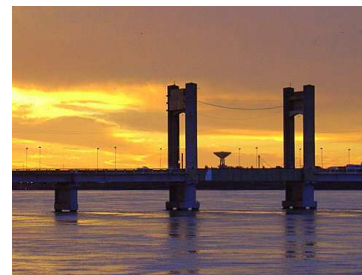


AIRDoc – An Approach to Improve the Quality of Requirements Documents: Dealing with Use Case Models

Ricardo Argenton Ramos

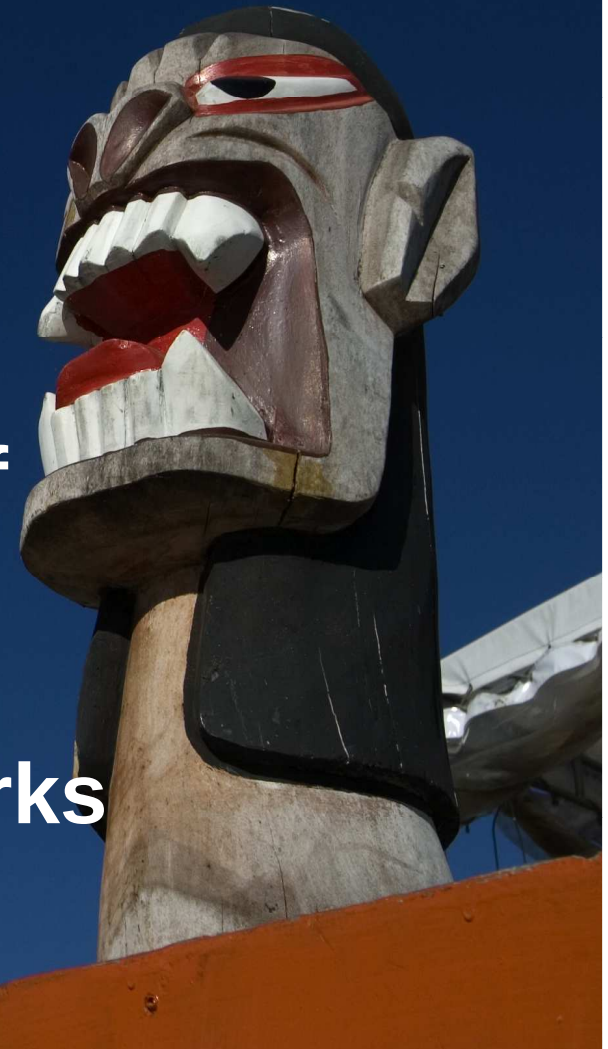
Orientador: Jaelson Castro

Co-orientador: João Araújo



Agenda

- Introduction – *The Problem and the Proposed Solution*
- AIRDoc
- An Experimental Validation of the AIRDoc Process
- Contributions and Future Works



Introduction (*the problem*) – 1/2

- Over the past few years, a set of typical issues seems to plague the Use Case Models. For example:
 - Use case that have been abandoned and are no longer meaningful,
 - Use case descriptions that are unnecessarily long and complex,
 - Information that is duplicated, scattered, tangled,
 - Among others ...

[Lilly 1999]

Introduction (*the problem*) – 2/2

The removal of these problems in early stages of software development process reduces the costs associated with software changes. These cost reductions could be three to six times more in later stages than during requirements activities [Pressman 2005], [Sommerville, 2004] .

How to Solve these Problems?

- Inspection techniques ?

[Travassos et al., 1999] [Fagan, 1986]

- Aspect Orientation ?

[Moreira et al., 2005], [Silva, L.; Leite, J. 2005], [Sousa, G; Castro, 2004]

- Good practices ?

[IEEE Std 830-1998], [IEEE 1061, 1998], [Firesmith, 2007]

- Metrics ?

[Fenton and Neil, 2000]

The Proposed Solution

We propose to use metrics to discover the potential problems

- The use of software metrics reduces subjectivity in the assessment and control of software quality by providing a quantitative basis for making decisions about software quality [IEEE 1061, 1998]

The Proposed Solution

We use the Goal Question Metrics approach to help the metrics application

- The GQM approach is based on the assumption that for an organization to measure in a purposeful way it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals [Basili et al.,1994]

The Proposed Solution

**We use the framework proposed by
the standard [IEEE 1061, 1998]**

The GQM approach needs to have a quality model to achieve the goal, question and metrics definitions.

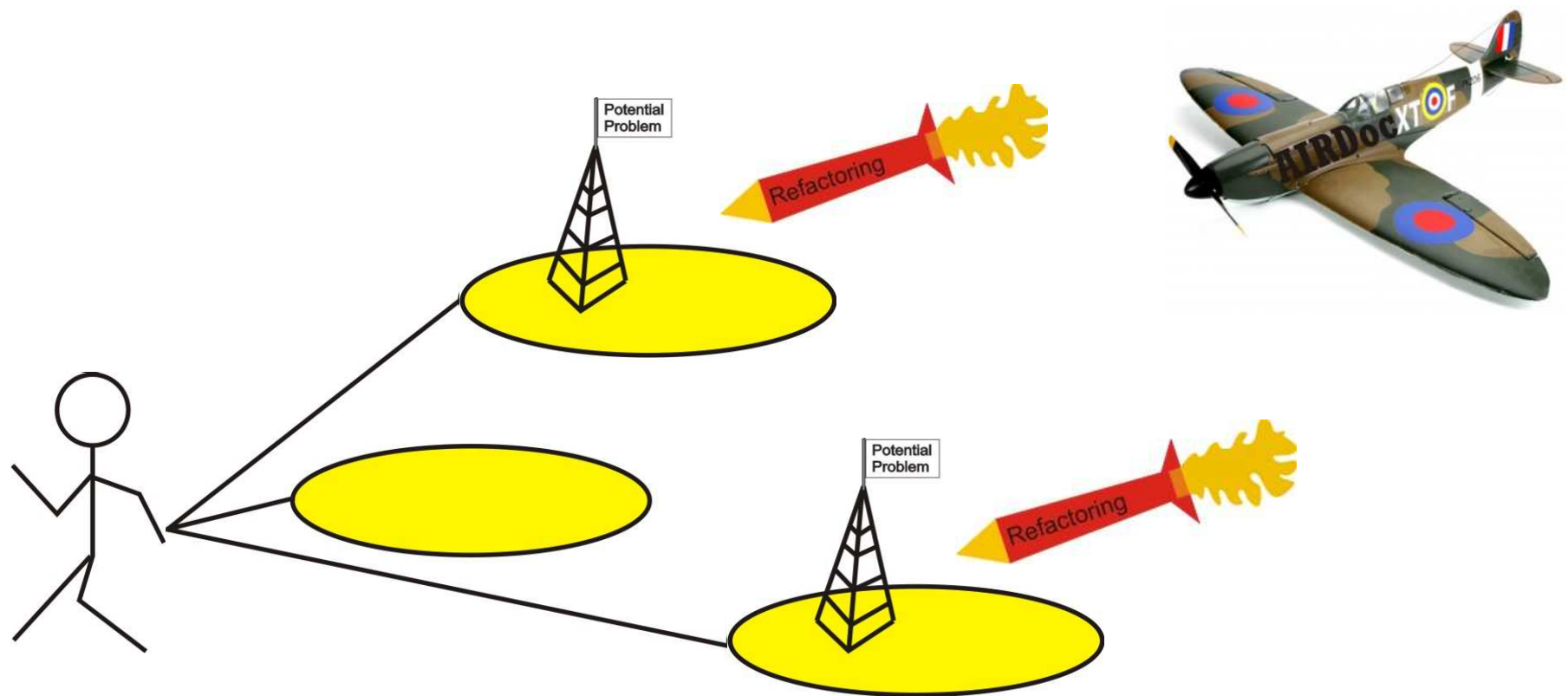
The Proposed Solution

Once, we have measured the appropriate quality factor, our AIRDoc process will be able to possible detect some potential problem.

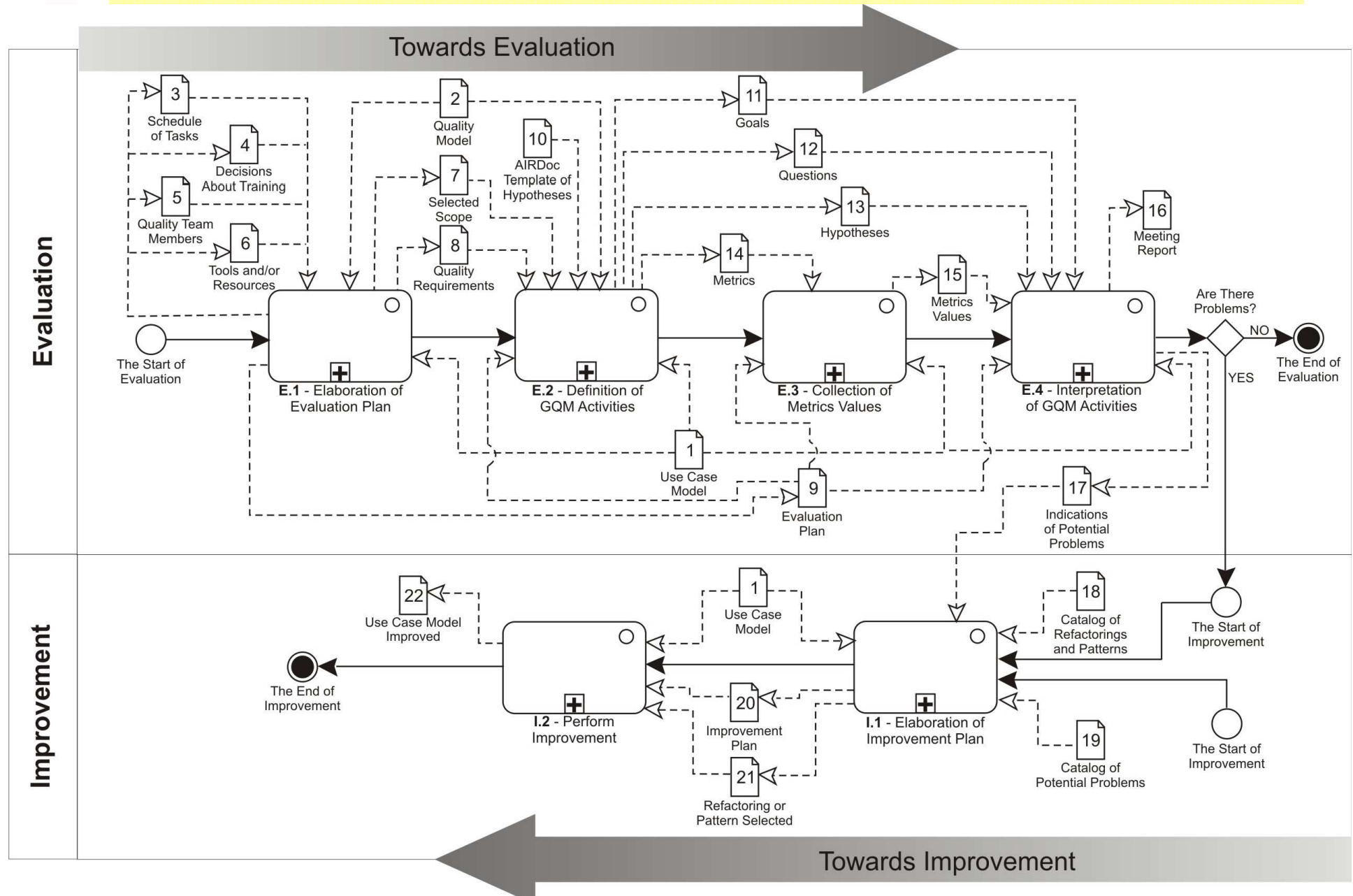
We propose the solution to the problems in terms of some refactorings to be performed

Refactorings can be classified according to which of the quality attributes they affect [Mens and Tourwé, 2004].

