Training

Inspection of Software Requirements Document

Wenhua Hu

whu10@crimson.ua.edu

The University of Alabama

Error - Faults

As per IEEE standard terminology:

- Fault is a concrete manifestation of an error in a software artifact.

- Error is a defect in the human thought process made while trying to understand given information, solve problems, or to use methods and tools. In the context of software requirements specifications, an error is a basic misconception of the actual needs of a user or customer.

Software Fault Types

Type	Description
Omission (0)	Necessary information about the system has been omitted from the software artifact.
Incorrect Fact (IF)	Some information in the software artifact contradicts information in the requirements document or the general domain knowledge
Inconsistent Information (II)	Information within one part of the software artifact is inconsistent with other information in the software artifact.
Ambiguous Information (AI)	Information within the software artifact is ambiguous, i.e. any of a number of interpretations may be derived that should not be the prerogative of the developer doing the implementation.
Extra Functionality (EF)	Information is provided that is not needed or used
Wrong Section (WS)	Other defects' e.g. a requirement may be found in an inappropriate section of the document

Requirement defects for Loan Arranger System– Example

- **Functional Requirement 2 (F2)**
  - There are two types of loans based on the amount of the loan: regular and jumbo. A regular loan is for any amount less than or equal to $275,000. A jumbo loan is for any amount over $275,000.
- **Functional Requirement 4 (F4)**
  - The application must respond to a loan analyst's request for information. In less than five seconds from submission of the request. The loan analyst must receive the results of a loan bundle optimization request within 6 seconds from submission of the request.

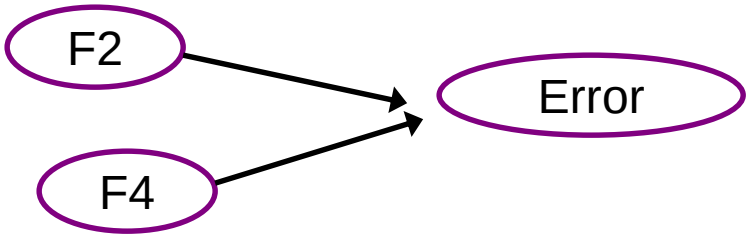
Error Abstraction

- Error abstraction process helps to abstract errors/mistakes from the faults. It includes doing:
  - Analysis of the fault lists
    - *Why each fault (in your fault report form) represents a defect in the SRS?*
  - Grouping of the related faults
    - *Group faults based on their categories or nature (e.g. II, IF, WS)*
  - Eliciting the underlying reasons for the occurrence of the faults
    - *Find pattern in the grouped faults and think of some believed reasoning for these faults to have occurred*
  - Write down the errors (***Mapping errors to faults***)

Error Abstraction

- F2 and F4 – Inconsistent Information (II) class.

  Error can be that: “ we do not know which data requirement should be correct .”

