Systematic Software Development

A State of the Practice Report from Germany

Kuhrmann, Marco; Méndez Fernández, Daniel

Published in:
Proceeding of the 10th IEEE International Conference on Global Software Engineering

DOI:
10.1109/ICGSE.2015.11

Publication date:
2015

Document version
Submitted manuscript

Citation for pulished version (APA):
Kuhrmann, M., & Méndez Fernández, D. (2015). Systematic Software Development: A State of the Practice Report from Germany. In *Proceeding of the 10th IEEE International Conference on Global Software Engineering* (pp. 51-60). IEEE. https://doi.org/10.1109/ICGSE.2015.11

Systematic Software Development: A State of the Practice Report from Germany

Marco Kuhrmann
The Mærsk Mc-Kinney Møller Institute
University of Southern Denmark
Campusvej 55, DK-5230 Odense M, Denmark
kuhrmann@mmmi.sdu.dk

Daniel Méndez Fernández
Faculty of Informatics
Technische Universität München
Boltzmannstr. 3, D-85748 Garching, Germany
mendezfe@in.tum.de

***Abstract*—The speed of innovation and the global allocation of resources to accelerate development or to reduce cost put pressure on the software industry. In the global competition, especially so-called high-price countries have to present arguments why the higher development cost is justified and what makes these countries an attractive host for software companies. Often, high-quality engineering and excellent quality of products, e.g., machinery and equipment, are mentioned. Yet, the question is: Can such arguments be also found for the software industry? We aim at investigating the degree of professionalism and systematization of software development to draw a map of strengths and weaknesses. To this end, we conducted as a first step an exploratory survey in Germany, presented in this paper. In this survey, we focused on the perceived importance of the two general software engineering process areas project- and quality management and their implementation in practice. So far, our results suggest that the necessity for a systematic software development is well recognized, while software development still follows an ad-hoc rather than a systematized style. Our results provide initial findings, which we finally use to elaborate a set of working hypotheses. Those hypotheses allow to steer the adaptation of our instrument in the future to eventually facilitate replications toward a more comprehensive theory on systematic globally distributed software development in practice.**

***Index Terms*—software development; software process; test process; software process improvement; survey research**

I. Introduction

Today, software development constitutes a vital component of our economy, strongly contributing to business value creation. As software development goes global, many companies change their business model in order to support global software development (GSD; [1]), for example, by appointing globally distributed resources for coding and testing activities in expectation of cost reductions. However, several studies (e.g., [2], [3]) illustrate the problems coming along with GSD. Therefore, companies also elaborate the key activities in the software development process, which are considered value-creating or business-critical to keep those onshore thus increasing the probability of project success.

Project success factors are well-explored in research. A number of studies investigate these factors from different angles, such as GSD [4], use of software processes [5], [6], feedback on mobile apps [7], understanding of software [8], and especially software process improvement (SPI) [9]. However, available studies do not yet allow for providing an overall picture of the state of practice of systematic software development, either because those studies focus on isolated development facets or because they have flaws hampering their reproducibility. One prominent study is, for instance, the Chaos Report [10] that aims at providing such an overall picture. However, this study is known to have some critically discussed flaws [11] whereby it—if at all—can only provide first superficial impressions. Other studies addressing the general perspective were mainly conducted in the context of research projects, e.g., [12], [13]. Although these studies follow a rigorous research method, they lack in continuity, i.e., they are usually conducted only once and trend analyses over a longer period of time was not in scope of the researchers. Nonetheless, we find a number of studies addressing quite focussed research questions, e.g., the investigation of the state of practice in requirements engineering [14], quality analyses of mobile apps [15], success factors for GSD [16], or success factors for SPI [9], [17]. By focusing on selected aspects, these studies provide important insights, which are, however, hard to integrate into a comprehensive global software development context. Furthermore, most of these studies are also conducted once providing one single snapshot of current practice and, thus, we still lack the option to analyze trends over time. Only few studies opt for a continuous replication to analyze effects over time and/or in different contexts, e.g., [14], [18].

