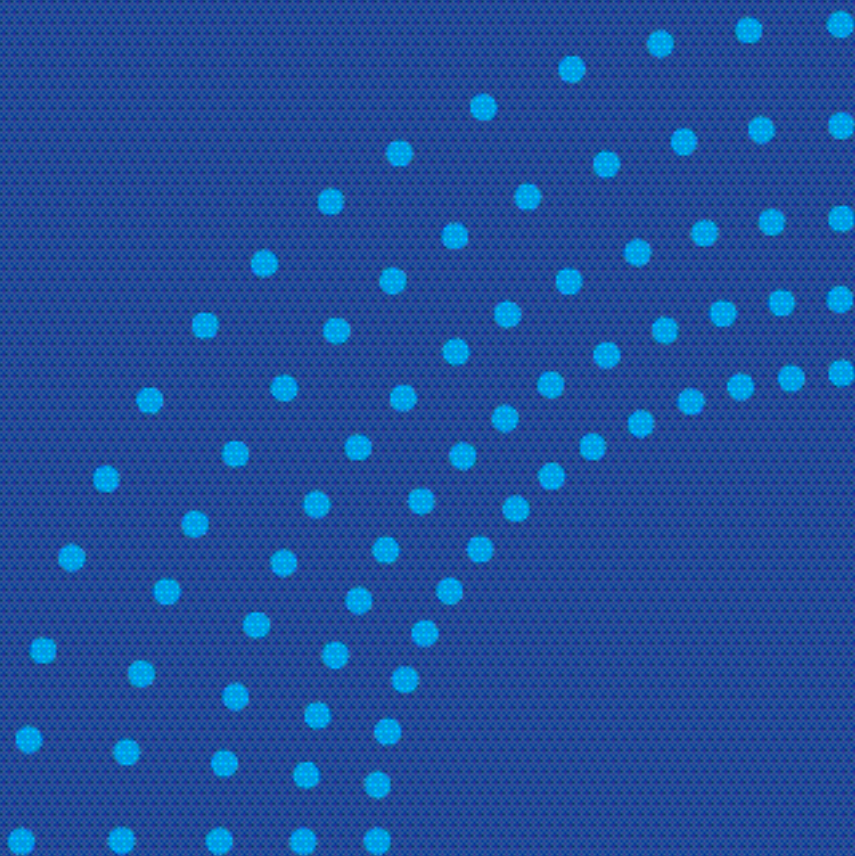


Dashboard development guide - How to build sustainable and useful dashboards to support software development and maintenance

CHALMERS |  UNIVERSITY OF GOTHENBURG

Department of Computer Science and Engineering

Mirosław Staron



Dashboard development guide

How to build sustainable and useful dashboards to support software development
and maintenance

Mirosław Staron

© Mirosław Staron, 2015

Report no 2015:02

ISSN: 1651-4870

Department of Computer Science and Engineering

Chalmers University of Technology and University of Gothenburg

Chalmers University of Technology

Department of Computer Science and Engineering

SE-412 96 Göteborg

Sweden

Telephone + 46 (0)31-772 1000

Göteborg, Sweden 2015

Dashboard development guide

How to build sustainable and useful dashboards to support software development and maintenance

Mirosław Staron

CHALMERS |  **UNIVERSITY OF GOTHENBURG**

Department of Computer Science and Engineering
CHALMERS | University of Gothenburg

Gothenburg, Sweden 2015

Dashboard development guide

How to build sustainable and useful dashboards to support software development and maintenance

Mirosław Staron, miroslaw.staron@gu.se

1 Introduction

Visualizing of organizational performance is a basis for the monitoring, controlling and improvement of the operations of organizations. Dashboards are often used for this purpose as they are a powerful tool to comprise relevant information in a single view providing graphical overview of the current status (Staron 2012). A *dashboard* is defined as an easy to read real-time user interface, showing graphical presentation of the current status (snapshot) and historical trends of an organizations Key Performance Indicators to enable decisions.

Dashboards can be used for multiple purposes and their design, technology and scope differ based on these usage scenarios:

1. Information radiators – dashboards designed to spread the information about the status to large audiences, often designed as information screen placed in central places for projects, teams, or groups.
2. Management dashboards – dashboards designed to provide information to the managers on the status of the project and the underlying parameters of the status, often designed as desktop reports with the possibility to drill-down in the data.
3. Business intelligence dashboards – dashboards designed to support product managers in accessing, visualizing and analyzing the data related to product development and its surrounding market, often designed as a desktop application with a potential for web-based access to reports.
4. Hybrid dashboards – dashboards combining two or three of the above usage scenarios.

In this document we describe how to develop and deploy a dashboard for visualizing software metrics. The document is intended for architects and designers of the dashboard and includes for following elements:

- Architecture of the dashboard
- Methods for selecting the right dashboard
- Overview of the techniques and tools for dashboard development
- Roles and responsibilities related to the dashboard development

The document is structured as follows. In section 2 we describe a reference development process for the dashboards based on the dashboard selection model (designed in Sprint 8) and the lean start-up principles of minimum-viable-product. In section 3 we present the details of how to select the right dashboard for the purpose of the organizations. In section 4 we describe a typical architecture of a dashboard and discuss its variants based on the usage scenarios. In section 5 we describe what a typical content of a software engineering dashboard is and in section 6 which roles are involved in the design of a dashboard and the responsibility of these roles. Section 7 concludes this document.

2 Dashboard development process

Dashboard should be developed iteratively in close collaboration with the users of the dashboards or the personas representing the users. However, the stages of the development process should progress from requirements elicitation where the dashboards are constructed to understand the information needs and their presentation to the maintenance of the dashboards where the corrective maintenance activities and support take place. The overview of the stages is presented in Figure 1.



FIGURE 1. DASHBOARD DEVELOPMENT PROCESS OVERVIEW

The stages can be briefly described as follows:

- **RQ Elicitation:** the goal of this stage is to collect high level expectations for the dashboard and create the first mock-ups of its content. The dashboard designers need to make interviews in the organization to identify the stakeholders, information providers and users of the dashboard. During this stage the dashboard designers need to work with the goals for the dashboard (e.g. by finding what the information needs are to be satisfied, which metrics to visualize, etc. (Staron, Meding et al. 2011)). The result is an information model for the indicators of the dashboard and the mock-up of its visual content.
- **Dashboard type selection** (see also section 3): the goal of this stage is to find the technology which is to be used to realize the dashboard. The result of this stage is a first prototype of the working dashboard as a feasibility study of the technology.
- **Dashboard design:** depending on the chosen technology the dashboard designers need to iteratively design and evaluate the dashboard. We recommend the concept of the Minimum Viable Product and the Build-Measure-Learn for this stage (Ries 2011). This stage should conclude with a working dashboard placed according to the initial requirements.
- **Impact evaluation:** after the dashboard has been put in place the dashboard designers need to observe what the impact the dashboard had on the organization. For this we recommend the theory of organizational learning by Goodman and Dean (Kontogiannis 1997). A successful dashboard, in this context, would show signs of influencing the practice at the company, which would show in the dashboard's indicators/metrics after the influenced change was introduced.
- **Dashboard maintenance:** the final stage is to place the dashboard in a maintenance where the dashboard designer or a dedicated person monitors that the dashboard is operational and that it shows the information required. The designer also needs to be involved in the updates of the dashboard once the company's goals change or the data sources change over time.