AIRDoc - Approach to Improve the Quality of Requirement Documents

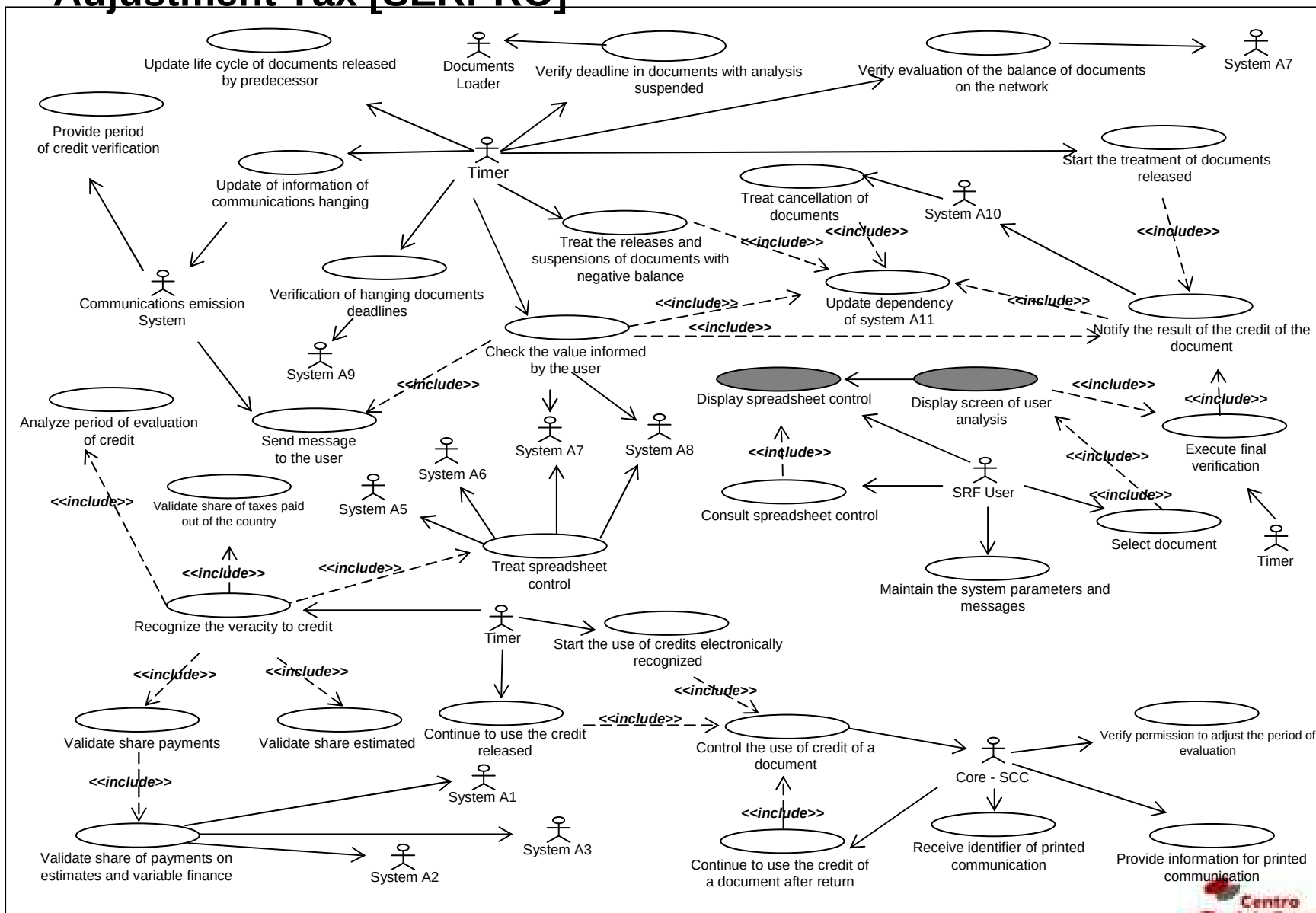


The AIRDoc

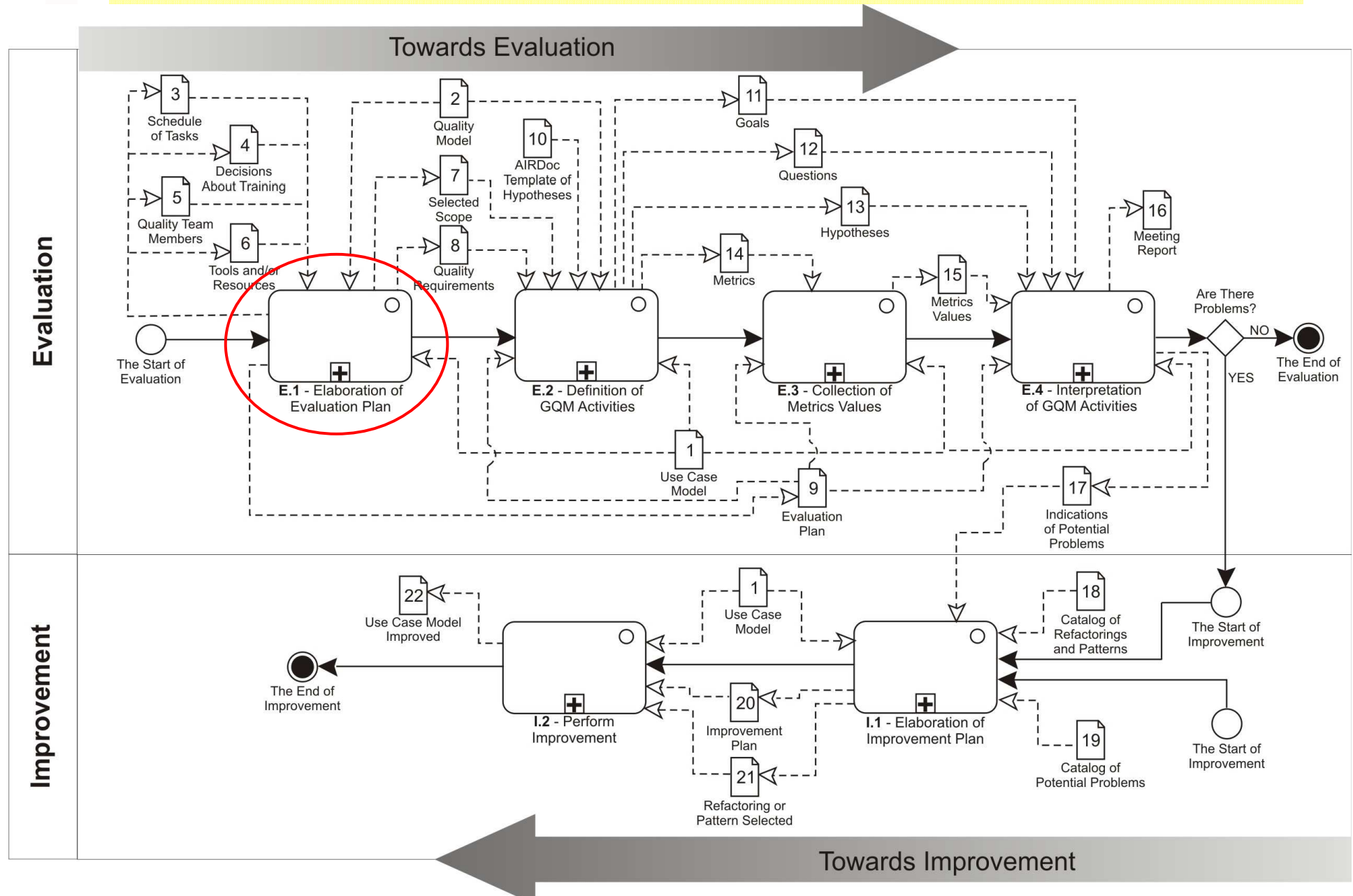


A Running Example

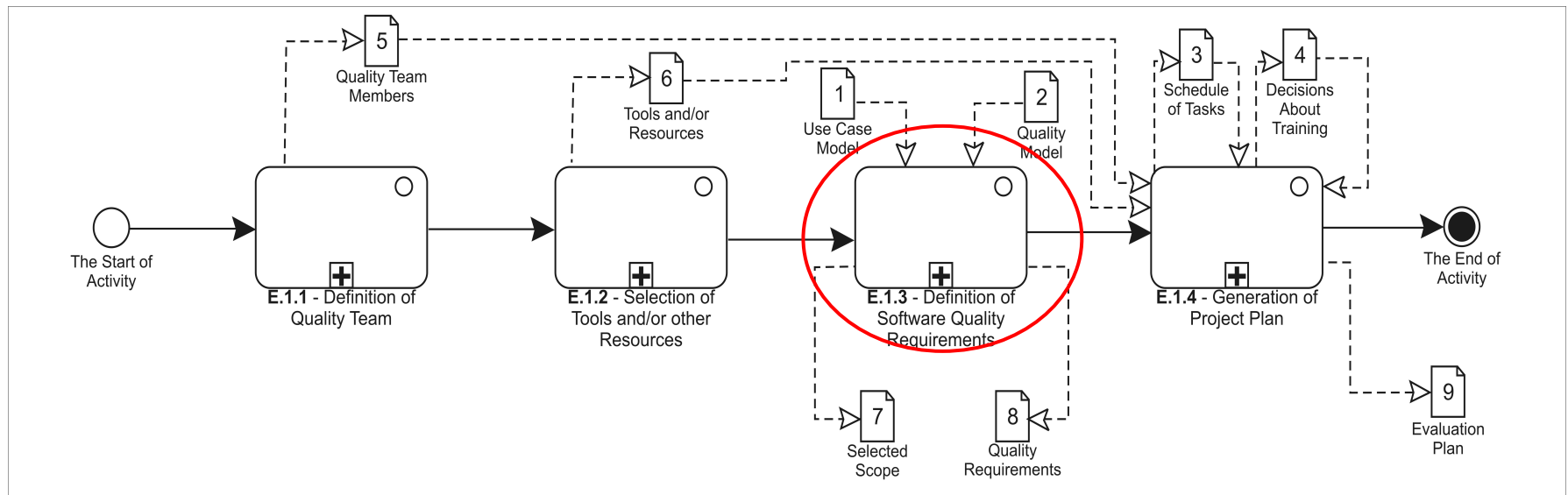
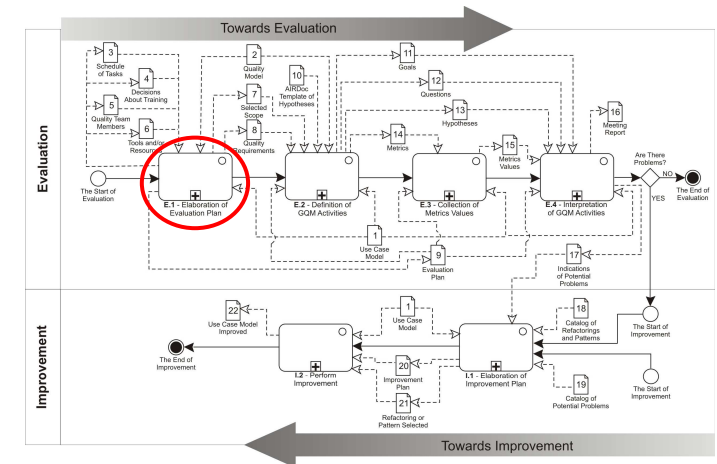
Adjustment Tax [SERPRO]



