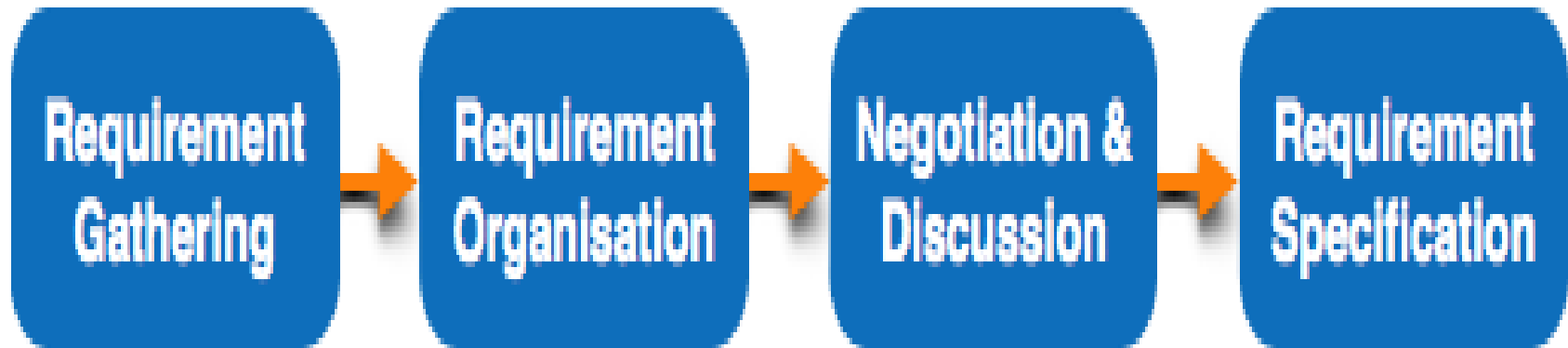


Chapter Three

Requirement Elicitation and Analysis

Requirements Elicitation and Analysis

- Requirements Elicitation is the process to find out the requirements for an intended software system by communicating with client, end users, system users and others who have a stake in the software system development.
- Requirement elicitation process can be depicted using the following diagram:



Cont'd...

- **Requirements gathering** - The developers discuss with the client and end users and know their expectations from the software.
- **Organizing Requirements** - The developers prioritize and arrange the requirements in order of importance, urgency and convenience.
- **Negotiation & discussion** - If requirements are ambiguous or there are some conflicts in requirements of various stakeholders, if they are, it is then negotiated and discussed with stakeholders.
 - Requirements may then be prioritized and reasonably compromised.
- ❑ The requirements come from various stakeholders.
 - To remove the ambiguity and conflicts, they are discussed for clarity and correctness.
 - **Unrealistic requirements** are compromised reasonably.
- **Documentation** - All formal & informal, functional and non-functional requirements are documented and made available for next phase processing.

Requirement Elicitation Techniques

- **There are various ways to discover requirements.**
 - ✓ **Interviews**
 - ✓ **Surveys**
 - ✓ **Questioners**
 - ✓ **Task analysis**
 - ✓ **Domain analysis**
 - ✓ **Scenarios**
 - ✓ **Brainstorming**
 - ✓ **Prototyping**

Cont'd...

1. Interviews

- Interviews are strong medium to collect requirements.
- Organization may conduct several types of interviews such as:
 - **Structured (closed) interviews**, the requirements engineer looks for answers to a pre-defined set of questions.
 - **Non-structured (open) interviews**, where there is no pre-defined agenda and the requirements engineer discusses, in an open-ended way, what stakeholders want from the system.
 - more flexible and less biased.
 - **Oral interviews**
 - **Written interviews**
 - **One-to-one interviews** which are held between two persons across the table.
 - **Group interviews** which are held between groups of participants.
 - They help to uncover any missing requirement as numerous people are involved.

Cont'd...

2. Surveys

- Organization may conduct surveys among various stakeholders by querying about their expectation and requirements from the upcoming system.

