THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Evaluation

Citation for published version:
Francois, P, Grohmann, S, Eck, K, Gavin, O, Koller, A, Nagy, H, Dirschl, C, Turchin, P & Whitehouse, H 2018, Evaluation. in K Feeney, J Davies, J Welch, S Hellmann, C Dirschl, A Koller, P Francois & A Marciniak (eds), *Engineering Agile Big-Data Systems.* River Publishers Series in Software Engineering, River Publishers, pp. 313-324.

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Engineering Agile Big-Data Systems

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 22. Jul. 2022

7

Evaluation

Pieter Francois[1], Stephanie Grohmann[1], Katja Eck[2], Odhran Gavin[3], Andreas Koller[4], Helmut Nagy[4], Christian Dirschl[2], Peter Turchin[5] and Harvey Whitehouse[1]

[1]University of Oxford, UK
[2]Wolters Kluwer Germany, Germany
[3]Trinity College Dublin, Ireland
[4]Semantic Web Company, Austria
[5]University of Connecticut, USA

7.1 Key Metrics for Evaluation

The evaluation of productivity, quality and agility requires concrete metrics to be evaluated prior to the introduction of ALIGNED tools. This gives us a baseline measurement for gains in the three evaluation areas. Once ALIGNED tools and processes are then deployed, concrete comparisons can be made to assess the progress, which results from ALIGNED tools and processes. The units over which evaluation takes place, and the measures over these units must be designed such that they can be assessed both prior to, and after, the integration of ALIGNED tools and processes.

In order to evaluate the tools that we produced during the ALIGNED project, we took the following steps:

- Baseline studies: an initial estimate of how the use cases perform before the introduction of ALIGNED tools.
- Studies on initial prototypes: focussed initial prototypes will be developed for three ALIGNED use cases in phase 1 of the project (up to month 9) that only depend on the work of a single technical workpackage (WP3, WP4, WP5) and the tools can be evaluated in this initial phase to gain rapid user insight and feedback.
- Longer-term evaluations based on the empirical evidence collected from the four use cases for ALIGNED methods and tools developed during phase 2 and phase 3 of the project.

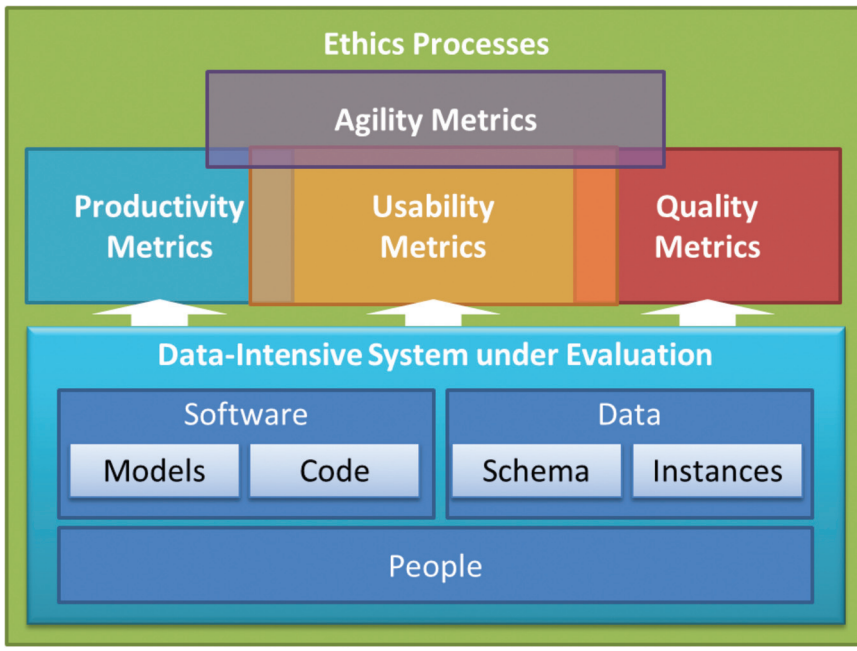

Figure 7.1 The ALIGNED Evaluation Framework.

