



UNIVERSITY OF GOTHENBURG

The Visualization of Software Quality Metrics  
*A Systematic Literature Review*

*Bachelor of Science Thesis in Software Engineering and Management*

Dur Abuzaid  
Scott Titang

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

## **The Visualization of Software Quality Metrics – A Systematic Literature Review**

Dur Abuzaid  
Scott Titang

© Dur Abuzaid, June 2014.

© Scott Titang, June 2014.

Examiner: Ana Magazinius

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering  
Göteborg, Sweden June 2014

# THE VISUALIZATION OF SOFTWARE QUALITY METRICS – A SYSTEMATIC LITERATURE REVIEW

Dur Abuzaid

[durr\\_abuzaid@hotmail.com](mailto:durr_abuzaid@hotmail.com)

Department of Computer Science and Engineering  
Software Engineering, Gothenburg, Sweden.

Titang Ayeah Scott

[scott.titang@gmail.com](mailto:scott.titang@gmail.com)

## ABSTRACT

Researchers and practitioners have put a lot of attention in the visualization of software quality metrics. However, there is little attempt to systematically review and classify the different approaches of the quality metrics. Our objective in this study is to conduct a systematic literature review to identify the types and purposes for visualizing software quality metrics, including an analysis of existing visual attributes, interaction mechanisms and the different types of evaluations that are performed on the visualizations. We use a thematic analysis on 18 studies that met our inclusion criteria and found out that graph-based visualization is the most commonly used to visualize quality metrics such as lines of code, McCabe's complexity and the number of methods. These metrics are mapped to the visual aspect of dimension, such as length, width, height and depth of the visual data, for instance, 3D boxes. In addition, we found out that the main purpose of employing visualization techniques is to improve the understanding of the structural characteristics of a software entity.

## KEYWORDS

Visualization techniques, software, quality metrics.

## I. INTRODUCTION

Software metrics and their visualization are two important features of measurement systems. Companies have developed measurement systems to guide them to monitor and control the status and progress of their products (Shollo and Pandazo, 2008). It provides objective information required by managers to make decisions that positively impacts their products (Antolić, 2008).

Software metrics are used to provide wide range of information about performance, quality, schedule and cost of software data. In this study we pay attention to quality related metrics of

software because (i) it is an important topic for researchers and practitioners within the software engineering community (Singh et al., 2013), (ii) the quality products offer, indicates the level of customer satisfaction and improves their competition in the market. However, there have been a variety of definitions of quality and its metrics depending on the actual goal. In some studies, quality is a measure of stability (Staron et al., 2013; Girba et al., 2005), usability (Wingkvist et al., 2012), complexity (Rosner et al., 2008; Bohnet and Dollner, 2011), functionality (Muto et al., 2011), and maintainability (Erdemir et al., 2011; Bauer and Heinemann, 2012), just to name a few.

The main objective for visualizing quality metrics is to improve the understanding and analysis of information about the software data (Denis et al., 2005). In academia, there have been a lot of discussions about visualization of quality metrics. For instance, Erdemir et al. (2011) used visualizations to simplify the comprehension and refactoring of complex software systems. Their visualization tool was able to extract and graphically visualize quality metrics and their relations for java source codes. Similarly, Varet et al. (2013) presented a visualization tool for software quality metrics that evaluate and helped them minimize the complexity of C and Ada source codes of an embedded system. While both authors have focused on the visualization of source codes, Wingkvist et al. (2010) and Knab et al. (2009) focused on the visualization of software documents. Wingkvist et al. (2010) visualized the quality of technical documents to gain an understanding of their uniqueness and usability. Knab et al. (2009) visualized the software problem reports to explore the effort estimation quality and bug-life cycle during the development process. However, these studies have drawbacks in because their results are framed within a given context which makes the generalization and applicability of their approach tricky in other contexts.

**Table 1:** Research questions and their motivations

Research Questions	Motivation
1. What are the different types and purpose of the software quality metrics visualized?	The goal is to get an overview of different types of quality metrics for software and identify their purpose and attributes and to investigate whether there is a relationship between different software quality metrics and their visualizations.
2. What types of information are provided by the visualized quality metrics?	To gain an understanding of the different types of values associated with quality metrics.
3. What types of visualization techniques are used to analyze the results of software quality metrics?	
4. What are the different objectives for employing visualization techniques for the analysis of software quality metrics?	To identify the main purpose of applying visualization techniques for quality metrics. Important for practitioners to know what they can do with the visualization of software quality metrics?
5. Which visual attributes are used in the visualization of software quality metrics?	To identify the main visual attributes mapped to quality metrics and their advantages in the visualization of quality metrics.
6. What types of interactions are performed on the different visualization techniques?	To highlight the most commonly interactions performed on the visualization techniques.
7. What types of evaluation performed on the visualization under investigation?	

The area of software quality metric visualization still poses a lot of challenges and requires special attention because (i) people are always surrounded by overwhelming information and (ii) there is no particular set of guidelines on how to select the most relevant visualization technique(s) for a given purpose. This is the case of one company located at the Lindholmen area in Gothenburg. To meet this goal, we need to scientifically identify and classify visualization approaches of quality metrics from existing literature – a systematic literature review. The knowledge gained from this systematic review will provide key findings relevant and applicable to their needs and concerns. We realized that other studies have also carried out a systematic literature review for the architecture of the system (Shahin et al., 2014), or the evolution of software (Novais et al., 2013), or the evolution of the architecture (Breivold et al., 2012). However, these studies were not concerned with the visualization of quality metrics.

The contents of this paper is organized and presented in different sections as follows: Section 2 presents our systematic process including the research questions. In Section 3, we present the results of and discuss the principal findings in Section 4. Threats to validity are presented in Section 5 and conclude the paper in Section 6.

## II. METHODOLOGY

We use a systematic literature review (SLR) in accordance to Kitchenham (2007) because it provides a well-defined process for identifying, evaluating and interpreting relevant studies available for a given set of research question(s). Kitchenham’s guideline of an SLR consists of three main phases: *review protocol*, which focuses on the search, selection, data extraction and synthesis strategies; *conducting the review and reporting the review*. The review protocol designed for this study consists of: (i) research questions, (ii) search strategy, (iii) criteria for inclusion and exclusion, (iv) study selection, (v) study quality assessment, and (vi) data extraction and synthesis.

### 2.1 Research Questions

The main objective of this paper is to systematically select and review existing studies to provide an overview of “*what software quality metrics visualization techniques are supported in existing studies, and what types of evaluations have been performed on them?*”

Our research question focuses on three areas: the different quality metrics visualized, the types of visualization techniques used, and their evaluations. Table 1 presents a set of sub questions necessary to answer our research

question, as well as their motivations. The results for these questions, which are directly linked to the objective of this SLR, can provide missing gaps and highlight the best practices of quality metric visualizations for researchers and practitioners, respectively.

## 2.2 Search Strategy

The strategy for collecting relevant literature is composed of two main elements: search keyword and data sources. The following initial steps are introduced to identify the relevant data sources and search terms:

- Trial search using various combinations of initial search keywords derived from the research questions.
- Consult with our supervisor who has experience in applying visualization techniques for software quality metrics, to get to obtain a better combination of keywords related source and literature on metric visualization.