Designing and maintaining of the dashboards depend on the chosen technology, therefore the designers of the dashboard need to evaluate the needs of the organization and choose the technology wisely. In the next section we describe a technique for selecting the right dashboard.

3 Selecting the right dashboard

To select the right dashboard we can use the dashboard selection model described in the following paper - (Staron, Niesel et al. 2015) which is based on similar principles as (Mellegard, Staron et al. 2012). The dashboard selection model consists of seven categories describing seven aspects of dashboards.

1. Type of dashboard - defining what kind of visualization is needed. Many dashboards are used as reports where the stakeholders input the data and require the flexibility of the format -- the alternative is named *report* whereas some require a strictly pre-defined visualization with the same structure for every update -- the alternative designated as *dashboard*. There is naturally a number of possibilities of combining the flexibility and the strict format, which is denoted by the scale between fully flexible and fully strict.
2. Data acquisition - defining how the data is input into the tool. In general the stakeholders/employees can enter the data into the tool -- e.g. making an assessment -- the alternative is named *manual* or they can have the data being imported from other systems -- this alternative is named *automated*. The previous selection of a dashboard for visualization quite often correlates to the selection of the automated data provisioning.
3. Stakeholders - defining the type of the stakeholder for the dashboard. The dashboards which are used as so-called information radiators often have an entire *group* as a stakeholder, for example a project team. However, many dashboards which are designed to support decisions often have an *individual* stakeholder who can represent a group.
4. Delivery - defining how the data is provided to the stakeholders. On the one hand the information can be delivered to a stakeholder in such forms as e-mails or MS Sidebar gadgets -- the alternative is *delivered* or on the other hand it can be fetched, which requires the stakeholder to actively seek the information in form of opening a dedicated link and searching for the information -- which is denoted as *fetched*.
5. Update - defining how often the data is updated. One alternative is to update the data *periodically*, for example every night with the advantage of the data being synchronized but with the disadvantage that it is not up-to-date. The other alternative is the *continuous* update which has the opposite effects on the timeliness and synchronization.
6. Aim -- defining what kind of aim the dashboard should fulfill. One of the alternatives is to use the dashboard as an information radiator -- to spread the *information* to a broad audience. The other option is to design the dashboard for a specific type of *decision* in mind, for example release readiness.
7. Data flow - defining how much processing of the data is done in the dashboard. One of the alternatives is to visualize the *raw data* which means that no additional interpretation is done and the other is to add the interpretations by applying analysis models and thus to visualize *indicators*.

Graphically the dashboard selection model can be presented as a set of “sliders” which allow to prioritize between polar in these dimensions – as presented in Figure 2.

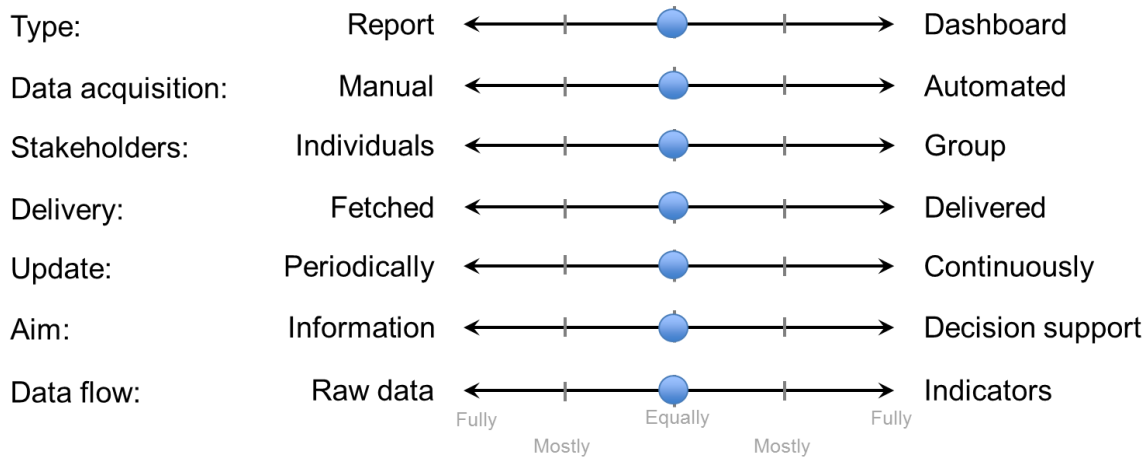


FIGURE 2. VISUAL REPRESENTATION OF DASHBOARD SELECTION MODEL

In the published paper we provide more details on what kind of combination of slider position correspond to which type of a dashboard.

However, regardless of the position of the slider or the type of the dashboard, each dashboard has the same architecture which is based on the “layered” architecture style.

4 Dashboard architecture

The layered architectural style is the most common one for dashboards as it allows to process the information as a “flow” without the need to provide star-like connections between all components of the dashboard. Depending on the type of the dashboard these component have different characteristics (e.g. wrt interactivity).

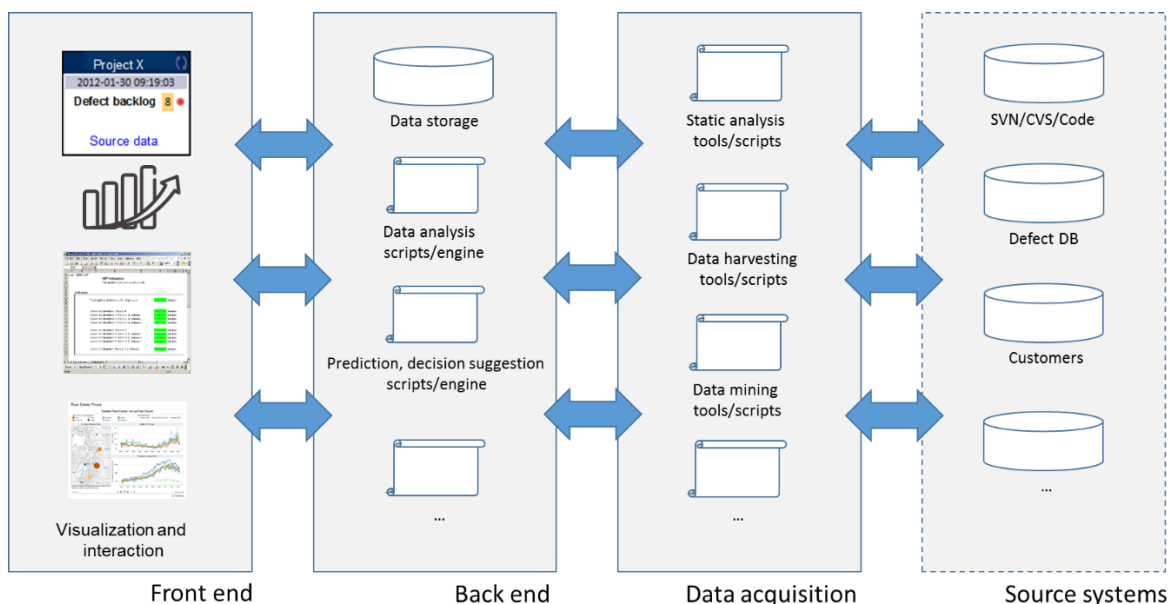


FIGURE 3. TYPICAL ARCHITECTURE OF A DASHBOARD

The front end is naturally the part of the dashboard which is the most visible one, but far from being the most important one. Depending on the type of the dashboard the set-up of the front-end can differ significantly. For the reporting dashboards the front end needs to be

interactive and support easy-to-use data input (e.g. reporting of time) whereas the visualization part is of less importance. For the information radiator dashboard the type the visualization and graphical layout are the most important elements whereas the data input is almost not required at all.