Problem Statement: We still lack an evidence-based instrument that supports creating a comprehensive view on the degree of professionalism and systematization of software development. Such an instrument needs to allow for a continuous study replication to analyze trends over time to enable researchers and practitioners to investigate the state of current practice, to direct research in a problem-driven manner, and to study the impact of proposed—particularly academic—solutions. In order to support the variety of available topic-specific studies, the instrument must provide a framework that allows for relating these studies and to analyze how the specific findings contribute to the improvement of the overall software engineering process.

Objective: Our objective is to lay an empirical foundation to determine the degree of systematization of software development in a global context. We aim at developing an open and reproducible instrument to continuously evaluate software development from the engineering perspective. Therefore, we focus on the general "top-level" software engineering processes *project-* and *quality management* to provide a bigger picture and to provide a framework to relate more specific studies. To support gathering a broader study population, we opt for survey research.

Contribution: In this paper, we present three contributions. First, we contribute initial results from a survey on the state of the practice of systematic software development conducted in Germany. We investigated the key process areas of project- and quality management (with a particular focus on test management). Furthermore, we studied continuous SPI to analyze problem awareness and the readiness to systematically improve current project- and quality management. Our second contribution is the survey instrument used and the data obtained. We provide insights into the development of this instrument, and disclose our raw data. The findings provided in the paper at hand are based on the *3ProcSurvey* [19], which is the first publicly conducted survey instance aiming at creating a reference dataset. Third, based on our results, we elaborate a set of working hypotheses that indicate into specific directions and trends based on the data obtained from Germany. Those hypotheses allow to steer the adaptation of our instrument to eventually facilitate the replications toward a more comprehensive theory on systematic globally distributed software development in practice.

Outline: The remainder of the paper is structured as follows: In Section II, we discuss context and background of our study, and related work. In Section III, we present the research design, and the respective results in Section IV. Finally, we present a first working theory in Section V, before concluding the study in Section VI.

II. Related Work

Finding proper studies to investigate the state of practice in software engineering is difficult [20]. Several studies were conducted over time to provide an overview of certain domains, state of the practice, and to elaborate gaps, challenges, and success factors. Yet, many studies are limited to a specific context normally not implementing a continuous instrument.

For instance, in Germany, the SUCCESS study [12] was conducted in 2005/2006 to investigate factors influencing software project success. In 2006, in the project IOSE-W^2 [13], a Germany-wide study, was conducted to elaborate the state of practice regarding process use and improvement in the context of GSD and in the development of software product lines. The study revealed that companies use a variety of software processes, barely use CMMI certification programs, and consider project management as the one management discipline that is highly versatile, whilst development and quality assurance are considered barely versatile. Furthermore, although companies support the assumption that standardization positively affects software development, reuse or product-line-based software development was not widely adopted at that time. Cusumano et al. [6] surveyed 104 projects operated in India, Japan, US, and Europe (and other) to study the state of the practice of using development practices. Their study revealed a diverse set of development practices. Furthermore, they concluded that developers write more code if they had a more complete specification, that good designs correlate with low defect rates and, notably, that adopting practices associated with flexible processes appears to compensate for incomplete specifications. Similarly, Jones [21] conducted a retrospective analysis of approx. 12,000 projects. By analyzing 25 standard practice categories, 30 occupation groups, defining 6 project types and 6 project size categories, he found no universally deployed software process. However, from the analyzed data, he concluded that projects implementing any kind of formal design have better success rates and, furthermore, that a continuous quality control seems to be the best success indicator. However, studies like [6], [12], [13], [21], or [5] are not repeated on a regular basis. Therefore, those snapshots do not allow to infer trends to visualize developments/improvements over time, or to project future developments. Even Cusumano et al. [6] only refer to previously conducted investigations yet not providing comprehensive trend analyses.

In other domains than general software engineering, an increasing number of studies investigates problems and success factors. Especially for SPI and GSD, the investigation of success factors plays an important role, e.g., [2], [3], [9]. Similar, there are studies at our disposal that monitor the implementation and success of agile methods, e.g., [18], [22]–[25]. A distinctive approach is followed by the NaPiRE study [14] in which, based on an international research network, the state of the practice in requirements engineering, including problems, causes, and effects, are investigated on a regular basis.

