










Requirements Engineering

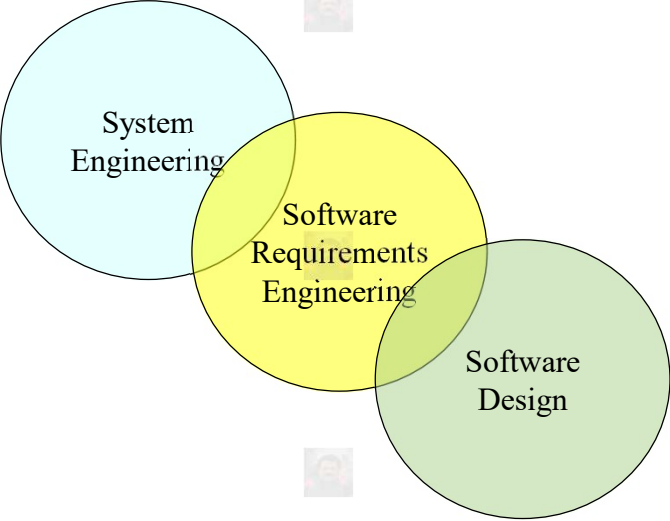
September 13, 2024

Nabendu Chaki, University of Calcutta

1



RE to Bridge the Gap



September 13, 2024

Nabendu Chaki, University of Calcutta

2

What is Requirements Engineering?



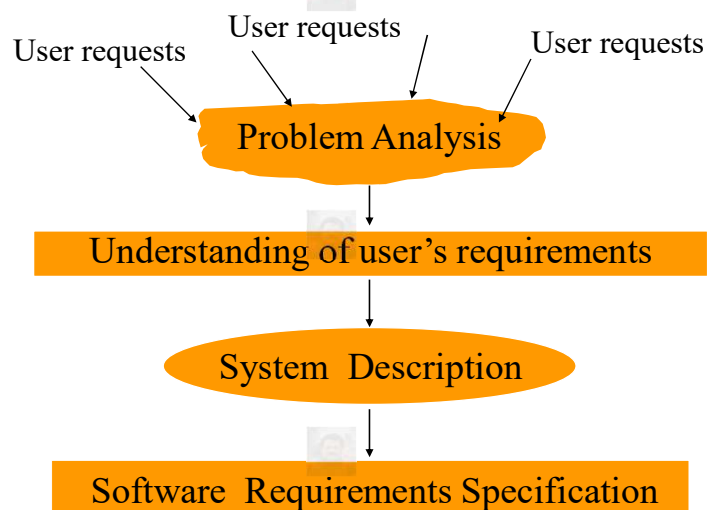
- A study of user needs for a problem to arrive at a definition of WHAT the software will do without describing how it will do it.
- A Software Requirements Specification (SRS) is a document containing *functional* and *non-functional* requirements for a system.

September 13, 2024

Nabendu Chaki, University of Calcutta

3

What is Requirements Engineering?



September 13, 2024

Nabendu Chaki, University of Calcutta

4

Types of Requirements



- **Functional requirements** define part of the system's functionality.
- **Non-Functional Requirements**
 - Implementation requirements which state how the system is to be implemented.
 - Performance requirements which specify a minimum acceptable performance for the system.
 - Constraints, e.g., the maximum acceptable time to deliver the system.

Challenges



- The requirements agreed may not actually reflect the real needs of the customer for the system.
- It is important to understand and unearth the hidden agendas – that's the real treasure box for a good system analyst.
- Requirements could be inconsistent and/or incomplete.

Challenges



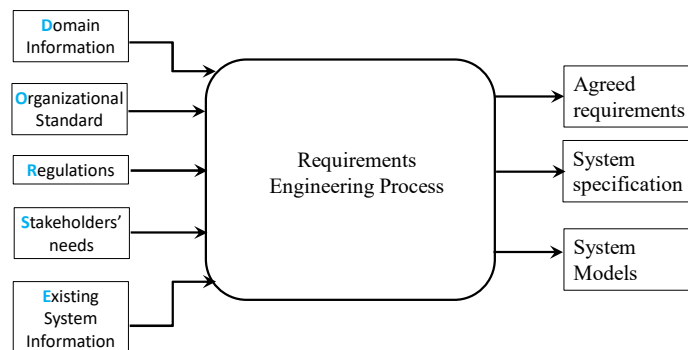
- It is expensive to make changes to requirements at a later stage of development.
- Often there are misunderstandings between customers, those developing the system requirements and software engineers developing or maintaining the system.