3. Questionnaires

- A document with pre-defined set of objective questions and respective options is handed over to all stakeholders to answer, which are collected and compiled.
- A **shortcoming** of this technique is, if an option for some issue is not mentioned in the questionnaire, the issue might be left unattended.

Cont'd...

4. Task analysis

- Team of engineers and developers may **analyze the operation** for which the new system is required.
- If the client already has some software to perform certain operation, it is studied and requirements of proposed system are collected.

5. Domain Analysis

- Every software falls into some domain category.
- The **expert people in the domain** can be a great help to analyze general and specific requirements.

Cont'd...

6. Scenarios

- Bridging the gap between user and developer:
 - *Scenarios*: Example of the use of the system in terms of a series of interactions with between the user and the system
 - *Use cases*: Abstraction that describes a class of scenarios

7. Brainstorming

- An informal debate is held among various stakeholders and all their inputs are recorded for further requirements analysis.

Cont'd...

8. Observation

- Team of experts visit the client's organization or workplace.
- They observe the actual working of the existing installed systems.
- They observe the workflow at client's end and how execution problems are dealt.
- The team itself draws some conclusions which aid to form requirements expected from the software.
- ❖ **Ethnography** is a technique from the social sciences which has proved to be valuable in **understanding actual work** processes.
 - involves an observer spending an extended period in a society or culture, making detailed observation of all their practices.

Cont'd...

9. Prototyping

- A prototype is **a demonstration system** which shows end-users and system stakeholders what facilities the system can provide.
- If you have vague or poorly understood requirements, you should consider developing a prototype system which **simulates the behavior of system software and hardware.**
- End-users can experiment with this to refine their ideas about the system requirements.

Benefits of prototype

- I. Prototyping makes the real meaning of requirements easier to understand.
 - If stakeholders do not fully understand the system requirements, they may agree to a requirements specification which does not reflect their real needs.
 - The ultimate system which is developed is likely to be unsuitable.
- II. Prototyping is the only effective way of developing system user interfaces.
 - If a prototype has been developed as part of the requirements process, this can reduce later development costs for the system.

Cont'd...

III. The prototype system may help establish the overall feasibility and usefulness of the system before high development costs are incurred.

IV. For some types of system, which are primarily software,

- the **final system** can sometimes be developed by modifying and adding facilities to the prototype.
- This means that a limited but usable system may be available relatively quickly.

Implementation of prototypes

- There are three possible approaches to system prototyping:
 - I. Paper prototyping** where a mock-up of the system is developed and used for system experiments.
 - II. 'Wizard of Oz' prototyping** where a person simulates the responses of the system in response to some user inputs
 - III. Automated prototyping** where a fourth generation language or other rapid development environment is used to develop an executable prototype.

Paper prototyping

- Is a **cheap** and surprisingly **effective** approach to prototype development.
- It is **most suitable** for establishing end-user requirements **for software systems**.
- Paper versions of the screens which might be presented to the end-user are drawn and various usage scenarios are planned.
- Analysts and end-users work through these scenarios to find users reactions to the system, the information they require and how they would normally interact with the system.

'Wizard of Oz' prototyping

- Is also **relatively cheap** as it does not require much software to be developed.
- The user interacts with what appears to be the system but his or her inputs are actually channeled to a person who simulates the responses of the system.
- **This approach is particularly useful** when a new system has to be developed based on an existing interface.
- Users are familiar with the interface and can see the interactions between it and the system functionality simulated by the 'Wizard of Oz'.

Automated Prototyping

- ❑ Developing an executable prototype of the system is a more expensive option.
 - It involves writing software to simulate the functionality of the system to be delivered.
 - It is very important that the prototype should be developed quickly and this means that you should use very high level languages and support environments for prototype development.

Cont'd...

- ❑ Depending on the class of system you might use any of the following.

1. Fourth generation languages based around database systems.

- These are good for prototyping applications which involve **information management**.
- However, they do have restrictions which are inherent in the interaction facilities which they provide.
- You must develop your prototype within these limitations and this may mean that some facilities (e.g. navigation around a database by using a graphical visualization of the data) may be impossible.

