Context in Industrial Software Engineering Research

Kai Petersen, Claes Wohlin
Blekinge Institute of Technology
Box 520 SE-372 25 Ronneby, Sweden
{kai.petersen | claes.wohlin}@bth.se

Abstract

In order to draw valid conclusions when aggregating evidence it is important to describe the context in which industrial studies were conducted. This paper structures the context for empirical industrial studies and provides a checklist. The aim is to aid researchers in making informed decisions concerning which parts of the context to include in the descriptions. Furthermore, descriptions of industrial studies were surveyed.

1 Introduction

Evidence-based software engineering aims at integrating evidence from different studies. The purpose is to give recommendations which solutions are the most useful for the software industry [3]. When integrating evidence from industrial studies we argue that context has a large impact on the conclusions. Often different studies have the same object of investigation (let's say an agile process), but the process is studied in different domains, different sizes of companies, different people with different roles and skills, different cultures, and so forth. This leads to two important implications: (1) The object of investigation is implemented in different ways to fit into the context. From our experience of doing research with industry we know that companies do not adopt general methods and models one to one, but tailor them to fit their specific needs; (2) The conclusion drawn in a specific study is only true in the context in which it was conducted, although some generalization may be possible as well. Though, if a solution works in many different contexts, this is an indication that the solution is generally applicable.

This means that in order to judge whether a specific solution can be successful it is necessary to describe the context as complete and accurate as possible for the considered object of study. When having a comprehensive description of the context, evidence-based software engineering can leverage on this richer description. For example, in systematic reviews, it is possible to classify the outcome of the studies based on the context. Thereby, companies can compare their context to the contexts described in the review, allowing them to make a good choice when selecting a solution. Furthermore, it will be easier to integrate evidence based on solutions for software engineering problems (e.g. specific processes, practices, and techniques) with good context descriptions. This helps to give answers to a question such as: How should a large-scale company developing embedded systems (further information about the context can be added here) handle their quality assurance?

To aid researchers in describing the context in a way to allow better aggregation of evidence, this paper makes the following novel contributions: (1) Provide a checklist to describe the context in industrial studies; (2) Conduct a small literature survey of industrial studies to determine to what degree industrial case studies covered the context facets identified in our checklist.

2 Related Work

Runeson and Höst [4] emphasize that case studies are embedded in a real-life context/environment. They provide a checklist to guide researchers in designing good case studies, and in judging the quality of case studies. In their checklists context is implicitly considered. Thus, the checklist presented here can serve as a complement to the checklists for case study research. Kitchenham et al. [2] provide three elements considered context in case studies, namely objectives of the study, baseline against which is evaluated, and external project constraints. The external constraints can be interpreted as the environment impacting the outcome of the study regarding a specific case. Though, no guidelines on how to consider context are provided. A later paper by the same authors states that it is important to describe as much of the context as possible [1]. Overall, the related work underlines the importance of a checklist for context descriptions.

3 A Checklist for Context Documentation

The checklist is structured in six different context facets, namely product, processes, practices and techniques, people, organization, and market (see Figure 1). In the center of the context facets the object of the study is shown. The object of the study interacts with the context. For example, when having an agile process as the object of study, the process is used to develop a product, is executed by people, interacts with other processes, and is supported by practices, tools and techniques. Furthermore, the object of study is embedded in an organization (as is the case for all other context facets). The organization is operating within a market.
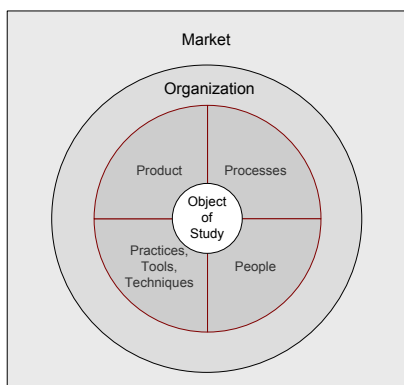

Figure 1. Context Facets

Each context facet comprises a set of context elements describing the facet. In the following we provide a description of the facets and examples of related elements.

Product: The product is the software system developed with the help of the object of study. Context elements:

- *Maturity:* The maturity of the product needs to be described. For example, by saying how long it was on the market, how many releases there were, etc.

- *Quality:* The product development is driven by different quality aspects (e.g. a development effort is undertaken to increase maintainability of the product).