The back end layer consists of all the components which support the visualization – data sources, files storing the metrics/indicators, scripts making predictions and similar components. These components are necessary to store the data acquired from source systems, allow to analyze the data and prepare for its visualization.

The data acquisition layer is a set of scripts and programs used to collect the data from source systems. It could be metrics tools, static analysis tools, scripts for mining data repositories and similar components. The responsibility of this layer is to harvest the data from the source systems (e.g. a source code repository) and place that data in form of metric values in the storage of the back end of the dashboard.

Finally the components which are “outside” of the dashboard, but are crucial for a dashboard to function (hence delineated using the dashed line) are the source systems. These systems are part of the normal operations of the company from which data can be acquired. Examples of such systems are source code repositories, defect databases, or integration engines (e.g. Jenkins).

5 Monitoring information quality

The architecture presented in the dashboard is based on the pipes-and-filters architecture with the data flow. Therefore it is important to monitor that the calculations are correct.

For the we recommend to implement the information quality indicators based on the previous research from the software center (Staron and Meding 2009) and (Staron and Wohlin 2006).

6 Dashboard content

A typical dashboard contains three elements:

- Heading explaining the content of the dashboard and its purpose
- Diagram visualizing the metrics
- Short explanation of the status and information in the diagram

In designing the pages of the dashboard the principles of cognitive perception abilities should be taken into account, such as:

1. Elements of the dashboard should be logically and conceptually related to each other
2. The number of elements in the dashboard (diagrams, text fields, explanations, buttons) should be no more than 7 (+2 if necessary) as this is the number of elements an average person can keep in the short term memory.
3. The use of colors should be limited to the minimum and the colors should extrapolate the diagrams and the important information in the dashboard.

An example of a dashboard is presented in Figure 4, which presents a set of metrics for architecture of a software product. These metrics are logically connected and shows the

changes in the architecture's components, complexity of the architecture and changes to the interfaces of the architecture. The dashboard is build using the Google chart framework.

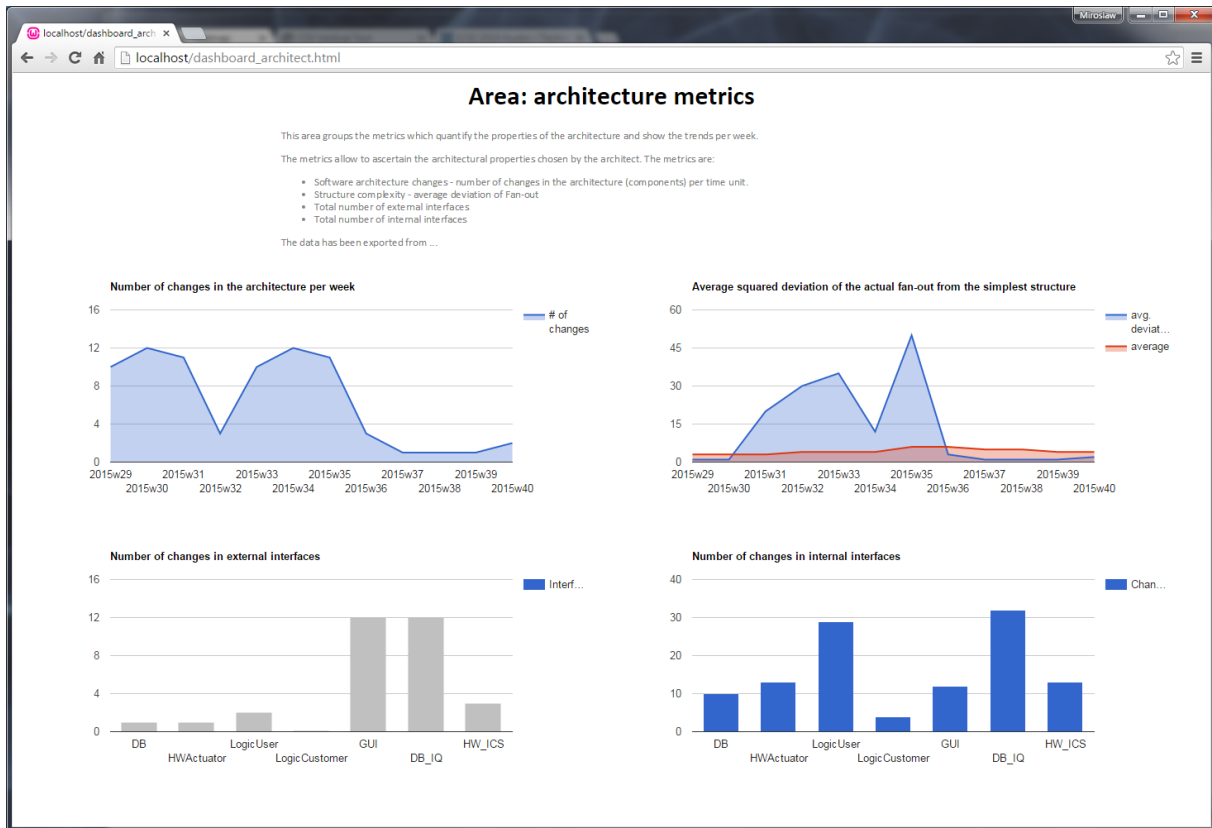


FIGURE 4. EXAMPLE OF A DASHBOARD - INTERACTIVE DASHBOARD FOR ARCHITECTURE METRICS

Another example of a dashboard (Figure 5) is the dashboard for the architectural dependencies visualizing implicit relationships in the architecture based on the previous studies in the software center (Staron, Meding et al. 2013) and outside (Mellegard, Staron et al. 2012). The dashboard contains only one diagram and shows how strongly different architectural components (A-R) are connected to each other.

