CS 487 – SOFTWARE ENGINEERING

Week 3 Engagement - Awareness of User Needs

Suhas Palani **A20548277** spalani3@hawk.iit.edu

Requirements engineering is a key phase in software development where we figure out what users need and what limitations we might face. It starts with a broad understanding and narrows down to detailed specifications, essentially creating an agreement between everyone involved—both the stakeholders and the developers.

Types of Requirements:
- **Functional:** Define system behaviors and responses to inputs and specific situations
- **Non-functional:** Impose constraints on system services or functions, often applying to the system as a whole
- **Domain:** Derived from the specific operational domain of the system

Non-functional Requirements Categories:
1. **Product:** Execution speed, memory requirements, usability, portability, failure rate
2. **Organizational:** Policies, process standards, delivery timeframes, programming languages, methodologies
3. **External:** Interoperability with other systems, legal and ethical requirements

Challenges in Capturing Requirements:
- **Ambiguity:** Balancing clarity and brevity, considering differences in human, user, and system languages
- **Confusion:** Distinguishing between requirement types, system goals, and design information
- **Amalgamation:** Multiple requirements contained in a single statement

System Requirements Challenges:
- **Structured Natural Language:** Standard forms and templates
- **Design Description Languages:** Like pseudo-code
- **Graphical Notations:** Use-case and sequence diagrams
- **Mathematical Specifications:** Based on concepts like finite-state machines

Requirements Engineering Process:
1. **Feasibility Study:** Figure out if building the system is a good idea and if it can be successfully integrated and maintained.
2. **Requirements Elicitation and Analysis:** Collaborate with stakeholders to uncover what's needed, while dealing with issues like vague requirements and gaps in terminology.
3. **Requirements Specification:** Clearly and precisely state what is needed.
4. **Requirements Validation:** Verify that the requirements satisfy the needs of the client and that they are testable, valid, consistent, full, and reasonable.

Ethnography in Requirements Gathering:

Ethnography was highlighted as an effective observational technique for understanding social and organizational requirements. It helps uncover requirements based on actual work practices rather than theoretical ones, emphasizing cooperation and awareness among stakeholders.

H-H-I (Human-Human-Interface) Notations:

- **Human-Human Interaction**: Focuses on effective communication between users and engineers to translate needs into technical requirements
- **Interface Specification:** Defines component interactions, including:
- **Procedural Interfaces (APIs):** Specify available procedures or functions
- **Data Structures:** Define data organization and access
- **Data Representations:** Establish data formatting and communication, especially for existing subsystems

Interface Specification:

- **Procedural Interfaces:** APIs for component interaction, ensuring adherence to protocols and data formats
- **Data Structures:** Organize and manage system data, ensuring consistency and efficiency
- **Data Representations:** Define data formatting and transfer, maintaining integrity and facilitating interoperability

The requirement engineering process is all about putting together and keeping up to date a complete document of what the system needs to do. It stresses the importance of clear communication, detailed interface specifications, and thorough validation checks. By concentrating on these areas, developers can better understand what users need and make sure the system fulfils both its functional and non-functional requirements effectively.