

Human Error Taxonomy

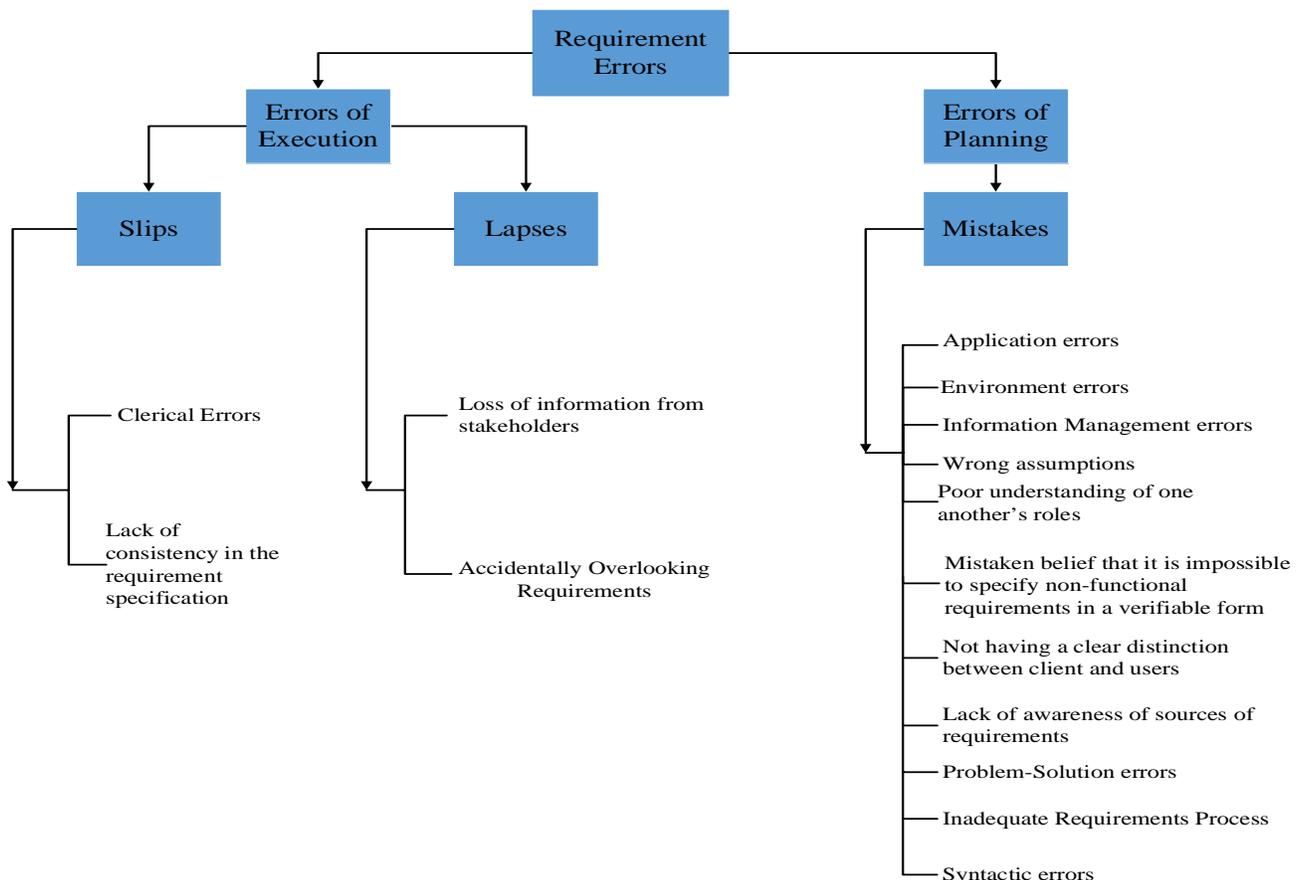
The Human Error Taxonomy (HET) provides a structure for requirement errors made during the software development process. The HET can be employed during software inspection to help *detect* and *classify* errors in requirement specifications. Errors organized in the HET have been identified through a systematic investigation of the software engineering literature. This process has allowed us to identify different types of errors that people make during software requirements development. Each error in HET represents a flaw in our thought process and can lead to one or more than one faults in the requirement specification.

The two primary categories of errors considered in HET are:

Errors of Execution: The failure of a planned action to be completed (executed) as intended.

Errors of Planning: The use of a wrong plan to achieve a goal (or aim).

In the figure below, notice that we divide execution errors and planning errors into three broad categories: *slips*, *lapses*, and *mistakes*. Slips occur when we intend to perform one action, but instead do another. You might intend to pour some more milk on your morning cereal but instead pour orange juice instead – a typical slip. Lapses are defined as forgetting one or more steps in a process. Perhaps you have had the experience of going off to another room to accomplish something, only to realize you have forgotten what you meant to do. That is a classic lapse. Mistakes are understood as a faulty plan. These errors will be explained in greater detail below.