### 2.2.1 Search terms

Our search terms are to match the paper titles and abstract from the electronic data sources. The search string for the study was: “quality metric” AND “software” AND “visualization”. They are derived according to the study topic and research questions. We use the Boolean AND to form our search query.

### 2.2.2 Data Sources

Scientific databases formed the main source for collecting potential relevant studies. We use 4 electronic databases – *ScienceDirect*, *IEEE*, *Scopus* and *Inspec*. Google Scholar and the ACM digital library were not included because the former generate irrelevant searched results with low precision (Shahin et al, 2014) and from the latter, we realized that a lot of articles were already in either one or more of the 4 databases. For instance, of the first 30 papers from ACM, *Inspec* contained 19 papers while *ScienceDirect* contained 15 papers.

## 2.3 Inclusion and Exclusion Criteria

In table 2, we present the inclusion and exclusion criteria with the main objective to retrieve all relevant studies from the database in different steps (see fig. 1). We limited the time period for our search because we intend to include only potential relevant studies, stored in the database for 14 years.

Thus we considered full studies in written English from conferences and workshops published between the years 2000 – 2014. We are interested in the recent studies of quality metric visualization because the field of software engineering has grown rapidly. The selected studies should propose or present a visualization approach for analyzing software quality related metrics. In addition, the study should report evaluations performed to support and validate their proposed technique.

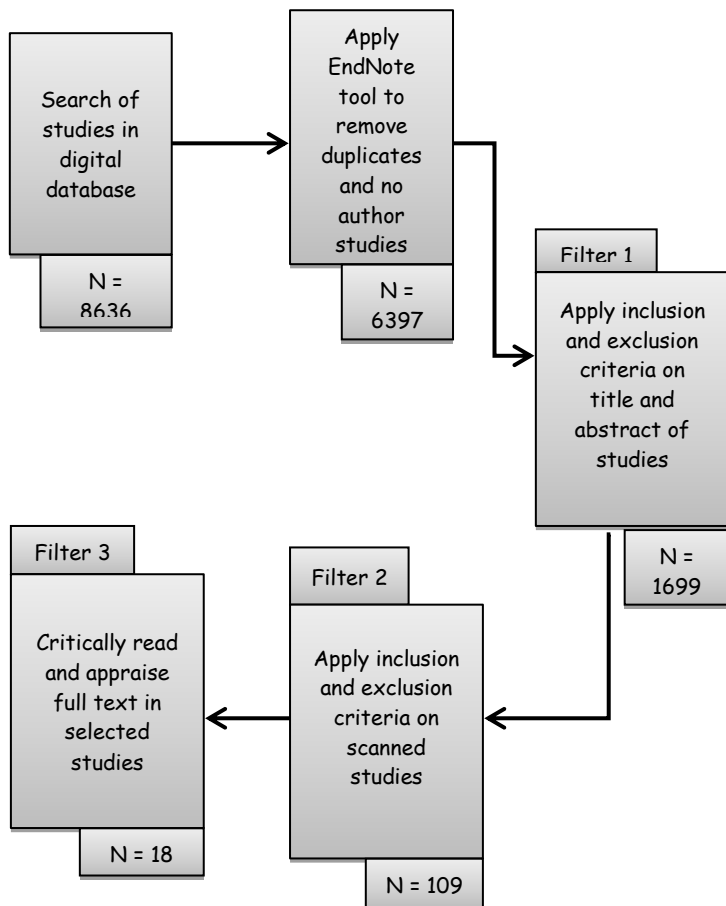
We exclude studies that do not explicitly relate to the visualization of quality metrics of software. For instance, studies that visualized quality related metrics of rendered data, architectural components and software models through simulations.

**Table 2:** Inclusion and exclusion criteria of the SLR

<i>Inclusion criteria</i>	
I <sub>1</sub>	A study that investigates a visualization technique on software quality related metrics
I <sub>2</sub>	A study must be a scientific research that is written in English and available in full text
<i>Exclusion criteria</i>	
E <sub>1</sub>	A study that only focuses on the evaluation of software visualization techniques or proposes evaluation guidelines
E <sub>2</sub>	A study that visualizes quality related metrics of other software data e.g. rendered images
E <sub>3</sub>	Reviews, presentations, posters, tutorial summaries, and panel discussions

## 2.4 Study Selection

The selected studies constitute the main primary papers relevant for the systematic review. It includes only studies with useful information relevant for answering the research questions presented in table 1.



**Figure 1:** Stages of the search process and number of selected studies in each stage

Figure 1 illustrates the number of studies selected at each phase. From the search phase, 8636 papers were obtained. These studies were retrieved and managed in EndNote X7 tool<sup>1</sup>. EndNote, a reference manager tool which helped us to retrieve, store and manage the searched studies, was used to manage the searched results. Duplicates and studies without authors were automatically detected and removed by the tool. For those studies that we could not decide on in Filter 1, they were retained for further investigation in Filter 2. In Filter 1, irrelevant studies were removed after we applied the inclusion and exclusion criteria on their titles and abstracts. This resulted into 109 studies.

They were scanned and the inclusion and exclusion criteria were applied to the approach and conclusion of the study. By scanning, we focused on three things: the software quality metric(s) identified in the study, the visualization approach used and the evaluations that are performed on the visualization. This was done without actually reading the full text of the paper. For each paper, the reasons for

their inclusion or exclusion decisions were recorded on a spreadsheet. To enhance the level of reliability of agreement of the included and excluded studies, the authors applied the Fleiss Kappa statistical measurement on 50 studies. The result from the measurement was 0.86, which is within the range for significant agreement. The authors discussed and solved any variations that surfaced for a given paper. In addition, a reassessment was made by the authors, together with supervisor, on whether to include the paper or not. This resulted into 18 studies selected for the final list (see Appendix A table 5).

## 2.5 Data Extraction and Synthesis

In the data extraction and synthesis process, the full texts of the final studies were read and all relevant information needed to address our research questions was extracted. The extracted information items for each study were recorded in a spreadsheet for further analysis. Table 3 shows the data item from which the relevant information in each study was extracted. We perform statistical and thematic analysis on the data collected. Statistical analysis is employed to provide a descriptive summary of our primary studies including publication venue and distribution of studies over the years. In order to answer our RQs, we employ a thematic approach (Taylor-Powell and Renner, 2003) to identify, examine, and record patterns in the data. Our thematic analysis consists of five steps, conducted as follows:

- i. *Get to know the data:* the data items presented in table 3, helped us to familiarize with contents in each study.
- ii. *Focus on the analysis:* in this step, we examined the data contents in detail to decide on how they would aid in answering the research questions. Table 3: Extracted data items for the research question