The AIRDoc

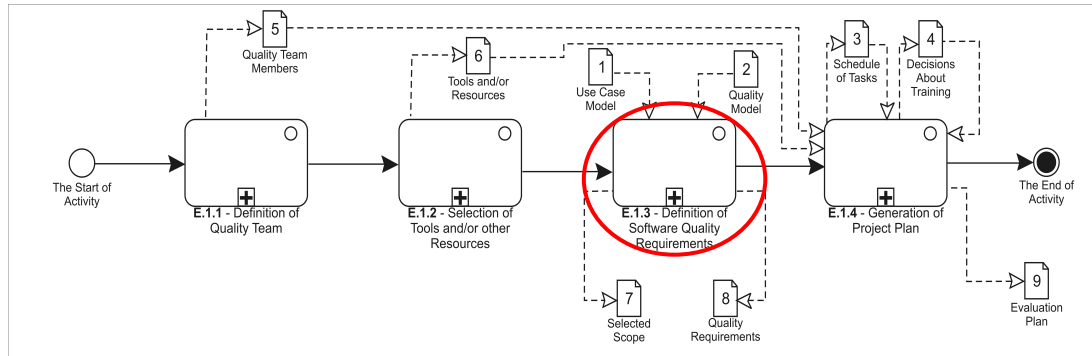


Elaboration of Evaluation Plan

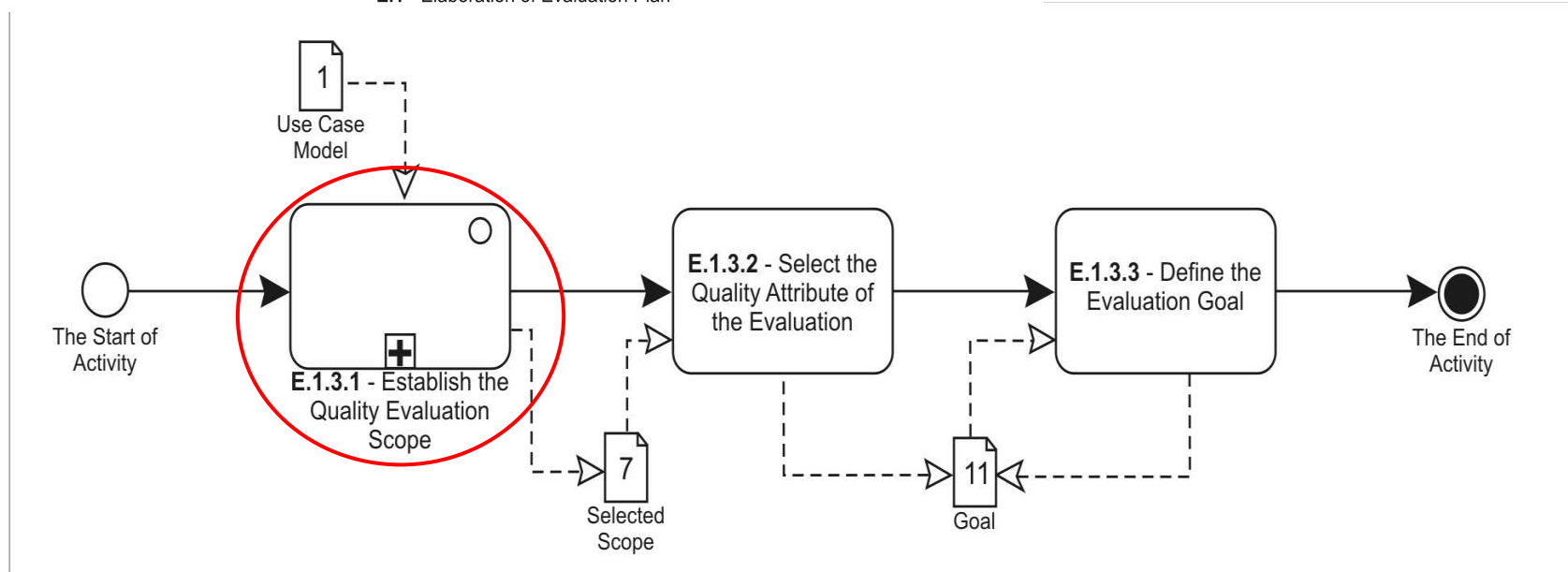


E.1 - Elaboration of Evaluation Plan

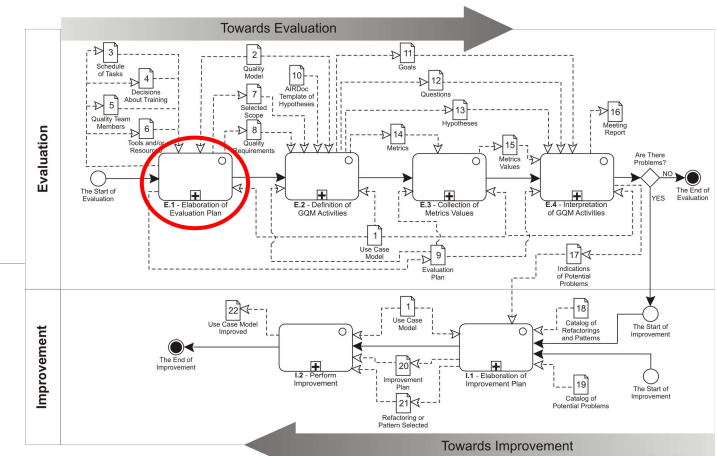
Definition of Software Quality Requirements



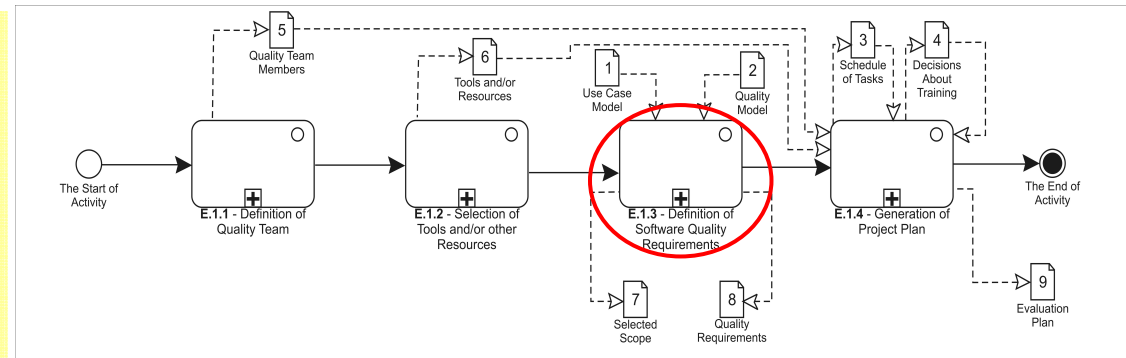
E.1 - Elaboration of Evaluation Plan



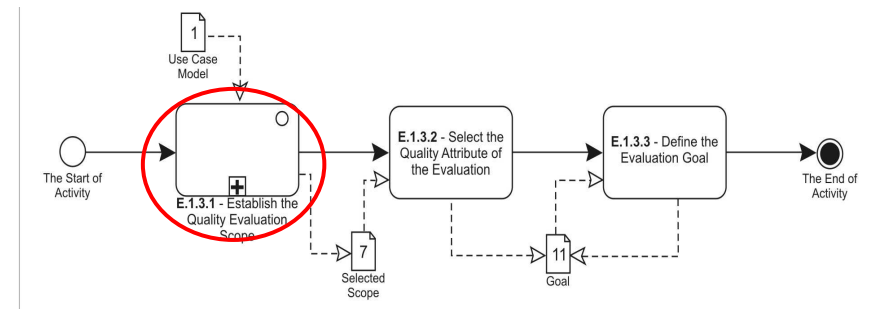
E.1.3 - Definition of Software Quality Requirements



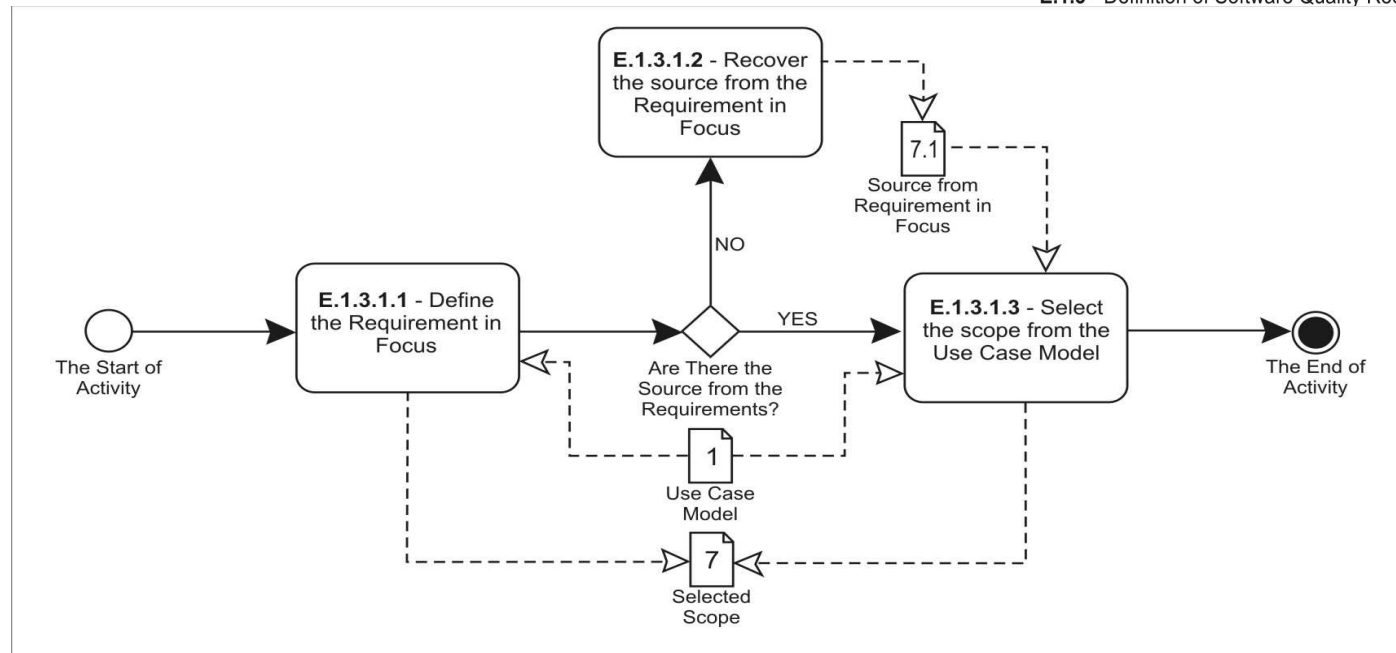
Establish the Quality Evaluation Scope



E.1 - Elaboration of Evaluation Plan

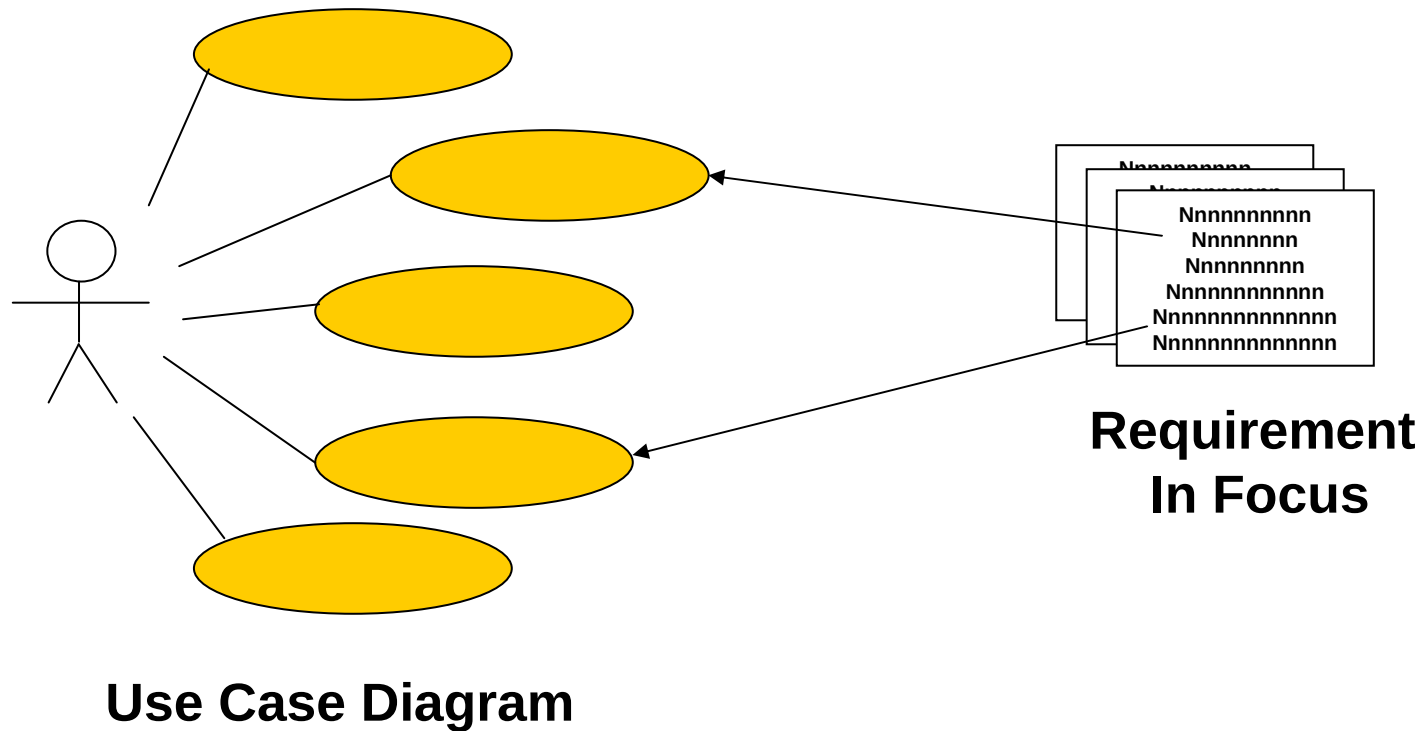


E.1.3 - Definition of Software Quality Requirements



E.1.3.1 - Establish the Quality Evaluation Scope

Mapping of: Requirement in Focus and Use Cases



Template to Describe the Requirement in Focus

Requirement in focus	<i>< Identify the requirement in focus by a name. Create a list with the name(s) of use case(s) that are directly related with the requirement in focus. In some cases it is necessary to describe the steps that are in other use cases.></i>
“Display Requirement”	use cases: 1 - “Display spreadsheet control”, 2 - “Display screen of user analysis”

