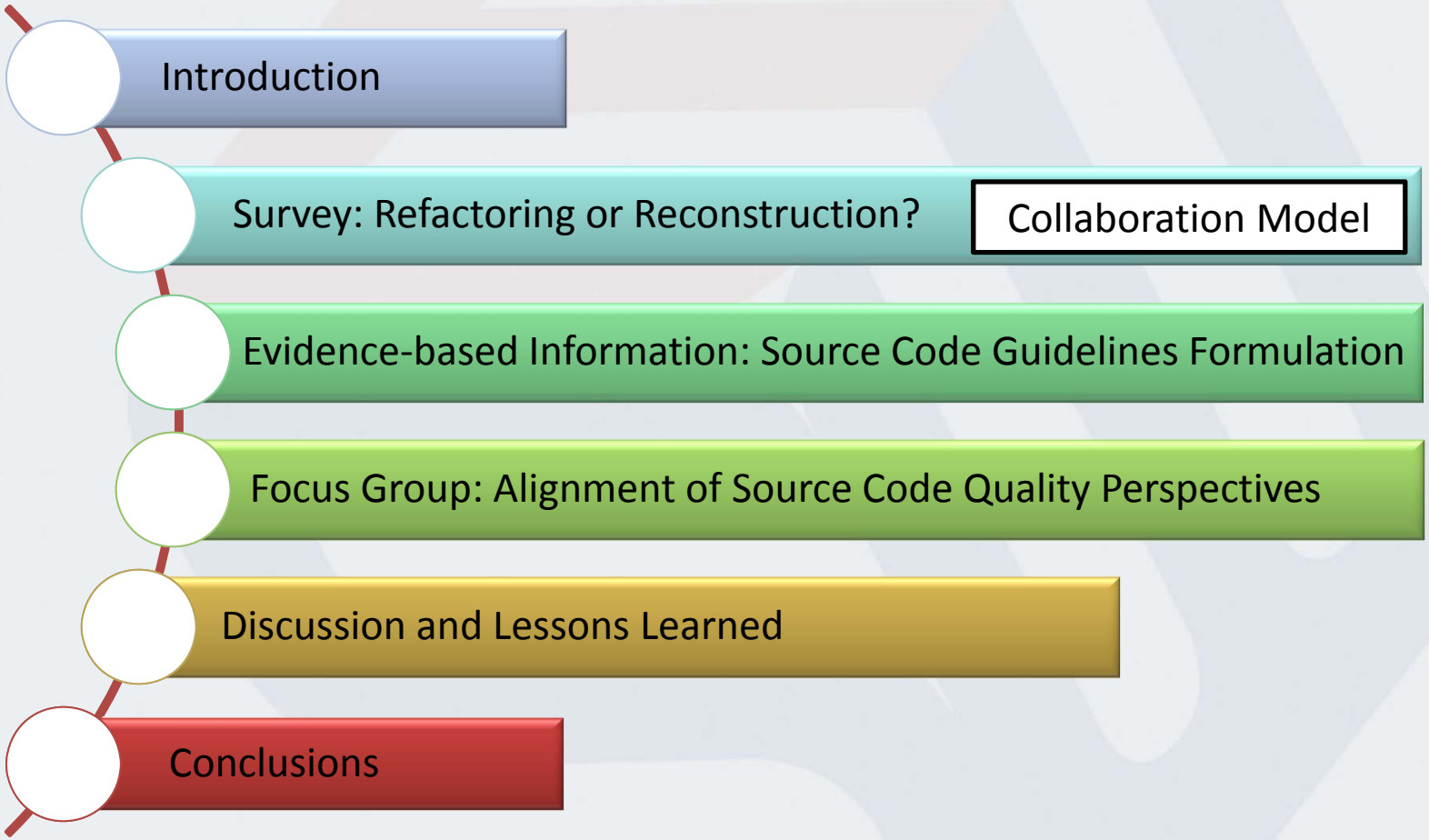
On the Alignment of Source Code Quality Perspectives through Experimentation: An Industrial Case

Talita Vieira Ribeiro

Guilherme Horta Travassos

Agenda


Introduction
Survey: Refactoring or Reconstruction?
Collaboration Model
Evidence-based Information: Source Code Guidelines Formulation
Focus Group: Alignment of Source Code Quality Perspectives
Discussion and Lessons Learned
Conclusions

