Effective dashboard design

Article in Cutter IT Journal · January 2013

Andrea Janes
FHV Vorarlberg University of Applied Sciences

Alberto Sillitti
Centre for Applied Software Engineering

Effective Dashboard Design

by Andrea Janes, Alberto Sillitti, and Giancarlo Succi

In the 19th century, the term "dashboard" was already being used to refer to a board in front of a carriage that stopped mud from being splashed (dashed) into the vehicle by the horse's hooves. Later, cars began using dashboards to inform the driver about the status of the car's various systems. When a problem arises, the colors of the dashboard indicators show how urgent the matter is (see Figure 1):

- **Red** indicators typically mean that the problem is serious and that some action is needed immediately. Red indicators such as "Check Engine" or "Low Oil" require the driver to halt the car right away to prevent further damage. Safety issues, such as a nonworking air bag, are also shown using red lamps.
- **Yellow** indicators show that some action is required soon, such as the yellow "Low Fuel" light.
- **Green** indicators inform the driver that some system is turned on, such as the "Low Beam Lights."

The dashboard is designed to ensure the correct functioning of the car. It is, so to speak, aligned to the business goal of the driver. It helps the driver achieve the goal, which is to drive from point A to point B.

In an organization, the term "dashboard" is used to describe a system that visualizes data useful for decision making.[1] Dashboards, as in a car, have the goal of informing while not distracting users from their actual task. Therefore, data in dashboards is summarized using charts, tables, gauges, and so on (see Figure 2). To allow users to interpret an element on the dashboard correctly, dashboards typically allow them to see the original data on which the summarization was based.

Some authors distinguish between dashboards and scorecards,[2] depending on whether the data measures performance (dashboard) or charts progress (scorecard). We treat the terms "dashboard" and "scorecard" as synonyms.

THE IDEAL DASHBOARD

Dashboards visualize data. Ideally, dashboards are useful. They are useful if they support their users in fulfilling their goals. Unfortunately, many dashboards are not designed to be useful.[3] Instead, they are designed to visualize as much data as possible, to demonstrate the graphical abilities of the dashboard, to impress potential customers, and so on.

To obtain a useful dashboard, our approach focuses on two aspects: selecting the "right" data and choosing the "right" visualization technique:

1. To choose the right data, we develop a measurement model (see next section) that defines *which* data we collect, together with the reasons *why* we collect it. For example, we could define that we collect McCabe's cyclomatic complexity of every method (what) because by using that measurement, we can decide to what extent we have to test that code (why). Once the collected data is linked to the reason *why* we need it, it is possible to correctly interpret the data and reuse it for future projects, since we are able to put it into the correct context.[4]

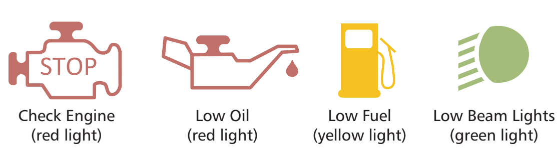

Figure 1 — Typical indicators in a car dashboard.

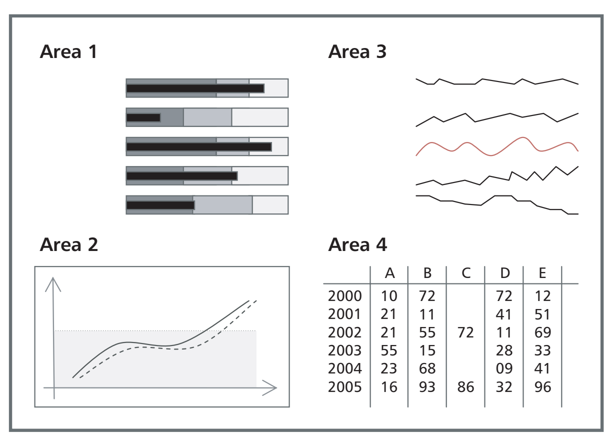

Figure 2 — A typical dashboard (adapted from Few).

2. The next step is to choose the right visualization techniques. We focus on visualizations that minimize the time needed to understand what has to be communicated.