Cont'd...

2. Very high-level languages such as Visual Basic or Smalltalk.

- These are general purpose programming language which usually come with a powerful development environment and access to a range of reusable objects.
- These **allow applications to be developed quickly** and with more flexibility than is usually provided with 4GLs.

Cont'd...

3. Internet-based prototyping solutions based on World-Wide-Web browsers and languages such as Java.

- Here, you have a ready-made user interface.
- You add functionality to it by associating segments of Java programs with the information to be displayed.
- These segments (called **applets**) are executed automatically when the page is loaded into the browser.
- This approach is a fast way of developing user interface prototypes but you must accept the inherent restrictions imposed by the browser.
- At the time of writing, this approach seems to have a lot of potential but is still immature.

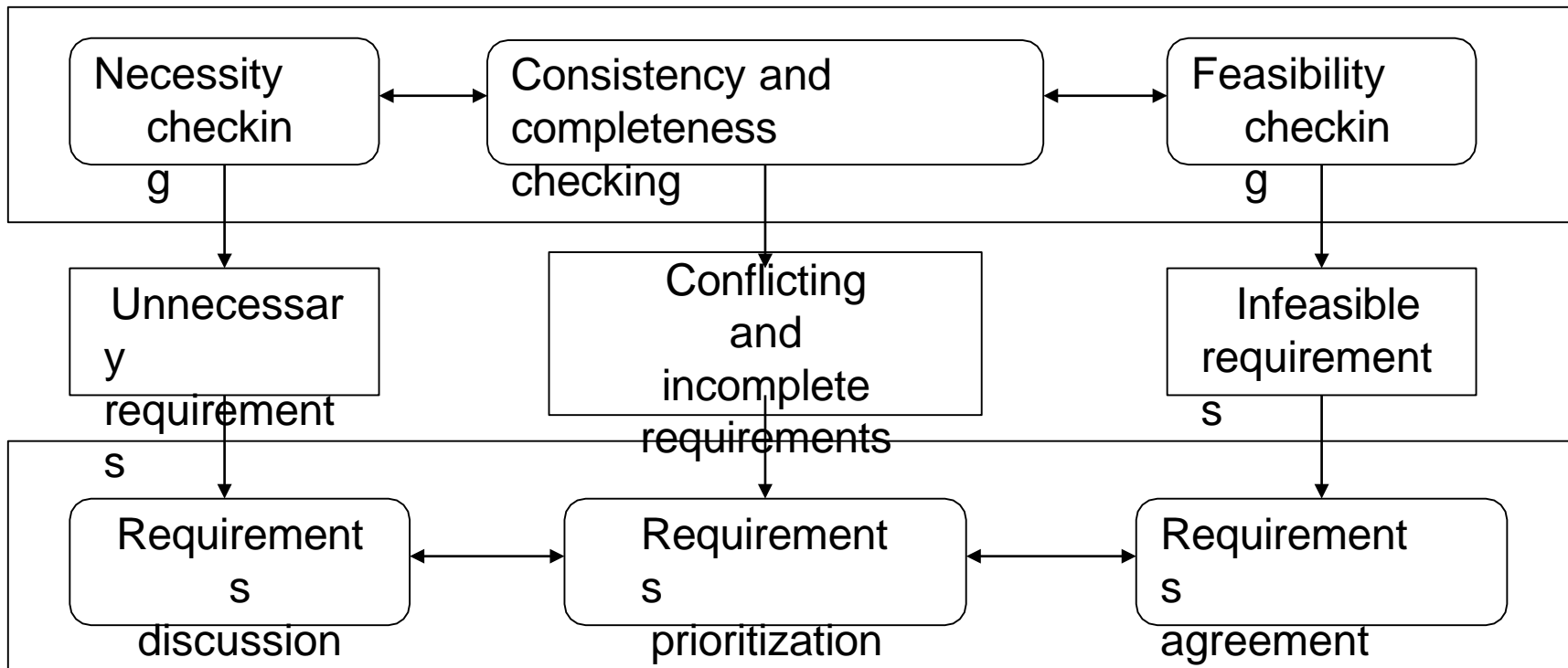
Cont'd...

