5.1. Vision

Most of the currently available solutions share similar use cases. They primarily offer the users the ability to observe a set of project metrics and create reports without actually providing the levers to control the project. Figure 5.1 shows a vision for the main use cases and actors of the conceptual cockpit. The use cases are presented from the point of view of the cockpit users.

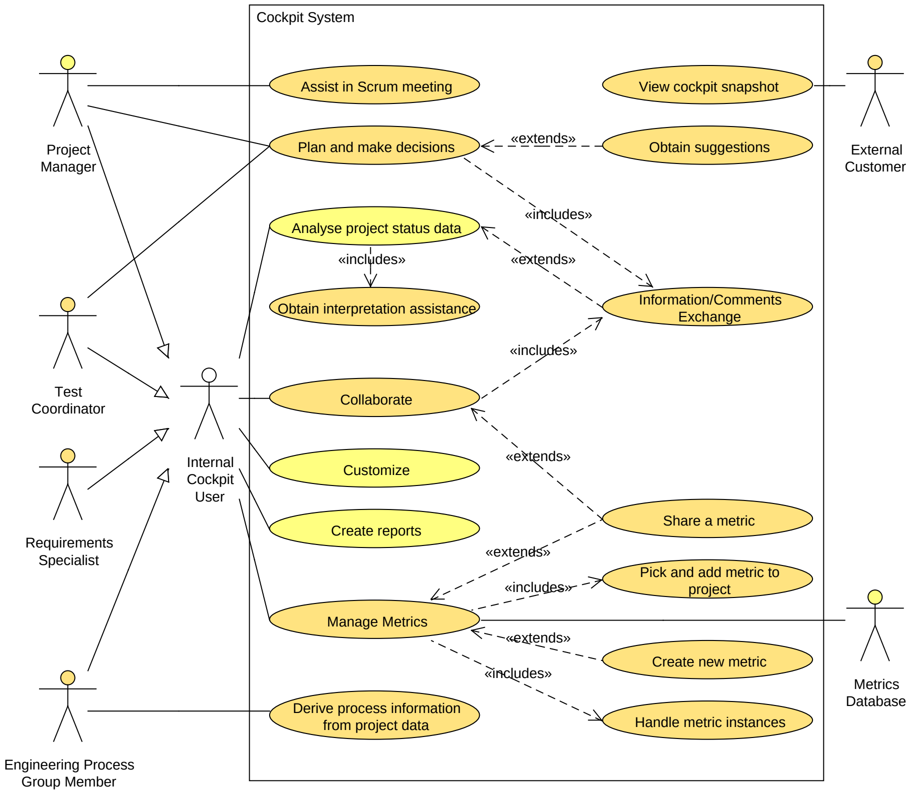

Figure 5.1.: Use cases of the conceptual cockpit (UML diagram)

The most limited set of actions is available to the external customers. They have access only to a dashboard - a snapshot of the cockpit. Operations besides viewing the data are not provided. None of the solutions that were analyzed in chapter 4 had identified the external customers as potential users. The role was discovered in the process of requirements engineering (see chapter 3) and corresponds to the needs of the industry.

Company-internal cockpit users, such as test coordinators and requirements specialists, have access to much more functionality (see figure 5.1). They are able to create metrics, add them to a selected project and manage their visibility and granularity. Internal cockpit users can collaborate within the cockpit by sharing metrics they have created and by exchanging information and comments. They can further on analyze the project data with the interpretation support of the cockpit system or with the help of the other members. Different default project visibility sets could be defined for the specific roles. All internal cockpit users are able to create reports. They can also customize the look of their cockpit.

The project manager can use all the features available to standard internal cockpit users, but can also use the cockpit for making decisions and controlling the overall project progress. Where applicable, the cockpit system assists the project manager by making suggestions for actions, based on the observed trends. The decisions can then be documented in the system and communicated to the other project members. The cockpit can be additionally used by the project manager during Scrum meetings to show the current state of a project. A more limited planning and decision-making actions set is available to test coordinators as well.

Members of the Engineering Process Group (EPG) are interested in the the improvement of the development process. They need to focus on the metrics themselves and not on their exact values for a concrete project. They can use the cockpit to manage the available metrics stored in a metrics database and analyze their usage, efficiency, etc. Besides the features available to standard internal cockpit users, EPG members can use the cockpit system to derive process improvement information and best practices from the project data.

In order to realize this functionality, a deeper understanding of the project cockpit domain is required. The next section presents a metrics meta model.

5.2. Metrics meta model

The most important entities within the project cockpit domain and their relations are presented here. The aim of this section is to give an overview of some of the properties of the conceptual cockpit and to establish a common understanding of the terminology. The model could be used as a guideline for a cockpit implementation.

