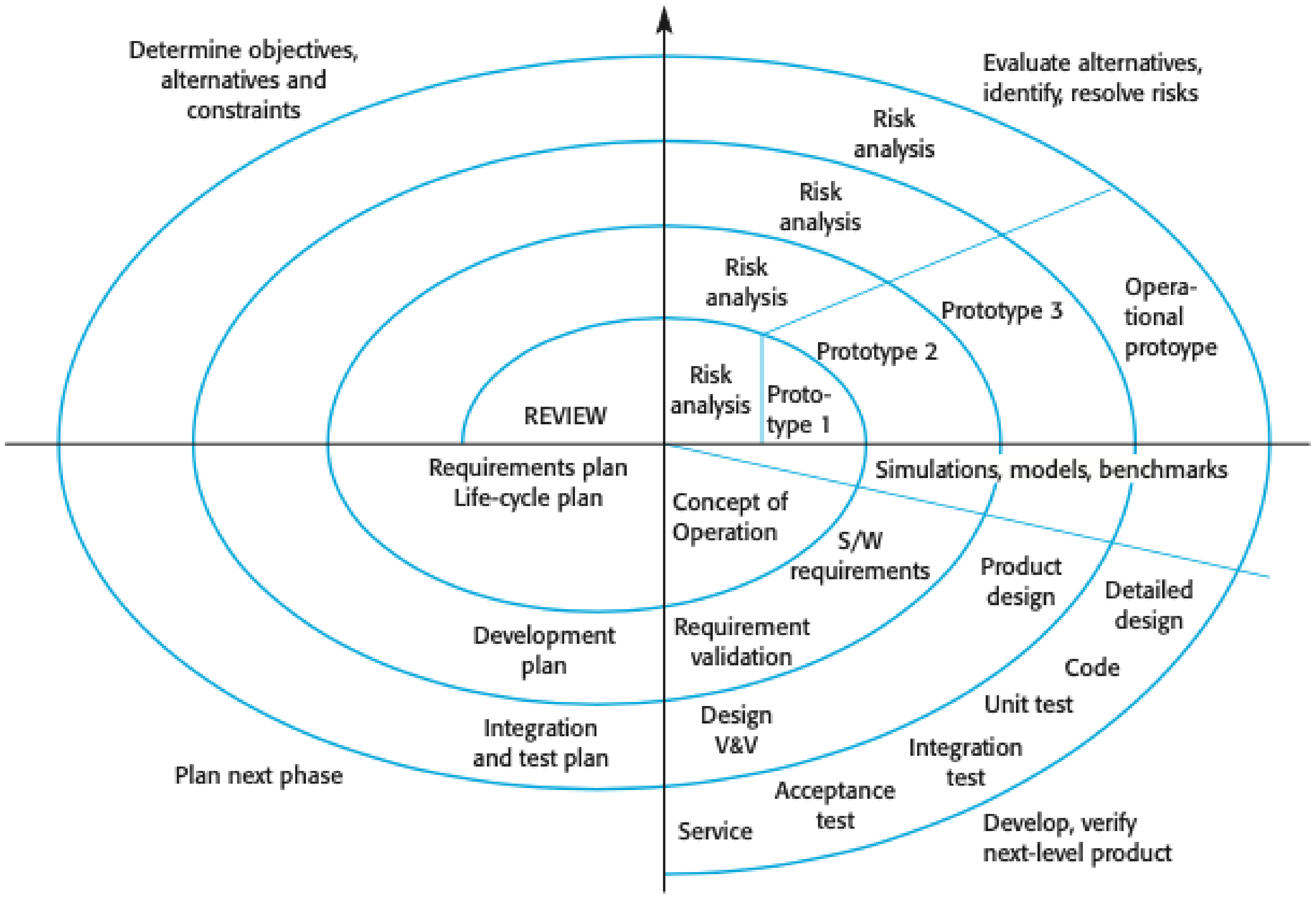
Requirements Validation Cycle

Repeated process.


1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritisation and Negotiation
4. Requirements Documentation

Requirements Validation Cycle

Starts to look like a spiral model



3. Contractually Agreeing

- At some point in a project, you must decide what **exactly what to build**.
- If this is for a customer, you want all the stakeholders and the team to agree exactly what will be built.
  - Otherwise, you and the customer may have vastly different ideas.
  - You cost for your idea.
  - **NOTE: They don't pay until their idea is achieved**.

Requirements Validation Techniques

- **Requirements reviews**:
  - Requirements are analysed systematically by a team who check for errors and inconsistencies.

- **Prototyping** (Lecture 08):
  - Developing an executable model of a system.
  - Use the model with end-users and customers to verify their needs and expectations.
  - Stakeholders can experience with the modeled system and provide feedback.

Requirements Validation Techniques

- **Test-case generation**:
  - Requirements should be testable.
  - If test is difficult or impossible to design, this commonly signifies that the requirements are challenging or impossible to design.
  - Developing tests from the user requirements before coding (test-driven development, TDD).