HOW TO CHOOSE THE "RIGHT" DATA

Generally speaking, we can measure everything that we can observe. Some things can be observed with our senses, while some things — such as the Higgs boson — require the use of expensive equipment such as the Large Hadron Collider in Geneva. Not everything that *can* be measured should be measured. Measurement costs, so the decision of what to measure has to be based on the expected benefit of the measurement.

By creating a measurement model, we document — starting from the measurement goal — which questions we want to answer and which data we have to collect to answer them. To achieve this, we have adopted the GQM+Strategies approach,[5] which is based on GQM (Goal-Question-Measurement) models.[6] We will explain the latter first.

A GQM model is defined on three levels:

1. The **goal** — the conceptual level — defines what we want to study and why. What is studied is the "object of study," the specific products, processes, and resources. Why something is studied identifies the reason, the different aspects taken into consideration, the considered points of view, and the environment.
2. The **questions** — the operational level — define (a) what parts of the object of study are relevant, and (b) what properties of such parts are used to characterize the assessment or achievement of a related goal. These properties are often called the "focus" of the study.

   Altogether, the questions specify which specific aspects of the object of study are observed to understand whether the goal is achieved or not. Questions are measurable entities that establish a link between the object of study and the focus. For example, if the object of study is a car and the focus is its environmental impact, a question could be: "How high are the car's carbon dioxide emissions?"
3. The **measures** — the quantitative level — define which data has to be collected to answer the questions in an objective (quantitative) way.

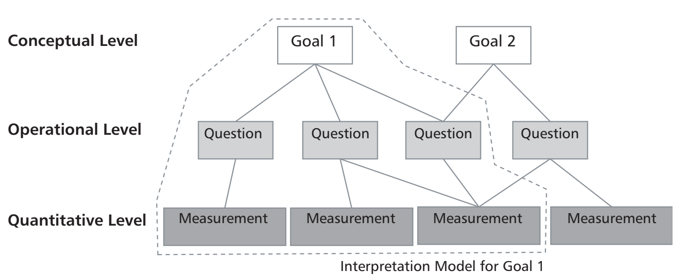

Figure 3 — A GQM model.

These three levels are complemented by an interpretation model that defines how to interpret the collected data to evaluate the measurement goal.

It is important not to confuse the focus with the point of view. The focus is the part of the object of study that is studied. It is an objective view on the object of study. The point of view describes who is measuring and represents the subjective view of the measurement goal.

The GQM model is a hierarchy of goals, questions, and measurements (see Figure 3). This hierarchy details what is measured and how the results have to be interpreted.

Showing How It's Done with the King of Wines

To explain how a GQM model defines the collection and interpretation of data, in the following example we use the model to evaluate the taste of a glass of Barolo, an Italian red wine.

The model is depicted in Figure 4. A precise definition of the goal — evaluating a glass of Barolo wine — helps us to obtain data (answers to the defined questions) that has an impact on the goal.

In this example, we evaluate the taste using three criteria: the sweetness, the aroma, and the flavor of the wine. To objectively measure the sweetness of the wine, we use an electronic oscillating U-tube meter. To measure

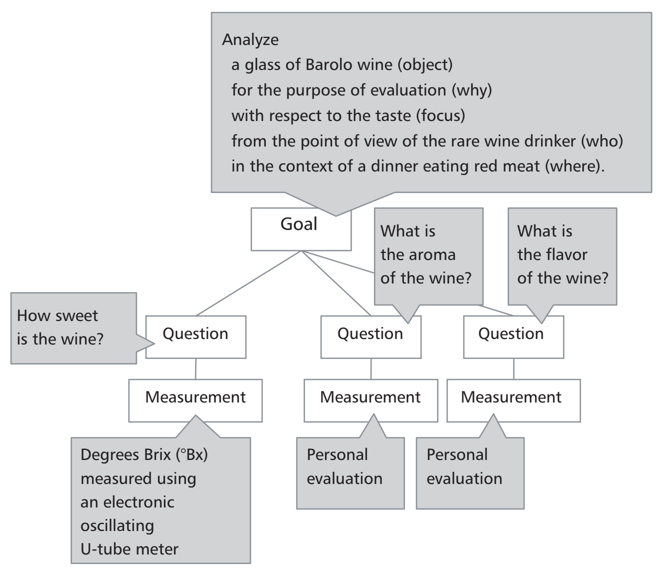

Figure 4 — A GQM model for evaluating the taste of a glass of Barolo wine.

aroma and flavor, we use the personal evaluation of our sommelier, who is an expert in the field. The sommelier's opinion is not that objective, but it is also not as subjective as asking a beginner.