Figure 5.2 shows the main entities of the cockpit metrics meta model. A central entity is the metric (refer to section 2.2 for the definition of a metric and more information on project metrics). A metric calculator corresponds to each concrete metric. It "knows" the formula for the calculation of the metric. Importing the raw data into the cockpit system is performed by specialized importers, communicating directly with the data sources. Data sources could be databases, repositories or the cloud. Storing data in the cloud could prove to be beneficial to a cockpit system.

Beside values, metrics are associated with various other types of data. Each has

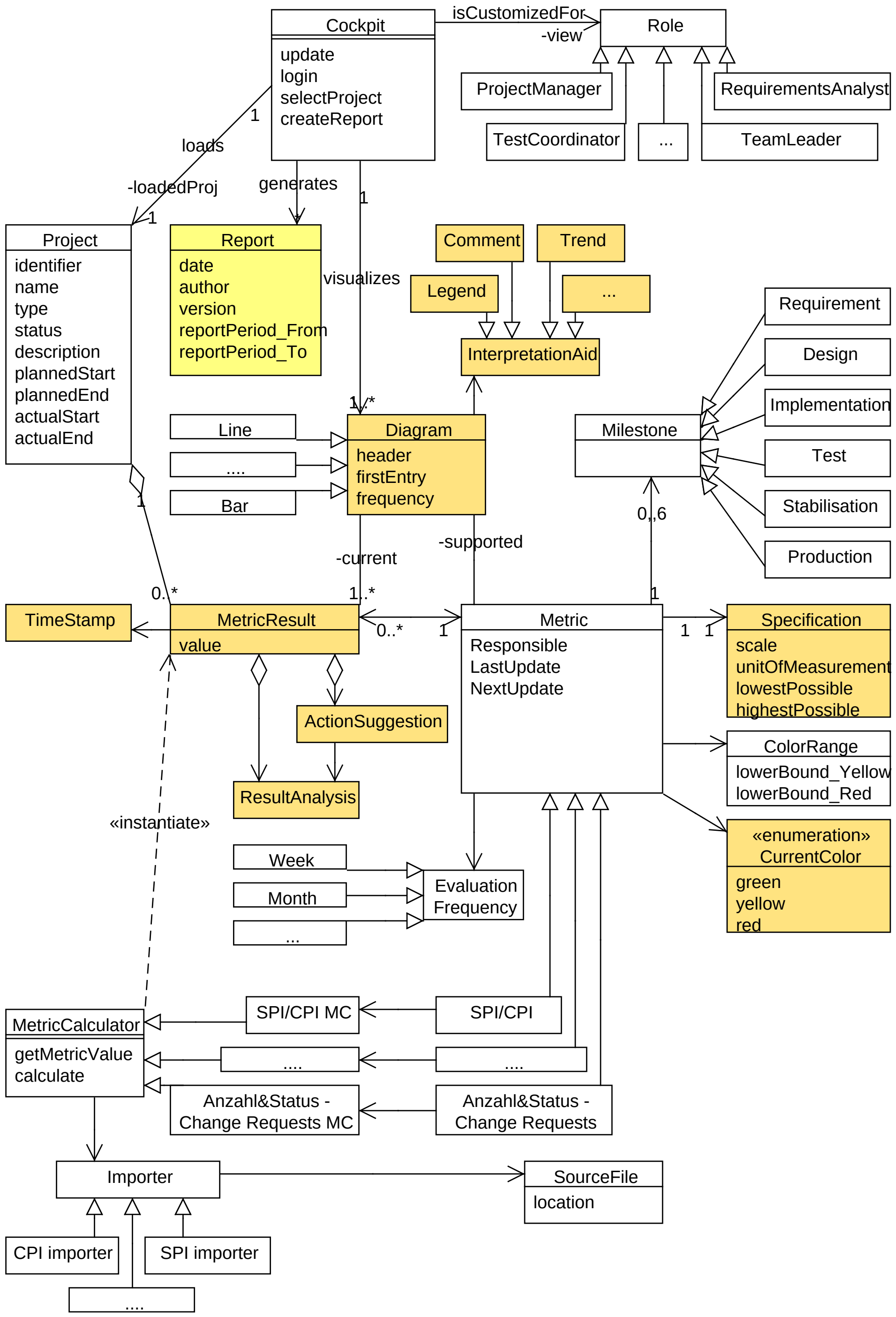

Figure 5.2.: Metrics meta model of the conceptual cockpit (UML diagram)

an evaluation frequency, which denotes the period of time between two automatic calculations of the metric. Information, regarding the properties of the metrics, such as the scale type (for example, nominal, interval, relational scale and so forth), the unit of measurement, etc. is stored in a metric specification.