<sup>1</sup> <http://endnote.com/>

**Table 3:** Extracted data items for the research question

#	Extracted Data	Description	RQ
D1	Meta-data information	The author, title, year etc of the study	Overview
D2	Metric Purpose	The purpose of quality metrics proposed or used in the study	RQ <sub>1</sub>
D3	Information Types	The type of information provided by the quality metric	RQ <sub>2</sub>
D4	Visualization techniques (VTs)	The visualization techniques used to analyze the quality metric	RQ <sub>3</sub>
D5	Purpose of VTs	Identify the purpose of visualization technique used in the study	RQ <sub>4</sub>
D6	Visual Attributes	The visual attributes used in the visualizations of quality metrics	RQ <sub>5</sub>
D7	Interactions	The interaction mechanisms used in the visualization	RQ <sub>6</sub>
D8	Validations	The purpose and types of validations carried out in the study	RQ <sub>7</sub>

- iii. *Categorize the information:* here, a list of all the extracted information relating to each data item is created. Codes are then generated and assigned to the extracted information, which helps with organizing and sorting the information into different categories or themes.
- iv. *Identify patterns and connections:* once themes are established, we assembled all the primary studies within their respective categories. This enhanced recognition and analysis of similarities or differences of studies within each category and their analysis.
- v. *Interpretation:* in this step, we attached meaning and significance of our analysis based on the categories in the previous step.

The thematic approach was employed for all data items except data item D4. Instead, we employed a predefined classification of visualization techniques according to Shahin et al. (2010). These are *graph-based*, *notation-based*, *matrix-based*, and *metaphor-based* visualization. The classification for data items D6 and D8 are presented. Data item D6 evaluates the evidence of visual attributes employed in the visualizations and resulted into two classifications of the visual attributes – *data* and *quality metrics*. Similarly, D8 resulted into three categories for the empirical validations performed on the visualization technique – on the *type*, *aspect*, and the *outcome* of the evaluation(s). In this study, type of evaluation is defined by its objective. We identified 3 sub-categories, which are: (i) *use case* – the goal here is to demonstrate implementation and analysis

of visualization technique, for a specific task; (ii) *user studies* – the goal is to determine the benefits of the visualization technique through a number of participants/users usually through questionnaires; (iii) *experiments*: the goal is similar to user studies however, the benefits of the visualization are determined from the comparison of predefined measurements against users' performances.

## 2.6 Quality Assessments

A quality assessment was performed on all of the 18 studies by using the quality assessment questions defined in table 4. They were used to assess the quality and to validate the result of the obtained primary studies. Quality assessment process was performed in accordance to the guidelines by Kitchenham (2007) in order to reduce the bias in the study, internal validity and external validity. During the data extraction process three possible answers could be chosen for each question either “yes”, “partially” or “no”. Table 4 shows the list of the quality assessment questions and description of the criteria used to evaluate them.

Each author reviewed the 18 primary studies independently using the quality assessment questions. In case of variations in judgment, the authors discussed the differences. From table 4, all 18 primary studies clearly stated the aim and objective of the study. Question 2 shows that 16 studies were answered positively except for two studies, where the authors had to estimate the aim and objectives of the study. The answers of Q3, Q4, Q4 and Q5 shows that all the primary studies propose visualization of the quality metrics and used quality attributes to visualize the data.

**Table 3:** Questions for quality assessment

#	Questions	Yes	Partially	No
1	Does the study define the aim and objectives clearly?	18	0	0
2	Does the study identify a software quality metric(s)?	16	2	0
3	Does the study use an approach to visualize the proposed metrics?	18	0	0
4	Does the study clearly indicate the data of analysis?	18	0	0
5	Are the visual attributes for the metrics and data clearly stated?	18	0	0
6	Does the study provide validations to the proposed the VTs?	18	0	0
7	Are the results of evaluation clearly stated?	17	1	0

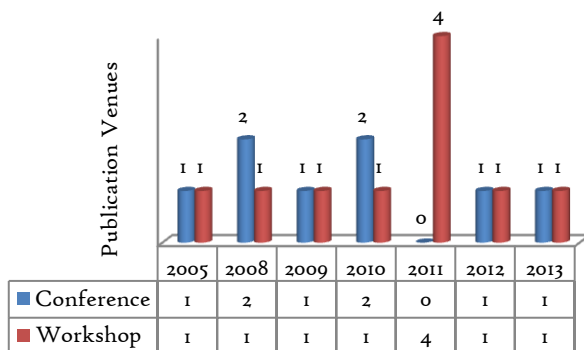
In Q6 and Q7 indicates that all the studies evaluated the visualization of quality metrics and almost all the studies clearly stated the result of the evaluation (except for 1 study).

### III. RESULTS

The goal of this study is to investigate the types of visualization approaches used to visualize software quality metrics and to present the evaluations that are performed on them. In this section, we present the results of the systematic review, structured in a manner consistent with the ordering of the research questions in table 2. As stated previously, this paper employs the systematic review approach and uses thematic analysis for data analysis and classification.

#### 3.1 Meta data of studies

Before presenting results of the extracted data, we report the classifications of the origins of the primary studies, for example, by year, type, and venue of their publication. This classification indicates the publication venue that mostly contributes to the visualization of software quality metrics.



**Figure 2:** Distribution of the number of studies by venue per year.

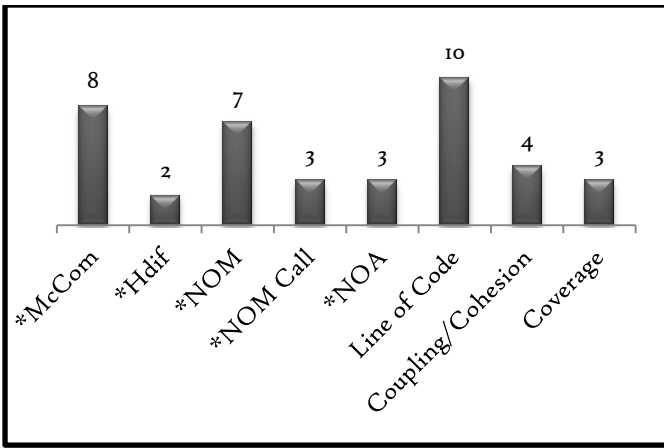
Figure 2 illustrates the different number of studies published throughout the year. Conferences and workshops provide the main

publication venues for our primary studies. Workshops provide the leading publication venue for the relevant studies (10 studies), though with a small margin as compared to conferences (8 studies). From the figure, we identify two important aspects to be taken into consideration up to and until 2011. Firstly, we notice an equal amount of effort within the research community to visualize quality metrics. However, there is a drop in the number of papers published in both venues, after this period. Nonetheless, according to Gu and Lago (2009), the presentation of the meta information of the studies and their classification and classification of the origins of primary studies does not provide the relevant information to answer our research questions.

#### 3.2 The purpose and types of quality metrics

The analysis of data item D2 provides answers to RQ1 presented in table 2. The purpose of quality metrics for the majority of the studies is to enhance maintainability of software. We found that the quality metrics proposed measure a specific attribute, characteristics or property of the given software entity. All the primary studies use more than one quality metric to describe abstract aspects of that entity. Figure 3 highlights the three sets of quality metrics frequently used – McCabe complexity [S3, S4, S5, S7, S8, S11, S14, S18], number of methods [S1, S11, S13, S15, S16, S17, S18], lines of code [S3, S4, S5, S8, S10, S11, S14, S16, S17, S18]. However, we realize that not all the quality metrics are reported in the studies. We also find out that the goal of the VTs and the purpose of quality metrics are highly correlated.





