Concept for a qualitative content analysis of scientific articles gathered from a Systematic Literature Review

LVA: IK Qualitative Forschungsmethoden 232.101
Dr. Peter Holtz WS 2015/16

Submitted by Johannes Bräuer, 0857361
johannes.braeuer@gmx.at
Mai 2016

Table of Content

Word count (exclusive table of content, list of figures, list of tables, references, and appendix): 8806

List of Figures

List of Tables

1 Introduction

Measuring object-oriented software design is an ongoing topic that has its roots back to 1994 when (Chidamber & Kemerer, 1994) and (Bansiya & Davis, 2002) proposed a metrics suite for measuring object-oriented design aspects. Based on this work, many authors continued to pursue the idea of expressing design with simple metrics and built their own measuring approaches therefore. In spite of solid results and successful achievements, it became obvious that software design is too complex and cannot be expressed by using simple metrics (Marinescu, 2004). As a consequence, new and more advanced measuring techniques established in order to provide better and more suitable support for software architects and engineers.

We – the project team for software quality at the software engineering institute - believe that design principles play a crucial role in building a good software design that has to fulfil quality characteristics. For measuring these design principles, checking metrics is not adequate because semantical aspects of the design are then completely ignored. Instead of relying on metrics, we defined rules (design best practices) that are associated with design principles. Thus, a rule violation may indicate a non-conformance of the design principle and to a design flaw. At this point, it is important to mention that we do not want to provide an additional measuring tool that just simple identifies rule violations that need to be fixed, we are eager to provide an approach that is capable to identify deeply-rooted design flaws.

When I was working on a section of my PhD thesis that is derived from this idea, doubt rose about the collected literature. The method for collecting this literature did not follow a defined process and was more or less an ad hoc literature review for getting familiar with the topic of measuring object-oriented software design. At this stage, I believe that I already identified papers which can be considered as fundamental work in this topic area. However, there is no guarantee that I captured all available approaches and also newer ones. For building my PhD thesis on a solid literature foundation and to support the research question of the thesis, I see the need for a systematic literature review (SLR).

Based on this reasoning, my advisor and I came to the agreement that a SLR is one of the next tasks for continuing the work on my PhD thesis. The aim of the SLR is to get a detailed overview of the current state for measuring and assessing object-oriented software design. The SLR follows the process suggested by (Kitchenham & Charters, 2007). This process describes the steps for performing a systematic literature review in the software engineering domain and is subdivided into four phases. Since the first two phases are clearly defined and straight forward, this seminar paper describes them in an overview. However, for the third and fourth phase – the data extraction and data synthesis phase, respectively – decisions for the qualitative content analysis must be defined within a concept.

The aim of this seminar paper is to formulize and to define this concept for qualitatively assessing the gathered articles. This concept must be well-conceived because the third and fourth phases of the SLR are time consuming and a repetition must be avoided. Although the results and findings from the gathered papers will be investigated for understanding the measuring approach, the main focus lies on understanding the ideas behind it. Consequently, the concept must be an analysis strategy that allows drawing meaningful conclusions. The work of (Mayring, 2010) provides support in this direction because he shows how to organize content in a structured manner.

The structure of this seminar paper is outlined as follows: Section 2 provides a brief overview of the systematic literature review process. Therefore, it shows the four phases of the process and the underlying research question (and guiding questions). In Section 3 the data extraction phase is discussed in more detail since this is the phase in which the concept will be applied. For testing the coding system, a pilot study on ten primary papers has been conducted. The qualitative discussion of the result is shown in Section 5. Lastly, Section 6 summarizes the lessons learned and provides an outlook on future work.

2 Systematic Literature Review

In order to extend the introduction and to explain the goal of the SLR in more detail, this section highlights the general process, the research questions and the current state.

2.1 Overview of the SLR Process

The SLR process aims to identify approaches for measuring software design by considering the entire literature that is currently available. The process contains the main steps that are required to conduct a systematic literature review as proposed by the guidelines of (Kitchenham & Charters, 2007). According to Figure 1, the process is divided into four different phases containing multiple steps.

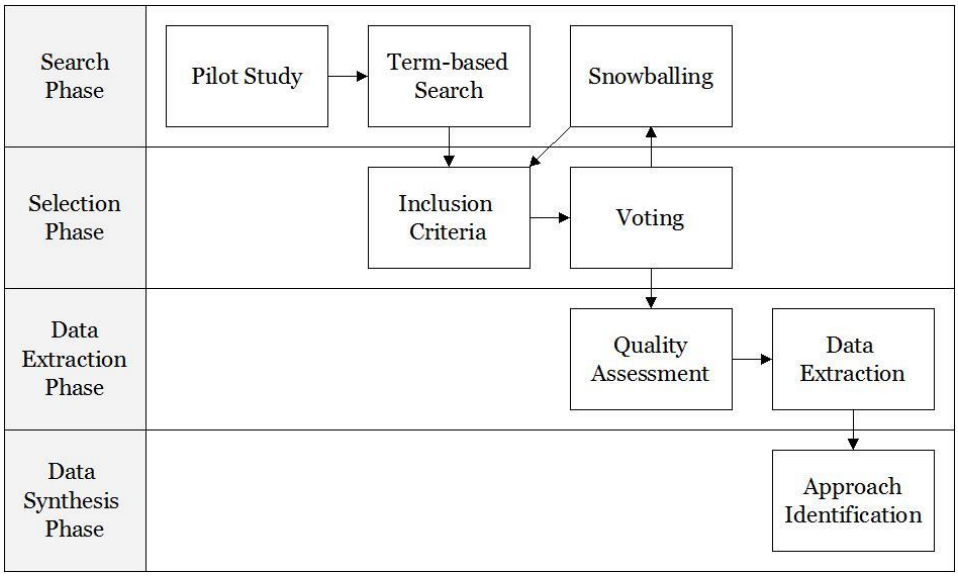

Figure 1: Overview of the SLR Process

The first task is a pilot study which needs to be carried out in the search phase. This pilot study has to follow a pre-defined protocol and tries to identify weaknesses in it. For example, the search terms are verified for their practical relevance and access to digital libraries is checked. The result of the pilot study is a new version of the protocol that exactly describes the characteristics of this SLR. With an updated version of the protocol, the search of papers can start.

To find potential papers for the SLR, different scientific databases are queried by using a set of suitable search terms. Most of the time, these database queries return a huge number of papers why it is necessary to identify totally irrelevant papers obtained from the term-based search. Therefore, the selection process uses formal inclusion and exclusion criteria. Moreover, in the voting step of the selection phase the remaining papers are rated by at least two individual researchers according to the paper title and abstract. This step also

identifies high value (hot) papers, which are used as origin, for the snowballing search. The snowballing search is a forward and backward search that aims to identify papers that are not covered by the term-based search but of high relevance. Nevertheless, the new papers must pass the inclusion criteria as well.