Introduction

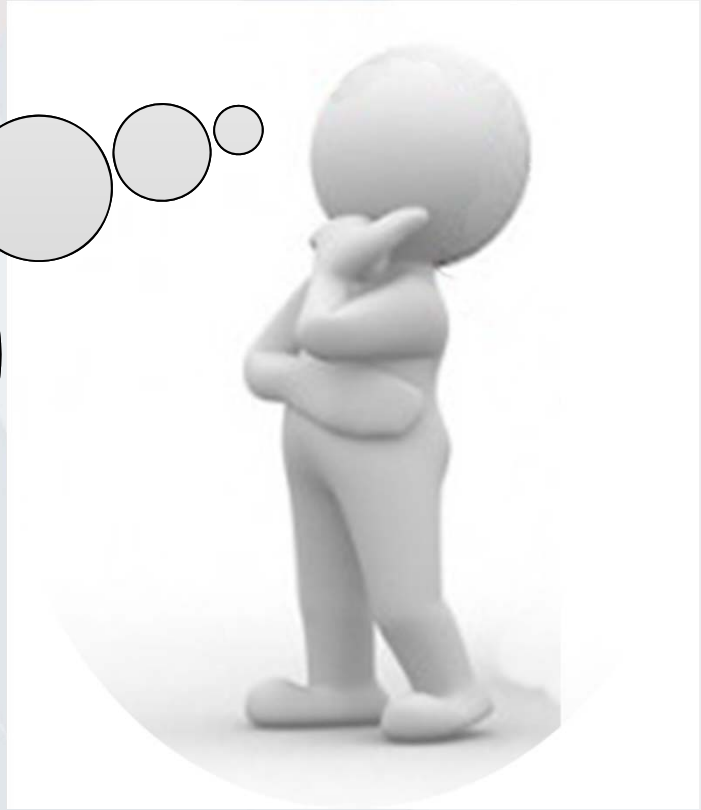
- Collaboration project between ESE Group and an embedded software development company in Brazil (Alpha Company);
- Too much rework due to repeated “refactoring” activities.

Introduction

What does source code quality mean for the developers?

What do they understand by “refactoring”?

What are the source code situations that make them observe the need for refactoring?



Survey

Refactoring or Reconstruction?

- Prior information that had to be taken into account for the decision of which empirical study strategy we should use:
  - i) different development teams are distributed;
  - ii) researchers and company are also apart – different states…
- Three distinct questionnaires were formulated to capture the answers to the questions we had:
  - developers' and projects' characteristics;
  - the importance of some product-based quality characteristics;
  - refactoring activities.