HET defines human errors that can happen in the following Requirement Engineering Activities:

- ❖ Elicitation: Discovering and understanding the user's needs and constraints for the system.
- ❖ Analysis: Refining the user's needs and constraints.
- ❖ Specification: Documenting the user's needs and constraints clearly and precisely.
- ❖ Verification: Ensuring that the system requirements are complete, correct, consistent, and clear.
- ❖ Management: Scheduling, coordinating, and documenting the requirements engineering activities (that is, elicitation, analysis, specification, and verification).

To help you get a better understanding of the Human Error Taxonomy and its application in requirements inspection, we will be examining requirement specifications for a *Loan Arranger system*.

Loan arranger system (LA): A loan consolidation organization purchases loans from banks and bundles them for resale to other investors. The LA application allows a loan analyst to select a bundle of loans that match the criteria provided by an investor. These criteria may include: 1) amount of risk, 2) principal involved, and 3) 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.

SLIPS

A slip occurs when a person carries out a *planned task* incorrectly or in the wrong sequence. Hence, a slip is an execution failure. Slips are generally attentional failures (carrying out a task incorrectly while executing a planned sequence of events due to lack of attention). A slip can only occur when a plan has been made. HET identifies 2 errors under the Slips category, which are described in Table 1.

Table 1 Slips

Clerical errors

Error Description:

Clerical errors are due to carelessness while performing mechanical transcriptions from one format or from one medium to another. Requirement specific examples include carelessness while writing specifications from elicited user needs.

Example:

Error: In the LA application system, the requirement author understood the difference between regular loans (i.e., for amount \leq \$275,000) and jumbo loans (i.e., for amount $>$ \$275,000), but while documenting the requirements, s/he recorded the same information for both types of loans.

Fault: The requirements for the jumbo loans incorrectly specify exactly the same behavior as for regular loans.

Lack of consistency in the requirement specification

Error Description:

Such errors occur when requirement authors are not able to articulate and organize the requirements in a consistent manner, even when they have a clear idea of user needs. This leads to requirements being specified in a disjointed form which makes interpretation very difficult.

Example:

Error: When creating the requirements document for the LA system, the requirement engineer does not use any type of logical organization for the requirements.

Fault: Because the requirements are not grouped logically, a requirement about the number of loan analysts that can use the LA system at a given time is inconsistent with the information provided in the 'Preconditions and Assumptions' section of the document.

LAPSES

Lapses happen when a goal is forgotten in the middle of a sequence of actions, or we omit a step in a routine sequence. Like slips, lapses are also execution failures. Lapses are generally memory related failures (forgetting a step while executing a planned sequence of events). HET identifies 2 errors under the Lapses category, which are described in Table 2.

Table 2 Lapses

Loss of information from stakeholders

Error Description:

Forgetting, discarding or failing to store information or documents provided by stakeholders. For example, information about some important user need is forgotten before being recorded by the person gathering requirements.

Example:

Error: The team member responsible for gathering requirements receives information about the format of reports (file, screen, or printout) from a loan analyst, but forgets to note it down.

Fault: Information about the report formats the loan analysts can receive is omitted.

Accidentally overlooking requirements

Error Description:

Many requirements, both functional and non-functional, are accidentally overlooked. This typically happens because the stakeholders who are the source of the requirements often assume that such requirements are obvious and go without saying.

Example:

Error: Stakeholders assume that abnormal termination and system recovery is a commonplace occurrence and will be handled by requirement analysts or the system design team. Therefore, they do not provide system recovery requirements.

Fault: Description of system recovery from abnormal termination has been omitted

MISTAKES

Mistakes happen as a result of *inadequate planning*. Mistakes are failures that happen when the actions proceed as planned, but the plan itself is not adequate to achieve the intended outcome.

HET contains 11 errors under the Mistakes category, which are described in Table 3.

Table 3 Mistakes

<p><i>Application errors</i></p> <p><u>Error Description:</u> Application errors arise from a misunderstanding of the application or problem domain. Additionally, a misunderstanding of some aspect of the overall functionality of the system may cause an application error</p> <p><u>Example:</u> Error: Lack of domain knowledge about how loans, investing, and borrowing works makes the requirement gathering person believe that he/she has received all information required to make decisions on removing loans in default from the repository, whereas his/her belief is incorrect. Fault: Omission of requirement about retaining information of those borrowers who are in default (even after the corresponding loans are deleted from the system).</p>
<p><i>Environment errors</i></p> <p><u>Error Description:</u> Environment errors are due to a misunderstanding or misuse of the hardware or software environment of a given project. Environment comprises all hardware and software used but not developed within a given project (for example, operating systems, devices, data base systems).</p> <p><u>Example:</u> Error: In documenting the LA application system, a standard template for requirement specification (for example, IEEE provides a standard template for SRS) was not used. This means that the right tools for requirement specification were not used. Fault: Requirements about system scope and performance were omitted.</p>
<p><i>Information Management errors</i></p> <p><u>Error Description:</u> Information Management errors are due to a lack of knowledge about standard procedures. This means the RE practitioners (requirement gathering persons and RE analysts) are not aware of the standard practices and procedures defined by the organization.</p> <p><u>Example:</u> Error: It is common procedure for requirement specifications to have error-handling information, but this specification does not contain any error-handling information. Error-handling information contains information about the mechanisms which are invoked when errors are encountered by the system. Fault: Omission of information about throwing error messages when the system encounters a problem (instead of just returning to previous screen, the system should throw an error message which makes the user aware of what went wrong).</p>

Wrong assumptions

Error Description:

The human error that happens here is creation of mistaken assumptions about features and opinions of stakeholders. Wrong assumptions about stakeholder opinions can be made by both, the requirements-gathering people or the requirements analysts.

