Software Requirements Engineering

Week 4

Asst. Prof. Bilge Kağan DEDETÜRK

The process support and management

- The requirements processes interact with all the other processes within the project life cycle, and we should put that into consideration while working on our requirements.

- Support processes ensure the operation of the company's central processes and overall operations of the company.

Support Processes

- Human Resource Management
  - Managing the people involved in the requirements process, their hiring, training, and so on.
- Financial Management processes
  - The budget allocated to the requirements process.
- IT processes
  - The infrastructure needed by the requirements process, such as the version control tools.

Support Processes

- Procurement and sourcing processes
  - Supplier selection processes, purchasing services, purchasing materials, tools, sourcing human and financial resources, etc.
- Operational processes
  - Routine operation, and organization management
- Risk Management processes
- Quality Management processes

The process support and management

- Support processes are usually more complex and complicated in larger companies

- In small companies, only one person can be responsible for a particular support process or activity.

Process quality and improvement

- A good quality requirement process and excellent software quality will allow you to deliver products with cost-effectiveness, on time, and high customer satisfaction.

- Any process needs continuous improvement.

- Our processes should always be adapted to changes.

Process quality and improvement

- Improvements can seek "incremental" improvement over time or "breakthrough" improvement all at once.

- You should first assess the current requirements process being utilized and recognize its qualities and deficiencies.

- The assessment doesn’t give you any improvement. It gives you information that you have to analyze to put your hands on the real problem.

Requirements process metrics

Requirements process metrics include:

- Number of requirements change
- Volatility
- Ambiguity
- Defects escaped
- Cost-effectiveness

Process Improvement Models: PDCA

- Plan: Identify an opportunity and plan for change.
- Do: Implement the change on a small scale.
- Check: Use data to analyze the results of the change and determine whether it made a difference.
- Act: If the change was successful, implement it on a broader scale and continuously assess your results.

Process quality and improvement

- The cost to change a system at different stages will vary depending on:
  - lifecycle model adopted
  - size and complexity of the system
  - Adherence to guidelines and standards

Process quality and improvement

- Other widely used continuous improvement methods:

  Six Sigma,

  Lean,

  Total Quality Management.

- Those models emphasize:
  - employee involvement and teamwork
  - measuring and systematizing processes
  - reducing variation, defects, and cycle times.

Requirements Elicitation

- Requirements elicitation is concerned with the origin of software requirements and how the software engineer can discover them.

- Even though the most famous term for this activity is “requirements gathering”.
  - Gathering means that the requirements already exist and are ready for collection and documentation.

Requirements Elicitation

- According to the business dictionary, “Elicitation” is:
  - drawing out,
  - calling forth,
  - evoking a response or reaction

- It is also termed:
  - requirements capture,
  - requirements discovery
  - requirements acquisition

Requirements Elicitation

- Requirements elicitation is the first stage in understanding the problem and the software is required to solve.

- It is also about establishing relationships between the development team and the customer.

Requirements Elicitation

- Analysts must mediate between the world of the software users, and other stakeholders, and the software engineer's technical world.

- A critical element of requirements elicitation is forming the project scope.
  - This involves describing the software being specified, and its purpose, and prioritizing the deliverables.

Issues and Problems in Requirements Elicitation

- Your client may not know exactly what the problem is.
- We should recognize and appreciate the users as domain experts; that’s why we should always try different elicitation techniques..

Issues and Problems in Requirements Elicitation

- Imagine working with a client who knows nothing about computers or software systems; it will be difficult for him to visualize his needs. There will be a lot of wishful thinking and impractical expectations from the stakeholders leading to non-realizable requirements.

- If your client is an expert in his field, he might skip some information thinking that it is obvious enough; “It goes without saying!”.

Issues and Problems in Requirements Elicitation

- “Language Diversity.” There is a communication gap between analysts and stakeholders.

- Culture also can make a problem.

Issues and Problems in Requirements Elicitation

- The user tells you the solution to his problems, not the problem itself.
- Telling you the solution and not knowing the root problem might restrict you from designing the best solution to the real problem.

Issues and Problems in Requirements Elicitation

- Customer get bored from the meeting and the questions.
- Select the wrong stakeholders and maybe fail to identify the correct stakeholders.
- Sometimes, an influential stakeholder will insist on being in the picture and being the only source of requirements even though he might not know all the information needed.

Issues and Problems in Requirements Elicitation

- If the assigned analyst to the project is not experienced or trained enough, he/she may not capture all the information provided accurately and miss important ones.

The “Undiscovered Ruins” Syndrome

- The search for requirements is like a search for undiscovered ruins: the more you find, the more you know remains. **You will never know when to stop looking.** How to cope with this syndrome:
  - Identify all the stakeholders of the system.
  - Taking the time to define the problem to be solved with the system
  - Employ techniques that help find some of those ruins
  - Have the stakeholders buy-into the requirements

The “Yes, But” Syndrome

- Whenever the users see the system, or even a small part of the system for the first time, their reaction could be, “Yes, but, …, now that I see it, what about this…? Wouldn’t it be nice …?

- Accepting the "Yes, But" syndrome as reality may lead to real insights that will help team members mitigate this syndrome in future projects.

- Techniques like prototyping, modeling, role-playing and storyboards help get the idea physically to the users.

Requirements Sources

- Requirements have many sources in typical software, and all possible sources must be recognized and evaluated.

- Dropping any of the requirements' sources could cause harm to the project.

Requirements Sources

- According to the SWEBOK, there are six primary requirements sources:
  - Goals
  - Domain knowledge
  - Stakeholders
  - Business rules
  - Operational environment
  - Organizational environment

Requirements Sources: Goals

- The term “goal,” sometimes called "business concern" or "critical success factor," refers to the software's overall, high-level objectives.

Feasibility study:

- A feasibility study is a short, focused study to assess a proposed project or system's practicality.
- Aims to uncover the strengths and weaknesses of an existing business or proposed project, opportunities, and threats present in the natural environment.
- The two criteria to judge feasibility are cost required and value to be attained.
- Software engineers need to pay attention to estimate the value and cost of goals. The feasibility study is a relatively low-cost way of doing this.
- Goals should be our primary source of high-level requirements

Requirements Sources: Domain Knowledge

- Domain knowledge provides the background against which all elicited requirements knowledge must be set to understand it.

- With domain knowledge, one can tell whether a requirement has been missed or conflicts with other requirements.

Requirements Sources: Stakeholders

- One of the many reasons the software can be unsatisfactory is to stress the requirements from one group of stakeholders than the others.

- The software engineer should be careful while identifying, representing, and managing the different stakeholders' viewpoints.

Requirements Sources: Business Rules

- Every business includes rules that represent the guidelines and the constraints of the business.

- Business rules, unlike domain knowledge, are specific to the business.

Requirements Sources: Operational Environment

- Software's requirements could be extracted from the operational environment where the software will be executed.

- Requirements could be, for example, timing constraints in real-time software or performance constraints in a business environment.

Requirements Sources: Organizational Environment

- Every organization has its own structure, culture, and internal politics.

Requirements Sources: Technology

- New products are often developed to take advantage of technological developments since their competitors were launched.