Why Survey 

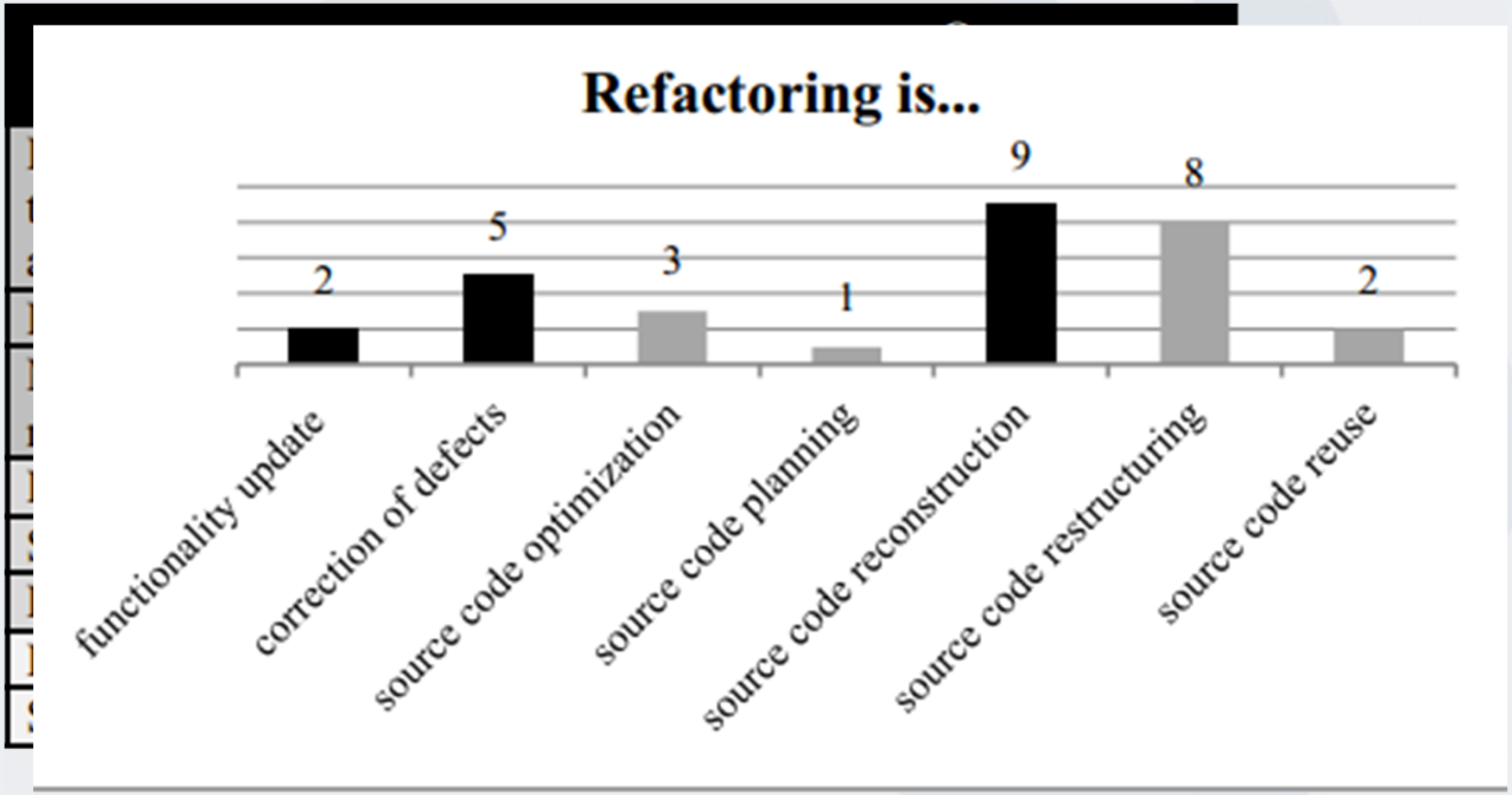
Survey

Refactoring or Reconstruction?

Problematic Situation	Occurrence Frequency
Low readability and/or understandability, that is, identifiers are located improperly and/or are not intuitive	ALWAYS
Difficulty in maintaining	
Method, functions and/or procedures with many responsibilities	
Difficulty in reusing	EVENTUALLY
Source code very fragmented	
Difficulty in extending functionalities	
Existence of legacy code	RARELY
Source code redundancy	