**Figure 3:** Analysis of different types of metrics  
 \*NOA = Number of Attributes; \*NOM call = Number of method call; \*NOM = Number of methods; \*McCom = McCabe’s Complexity; \*Hdif = Halstead’s difficulty

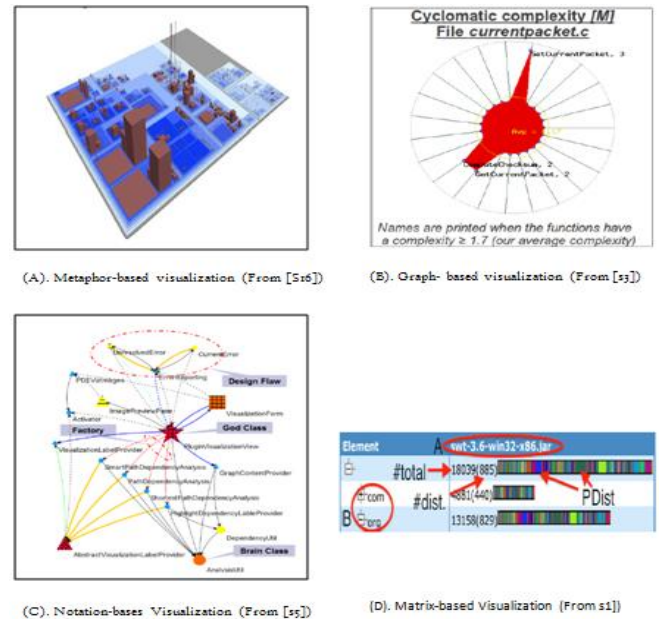
### 3.3 Types of information for quality metrics

This section presents findings to answer RQ<sub>2</sub>, “What types of information are provided by these quality metrics?” The results are obtained by analyzing data item D<sub>3</sub> and we classified the types of information into three categories – *numeric, ratio or scale*. The value of quality metrics in 15 studies [S<sub>1</sub>, S<sub>3</sub>, S<sub>4</sub>, S<sub>5</sub>, S<sub>7</sub>, S<sub>8</sub>, S<sub>9</sub>, S<sub>10</sub>, S<sub>11</sub>, S<sub>12</sub>, S<sub>13</sub>, S<sub>14</sub>, S<sub>15</sub>, S<sub>16</sub>, S<sub>18</sub>] are numeric. This can be attributed to the fact that numerical values are most precise and can be represented graphically among the ratio and scale. Moreover, in case of a higher numeric value, some studies [S<sub>4</sub>, S<sub>12</sub>] consider only the predefined threshold as the value for the quality metric. 6 studies [S<sub>2</sub>, S<sub>6</sub>, S<sub>8</sub>, S<sub>14</sub>, S<sub>17</sub>, S<sub>9</sub>] use ratios to represent metric values. For instance, [S<sub>2</sub>] uses ratio as a measure of a fraction of a document that is unique, and [S<sub>6</sub>] uses ratio to indicate the proportion of software’s specifications that have been implemented. Only 3 studies [S<sub>12</sub>, S<sub>14</sub>, and S<sub>17</sub>] used scaling to represent the value of quality. The value of quality metrics are scaled to assist data interpretation as some studies report that it is difficult to interpret the values of quality without knowing the context.

### 3.4 Types of visualization techniques

This section presents the results to answer RQ<sub>3</sub>. Visualization techniques are developed to visualize the different types of quality metrics presented in Section 3.2. To present our findings we employ Shahin’s pre-defined classification of visualization techniques – *Graph-based, notation-based, matrix-based and metaphor-based* visualizations. Examples of these visualization techniques are presented in figure 4. In figure 5,

we highlight the actual proportion of these techniques employed in the primary studies. Some studies employed more than one visualization techniques, for instance [S<sub>3</sub>, S<sub>12</sub>, S<sub>18</sub>]. Our result indicates that graph-based visualization is the most widely used visualization techniques. In addition, the visualizations contained more than one view – global and specific – to get an overview of the software entity under investigation and to identify hotspots that require greater attention, respectively.



**Figure 4:** Examples of four visualization techniques used in software visualization.

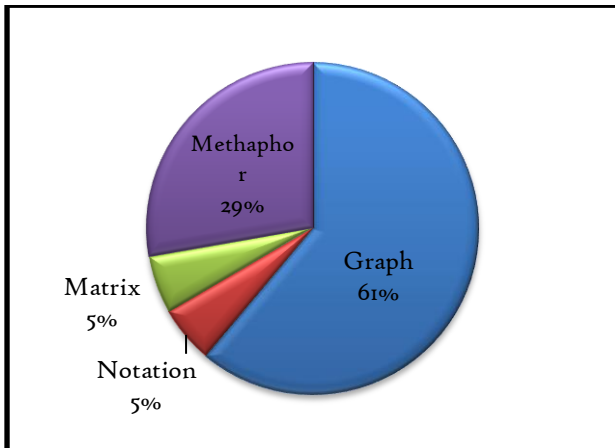
#### 3.4.1 Graph-Based Visualizations

In figure 5, this particular technique accounts for 61% of the primary studies that employ this techniques, for instance, [S<sub>3</sub>] uses the approach to represent the signatures of the different functions determined by the values of its computed metrics. They use the Kiviat diagram to indicate functions that require attention and can significantly impact the level of maintenance efforts. In a similar study, [S<sub>17</sub>] uses graph-based visualization to visualize the evolution of class hierarchies.

#### 3.4.2 Notation-Based Visualizations

The result obtained indicates that only one study [S<sub>5</sub>] use notation-based technique. The technique relates to the specific notation-based visualization, one of the three modeling techniques defined in Shahin et al. (2010). Unlike UML and SysML (system modeling language), specific notation-based visualizations are new, customizable notations developed for a

specific purpose in software visualization. The work reported in [S5] visualizes the internal structure of the quality of a software entity and the relation between them.



**Figure 5:** Proportion and number of studies per visualization approach

### 3.4.3 Metaphor-based Visualizations

In the metaphor visualization based, the visualization process is more intuitive and effective (Bentrad and Meslati, 2011), improves the perception of quality (Bohnet and Dollner, 2011) and facilitates diagram interpretation (Varetl et al., 2013). This is because metaphor-based visualization uses familiar physical world properties to visualize software entities and their relationships. In this study, about 29% of the studies adopt this approach [S3, S4, S11, S16, S18]. Two of these studies [S4, S16] employ software maps, which visualizes classes as “buildings” and packages as “districts”, to monitor and maintain the current state of source codes and their dependencies. On a similar note, [S3] uses city metaphor to represent the function of a class as “buildings” and height and position of the “buildings” as lines of code and complexity respectively.