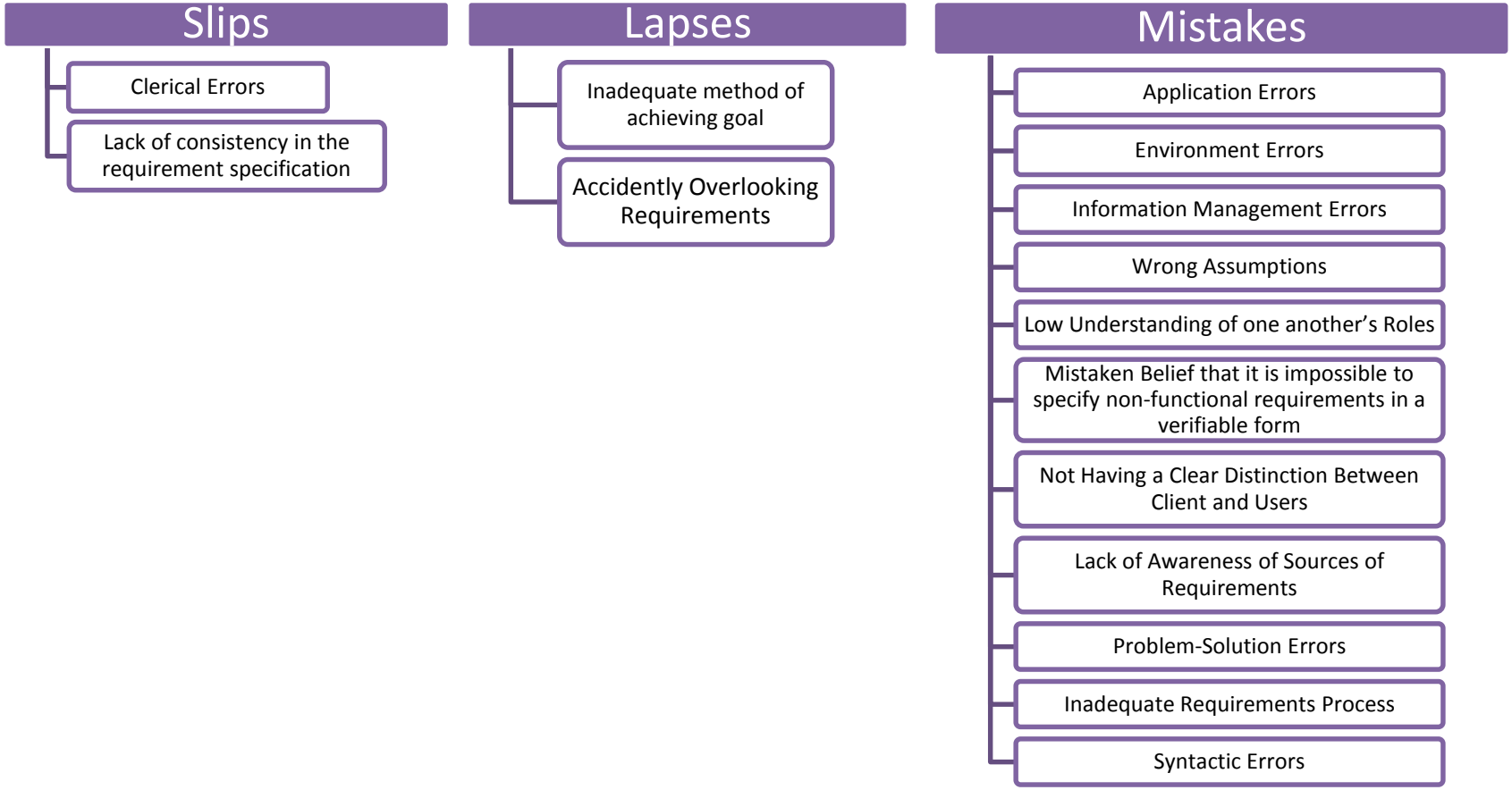


- Remember:
  - It is not always the case that you will find an error responsible for multiple fault (as in above example) ; Error can be responsible for single faults, and
  - Patterns can also be found between errors in different classes

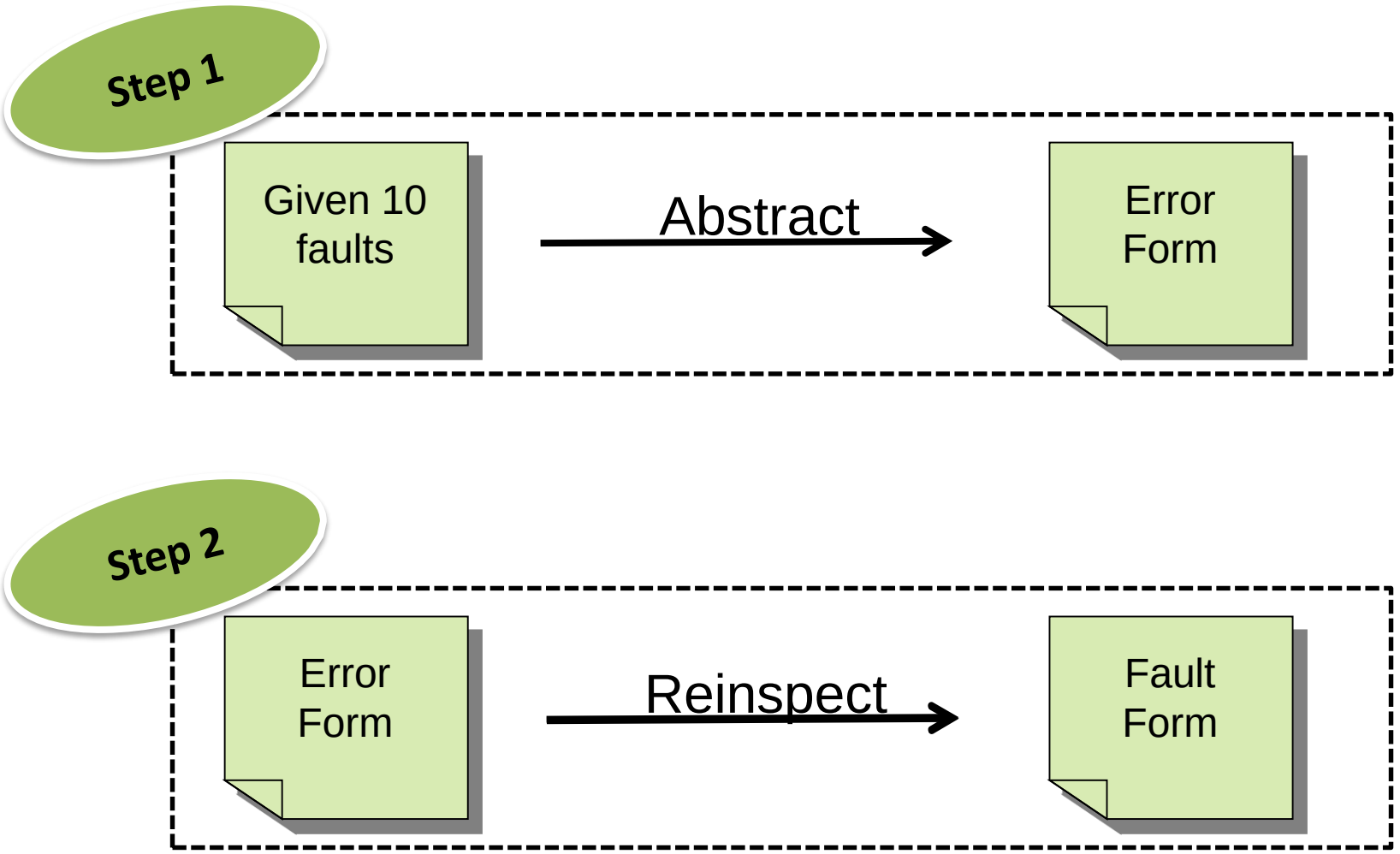
Error Abstraction

- Abstracting errors from faults is a very creative process.
- To support the error abstraction process, you can use the *Human Error Taxonomy* that describes the different types of errors that can occur during the development of requirement document.