- ❖ In general, because the performance, efficiency, reliability, etc., of a prototype is quite different from a finished system, you can only use prototypes to discover the functional requirements for a system.

Requirements Analysis and Negotiation

- Requirements analysis and negotiation are inter-leaved activities and join to form a major activity of the requirements engineering process.

Requirements Analysis

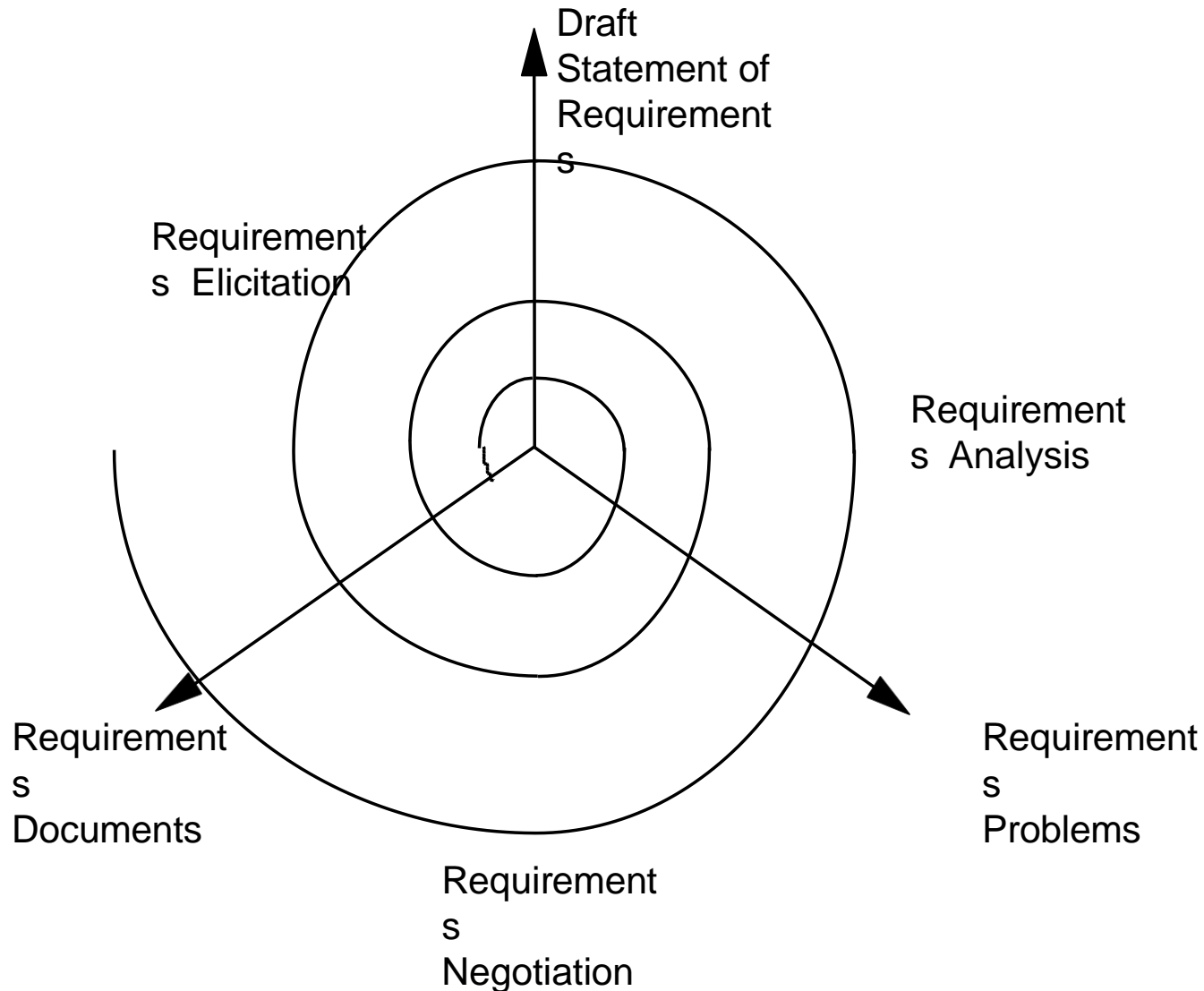


Requirements Negotiation

Requirements Analysis

- The **aim** of requirements analysis is to **discover problems with the system requirements**, especially incompleteness and inconsistencies.
- Some analysis is inter-leaved with requirements elicitation as problems are sometimes obvious as soon as a requirement is expressed.
- ❑ Detailed analysis usually takes place after the initial draft of the requirements document is produced.
- **Analysis is concerned with incomplete set of requirements, which has not been discussed by stakeholders.**

Iterative Aspects of Elicitation, Analysis, and Negotiation



Comments on Requirements Analysis