September 13, 2024

Nabendu Chaki, University of Calcutta

7

RE process - inputs and outputs



September 13, 2024

Nabendu Chaki, University of Calcutta

8

Requirements Engineering Activities



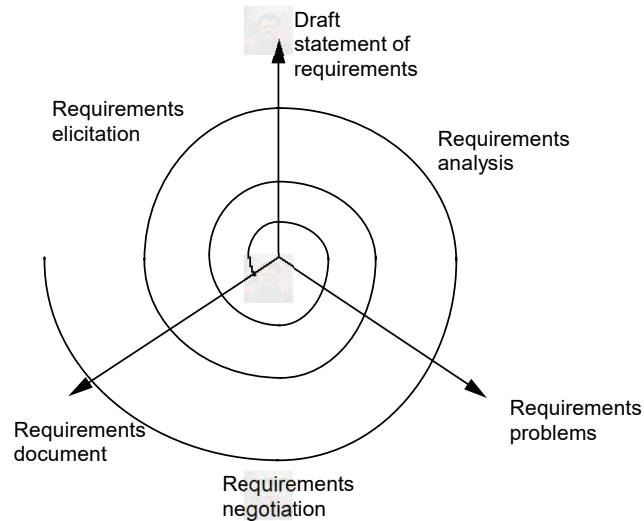
- Requirements *elicitation*
 - Requirements are discovered through consultation with stakeholders
- Requirements *analysis*
 - Requirements are analyzed to check for completeness, consistencies, correctness, and other quality aspects
- Requirements *negotiation*
 - Conflicts are resolved through requirements negotiation
- ...

September 13, 2024

Nabendu Chaki, University of Calcutta

9

Elicitation, analysis and negotiation



September 13, 2024

Nabendu Chaki, University of Calcutta

10

Requirements Engineering Activities



- Requirements *documentation*
 - A requirements document is produced. This is popularly referred as Software Requirement Specification (*SRS*).
- Requirements *validation*
 - The requirements document is checked for consistency and completeness
- Requirements *management*
 - The process of managing change to the requirements for a system