### 3.4.4 Matrix-based Visualization

The main goal of matrix based techniques is to provide additional information for large graphs. Similar to notation-based visualizations, matrix based is employed only by one study [S1]. Matrix-based visualization is used in [S1] understand the dependencies of software to external API libraries. This visualization provides information on the role of specific API's and the degree of API dependency between packages. This visualization use tabular representation to compare information on the role of specific API's. The degree of dependency provides information on specific role of the API.

## 3.5 Purpose of Visualization techniques

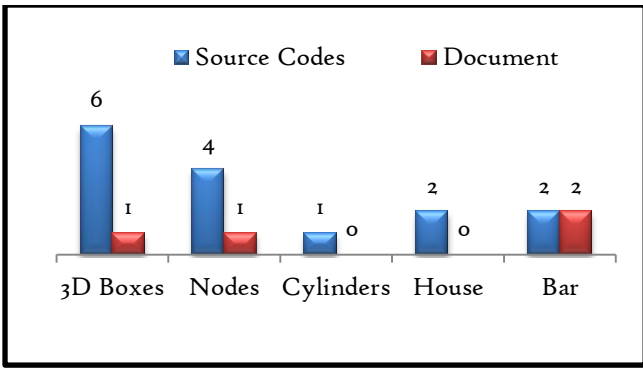
This section present finding relating to data item D5 which answers RQ4: *What are the different objectives for employing visualization techniques?* We use the thematic approach presented in Section 2.5 and identified five classifications based on the motivation reported in the studies for the adoption of a particular visualization techniques. They are presented below together with the number of studies in each category:

- *Category 1* – Improve the understanding of dependencies between software entities [S13 S10 S5 S1]
- *Category 2* – Improve the understanding of structural characteristics of software entities [S17 S16 S15 S14 S12 S11 S9 S8 S7 S5 S4 S3 S2]
- *Category 3* – Improve the understanding of software evolution [S11]
- *Category 4* – Enhance collaboration of software development [S18]
- *Category 5* – Provide traceability between software entities and software specification [S6]

The result indicates that of the 18 included studies, the majority focus on the static properties of the software entity under investigation, to get an understanding of its level of quality. The results are as expected because quality aspects related to performance of software entity were not included in the study. The last two categories contain only one study each and this might indicate the emergence of new areas to visualize software quality metrics. Finally, the second category has more studies than category four because software entities are made up of different components. Components can be further broken down to desired level of granularity (classes, methods, interfaces) which communicate or depend on each other.

### 3.1 Visual attributes

In this section we answer RQ 5. Using the thematic analysis, we classified visual attributes into two categories: visual attributes for the data and visual attributes for the metrics. The results are graphically represented in figure 6 and figure 7 to indicate the number of studies that use visual attributes to map the data and metrics respectively.



**Figure 6:** Visual attributes mapped to the data

It should be observed that because some studies have more than one view in the visualization, they also use more than one visual attribute for the same data and metric, the sum exceeds the number of the reviewed studies.

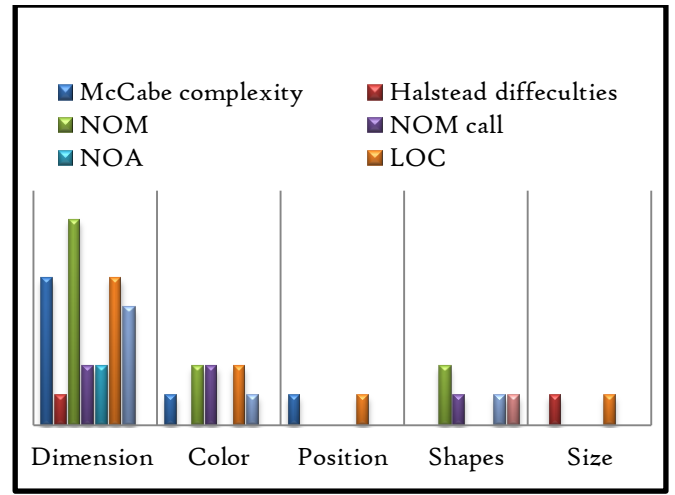
### 3.1.1 Visual attributes for the data

Figure 6 shows that source codes are the most frequently used data. All studies mapped classes to one or multiple shapes. From the figure, we can observe that 3D boxes (38.8%) and nodes (28%) are preferred over the other types of shapes. We realize that the type of VT determines the kind of visual attribute mapped to the data. Boxes and bars are mostly associated with graphical visualizations [S3, S4] while houses/buildings are related to metaphor-based visualizations [S11, S18].

### 3.1.2 Visual attributes for the metrics

Dimension, such as height, width, length and depth, is the most frequently used visual attribute for quality metrics. Our results indicate that 14 studies, which accounts for about 78%, mapped one or more dimensional aspects to a quality metric. The results from this is consistent and as expected because 3D boxes has multiple dimensions - height, depth, width - that are easily mapped as visual attributes for quality metrics. In addition, the dimensions of an object or shape are fairly easy to compute, implement, and process by the human visual system.

As shown in figure 7, NOM is mostly mapped to the visual attribute of dimension (7 studies). The results also indicate that LOC is the only quality metric that is mapped to all the visual attributes, except the visual attribute “shape”.

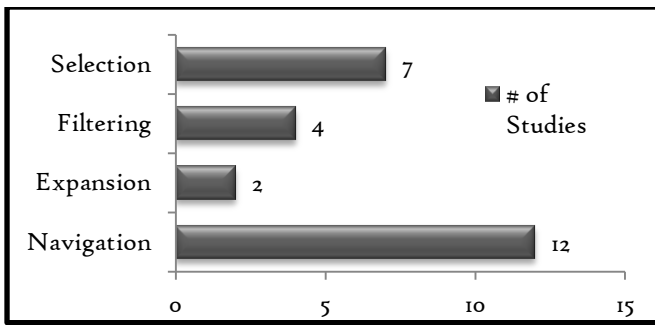


**Figure 7:** Visual attributes mapped to quality metrics

Only a few studies map position to quality metrics. We observe that in different studies, different visual attributes are used for a similar quality metric. For instance, lines of code (LOC) are mapped to the height [S3, S11, S16] and to the width [S18] while McCabe’s complexity is mapped to the height of roof [S18] and to the position [S3]. Unlike other studies [S3, S6, S11, S12, S15, S16, S17, 18], color is not the prime attributes for metrics although it was acknowledge as influential and sparsely used in the primary studies.

## 3.6 Mechanisms of Interactions

Interactions in visualizations have significant implication because it enhances the interpretation of the visual data. The results obtained from the data are not surprising. Apart from [S2] and [S6], all other studies incorporated a number of interaction mechanisms in their visualization. We classified interactions into 4 categories based on the thematic approach (see figure 8). Navigation comprises of interactions such as zoom, mouse hovering, rotation and movement, to the desire object of focus. Selection and expansion are used to show detailed information on demand or an informative summary of the entity. Filtering was implemented to simply view and express of the object under analysis.