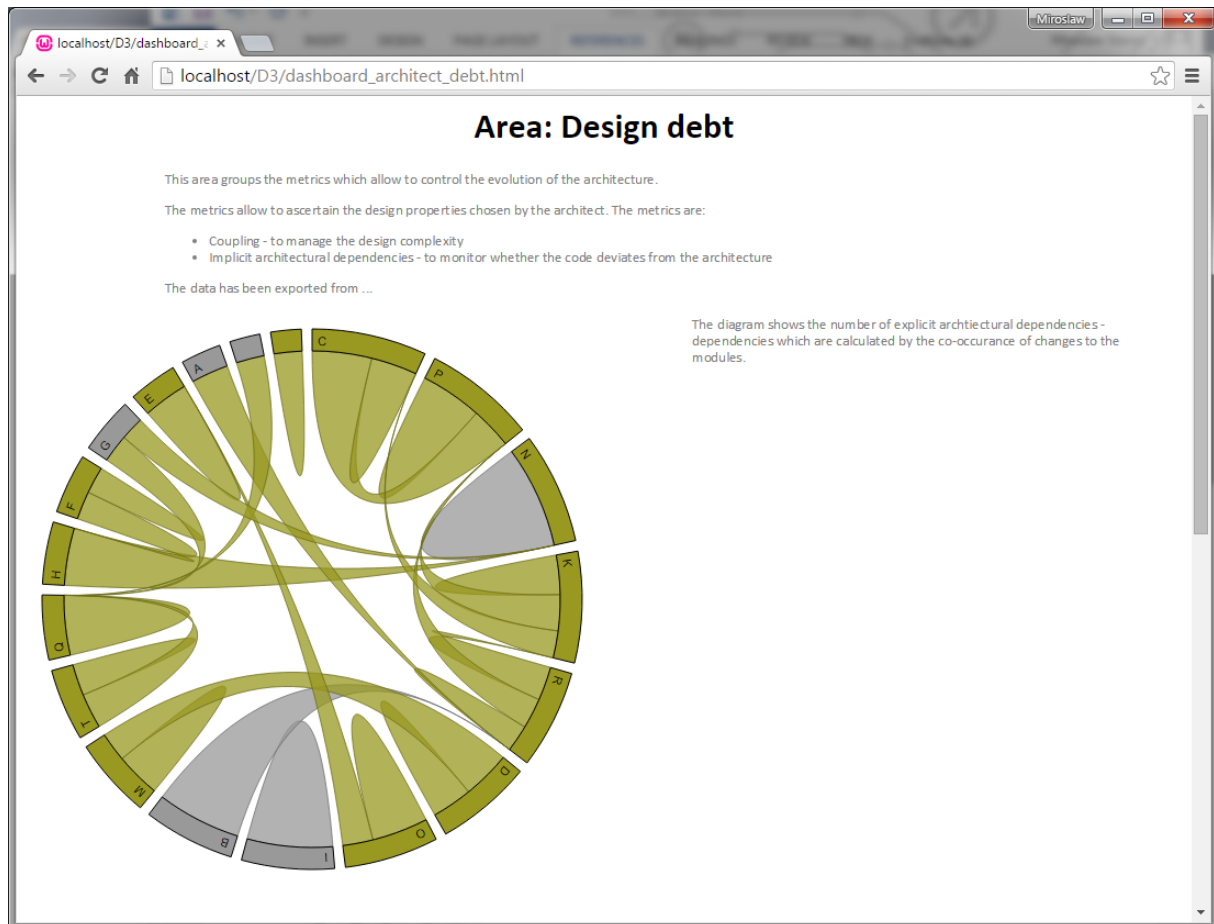


FIGURE 5. ARCHITECTURAL DEPENDENCIES DASHBOARD

The presented dashboards illustrate the principles of using graphs to communicate the information and show the simplicity required to prepare a dashboard which should be an information radiator.

The set of metrics which we collected as part of the literature studies, with the links to the corresponding papers, is presented in Appendix A.

7 Technologies

The choice of technology depends primarily on the use of the dashboard and the resources available. Below we presents a subset of technologies with a short description of their advantages and disadvantages.

A number of technologies and framework exists which can support the development of a dashboard, for example:

- Dashing.io (open source): <http://dashing.io/> - a ready-to-use dashboard software based on XML file links to the web server. The framework is simple to set up, but limited in its graphical abilities. It also requires a backbone processor of data as it cannot process the data itself.
- The dash (free): <https://www.thedash.com/> - an alternative to dashing.io, with similar requirements on backbone processor scripts, but more flexible in terms of available visualizations (e.g. diagrams).

- Google dashboard (free): https://developers.google.com/apps-script/articles/charts_dashboard - a set of simple-to-set-up javascript and SVG based charts which can be customized very easily. The main advantage is that it is simple and easy to use but it also requires backbone processing of the data.
- D3 (Data Driven Documents, open source): <http://d3js.org/> - a more flexible (powerful and expressive) alternative to Google charts/dashboard.
- Tibco Spotfire: http://spotfire.tibco.com/products/spotfire-desktop?gclid=CjwKEAjwkK6wBRCcoK_tIoT-zFASJAC7RArifNQV5JgnHYXKOVyhwDlfgKdTj0b3ei4xyJBgn6VqhoCLO3w_wcB – a business intelligence tool which allows to easily create drill-down reports and dashboards. The main advantage is that once the data is in a database the tool has a graphical way of creating the charts (no programming needed as in the previous techniques); the main disadvantage is that it is commercial and that setting up the database and importing the data requires programming and more effort than in the case of the scripts for the previous techniques.
- Tableau: <http://www.tableau.com/> and <http://www.tableau.com/learn/whitepapers/5-best-practices-for-effective-dashboards> - an alternative to Spotfire.
- Qlikview: <http://www.qlik.com> – another alternative to Spotfire

8 Roles and responsibilities

The roles and responsibilities in the dashboard design reflect the roles in the international standard ISO/IEC 15939 - Software and Systems Engineering – Measurement processes (IEEE 2007) and the process of development of measurement systems (Staron and Meding 2009, Staron, Meding et al. 2009, Staron, Meding et al. 2011) and have been shown to be important for the robust design of the entire measurement program (Staron and Wohlin 2006, Staron and Meding 2015). Table 1 presents the roles and responsibilities.

TABLE 1. ROLES AND RESPONSIBILITIES IN DASHBOARD DEVELOPMENT

Role	Responsibility
Stakeholder	Product owner of the dashboard; acts as a customer for the dashboard providing: <ul style="list-style-type: none"> • Information needs • Evaluation of the dashboard
Metric designer	Designer and developer of the dashboard; responsible for the technical part of the development and maintenance of the dashboard. In particular: <ul style="list-style-type: none"> • Develop the dashboard • Develop the visualization and update mechanisms • Monitor the daily operation of the dashboard
Measurement sponsor	Sponsor paying for the development and maintenance of the dashboard.
Measurement analyst	A specialist in the metrics area designing the metrics to be included in the dashboard; the responsibilities include: <ul style="list-style-type: none"> • Designing of the metrics according to the international standards ISO/IEC 15939, ISO/IEC 25xxx and metrology (e.g. fulfilling the properties of well-constructed measures)

	<ul style="list-style-type: none"> • Assessment of the validity of the metrics proposed by the metric champions • Maintaining the validity of the metrics over time
Metric champion	<p>A specialist in the product/process/management area proposing new metrics/changes to the existing metrics based on the information needs of the organization, in particular:</p> <ul style="list-style-type: none"> • Articulate the information need for a particular area or metric • Propose new base and derived measures, indicators • Propose the measurement method and measurement function • Support the metric designer and measurement analyst in defining the right metric and its visualization • Develop the value proposition of the metrics (Staron and Meding 2015)
Measurement librarian	<p>A dedicated person for cataloguing the dashboards, metrics and related good/bad practices, in particular:</p> <ul style="list-style-type: none"> • Collecting the lessons' learned from the usage of each dashboard and metric • Evaluate the value of the metrics • Maintain the measurement experience base as specified in ISO/IEC 15939
Measurement program leader	<p>Coordinating the measurement team and the measurement program; assuring that all relevant information needs are prioritized and satisfied</p>

The roles presented in the table can be either full-time or part-time roles depending on the size of the organization and its measurement program. It is important, however, that the number of individuals is at least two – playing the roles of stakeholders and metric champions on the one side and the designers and measurement analysts at the other side.

