Requirement Error Taxonomy

Loan arranger system (LA): The LA application supports the business of a loan consolidation organization. This type of organization makes money by purchasing loans from banks and then reselling those loans to other investors. The LA allows a loan analyst to select a bundle of loans that have been purchased by the organization that match the criteria provided by an investor. This criterion may include amount of risk, principal involved and expected rate of return. When an investor specifies investment criteria, the system selects the optimal bundle of loans that satisfy the criteria. The LA system automates information management activities, such as updating loan information provided monthly by banks.

Automated ambulance dispatch system (AAD): This system supports the computer-aided dispatch of ambulances to improve the utilization of ambulances and other resources. The system receives emergency calls, evaluates incidents, issues warnings, and recommends ambulance assignments. The system should reduce the response time for emergency incidents by dispatching decisions based on recommendations made by system.

PEOPLE ERRORS

- The communication errors class (Table 1) describes errors that result from poor or missing communications among the various stakeholders involved in developing the requirements.

Table 1 Communication Errors

Lack of communication among stakeholders including communication with customers/users within the team and between teams.
Unclear lines of communication and authority leading to lack of consensus on technical standards and approaches with teams.
Lack of communication of changes made to a document.

Example:
Error: Customer did not communicate that the LA system should be used by between one and four users (loan analysts) simultaneously.
Fault: Omitted functionality because the requirements specify operations as if they are performed by only one user at a time.

- The participation error class (Table 2) describes errors that result from inadequate or missing participation of important stakeholders involved in developing the requirements.

Table 2 Paticipation Errors

No involvement of all the stakeholders
Lack of involvement of users at all times during requirement development

Involving only selected users to define requirements due to the internal factors like rivalry among developers or lack of the motivation
Lack of mechanism to involve all the users and developers together to resolve the conflicting requirements needs

Example:
Error: Bank lender, who was not involved in the requirements process, wanted the LA application system to handle both fixed rate loans and adjustable rate loans.
Fault: Omitted functionality as requirements only consider fixed rate loans.

- The domain knowledge error class (Table 3) describes errors that occur when requirement authors lack knowledge or experience about the problem domain.

Table 3 Domain Knowledge Errors

Lack of domain knowledge or lack of system knowledge
Complexity of the problem domain
Lack of appropriate knowledge about the application
Complexity of the task leading to misunderstandings
Lack of adequate training or experience of the requirement engineer
Lack of knowledge, skills, or experience to perform a task
Some properties of the problem space are not fully investigated
Mistaken assumptions about the problem space

Example:
Error: Requirement author lacks knowledge about the relative priority of emergency types within the AAD domain.
Fault: The functionality is incorrect because the requirements contain the wrong ambulance dispatch algorithm.

- The specific application error class (Table 4) describes errors that occur when the requirement authors lack knowledge about specific aspects of the application being developed (as opposed to the general domain knowledge).

Table 4 Specific Application Errors

Lack of understanding of the particular aspects of the problem domain
Misunderstandings of hardware and software interface specification
Misunderstanding of the software interfaces with the rest of the system
User needs are not well understood or interpreted while resolving conflicting requirements
Mistakes in expression of the end state or output expected
Misunderstandings about the timing constraints, data dependency constraints, and event constraints
Misunderstandings among input, output and process mappings

Example:
Error: Requirement author does not understand the order in which status changes should be made to the ambulances in the AAD system.
Fault: The requirements specify an incorrect order of events, the status of the ambulance is updated after it is dispatched rather than before, leaving a small window of time in which the same ambulance could be dispatched two times.

- The process execution error class (Table 5) describes errors that occur when requirement authors make mistakes while executing the requirement elicitation and development processes regardless of the adequacy of the chosen process.

Table 5 Process Execution Errors

Mistakes in executing the action sequence or the requirement engineering process, regardless of its adequacy
Execution or storage errors, out of order sequence of steps and slips/lapses on the part of people executing the process

Example:
Error: Misunderstanding of the ordering of the transactions involving the creation or deletion of records in the database for the LA system.
Fault: The requirements incorrectly specify how the LA should handle transactions. They state that transactions should be resolved based on the order in which the processing completes rather than in the order in which the requests were received.

- The other human cognition error class (Table 6) describes other errors that result from constraints on the cognitive abilities of the requirement authors.

Table 6 Other Human Cognition Errors

Mistakes caused by adverse mental states, loss of situation awareness
Mistakes caused by ergonomics or environmental conditions
Constraints on humans as information processors e.g., task saturation

PROCESS ERRORS

- The inadequate method of achieving goals and objective error class (Table 7) describes errors that result from selecting inadequate or incorrect methods for achieving the stated goals and objectives.