Inspired by international continuous endeavors to elaborate a reliable theory of the current state of (discipline-specific) practice, our study is intended to establish an instrument to monitor professionalism in and systematic software management and development on a regular basis. Therefore, we selected key disciplines crucial for industrialized software development. The initial survey instrument was developed based on the experiences from past studies in order to investigate expectations as well as actual approaches. The survey already allows for elaborating a first impression of gaps ("to-be" vs. "as-is") and to derive areas of action. Due to the (planned) repetition, the survey aims at outlining trends and making projections regarding future trends in the long rung.

III. Study Design

In this section, we describe the overall study design. We introduce the research questions, provide the data collection and analysis procedure, and discuss the validity procedures.

A. Research Questions

The goal of the study at hand is to study the state of practice of software development, in a first step focused on project- and quality management using German companies as a sample population. The study addresses industry expectations as well as practical implementation and also aims at investigating if companies are aware of gaps and accordingly implement improvement programs. In order to address the research goals, we formulate the research questions from Table I.

TABLE I
RESEARCH QUESTIONS AND RATIONALE.

No.	Question
RQ_1	*What is the expectation of a good software development?* aims at investigating the expectations of software companies regarding development processes (incl. project organization and management) and quality management, e.g., project cost, quality, collaboration, and tool support.
RQ_2	*What is the status quo in software development?* aims at determining the current state of practice, how these development- and quality management processes are actually implemented, e.g., the actual process, company-wide and project-specific processes, tailoring, metrics, and controlling.
RQ_3	*How are software processes currently improved?* aims at elaborating how software companies conduct software process improvement (SPI) by, e.g., analyzing the implementation of SPI programs.
RQ_4	*Are there observable patterns of expectations and the current state?* Finally, RQ4 aims at investigating whether patterns of expectations exist to enable (future) analysis of gaps between current practices and the expectations the participants have.

B. Data Collection Procedure

We used a questionnaire[1] comprising 33 questions in total (6 devoted to meta data and general information, 9 questions on expectations and 13 on state of practice, and 5 questions on SPI). Question types were single- and multiple choice, free text, and rating (mapped to a 5 point Likert scale with an option for the respondents to select that they have no opinion or don't know the answer), all together yielding in 164 variables. The questionnaire targets practitioners and was developed based on several previously conducted surveys (e.g., [26] and the Iqnite study[2]), whereas the questions of the IOSE-W^2 survey provided the master template.

The survey remained open from October 2012 to January 2013. Invitations were sent using personal contacts, mailing lists and events of the Federal Association for Information Technology, Telecommunications and New Media (BITKOM), German Computer Society (Gesellschaft für Informatik, GI e.V.), Twitter, and LinkedIn. We intentionally sacrificed the ability to calculate the response rate to gain a high visibility of our first exploratory study within the communities.

C. Analysis Procedure

The data of the survey contains a mix of information about the participants' background, the domains the companies are working in and the standards used, as well as expert opinions of the participants involved in those companies. Moreover, we do not exclusively rely on random samples as we opted for specific distribution channels we had access to. Finally, regarding the expert opinions, we express the subjects' opinions with Likert scales, which are specified with ordinal scales with no interval data, i.e., the distances between the single values in the variables (e.g., "strongly agree", "agree", and "disagree") are not equally distributed. In other cases, we have open questions or certain variables in nominal scales, e.g., the companies either apply certain methods or they do not. Therefore, we rely on different procedures for the data analysis.

For the analysis of the answers given to RQ_{1-3}, we rely on descriptive statistics without hypothesis testing given we had no underlying theory of testable expectations but that the goal is to elaborate such a theory. Furthermore, we answer RQ_4 performing a correlation analysis over all variables. In the correlation analysis, we check whether we can find general relationships between answers to our questions. Therefore, we calculate Kendall's τ as correlation coefficient for ordinal data (and binary answers coded as 0 and 1). We are only interested in 'stronger' relationships whereby we filter τ values larger than 0.5 and smaller than -0.5 (cf. discussion in [27]). Given that we have little control over the population (regarding, e.g., its representativeness) and the sample size, we do not expect a high significance or even a high expressiveness of a significance test at all. We therefore take the remaining relationships as candidates for inclusion and discuss if there are reasonable explanations or interpretations provided by existing (further) studies.