Example:

Error: Requirement gathering person assumes that error-handling is a task common to all software projects. The wrong assumption leads the requirement gathering person into concluding that error-handling will be taken care of by programmers and he/she does not gather error-handling information.

Fault: Information about what happens if invalid data is received from a lender has been omitted.

Poor understanding of one another's roles

Error Description:

Domain knowledge and perspectives vary between roles, which necessitates considerable communication among members of the software engineering team. Without proper understanding of developer roles, communication gaps may arise, either by failing to communicate at all (due to lack of understanding that other roles are impacted) or by ineffective communication (e.g. missing tacit requirements due to lack of insight into the customer's domain.)

Example:

Error: It was not clear among team members, who needed to elicit the requirements of a bank lender, which affected the participation of an important stakeholder during the requirements process.

Fault: Omitted functionality as requirements of a bank lender (i.e., the LA application system to handle both fixed rate loans and adjustable rate loans) were not recorded in the specification.

Mistaken belief that it is impossible to specify non-functional requirements in a verifiable form

Error Description:

Major causes of this problem are the prevalent myths that it is too costly, too difficult, and even impossible to produce good requirements, especially nonfunctional requirements, during the software engineering process. These myths are especially prevalent with regard to quality and specialty engineering requirements (e.g., availability, interoperability, performance, portability, safety, security, and usability), where there is still a prevailing but mistaken belief that it is impossible to specify these requirements in a verifiable form.

Example:

Error: An absence of any performance, security, usability, or availability requirements suggests that all non-functional requirements were overlooked.

Fault: Omission of performance requirements, security requirements and other non-functional requirements.

Not having a clear distinction between client and users

Error Description:

If RE practitioners are not able to distinguish between clients and end users and they fail to gather and analyze requirements from the user's point of view.

Clients are frequently unable to describe the knowledge and abilities of the end users of the proposed system.

Example:

Error: The requirement gathering person failed to gather information from the actual end user of LA system, the Loan Analyst.

Fault: No functional requirement to edit loan information has been specified whereas 'Purpose' specifies loans can be edited.

Lack of awareness of sources of requirements

Error Description:

Requirements gathering person is not aware of all stakeholders which he/she should contact in order to gather the complete set of user needs. Sources of requirements include all different types of end users of the system being built and all the decision-makers from project sponsoring organization (also called the customers or the clients)

Example:

Error: Requirement gathering person was not aware of all end users and clients and did not gather the needs of a bank lender (one of the end users of LA system). This end user wanted the LA system to handle both fixed rate loans and adjustable rate loans.

Fault: Omitted functionality as requirements only considers fixed rate loans.

Problem-Solution errors

Error Description:

Problem-Solution errors are due to not knowing, misunderstanding, or misuse of problem solution processes. This kind of errors occur in the process of finding a solution for a stated and well-understood problem. If RE analysts do not understand the correct use of problem-solving methods and techniques, they might end up analyzing the problem incorrectly, and choose the wrong solution.

Example:

Error: The requirements engineer misunderstood that some crucial functionality had to be delivered before other functionality, so he/she chose the waterfall lifecycle rather than an incremental one.

Fault: Required functionality cannot be delivered on time to the customer.

Inadequate Requirements Process

Error Description:

The requirements engineering plan is incomplete or inadequate for the current project. The development team did not fully appreciate all of the steps needed to ensure the final software was complete and met specifications. As a result, one or more steps in the process was omitted from the plan.

Example:

Error: Requirements engineering plan did not have sufficient requirement traceability measures. As a result, a requirement describing the ability of loan analyst to change the borrower information cannot be traced to any user need.

Fault: An extraneous requirement is included that could result in extra, unnecessary work for the developers and perhaps some unwanted or undesirable functionality in the final product.

Syntactic errors

Error Description:

Syntax describes the grammatical rules for specifying correct arrangement of elements to produce meaning. In English, word order conveys grammatical role. For example, 'The dog bit the boy' means something quite different from 'The boy bit the dog' even though both contain the same words. In the context of RE, syntactic errors can happen in the following 2 ways:

- Incorrect usage of grammatical rules of natural language, or
- Incorrect usage of rules, symbols, and standards used in a formal requirement specification technique like UML diagramming.

Example:

Error: The requirements engineer did not realize that association links on UML diagrams must include navigability arrows when one use case (functionality) 'extends' another.

Fault: An association link on a UML diagram between two classes lacks a navigability arrow depicting the directionality of association. The diagram is ambiguous and can be misunderstood.