There were three key target areas for the impact of ALIGNED methods and tools on the development and evolution of data-intensive systems: productivity, quality, and agility. Each of these is defined below to allow cross-tool and cross-use case comparisons to be made. In addition, each target area can be split into data and software aspects as well as system-wide measures, for example data management productivity, software development productivity and overall system productivity. For data management, it is often useful to split tasks into schema-oriented and dataset or instance-based measures since these often have different actors, timeframes and scopes. Figure 7.1 illustrates the ALIGNED evaluation framework, which is made up of the data-intensive system under study and the four evaluation aspects plus ethics processes covered by this handbook.

An important aspect of system evaluation that has cross-cutting impacts on quality, productivity, and agility is the well-developed concept of usability[1] and ALIGNED performed usability evaluations on all tools developed within the project.

[1]Ergonomics of Human System Interaction ISO 9241, in particular part 11 – Human-Computer Interaction, 1998.

In general, ALIGNED stressed quantitative evaluation over qualitative measures (information or data based on quantities obtained using a quantifiable measurement process) as befits automated systems such as model-driven software tools. However, the nature of systems development and maintenance (evolution) are that of a socio-technical system and as such qualitative evaluation (qualities that are descriptive, subjective or difficult to measure) based on user feedback were used to supplement quantitative evaluations. This is especially true in cases where informal or semi-automated human-based systems are either currently deployed (for baseline studies) or are necessary to produce the best outcomes (e.g., domain expert-based data curation).

7.1.1 Productivity

For evaluation purposes, we understand productivity as being a measure of the amount of human effort required to produce some unit of software, schema or dataset change for a given use case scenario. This effort may be measured in person-hours, but other measures are possible such as task completion time, task completion rate, or task error rate. For largely user-interface-driven processes, there are a number of popular keystrokes[2] or click-based models[3] for estimating productivity. For software engineering, there is prior work on evaluating the productivity of new engineering processes that should be considered.[4] In the first instance, it is possible to find a number of proxies which, when taken together, may act as a crude guide to measuring software size. Lines of Code,[5] Control-flow or Cyclomatic complexity,[6] and various feature counts[7] have traditionally acted as primitive metrics for software scale and complexity.

[2]The Keystroke Level Model for User Performance Time with Interactive Systems S. Crad, T. Moran, A. Newell, CACM, v23 n7, July 1978.

[3]Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance W. Gray, B. John, M. Atwood, Human-Computer Interaction, Vol. 8, Issue 3.

[4]Measuring and predicting software productivity: A systematic map and review K. Petersen, Information and Software Technology, Vol. 53, Issue 4, pp. 317–343.

[5]A Survey on Impact of Lines of Code On Software Complexity S. Bhatia, J. Malhotra, ACM SIGSOFT Software Engineering Notes, Vol. 39, pp. 1–6.

[6]Cyclomatic Complexity Metric for Component Based Software S. Chidamber, C. Kemerer, International Conference on Advances in Engineering and Technology Research (ICAETR), pp. 1–4, 2014.

[7]A metrics suite for object oriented design U. Tiwari, S. Kumar, IEEE Transactions on Software Engineering, Vol.20, No. 6, pp. 476–493.

There are also several cases in which cross-cutting productivity concerns are of importance, the one most particularly relevant to ALIGNED being the productivity costs of parallel development of software, schema, and datasets changes. In this case, productivity measures should look at the cost of changes from one area to the others in terms of productivity.

7.1.2 Quality

Quality is generally taken as the assessment of "fitness for purpose"[8] of the output of a given tool, process, or method. The measurement of quality is generally more context-dependent, and different measures are used in the areas of software, schema, and data.

For software quality, evaluation of software generation tools is difficult, especially as ideal tools produce no defects, and validating the absence of something is hard. It is possible to measure "churn" of software development or counts of bugs found and that can act as metrics for software quality and reliability.[9]

For data, we assess the ability of the data to satisfy properties, which are either desirable or required by consumers of the data. In particular, we will reuse the methods of assessment of Linked Data Quality defined by Zaveri et al.[10] This gives us 27 separate dimensions on which to evaluate data quality and specifies multiple metrics for all of them.