- **Analysts** read the requirements, highlight problems, and discuss them in requirements review meetings.
 - This is a time-consuming and expensive activity.
- Analysts have to think about implications of the draft statements of requirements.
- ❖ People do not think in the same way and different analysts tackle the process in different ways.
- It is not possible to make this activity a structured and systematic process.
 - It depends on the judgment and experience of process participants.

Requirements Analysis Stages

- I. Necessity checking
- II. Consistency and completeness checking
- III. Feasibility checking

Necessity Checking

- The need for the requirement is analyzed.
- In some cases, requirements may be proposed which don't contribute to the business goals of the organization or to the specific problem to be addressed by the system.

Cont'd...

Consistency and Completeness Checking

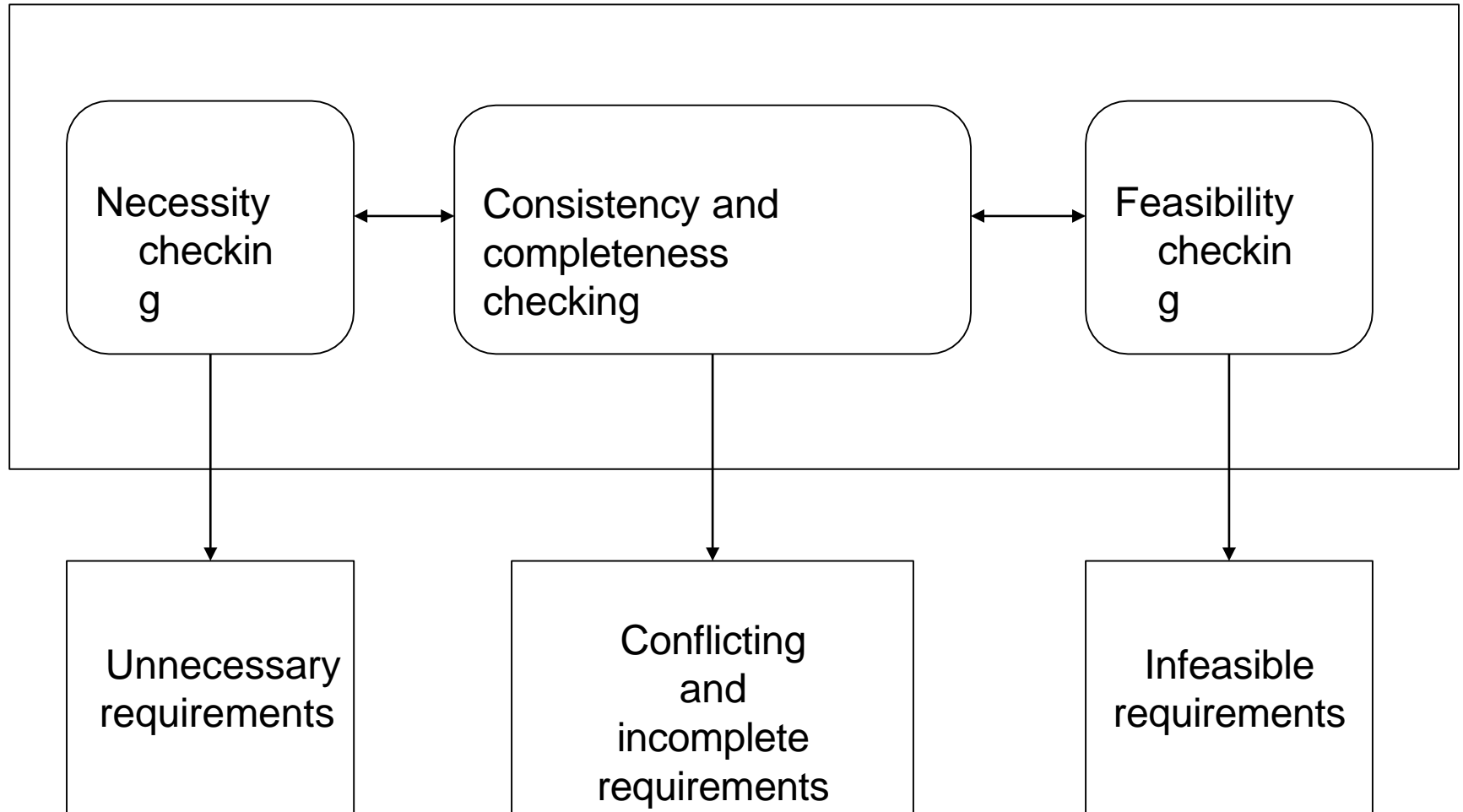
- The requirements are cross-checked for consistency and completeness.
- **Consistency** means that **no requirements should be contradictory**
- **Completeness** means that **no services** or constraints which are needed have been **missed out**.