September 13, 2024

Nabendu Chaki, University of Calcutta

11

Requirements Elicitation

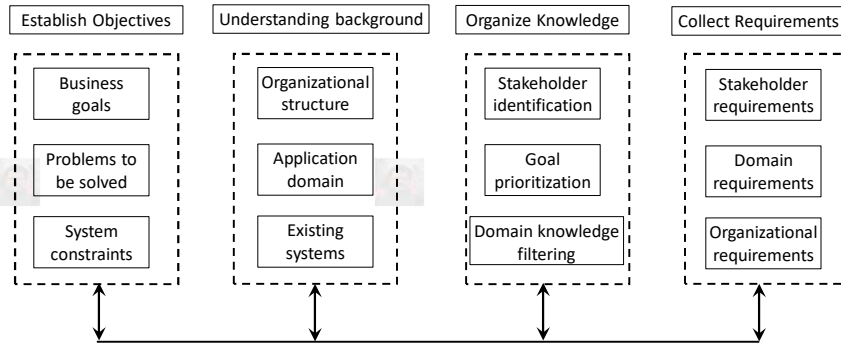


September 13, 2024

Nabendu Chaki, University of Calcutta

12

Requirements Elicitation Process

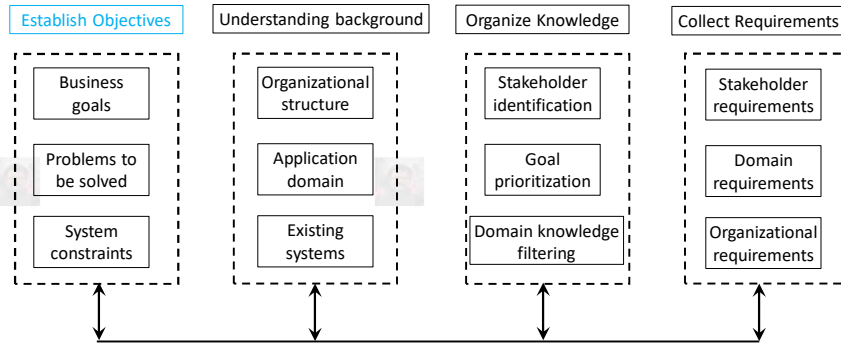


September 13, 2024

Nabendu Chaki, University of Calcutta

13

Requirements Elicitation Process

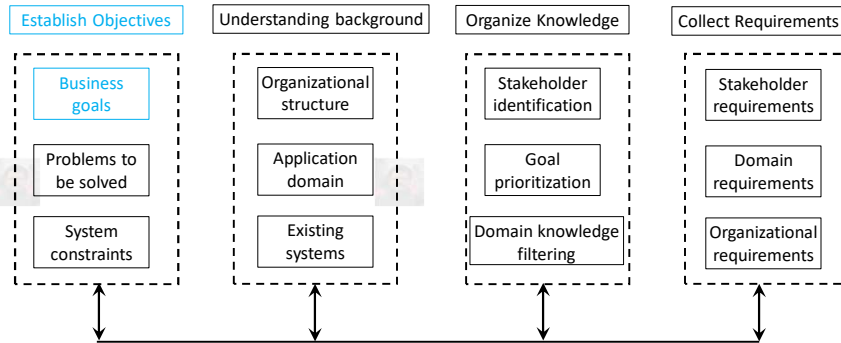


September 13, 2024

Nabendu Chaki, University of Calcutta

14

Requirements Elicitation Process

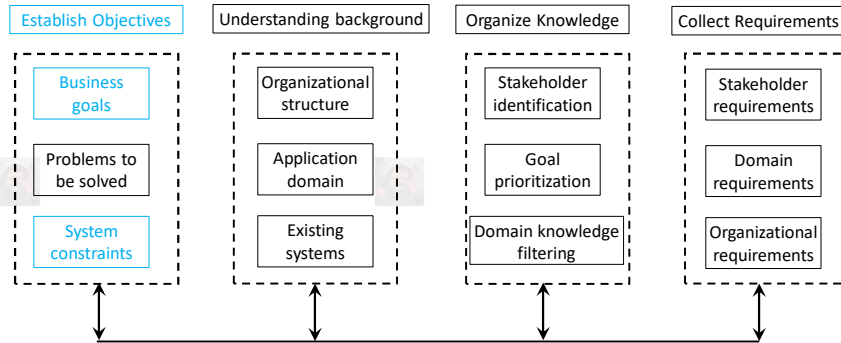


September 13, 2024

Nabendu Chaki, University of Calcutta

15

Requirements Elicitation Process

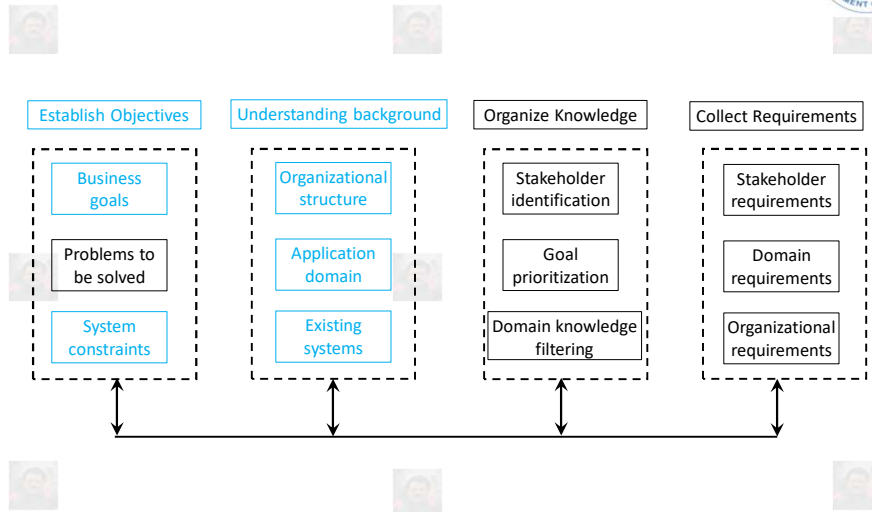


September 13, 2024

Nabendu Chaki, University of Calcutta

16

Requirements Elicitation Process

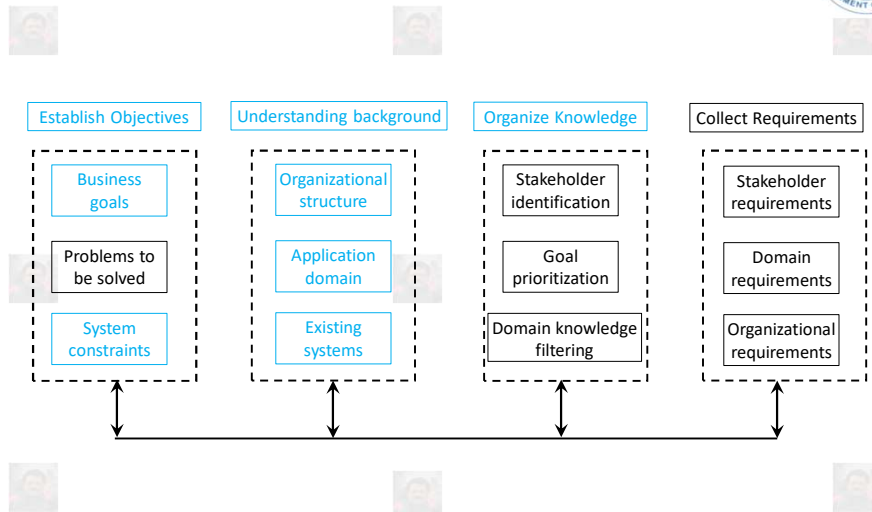


September 13, 2024

Nabendu Chaki, University of Calcutta

17

Requirements Elicitation Process

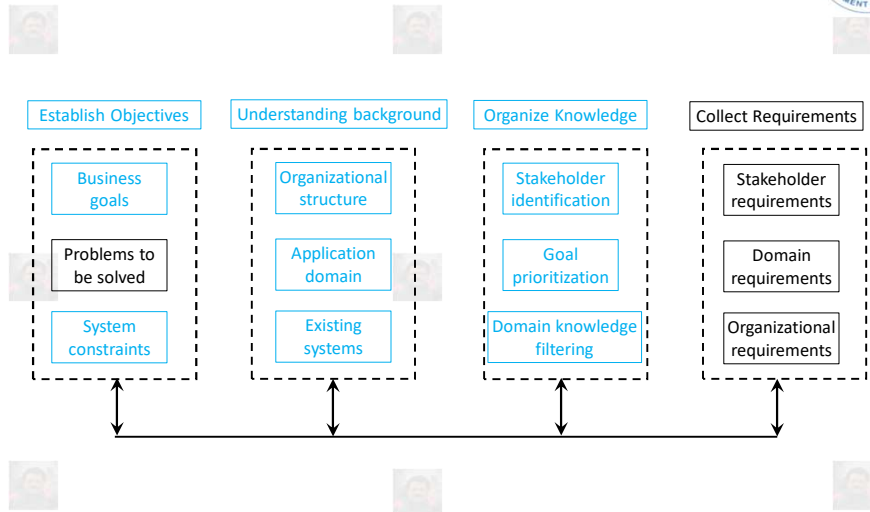


September 13, 2024

Nabendu Chaki, University of Calcutta

18

Requirements Elicitation Process

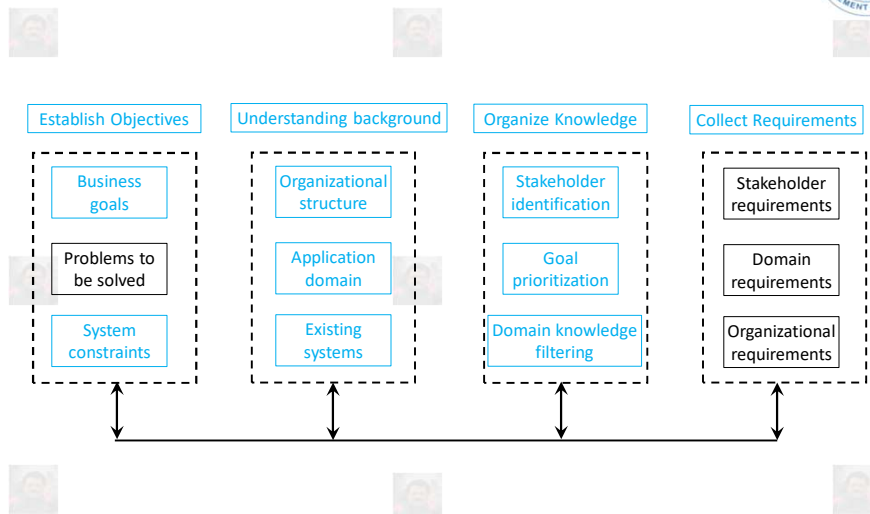


September 13, 2024

Nabendu Chaki, University of Calcutta

19

Requirements Elicitation Process

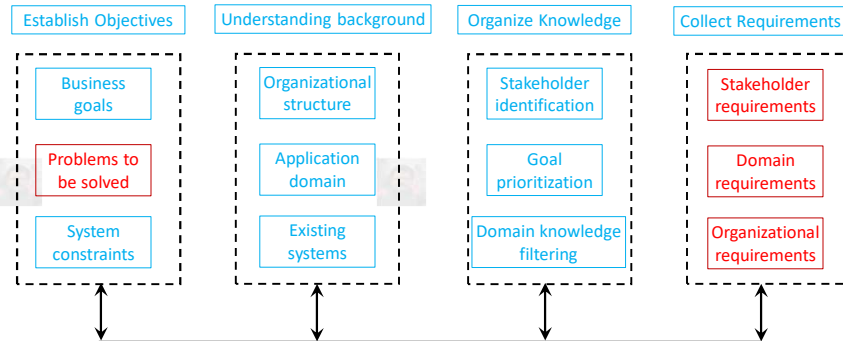


September 13, 2024

Nabendu Chaki, University of Calcutta

20

Requirements Elicitation Process



September 13, 2024

Nabendu Chaki, University of Calcutta

21