7.1.3 Agility

We define agility as the speed at which the ALIGNED tools can be adapted and reconfigured in the face of ongoing changes in requirements. It is often measured in terms of the human effort required to enact the change and so is closely related to productivity measures. When software or data management task sizes are combined with measurements of man-hours spent on development, some approximations can be made for notions of agility.[11]

---

[8]The Quality Control Handbook J. Juran, McGraw-Hill, New York, 1974.

[9]Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities S. Yonghee, A. Meneely, L. Williams, J. Osborne, IEEE Transactions on Software Engineering, Vol. 37, No. 6, pp. 772–787.

[10]Quality Assessment Methodologies for Linked Open Data A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, S. Auer, Semantic Web Journal

[11]Survey on agile metrics and their inter-relationship with other traditional development metrics S. Misra, M. Omorodion, ACM SIGSOFT Software Engineering Notes. Vol. 36, Issue 6, pp. 1–3.

Agility for our use cases will often be measured with respect to parallel co-development of software, schema and datasets as agility is a cross-cutting concern. For instance, a change to a schema or ontology will generally require both migration of datasets, as well as changes to the programme interface to consumption of the data.

7.1.4 Usability

ISO 9241[12] on human computer interaction defines usability as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use". Effectiveness and efficiency can be measured through productivity-style measurements of task outputs and work rates. However, it is also considered valuable to analyse the user error rates generated and the quality of work produced (linking to our quality measures). Satisfaction is probably the hardest aspect to accurately measure but we will deploy System Usability Scale (SUS)[13] user questionnaires as a baseline. It is simple, ten-item attitude Likert scale giving a global view of subjective assessments of usability. Despite its simplicity, SUS is well-understood and widely deployed, and this gives us access to decades of comparative usability studies and analysis to interpret SUS results.[14] In addition, it is easy to augment SUS with additional questions that are specialised to the system under study or which follow recent best practice in user surveys such as Perlmans Practical Usability Evaluation questionnaire.[15]

In addition to questionnaire-based approaches to usability, we deployed, where appropriate, the "thinking-aloud" protocol where participants are asked to verbalise their thinking while performing a task.[16] Other techniques deployed are "co-discovery", where participants are asked to verbalise their thinking while performing a task and "retrospective testing" or "coaching".[17]

---

[12]Ergonomics of Human System Interaction ISO 9241, in particular part 11 – Human-Computer Interaction, 1998.

[13]SUS: a "quick and dirty" usability scale J. Brooke, Usability Evaluation in Industry. London: Taylor and Francis, 1986.

[14]An empirical evaluation of the system usability scale A. Bangor, P.p T Kortum, and J. T. Miller, Intl. Journal of Human-Computer Interaction, Vol. 24, Issue 6, pp. 574–594, 2008.

[15]Practical usability evaluation G. Perlman, CHI'97 Extended Abstracts on Human Factors in Computing Systems. pp. 168–169, ACM, 1997.

[16]Protocol analysis: verbal reports as data, revised edition K. A. Ericsson, H.A. Simon MIT Press, Cambridge, MA, 1993.

[17]Usability Engineering 2nd edition J. Neilsen, Morgan Kaufmann, San Francisco, 1994.

7.2 ALIGNED Ethics Processes

This section provides a set of guidelines followed by the coordinators of ALIGNED pilot studies and trials. Specific instructions are provided for each step in the life cycle of these pilot studies that involves ethical considerations. Taken together, these guidelines provide ALIGNED collaborators with detail on when and how to engage with the Ethics and Society sub-committee of the ALIGNED project and on how to ensure the pilot studies and trial confirm to both relevant national and EU regulation.

Over the life cycle of a pilot study, coordinators need to engage with ten sets of action points.