Human Error Taxonomy



Outline – Exercise



Exercise Tasks

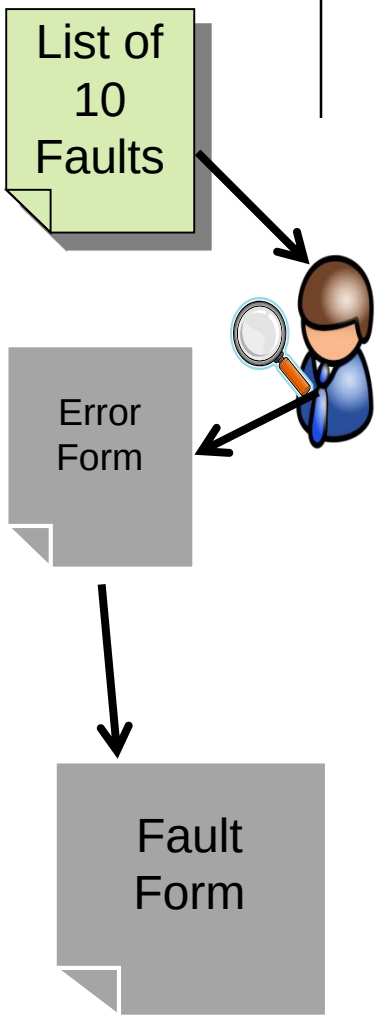
Fault #	Line#	Requirement #	Fault Class	Description

- Abstract and Classify Errors
- Fill an Error Report Form

Error #	Related Faults #	Error type	Description	Time Found	Breaks (Time and Date)

- Inspect SRS using errors to find the rest of the faults.
- Fill a Fault Report form

Fault # (beginning from 11)	Error# (from sample error form)	Error type	Line #	Requirement #	Fault Class	Description	Time Found	Breaks (Time and Date)

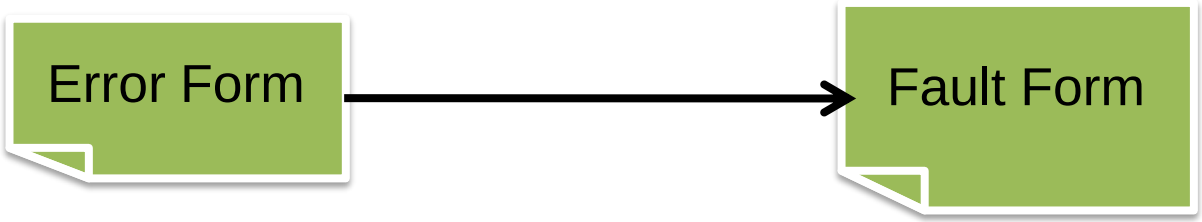


Error Form : Example

Error #	Related Fault #	Error types	Description of Error	Time found	Break (time and date)
1	1	Slips – clerical error	……. …….	9:30 AM	
2	3, 7	Lapses – accidently overlooking requirements	……. ……	10:00 AM	Break:10 AM; 5th Feb
3	4	Mistakes – application error	……. ……	1 PM	Resume 12 PM; 6th Feb

Inspecting SRS: Use error information to find more faults

- Use the errors identified on your “Error Form" and your knowledge of the Human Error Taxonomy, inspect the SRS document:
  - For each error in the “Error Form”, inspect the SRS for fault(s) caused by it
  - For each new fault found, complete a row in the “Fault Form"
  - An error can cause one or more faults
  - If you find a new fault that not related to one of the errors you abstracted in the “Error Form”, record this fault information in “Fault Form”, then create a new error on the “Error Form” that covers this fault, and see if there are any other faults related to this new error.



Fault Form

Error #	Fault #	Human Error (HET)	Description of Error	Time found	Break (time and date)

Fault # (beginning from 11)	Error# (from sample error form)	Error type	Line #	Requirement #	Fault Class	Description	Time Found	Breaks (Time and Date)

Fault From : Example

Fault #	Error# (from the error-form)	Error Type	Line #	Requirement #	Fault Class	Description	Time Found	Breaks (Time and Date)
11	5	Slips – clerical error	35	FR 3	II	…….	9:45 AM	
12	2	Lapses – accidently overlooking requirements	234	FR7	MF	…….	9:50 AM	
13	7	Mistakes – application error	309	FR10	**MI**	…….	9:59 AM	

Thank you!

whu10@crimson.ua.edu