The interpretation model for the example in Figure 4 has to state how the collected data answers the questions and how the answers to the questions influence the measurement result. The model has to specify how to combine the results to evaluate how well Barolo wine tastes to a rare wine drinker who is eating red meat.

As we said above, the definition of measurement goals is critical to the successful application of the GQM approach. To facilitate the definition of precise measurement goals, the authors of the GQM approach developed a goal template (see Figure 5). The goal template requires the dashboard's designer to state the purpose of the measurement (what it is measuring and why), the perspective (what specifically is observed, the focus, and from which point of view the observation is made), and the environment (in which context the measurement takes place).

The explicit formulation of the purpose, the perspective, and the environment helps us to understand which data is needed to fulfill the measurement goal and to understand how to interpret the collected data.

Once we have defined one or more goals, we define questions that characterize the goal in a quantifiable way and the measurements to describe the data that will be used to answer the questions. GQM questions can be classified into three groups:[7]

1. Questions that characterize the object of study with respect to the overall goal (e.g., "Is Barolo considered a superb, good, or miserable wine?")
2. Questions that characterize relevant attributes of the object of study with respect to the focus (e.g., "What is the aroma of the wine: spicy, smoky, oaky, etc.?" or "What is the sensation of the wine: sparkling, acidic, crisp, etc.?")
3. Questions that evaluate relevant characteristics of the object of study with respect to the [focus] (e.g., "Is the taste satisfactory from the viewpoint of a rare wine drinker?" or "Does the taste match well with the meat?")

After defining the questions, we have to define which measurements we are going to collect to answer them, as multiple measurements might be used to answer the same question. As we have seen before, the sweetness of wine can be measured using an electronic oscillating U-tube meter, but it could be also evaluated using a personal evaluation.

The selection of measurements to answer the developed questions depends on different factors, such as the amount and quality of data that is already available, the cost-benefit ratio of performing a specific measurement, the level of precision needed, and so forth.

A Hierarchy of Goals

So far we have not mentioned that measurement goals can occur on different levels in the organization. In fact, every activity within an organization is a means to an end, a part of the organizational strategy to achieve an organizational goal. In any organization we can observe a goal hierarchy: beginning from the main organizational goal, all subsequent goals are derived from the previous goal.

In the example in Figure 6, the organization decides to obtain a given financial goal through an increase in

Purpose

Analyze:	*a glass of Barolo wine* (objects: process, products, resources)
for the purpose of:	*evaluation* (why: to characterize, evaluate, predict, motivate, improve)

Perspective

with respect to:	*the taste* (focus: cost, correctness, changes, reliability, …)
from the point of view of:	*a rare wine drinker* (who: user, customer, manager, developer, corporation, …)

Environment

in the following context:	*a dinner eating meat* (where: environmental factors influencing the measurement)

Figure 5 — A GQM goal template.

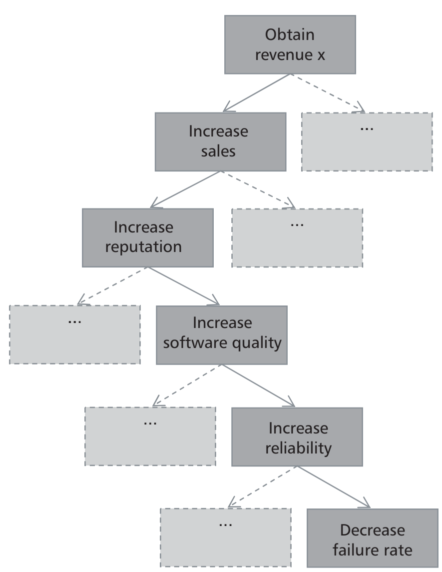

Figure 6 — A goal hierarchy.

sales, which should be achieved through an increase in the reputation of the company. (Alternative options at each step in the hierarchy would be shown in the dashed boxes.) The organization believes it can attain this increase in reputation by increasing its software quality. Moreover, in this example, the software department decides that it should achieve this through an increase in the reliability of the software, which implies a decrease in the failure rate.