- BEFORE THE START OF THE PILOT STUDY: Coordinators need to familiarise themselves thoroughly with the Ethics section of the contract signed between the ALIGNED project and the EC. This is an important first step to understand the full range of potential ethical issues at stake when setting up a pilot study.
- BEFORE THE START OF THE PILOT STUDY: Coordinators need to obtain the appropriate internal institutional ethical approval. The bodies responsible for internal institutional approval are your first port of call to ensure that the pilot study respects institutional, national and European regulation. This is especially important for any pilot study that involves the storage of personal data as some categories of these data are classed as 'sensitive' (e.g., health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction), and this data may only be processed according to specific rules. The ALIGNED Ethics and Society sub-committee has the details of the relevant institutional bodies for each partner.
- BEFORE THE START OF THE PILOT STUDY: Coordinators need to forward the institutional ethical approval obtained to the ALIGNED Ethics and Society sub-committee. This step is crucial as it is vital that the ALIGNED project forwards all ethical approvals to the EC. Furthermore, this will allow the Ethics committee to double check that all necessary steps have been taken and that the pilot study fulfils all necessary requirements.
- BEFORE THE START OF THE PILOT STUDY: As most pilot studies will involve voluntary participants, the coordinator must submit the consent form to be used to the ALIGNED Ethics and Society sub-committee. This consent form must be modelled on the template attached in appendix A and any change to the template must be approved by the ALIGNED Ethics and Society sub-committee.

- BEFORE THE START OF THE PILOT STUDY: As most pilot studies will involve voluntary participants, the coordinator needs to ensure that all staff associated with the pilot study fully understand the ethical considerations when handling voluntary participants. For this, all staff need to familiarise themselves with the relevant Ethics sections of the contract signed between ALIGNED and the EC. Special attention must be paid to those sections dealing with the recruitment of voluntary participants, the control of their personal data, the nature of their participation, the right of participants to cancel their involvement at any time in the process, the rights of voluntary participants to privacy and appropriate treatment, and the definition of informed consent. It is absolutely vital that no participation can take in any form without informed consent.
- AT THE START OF THE PILOT STUDY: the coordinator will ensure that sufficient measures are in place to store all personnel data password protected and all 'sensitive' personnel data encrypted.
- AT THE START OF THE PILOT STUDY: the coordinator, in collaboration with the ALIGNED Ethics and Society sub-committee, will prepare and share with the voluntary participants detailed information on the procedures that will be implemented for data collection, storage, protection, retention, and destruction. The ALIGNED Ethics and Society sub-committee will provide the coordinator with input to ensure that this information conforms to national and European legislation.
- THROUGHOUT THE ENTIRE LIFESPAN OF THE PILOT STUDY: the coordinator needs to assess on a continuous basis whether any of the ALIGNED methodologies result in discriminatory practices or unfair treatment. The pilot study coordinator needs to inform the ALIGNED Ethics and Society sub-committee even in case of the slightest doubt that the pilot study results in discriminatory practices or unfair treatment.
- DURING AND AFTER THE PILOT STUDY: In the case of incidental findings of value arising from research activities (e.g., psychological trauma arising from productivity-related questions), the coordinator needs to inform participants when such results will be disseminated. Participants will be given the right to withdraw their information.
- AFTER THE PILOT STUDY: as personnel data can only be archived during the lifespan of the ALIGNED project and thus needs to be deleted at the end of the project, the coordinator will work together with the ALIGNED Ethics and Society sub-committee to ensure the deletion of all personal data.

7.3 Common Evaluation Framework

Productivity, quality, and agility are the three dimensions that are most usefully measured in order to practically evaluate data-intensive systems. However, they are not separate dimensions but in fact have close semantic connections between them.

7.3.1 Productivity

Productivity is the overarching dimension used to measure the performance of all work systems – the ratio of the value provided by a service to the cost of delivering the service. If we were to implement two alternative systems in parallel and maintain them over time so that they provided exactly the same service, the relative cost would provide us with an unambiguous guide as to which system had performed better. Similarly, if we were to spend exactly the same time and money on delivering the same service over a period of time, through two alternative systems, the relative value provided by each would again tell us which system had performed better.

However, while costs are normally reasonably easy to measure, the value provided by a system can be more difficult as systems can be embedded within larger systems and provide value that cannot easily be distilled into economic units.