Table 7 Inadequate Method of Achieving Goals and Objectives Errors

Incomplete knowledge leading to poor plan on achieving goals
Mistakes in setting goals
Error in choosing the wrong method or wrong action to achieve goals
Some system-specific information was misunderstood leading to the selection of wrong

method
Selection of a method that was successful on other projects
Inadequate setting of goals and objectives
Error in selecting a choice of a solution
Using an analogy to derive a sequence of actions from other similar situations resulting in the wrong choice of a sequence of actions
Transcription error, the developer understood everything but simply made a mistake

Example:
Error: The requirements engineering misunderstood that some crucial functionality had to be delivered before other functionality, so s/he chose the waterfall lifecycle rather than an incremental one.
Fault: Required functionality cannot be delivered on time to the customer.

- The management error class (Table 8) describes errors that result from inadequate or poor management processes.

Table 8 – Management Errors

Poor management of people and resources
Lack of management leadership and necessary motivation
Problems in assignment of resources to different tasks

Example:
Error: In the LA system, the same requirement engineer is assigned to document the borrower's risk requirement and the loan's risk requirements. These contrasting tasks result in a mental lapse when understanding the inputs required to successfully produce the risk estimates.
Fault: The functionality for calculating risk is incorrect.

- The requirement elicitation error class (Table 9) describes errors that result from the use of an inadequate requirement elicitation process.

Example:
Error: In the AAD system, the requirements engineers are not able to elicit requirements about system response time for emergency incidents or about error handling.
Fault: Performance and other non-functional requirements are omitted.

Table 9 Requirement Elicitation Errors

Inadequate requirement gathering process
Only relying on selected users to accurately define all the requirements
Lack of awareness of all the sources of requirements
Lack of proper methods for collecting requirements

- The requirement analysis error class (Table 10) describes errors committed during the requirement analysis process.

Example:
Error: In the LA system, the analysis process was not able to identify how the system should respond if multiple transactions are requested that include the same loan before the system is updated.
Fault: The requirements omit this situation leaving it undefined and possibly erroneous.

Table 10 Requirement Analysis Errors

Incorrect model(s) while trying to construct and analyze solution
Mistakes in developing models for analyzing requirements
Problem while analyzing the individual pieces of the solution space
Misunderstanding of the feasibility and risks associated with requirements
Misuse or misunderstanding of problem solution processes
Unresolved issues and unanticipated dependencies in solution space
Inability to consider all cases to document exact behavior of the system
Mistakes while analyzing requirement use cases or scenarios

- The requirement traceability error class (Table 11) describes that result from an inadequate or incomplete requirement traceability process.

Example:
Error: In the LA system, a requirement describing the ability of loan analyst to change the borrower information can not be traced to any user need.
Fault: An extraneous requirement is included that could result in extra, unnecessary work for the developers.

Table 11 Requirement Traceability Errors

Inadequate/poor requirement traceability
Inadequate change management, including impact analysis of changing requirements

DOCUMENTATION ERRORS

- The requirement organization error class (Table 12) describes errors committed while organizing the requirement during the documentation process.

Table 12 Requirement Organization Errors

Poor organization of requirements
Lapses in organizing requirements
Ineffective method for organizing together the requirements documented by different developers

Example:
Error: When creating the requirements document for the LA system, the requirement engineer does not use any type of logical organization of the requirements.
Fault: Because the requirements are not grouped logically, a requirement about how to display the results of a report was omitted from the requirements document.

- The no use of documentation standard error class (Table 13) describes errors committed because the requirement author did not use a standard for documenting the requirements.

Example:
Error: In documenting the LA application system, the IEEE Standard was not used.
Fault: Requirements about system scope and performance were omitted.

Table 13 No Use of Standard for Documenting Errors

No use of standard format for documenting requirements
Different technical standard or notations used by sub teams for documenting requirements

- The specification error class (Table 14) describes general errors that can occur while specifying the requirements regardless of whether the developers correctly understood the requirements.

Table 14 Specification Errors

Missing checks (item exists but forgotten)
Carelessness while organizing or documenting requirement regardless of the effectiveness of the method used
Human nature (mistakes or omissions) while documenting requirements
Omission of necessary verification checks or repetition of verification checks during the specification

Example:
Error: In the LA application system, the requirement author understood the difference between regular loans (i.e., for amount <= $275,000) and jumbo loans (i.e., for amount >$275,000), but while documenting the requirements, s/he recorded the same information for both type of loans.
Fault: The requirements for the jumbo loans incorrectly specify exactly the same behavior as for regular loans.