**Figure 8:** Mechanisms of interaction used in the studies

### 3.7 Evaluation done on study

As explained in Section 2.5, we analyzed the evaluations carried out on the different VTs according to three categories – by *type*, *aspect* and *outcome* of the evaluation. The findings to this section provides answers to RQ7 as shown in table 3, the evaluations reported from these studies aim to improve task comprehension and analysis within a particular context and a given set of data. 13 studies (72%) conducted use cases to validate their VTs, with the primary goal to demonstrate how to use the VTs. The outcome for most VTs was effective except for 2 studies [S8, S13] for which the outcome was not as expected. For instance, [S8] performed an evaluation with the hypothesis that their visualization technique will improve the accuracy and time it takes to carry out a given task. However, the outcome indicated otherwise because the margin of the accuracy and time it took to complete a task, compared to text based approach, was statically insignificant. [S13] had concerns with scalability instead.

## IV. DISCUSSION

We reviewed 18 studies and proposed classifications that aid to answer the research questions presented in Section 2.1. We have investigated the visualization approaches for quality metrics and the evaluations performed on them. Our review indicates that the number of studies is decreasing. We believe that despite its significance, the field of quality metrics visualization remains challenging for two reasons. Firstly, quality metrics have been elaborated and expanded in many different directions making it difficult to provide a unified and coherent picture (Bertini et al., 2011). Secondly, there is no one given approach for a

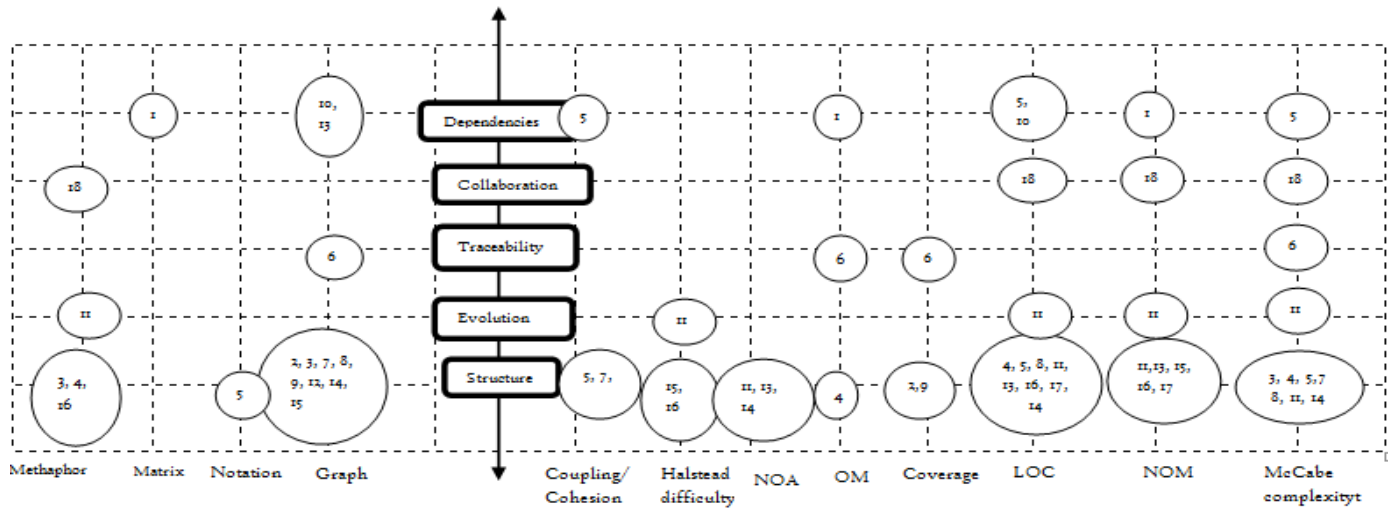
specific purpose. The latter has led to increase in the number of visualization techniques that aim to provide efficient ways to communicate and visualize quality metrics in an interactive manner within a specific context.

The remainder of this section is focuses on highlighting the main findings and presents the threats to validity in the study.

We found that a large number of studies used the graph-based visualization technique, to visualize, especially, the structural properties of a software entity. We also found out that on average, the visualization approaches presented in the studies contain a number of views – global and specific views. The global view is similar to a dashboard, and is adopted to get an overview and identify potential hotspots within the structure of the software entity. The specific view on the other hand, is used to refine and examine these hotspots in detail to address the areas that require greater attention. The Kiviat diagram fulfills this purpose best. In addition to views, interactions are also key elements that enhance the interpretation of information and the most common type of interaction identified was navigation.

For researchers, figure 8 also aids in the identification of gaps and possible future research areas. It is evident that some of the different approaches have been underexplored, e.g., the matrix-based visualization approach. We found that 3 studies visualized the level of quality of software documents as opposed to classes or packages. In order to reduce maintenance costs and increase product quality, it is also important to investigate the quality aspects of software documents as well. Furthermore, only one study used the graph-based approach to visualize trace the coincidence between specification and implementation.

We also realized that the amount of collaboration in the area of quality metric visualization is rather low. The majority of the studies that use classes as their visual data did not benchmark their findings against other approaches. Instead, they introduce a novel approach to analyze quality metrics rather than expand or add value to already existing approaches. Besides that, the mapping of the same quality metrics, within similar contexts, to different visual attributes indicates lack of collaboration in this area.



**Figure 8:** Quality Metrics x Visualizations x Goal

A systematic use of common terminology and the mapping of visual attributes will organize and foster better guidelines for future research or practices.

A number of studies complemented their proposed approach with validations to illustrate task comprehension and analysis and on the visual presentation. The most common type of evaluation is use cases, which demonstrates how to analyze and interpret the results. However, although the outcome in many studies focuses on obtaining insights from the approach, they do not suggest guidelines on how the insights will be used. In some studies, the validations are carried out on the visual aspects of the approach. We consider this to be rather weak, because although this affects the human visual system, there is no guarantee it will enhance task comprehension or analysis. Instead, issues such as scalability and generalizability should be prime concerns in the evaluation of visualization approaches.

For practitioners, this review offers different visualization techniques that examine and enhance the visualization of software quality metrics from a given context. It is necessary to take this into consideration when embarking on a specific visualization approach. We suggest that practitioners should examine and analyze their contents and characteristics of their projects before tailoring the desired visualization approach. In order to enhance data interpretation, we recommend practitioners to (i) map visual attributes to the quality metrics; (ii) adopt a variety of visualization views within the visualization approach; (iii) apply some

mechanisms of interactions as deemed necessary.

## V. THREATS TO VALIDITY

In this section we present the threats to validity of our study and a detailed description on the strategies used to mitigate them.

### 5.1 Threats to the search string

The fact that our research string was general and broad does not indicate that we have addressed all previous studies. Since our search strings were broad, 8636 of studies were retrieved from the database. However, there is always a risk that some relevant papers were not retrieved by the search string. Also, the authors might have missed some relevant papers because (i) we excluded ACM and Springerlink digital library because we could not retrieve massive amount of studies, we could only download one study at a time; (ii) we did not have access to more than 1000 studies from ScienceDirect digital library; (iv) we did not conduct an analysis to find which digital libraries best fit the field of our research topic; and (v) We did not conduct a snowballing due to time limitation of the study. However, because this study is conducted within a short time frame, the authors did not mitigate these threats. Hence, this threat is assessed as moderate.