7.3.2 Quality

In the context of information systems, quality is a proxy measure for value. The better the Quality of Service (QoS), the greater the value provided by the system. If this is not the case, then the QoS has not been well defined. In general, therefore, if two systems provide the same QoS, we can compare them directly in terms of costs. In data-intensive systems, we are primarily focussed on the data quality because much of the behaviour of the system is driven by data. However, data quality only has meaning in the context of the services that are based on the data. We care about the overall service quality, and data quality is only interesting to the extent that it affects the business value provided by the system.

In any given system, it should be the case that improving quality increases the value provided by the system and vice versa. Quality is a multi-dimensional concept,[18] often with complex non-linear interactions

[18] A metrics suite for object oriented design U. Tiwari, S. Kumar, IEEE Transactions on Software Engineering, Vol. 20, No. 6, Pages 476–493.

between variables in different dimensions. For any given system, we can imagine a function Qual(sys) $\rightarrow$ \$n which generates the value provided by a given system. In practice, we normally really want to know Qual(sys') $\geq$ Qual(sys), the effects of a given change in a system. We need a function which, for any given change to a system, will tell us what the change to business value will be. Our quality model defines the variables that will be passed to this function, and the function's implementation defines how changes to the values of variables impact service value.

7.3.3 Agility

We would like to be able to forecast the performance of systems and not just compare them in retrospect. Agility is essentially a measure of future productivity which attempts to capture such a forecast. How much future value will this system provide and at what cost? The trouble with this measure, of course, is that we do not know what opportunities for value the future holds. For any given system, agility to make changes that we never end up wanting to make have essentially no value. This means that agility, like quality, is very domain and context-dependent. We therefore need to know which types of changes are likely to be important in a given system before we can assess its agility. Because this is a prediction about the future, it can never be more than probabilistic, but previous behaviour is normally a good guide to future behaviour, so we can normally extract at least some characteristics of the types of changes that are important in a particular domain by observing existing systems.

In data-intensive systems, scale – considered as the volume, velocity and complexity of the data – tends to have significant influence upon the system's agility and tends to increase over time. As a general rule of thumb, service value and cost both increase with scale. Therefore, one of the most important aspects of understanding a data-intensive system's agility is understanding the interaction between these two variables and the different components of scale – in the context of the likely evolution of the system over the course of its operation.

Ultimately, the value of any work-system can be characterised by its productivity curve over time. The more agile the system, the more this curve will tend to rise in the future; the less agile the system, the quicker it will fall. This is because the more agile the system, the quicker and more effectively changes can be tested to meet emerging requirements. The most important way to compare systems is the net value that they deliver over their lifetime. We cannot know this in advance, but we can normally make reasonable

predictions based on proxies for agility in any given context and use them to predict the likely future productivity trajectory.

Because the dimensions and metrics used in any given data-intensive system are heavily dependent on the specific context of the service, they cannot be directly compared. In one context, better accuracy and precision of data might be considered to have a uniformly positive effect on Quality of Service. In another context, it might cause the system to crash (e.g., because it causes the program to trigger a bug in a floating point operation that was not used when the data had lower precision).

Rather than comparing data quality directly, we can compare it indirectly through the cost of providing a given quality of service. There are several aspects that must be considered in this comparison:

Data Curation Cost: The cost of maintaining the data at a given quality level (to provide constant Quality of Service) over a period of time, given changes in scale. There are two particularly important data quality levels that are worth focussing on here. DQmin is the minimum level of data quality required in order for the service to work. The threshold is multi-dimensional and complex and includes, among other things, all the database conditions which cause the software service to crash. The second quality threshold worth considering is DQmax – the maximum level of data quality that the service can exploit. Examples of data quality that exceeds DQmax: data stored as floating points with high precision that are then cast to integers by a program, metadata about data semantics that is not used by programs. As a general rule, there is no return on investment for exceeding DQmax. Between these two thresholds, quality can vary in any number of dimensions. If the dimensions used are well chosen for the service, then increases in quality will translate into an increase in the overall quality of service provided, and if the service is well aligned with the business needs, this translates directly into increased business value.

Data Agility Cost: the cost of increasing the overall value provided by the system by using existing data in a new way – for example, how much time and money is required to make a slice of the data in a database available for use by a new program (with whatever data-formatting requirements it has). The cost includes any changes to the code of programs that consume the data, everything that is required to produce and deliver the new service.