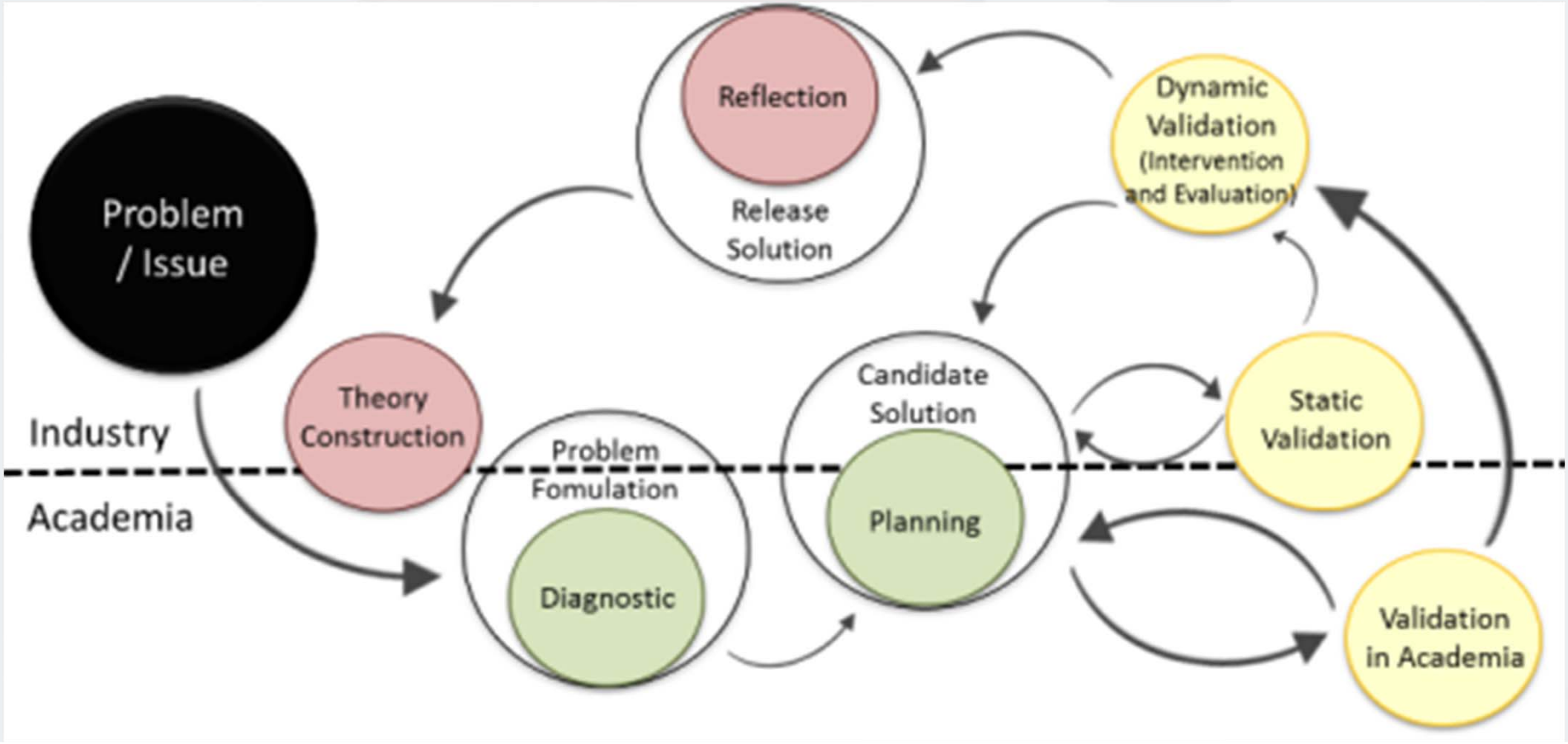
Survey

Refactoring or Reconstruction?



Unusual frequency for the realization of refactoring: **monthly; weekly; daily**.

Collaboration Model Designed



Based on:

T. Gorschek, C. Wohlin, P. Carre and S. Larsson, "A model for technology transfer in practice," *IEEE Softw.*, vol. 23, no. 6, pp. 88-95, Nov 2006.

P. S. M. dos Santos and G. H. Travassos, "Action research can swing the balance in experimental software engineering," *Advances in Computers,* vol. 83, pp. 205-276, May 2011.

Evidence-based Information:

Source Code Guidelines Formulation

Guidelines for Readability and Understandability: focusing on what was important for the company in terms of source code quality!

Evidence-based Information:
Source Code Guidelines Formulation

Guidelines for Readability and Understandability: focusing on what was important for the company in terms of source code quality!

- **Systematic Literature Review:**
  - RQ1 – Which attributes are used to evaluate source code **readability and understandability**?
  - RQ2 – What are the measurement procedures of these attributes?
  - RQ3 – What are the existing relations between the attributes identified and the source code quality characteristics?

TITLE-ABS-KEY((metric OR measure OR attribute OR predictor OR evaluation OR assessment OR improvement OR style OR standard OR pattern) AND (readability OR understandability OR understandability OR identifier OR naming OR comment) AND ("software quality" OR "software readability" OR "software comprehension" OR "software understanding" OR "program quality" OR "program readability" OR "program comprehension" OR "program understanding" OR "code quality" OR "code readability" OR "code comprehension" OR "code understanding"))