Elicitation Stages



- Objectives
 - The organisational objectives should be established including general goals of the business, an outline description of the problem to be solved, necessity of the system and the constraints on the system.

September 13, 2024

Nabendu Chaki, University of Calcutta

22

Elicitation Stages



- Background information about the system includes information about
 - the organisation where the system is to be installed,
 - the application domain of the system and
 - information about existing systems

September 13, 2024

Nabendu Chaki, University of Calcutta

23

Elicitation Stages



- Knowledge organisation
 - The large amount of knowledge which has been collected in the previous stage must be organised and collated.
 - Stakeholders are identified
 - Goals are prioritized

September 13, 2024

Nabendu Chaki, University of Calcutta

24

Elicitation Stages



- Requirements collection
 - System stakeholders are consulted to discover their requirements.
 - Requirements that comply to the application domain are listed
 - Organization specific requirements are listed

September 13, 2024

Nabendu Chaki, University of Calcutta

25

Elicitation Problems



- *Not enough time* for elicitation
- *Inadequate preparation* by engineers
- *Stakeholders are not convinced* about the need for a new system

September 13, 2024

Nabendu Chaki, University of Calcutta

26

Elicitation Techniques



- Interviews
- Scenarios
- Ethnography
- Requirements reuse
- Prototyping

September 13, 2024

Nabendu Chaki, University of Calcutta

27

Interviews



- Requirements engineer or analyst interviews different stakeholders and builds up an understanding of their requirements.
- Types of interview
 - **Closed** interviews: Pre-defined set of questions
 - **Open** interviews: Open-ended discussion with the stakeholders to know what they want from the system

September 13, 2024

Nabendu Chaki, University of Calcutta

28

Interviewing Essentials



- Interviewers must be open-minded and should **not** approach the interview with **pre-conceived notions** about what is required
- **Stakeholders must be given a starting point for discussion.** This can be a question, a requirements proposal or an existing system
- Interviewers must be **aware of organisational politics** - many real requirements may not be discussed because of their political implications

September 13, 2024

Nabendu Chaki, University of Calcutta

29

Scenarios



- **Scenarios are stories that explain how a system might be used.** They should include
 - a **description** of the system state **before** entering the scenario
 - the **normal flow** of events in the scenario
 - **exceptions** to the normal flow of events
 - information about **concurrent activities**
 - a **description** of the system state **at the end** of the scenario