### 5.2 Study Selection

To avoid bias in the study selection phase and to minimize the risk of excluding relevant studies, we developed a review protocol that consists of several steps of selection process. The protocol was reviewed and discussed with a supervisor, who has previous experiences in systematic

literature review. The review protocol was followed strictly by the authors as the following:

- Collaboration during the study selection phase.
- The study was excluded when both authors agree that the title, abstract or the full text do not fulfill the inclusion criteria.
- Consult a supervisor if there are differences in opinions about certain articles.
- We applied and reviewed the inclusion and exclusion criteria on 109 studies in Filter 2 (see fig 1), with the supervisor.
- Further the list of both excluded and included studies were reviewed by the supervisor so, there are less likely to exclude relevant papers unintentionally.

### 5.3 Data extraction

To minimize the bias in the data extraction phase, the authors created a data extraction table to be filled in with the relevant information. To reduce the bias, both authors read all the 18 primary studies independently and filled in the data extraction form. Later on, the authors discussed the results of each study. The supervisor provided his opinion when there were certain differences between the authors' opinions.

### 5.4 Publication bias

This threat is considered insignificant, because the aim of our study is to identify the different software quality metrics, and they are not categories as positive or negative is just a mean of representing and gathering information of visualizing software quality metrics. Hence, all the identified software quality metrics will be published. Moreover, it is possible that some relevant studies were missed during the study selection process. This could be mitigated by snow-balling the primary studies. However, we did not mitigate this issue.

## VI. CONCLUSION

To gain a holistic overview of the visualization approaches for software quality metrics, we

performed a systematic literature review. We systematically selected 18 studies, based on the inclusion and exclusion criteria, and thematically analyzed them to answer our research question: "*what software quality metrics visualization techniques are supported in existing studies, and what types of evaluations have been performed on them?*" Despite the variation of the types and purposes of these approaches, the results reported in this paper enable us to make the following conclusions:

- The purpose of quality is to measure the maintainability aspects of a software entity and the most widely used metrics of quality are, in descending order; lines of code, McCabe's complexity, and number of functions.
- The majority of studies employ visualization techniques mainly to improve the understanding of the structural aspects for software entities. Graph-based visualizations technique is the most widely used approach for this purpose. And to better interpreted the visualized results, the data and metrics are mapped to different visual attributes.
- A large percentage of the studies validate their approaches through use cases focusing on task comprehension and analysis activities.

However, there is room for future research. Firstly, we could investigate the use of notation-based visualization to improve the understanding of dependencies of software modules, rather than graph-based visualization. Notations are best used for describing the behavioral aspects of software entity especially their relationships or dependencies.

Secondly, only three studies visualized software documents. Thus, more research can be conducted on how the quality attributes of software documents like readability and understandability can be measured and visualized.

Finally, we could transfer the knowledge gained from this study into an industrial setting. The main objective will be to validate in practice how these results can facilitate the development

and implementation of quality metric visualization.

## **Acknowledgements**

Our gratitude goes to the company and the contact persons, for their collaboration and assistance in carrying out this research. We equally thank our supervisor for his guidance and support throughout the research period. Finally, we thank the staff at the department for their suggestions and comments.

**Table 5:** Appendix A. Detailed information of the primary studies

Num	Title	Author(s)	Venue/ Acronym	Year
[S <sub>1</sub> ]	Understanding API Usage to Support Informed Decision Making in Software Maintenance	Veronika Bauer, Lars Heinemann	European Conference on Software Maintenance and Reengineering ( CSMR)	2012
[S <sub>2</sub> ]	A Metrics -Based Approach to Technical Documentation Quality	Anna Wingkvist, Morgan Ericsson, Rudiger Linke, Welf lowe	International Conference on the Quality of Information and Communications Technology (QUATIC)	2010
[S <sub>3</sub> ]	METRIX: a new tool to evaluate the quality of software source code	Antoine Varet, Nicolas Larriou, Leo Sartre	AIAA Infotech@Aerospace (I@A) Conference(AIAA)	2013
[S <sub>4</sub> ]	Monitoring Code Quality and Development Activity by Software Maps	Johannes Bohent, Jurgen Dollner	IEEE/ACM Workshop on Managing Technical Debt (ICSE)	2011
[S <sub>5</sub> ]	Equality: A Graph Based Object Oriented Software Quality Visualization Tool	Ural Erdemir, Umut Tekin Feza Buzluca	IEEE International Workshop on Visualizing Software for Understanding and Analysis. (VISSOFT)	2011
[S <sub>6</sub> ]	Improvement of a Visualization Technique for the Passage Rate of Unit Testing and Static Checking and its Visualization	Yuko Muto, Kozo Okano Shinji Kusumoto	Joint Conference of the Software Measurement. Workshop on Software Measurement and Conference on Software Process and Product Measurement. (IWSM/MESURA)	2011
[S <sub>7</sub> ]	Exploring the Evolution of Software Quality with Animated Visualization	Guillaume Langelier, Hourai Sahraoui, Pierre Poulin	IEEE Symposium on Visual Languages and Human-Centric Computing. (VL/HCC)	2008
[S <sub>8</sub> ]	Supporting the evolution of software visualization tool through usability studies	Marcus, A., Comorski, D., Sergeyev, A.	International Workshop on Program Comprehension. (IWPC)	2005
[S <sub>9</sub> ]	Smart views for analyzing problem reports: tool demo	Patrick Knab, Harald Gall, Martin Pinzge	European Software Engineering Conference andthe ACM SIGSOFT Symposium on the Foundations of Software Engineering. (ESEC/FSE)	2009
[S <sub>10</sub> ]	Sextant: A Tool to Specify and Visualize Software Metrics for Java Source-Code.	Victor Winter, Carl Reinke Jonathan Guerrero	International Workshop on Emerging Trends in Software Metrics. (WETSOM)	2013
[S <sub>11</sub> ]	Software Visualization with Audio Supported Cognitive Glyphs	Sandro Boccuzzo, Harald C.Gall	IEEE International Conference on Software Maintenance. (ICSM)	2008
[S <sub>12</sub> ]	Software Metrics in Static Program Analysis	Andreas Vogelsang, Ansgar Fehnker, Ralf Huuck, Wolfgang Rief	International Conference on Formal Engineering Methods. (ICFEM)	2010
[S <sub>13</sub> ]	Understanding the Use of Inheritance with Visual Pattern	Simon Denier, Houari Sahraoui	International Symposium on Empirical Software Engineering and Measurement .(ESEM)	2009