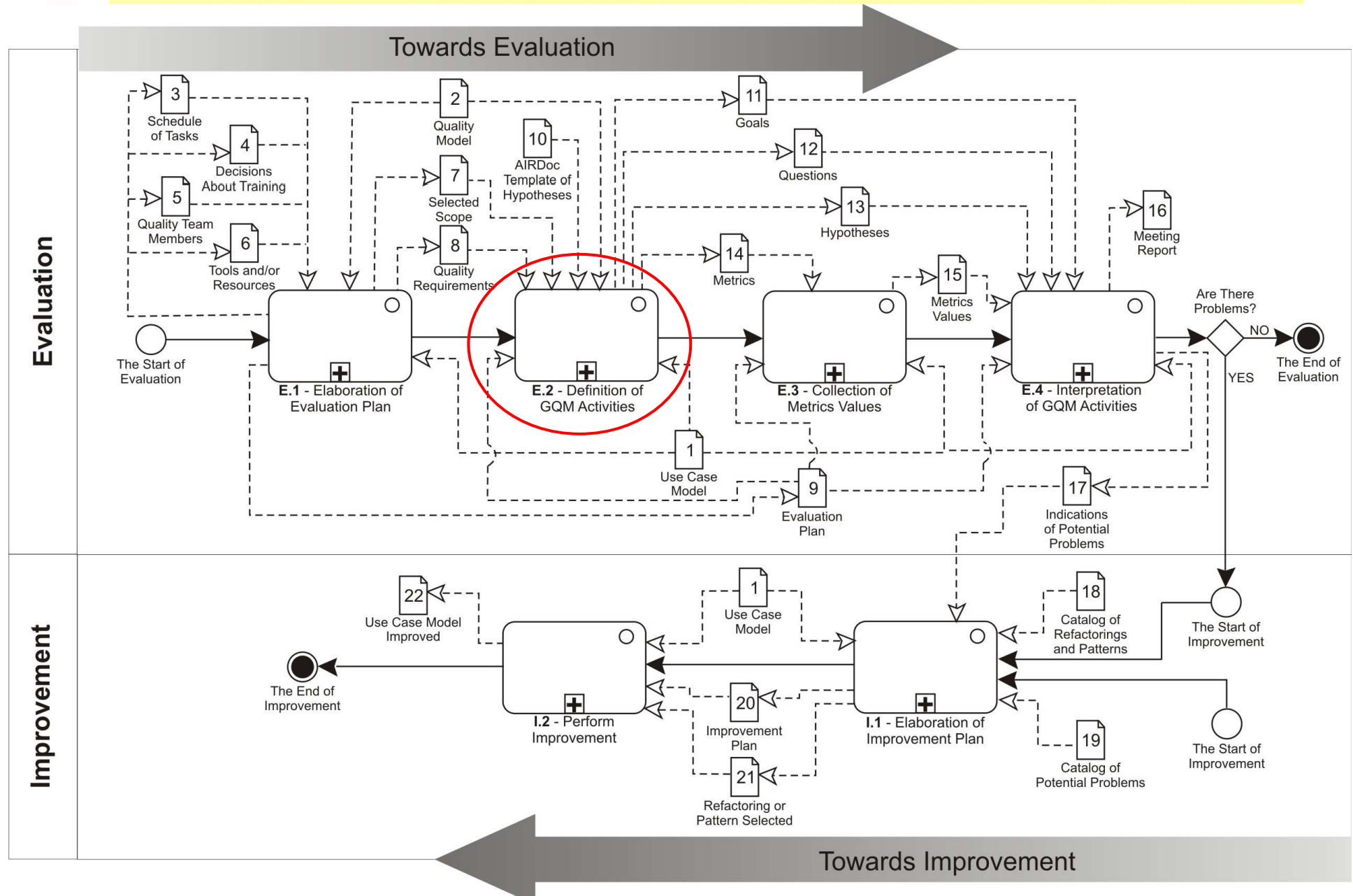
Template to describe the Source from the Requirement in Focus

Source from Requirement in Focus	<i><Describe the information about the source, such as:</i> <i>- description about the stakeholder who originated the requirement;</i> <i>- type of source (interview, annotation, protocols, laws, rules, etc.);</i> <i>Include the description where the requirement existence is evidenced.></i>
Source from “Display Requirement”	<i>Stakeholder - SRF User.</i> <i>The sources from the requirement in focus are dispersing on:</i> <i>- the laws nº 11.773/2008 (DOU of 18.9.2008) and 10.833/2003 (DOU of 30.12.2003)</i> <i>- meeting reports from the SERPRO Units.</i>

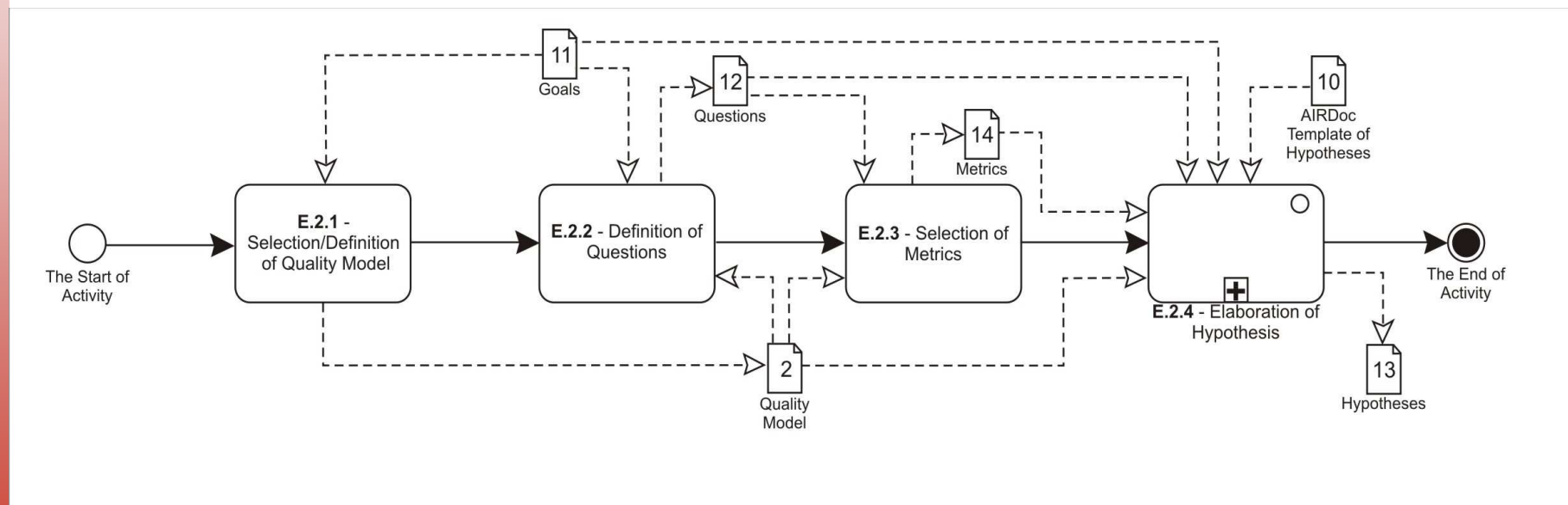
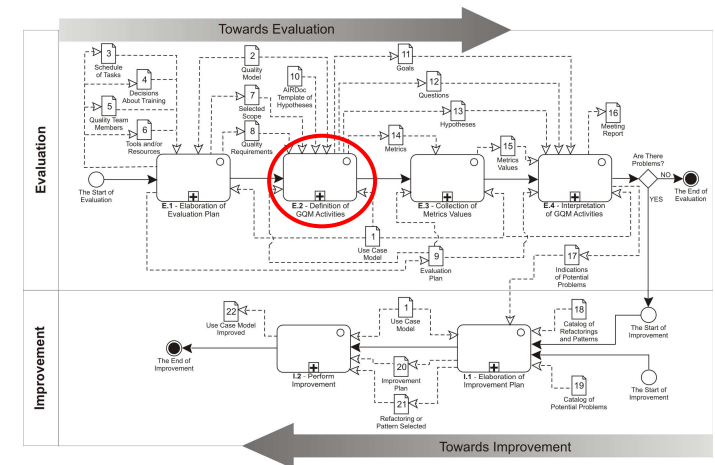
Template of the evaluation goal

Evaluate in	<scope>	<the quality attribute>	of	<requirement in focus>
Evaluate in	Adjustment Tax	the maintainability	of	the display requirement

The AIRDoc

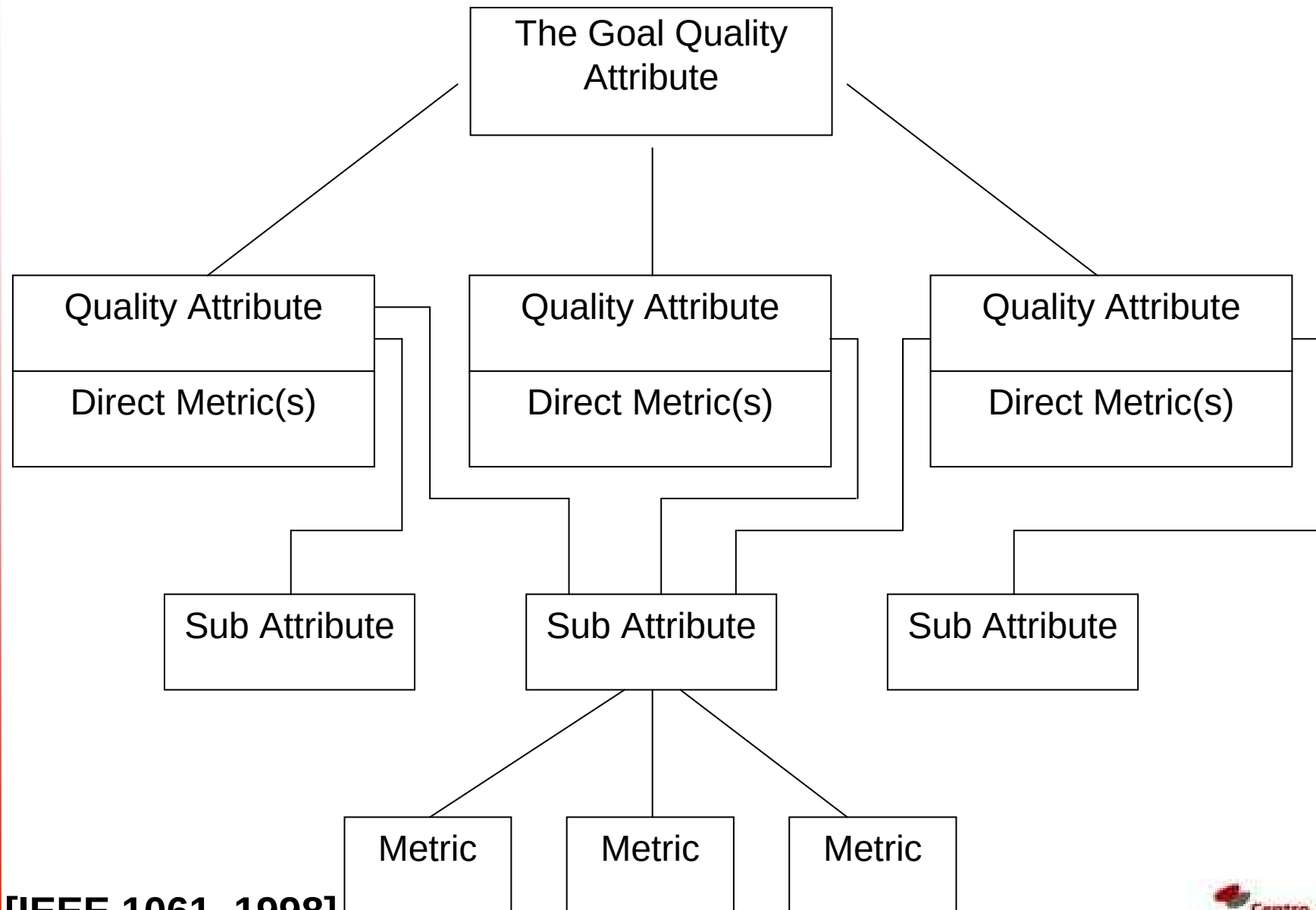


Definition of GQM Activities



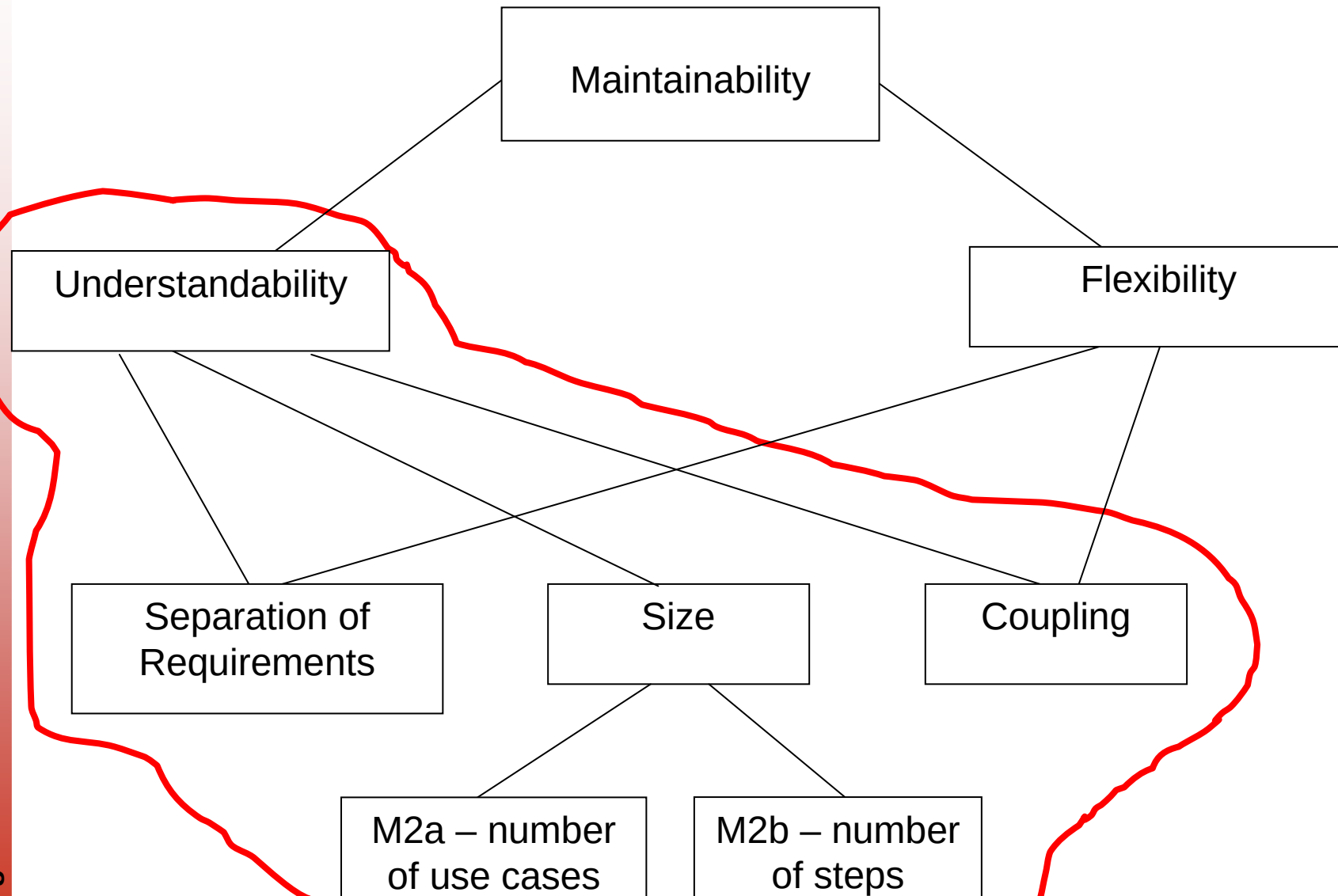
E.2 - Definition of GQM Activities

Selection/Definition of Quality Model



[IEEE 1061, 1998]