September 13, 2024

Nabendu Chaki, University of Calcutta

30

Scenarios



- Scenarios are examples of **interaction sessions** which describe how a user interacts with a system
- **Discovering scenarios** exposes possible system interactions and reveals system facilities which may be required

September 13, 2024

Nabendu Chaki, University of Calcutta

31

Scenarios



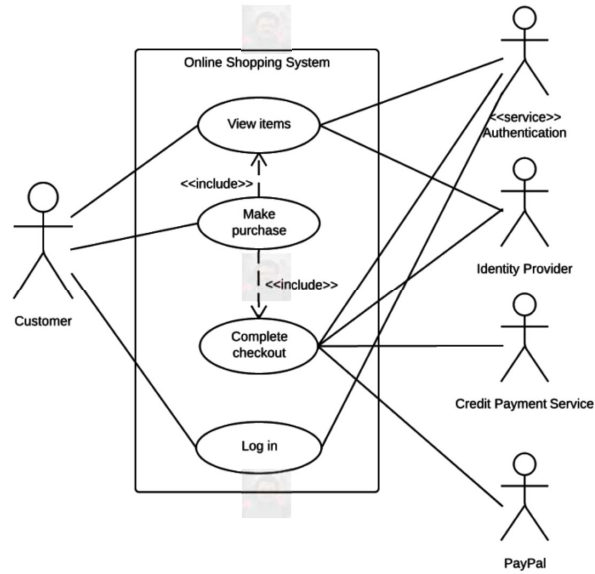
- Scenarios are **real-life examples** of how a system can be used.
- **Based on a practical situation** - stakeholders can **relate to those** and react.
- **Scenarios are a structured form of user story** with
 - A description of the **starting situation**;
 - A description of the **normal flow of events**;
 - A description of **what can go wrong**;
 - Information about other **concurrent activities**;
 - A description of the state **when the scenario finishes**.

September 13, 2024

Nabendu Chaki, University of Calcutta

32

Use Case Diagram

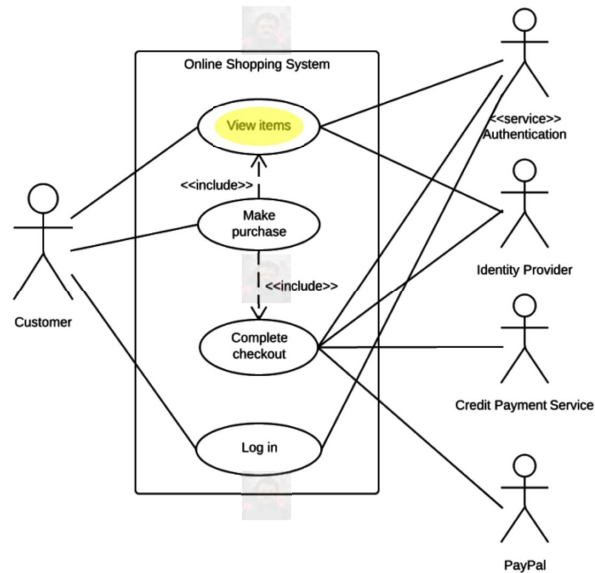


September 13, 2024

Nabendu Chaki, University of Calcutta

33

Use Case Diagram

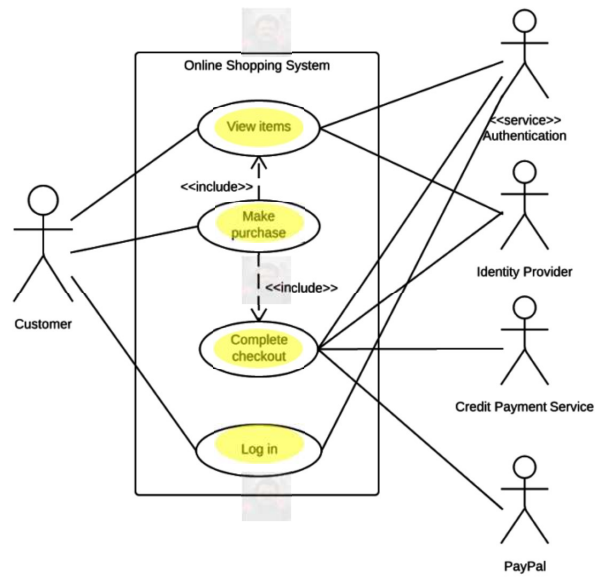


September 13, 2024

Nabendu Chaki, University of Calcutta

34

Use Case Diagram



September 13, 2024

Nabendu Chaki, University of Calcutta

35

Use Case Diagram

- Use cases **identify the actors in an interaction** and which describe the interaction itself.
- A set of use cases should describe all **possible interactions** with the **system**.
- High-level **graphical model** supplemented by more detailed tabular description .
- UML **sequence diagrams** may be used **to add detail** to use-cases by showing the sequence of event processing in the system.