Feasibility Checking

- The requirements are checked to ensure that they are feasible in the context of the **budget** and **schedule** available for the system development.

Requirements Analysis Process

Requirements Analysis



Analysis Techniques

- **Analysis checklists**
 - A checklist is a list of questions which analysts may use to assess each requirement.
- **Interaction matrices**
 - Interaction matrices are used to **discover** interactions between requirements and to highlight **conflicts** and **overlaps**.

Cont'd...

Analysis Checklists

- Each requirement may be assessed against the checklist
- When potential problems are discovered, these should be noted carefully
- They can be implemented as a spreadsheet, where the **rows** are labeled with the **requirements identifiers** and **columns** are the **checklist items**
- They are useful as they provide a reminder of what to look for and reduce the chances that you will forget some requirements checks.

Cont'd...

- They must evolve with the experience of the requirements analysis process.
- ❑ The questions should be general, rather than restrictive, which can be irrelevant for most systems.
- ✓ Checklists should not include more than ten items, because people forget items on long checklists reading through a document.
- ❖ The following list shows a possible requirement analysis checklist:

Checklist Items and their Description

- **Premature design**
 - Does the requirement include premature design or implementation information?
- **Combined requirements**
 - Does the description of a requirement describe a single requirement or could it be broken down into several different requirements?
- **Unnecessary requirements**
 - Is the requirement ‘gold plating’? That is, is the requirement a cosmetic addition to the system which is not really necessary?

Cont'd...

■ **Use of non-standard hardware**

- Does the requirement mean that non-standard hardware or software must be used?
 - To make this decision, you need to know the computer platform requirements.

■ **Conformance with business goals**

- Is the requirement consistent with the business goals defined in the introduction to the requirements document?

Cont'd...

■ **Requirements ambiguity**

- Is the requirement ambiguous i.e., could it be read in different ways by different people? What are the possible interpretations of the requirement?

■ **Requirements realism**

- Is the requirement realistic given the technology which will be used to implement the system?