After completing the search phase and selection phase, all remaining papers must be read in detail. This is necessary for assessing the quality of the paper and for extracting the required information about the different approaches. Since this seminar paper defines the concept for the data extraction and synthesis phase, Section 3 and 4 provide more details therefore.

2.2 Research Questions of the SLR

From a general viewpoint, the SLR should reveal an answer to the following research question (RQ): *Is the research area of assessing object-oriented design ready to move on, or should we critically rethink the proposed approaches?*

In order to answer the RQ of this SLR, it is necessary to identify existing approaches for measuring and/or assessing object-oriented software design. Additionally, we want to figure out which approaches are addressing design principles or object-oriented language features and whether there is a trend showing that metric-based approaches have been becoming less relevant. Since it is important to highlight the advantage of an approach, we are interesting in the intention of measuring and assessing design as well as the perceived suitability. Accordingly, the SLR has been aligned to following guiding questions (GQ):

- Research Interest 1: Design Measuring Approaches
    - **GQ1.1**: Which approach is used for the design quality measurement?
    - **GQ1.2**: What is the target of the design quality measurement?
- Research Interest 2: Design Assessment Approaches
    - **GQ2.1**: Which approach is used for the design quality assessment?
    - **GQ2.2**: What is the target of the design quality assessment?
- **GQ 3**: How important are design principles for measuring/assessing object-oriented software design?
- **GQ 4**: Which approaches are explicitly dealing with object-oriented features of software design?
- **GQ 5**: Is there a trend showing that (pure) metric-based measuring/ assessing approaches are fading into the background?
- **GQ 6**: What is the underlying purpose of measuring/assessing object-oriented software design? (intention behind the approach)
- Research Interest 7: Perceived Suitability
    - **GQ 7.1**: To which extent are the measuring/assessing approaches suitable within the applied validation context?
    - **GQ 7.2**: To which extent do the measuring/assessing approaches confirm their intention?
- **GQ 8**: Which application purpose of quality models is addressed by the proposed approach?

Research interest 1 is subdivided into **GQ 1.1** and **GQ 1.2** and aims on identifying approaches and groups of approaches to measure object-oriented design. Measuring object-oriented design means that source code or software artifacts are used to calculate design related software characteristics. **GQ 2.1** and **GQ 2.2** ask the same question but from the viewpoint of assessing design. Consequently, we are eager to identify groups of ideas that are bringing measuring results in relation and for deriving design improvements. The research interest 7 deals with the suitability of a particular approach. As this work argues about the benefits and drawbacks of proposed approaches, it is important to identify indicators for the strengths and weaknesses of an approach. This is addressed by **GQ 7.1** and **GQ 7.2**.

Even though **GQ 6** already addresses the main purpose of the proposed approach, we added **GQ 8** to the investigation that is based on the work of (Kläs, Heidrich, Münch, & Trendowicz, 2009). There the authors discuss a classification schema for quality models and identified six application purposes of quality models that are: to specify, measure (monitor), assess (control), improve, manage, and estimate (predict) the target of interest. This SLR does not explicitly deal with (design) quality model, but the proposed approaches address certain aspects of the six application purposes. Consequently, we want to figure out which of the six purposes is well covered and where are still white spots.

2.3 Current State of the SLR

At the current state of the SLR, the voting is completed meaning that the quality assessment is coming next as depicted in Figure 1. For briefly summarizing the progress till the current state, the term-based search represents the starting point. In order to find potential papers for the SLR, we queried the following online scientific databases that are dominating the software engineering domain: IEEExplore, ACM, ScienceDirect, and SpringerLink. For querying these databases, we defined search strings composed of pre-defined search terms. Figure 2 presents such a search string and shows that we were searching for papers containing a search term in the title or abstract.

```
(
(p_Title:"measurement" OR "Abstract":"measurement") OR
(p_Title:"measuring" OR "Abstract":"measuring")  OR
(p_Title:"measure" OR "Abstract":"measure")
)
AND
(p_Title:software design OR "Abstract":software design)
AND
(
(p_Title:"design principle" OR "Abstract":"design principle") OR
(p_Title:"design principles" OR "Abstract":"design principles")
)
```

Figure 2: Search String for Term-based Search

Unfortunately, it was not possible to build one search string that logically connects all search terms because the online databases restrict the number of search terms per query. Consequently, we had to trigger the search strings individually. Next to the overhead of effort that was necessary due to this limitation, duplicated entries hat to be removed too. Nevertheless, the term-based search followed by the application of formal inclusion and exclusion criteria returned **327** potential articles.

This set of articles was then handed over to the voting process of the SLR which aims to reduce this list to those papers that can provide a contribution to the research question.

Therefore, two researchers read the title and abstract of each paper in order to assess the potential contribution. Since this assessment is not trivial and can depend on someone's opinion, we defined and followed inclusion and exclusion criteria. For including a paper to the final set, it must fulfil at least on inclusion criteria, which are RQ-related. However, when a paper also fulfills at least on exclusion criteria then it will be removed from the final set.

After the first voting round of the initial set of 327 primary papers, we came to the result summarized in the following table:

Table 1: Result of Voting Procedure

		Johannes		
		Yes	No	Total
Reinhold	Yes	87	31	*118*
	No	15	194	*209*
		102	*225*	*327*

Agreement:	87	194	*281*
By Chance:	36.81	143.81	*181.4*

Cohen's Kappa:	**0.68**

According to (Landis & Koch, 1977) the Cohen's Kappa value of 0.68 reaches a substantial agreement that fulfils our requirements for accepting the assessment. This decisions is underlined by the opinions of (Bortz & Döring, 2007) and (Greve & Wentura, 1997) who consider a value between 0.60 and 0.75 as reasonable.

Table 1 depicts that the two reviewers had a different opinion on 46 (31 + 15) publications. Consequently, they were reconsidered based on the RQ-related inclusion and exclusion criteria. After fixing these conflicts, 20 papers passed the voting procedure and 26 papers were rejected due to their weak contribution to the research question. In total, 122 (102 + 20) papers passed the voting step.

3 Data Extraction Phase

The data extraction is performed for the final set of articles that has been selected through the search and selection process described above. Aim of this phase is to qualitatively assess the remaining papers. In other words, some papers may be promising based on their title and abstract, but they do not provide a meaningful contribution to the research question. Consequently, this data extraction phase has to provide deeper inside into the quality of the papers, which is necessary:

- to provide still more detailed inclusion/exclusion criteria,
- to investigate whether quality differences provide an explanation for differences in study results,
- as a means of weighting the importance of individual studies when results are being synthesized,
- to guide the interpretation of findings and determine the strength of inferences, and
- to guide recommendations for further research.

3.1 Paper Quality Assessment

Although the voting process should have identified papers that are important for this SLR, an explicit quality assessment is needed to assist the primary study selection on the one hand and to assist the data analysis and synthesis on the other hand (Kitchenham & Charters, 2007). Therefore, it is necessary to read or at least to skim all papers for getting an understanding of their importance for the SLR. However, to ensure the reliability of this SLR, it is recommended to define a quality assessment instrument that captures meta-data of the papers and the assessments of reviewers (Kitchenham & Charters, 2007).