- *Size:* Size of the product measured in, for example, lines of code, or number of function points. The size is an important indicator for product complexity.

- *System type:* The system can be of different types, such as an information system, embedded system, web-application, or distributed system.

- *Customization:* This means that the product is either general, or customizable and can be tailored to different market segments.

- *Programming language:* This element describes the programming language in which the system was developed.

Processes: The process describes the work-flow of the development. Context elements:

- *Activities:* The activities are different steps in the development process (e.g. specifying requirements).

- *Work-flow:* The work-flow describes the order in which activities are executed (including branching, merging, iterations, etc.).

- *Artifacts:* Artifacts are the results of the activities (e.g. the requirements specification).

Practices, Tools, Techniques: Practices, tools, and techniques describe systematic approaches that are used in the organization, and are interacting with the object of study. Context elements:

- *CASE tools:* This element describes tools that are used to support or automate software development (for example, integrated development environments, automated test tools, etc.).

- *Practices and Techniques:* This can be systematic approaches interacting with the object of study. For example, when studying an agile process practices could be time-boxing, frequent integration, or pair programming.

People: The human factor is very important when studying software development, as it has a major impact on the success of software development. Thus, the factor has to be covered in the context. Context elements:

- *Roles:* This element describes what type of roles are involved in using the object of study. This includes a description of the main responsibilities and authorities associated with the roles. For example, which roles make use of a specific technique studied in a project.

- *Experience:* Experience is concerned with the areas that people affected by the object of study have worked in, and for how long. Furthermore, education and trainings are of interest.

Organization: The organization describes the company structure in which the other context facts and the solution are embedded in. Examples for elements are:

- *Model of overall organization:* The organization model describes how the company is organized, such as matrix-organization or hierarchical organization. Furthermore, it could be discussed whether the organization is flexible or strict.

- *Organizational unit:* The organizational unit is a part of the organization closely interacting with the object of study. For example, this can be a project (temporary existing unit) or department (permanent unit). The organizational unit can be complemented with management related information such as responsibilities of the unit, and size measured as number of persons involved.

- *Certification:* This tells whether the organization is certified (e.g. ISO/CMMI).

- *Distribution:* The organizational units are either collocated or distributed (nationally or internationally).

Market: The market represents the customers and competitors. Elements describing the market are:

- *Number of customers:* This element refers to bespoke versus market-driven development. In bespoke development the customer is known and can be addressed, and a contract is established already between developer and customer. On the other hand, market-driven development targets a large and open market of potential customers that might buy the product after release.

- *Market segments:* The market segments describe groups of customers that share a common need.

- *Strategy:* The strategy describes how to address the market in the long-term. For example, the company can run a niche-strategy where they compete with a special product (e.g. in terms of extremely high quality) or a strategy to compete on a low price.

- *Constraints:* The market can put constraints on software development, such as a very short time-to-market, or certifications (e.g. CMMI).

We would like to emphasize that context has a large impact on the solution, and also that context elements influence each other. For example, when having several market segments that share a common need, it is normal to develop a product that is customizable to address as many market segments as possible with little effort, which would call for product-line approaches as a solution. How the solution would look like is further influenced by the complexity of the product. Another example is the strategy on the market-level. If we would have a strategy to target a market with low costs, then testing solutions would look different in comparison to a market where a niche is addressed, calling for very high quality.

Regarding the context elements it is important to mention that we do not claim completeness. The checklist is rather a first step to raise awareness that there are multiple facets that ought to be covered in context descriptions. In further work, the aim should be for a more complete description of context elements, which would result in a comprehensive framework for context description.

Not all elements can be described in industrial studies. Rather a sub-set of elements should be described that are most relevant for the conclusions drawn in relation to the object of study. That is, the checklist should be gone through and informed decisions should be made of what to include and what not to include. To motivate researchers considering context, it is important to emphasize context more in guidelines for systematic reviews and case study research.

4 Context in Industrial Studies

The research question (RQ) for the survey is: *To what degree do industrial studies cover the context facets presented in this study in their descriptions?*

In order to provide at least a partial answer to the question, we limit our search to the journal of empirical software engineering (EMSE). The reasons for doing so are: (1) the journal has an explicit focus on publishing high quality empirical work; and (2) there are no strict page limitations which does not give authors a reason to neglect descriptions of the context.