9 Summary and wrap-up

Using dashboard for visualizing the organizational performance has gained a considerable attention in recent years. Together with the coining of the concept of *information radiators* for Agile software development teams the number of frameworks supporting this kind of information dissemination has increased exponentially.

In this document we presented the main guidelines on how to develop a dashboard for an organization. We have presented the process of selecting a dashboard, a tool for choosing the type of the dashboard, principles of building a dashboard and a set of roles involved in the development of a dashboard.

Further reading

In this document we focused on dashboards for software development support. However, there exists a number of tutorials on how to construct a dashboard without the specific focus on software engineering, for example:

- Visualization aspects in software engineering (focused on graphics): Telea, A. C. (2014). Data visualization: principles and practice. CRC Press (Telea 2014).
- Visualization of code repositories (Voinea, Telea et al. 2005, Telea and Auber 2008)
- Visualization of areas of interest in software architecture (Byelas and Telea 2006)
- Designing and building great dashboards: https://www.geckoboard.com/blog/building-great-dashboards-6-golden-rules-to-successful-dashboard-design/#.VgwU5_mqqko
- Digital dashboards: Strategic and tactical: <http://www.kaushik.net/avinash/digital-dashboards-strategic-tactical-best-practices-tips-examples/>
- Building dashboards that people love to use: [http://www.cpoc.org/assets/Data/guide to dashboard design1.pdf](http://www.cpoc.org/assets/Data/guide%20to%20dashboard%20design1.pdf)
- Examples of 24 web dashboards: <https://econsultancy.com/blog/62844-24-beautifully-designed-web-dashboards-that-data-geeks-will-love/>
- How to build an effective dashboard: <http://www.isixsigma.com/methodology/metrics/build-a-visual-dashboard-in-10-steps/>
- Dashboard gallery: <http://www.liquidplanner.com/support/articles/dashboard-gallery/>

Appendix A – Metrics portfolio

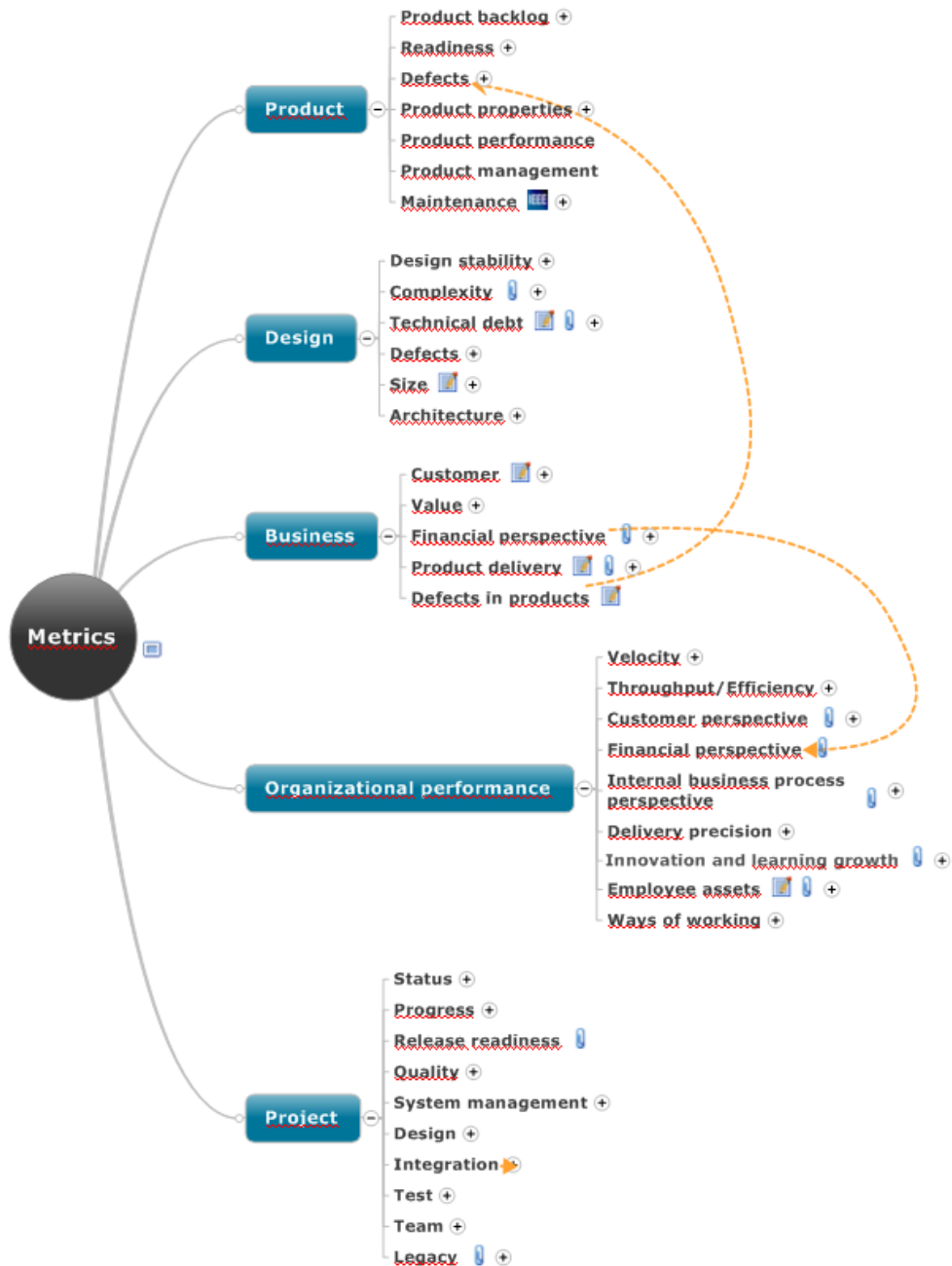


FIGURE 6. MINDMAP WITH THE METRICS COLLECTED IN THE PORTFOLIO

Product

Product backlog

- Product backlog, http://link.springer.com/chapter/10.1007/978-3-642-21843-9_5
- Code coverage, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Readiness

- Readiness/Running tested features (RTF), http://agile2009.agilealliance.org/files/session_pdfs/Rawsthorne_AgileMetrics_v6d.pdf
- Number of passed acceptance tests, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Defects