Requirements Validation

- You present it to your “boss and colleagues”.
  - You had to explain things to the audience.
  - This is a first sanity check.
  - Does it make sense when you tell your “boss”.

Internal

- You present it back to participants/clients/users.
  - Do they agree with your understanding?
  - Do they agree with what you think is “most important”.

External

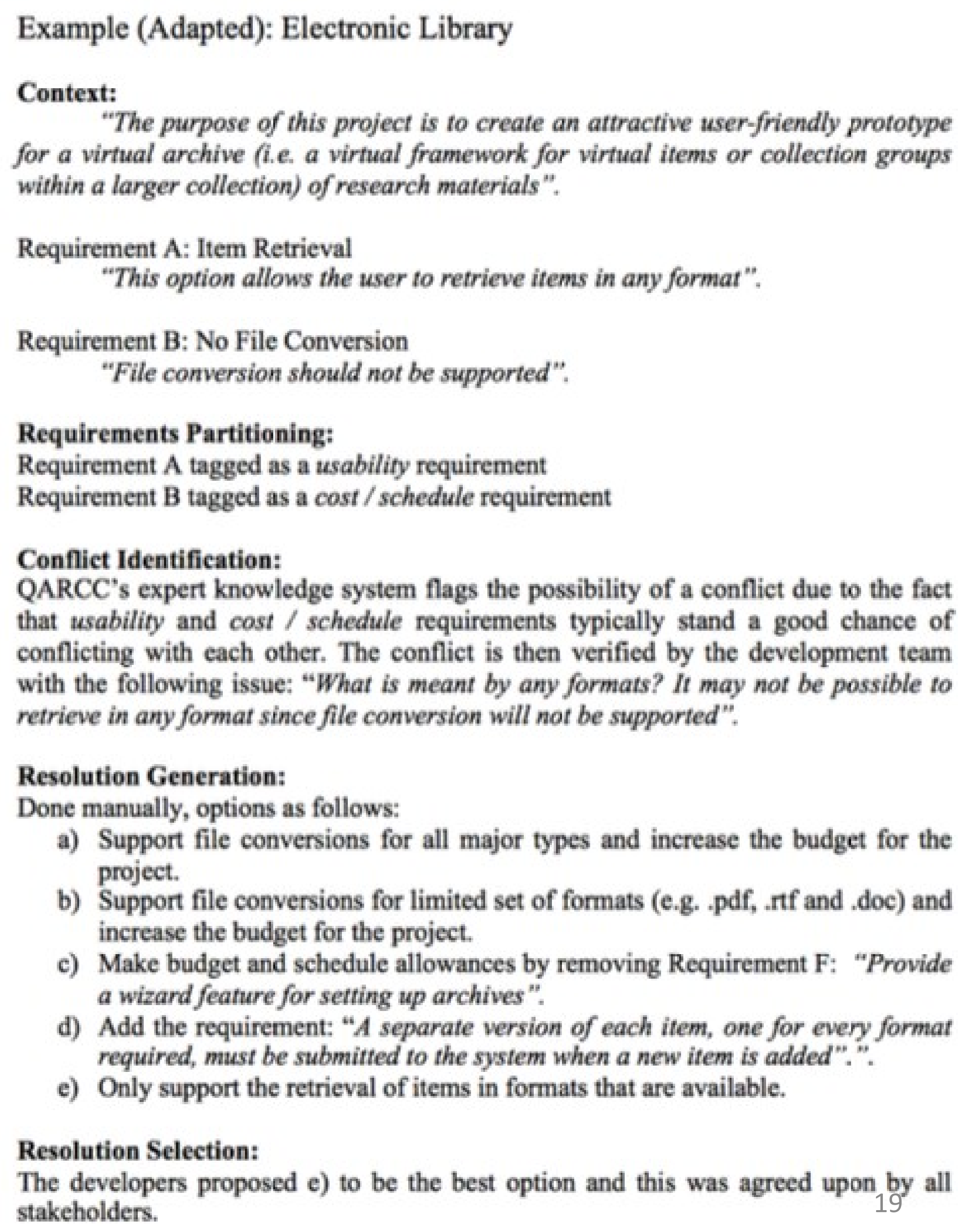
Requirements Validation – Internal

- Using a focused method – a Requirements Review which appear in several stages.
- You want to do this with your team first.
  - It's like a practice run for when you present things to your client.
    - The full requirements, the time plan, the requested budget to achieve it, etc.
- Get a manager, the requirements leader, a developer, a quality manager, and the client manager (if different) together in a room.
- Two **benefits**:
  1. The client manager gets a clear picture before taking it to the client.
  2. If you can explain it to them, without having trouble, then you are ready to take it to the client.

Requirements Conflicts?

- What happens when you find gaps in your understanding?
    - You are not ready to move to external validation!
- If it's a missing aspect, then you need to do more elicitation.
- It it's a conflict, you need to document the conflicting ideas.
    - And resolve with client.
- Don't go to full external validation until you are ready.

https://camilofitzgerald.wordpress.com/



Example (Adapted): Electronic Library

Context:

"The purpose of this project is to create an attractive user-friendly prototype for a virtual archive (i.e. a virtual framework for virtual items or collection groups within a larger collection) of research materials".