For building a quality assessment instrument, we defined a list of quality criteria (QC) and questions related to the GQs as shown in Table 2. These questions were derived from the GQ as indicated by the identifier in the second column. The list of possible answers was pre-defined and became extended when there was the need therefore. For assessing the scientific strength of the publications, we added the three general characteristics of scientific research, which are: objectivity, validity, and reliability. These three properties need to evaluated by using a three-point scale: “not addressed”, “partially addressed”, and “fully addressed”. To assign an assessment to these properties, the reviewers have to figure out how well the authors compare their work with others, how well the authors describe the validation process, and whether threats do validity are discussed or not.

Table 2: Quality criteria and questions for data extraction

Classification/Question	**GQ**
Conference or Journal the paper was published	QC
Year the paper was published	GQ 5
# of Citations (reported by Google Scholar)	QC
Which approach is used for the design quality measurement? - Manual approach - Machine learning-based approach - Metric-based approach - Rule-based approach - Symptom-based approach - Probabilistic approach - Visualization-based approach	GQ 1.1
Which approach is used for the design quality assessment? - Aggregation/weighting function - Benchmark-based approach - Learning method - Manual approach	GQ 2.1
What is the target of the design quality measurement/assessment? - Design in general - Technical property - -ility aspect - Design principle	GQ 1.2 GQ 2.2
How important are design principles for measuring/assessing object-oriented software design? - Design principle in a broader sense	GQ 3 GQ 4

- Design principle in a narrow sense - OO-Features - no Principle, nor OO-Feature	
What is the underlying purpose of measuring/assessing object-oriented software design? (intention behind the approach)	GQ 6
Which application purpose of quality models is addressed by the proposed approach? - Specify - Measure/Monitor - Assess/Control - Improve - Manage - Estimate/Predict	GQ 8
Is related work considered and compared to the proposed approach?	QC
Are threats to validity discussed?	QC
Do the authors describe the research design of the validation?	QC
What is the validation context? - Student Project, Example - Open Source Project (Case Study) - Industrial Project (Case Study) - Empirical Investigation	QC
How suitable is the approach within the validation context?	GQ 7.1
What is the overall validation result?	GQ 7.2

3.2 Data Extraction Strategy

After qualitatively assessing the primary papers, the next and major step of the SLR focuses on the data extraction that should follow a pre-defined process for being efficient and effective. The data extraction is time consuming because every paper must be read in detail and logical conclusions must be drawn based on the content of the paper and the understanding of the reviewers. For avoiding the situation that each paper must be read multiple times, this section defines a two-step process that has to be followed when analyzing the primary papers in detail.

As mentioned, the analysis of a primary paper consists of two steps. First, the paper has to be coded based on a pre-defined – but not limited – coding system while reading it the first time. Next to the research question related codes, this coding system contains codes that should yield interesting aspects in the domain of measuring object-oriented design. For instance, approaches for normalizing measuring results, which are not part of a research question, may disclose some insights in new ideas and concepts. For the second step of the analysis, the reviewer has to answer a questionnaire (form) which exclusively contains questions related to the guiding questions. Consequently, the form has to be completed when a design assessment or evaluation approach is discussed in the paper.

In the next part of this section the coding-system and the questionnaire are explained. For adequately applying the coding-system, a coding guideline is defined. This coding guideline

provides a definition, an example, and a coding rule for each code and explains how to extend the coding-system when an additional code is required. The questionnaire is described by the questions it contains and how they are related to the guiding questions of this SLR.

3.2.1 Coding System

For explaining the coding system, Tables 3 to 10 show the coding guidelines for seven code groups. As proposed by (Mayring, 2010), the guidelines provide the code name, a definition, and an anchor example that should clarify the context for assigning the code. Due to the hierarchical structure of the coding system, it is necessary to show how sub-codes are related to their super-code. To illustrate this hierarchy the "ʟ" indicates the next lower – and more specific - level of the hierarchy. The categories have been inductively determined by reading approximately 25 primary papers. Since a set of remaining papers could deal with another aspect uncovered by the categories, the list can be extended when needed.

Table 3: Coding Guideline for Measuring approaches

Code	Definition	Anchor Example
Measuring approach	-	-
ʟ Manual approach	A manual approach is a human-centric activity that aims on finding design defects based on analysis and interpretation from software maintainers (Ouni, Kessentini, & Sahraoui, 2014).	[…] "identifying and fixing these kinds of problems was an entirely manual operation. Knowledge of design patterns and code smells remained the best method for finding these problems within the system."
ʟ Machine learning-based approach	A machine learning-based approach uses a training set to adjust an algorithm before it is applied on the target source code.	"In this paper we use Naïve Bayers graph theory, to train our assessing approach for more accurate predictions."
ʟ Metric-based approach	A metric-based approach is calculating a single metric value for a certain design aspect based on design entities such as inheritance trees, class characteristics, or method calls.	"We use well known object oriented design quality metrics and find correlation among them to formulate a quality rank that is an indicator to the overall quality of any object oriented software."
ʟ Rule-based approach	A rule-based approach relies on a knowledge pool of design best practices and identifies violations of these best practices directly in the source code.	"Experts use a set of design best practices to check the compliance of design goals."
ʟ Symptom-based approach	A symptom-based approach does not use metrics or rules but rather follows descriptions of design flaws to find defects.	"The originality of DETEX stems from the ability for software engineers to specify smells at a high level of abstraction using a consistent vocabulary and domain-specific language for automatically generating detection algorithms."
ʟ Probabilistic approach	Probabilistic approaches extend the idea of measuring single metrics by using algorithms and	"The Analytical Hierarchy Process (AHP) is a structured technique to analyze the complex decision using pairwise comparisons."

		graph theory for deriving more meaningful results.	
	ʟ **Visualization-based approach**	A visualization approach uses visualization techniques to express frictions within a software design.	"Using this visualization tool, design flaws can be localized more easily and a starting point for the design investigation can be selected."
	ʟ **other**	There is another measuring approach not mentioned in the list above.	No example available.

Table 4: Coding Guideline for Assessment approaches

	Code	Definition	Anchor Example
	Assessment approach	-	-
	ʟ **Aggregation/weighting function approach**	An aggregation/weighting function approach uses a defined evaluation function to compose multiple evaluations to one single value.	"When the source code classification is finished, we have to measure the quality of code with Maintainability Index."
	ʟ **Learning method**	A machine learning-based approach uses a training set to adjust an algorithm before it is applied on the target software design.	"We will therefore use Xpose for the fit data set from which we build our prediction models."
	ʟ **Manual approach**	A manual assessment approach does not relay on tool support but rather on expert knowledge.	"In […], manual assessment is directed towards analyzing the results of design analysis tools. Faced with a large design artifact, experts get a better understanding of design problems when they examine tool results."
	ʟ **Benchmark-based approach**	A benchmark-based assessment approach compares a particular design assessment to a pool of assessments in benchmark base.	"This technique is used to derive the thresholds from a set of selected reference projects."
	ʟ **other**	There is another assessment approach not mentioned in the list above.	No example available.