- Defect backlog, <Http://www.sciencedirect.com/science/article/pii/S0950584910000832>, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Defects carried over to next iteration, Http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1667571&tag=1
- Number of External Trouble Reports (TR) http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- # of system failures (ISO/IEC 25021)
- # of failures, QME #7 (ISO/IEC 25021)
- # of faults (ISO/IEC 25021)
- # of errors (ISO/IEC 25021)
- # of fatal errors (ISO/IEC 25021)
- Number of problem reports
- Critical problem reports per normalization unit per year
- Major problem reports per normalization unit per year
- Minor problem reports per normalization unit per year
- Problem reports per normalization unit per year
- Problem report fix response time formulas
- Major problem report fix response time
- Minor problem report fix response time
- Problem report fix response time
- Overdue problem report fix responsiveness formulas
- Major overdue problem report fix responsiveness
- Minor overdue problem report fix responsiveness
- Overdue problem report fix responsiveness
- On-time delivery formulas
- On time items delivery
- On time service delivery
- Service impact outage formulas
- Service impact all causes outage frequency per NU per year
- Service impact all causes outage downtime per NU per year
- Service impact product attributable outage frequency per NU per year
- Service impact product attributable outage downtime per NU per year

- Network impact outage
- Network element impact outage frequency - Customer attributable
- Network element impact outage (weighted) downtime - Customer attributable
- Network element impact outage frequency - Product attributable
- Network element impact outage (weighted) downtime - Product attributable
- Engineering or installation caused outage formulas
- Engineering caused outage frequency
- Installation caused outage frequency
- Field replaceable unit returns formulas
- Early return index
- Long-term return rate
- Normalized one-year return rate
- Corrective fix quality
- Software fix quality
- Software problem reports formulas
- Critical software problem reports per normalization unit per year
- Major software problem reports per normalization unit per year
- Minor software problem reports per normalization unit per year
- Service quality formulas
- Defective service transactions

Product properties

- Total product size, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1609800>
- Maturity/Software reliability growth, [Http://web.student.chalmers.se/~rakeshr/files/SRGM_embedded_journal.pdf](http://web.student.chalmers.se/~rakeshr/files/SRGM_embedded_journal.pdf)
- Branding
- Product global awareness

Maintenance, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=131381

Change

- Change count per X (e.g. category like fix, enhance, restructure), <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per status, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per maintenance type, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per change effort, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per defect source, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per quality focus, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per change span, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per detection, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per change span, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - per developer span, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
- Average number of change size, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
- Change request backlog, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Number of changes per type, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>

- Proportion of defect type (maintenance/development), <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
- % of content changes per delivery, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- KLOC change to the code, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Current change backlog, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Code change metric (CM), custom, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Change interval, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>

Time

- Time trend in change count, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
- Total test time, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Time to close urgent software change requests (SCF), <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- % of duplicate and invalid change requests closed by month, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- % of on-time deliveries, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Proportion of time trend, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - Delayed, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - Solved/unsolved, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - Rejected/non-rejected, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - Change interval used to close urgent requests, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
- Time trend in change count per maintenance type, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>

Effort

- Staff days expended/change type, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Cost/delivery, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Proportion of Change effort (development/maintenance), <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
- Cost/activity, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Change effort per, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - activity, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - maintenance type, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - change size, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - change count, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
 - origin, <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>

Product

- Software Reliability, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
 - Total failures, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
 - MTTF, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
 - Remaining failures, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
 - Time to next failure, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Computer resource utilization, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>

- Proportion of defect source (maintenance/development), <http://onlinelibrary.wiley.com/doi/10.1002/smr.412/epdf>
- # of interruptions (ISO/IEC 25021)
- Complexity, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>
- Software maintainability, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565000>

Design

Design stability

- Product/code stability, http://www.cse.chalmers.se/~mirosław/papers/2013_mensura_heatmaps.pdf
- System design instability, <http://www.sciencedirect.com/science/article/pii/S016412120400007X>
- Code churn, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- % Of ontime delivery of development projects, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>

Complexity, Http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6747165

- Model based, <Http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4351352>
 - Average number of transitions per capsule
 - Average number of choice points per capsule
 - Average number of capsule operations per capsule
 - Ratio choice points per states
 - Average visual cyclomatic complexity per capsule
 - Average number of defers/recalls per capsule
 - Average capsule size
- Cohesion, <http://agile.csc.ncsu.edu/SEMaterials/OOMetrics.htm>
- Coupling, <http://agile.csc.ncsu.edu/SEMaterials/OOMetrics.htm>
- # of error messages (ISO/IEC 25021)
- # of steps of procedure (ISO/IEC 25021)
- Task complexity (ISO/IEC 25021)
- # of operations (ISO/IEC 25021)
- Average nesting depth of #ifdefs (variability), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6062078

Technical debt, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5457755

- Initial quality, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Design debt, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Architectural dependencies, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6619529>

Defects

- Pre-release defect density (test defect/KLOEC), <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- Total defect density (pre-release + post-release defects/KLOEC), <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- Post-release defect density (released defects/KLOEC), <http://www.sciencedirect.com/science/article/pii/S1383762106000671>

- Total defect density (pre-release + post-release defects/function points), <http://www.sciencedirect.com/science/article/pii/S1383762106000671>

Size

- LOC, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Functional size of the product (ISO/IEC 25021)
- # of data items (ISO/IEC 25021)
- # of messages (ISO/IEC 25021)
- # of use cases (ISO/IEC 25021)
- Variability
 - # of feature constants, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6062078
 - Lines of feature code, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6062078
 - Scattering degree, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6062078
 - Tangling degree, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6062078
- # of the databases (ISO/IEC 25021)
- # of memory (ISO/IEC 25021)
- Model Size, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4351352>
- Class size, <http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=1392720>

Architecture

- Degree of impact of change, http://resources.sei.cmu.edu/asset_files/Presentation/2014_017_001_88189.pdf
- Number of defects injected to component, http://resources.sei.cmu.edu/asset_files/Presentation/2014_017_001_88189.pdf
- Number of components, <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=88199>, [download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32](http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=88199;download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32)
- Number of connectors, <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=88199>
- Number of symbols, <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=88199>
- Software architecture changes, <https://www.cs.umd.edu/~basili/publications/proceedings/P114.pdf>
- Coupling, <https://www.cs.umd.edu/~basili/publications/proceedings/P114.pdf>, http://www.nasa.gov/centers/ivv/ppt/172467main_Hany_Ammar_Architectural_Level_SW_Metrics.ppt
- Cohesion, http://www.nasa.gov/centers/ivv/ppt/172467main_Hany_Ammar_Architectural_Level_SW_Metrics.ppt
- Error propagation, http://www.nasa.gov/centers/ivv/ppt/172467main_Hany_Ammar_Architectural_Level_SW_Metrics.ppt
- Class dynamicity, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1605177>
- Number of classes per use case, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1605177>
- Number of use cases per class, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1605177>
- Architecture weight, <http://dl.acm.org/citation.cfm?id=512059>
- Architecture preservation factor, <http://dl.acm.org/citation.cfm?id=512059>
- Number of processing units, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>

- Number of active data repositories,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of passive data repositories,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of persistent components,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Ratio of persistent components/total number of units,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Ratio of persistent components/total number of units,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Ratio (computational + process) / total number of units,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of control links,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of data links,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of synchronization links,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of asynchronization links,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of types of communication mechanisms,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of fan-out of process units,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of fan-in of process units,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of fan-out of computational units,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of fan-in of computational units,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Max ratio function/component,
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>

