the JSF template pages (Facelets) are described. In the last step the JSF source code for the managed beans and the Facelets is generated [9], [24].

2.4. Metrics

As defined in the title, the main task of this bachelor thesis deals with the extension of a dashboard functionality. A project dashboard can be defined as a tool for the visualization of different project metrics (see section 2.5). Therefor metrics are an important basis of this bachelor thesis.

The term ‘metric’ is intuitively related to the terms ‘measure’ or ‘measurement.’ Hence, the term ‘metric’ is often explained in literature by using the terms ‘measure’ or ‘measurement’ and vice versa [30]. However, measurement is defined in ISO/IEC 15939 as a *“set of operations having the object of determining a value of a measure.”* [19] The IEEE defines the term ‘metric’ as *“a quantitative measure of the degree to which a system, component, or process possesses a given attribute.”* [17] Hence, a metric is a mapping of a measured property to a scalar or vectorial scale [25], while a measurement is simply a value.

Metrics are an objective assessment of an artifact and determine the quantitative characteristics of an artifact. This means, that the artifact characteristics are mapped to a numeric value or a vector of numeric values. Different artifacts have a different degree of structuring. It is more difficult to represent poorly structured artifacts in a metric than the well-formalized artifacts [3].

Referring to project development, metrics are *“objectively measurable attributes”* [8] used to provide information about the project state. In the management area metrics are important *“decision-support tools”* [10] used to validate, control and report collected data. They help to find an effective and efficient direction of the development and organization process and make it more predictable.

Software Metrics

In software engineering the term ‘metric’ is extended by the term ‘software metric’ and uses to determine quality properties of software system and the software development process during its lifecycle. Software metrics help to estimate the required quality properties like maintainability, efficiency, flexibility, portability etc. as well as software development costs, project scheduling or risk register [8].

In [25] an ‘ideal’ metric is defined as a metric, which provides an accurate characterization of a measured software object, restricted to the measured property. The authors define seven main requirements for the software metrics: differentiability, comparability, reproducibility, availability, relevance, profitability and plausibility. They also present the following categorization of the software metrics, which are classified according to their application area.

- **Cost metrics**. Cost metrics are used to measure the project development effort, required human resources and project development time.
- **Error metrics**. Error metrics measure the number of detected or expected errors.
- **Volume metrics**. Volume metrics measure the size of a software entity.
- **Quality metrics**. Quality metrics are used to measure the quality of software.

Software metric are also used to determine requirements of systems and for the prediction of errors and defects in code. Fenton [10] defined three other types of software metrics:

- **Process metrics**. Process metrics are used to measure characteristics related to the development process of a software system. In this case, time is the central measuring characteristic, but also human resources and process schedules.
- **Product metrics**. Product metrics measure characteristics, which refer to products. The measured objects are specifications, code or test data. Characteristics are for instance lines of code (LOC) or identified faults in code and tests.
- **Resource metrics**. Resource metrics measure available resources. The measured objects are characteristics like age or memory size.

The Goal Question Metric Approach

Since the range of quality characteristics that are required from the system, is large, it can be difficult to define proper metrics to cover all significant information needs.

There are many studies, which researched the problem defined above and suggest possible approaches as a solution. An approach developed by Basili, Cardela and Rombach, which is called Goal Question Metric (GQM), became very popular. The GQM approach is based on templates, which help to define measurement goals. The measurement model obtained by the GQM approach describes a measurement system and rules for the data interpretation, where goals, question and metrics are related to each other in a certain way. The measurement model of the GQM approach is divided into three levels, which are illustrated in figure 2.5 [4].

On the first level, called conceptual level (see fig. 2.5), the measurement **goals** for the visual quality characteristics are defined. This characteristics include software artifacts, software activities and resources. On the operational level, a set of questions is generated, which serve to interpret and understand the goals specified on the conceptual level. The answers to the questions are defined in

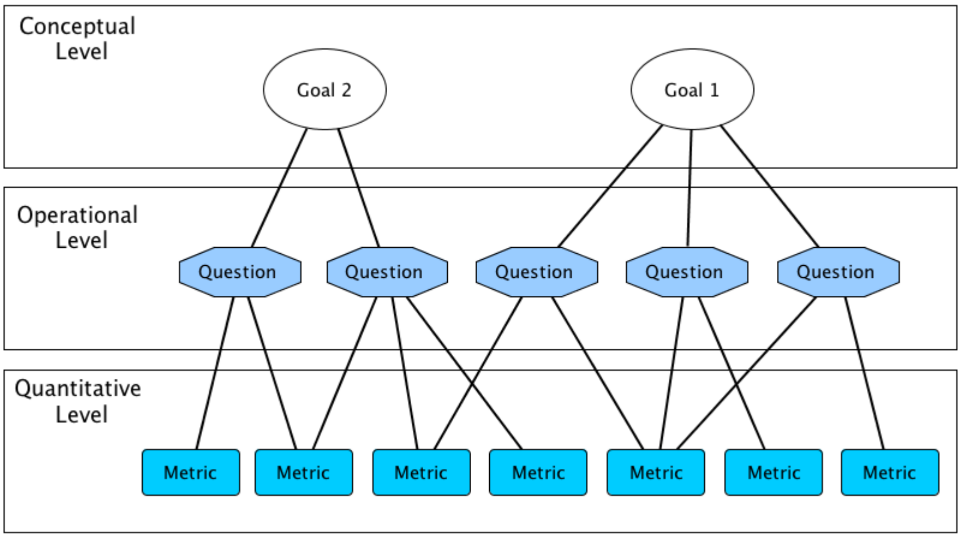

Figure 2.5.: The hierarchical Structure of the Goal Question Metric Model (from [4]).

the form of metrics on the third level, called quantitative level.

The GQM approach proceeds top-down and starts with the identification of metrics for the defined goals. In the next step the bottom-up proceeding is performed, where the results, that are obtained by the measurement, are validated, analyzed and interpreted [4].

2.5. Dashboard

A Dashboard (or also cockpit, control and management cockpit) is a general term for a software tool, which is employed for visualization, monitoring, analysis and evaluation of information needs. The information needs displayed on the dashboard are the different predefined metrics represented as text and graphics [11].

In the previous chapter metrics were defined as a tool to support management decisions, that allows to predict the development process. In reality, the different numerical values of metrics can be hard to interpret and analyze. A dashboard combines and illustrates the collected data values in an appropriate form, which allows to find critical or important information and to achieve objectivity. Moreover, the data represented as a graphic or an image in the dashboard are more “powerful” for the human awareness and can be better understood and quicker processed than numbers [12].

Classification of Dashboards