Such a goal hierarchy fulfills two functions. First, it describes abstract goals in more detail; for example, "increase reliability" better describes what is meant by "increase software quality." Second, it describes means-end relationships, such as to increase the reputation of the company being the means to increase sales.[8]

Deciding how a given goal is to be achieved — what means will be used to achieve it — is defined by the strategy. The organization accepts the strategy as the right approach to achieve its goals, given its context and assumptions.[9] The strategy explicitly states what steps the organization has to take to obtain the desired goal.

At this point, we see how the GQM$^+$Strategies approach extends the GQM model. It considers the goal hierarchy motivated by the organizational strategy and creates a measurement model that links business goals to measurement goals. Figure 7 illustrates the concept: every element of the goal hierarchy is linked to a GQM model that measures the achievement of the business goal at that level.

The GQM$^+$Strategies measurement model describes what we call the "ideal dashboard." It measures the achievement of the organizational strategy (i.e., the

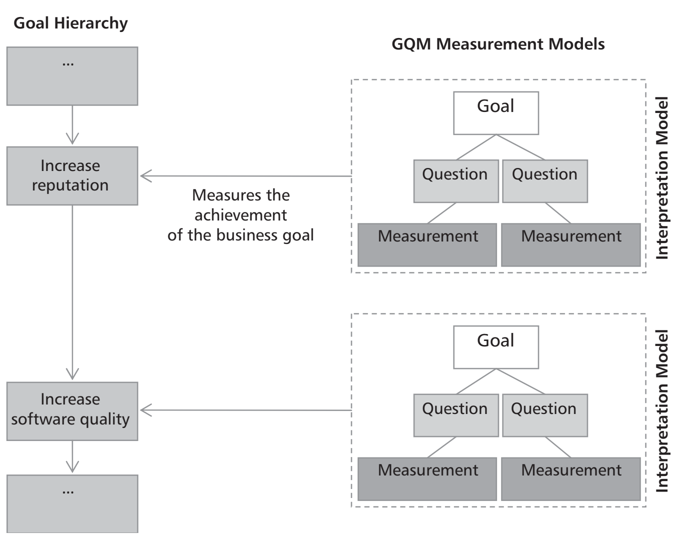

Figure 7 — A GQM$^+$Strategies measurement model.

business goals). To measurements it adds the context, meaning the reasons for collecting them, and the business strategy that justifies this.

It is not always feasible to elaborate a detailed GQM$^+$Strategies measurement model; for example, if the strategy is frequently changing as a reaction to a volatile market. As we said earlier, measurement costs. Therefore it is up to the organization to decide which level of monitoring and what kind of information are worth collecting and visualizing.

Authors typically distinguish between strategic, tactical, and operational dashboards. Seen from the perspective of the GQM$^+$Strategies measurement model, a strategic dashboard visualizes the achievement of the upper goals of the goal hierarchy, a tactical dashboard deals with the goals in the middle, and an operational dashboard handles the most detailed goals at the bottom.

It is out of the scope of this article to discuss *how* the data required to evaluate the measurement models is obtained. We recommend automating this step as much as possible using noninvasive measurement.[10, 11]

HOW TO CHOOSE THE "RIGHT" VISUALIZATION

Dashboards can be designed in a variety of ways. There is no one right or wrong way — it depends on the requirements the dashboard has to fulfill. In this section we want to discuss two usage scenarios for a dashboard: "pull" and "push."[12]

In the pull scenario, the user wants to get a specific piece of information and uses the dashboard to obtain it. In such a case, aspects of technology acceptance become important, such as the dashboard's perceived usefulness and perceived ease of use.[13] Important considerations include:[14]

- The dashboard should help the user to understand the context of the data: why it was collected, how it should it be interpreted, how it can be used in future projects, etc.
- The dashboard should help the user understand the meaning of the data. Visualizations should require minimal effort to get the conveyed message, be coherent, allow the user to choose the level of detail of the data, etc.

In the push scenario, the dashboard has to be designed so that important information is pushed to the user. That is, it must capture the user's attention and inform him.