- Min ratio function/component, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Average number function/component, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of logical groupings (cluster/domain), <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of architectural styles, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Number of violations of architectural styles, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Structure complexity, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf> , http://ac.els-cdn.com/0164121288900210/1-s2.0-0164121288900210-main.pdf?_tid=46d7c348-7c9f-11e5-8f9a-00000aab0f27&acdnat=1445946103_b7f396bceafd4d95cfb7907d1c48e25b
- System strength, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5DCA002B15FEB6B1CF60136C5D328E32?doi=10.1.1.71.6972&rep=rep1&type=pdf>
- Architecture adaptability index, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.2333&rep=rep1&type=pdf>
- Software adaptability index, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.2333&rep=rep1&type=pdf>
- Number of 3rd party components, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Total number of external interfaces, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Total number of internal interfaces, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Total number of specialized components, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of functionality critical components, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of architectural revisions, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of interface types, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of versions, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of generic components, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of redundant components, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of subsystems, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of services, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>
- Number of concurrent components, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=685597>

Business

Customer

- ISP/Availability
- Market share

- Survey of customer satisfaction
- Return rate
- Percent of sales from new products
- Percent of sales from proprietary products
- On-time delivery defined by customer
- Share of key accounts' purchases
- Ranking by key accounts
- Number of cooperative engineering efforts
- Customer complaints
- Complaints resolved on first contact
- Response time per customer request
- Direct price
- Price relative to competition
- Total cost to customer
- Average duration of customer relationship
- Customers lost
- Customer retention
- Customer acquisition rates
- Percent of revenue from new customers
- Number of customers
- Annual sales per customer
- Win rate
- Customer visits to company
- Hours spent with customers
- Marketing cost as a percentage of sales
- Number of ads placed
- Number of proposals made
- Brand recognition
- Response rate
- Number of trade shows attended
- Sales volume
- Share of target customer spending
- Sales per channel
- Average customer size
- Customers per employee
- Customer service expense per customer
- Customer profitability
- Frequency (number of sales transactions)
- % Of budget dedicated to customer analysis or verification, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- % Of Customer driven projects, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- Perceived customer satisfaction, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Net promoter score, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Number of technical documents

- Number of technical documents reviewed
- Number of technical documents reviewed and approved

Value

- Business value delivered, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1667571&tag=1
- Return of Value

Financial perspective, <http://www.sciencedirect.com/science/article/pii/S1044500510000831>

- Operating income
- Sales growth
- Return on investment
- The rate of achieving budget
- Shareholders
 - Market share
 - Net sales
 - Orders booked
- Cash flow
- Quarterly sales growth and operating income by division
- Increased market share
- Return on Capital
- Financial costs, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Financial revenues, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Profit and loss, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Product delivery, [Http://www.sciencedirect.com/science/article/pii/S0377221705000135](http://www.sciencedirect.com/science/article/pii/S0377221705000135)

- Delivery speed
- Delivery reliability
- New product introduction
- Number of technical documents delivered per customer
- New product development time
- Manufacturing lead-time
- Customer responsiveness
- Number of technical documents delivered, reviewed and approved, per customer

Defects in products

See also: [Defects](#)

Organizational performance

Velocity

- Velocity - completed SPs, http://agile2009.agilealliance.org/files/session_pdfs/Rawsthorne_AgileMetrics_v6d.pdf
- Cycle-Time per Feature, http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- Days Open, External Trouble Reports, http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- Builds per iteration

Continuous integration metrics, See also: [Integration](#)

- Code coverage, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4599529
- Static bug detection, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4599529
- Pulse, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1609800> ,
http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- Defect turnaround time
- Build time

Throughput/Efficiency

- Functionality / Work Effort, http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- Productivity, <http://www.sciencedirect.com/science/article/pii/S0950584910002156>
- Flow, http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- Waste
- Deliveries per month
- Product burndown, http://agile2009.agilealliance.org/files/session_pdfs/Rawsthorne_AgileMetrics_v6d.pdf
- Impediments
- Value efficiency, <http://onlinelibrary.wiley.com/doi/10.1002/spe.975/full>
- Inventory of phase, <http://onlinelibrary.wiley.com/doi/10.1002/spe.975/full>
- # of function test cases developed per week, http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- # of function test cases planned but not developed (queue),
http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- # of features integrated per week, http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- # of features planned in the integration plan to date but not integrated (queue),
http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- Fault closing speed, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1609800>
- # of defects reaching the state "closed" per week, http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- # of defects not in the state "closed" (queue), http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- # of system test cases planned for execution up to a given week (queue),
http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- # of acceptance test cases executed per week, http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- # of acceptance test cases planned for execution up to a given week (queue),
http://link.springer.com/chapter/10.1007%2F978-3-642-21843-9_3
- Capacity, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Time efficiency, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

- Value efficiency, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Effort (ISO/IEC 25021)
- Defect removal efficiency (test defects/total defects),
<http://www.sciencedirect.com/science/article/pii/S1383762106000671>

Customer perspective, <http://www.sciencedirect.com/science/article/pii/S1044500510000831>

- Customer response time
- On time delivery
- Manufacturing/service lead time
- Customer Service Request (CSR) Turnaround Time, http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- Business Value / Work Effort, http://link.springer.com/chapter/10.1007%2F978-3-642-38314-4_12
- Accuracy of interpretation of customer requirements,
<http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>

Internal business process perspective,

<http://www.sciencedirect.com/science/article/pii/S1044500510000831>

- Number of customer complaints
- Percent of shipments returned due to poor quality
- Number of warranty repair requested by customers
- Ratio of defective output/total output
- Manufacturing geometry vs. competition
- Cycle time
- Breakeven time
- Inventory turnover
- Average lead-time
- Community involvement
- Patents pending
- Cycle time improvement
- Unit cost
- Silicon efficiency
- Engineering efficiency
- Actual introduction schedule vs. plan
- Lead user identification
- Waste reduction
- Number of positive media stories
- Leadtime for development, http://link.springer.com/chapter/10.1007%2F978-3-642-30350-0_8

Delivery precision

- % of projects respective cost and budget, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- % of respected milestones, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- Average project delay, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- Delivery of product to cost (as quoted), <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- Actual vs target time of project completion, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>

- Cost of delay, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Innovation and learning growth,

<http://www.sciencedirect.com/science/article/pii/S1044500510000831>

- Number of new service/product launch
- Time to market of new products/services
- On job training hours
- Employees' suggestions
- Time to develop next generation
- Process time to maturity
- Percent of products that equals 80% of sales
- New product introduction vs. competition
- Innovativeness rating, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- Costs of investments, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Employee assets, <http://amr.aom.org/content/26/3/446.short>

- Number of employees
- Number of employees per site
- Number of consultants
- Value added per employee
- Employee participation in professional and trade organizations
- Motivation index
- Number of consultants per site
- Employee satisfaction, <http://www.sciencedirect.com/science/article/pii/S1044500510000831>
- Average R&D personnel turnover, <http://onlinelibrary.wiley.com/doi/10.1111/radm.12074/pdf>
- Perceived skill level, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Ways of working