Table 5: Coding Guideline Target of a Measurement/ an Assessment

	Code	Definition	Anchor Example
	Target	-	-
	ʟ **Design in general**	The design measuring or assessing approach addresses design in general.	"We select […] to meet our purpose to formulate a general rule for assigning a quality rank as an indicator of overall design quality of any object-oriented software."

L Technical property	The design measuring or assessing approach addresses a technical property such as: coupling, cohesion, complexity, or modularity.	"This approach can increase High cohesion and Low coupling that effect to the internal and the external of software."
L -ility	The design measuring or assessing approach addresses an –ility aspect such as: maintainability, reliability, understandability, etc.	"It will address some major criteria of a quality 00 design; and will seek to map object oriented metrics to quality factors such as reliability, complexity and reusability."
L Design principle	The design measuring or assessing approach addresses a desing principle (in a narrow sense).	"Design Principles-Based View: This view shows the causal relationship between a design problem and possible violations of design principles."

Table 6: Coding Guideline for Design Principle Codes

Code	Definition	Anchor Example
Design Principles	-	-
L Design principles (in a broader sense)	Design principles in a broader sense are: Coupling, Cohesion, Complexity/Size, and Encapsulation.	"Low inter module coupling, high cohesion and low complexity have always been deemed to be important attributes of object OO software systems."
L Design principles (in a narrow sense)	Design principles in a narrow sense are, e.g.: Information Hiding, Open Closed Principle, Single Responsibility Principle, etc.	"Design Principles-Based View: This view shows the causal relationship between a design problem and possible violations of design principles."
L OO-Features	OO-Features are: Abstraction, Inheritance, Polymorphism, and Encapsulation.	"Based on this idea, we use the measures to check the encapsulation of packages […]."
L no Principle, nor OO-Feature	This covers design metrics or other measurable artifacts.	No example available.

Table 7: Coding Guideline for Intention Codes

Code	Definition	Anchor Example
Purpose / Intention	-	-
L Improving software/design quality	The proposed approach aims on improving software or design quality.	"The third section is source code improvement with refactoring."
L Detecting (code/design) smells	The proposed approach aims on detecting code or design smells.	"DECOR, a method that embodies and defines all the steps necessary for the specification and detection of code and design smells."
L Detecting error proneness	The proposed approach aims on detecting error	"Thus, the benefit of using the fault-proneness model is

	proneness of a software design.	proportional to the number of faults it detects above what the size-based model can find."
L Predicting (code/design) smell evolution	The proposed approach aims on predicting the evolution of code or design smells.	"This paper aims to give support hints for the evaluation of the code and design quality of a system and in particular we suggest to use metrics computation and antipatterns detection together."
L Predicting quality assurance	The proposed approach aims on predicting the quality assurance of a software design.	Currently, no anchor example available.
L Measuring/evaluating an –ility aspect	The proposed approach aims on measuring or assessing an –ility aspect such as: maintainability, reliability, understandability, etc.	"Presenting the relation between these measurable designs attributes as independent variables and the quality attributes (-ility attributes) as dependent variables."
L Measuring/evaluating design in general	The proposed approach aims on measuring or assessing software design in general.	"We look into measuring the quality of Object Oriented designs during both software development and re-development processes."

Table 8: Coding Guideline for Quality Characteristics Codes

Code	Definition	Anchor Example
Quality characteristics	-	-
L Objectivity	The objectivity focuses on the related work considered in the paper.	"In this [related work] section we delve into each type and discuss whether and to what extent these methodologies can be used to realize the insights from our survey."
L Validity	The validity focuses on the discussion of threats to validity.	"The work is threaten by the following validation concerns."
L Reliability	The reliability focuses on aspects of the research design and research method of the validation.	"Xpose was implemented under Sun's JDK 1.2, using the Swing library classes for the GUI and a prefabricated XML Parser2 also developed at Oracle. Xpose consists of 144 Java classes, with a total of 1,774 methods. Some of the 144 classes were reused with modifications."

Table 9: Coding Guideline for Validation Context Codes

Code	Definition	Anchor Example
Validation Context	-	-
L Student project or example	A student project is part of a course work of students and junior software engineers.	"An example has been selected from many programs with which we have experimented to illustrate the use of the Metric Facility in

		RA.”
⌊ **Open-source project**	An open-source project is a project developed by an open-source community with public access to the source code.	“We tested our proposed formula on several open source software systems of different levels of design quality and validate by comparing the test result with expected levels of design quality.”
⌊ **Industrial project**	An industrial project is developed within an organizational context and by senior engineers. Such a project has practical relevance and is applied in organizational tasks.	“MIDAS has been successfully used to steer design assessment in three separate CT DC AA projects.”

Table 10: Coding Guideline for perceived Suitability Codes

Code	Definition	Anchor Example
(perceived) Suitability	-	-
⌊ **high**	The approach proposed in the paper perceives a high suitability in the applied context.	“Feedback from our MIDAS evaluation shows that using principles and constraints to report violations are very useful.”
⌊ **moderate**	The approach proposed in the paper perceives a moderate suitability in the applied context.	“[…]the important role that experts play in the MIDAS method. The second issue was the lack of effective and powerful tools (that would report more relevant issues, report lesser number of false positives, etc.) due to which a larger investment was needed in manual reviews.”
⌊ **low**	The approach proposed in the paper perceives a low suitability in the applied context.	Currently, no anchor example available.
⌊ **N/A – not available**	No statement regarding this concern.	No example available.

3.2.2 Questionnaire

Exclusively for those papers that are categorized as measuring approach or assessing approach, an additional questionnaire has to be completed next to the application of the coding system. This questionnaire aims revealing answers and deeper insight regarding the guiding questions and will be used to better draw conclusions. The structure of the questionnaire is divided into eight groups as shown in Table 9. This table also depicts the questions and considerations for each group.

Table 11: Data Extraction Questions

#	Label	Questions / Considerations
D1	Executive Summary	• Paper Title and Reviewer • Describe the research context of the paper (industry, academic, product, etc.)? • What is presented in the paper (approach, concept, idea, etc.)? • Relevance of the approach (research, practice)? • Which application scenarios are mainly targeted by the approach? • Notes to the approach.
D2	Measuring / Assessment Approach	• What is the problem addressed by this approach? • What is the benefit of using this approach?
D3	Design Paradigm	• On which design paradigm is the approach leaned on? - Design principles - Code smells - Design smells - Design patterns
D4	Design Model	• Does the approach rely on a formal design model? • Is the model complete?
D5	Design Improvements	• Does the approach provide recommendations for design improvements? • In which way does a software engineer or architect get support?
D7	Validation	• Is the approach validated in an industrial or an academic environment? • Is the approach validated on an open-source or industrial project? • Size of validations (# of projects) • Notes for validation.
D8	Tool Support	• Is there a tool support for the approach?

