UMARU MUSA YAR'ADUA UNIVERSITY, KATSINA

COMPUTING SCIENCE DEPARTMENT

INS 204: SYSTEM ANALYSIS AND DESIGN

LECTURE NOTE 2

LECTURE TWO

REQUIREMENTS GATHERING

Requirements Gathering: Understanding Stakeholder Needs

Requirements gathering is a critical step in the System Development Life Cycle (SDLC) that involves collecting information from stakeholders to understand their needs and expectations for a new system or enhancement to an existing system. The primary goal is to ensure that the final system meets the users' requirements and provides value to the organization.

Practical Examples of Requirements Gathering and Understanding Stakeholder Needs

Example 1: Developing a New Customer Relationship Management (CRM) System

Scenario: A company decides to develop a new CRM system to better manage customer interactions, sales processes, and marketing campaigns.

Steps in Requirements Gathering:

1. Identify Stakeholders:
   - *Stakeholders*: Sales team, marketing team, customer service representatives, IT department, senior management.
   - *Objective*: Ensure all relevant perspectives are considered.

2. Conduct Interviews:
   - *Sales Team Interview*:
     - Questions: What are the current challenges with the existing system? What features would make your job easier?
     - Insights: Sales team needs better tracking of customer interactions and integration with email.
   - *Marketing Team Interview*:
     - Questions: How do you currently manage campaigns? What reporting capabilities do you need?
     - Insights: Marketing requires advanced segmentation and real-time campaign performance tracking.

3. *Facilitate Focus Groups*:
   - Description: Organize sessions with representatives from different teams to discuss their needs collaboratively.
   - Outcome: Identify common needs, such as a unified customer view and easy data entry forms.
4. *Distribute Surveys*:
   - Description: Create surveys to gather broader input from all stakeholders.
   - Questions: Rate the importance of various features (e.g., mobile access, analytics, integration with social media).
   - Results: Quantitative data to prioritize features based on stakeholder preferences.
5. *Observation and Shadowing*:
   - Description: Observe stakeholders as they use the current system to identify pain points and inefficiencies.
   - Findings: Customer service reps spend too much time switching between systems, highlighting the need for integration.
6. *Document Analysis*:
   - Description: Review existing documentation, such as process manuals, reports, and system logs.
   - Outcome: Understand current workflows and data usage patterns.
7. *Workshops and Brainstorming Sessions*:
   - Description: Conduct collaborative workshops to generate ideas and solutions.
   - Outcome: Detailed feature lists and workflow diagrams.

Example Requirements Document:

- Functional Requirements:
  - Track customer interactions across multiple channels (email, phone, social media).
  - Provide a unified customer view accessible by sales, marketing, and customer service.
  - Enable advanced customer segmentation and targeted marketing campaigns.
- Non-Functional Requirements:

- System must be accessible via mobile devices.
- Must support integration with existing email and social media platforms.
- Require high availability and data security measures.

Example 2: Enhancing an E-commerce Platform

Scenario: An online retailer wants to enhance its e-commerce platform to improve user experience and increase sales.

Steps in Requirements Gathering:

1. *Identify Stakeholders:*
   - Stakeholders: Customers, product managers, IT team, customer support, warehouse staff.
   - Objective: Capture diverse viewpoints to ensure comprehensive requirements.
2. *Customer Feedback and Surveys:*
   - Description: Collect feedback from customers through surveys and reviews.
   - Questions: What features do you find most useful? What challenges do you face while shopping online?
   - Insights: Customers want faster checkout, personalized recommendations, and better mobile usability.
3. *Focus Groups with Customers:*
   - Description: Conduct focus groups to delve deeper into customer preferences and pain points.
   - Outcome: Customers express a need for a more intuitive navigation and improved search functionality.
4. *Interviews with Internal Stakeholders:*
   - Product Managers: Discuss feature priorities, competitive analysis, and strategic goals.
   - Customer Support: Identify common customer complaints and frequent support issues.
   - Warehouse Staff: Understand inventory management challenges and fulfillment process inefficiencies.

5. *Competitive Analysis:*
    - Description: Analyze competitors' platforms to identify strengths and potential improvements.
    - Outcome: Identify industry best practices and innovative features that could be incorporated.
6. *Usability Testing:*
    - Description: Conduct usability tests with users to observe how they interact with the current platform.
    - Findings: Users struggle with the checkout process and often abandon their carts.

Example Requirements Document:

- **Functional Requirements:**
    - Simplify the checkout process to reduce cart abandonment rates.
    - Implement personalized product recommendations based on user behavior.
    - Enhance search functionality with filters and auto-suggestions.
- **Non-Functional Requirements:**
    - Ensure the platform is optimized for mobile devices.
    - Improve page load times to enhance user experience.
    - Implement robust security measures to protect user data.

Requirement Elicitation Techniques: Interviews, Questionnaires, etc.

Requirement elicitation techniques are methods used by system analysts to gather information from stakeholders regarding their needs, preferences, and expectations for a system. These techniques are crucial for understanding the scope and requirements of a project and play a significant role in the success of system development efforts. Let's delve into these techniques, exploring their meanings, features, merits, demerits, and practical applications:

1. **Interviews:**
   *Meaning***:** Interviews involve direct interaction between the system analyst and stakeholders to gather information. This can be done one-on-one or in a group setting.

Features:

- Personal interaction allows for in-depth exploration of ideas and concerns.
- Flexibility to adapt questions based on the interviewee's responses.
- Opportunity to clarify ambiguous or incomplete information.

Merits:

- Rich qualitative data obtained through open-ended questions.
- Allows for building rapport and establishing trust with stakeholders.
- Provides insights into stakeholders' perspectives and priorities.