To assist the interpretation of a metric, fixed thresholds for their values can be defined. For example, if the value of a given metric goes below a certain minimum threshold, it is considered unacceptably low. Traffic light colors can be assigned to the value ranges. The colors are used in the user interface to direct the attention of the analyst to the metrics with problematic values in a glance. As further support to the cockpit user, the possibility to conduct metrics analysis and to make notes within the cockpit is offered. Actions for handling problems can be suggested and discussed by the project members.

One of the fundamental features of the cockpit is visualization of data. The cockpit contains multiple diagrams that display metrics. Each metric has a set of possible diagram types for their visualization. The set of possible diagram types depends on the chosen charting framework.Diagrams are accompanied by interpretation aids, such as a legend, trend interpretation, user comments, etc.

The cockpit system supports various account types, as described in section 5.1. Project managers, test coordinators, requirements analysts, members of the engineering process group and external customers have access to different sets of features. Authorization is required for loading a project. After login, the user obtains a view of the project that corresponds to the user's role, i.e, the corresponding set of metrics and their granularity, the set of allowed features within the cockpit system as well as the assigned access to collaboration data. Reports can be generated from the data in the cockpit.

5.3. Central features

Some of the main features that go beyond those of a standard cockpit solution are presented in this section.

Extension of the metrics framework
Metrics are stored into a central data warehouse, from where they can be accessed by all team members. A team member can define a new metric - its calculation, purpose, interpretation, supported visualization types, possible measurement frequencies, etc. This metric can then be added to a project by other team members as well. Once added to a project, the metric is "instantiated" with the data from the concrete project. The personal project view of a team member comprises of containers, to which a metric can be loaded.

Role-specific cockpit views
Every project role has a specific view on the project data. When a new member is assigned to a project, he/she gets a role-specific cockpit, which contains the

default metrics, relevant to the role of the project member. Additional metrics, shared by the other project members are also displayed.

Role-specific metrics, shared metrics and granularity
A project members might define metrics, visible only to themselves. This feature might be used, for example, when "trying out" a new metric. In this case, the metric might also be saved locally. Project members can define metrics, relevant to all members of a specific role as well. Such metrics are persisted in the common data warehouse to facilitate their reuse. For example, a test manager could create a metric, which is also visible to the project manager. A project manager does not need to view the metrics of the test manager in the same detail, therefore the cockpit should support the setting of role-specific granularity of the metrics.

Apart form making metrics available to certain roles, they could also be shared temporarily with a given team member on demand. A project manager could ask a test manager to share a metric in order for the project manager to obtain additional insight on the reasons for a trend. Publishing and sharing metrics could improve the visibility into a project through collaboration and easier exchange of information.

Collaboration
In order to react to the trends and make decisions, project members could exchange information and opinions, discuss tendencies and measures. The multiple benefits of collaboration are discussed in 2.5. The cockpit supports this types of collaboration by allowing the project members to communicate inside the cockpit. This feature is not meant to replace face-to-face human communication, but is intended to support the quick exchange of specific kinds of information. It can be helpful in the process of decision making, where backtracking of the reasons for taking a decision is important.

Assistance and metrics interpretation
The cockpit assists the interpretation of metrics, diagrams and trends using visual and textual means. As found in the requirements engineering phase, such assistance is of highest importance to new project members, but could be helpful for people, using the cockpit regularly as well. Visual aids, such as traffic light coloring of the metric results, could make a diagram understandable at a glance. Setting markers inside a diagram might be used to add additional information to a diagram. For example, a marker could denote a value that needs to be analyzed and discussed.

Textual assistance introduces cockpit users to the meaning of a metric. More than one type of information can be derived usually from a single diagram. The information needs of the project members differ as well. Legends and role-specific textual aids could therefore be used to support easier goal-oriented interpretation of metrics diagrams.

Customization

Each project member's cockpit can be customized beyond the choice of displayed metrics. Metrics, their interpretation aids, discussion block, etc. are kept in a container whose position can be adjusted. Further personalization settings, such as color schemes, could also be thought of.

5.4. Logical Architecture

Figure 5.3 presents a three-layer solution for the architecture of the conceptual cockpit, addressing the earlier discussed aspects. The elements of the logical architecture are abstract components. A more detailed look into the layers of the architecture is offered below.