3.2.3 Tool Support

MAXQDA 12

For coding the scientific articles, we decided to use MAXDQA version 12 since the JKU holds multiple licenses and colleagues made good experience in using this tool for coding interview results. By the first impression and by coding about ten primary papers, MAXQDA works intuitively and as intended. For example, the code system can be defined as a hierarchical tree as shown in Figure 3. Additionally, each code can be assigned to a particular

color. This easies the task of skimming already coded papers. In order to assign a code to a text passage, the code can be moved to a selected text passage by using drag-and-drop.

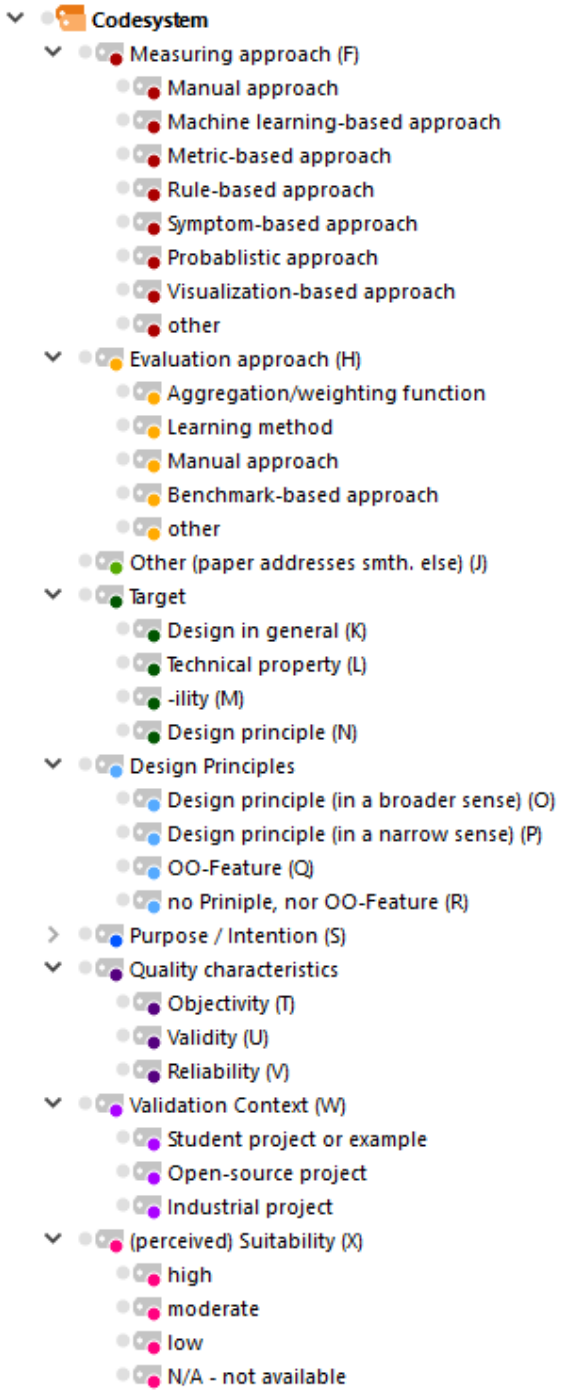

Figure 3: Code System in MAXQDA 12

Google Forms

The questionnaire for the data extraction phase is implemented in Google Forms. This online service gives us the possibility to evaluate the primary papers in a distributed manner and to store the results centralized in one online database. For filling out a questionnaire, a reviewer has to open a ULR and starts by entering the paper title which is required. Afterwards, the reviewer has to answer the questions as listed in Table 9. Finally, the form needs to be submitted before the reviewer can continue evaluating the next paper.

4 Data Synthesis Phase

The data synthesis focuses on collecting and summarizing the results of the paper assessments. This synthesis process can be descriptive and non-quantitative, but it is sometimes possible to complement a descriptive synthesis with a quantitative summary (Kitchenham & Charters, 2007). Although it is important to have a clear understanding of the data synthesis activities right at the beginning of the SLR and when writing the review protocol, some issues cannot be resolved until the data is actually analyzed (Kitchenham & Charters, 2007). In our case, this issue will come up since we are not familiar with all design assessment approaches and some approaches will be discovered while reading and studying the scientific articles.

According to (Popay et al., 2006), the synthesis process is the key element of each systematic review. The reason is that it brings together the findings form the individual papers and draws conclusions based on the results. For conducting a data synthesis, a quantitative or a narrative approach can be used (Popay et al., 2006); sometimes both are used in combination.

4.1 Narrative (Descriptive) Synthesis

Although (Popay et al., 2006) discuss multiple tools and techniques for developing a preliminary synthesis, this seminar paper focuses on those techniques that are considered as most relevant for the SLR. Actually, they are: textual descriptions, grouping and clustering, and tabulation.

4.1.1 Textual Descriptions

As recommended by (Popay et al., 2006) a good starting point in a narrative data synthesis is to compose a descriptive paragraph on each investigated paper. Therefore, it is important to produce these narrative descriptions in a systematic manner and to include the same information for all studies if available.

When mapping this idea to the method of this SLR, we are well prepared for producing these textual descriptions because the data extraction questionnaire (form) is especially designed for this task. The form contains text boxes that allow summarizing the questioned aspect in a structured way. Moreover, some questions are mandatory for ensuring the highest possible data quality. While the textual descriptions are a useful way for becoming familiar with the included papers and ideas, it can be very difficult to discern patterns across the design assessment approaches from these textual descriptions; particularly, when there is a large number of design assessment approaches.

4.1.2 Grouping and Clustering

Depending on the definition of inclusion and exclusion criteria as well as the size and quality of the existing literature, the pool of papers that need to be synthesized can be very large (Popay et al., 2006). While a statistical meta-analysis on this pool can be conducted automatically and with tool support, a narrative synthesis will fail because of the heavy workload. Consequently, it is necessary to divide the pool of papers into smaller groups. Although grouping papers is done at an early stage of the data extraction or data synthesis phase, it may be required to redefine the initial groups as a deeper understanding leads to another viewpoint.

For grouping the primary papers of this SLR, we consider to use the categorization codes listed in Table 3 and 4. The codes are deductively derived and each paper should fit within one of these. When there is the need to add a new group, the code can be added to the coding system. After grouping the papers, we believe that it is easier to detect patterns within and across various groups.

4.1.3 Tabulation

For presenting the result of both a qualitative and quantitative synthesis, tabulation is a common approach as it can be used at any stage of the process (Popay et al., 2006). For example, tabulation can be applied for showing an overview of the descriptions of the included primary papers and for deriving patterns between them. Nevertheless, the approach of showing data in table is typically used for providing details of study designs, of the quality assessment, or of the outcome of measures (Popay et al., 2006). The reason therefore is that the data can be presented in a structured manner by having the data separated by columns.