Requirement A: Item Retrieval

"This option allows the user to retrieve items in any format".

Requirement B: No File Conversion

"File conversion should not be supported".

Requirements Partitioning:

Requirement A tagged as a *usability* requirement

Requirement B tagged as a *cost / schedule* requirement

Conflict Identification:

QARCC's expert knowledge system flags the possibility of a conflict due to the fact that *usability* and *cost / schedule* requirements typically stand a good chance of conflicting with each other. The conflict is then verified by the development team with the following issue: "*What is meant by any formats? It may not be possible to retrieve in any format since file conversion will not be supported*".

Resolution Generation:

Done manually, options as follows:

a) Support file conversions for all major types and increase the budget for the project.
b) Support file conversions for limited set of formats (e.g. .pdf, .rtf and .doc) and increase the budget for the project.
c) Make budget and schedule allowances by removing Requirement F: *"Provide a wizard feature for setting up archives"*.
d) Add the requirement: "*A separate version of each item, one for every format required, must be submitted to the system when a new item is added*".".
e) Only support the retrieval of items in formats that are available.

Resolution Selection:

The developers proposed e) to be the best option and this was agreed upon by all stakeholders.

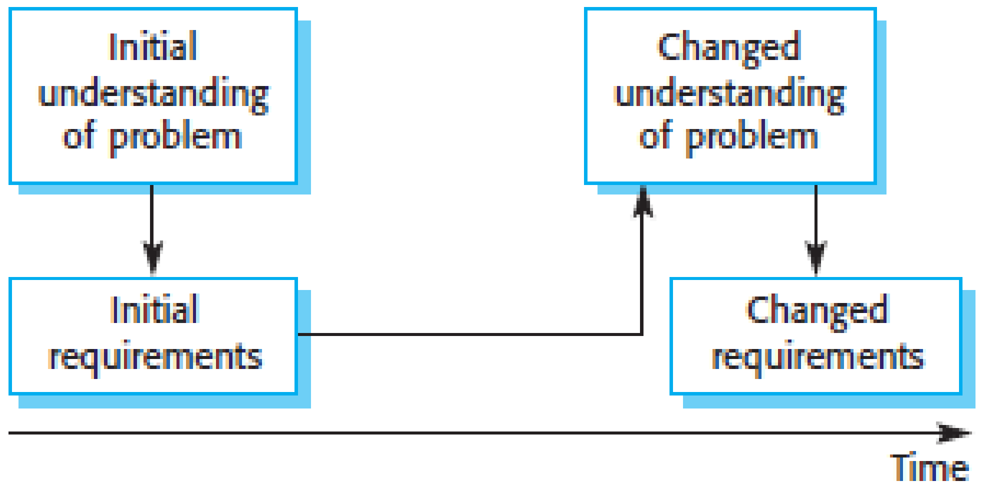
Requirements Validation – External

- But at some point, you've got to agree a plan with a client.
  - Essentially a Requirements Review with the client.
  - And you don't want to look like a fool when you do (hence you should do the internal validation first).
- This time the people in the room are the **key people from both companies**.
  - Probably not a developer, and tester and quality manager, etc.
  - But e.g., the manager, and finance manager from the client's side.
- By the end of this final validation, you should have "the plan"
  - Budget, time, requirements etc.
  - Performance indicators and evaluation metrics
  - Because if you are wrong, it's only going to create delay, or take you over budget.

Requirements Change

Requirements Change

- Requirements for large software systems are **always changing**.
  - To address problems that cannot be completely defined.
  - Requirements need to be evolved to reflect this changed problem.
- Once system is installed and regularly used, new requirements always pop-up.
  - Due to **errors** in the original requirements that need corrections.


Initial understanding of problem
Changed understanding of problem
Initial requirements
Changed requirements
Time

Requirements Change

- Changes of business environment of the system:
  - Constant changes of business and **technical environment of the system** – new hardware or update of existing hardware; introduction of new regulations which require system compliance.
  - **Funders and users of the systems are usually different** – end-user requirements are not fully implemented due to budgetary constraints.
  - Conflicting or **contradictory priorities** among diverse stakeholders.

Requirements Management (Later Lecture)

- Formal **process for making change proposals** and linking these to system requirements.
- Should start as soon as a draft of the requirements document is ready.

- Agile development processes are designed to cope with requirements that changes during the development process (later lecture).
  - Does not go through formal requirements management process.
  - Changes usually benefit some stakeholders and not others (hard to satisfy everyone).
  - Need of independent authority to balance the needs of all stakeholders.

Summary

- Requirements Validation
  - Process to verify the requirements with customers.
  - Check for errors and inconsistencies.
- Techniques
  - Requirement reviews
  - Prototyping
  - Test-case generation
- Requirements conflict?
  - How to solve it?
  - Internal vs external validation

THANK
YOU