Whether a dashboard is more suited to the push or pull scenario depends on how much effort a user has to invest to see the dashboard. A dashboard that pushes the information to the user has the advantage of informing him in unexpected, unforeseen situations about problems, anomalies, and the like. A dashboard that is designed to support the pull scenario should offer more possibilities to explore the data, to filter and to search, to investigate the reasons that caused the data, and so on.

To set up a dashboard that is used in a push scenario, we found the following considerations important:[15, 16]

- The user should be able to see the dashboard without any effort. For example, in a car, the dashboard is built in such a way that it is in the range of vision of the driver. An organizational dashboard should be displayed on a monitor in the corridor or somewhere in the office where many are passing by. The information will be pushed to the users without their active participation. An example for such a dashboard is the Andon board, used in lean manufacturing, which is placed so that everybody can see if there is a problem on the assembly line (see Figure 8).
- The user should not need to interact with visualizations to understand the data. The charts have to be designed so that an interaction is only necessary when the user switches into "pull" mode (i.e., the dashboard got the attention of the user and she wants to investigate further).
- Arrange the data to minimize the time needed to consult the dashboard. Always place the same information in the same spot. Allow the user to develop habits; for example, every morning as he passes by with his coffee in hand, he can check the current size of the error log displayed in the upper-right corner of the dashboard.
- Guide the attention of the user to important information. There are different mechanisms you can use to draw the attention of the user, but make sure not to overuse them. If everything on the dashboard is blinking, the user will ignore it.
- Since we want users to look at the dashboards by choice, there are also aesthetic factors to consider. Displaying dashboard elements in a visually appealing way can increase the user's interest in looking at the dashboard.

To highlight important data, we use a technique called "pre-attentive processing." Researchers have identified different graphical properties (grouped into form, color, motion, and spatial position) that cause people to process information before they are even paying conscious attention to it.[17] Pre-attentive processing elements have the advantage of being processed (i.e., understood) faster than elements that are not pre-attentive.[18]

In Figure 9, both boxes a and b contain numbers. If we look at box on the top (Figure 9a) and try to count the number of 3s, we have to process the numbers sequentially. That is, we have to look at each number separately and decide if it resembles the form of the number 3.

It is much easier to count the number of 3s in the bottom box (Figure 9b). This is because we identify color/shading differences faster than the meaning of a symbol.

How strongly something is noticed pre-attentively depends on how different the highlighted element is from the others and how different the other elements

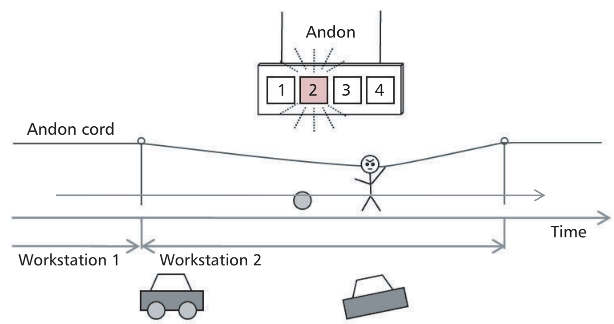

Figure 8 — An Andon board in lean manufacturing.

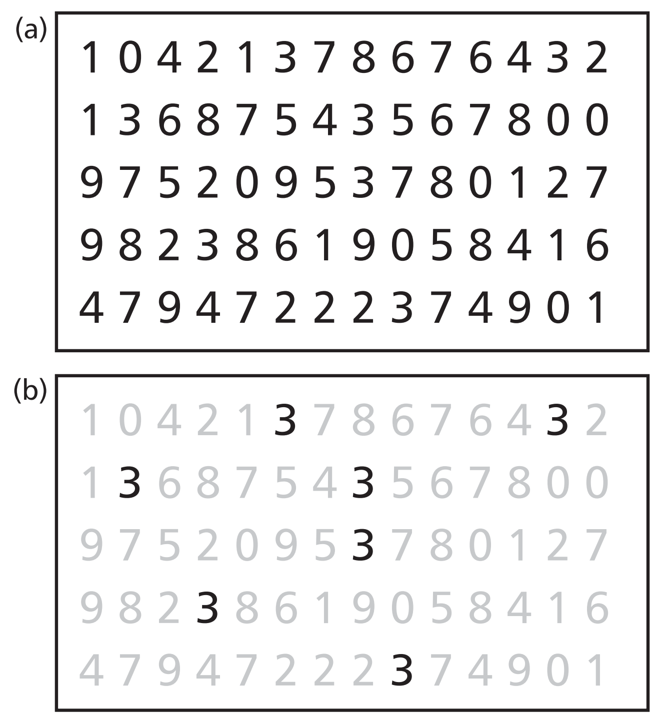

Figure 9 — How many 3s are present in the box?

are from each other. Moreover, combining two pre-attentively processed properties (such as color and shape in Figure 10) doesn't work. The information cannot be processed pre-attentively but will again require sequential processing.