An Example of Quality Model



Definition of Questions

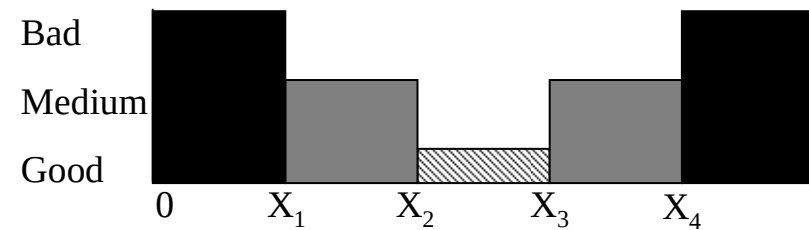
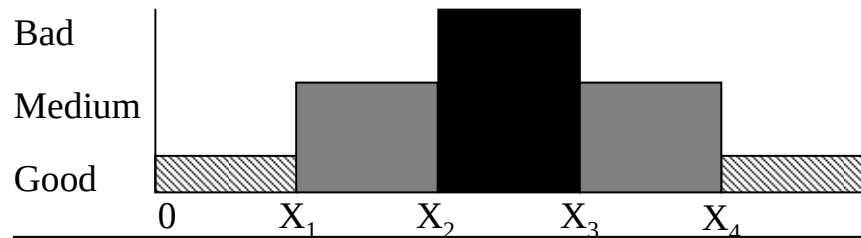
Quality attribute	Question	Source of the Answer
<name of the attribute or sub attribute >	<i><question(s) that when answered will provide the insights necessary to achieve the goal. The questions will be answered basically by the words: Good, Medium or Bad></i>	<i><the sources (metrics or other questions) that need be achieve to answer the question></i>
	Template: How good is the <quality attribute> from the <requirement in focus>?	
Understandability	How good is the understandability from the display requirement?	Q1.1. How good is the size from the display requirement? Q1.2. How good is the separation of requirements from the display requirement? Q1.3 - How good is the coupling from the display requirement?

Selection of Metrics

Questions	Metrics	Details
<i><question that are related with the metric></i>	<i><description of the metric></i>	<i><details about: the required value, how to obtain the value, among others></i>
Q1.1 - How good is the size from the display requirement?	M2a – How many use cases are required to specify the display requirement?	Count the number of use cases where there is, at least, one step that contributes to the specification of display requirements.
	M2b – How many steps are required to specify the display requirement?	Count the total numbers of steps that describe the display requirement.

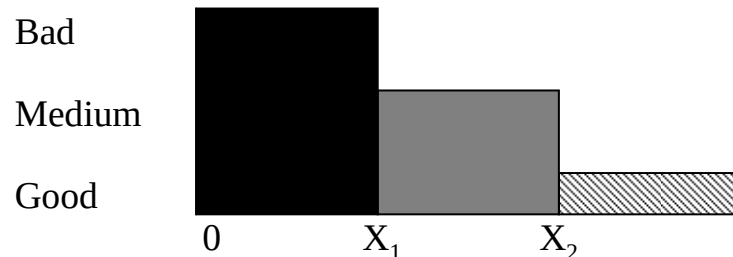
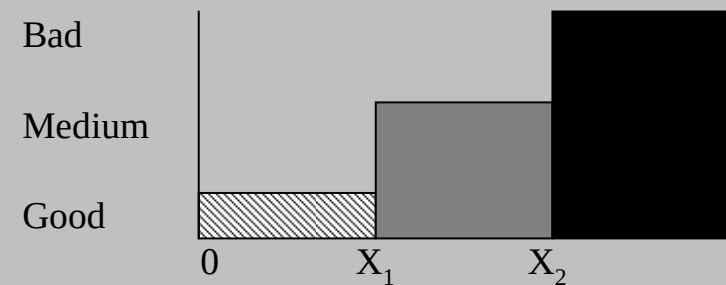
Create a Premisse

Scales for transformation of numerical values



Template:

The value of *<metric/function>* is "good" if its value is in the range $[0, x_1]$, is "medium" if its value is in the range $[x_1, x_2]$ and is "bad" if its value is in the range $[x_2, \infty]$.



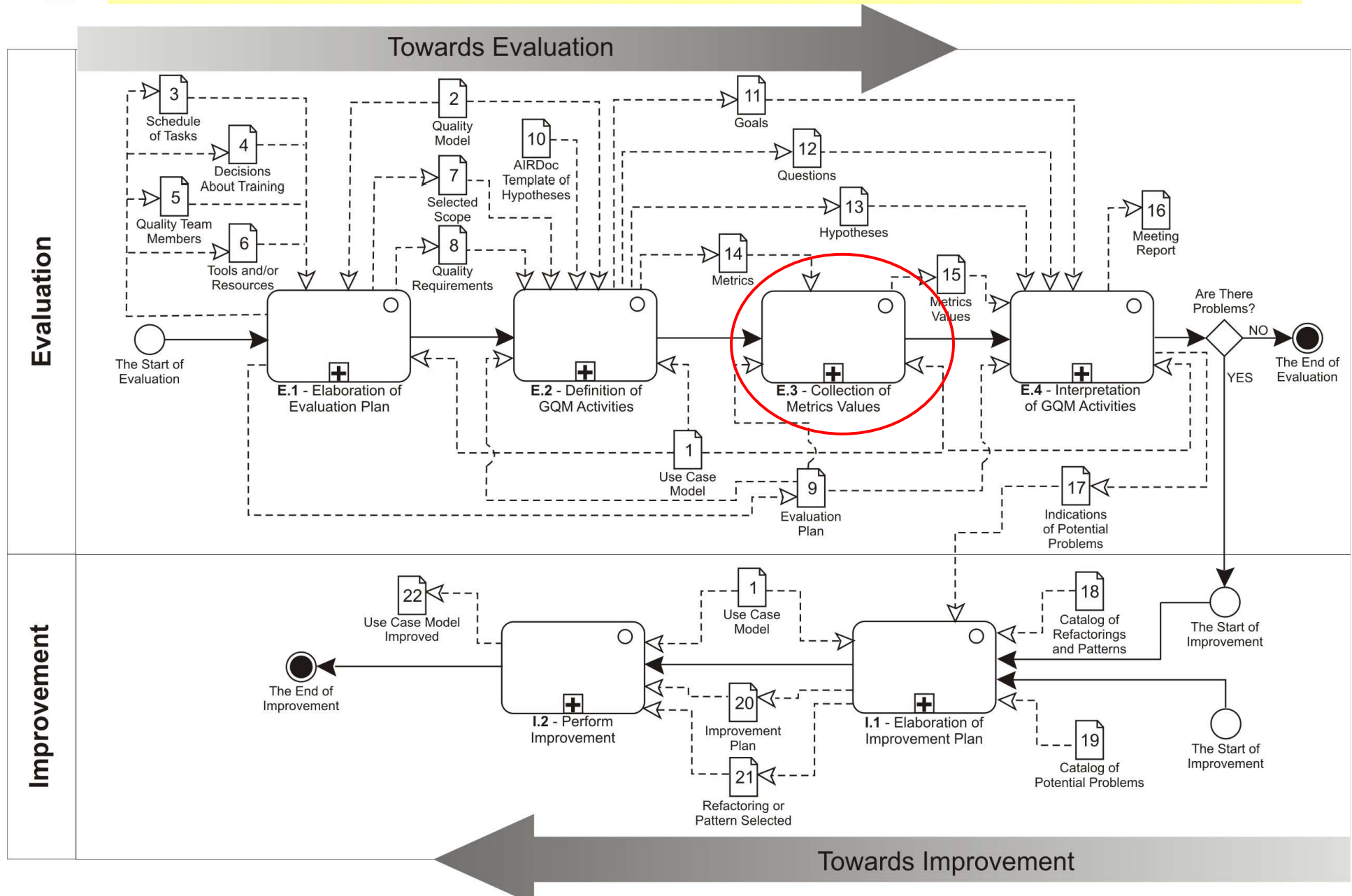
Create a Premisse

Metric	Possible values	Premise
<i><name or some other identification of the metric></i>	<i><range of possible values></i>	<i><create a premise analyzing the range of possible values and transform it in a scale of 3 values: Good, Medium and Bad></i>
M2a – How many use cases are required to specify the display requirement?	Function2 = M2b/M2a M2a [1 - 50] M2b [1 - 800] Type of Scale: Increasing [1 - 35] – Good [36 - 65] – Medium [66 - 800] – Bad	The value of the Function2 is "good" if its value is in the range [1, 35], is "medium" if its value is in the range [36, 65] and is "bad" if its value is in the range [66, ∞].
M2b – How many steps are required to specify the display requirement?		

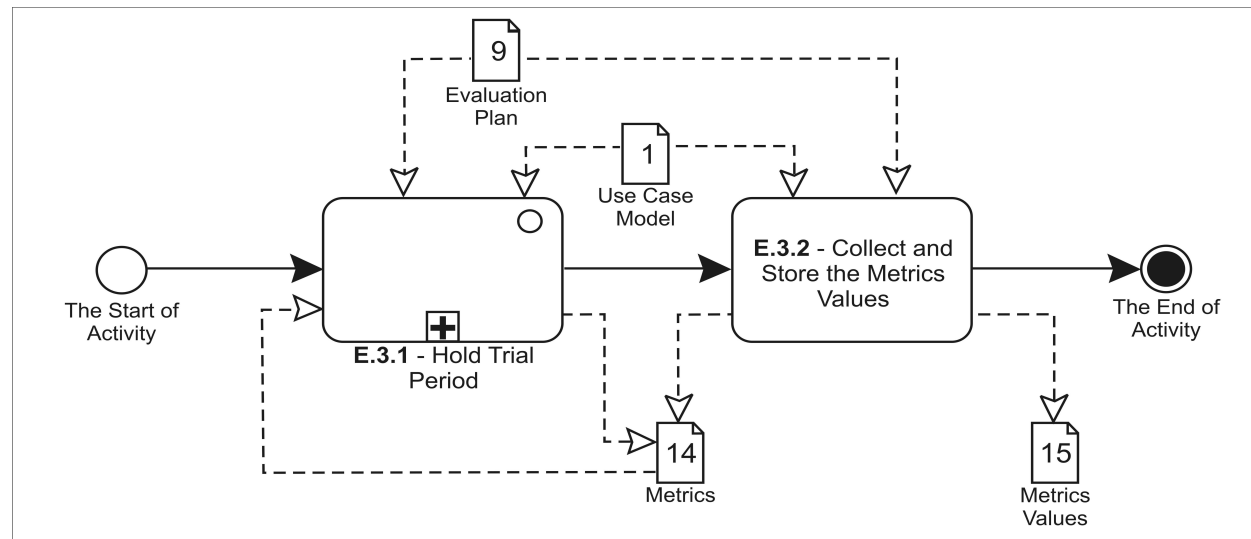
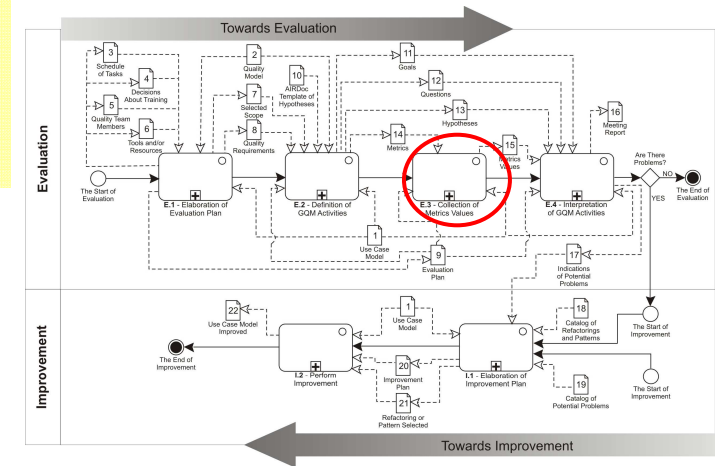
Elaborate Hypothesis to Each Question