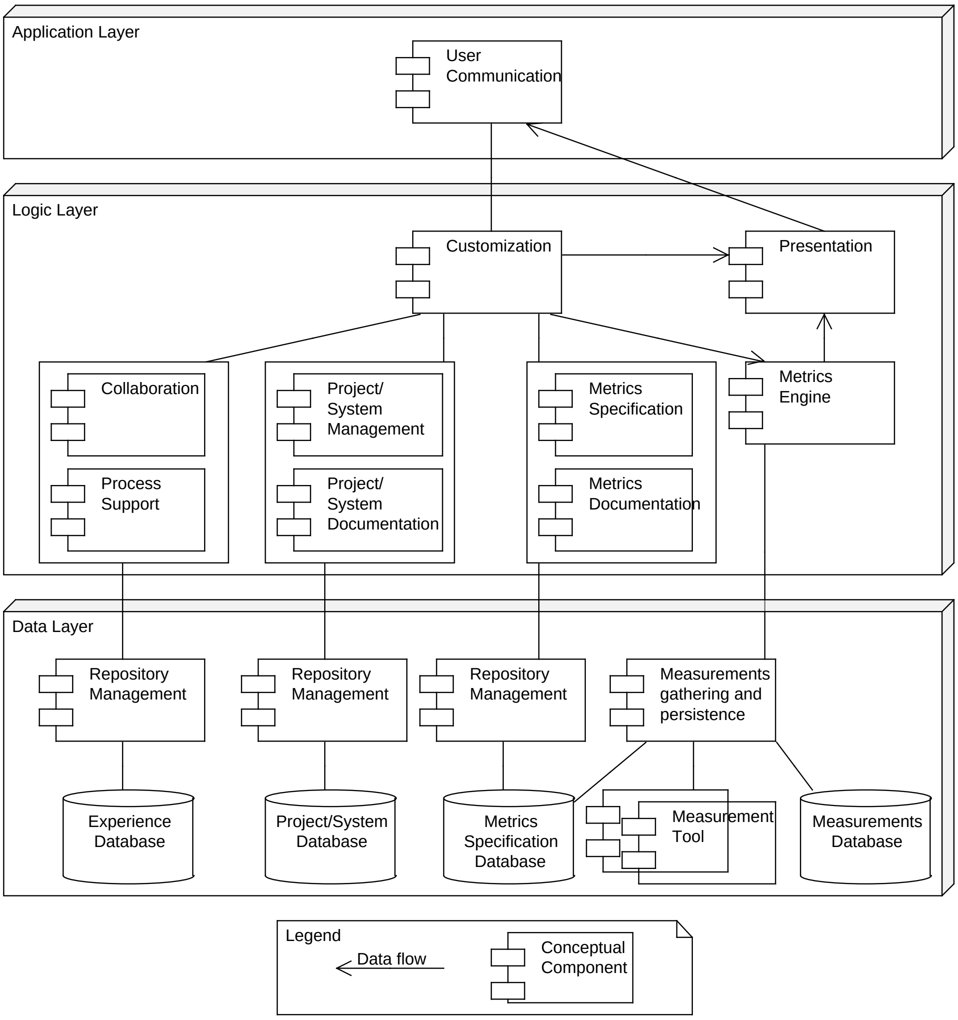

Figure 5.3.: Logical Architecture (UML diagram)

Data layer

The data layer is responsible for the retrieval and persistence of data, required by the cockpit. It handles the communication with the cockpit data storage unit and with the external source data systems. The source data usually originates from the databases of effort-tracking, bug-tracking, customer relationship management (CRM) systems, enterprise resource planning (ERP) systems and other operational systems. Measurement data could be obtained from other external sources, such as measurement tools. The latter is performed by the "Measurements gathering and persistence" component. Project data, raw measurements, metrics specifications and past experience are kept separately. A data warehouse solution (see section 2.4) is suggested to be used. On top of the data layers is placed the logic layer.

Logic Layer

The logic layer contains the functionality of the cockpit system. A central component is the "Customization" component, which coordinates the work of the components in the logic layer and actively communicates with the application layer. It delegates tasks to other components in the logic layer and prepares the data that needs to be visualized. How this data is visualized is decided in the "Presentation" component. It gets the project data, collaboration data and metrics documentation from the Customization component. The values of the metrics are calculated based on the gathered measurements by the "Metrics engine" and then passed on to the "Presentation" component. The latter then passes the data to the application layer.

Application Layer

The interaction with the user is handled by the "User communication" component. It gets the data, prepared for visualization by the logic layer, and displays it to the user. The actions made by the user within the cockpit are passed back to the "Customization" component for processing. An action could be, among others, the choice of a project, the choice of a team role, filtering the data, drill-downs, adding comments to a diagram, changing the order of the diagrams, personalizing the cockpit, defining new metrics, etc. The reaction of the cockpit to those actions is decided within the logic layer.