COMBINING THE "RIGHT" DATA WITH THE "RIGHT" VISUALIZATION TECHNIQUES

In this section, we present a dashboard we designed for a software development team, combining the concepts discussed above. The starting point for the dashboard's development was the development of the goal hierarchy. The top business goal for the team was to increase development productivity. Starting from that, we developed subgoals such as "Improve component reuse" and "Reduce average development effort."

For each business goal, the development team together with the project manager developed measurement goals and modeled them as GQM models (see Figure 11 for an example).

To display the measurements, we then used colored tiles that visualize the outcome of the measurement, the trend (if the value is decreasing, stable, or increasing over time), and the classification of the measurement as "good" (green), "warning" (amber), or "critical" (red).

It is also possible to use shades of gray, patterns, or a different line thickness if the dashboard has to be accessible to color-blind people. For example, in the US, the Section 508 Amendment to the Rehabilitation Act of 1973 states that "Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element." While this legal requirement applies only to US federal agencies, it is a good idea to add a second distinctive element, such as line thickness, or to choose colors that are different enough that they become distinctive shades of gray for a color-blind person.

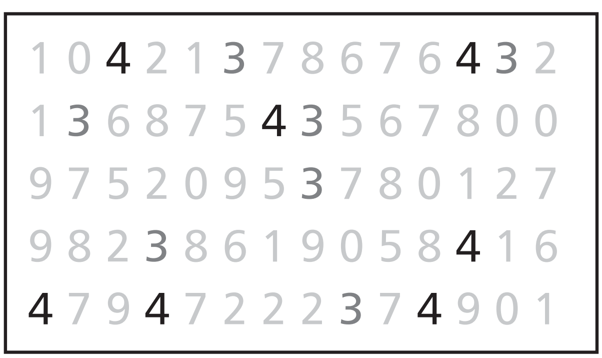

Figure 10 — How many 3s and 4s are present in the box?

The tile in Figure 12 is red (or dark gray if depicted in shades of gray), meaning that the measurement is classified as "critical" and requires the attention of the team. The name of the measurement is "A," and the current value is 10. The arrow shows that since the last time the measure was evaluated, the value has increased.

Figure 13 depicts a dashboard based on such tiles. Each tile represents the outcome of a measurement. The measurements are grouped with their questions and goals. We use colors to depict the status of a measurement to allow users to look the dashboard using pre-attentive processing. The idea is that green (light gray) tiles do not need to be read; they can be ignored. Amber (medium gray) and red (dark gray) tiles require attention; developers and managers should have a look at them.

The dashboard represents the measurement goals connected to one business goal. In our dashboard implementation, the values displayed within a measurement tile can either originate from actual data or be the result of another measurement goal.

Figure 14 illustrates how the dashboard described here allows users to navigate through the GQM+Strategies measurement model. In Step 1, the user clicked on the red tile related to the business goal "Increase reputation." That tile is not calculated using actual data but rather is the result of a measurement goal that belongs to the business goal "Increase software quality." Therefore, after clicking on the tile, the user obtains the GQM model for the business goal "Increase software