Question	<question>
Premise	<premise (created in step E.2.4.1)>
Function	<if some premise is based on a function, it should be described here>
Hypothesis	The <quality attribute/sub attribute> from the <requirement_in_focus> is <Good/Medium/Bad>. Because the value of the <metric/function> is <equal/lower/higher> to/than <metric/function value> (and <equal/lower/higher> than <metric/function value>) Note: At least three hypotheses must be elaborated, each one to each value “Good, Medium and Bad”
Note	<if necessary insert some note about the hypothesis>
Question	Q1.1 - How good is the size from the display requirement?
Premise	The value of the Function2 is "good" if its value is in the range [1, 35], is "medium" if its value is in the range [36, 65] and is "bad" if its value is in the range [66, ∞].
Function	Function2 = M2b/M2a
Hypothesis	H1.1a The size from the display requirement is <i>Good</i>. Because the value of the Function2 is <i>lower</i> than 35. H1.1b The size from the display requirement is <i>Medium</i>. Because the value of the Function2 is higher than 36 and lower than 65. H1.1c The size from the display requirement is <i>Bad</i>. Because the value of the Function2 is <i>higher</i> than 66.
Note	None

The AIRDoc



Collection of the Metrics Values

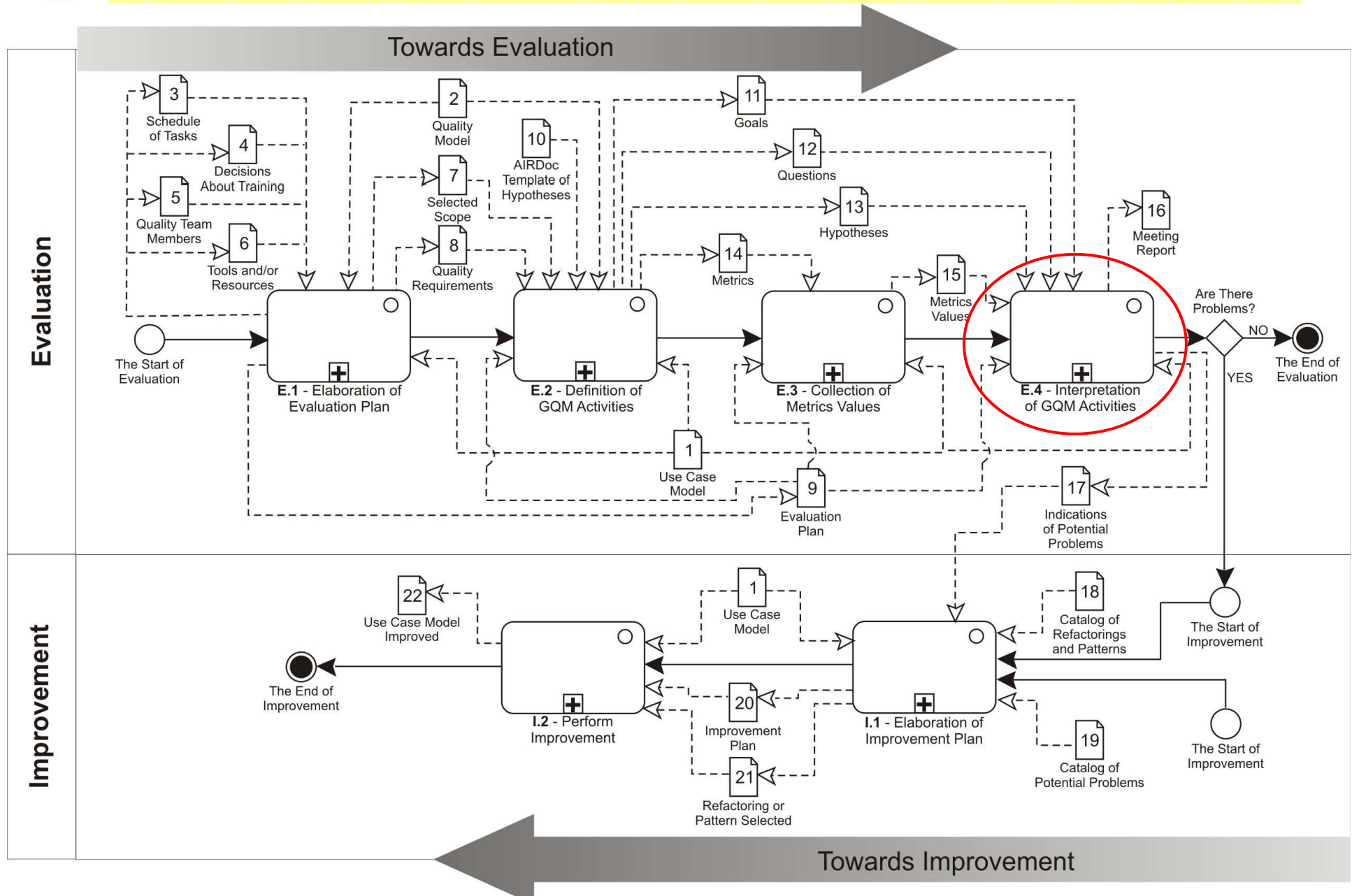


E.3 - Collection of Metrics Values

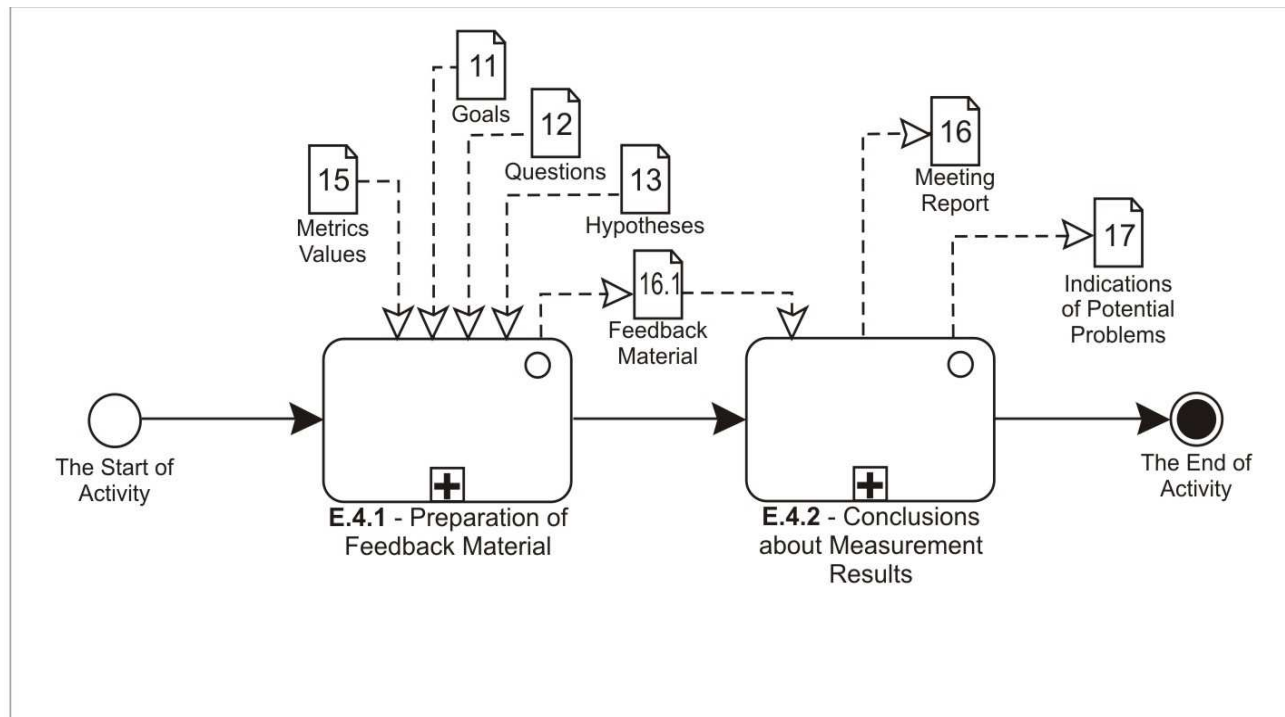
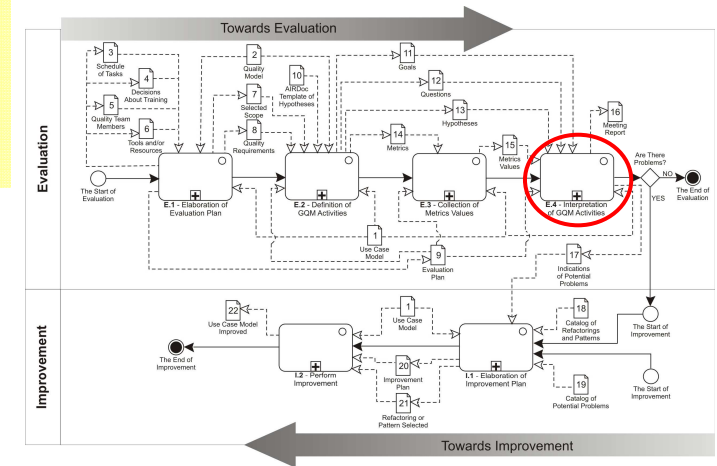
Collect and Store the Metrics Values

Metric	Value
<i><metric></i>	<i><numerical value obtained by direct measurement></i>
M2a – How many use cases are required to specify the display requirement?	2
M2b – How many steps are required to specify the display requirement?	798

The AIRDoc



Interpretation of GQM Activities



E.4 - Interpretation of GQM Activities

Accept the hypothesis and Answer the Questions

Question	Answer	Note
<i><question></i>	<i><hypothesis accepted></i>	<i><some note about the question or the answer></i>
Q1.1 - How good is the size from the display requirement?	H1.1c The size from the display requirement is Bad. Because the value of the Function2 is higher than 66.	The value obtained in the Function2 = $M2b/M2a$ is $798/2 = 399$.

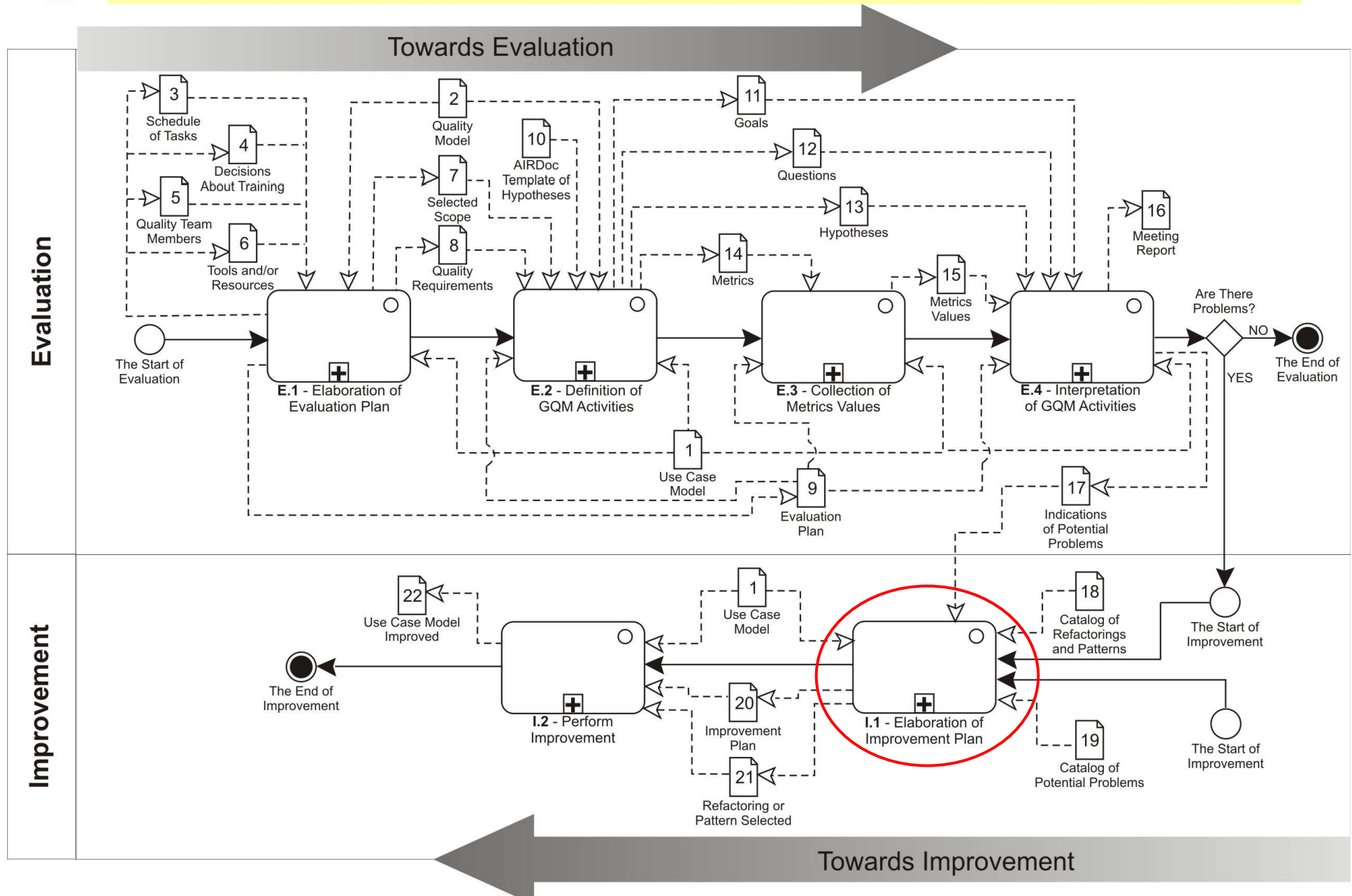
Analyze and Interpret the Hypotheses Questions and Goals

Question	Answer in analysis	Note
<question>	<i><hypotheses accepted with represent a Bad or Medium values and/or hypotheses rejected with represent a Good value></i>	<i><some note about the analysis></i>
Q1.1 - How good is the size from the display requirement?	Accepted -> H1.1c The size from the display requirement is Bad. Because the value of the Function2 is higher than 66.	The two use cases that describe the “display requirement” contain a lot of steps.