Actually, tabulation is used at multiple stages of this SLR. For instance, to summarize the search result of the term-based search, the paper assessment of the voting process, or the gathered information form the data extraction phase. Although tables facilitate the presentation of data, the layout of the tables can become very complex and confusing. This is the case when columns are linked with each other.

4.2 Qualitative Synthesis

When the SLR contains primary papers and studies that integrate experiments or studies that comprise natural language conclusions, it may occur that different researcher may have interpreted terms and concepts with different meaning (Kitchenham & Charters, 2007). As a results, it is necessary to clean up this data by one of the following approaches (Noblit & Hare, 1999):

- *Reciprocal Translation*: When papers are dealing with similar things and researchers are adding additional information, it is supportive to translate each interpretation into each of the other interpretation.
- *Refutational Synthesis*: When there is a refutation of a claim, it is necessary to understand both the refutation and the publication containing the claim (hypothesis). This provides the possibility to analyze the refutation in detail.
- *Line of Argument Synthesis*: The problem related to this approach roots in the problem that researches may run into concerns about their conclusions of a topic based on a set of selected studies (primary papers) which consider just a part of the problem. For dealing with this uncertainty, the studies need to be analyzed individually. Afterwards, the set of studies should be analyzed as a whole what is similar to a descriptive synthesis – discussed above. Consequently, important issues are identified and underlying approaches should be documented and tabulated.

5 Pilot Study for Quality Assurance

Since the SLR is still going on due to time constraints of two project partners, this section will not show the final result of the work. For the full report I refer to the article that presents the entire work and will (hopefully) get accepted by our target journal. Instead of the final result, this section discusses a pilot study we conducted in order to test the coding system.

Before we handed out the primary papers to the participants, who are assessing the quality, we conducted a workshop to explain the coding system and discussing misinterpretations. For this pilot study we used a set of ten papers – see Appendix Part A – from which each participant received five papers for the first assessing round. Since the two project partner did not hold a license of MAXQDA, we used an Excel spreadsheet for reporting the assessments. This was manageable due to the low number of articles.

After the first round, which took approximately two hours (~25 minutes for each paper), I collected the assessments and reassigned the participants to five other papers of the initial set. The five papers were categorized again according to the coding guidelines. As a result, we received two different assessments for ten papers, which were then used to analyze deviations and misunderstandings. The following tables show the codes that were used for classifying each of the ten papers by the two different reviewers. A qualitative discussion of the result reveals the differences.

Table 12: Pilot Study - Assessment of Paper 1

Paper:	Investigating the Impact of Code Smells on System's Quality: An Empirical Study on Systems of Different Application Domains	
	Reviewer A	**Reviewer C**
Measuring Approach	Symptom-based approach	Symptom-based approach
Assessing Approach	Aggregation/weighting function	Aggregation/weighting function
Focus	- Design in general - Technical property	- Design in general
Design Principles	- DP i.a.b.s	- DP i.a.b.s
Intention	Improving software/design quality	Improving software/design quality
Objectivity	1	2
Validity	1	2
Reliability	1	1
Validation Context	Open Source Project	Open Source Project
Perceived Suitability	low	very low

The difference between the assessment of reviewer A and C is the code of T*echnical property* assigned to the measuring/assessing focus. After discussing this aspect, both came to the conclusion that the paper does not address a technical property such as coupling, cohesion, complexity, or modularity. Next to the difference in the coding, reviewer C assessed objectivity and validity with higher values. In fact, the paper has weaknesses in both quality criteria why the best evaluation is not justified.

Table 13: Pilot Study - Assessment of Paper 2

Paper:	Detecting design flaws via metrics in object-oriented systems	
	Reviewer A	**Reviewer C**
Measuring Approach	Metric-based approach	Metric-based approach
Assessing Approach	/	Aggregation/weighting function
Focus	- Design in general	- Design in general
Design Principles	- No Principle, nor OO-Feature	- No Principle, nor OO-Feature
Intention	Detecting (design/code) smells	Detecting (design/code) smells
Objectivity	2	2
Validity	1	2
Reliability	2	2
Validation Context	Industrial Project	Industrial Project
Perceived Suitability	high	high

This case is problematic, because reviewer A does not find an assessment approach compared to reviewer C. According to reviewer C, an aggregation/weighting function is proposed there. At the first glance this might be correct; however, the aggregated value is used to detect design flaws and not to compare the design assessments with previous releases or other projects. Thus, the paper does not propose an assessment approach.

Table 14: Pilot Study - Assessment of Paper 3

Paper:	Assessment of Package Cohesion and Coupling Principles for Predicting the Quality of Object Oriented Design	
	Reviewer A	**Reviewer C**
Measuring Approach	Metric-based approach	Metric-based approach
Assessing Approach	Aggregation/weighting function	Aggregation/weighting function
Focus	- Design Principle	- Design Principle
Design Principles	- DP i.a.b.s	- DP i.a.b.s
Intention	Predicting quality assurance	Predicting quality assurance
Objectivity	2	2
Validity	0	0
Reliability	0	1
Validation Context	Student Project	Student Project
Perceived Suitability	low	very low

These assessments are almost a perfect match. This might result from the fact that the goal of the paper is well specified and the codes for *Focus* and *Design Principles* are precisely defined and reflect the intention of the paper.

Table 15: Pilot Study - Assessment of Paper 4

Paper:	Investigating object-oriented design metrics to predict fault-proneness of software modules	
	Reviewer A	**Reviewer D**
Measuring Approach	Metric-based approach	Metric-based approach
Assessing Approach	Learning method	Learning method
Focus	- Technical property	- Technical property
Design Principles	- no Principle, nor OO-Feature	- DP i.a.b.s
Intention	Predicting error proneness	Predicting error proneness
Objectivity	2	2
Validity	2	1
Reliability	2	2
Validation Context	Open Source Project	Open Source Project
Perceived Suitability	/	very low

While the codes for *Measuring Approach*, *Assessing Approach*, *Focus* and *Intention* are equal, reviewer A thinks that there is no principle or OO-feature addressed. This is true, since the design principle (in a broader sense) that is mentioned in the introduction is actually not addressed by the measuring or assessing approach.

Table 16: Pilot Study - Assessment of Paper 5

Paper:	DECOR: A Method for the Specification and Detection of Code and Design Smells	
	Reviewer A	**Reviewer D**
Measuring Approach	Symptom-based approach	Symptom-based approach
Assessing Approach	/	/
Focus	- Design in general	- Design in general
Design Principles	- No Principle, nor OO-Feature	- No Principle, nor OO-Feature
Intention	Detecting (design/code) smells	Detecting (design/code) smells
Objectivity	2	2
Validity	2	2

Reliability	2	2
Validation Context	Open Source Project	Open Source Project
Perceived Suitability	high	very high

Another perfect match is given with paper 5 since both reviewers assessed the paper almost equally. This might result from the precisely defined aim of the paper and the high quality regarding the scientific criteria.