Model Agility Cost: the cost to change the overall behaviour of the system in situations which require changes to the structure of the data. This includes the costs of changing the structure of the data, changing the software to encode the new behaviour, and returning the QoS to the level that it had

before the change. The last part is important, because, for example, when changing the structure of a SQL database, all the existing programs that use the changed part of the database normally stop functioning. The measure of agility therefore includes all of the effort required to return the existing system to the QoS level that it had before the structure was changed, as well as achieving the required QoS for the new behaviour.

Based on this interpretation of productivity, agility and quality, a general methodology for evaluating and comparing data-intensive systems can be derived.

- Start by defining the overall value provided by the system and identifying proxies where possible.
- Define the data quality dimensions that are most important for the domain and how they translate into changes in quality of service and value.
- Define the data quality metrics and thresholds that are most important for the context.
- Take a given quality threshold and estimate the data maintenance, data agility, and model agility costs of maintaining that threshold over time.
- Forecast the evolution of the system and how the value it provides will depend on data and model agility and the characteristic requirements of the domain.
- The total cost of providing a service can be compared as the cost of maintaining a given quality of service over the lifetime of the system, plus the cost of model and data agility to support the required changes to the service, multiplied by their frequency.

It is important to emphasise that these comparisons are only valid at a particular quality of service level and should be made at the broadest possible level, where for example, manual processes are included where they are required in a given approach to achieve a given quality level.

7.4 ALIGNED Evaluation Ontology

In parallel with the development of the common methodological framework described above, ALIGNED has developed an ontology for the description of evaluation results (Figure 7.2). It contains classes and properties designed to capture the most important types of evaluation metrics and related concepts. The ontology is available at: https://github.com/nimonika/ALIGNED_Ontologies/blob/master/evaluate.owl

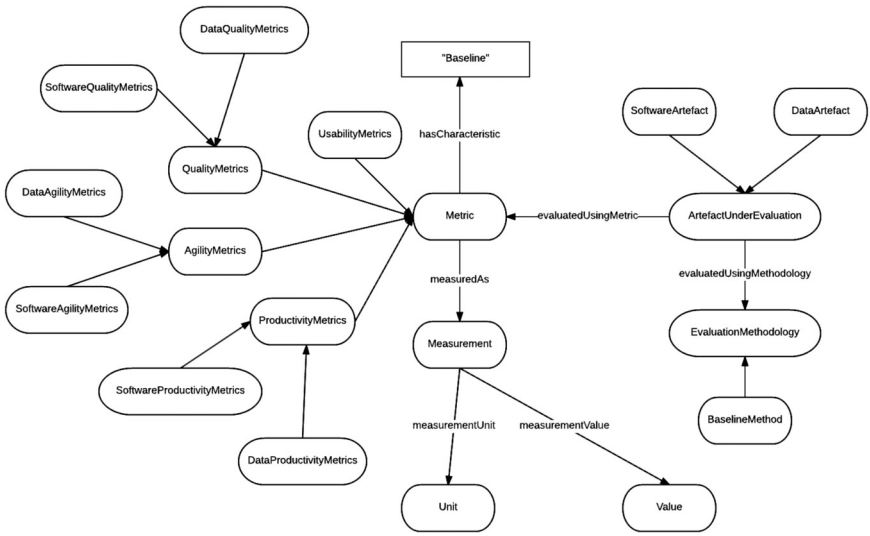

Figure 7.2 ALIGNED metrics ontology – classes.

At the core of the ontology is the concept of a metric. This is any property of the system that can be measured or analysed, such as the error rate of historical data variables in the Seshat: Global History Databank, or the number of data constraint violations on import in PoolParty. Metrics can be related to quality, agility, productivity, or agility and can be further subdivided into data and software metrics. A metric also includes information about the baseline of the metric (its initial value before any changes are made, used as a comparison to show change) and how it is measured.

These metrics are used to analyse an artefact. An artefact is any system or subsystem that is being evaluated for data and software quality analysis purposes. This also contains information about how the artefact is being analysed. The collection of metrics and the evaluation methodology provide a description of how the system in question is being analysed.