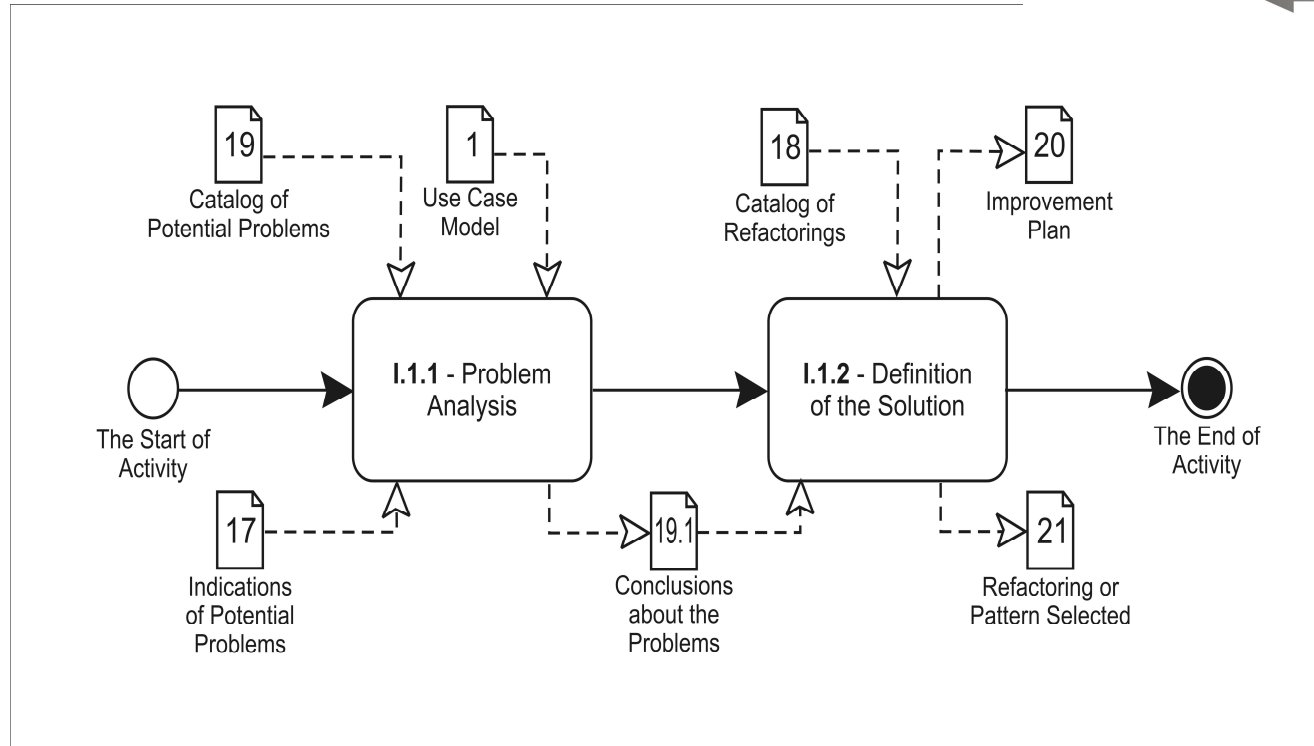
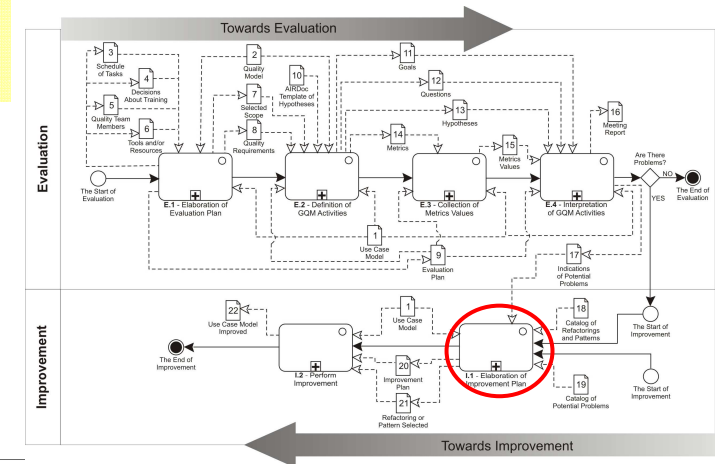
Create a Document to Indicate where the Worst Results were Found

Potential Problem	Localization
<i><indicate the type of the problem></i>	<i><indicate the name of use case(s) and, if necessary, the specific step(s)></i>
The two use cases that describe the “display requirement” contain a lot of steps.	Use case 1 – “Display spreadsheet control”. Use case 2 – “Display screen of user analysis”. Note: All steps of both use cases describe the “display requirement”

The AIRDoc



Elaboration of Improvement Plan



I.1 - Elaboration of Improvement Plan

Large Use Case Problem

Context

Large Use Case occurs when (i) a use case is trying to handle several different requirements at the same time or (ii) there are many alternative flows and steps.

This problem is particularly significant when the maximum size of each use case has already been set by the organization's Software Quality Assurance Team.

Possible Solutions

Use the Extract Use Case refactoring [Ramos et al., 2007c] to extract information related to a given concern and insert it into a new use case. This operation could be repeated for each major concern addressed by this large use case. This solution needs to be analyzed with caution, because it may increase the number of the use cases. To solve the problem of the increase of the use cases number, the *Package Use Cases* refactoring could be applied.

If the flows or other components of a use case could be moved to another use case, the *Move Activity* refactoring [Ramos et al., 2007c] could be used.

After extracting or relocating requirements, we sometimes need to rename the use case to better express the intention of the newly created one or of the one that was modified. In this case, the *Rename Use Case* refactoring [Ramos et al., 2007c] could be used to provide more appropriated names.

This refactoring opportunity is particularly important when there is a limit on the size of each use case, set by the organization's Software Quality Assurance Team.

Another possible solution is to use the *Extract Early Aspectual Use Case* refactoring [Ramos et al., 2008a]. This solution employs aspect-oriented requirements engineering and may be a favorable option if the requirements engineer desires to work with Aspect-Oriented Development of Software.

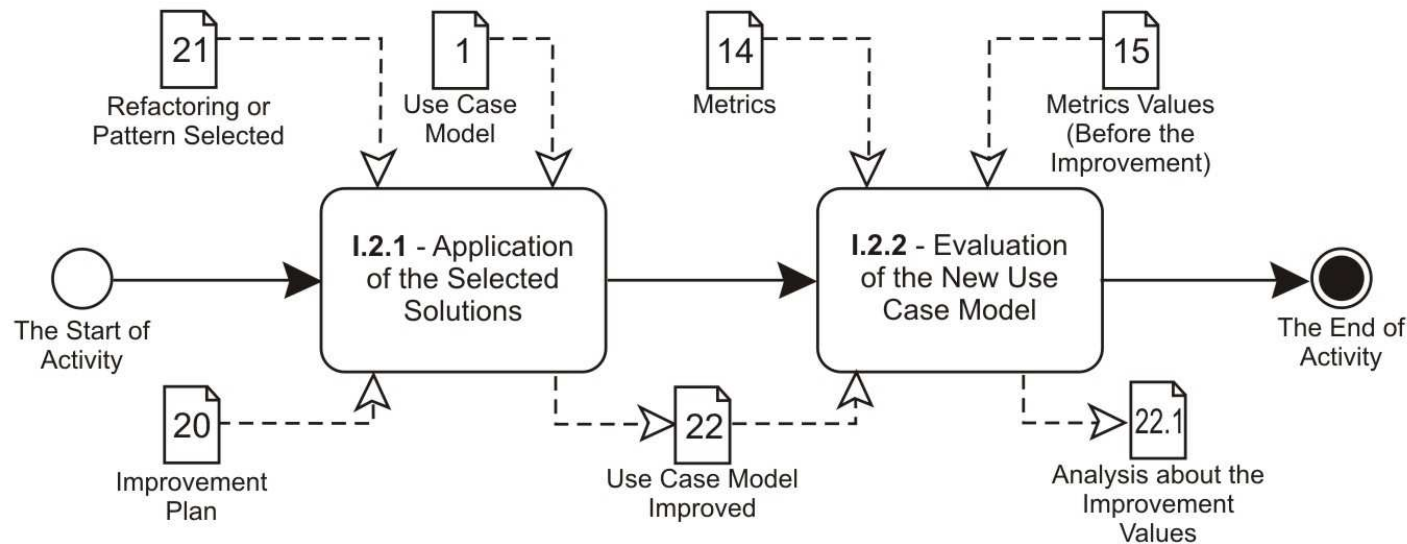
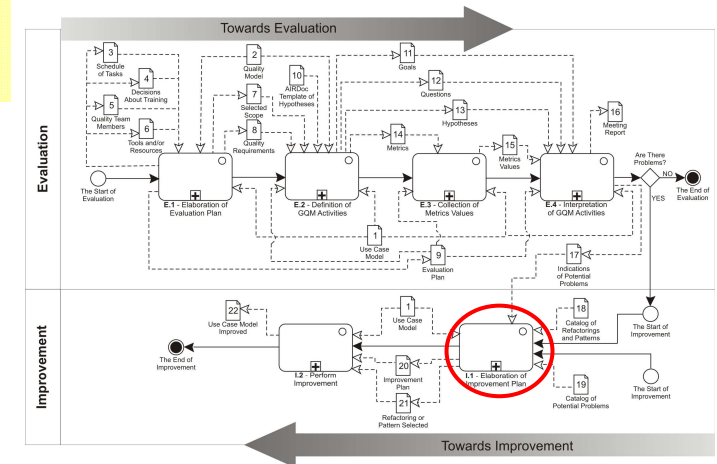
Problem Analysis

The Potential Problem	Selected Solution	Analysis of Cost and benefit
<i><name of the potential problem in agreement the catalog of Potential Problems ></i>	<i><list of the refactorings to solve the potential problems></i>	<i><describe the possible cost and benefits envisage with the application of the refactorings></i>
Large Use Case	Extract Use Case Package Use Case	The selected solution will have the cost of rearrange the use cases that describe the display requirement with the intention of decrease the size of it. We infer that this rearrangement will benefit the maintainability of this requirement.

Extract Use Case

Context	A set of inter-related information is used in several places or could be better modularized in a separate use case. Alternatively a use case description is too large or contains information related to a concern that is scattered across several use cases or is tangled with other concerns.
Solution	Extract the information to a new use case and name it according to the context.
Motivation	This refactoring should be applied when there are large use cases descriptions that can be split into two or more new use case(s). These large use cases include a great deal of information that is difficult to understand. Furthermore, it is not easy to locate the needed information quickly [Alexander and Stevens 2002], [Sommerville, 1997].