Table 17: Pilot Study - Assessment of Paper 6

Paper:	Product Metrics for Automatic Identification of "Bad Smell" Design Problems in Java Source-Code	
	Reviewer B	**Reviewer C**
Measuring Approach	Metric-based approach	Metric-based approach
Assessing Approach	Aggregation/weighting function	Aggregation/weighting function
Focus	- Design in general	- Design in general
Design Principles	- DP i.a.b.s	- DP i.a.b.s
Intention	Improving software/design quality	Detecting (design/code) smells
Objectivity	1	2
Validity	0	0
Reliability	1	1
Validation Context	Student Project, Examples	Student Project, Examples
Perceived Suitability	low	low

Using a set of metrics aggregated to various evaluation values of "bad smells" is the main idea of this paper. Whereas reviewer C considers the intention of the paper in just detecting "bad smells", reviewer B finds parts that suggest improvements of the design quality. After a short discussion regarding this difference, both conclude that the improvements are not applicable in practice and are not the main intention of the paper.

Table 18: Pilot Study - Assessment of Paper 7

Paper:	MIDAS: A Design Quality Assessment Method for Industrial Software	
	Reviewer B	**Reviewer C**
Measuring Approach	Metric-based approach	Rule-based approach
Assessing Approach	Manual approach	Manual approach
Focus	- ility - Design principle	- ility - Design principle
Design Principles	- DP i.a.n.s	- DP i.a.n.s
Intention	Improving software/design quality	Improving software/design quality
Objectivity	2	2
Validity	0	0
Reliability	2	1
Validation Context	Industrial project	Industrial project
Perceived Suitability	moderate	high

Even though it is clear that the paper uses design principles (in a narrow sense) to assess maintainability (-ility), both reviewers disagree in the measuring approach. The reason therefore is that the authors of the paper refer to the term of rule when talking about a set of combined metrics. Thus, the reviewers run into the problem that the authors used the term of rule in an imprecisely manner, resulting in a different interpretation. After discussing

the difference, the reviewers agreed that these rules are not rules in sense of design heuristics or design best practices but rather a combination of metrics.

Table 19: Pilot Study - Assessment of Paper 8

Paper:	An Empirical Analysis of Object-Oriented Metrics for Java Technologies	
	Reviewer B	**Reviewer D**
Measuring Approach	Metric-based approach	Metric-based approach
Assessing Approach	/	/
Focus	- Technical property	- Technical property
Design Principles	- OO-Feature	- OO-Feature
Intention	Improving software/design quality	Improving software/design quality
Objectivity	1	2
Validity	1	1
Reliability	1	2
Validation Context	Open source project	Open source project
Perceived Suitability	low	/

Due to the nature of paper 8, the reviewers had a good agreement in their assessments. In fact, the paper analyzes metrics that are used for measuring design aspects, but it does not propose an assessing approach. At some points it groups the metrics according to technical properties and object-oriented features. Consequently, the use of the right codes is obvious; except for the perceived suitability. In this concern, both have a different opinion. Reviewer D does not find a reason for assigning this code.

Table 20: Pilot Study - Assessment of Paper 9

Paper:	Automated design flaw detection in object-oriented systems	
	Reviewer B	**Reviewer D**
Measuring Approach	Symptom-based approach	Symptom-based approach
Assessing Approach	Aggregation/weighting function	/
Focus	- Technical property	- Technical property
Design Principles	- OO-Feature	- No Principle, nor OO-Feature
Intention	Improving software/design quality	Improving software/design quality
Objectivity	1	1
Validity	0	0
Reliability	2	1
Validation Context	Open source project	Open source project
Perceived Suitability	very low	very low

Similar to the assessments of paper 2, both reviewers have a different perception regarding the assessment approach. While reviewer D does not identify an assessment approach, reviewer B claims that there is an aggregation/weighting function proposed. In fact this case is conflicting since there is an aggregation function to derive a single value, but it is not used in the context of an assessment.

Table 21: Pilot Study - Assessment of Paper 10

Paper:	Constructing models for predicting extract subclass refactoring opportunities using object-oriented quality metrics	
	Reviewer B	**Reviewer D**
Measuring Approach	Metric-based approach	Metric-based approach
Assessing Approach	Aggregation/weighting function	Aggregation/weighting function

Focus	- Technical property	- Technical property
Design Principles	/	/
Intention	Predicting refactoring	Predicting refactoring
Objectivity	2	2
Validity	2	2
Reliability	2	2
Validation Context	Open Source Project	Open Source Project
Perceived Suitability	moderate	high

The last comparison of the tenth paper shows a good result, with just one difference; namely, in the perceived suitability. This may result from the clear goal of the paper and the codes that are available for evaluating it.

6 Future Work and Lessons Learned

The process of building the coding system had both a deductive and inductive character. Based on our understanding of measuring and assessing object-oriented design, we specified the top level classifiers and their direct sub-classifier. However, there was no guarantee that this first idea of classification captured all articles from the SLR. Thus, we investigated 25 reference papers and derived further classifiers inductively. According to (Mayring, 2010), building a category system inductively is more suitable for a qualitative content analysis because the result – the category system – is closer to the objects of investigation. I agree to this opinion since missing categories could be identified by the consideration of the reference papers.

After specifying the category system, (Mayring, 2010) suggest a revision step including 10-50% of the material. We conducted this revision as part of a workshop that was used to train the participants on the coding system. The result of this revision – pilot study – is depicted and qualitatively discussed in the previous section. We learned from the pilot study that the participants agree in the assignment of the code for measuring and assessing approach. When it comes to the focus of the paper and whether it addresses design principles, then there are minor deviations. The reason therefore is that there are papers that mention, for example, a design principle, but they do not investigate it in more detail. Thus, a reviewer could assign a code to the paper while a more critical reviewer did not see the need therefore.

An assessment aspect that is not extensively discussed in the previous section is the evaluation for perceived suitability (pink row). Assessing the perceived suitability is difficult and sometimes very subjective. We figured out that this code needs a more specific definition since it is currently too vague. However, we do not want to achieve a perfect agreement in this assessment aspect since this code should work as indicator of the strength of the paper. We could observe that this indicator is linked with the three quality characteristics: objectivity, validity, and reliability. In case a paper gets a high assessment for the three characteristics, it is likely that the perceived suitability is high or at least moderate. Thus, the classification of the perceived suitability works as an additional distinctive feature between strong and weak papers.

The future work of this SLR will concentrate on completing the assessment for all 122 papers. Therefore, the set of papers is divided into four parts and every reviewer has to

check one sub-set. After this first round, the assessments will be cross-checked by another reviewer to get at least two meanings on one article. With the classification in MAXQDA, we plan to answer the guiding questions in a quantitative manner first. For instance, we would like to show the distribution of measuring approaches and which area of design measurement is not addressed. This can be discussed based on the top level classification of the papers. Additionally, we also want to show which groups of, for instance, assessment approaches are available and how they differ in their characteristics. Therefore, it is necessary to check the papers in a qualitative way in order to understand the intention behind the proposed approach. MAXQDA perfectly supports this step as we already assigned a code to the relevant parts which can be easily queried by the tooling.