[S14]	WikipediaViz: Conveying Article Quality for Casual Wikipedia Readers	Fanny Chevalier, Stephane Huot Jean-Daniel Fekete	IEEE Pacific Visualization Symposium (PacificVis)	2010
[S15]	Visualization of Coupling and Programming to Interface for object-Oriented System	Peter Rosner, Srikumar Viswanatha	International Conference Information Visualization (IV)	2008
[S16]	2D and 3D Visualization of AspectJ Programs	Sassi Bentradi, Djamel Meslati	International Symposium on Programming and Systems (ISPS)	2011
[S17]	Characterizing the Evolution of Class Hierarchies	Tudor Girba, Michele Lanza Stephane Ducasse	European Conference on Software Maintenance and Reengineering. (CSMR)	2005
[S18]	An approach for Collaborative Code Reviews using Multi Touch Technology	Sebastian Muller, Micahael Wursch, Thomas Fritz, Harald C. Gall	International Workshop on Cooperative and Human Aspects of Software Engineering. (CHASE)	2012

## References

- Z. Antolic. "An example of using key performance indicators for software development process efficiency evaluation. Technical report". *Technical Report, R&D Center, Ericsson Nikola Tesla dd*, 2008.
- V. Bauer and L. Heinemann. "Understanding API usage to support informed decision making in software maintenance". In *Software Maintenance and Reengineering (CSMR), 2012 16th European Conference on, IEEE*, pp. 435-440, March. 2012.
- S. Bentradi and D. Meslati. "2D and 3D visualization of aspectj programs". In *Programming and Systems (ISPS), 2011 10th International Symposium on, IEEE*, pp.183-190. April 2011.
- S. Boccuzzo and H. C. Gall. "Software visualization with audio supported cognitive glyphs". In *Software Maintenance, 2008. ICSM 2008. IEEE International Conference on, IEEE*, pp. 366-375, Oct.2008.
- J. Bohnet and J. Dollner. "Monitoring code quality and development activity by software maps". In *Proceedings of the 2nd Workshop on Managing Technical Debt, ACM*, pages 9-16, May.2011.
- H. P. Breivold, I. Crnkovic, and M. Larsson. "A systematic review of software architecture evolution research". *Information and Software Technology*, 54(1):16-40, January.2012.
- F. Chevalier, S. Huot, and J.-D. Fekete. "Wikipediaviz: Conveying article quality for casual wikipedia readers". In *Pacific Visualization Symposium (PacificVis), 2010 IEEE, IEEE*, pp. 49-56, March.2010.
- S. Denier and H. Sahraoui. "Understanding the use of inheritance with visual patterns". In *Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on, IEEE*, pp.79-88, Oct.2009.
- U. Erdemir, U. Tekin, and F. Buzluca. "E-quality: A graph based objectoriented software quality visualization tool". In *Visualizing Software for Understanding and Analysis (VISSOFT), 2011 6th IEEE International Workshop on. IEEE*, pp.1-8. Sept.2011.
- A. Fehnker, R. Huuck, A. Vogelsang, and W. Reif. "Software metrics in static program analysis". In *International Conference on Formal Engineering Methods*, pp.485-500, Shanghai, China, Nov 2010. Springer
- J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971
- T. Girba, M. Lanza, and S. Ducasse. "Characterizing the evolution of class hierarchies". In *Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on, IEEE*, pp. 2-11, March. 2005.
- D. Gracanin, K. Matkovic, and M. Eltoweissy. "Software visualization". *Innovations in Systems and Software Engineering*, 1(2):221-230, July.2005.
- Q. Gu and P. Lago. "Exploring service-oriented system engineering challenges: a systematic literature review". *Service Oriented Computing and Applications*, Springer-Verlag, 3(3):171-188, September.2009.
- B. Kitchenham, R. Pretorius, D. Budgen, O. Pearl Brereton, M. Turner, M. Niazi, and S. Linkman. "Systematic literature reviews in software engineering—a tertiary study". *Information and Software Technology*, 52(8):792-805, August. 2010
- P. Knab, P. Knab, and M. Pinzger. "Smart views for analyzing problem reports: tool demo". In *Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, ACM, pp. 289-290, Aug.2009.
- G. Langelier, H. Sahraoui, and P. Poulin. "Exploring the evolution of Software quality with animated visualization". In *Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on, IEEE*, pp.13-20, Sept. 2008.
- A. Marcus, D. Comorski, and A. Sergeyev. "Supporting the evolution of a software visualization tool through usability studies".

In *Program Comprehension, 2005. IWPC 2005. Proceedings. 13th International Workshop on*, IEEE, pp.307–316, May. 2005.

S. Muller, M. Wursch, T. Fritz, and H. C. Gall. “An approach for collaborative code reviews using multi-touch technology”. In *Cooperative and Human Aspects of Software Engineering (CHASE)*, 2012 5<sup>th</sup> International Workshop on, IEEE, pp.93–99, June. 2012.

Y. Muto, K. Okano, and S. Kusumoto. “Improvement of a visualization technique for the passage rate of unit testing and static checking and its evaluation”. In *Software Measurement, 2011 Joint Conference of the 21st Int’l Workshop on and 6th Int’l Conference on Software Process and Product Measurement (IWSM-MENSURA)*, IEEE, pp.279–284, Nov.2011.

R. L. Novais, A. Torres, T. S. Mendes, M. Mendonca, and N. Zazworka. “Software evolution visualization: A systematic mapping study”. *Information and Software Technology*, 55(11):1860–1883, November .2013

P. Rosner and S. Viswanathan. “Visualization of coupling and programming to interface for object-oriented systems “. In *Information Visualisation, 2008. IV’08. 12th International Conference*, IEEE, pp.575– 581, July. 2008.

M. Shahin, P. Liang, and M. R. Khayyambashi. “Improving understandability of architecture design through visualization of architectural design decision”. In *Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge*, ACM, pages 88–95. May.2010.

A. Shollo and K. Pandazo. “Improving presentations of software metrics indicators using visualization techniques”. rapport nr.: Report/IT University of Goteborg 2008: 020, 2008

B. Singh and S. P. Kannoja. A review on software quality models. Technical report, IEEE, 2013.

M. Staron, J. Hansson, R. Feldt, A. Henriksson, W. Meding, S. Nilsson, and C. Hoglund. “Measuring and visualizing code stability—a case study at three companies”. In *Software*

*Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSP-MENSURA)*, 2013 Joint Conference of the 23<sup>rd</sup> International Workshop on, IEEE, pp.191–200. Oct. 2013.

E. Taylor-Powell, M. Renner. “Analyzing Qualitative Data”. *Program development and evaluation. University of Wisconsin-Extension, Cooperative Extension*, 2003

A. Varet, N. Larrieu, L. Sartre, et al. “Metrix: a new tool to evaluate the quality of software source codes”. In *AIAA Infotech@ Aerospace (I@A Conference)*, 2013.

A. Wingkvist, M. Ericsson, R. Lincke, and W. Lowe. “A metrics-based approach to technical documentation quality”. In *Quality of Information and Communications Technology (QUATIC)*, 2010 Seventh International Conference on the, IEEE, pp.476–481, Oct. 2010.

V. Winter, C. Reinke, and J. Guerrero. “Sextant: A tool to specify and visualize software metrics for java source-code”. In *Emerging Trends in Software Metrics (WETSoM)*, 2013 4th International Workshop on, IEEE, pp.49–55, May.2013