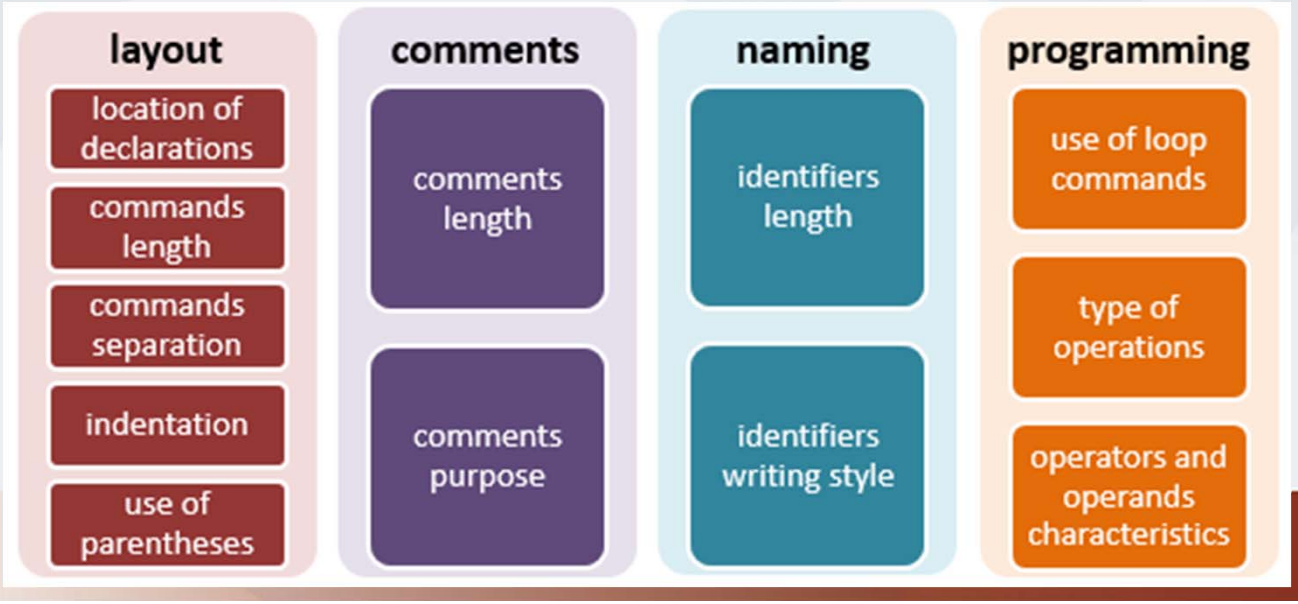
Scopus

236 articles returned;
18 were selected;
59 source code quality attributes.

Evidence-based Information:
Source Code Guidelines Formulation

Guidelines for Readability and Understandability: focusing on what was important for the company in terms of source code quality!

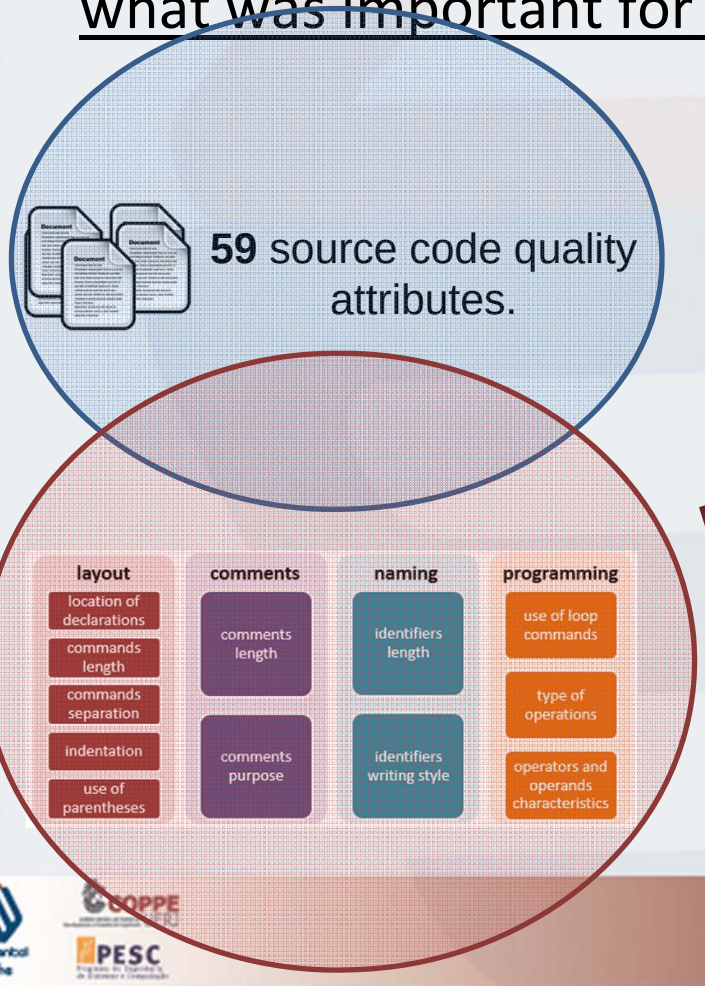
- **Gathering Alpha Company's Source Code Information:**
  - Assigned task: "Identify three source code snippets (one with high, another with low and another with medium readability and understandability) not written by you, but which you work with in your daily activities…"