Additionally, MAXQDA provides a feature that allows grouping the papers visually. Hence, we do have support for the technique of grouping papers. (Popay et al., 2006) mentions that this technique is useful for a preliminary data synthesis. I conducted this technique on a set of papers and agree with (Popay et al., 2006) because visually grouping papers helps in ordering thoughts and to derive conflicting or supportive relations between the identified groups.

For the data extraction strategy of this SLR, a questionnaire is used. This questionnaire is a supporting tool for the explication, which is a technique that uses content-based material to explain the investigated text (Mayring, 2010). The questionnaire contains a set of questions that has to be answered according to the information provided by the paper. Besides, notes and remarks can be added for supporting the data synthesis phase in identifying groups and related papers.

All in all, this concept for the qualitative content analysis of a set of scientific articles gathered from the SLR is based on a solid category system that will fit the further work. This category system is aligned to the guiding questions that are used to answer the research question: *Is the research area of assessing object-oriented design ready to move on, or should we critically rethink the proposed approaches?* Hence, a contribution to the research community can be provided and my dissertation has a strong base for proposing a novel measuring and assessing approach of object-oriented design.

7 References

Bansiya, J., & Davis, C. G. (2002). A hierarchical model for object-oriented design quality assessment. *IEEE Transactions on Software Engineering*, *28*(1), 4–17.

Bortz, J., & Döring, N. (2007). *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler: Limitierte Sonderausgabe*. Springer-Verlag.

Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, *20*(6), 476–493.

Greve, W., & Wentura, D. (1997). *Wissenschaftliche Beobachtung: eine Einführung*. Beltz, Psychologie Verlags Union.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*.

Kläs, M., Heidrich, J., Münch, J., & Trendowicz, A. (2009). CQML Scheme: A Classification Scheme for Comprehensive Quality Model Landscapes. In *Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications, SEAA '09* (pp. 243–250).

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159–174. http://doi.org/10.2307/2529310

Marinescu, R. (2004). Detection strategies: metrics-based rules for detecting design flaws (pp. 350–359). Presented at the Software Maintenance, 2004. Proceedings. 20th IEEE International Conference on, IEEE. http://doi.org/10.1109/ICSM.2004.1357820

Mayring, P. (2010). Qualitative Inhaltsanalyse. In G. Mey & K. Mruck (Eds.), *Handbuch Qualitative Forschung in der Psychologie* (pp. 601–613). VS Verlag für Sozialwissenschaften.

Noblit, G. W., & Hare, R. D. (1999). Chapter 5: Meta-Ethnography: Synthesizing Qualitative Studies. *Counterpoints*, *44*, 93–123.

Ouni, A., Kessentini, M., & Sahraoui, H. (2014). Chapter Four - Multiobjective Optimization for Software Refactoring and Evolution. In A. Hurson (Ed.), *Advances in Computers* (Vol. 94, pp. 103–167). Elsevier.

Popay, J., Roberts, H., Sowden, A., Petticrew, M., Arai, L., Rodgers, M., … Duffy. (2006). Guidance on the conduct of narrative synthesis in systematic reviews: A product from the ESRC Methods Programme.

8 Appendix

A. Set of Papers for Pilot Study

1. (Fontana, Ferme, Marino, Walter, & Martenka, 2013) - Fontana, F. A., Ferme, V., Marino, A., Walter, B., & Martenka, P. (2013). Investigating the Impact of Code Smells on System's Quality: An Empirical Study on Systems of Different Application Domains. In 2013 29th IEEE International Conference on Software Maintenance (ICSM) (pp. 260–269). http://doi.org/10.1109/ICSM.2013.37, Indexed by: IEEE

2. (Marinescu, 2001) - Marinescu, R. (2001). Detecting design flaws via metrics in object-oriented systems. In 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems, 2001. TOOLS 39 (pp. 173–182). http://doi.org/10.1109/TOOLS.2001.941671, Indexed by: IEEE

3. (Atole & Kale, 2006) Atole, C. S., & Kale, K. V. (2006). Assessment of Package Cohesion and Coupling Principles for Predicting the Quality of Object Oriented Design (pp. 1–5). Presented at the Digital Information Management, 2006 1st International Conference on, IEEE. http://doi.org/10.1109/ICDIM.2007.369321, Indexed by: IEEE

4. (Rathore & Gupta, 2012) - Rathore, S. S., & Gupta, A. (2012). Investigating object-oriented design metrics to predict fault-proneness of software modules. In 2012 CSI Sixth International Conference on Software Engineering (CONSEG) (pp. 1–10). http://doi.org/10.1109/CONSEG.2012.6349484, Indexed by: IEEE

5. (Moha, Guéhéneuc, Duchien, & Le Meur, 2010) - Moha, N., Gueheneuc, Y.-G., Duchien, L., & Le Meur, A.-F. (2010). DECOR: A Method for the Specification and Detection of Code and Design Smells. IEEE Transactions on Software Engineering, 36(1), 20–36. http://doi.org/10.1109/TSE.2009.50, Indexed by: IEEE

6. (Munro, 2005) - Munro, M. J. (2005). Product Metrics for Automatic Identification of "Bad Smell" Design Problems in Java Source-Code. In Software Metrics, 2005. 11th IEEE International Symposium (pp. 15–15). http://doi.org/10.1109/METRICS.2005.38, Indexed by: IEEE

7. (Samarthyam, Suryanarayana, Sharma, & Gupta, 2013) - Samarthyam, G., Suryanarayana, G., Sharma, T., & Gupta, S. (2013). MIDAS: A Design Quality Assessment Method for Industrial Software. In Proceedings of the 2013 International Conference on Software Engineering (pp. 911–920). Piscataway, NJ, USA: IEEE Press. Retrieved from http://dl.acm.org/citation.cfm?id=2486788.2486912, Indexed by: ACM / IEEE

8. (Farooq, Braungarten, & Dumke, 2005) Farooq, A., Braungarten, R., & Dumke, R. R. (2005). An Empirical Analysis of Object-Oriented Metrics for Java Technologies. Presented at the 9th International Multitopic Conference, IEEE INMIC 2005, IEEE. http://doi.org/10.1109/INMIC.2005.334410, Indexed by: ACM / IEEE

9. (Trifu, Seng, & Genssler, 2004) - Trifu, A., Seng, O., & Genssler, T. (2004). Automated design flaw correction in object-oriented systems. In Eighth European Conference on Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings (pp. 174–183). http://doi.org/10.1109/CSMR.2004.1281418, Indexed by: IEEE

10. (Al Dallal, 2012) Al Dallal, J. (2012). Constructing models for predicting extract subclass refactoring opportunities using object-oriented quality metrics. Information and Software Technology, 54(10), 1125–1141. http://doi.org/10.1016/j.infsof.2012.04.004, Indexed by: ACM