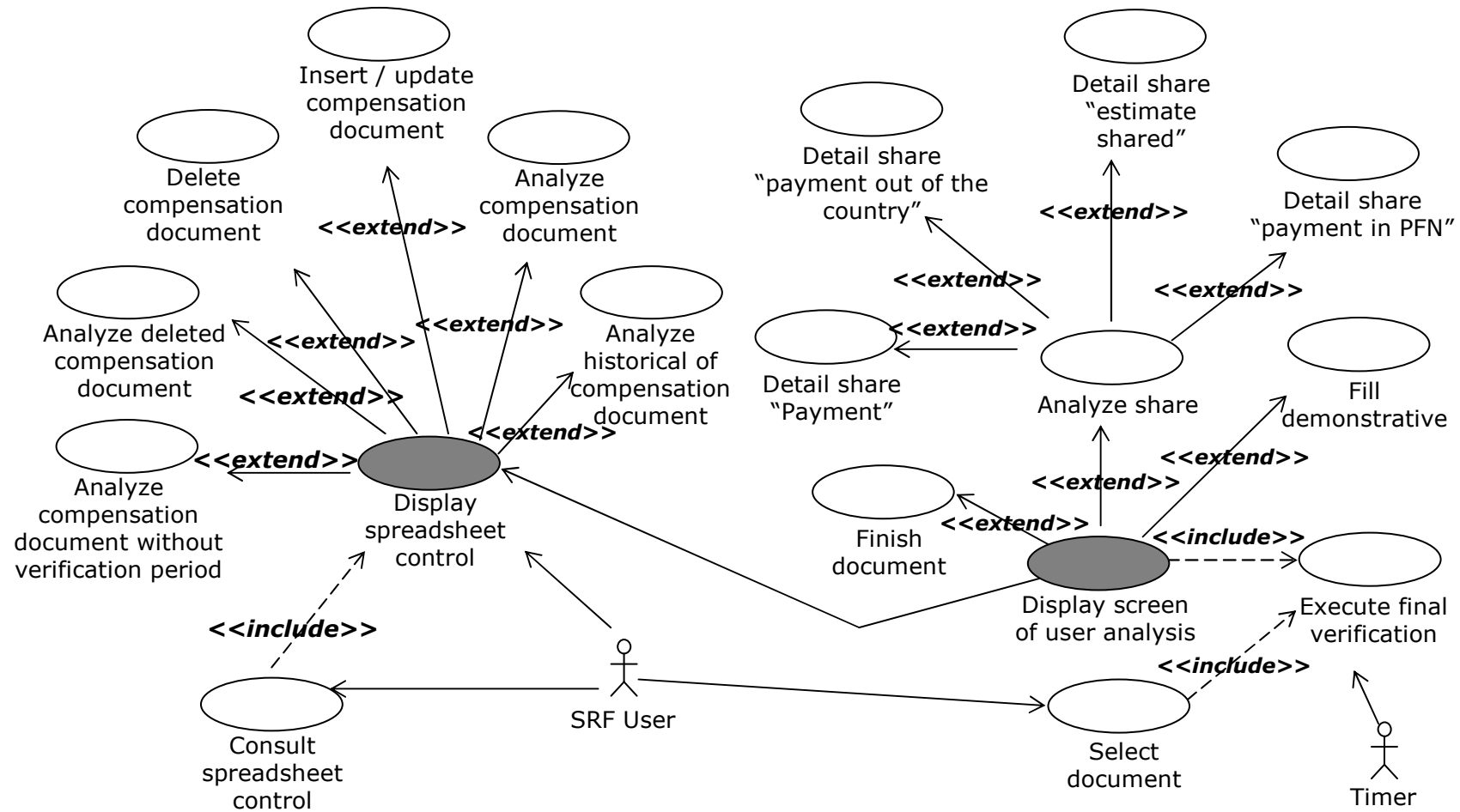
Perform Improvement



I.2 - Perform Improvement

Use Case Model After the Improvement

I) Apply the solutions selected

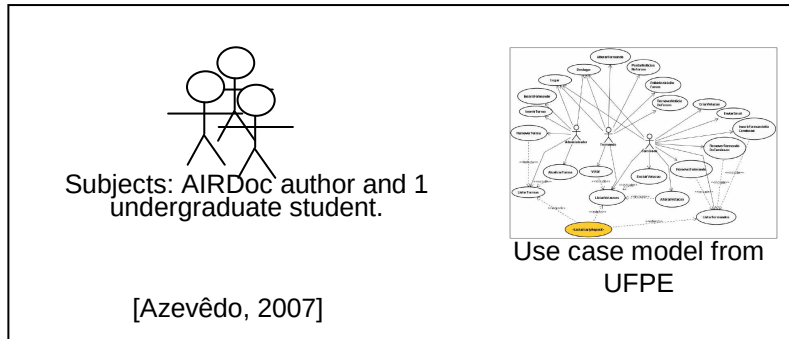


The Extract Use Case Refactoring was applied

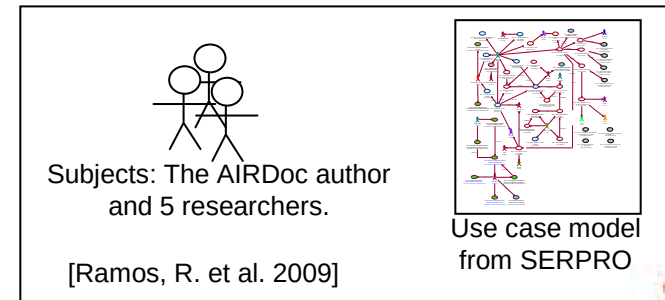
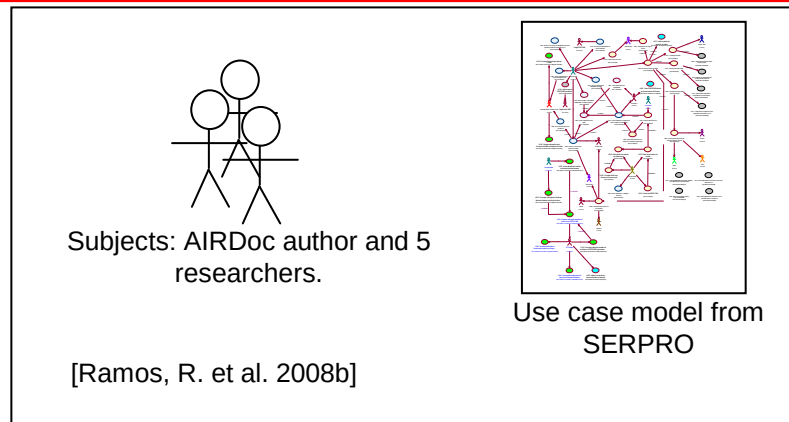
An Experimental Validation of the AIRDoc Process

Quantitative Studies
and
Qualitative Studies

Quantitative Studies



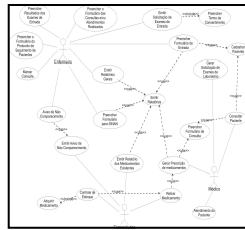
Applying an aspect oriented solution



Qualitative Studies

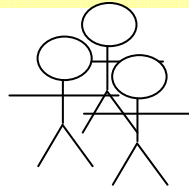
(1)

Qualitative evaluation



Use case model
from Juazeiro
Hospital

Quality: Feasibility



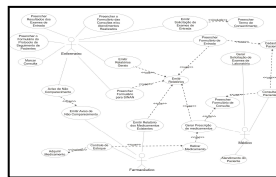
Subjects: 18 undergraduate students
and 2 professional developers.

Groups A, B, C, D, E and F - each one
composed by 3 undergraduate students.

Group G - composed by 2 professional
developers.

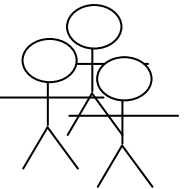
(2)

Comparative evaluation AIRDoc Vs. AD-HOC



Use case model
from Juazeiro
Hospital

Time spent, number
of problems identified
and used solutions.



Subjects: 24 undergraduate students and
4 professional developers.

Group H and I - each one composed by 3
undergraduate students.

Group J - composed by 2 professional
developers.

Qualitative Evaluation (1) - Hypotheses

- **H1** - the AIRDoc process *is acceptable* to improve a use case model, from the feasibility viewpoint.
- **H0** - the AIRDoc process *is not acceptable* to improve a use case model, from the feasibility viewpoint.

Conclusions about the Quality Evaluation

- Although some groups made more and others less refactoring, all groups ended up with very similar final use cases models and were able to improve the initial use case model. This may give us some clue that the AIRDoc process helped to improve the use case model.
- However, everyone had the same training. This may have influenced the results.

Comparative Evaluation (2) - Hypotheses

- **H1:** The AIRDoc is *better* than the ad-hoc method to improve the quality of use cases models.
- **H0:** The AIRDoc is *worse* than the ad-hoc method to improve the quality of use cases models.

Conclusions about the Comparative Evaluation

- The ad-hoc approach restructured the use case model based on the experience of the subjects. Thus, the quality improvement is totally dependent on the subjects. On the other hand, the AIRDoc process assists any type of subjects.
- The time spent using the AIRDoc process may be reduced if the subjects know the domain of the use case model and they have experience in the application of metrics to use cases.

Contributions

- We proposed a process to perform the evaluation and improvement in Use Case models.
- This process is based on GQM [Basili et al., 1994] and complies with the IEEE Standard for a Software Quality Methodology [IEEE 1061, 1998] and with the IEEE Recommended Practice for Software Requirements Specifications [IEEE 830, 1998].

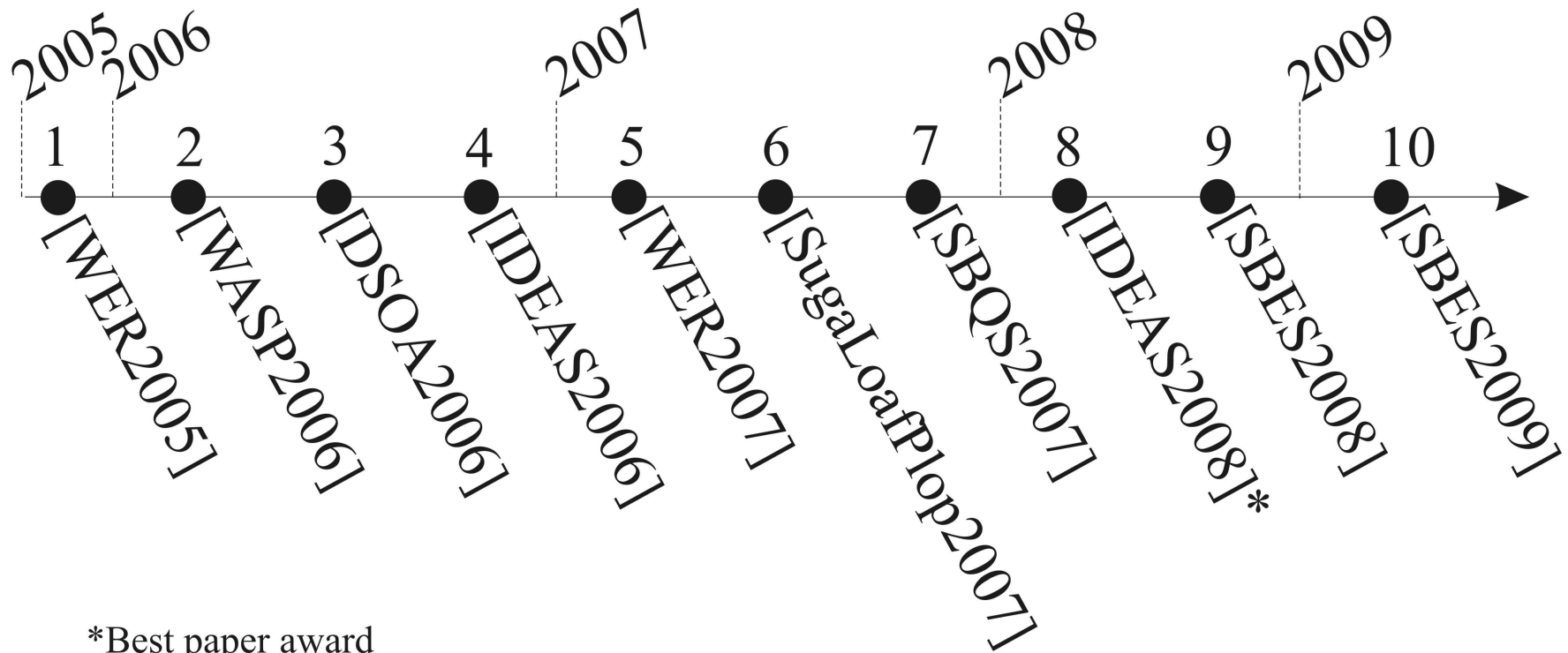
Contributions

- The AIRDoc process includes a catalog of known problems which may help to better categorize the potential problems. It also provides a refactorings catalog which to can assist the user to improve the use case model quality.

Future Works

- To implement a semi-automatic measurement mechanism to speed up its use and analysis
- To extend the AIRDoc process to evaluate and improve other modeling techniques, such as i* models [Yu, 1995] and models generated by the OO-method
- Replication of the experimental studies with professional developers that work specifically with UML development.

Related Publications



*Best paper award

Other 20 publications (in the period 2005 to 2009) were not reported here because they do not directly deal with the AIRDoc development. Those publications include: 3 books, 1 book chapter and 14 more publications in national and international conferences/workshops.