September 13, 2024

Nabendu Chaki, University of Calcutta

36

Ethnography



- People often find it hard to describe what they do because it is so natural to them. *The best way to understand it is to observe them at work.*
- An ethnographer observes people at work and build up a picture of how work is done.
- Ethnography is a technique from the social sciences that has proved to be valuable in understanding actual work processes
- Requirements that are derived from the way that people actually work rather than the way in which process definitions suggest that they ought to work.



September 13, 2024

Nabendu Chaki, University of Calcutta

37

Ethnography Guidelines



- Actual work processes often differ from formal, prescribed processes
- Assume that people are good at doing their job and look for non-standard ways of working
- Spend time getting to know the people and establish a trust relationship
- Keep detailed notes of all work practices. Analyse them and draw conclusions from them



September 13, 2024

Nabendu Chaki, University of Calcutta

38

Ethnography Guidelines



- Combine observation with open-ended interviewing
- Organise regular de-briefing session where the ethnographer talks with people outside the process
- Combine ethnography with other elicitation techniques

September 13, 2024

Nabendu Chaki, University of Calcutta

39

Requirements Reuse



- Reuse involves taking the requirements which have been developed for one system and using them in a different system
- Requirements reuse *saves time and effort* as reused requirements have already been analysed and validated in other systems
- *Systematic reuse* could lead to larger cost savings

September 13, 2024

Nabendu Chaki, University of Calcutta

40

Reuse Possibilities



- whether the requirement is *concerned with providing application domain* info.
- whether the requirement is *concerned with the style of information presentation*. Reuse leads to a consistency of style across applications.
- whether the requirement *reflects company policies* such as security policies

September 13, 2024

Nabendu Chaki, University of Calcutta

41

Prototyping



- A prototype is an initial version of a system which may be used for experimentation
- Rapid development of prototypes is essential so that they are available early in the elicitation process

September 13, 2024

Nabendu Chaki, University of Calcutta

42

Prototyping Benefits



- The prototype allows users to experiment and discover what they really need to support their work
- Establishes feasibility and usefulness before high development costs are incurred

September 13, 2024

Nabendu Chaki, University of Calcutta

43

Prototyping Benefits



- Essential for developing the 'look and feel' of a user interface
- Can be used for system testing and the development of documentation
- Forces a detailed study of the requirements which reveals inconsistencies and omissions

September 13, 2024

Nabendu Chaki, University of Calcutta

44

Types of Prototyping



- Throw-away prototyping
 - intended to help elicit and develop the system requirements.
 - The requirements which should be prototyped are those which cause most difficulties to customers and which are the hardest to understand. Requirements which are well-understood need not be implemented by the prototype.

September 13, 2024

Nabendu Chaki, University of Calcutta

45

Types of Prototyping



- Evolutionary prototyping
 - intended to deliver a workable system quickly to the customer.
 - requirements that should be supported by the initial versions of this prototype are those that are well-understood and that can deliver useful end-user functionality.
 - It is only after extensive use that poorly understood requirements should be implemented.

September 13, 2024

Nabendu Chaki, University of Calcutta

46

Prototyping Costs and Problems



- *Training costs* - prototype development may require the use of special purpose tools
- *Development costs* - depend on the type of prototype being developed
- *Incompleteness* - it may not be possible to prototype critical system requirements

September 13, 2024

Nabendu Chaki, University of Calcutta

47

Prototyping Costs and Problems



- Extended development *schedules* - developing a prototype may extend the schedule although the prototyping time may be recovered because rework is avoided.
- *Lack of formalism* – there could be a tendency to “do whatever you get from user” leading to a lack of quality.

September 13, 2024

Nabendu Chaki, University of Calcutta

48



Questions?