Within the journal we focus on empirical studies conducted in industry. In order to identify these we searched the journal with the following search string:

- Industrial OR case study IN title/abstract

Only industrial studies are included. Experience reports, short papers, surveys, experiments and case studies of open source software were excluded. To identify the context facets described in the studies we read the method sections of the identified articles and noted down which context facets and elements were covered.

One threat to validity is the bias of the main author when classifying the articles and extracting the context elements. To reduce the risk the author used the keywords as an aid to classify the papers. Furthermore, a template for data extraction was developed. Another threat is that we only focused on the EMSE journal, which reduces the coverage of topics. However, we believe that similar observations will be made when considering other sources.

Results: The search returned 56 studies, of which 27 were industrial studies relevant for the analysis. Only three studies cover five and more context facets. Eleven studies cover three to four facets. The remaining part (13 studies) only cover less than three facets.

As which context facets to cover depends on the object of study we grouped the studies. The identified groups regarding the object of investigation are: software reliability (SR) covering topics such as predicting faults in fault-prone

Table 1. Coverage of Research Facets Considering Objects of Investigation

Facets	SR(8)	SP(4)	QA(7)	SC(2)	UML(2)	RM(1)	RES(1)	AE(1)	CS(1)	Total(27)
Product	8	4	6	2	2	1	1	1	1	26
Process	2	2	3	1	1	0	0	0	0	9
Pract./Tool	3	2	5	0	1	1	0	0	0	12
People	0	1	1	0	2	0	0	0	1	5
Organization	0	4	4	1	2	1	0	0	1	13
Market	0	2	1	0	1	0	0	0	0	4

models, reliability growth models, and quality predictions; software process (SP) covering topics such as evaluating processes, and software process improvements; quality assurance (QA) evaluating quality assurance approaches such as testing techniques, reuse for QA, or inspections; size and cost estimation (SC); object orientation and UML (UML); requirements management (RM); software restoration (RS); architecture evaluation (AE); maintainability through code smells (CS).

Table 1 shows the coverage of the facets grouped by the objects of investigation mentioned before. The top line of the table shows the abbreviation of the objects of investigation and the total number of papers in brackets. The table allows us to make the following observations:

- Almost all studies consider it important to provide information of the product being studied. This mostly included information of the type of product, and its complexity.

- Studies investigating SP, QA, and the UML studies have the highest coverage of facets. This indicates that some objects of investigation seem to aim at a higher coverage of the context facets. This supports our proposal regarding the usage of the checklist, i.e. identifying those context facets and elements that are likely to influence the outcome of the study. This will differ depending on the object of investigation.

- Even though there is a trend visible which objects of investigation require higher coverage of context facets, the studies did not agree on which context facets are important to mention. For example, only a sub-set of studies investigating SR consider it important to describe processes and practices/tools (see Table 1). Similar patterns can be observed for the other objects of investigation as well (e.g. processes, practices/tools, and market for SP). Thus, this supports the need for a checklist of context facets and elements to increase awareness of potentially relevant context facets, and by that aiding the researcher in making informed decisions of what to include and not to include. Furthermore, the checklist can be used in different commu-

nities (focusing on a specific object of investigation) to discuss and arrive at a common ground for context descriptions.

5 Conclusion

This paper proposes a checklist for the description of context, consisting of context facets (product, process, practice/tools, people, organization, market) and related context elements. A literature review of industrial studies showed that studies investigating a similar object do not agree on which context facets are important to mention. Our checklist aims to help researchers to take informed decisions on what to include and not to include. The checklist also serves as a basis for discussion to reach a consensus in different communities which context facets/elements are most important for their focus area. In future work the checklist has to be further extended. We also plan to do in-depth analysis of a selection of articles focusing on context facets as well as elements.

References

[1] B. Kitchenham, S. L. Pfleeger, L. Pickard, P. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Software Eng.*, 28(8):721–734, 2002.

[2] B. Kitchenham, L. Pickard, and S. L. Pfleeger. Case studies for method and tool evaluation. *IEEE Software*, 12(4):52–62, 1995.

[3] B. A. Kitchenham, T. Dybå, and M. Jørgensen. Evidence-based software engineering. In *Proceedings of the 26th International Conference on Software Engineering (ICSE 2004)*, pages 273–281, 2004.

[4] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.