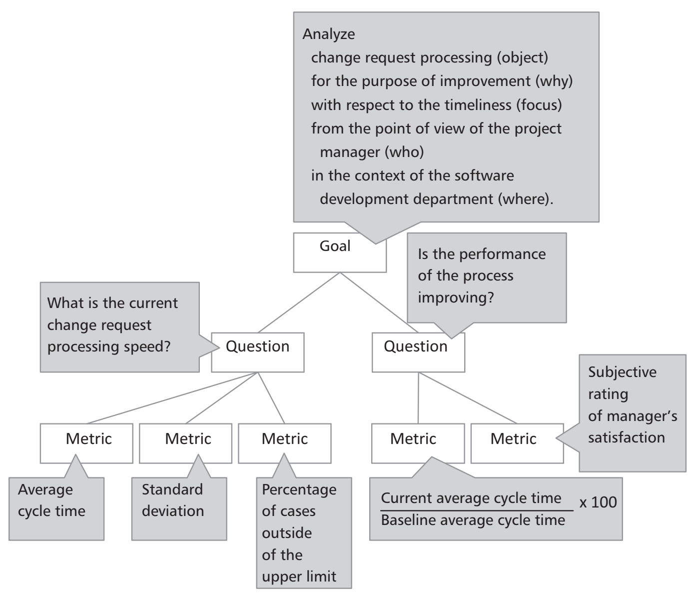

Figure 11 — GQM model to evaluate the timeliness of change request processing.

quality." Moreover, the user sees why the tile of the business goal "Increase reputation" was marked "critical": one of the tiles of the business goal "Increase software quality" is critical.

Tiles do not necessarily need to be connected to other GQM models. In Figure 14, Step 2, the user clicks on the red tile that is the result of an analysis performed over the source code repository. This analysis identified a problem with a module stored in the repository and reports a quality issue using a bar chart.

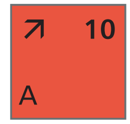

Figure 12 — A tile that represents a measurement outcome.

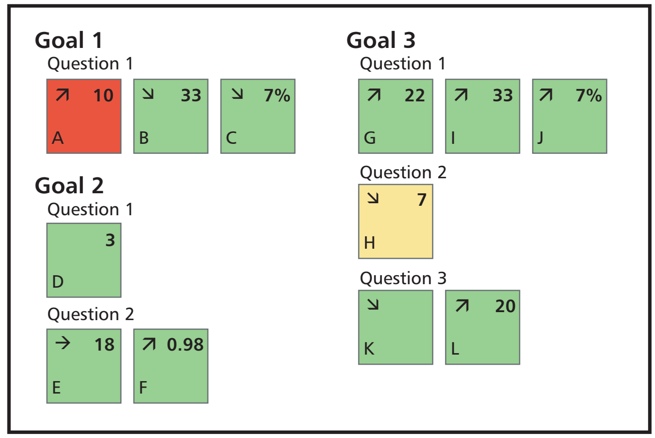

Figure 13 — A dashboard showing the measurement outcomes as tiles.

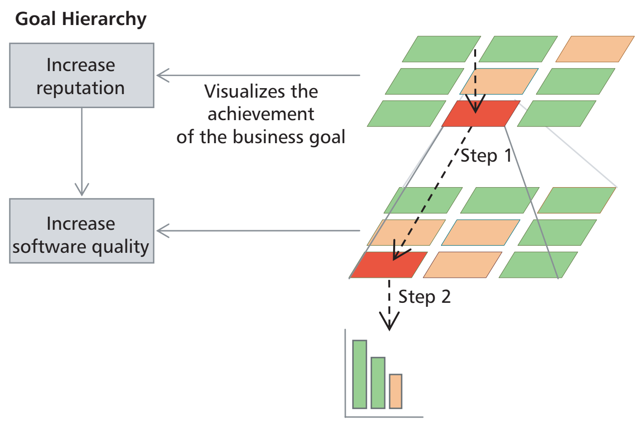

Figure 14 — Possible drill-down path in the dashboard.

CONCLUSIONS

The dashboard we have described here illustrates how to use a GQM+Strategies measurement model as a basis for developing a dashboard that supports its users to achieve their business goals. In our case study, this implementation led to the following conclusions:

- A dashboard is successful to the extent that it follows the findings of the Technology Acceptance Model — that is, according to its perceived usefulness and perceived ease of use.
- Dashboards that follow a "push" approach get more attention than dashboards following a "pull" approach, but they allow less freedom in their design.
- For a dashboard to be useful, it has to contain data related to the business goals. The development of such a dashboard requires the involvement of management and experienced collaborators.[19]
- The development of a dashboard is a continuous process, since the organization — ideally — is always learning. It is important to constantly ask whether it is time to update the current business goals, assumptions, strategy, and measurement goals.
- The visualization style you adopt is crucial — it has a huge impact on the acceptance of the dashboard.