■ **Requirements testability**

- Is the requirement testable, that is, is it stated in such a way that **test engineers can derive a test which can show if the system meets that requirement?**

Requirements Interactions

- A very important objective of requirements analysis is to discover the interactions between requirements and to **highlight requirements conflicts and overlaps.**
- A **requirements interaction matrix** shows how requirements interact with each other, which can be constructed using a spreadsheet.
 - Each requirement is compared with other requirements, and the matrix is filled as follows:
 - For requirements which **conflict**, fill in a **1**
 - For requirements which **overlap**, fill in a **1000**
 - For requirements which are **independent**, fill in a **0**
- **Consider the following example**

An Interaction Matrix

Requirement	R1	R2	R3	R4	R5	R6
R1	0	0	1000	0	1	1
R2	0	0	0	0	0	0
R3	1000	0	0	1000	0	1000
R4	0	0	1000	0	1	1
R5	1	0	0	1	0	0
R6	1	0	1000	1	0	0

Comments on Interaction Matrices

- If you can't decide whether requirements conflict, you should assume that a conflict exists.
- If an error is made it is usually fairly cheap to fix; it can be much more expensive to resolve undetected conflicts.
- In the example, we are considering, we can see that R1 overlaps with R3 and conflicts with R5 and R6.
- R2 is an independent requirement.
- R3 overlaps with R1, R4, and R6.
- The advantage of using numeric values for conflicts and overlaps is that you can sum each row and column to find the number of conflicts and the number of overlaps.
- ❖ Requirements which have high values for one or both of these figures should be carefully examined.

Cont'd...

- A large number of conflicts or overlaps means that any changes to that requirement will probably have a major impact on the rest of the requirements
- ❖ Interaction matrices work only when there is relatively small number of requirements, as each requirement is compared with every other requirement
 - The upper limit should be about **200** requirements.
- These overlaps and conflicts have to be discussed and resolved during **requirements negotiation**, which we'll discuss next.

Requirements Negotiation

- Disagreements about requirements are inevitable when a system has many stakeholders.
- **Conflicts** are not ‘failures’ but reflect **different stakeholder needs and priorities**.
- **Requirements negotiation** is the process of **discussing requirements conflicts** and **reaching a compromise that all stakeholders can agree to**.
 - In planning a requirements engineering process, it is important to leave enough time for negotiation.
 - Finding an acceptable compromise can be time-consuming.
- The final requirements will always be a compromise which is governed by the needs of the organization in general, the specific requirements of different stakeholders, design and implementation constraints, and the budget and schedule for the system development.

Requirements Negotiation Stages

1. Requirements Discussion

- Requirements which have been highlighted as problematic are discussed and the stakeholders involved present their views about the requirements.