Agile, Pair programming, http://link.springer.com/chapter/10.1007/978-3-642-20677-1_15

- Score
- Change propagation
- Defect density - proportion of bad methods
- Defect density
- Passed test cases

Project

Status

- Inventory, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- # items needing rework, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Overall state, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Progress

Monitoring release

- Release readiness, http://link.springer.com/chapter/10.1007%2F978-3-642-30350-0_7
- Release Burnup, Http://agile2009.agilealliance.org/files/session_pdfs/Rawsthorne_AgileMetrics_v6d.pdf
- Earned Value Metrics (CPI and SPI),
Http://agile2009.agilealliance.org/files/session_pdfs/Rawsthorne_AgileMetrics_v6d.pdf
- Earned Business Value,
Http://agile2009.agilealliance.org/files/session_pdfs/Rawsthorne_AgileMetrics_v6d.pdf
- Progress within a sprint,
Http://agile2009.agilealliance.org/files/session_pdfs/Rawsthorne_AgileMetrics_v6d.pdf ,
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1609800>
 - Sprint Task Hour Burndown
 - Checklist Item Burnup
 - Story Point Burnup
 - Graduated Story Point Burnup
- Feature burndown chart, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1667574 Duration in time units, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Schedule slippage, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Value transition, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Value added time, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Non-value added time, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- # of work items, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Failure load, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Rework rate, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3
- Duration (ISO/IEC 25021)
- Release readiness, http://link.springer.com/chapter/10.1007%2F978-3-642-30350-0_7

Quality

Defects

- Faults per iteration, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1609800>
- Fault slip-through, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- requirements review slippage, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- unit test slippage, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- function test slippage, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- system test slippage, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- integration test slippage, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- acceptance test slippage, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # of accessible functions (ISO/IEC 25021)
- # of user problems (ISO/IEC 25021)
- # of records (ISO/IEC 25021)

System management

- Requirements stability
- Change requests

Design

- # high level requirements, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # detailed requirements, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # requirements in design and implementation, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # in test, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # change requests under review, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # approved CRs, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # CRs ready for impact analysis, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # CRs in test, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # story points, http://link.springer.com/chapter/10.1007%2F978-3-642-44930-7_3

Integration

- # of builds
- Integration speed

Test

Test driven development, http://research.microsoft.com/en-us/groups/ese/nagappan_tdd.pdf

- Defect rate (defect density), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1251029
- Average percentage of faults detected, <http://www.sciencedirect.com/science/article/pii/S0950584911001947>
- Fault-Adequate Test set size (FATE), <http://www.sciencedirect.com/science/article/pii/S0950584911001947>
- Average Test Effort Index, <http://www.sciencedirect.com/science/article/pii/S0950584911001947>

General

- Test coverage, <https://www.computer.org/csdl/maqs/so/1990/02/s2065.pdf>
- Coverage – Microsoft, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5315981&tag=1
- Test sufficiency, <https://www.computer.org/csdl/maqs/so/1990/02/s2065.pdf>
- Test classes to New/Changed classes, <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- Unit tests per user story
- New classes with corresponding test classes, <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- Test LOC/source LOC, <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- Test first design, <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- Automated unit tests, <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- Customer acceptance tests, <http://www.sciencedirect.com/science/article/pii/S1383762106000671>
- # unit tests, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # function tests, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # integration tests, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # system tests, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>
- # acceptance tests, <http://www.sciencedirect.com/science/article/pii/S0164121210000403>

of test cases (ISO/IEC 25021)

- Functional (Fitness) tests per user story

Team

- Team size, <https://www.gartner.com/doc/1962817/balance-size-skills-agile-team>
- Team member loading
- Workload, http://link.springer.com/chapter/10.1007%2F978-3-642-30350-0_8

Legacy, Http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1667571&tag=1

- Obstacles carried over into next iteration
- User Stories carried over into next iteration
- Iteration mid point inspection
- Obstacles cleared per iteration

References

- Byelas, H. and A. Telea (2006). Visualization of areas of interest in software architecture diagrams. Proceedings of the 2006 ACM symposium on Software visualization, ACM.
- IEEE (2007). IEEE Std 15939–2007 IEEE Systems and Software Engineering—Measurement Process, IEEE–SA.
- Kontogiannis, K. (1997). Evaluation experiments on the detection of programming patterns using software metrics. Reverse Engineering, 1997. Proceedings of the Fourth Working Conference on, IEEE.
- Mellegard, N., M. Staron and F. Torner (2012). A light-weight defect classification scheme for embedded automotive software and its initial evaluation. Software Reliability Engineering (ISSRE), 2012 IEEE 23rd International Symposium on, IEEE.
- Ries, E. (2011). The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses, Random House LLC.
- Staron, M. (2012). "Critical role of measures in decision processes: Managerial and technical measures in the context of large software development organizations." Information and Software Technology(0).
- Staron, M. and W. Meding (2009). Ensuring Reliability of Information Provided by Measurement Systems. Software Process and Product Measurement, Springer Berlin / Heidelberg.
- Staron, M. and W. Meding (2009). Using Models to Develop Measurement Systems: A Method and Its Industrial Use. Software Process and Product Measurement. A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego and J. Brunekreef. Amsterdam, NL, Springer Berlin / Heidelberg. **5891**: 212-226.
- Staron, M. and W. Meding (2015). Measurement-as-a-Service—A New Way of Organizing Measurement Programs in Large Software Development Companies. Software Measurement, Springer: 144-159.
- Staron, M. and W. Meding (2015). "MeSRAM - A Method for Assessing Robustness of Measurement Programs in Large Software Development Organizations and Its Industrial Evaluation " Journal of Systems and Software **n/a**(accepted for publication).

- Staron, M., W. Meding, C. Hoglund, P. Eriksson, J. Nilsson and J. Hansson (2013). Identifying Implicit Architectural Dependencies Using Measures of Source Code Change Waves. Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on, IEEE.
- Staron, M., W. Meding, G. Karlsson and C. Nilsson (2011). "Developing measurement systems: an industrial case study." Journal of Software Maintenance and Evolution: Research and Practice **23**(2): 89-107.
- Staron, M., W. Meding and C. Nilsson (2009). "A framework for developing measurement systems and its industrial evaluation." Information and Software Technology **51**(4): 721-737.
- Staron, M., K. Niesel and W. Meding (2015). Selecting the Right Visualization of Indicators and Measures – Dashboard Selection Model. Software Measurement. A. Kobyliński, B. Czarnacka-Chrobot and J. Świerczek, Springer International Publishing. **230**: 130-143.
- Staron, M. and C. Wohlin (2006). An Industrial Case Study on the Choice between Language Customization Mechanisms. 7th International Conference, PROFES 2006, Amsterdam, The Netherlands, Springer-Verlag.
- Telea, A. and D. Auber (2008). Code flows: Visualizing structural evolution of source code. Computer Graphics Forum, Wiley Online Library.
- Telea, A. C. (2014). Data visualization: principles and practice, CRC Press.
- Voinea, L., A. Telea and J. J. Van Wijk (2005). CVSscan: visualization of code evolution. Proceedings of the 2005 ACM symposium on Software visualization, ACM.