Evidence-based Information:

Source Code Guidelines Formulation

Guidelines for Readability and Understandability: <u>focusing on what was important for the company in terms of source code quality!</u>



ID	CODING GUIDELINES – VERSION 1
	FILES AND FOLDERS
1	Use filename extensions according to the file contents
	LAYOUT
13	Variables must be declared together and at the beginning of their valid scope
	COMMENTS
17	Write a comment to explain the purpose of each function
	NAMING
21	Global constants and variables must be identified as such
	PROGRAMMING ORIENTATION
-	Avoid the presence of commented code no longer used

Focus Group

Alignment of Source Code Quality Perspectives

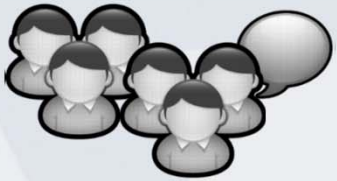
- We decided to plan and execute a FG with a fourfold purpose:
  - To characterize the applicability of 10 guidelines;
  - To start the internalization of the guidelines within the company (align the different quality perspectives);
  - To help on the exchange of experience and knowledge among the developers;
  - To gather as much information as possible to enable us to reformulate the guidelines or create new ones in case of need.

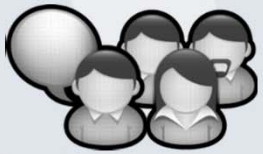
Why Focus Group 

Focus Group

Alignment of Source Code Quality Perspectives

It is not applicable for **embedded software**, because...	It is applicable for **embedded software** and...			**Coding Guidelines**	It is not applicable for **software web**, because...	It is applicable for **software web** and...		
	It does not bring benefits in cases which...	It helps in cases which...	It would be of more help if...			It does not bring benefits in cases which...	It helps in cases which...	It would be of more help if...
				<name of a CG>				
				<name of another CG>				





Talita Ribeiro
Guilherme Travassos

ID	CODING GUIDELINES – FINAL VERSION
	FILES AND FOLDERS
1	Organize cohesively folders and project files
2	Use filename extensions according to the file contents
3	Each module must be compound of at least one definition and one implementation file
4	Use the following order to organize definition files
5	Use the following order to organize implementation files
	LAYOUT
6	The indentation must be consistent
7	Use indentation in logical expressions with two or more operators not equal
8	The use of curly brackets for blocks identification must be consistent (2TBS style)
9	Pay attention to the use of consistent spacing (1 space) among different syntactic structures and program elements
10	Use parentheses to enclose operands in expressions
11	Use blank lines to separate extensive statements of other statements and to separate blocks of statements with different purposes
12	Each line must have only one declaration
13	Each line must have only one simple statement
14	Line length must be up to 80 characters
15	Variables must be declared together and at the beginning of their valid scope
16	Constants must be declared together and at the beginning of their valid scope
	COMMENTS
17	Select a language (English or Portuguese) to write the comments
18	Write an identification comment (header) for each file
19	Write a comment to explain the purpose of each function
20	Line comments must be used only in specific cases
	NAMING
21	Select a language (English or Portuguese) to write the identifiers
22	The identifiers must be easy to be remembered
23	Pay attention to the proper use of prefixes and suffixes in program elements
24	The program elements identifiers must be consistent in theirs style
25	The constants must be symbolic
	PROGRAMMING ORIENTATION
-	Avoid the use of "go to"
-	Avoid the use of ternary operation ("?") when the statement length exceed the stated line length
-	Avoid the presence of commented code no longer used
-	Pay attention to the code solution size given to the problem (in blocks, functions and files). This might be an indication that either the problem is not well structured or the structural complexity of the solution is high

Discussion and Lessons Learned

Discussion and Lessons Learned

- *General Challenges?*
  - Confidentiality issue.
- *Industry Challenges?*
  - The leader of the development teams was committed to the collaboration project, but the teams had their own manager and not all of them were opened to changes.
- *Academic Challenges?*
  - Anxiety for quick solutions.
- *Research Challenges?*
  - Selection of suitable research methods.

Conclusions

- Industry-academia collaboration guided by different empirical studies;
- Apparent triviality of the issues identified in the company → industry still faces problems which academia argues have been solved;

Are available software technologies applicable for companies?

Do companies see the importance of some disseminated practices for the software process development?

On the Alignment of Source Code Quality Perspectives through Experimentation: An Industrial Case

Talita Vieira Ribeiro

Guilherme Horta Travassos