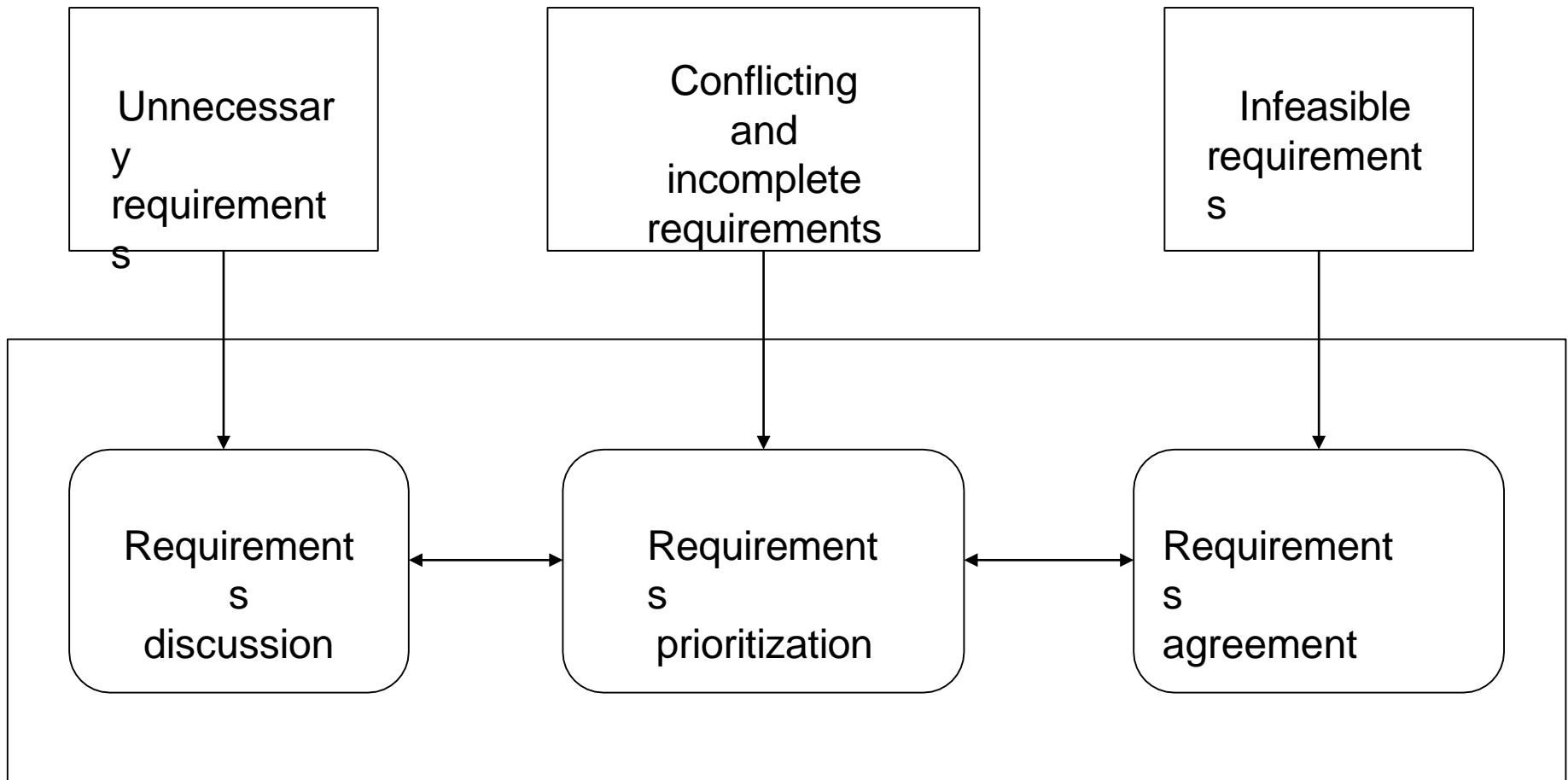
2. Requirements Prioritization

- Disputed/unclear requirements are prioritized to **identify critical requirements** and to help the decision making process.

3. Requirements Agreement

- Solutions to the requirements problems are identified and a compromised set of requirements are reached.
- Generally, this will **involve making changes to some of the requirements.**

Requirements Negotiation Process



Requirements
Negotiation

Comments on Requirements Negotiation

- ❑ In principle, requirements negotiation should be an objective process.
- The judgments should be the requirements for the system and should be based on technical and organizational needs.
- Reality is, however, often different.
- ❑ Negotiations are rarely conducted using only logical and technical arguments.
- They are influenced by organizational and political considerations, and the personalities of the people involved.
- A strong personality may force their priorities on other stakeholders.
- Requirements may be accepted or rejected because they strengthen the political influence in the organization of some stakeholders.
- End-users may be resistant to change and may block requirements,⁴¹ etc.

Cont'd...

- The majority of time in requirements negotiation is usually spent resolving requirements conflicts.
- A requirement conflicts with other requirements if they ask for different things.
- Conflicts should not be viewed as ‘failures’, they are natural and inevitable – rather healthy.
- **Meetings are the most effective way to negotiate requirements and resolve requirements conflicts.**
- All requirements which are in conflict should be discussed individually.
- ❑ **Negotiation meetings should be conducted in three stages.**

Stages of Negotiation Meetings

1. Information Stage

- The nature of the problems associated with a requirement is explained.

2. Discussion Stage

- A discussion stage where the stakeholders involved discuss how these problems might be resolved
 - All stakeholders with an interest in the requirement should be given the opportunity to comment.
 - Priorities may be assigned to requirements at this stage.

Cont'd...

3. Resolution Stage

- A resolution stage where actions concerning the requirement are agreed
 - These actions might be to delete the requirement, to suggest specific modifications to the requirement or to elicit further information about the requirement.

Thank You!
?