ENDNOTES

[1]Few, Stephen. *Information Dashboard Design*. O'Reilly Media, 2006.

[2]Eckerson, Wayne W. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. 2nd edition. John Wiley & Sons, 2011.

[3]Few. See 1.

[4]Rombach, H. Dieter, and Bradford T. Ulery. "Improving Software Maintenance Through Measurement." *Proceedings of the IEEE*, Vol. 77, No. 4, 1989.

[5]Basili, Victor, Jens Heidrich, Mikael Lindvall, Jürgen Münch, Myrna Regardie, Dieter Rombach, Carolyn Seaman, and Adam Trendowicz. "Linking Software Development and Business Strategy Through Measurement." *Computer*, Vol. 43, No. 4, April 2010.

[6]Basili, Victor R., and H. Dieter Rombach. "Goal Question Metric Paradigm." In *Encyclopedia of Software Engineering*. 1st edition. John Wiley & Sons, 1994.

[7]Basili, Victor R., Gianluigi Caldiera, and H. Dieter Rombach. "The Goal Question Metric Approach." In *Encyclopedia of Software Engineering*. 1st edition. John Wiley & Sons, 1994.

[8]Schwarz, Rainer. *Controlling-Systeme: Eine Einführung in Grundlagen, Komponenten und Methoden des Controlling*. Gabler Verlag, 2002.

[9]Basili et al. See 5.

[10]Danovaro, Emanuele, Andrea Janes, and Giancarlo Succi. "Jidoka in Software Development." *Proceedings of OOPSLA 2008*. ACM, 2008.

[11]Janes, Andrea, Alberto Sillitti, and Giancarlo Succi. "Non-Invasive Software Process Data Collection for Expert Identification." *Proceedings of the International Conference on Software Engineering & Knowledge Engineering*. Knowledge Systems Institute, 2008.

[12]Janes, Andrea, and Giancarlo Succi. "To Pull or Not to Pull." *Proceedings of OOPSLA 2009*. ACM, 2009.

[13]Venkatesh, Viswanath, and Hillol Bala. "Technology Acceptance Model 3 and a Research Agenda on Interventions." *Decision Sciences*, Vol. 39, No. 2, May 2008.

[14]Few. See 1.

[15]Few. See 1.

[16]Ware, Colin. *Information Visualization: Perception for Design*. 3rd edition. Morgan Kaufmann, 2012.

[17]Ware. See 16.

[18]Ware. See 16.

[19]For an example of tool support for collecting experience, see: Danovaro, Emanuele, Tadas Remencius, Alberto Sillitti, and Giancarlo Succi. "PKM: Knowledge Management Tool for Environments Centered on the Concept of the Experience Factory." *Proceedings of the 30th International Conference on Software Engineering*. IEEE, 2008.

Andrea Janes is Assistant Professor of Software Engineering at the Free University of Bolzano-Bozen (Italy). His research interests include lean software development, value-based software engineering, and empirical software engineering. Mr. Janes holds a master of science degree in business informatics from the Technical University of Vienna and is pursuing a PhD in informatics at the University of Klagenfurt (Austria). He can be reached at ajanes@unibz.it.

Alberto Sillitti is Associate Professor of Computer Science at the Free University of Bolzano-Bozen. He has been involved in several EU-funded projects related to open source software, services architectures, and agile methods in which he applies noninvasive measurement approaches. Additional research areas include mobile and Web services. Dr. Sillitti has served as a program committee member of several international conferences and as program chair of OSS 2007, XP2010, and XP2011. He is the author of more than 80 papers for international conferences and journals. Dr. Sillitti holds a PhD in electrical and computer engineering from the University of Genoa. He can be reached at asillitti@unibz.it.

Giancarlo Succi is a Senior Consultant with Cutteer Consortium's Agile Product & Project Management *practice. He is a Professor of software engineering and Director of the Center for Applied Software Engineering at the Free University of Bolzano-Bozen. Dr. Succi's research areas include agile methodologies, open source development, empirical software engineering, software product lines, software reuse, and software engineering over the Internet. He is the author of one book and more than 100 papers published in international conferences proceedings and journals. He can be reached at gsucci@cutter.com.*