Demerits:

- Time-consuming, especially for large groups or multiple interviews.
- May be influenced by interviewer bias or misinterpretation.
- Potential for interviewees to provide socially desirable responses.

Practical Application: Used in various stages of system development, including initial requirements gathering, clarification of requirements, and validation of proposed solutions.

2. Questionnaires:

Meaning: Questionnaires are written sets of questions administered to stakeholders to gather information. They can be distributed electronically or on paper.

Features:

- Standardized format ensures consistency in data collection.
- Can be distributed to a large number of stakeholders simultaneously.
- Anonymity may encourage more honest responses, especially for sensitive topics.

Merits:

- Efficient for gathering data from a large and diverse group of stakeholders.
- Standardized responses facilitate quantitative analysis.
- Cost-effective compared to interviews in terms of time and resources.

Demerits:

- Limited depth of information compared to interviews.
- Lack of opportunity for clarification or follow-up questions.
- Low response rates or incomplete responses may affect data quality.

Practical Application: Suitable for collecting baseline data, obtaining feedback on proposed solutions, or conducting surveys to understand stakeholder preferences.

3. Workshops/Brainstorming Sessions:

Meaning: Workshops or brainstorming sessions involve bringing together stakeholders in a facilitated group setting to generate ideas, discuss requirements, and solve problems collaboratively.

Features:

- Encourages active participation and collaboration among stakeholders.
- Facilitator guides the discussion and ensures all voices are heard.
- Enables rapid idea generation and consensus building.

Merits:

- Harnesses collective intelligence and creativity of stakeholders.
- Fosters a sense of ownership and commitment to the project.
- Allows for immediate feedback and iteration on ideas.

Demerits:

- Requires careful planning and facilitation to manage diverse opinions and avoid conflicts.
- May be challenging to schedule and coordinate participation from all relevant stakeholders.
- Dominant personalities or groupthink may influence outcomes.

Practical Application: Effective for exploring complex issues, generating innovative solutions, and building consensus among stakeholders on project requirements and goals.

Requirements Analysis and Documentation

Requirements analysis and documentation is a crucial phase in the System

Development Life Cycle (SDLC) where system analysts gather, analyze, and document the needs and expectations of stakeholders regarding a system or software project. This phase aims to ensure a clear understanding of the desired system functionalities, features, and constraints before proceeding with system design and development. Let's delve into this process in detail:

1. **Requirements Analysis:**

   Definition: Requirements analysis involves systematically studying stakeholder needs, expectations, and constraints to identify, clarify, and prioritize system requirements.

   Process:

   1. Requirements Elicitation: Utilize various techniques like interviews, questionnaires, and workshops to gather information from stakeholders.
   2. Requirements Documentation: Organize and document gathered requirements in a structured format, ensuring clarity and completeness.
   3. Requirements Validation: Verify and validate requirements with stakeholders to ensure accuracy and alignment with business objectives.
   4. Requirements Prioritization: Prioritize requirements based on their importance, feasibility, and impact on project success.
   5. Requirements Traceability: Establish traceability links between requirements and other SDLC artifacts to ensure comprehensive coverage and manage changes effectively.

   ***Example*:** Consider a project to develop an online banking system. During requirements analysis, system analysts gather information from various stakeholders, including bank employees, customers, and IT staff. They identify key functionalities such as account management, fund transfer, and bill payment. Through interviews and workshops, analysts clarify requirements related to security, user roles, and reporting. Once requirements are gathered, they are documented and validated with stakeholders to ensure accuracy and alignment with business needs.

2. **Requirements Documentation:**

   Definition: Requirements documentation involves capturing and organizing gathered requirements in a formal document that serves as a reference for system design and development activities.

Features:

- Clear and Concise: Requirements should be articulated in a clear and concise manner, avoiding ambiguity or ambiguity.
- Structured Format: Organize requirements systematically, typically including sections such as functional requirements, non-functional requirements, and constraints.
- Traceability: Establish traceability links between requirements and other SDLC artifacts, such as design documents and test cases, to ensure alignment and manage changes.
- Version Control: Maintain version control of requirements documents to track changes and facilitate collaboration among project stakeholders.

***Example*:** Continuing with the online banking system example, requirements documentation would include detailed descriptions of functional requirements such as user authentication, account balance inquiry, and transaction history retrieval. Non-functional requirements related to performance, security, and usability would also be documented. Each requirement is assigned a unique identifier, and traceability links are established with corresponding design documents and test cases.

Importance of Requirements Analysis and Documentation:

1. Alignment with Stakeholder Needs: Ensures that the developed system meets the expectations and requirements of stakeholders.
2. Clarity and Consistency: Provides a clear and consistent understanding of system functionalities, reducing ambiguity and misunderstandings.
3. Basis for Design and Development: Serves as a foundation for system design, development, and testing activities, guiding the implementation process.
4. Risk Management: Helps identify and address potential risks and challenges early in the project lifecycle, minimizing rework and cost overruns.
5. Change Management: Facilitates effective change management by providing a baseline for assessing the impact of proposed changes on project scope and objectives.

Recommended Text books:

- Tilley, S., & Rosenblatt, H. J. (2016). *Systems analysis and design* (11th Ed.). Cengage Learning.
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). *Systems analysis and design in a changing world* (7th Ed.). Cengage Learning.
- Hoffer, J. A., George, J. F., & Valacich, J. S. (2016). *Modern systems analysis and design* (8th Ed.). Pearson.
- Dennis, A., Wixom, B. H., & Roth, R. M. (2018). *Systems analysis and design* (7th Ed.). Wiley.