D. Validity Procedures

This first survey round has more a curiosity-driven, exploratory character whereby we intentionally sacrificed the possibility to calculate the response rate. Similarly, as we used our data to also explore a first contextualization of our population, we cannot yet critically reflect on the representativeness of our sample (let alone as we cannot guarantee that one respondent represents one company only). Still, our instrument is based on IOSE-W^2 questionnaire which should increase the construct validity of the survey (similar questions, similar variable ranges, etc.). As a means to additionally increase the validity, we conducted a self-contained, iterative validation phase with 4 researchers before initiating the survey. In that phase, we filled out the implemented questionnaire without, however, a pilot data analysis given the exploratory character of the survey. Due to this character, we do also not make any claims about the external validity which can only be supported by independent replications for which we lay the foundation with the survey at hands. Finally, during the data analysis and the interpretation, we relied on researcher triangulation to support for an unbiased interpretation of the data.

IV. STUDY RESULTS

In this section, we present the study results. In Sect. IV-A, we give an overview of the study population to deliver context information. In subsequent sections, we answer our research questions in a step-wise manner.

[1]Due to space limitations, the whole questionnaire is available for download from: http://goo.gl/UBS1mK. The package also contains the raw data and the correlation analysis files.

[2]For further information, see: http://iqnite2012.questionpro.com/

A. Study Population

The questionnaire was open, whereas the contributing institutions (BITKOM, GI, and TUM) used their mailing lists and social media channels to attract participants. All together, 474 people viewed the questionnaire, 97 started the survey, and 38 completed the survey. However, as all questions were intentionally kept optional, the number of responses varies between n_{min}=35 and n_{max}=51. Therefore, in the following presentation of the results, we summarize the results in relation to the respective number of responses n.

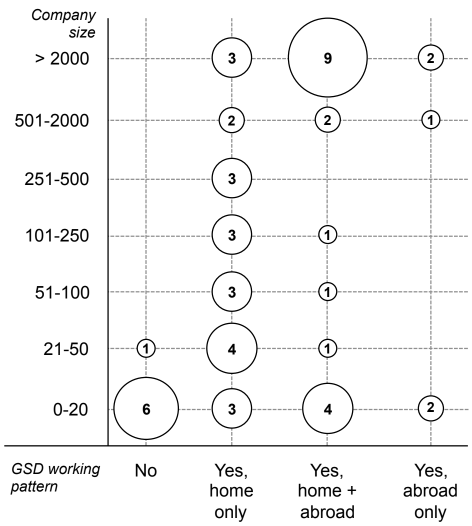

Fig. 1. Map of participants' company size and GSD work pattern (n=51).

Figure 1 shows the distribution of the sizes of the companies in which the respondents worked. We have companies of all size and working in different domains, e.g., "classic" software development, automotive, medicine, logistics, and avionics. Only 7 out of 51 companies mention to be not involved in GSD. For the remaining companies, regardless of their size, (globally) distributed work is part of their general project organization, and even very small companies team up with partners in other countries.

TABLE II
Relevance of disciplines for previous project success and expectations for improvement opportunities (Likert scale: 1=not relevant at all, . . . , 5=strongly relevant, Presentation: mode (mean), n=50).

	PM	DEV	TST
Importance for project success so far	5 (4.63)	4 (4.26)	4 (4.28)
Expected improvement potential	5 (3.92)	4 (3.96)	4 (3.96)

Apart from the metadata-related questions, the questionnaire started with two general questions to gather the respondents' opinion regarding success factors of past projects and regarding the biggest improvement potential. Table II illustrates that all respondents consider the development process (DEV) and test processes (TST) relevant to the project success. 76% of the respondents consider project management (PM) highly relevant for the project success (only one respondent selected "don't know"). Table II also shows the respondents' opinion regarding the biggest improvement potential. Again, DEV and TST are mentioned of high potential, and PM is considered of very high potential.

TABLE III
Expectations regarding the use of development and test processes (Likert scale: 1=not relevant at all, . . . , 5=strongly relevant, Presentation: mode (mean), dn=don't know, n=variable).

We can...	DEV	dn	n	TST	dn	n
reduce project cost	4 (3.79)	2	40	4 (3.34)	3	38
improve product quality	4 (4.20)		40	5 (4.32)	1	38
improve communication within/between projects	4 (3.95)	1	40	4 (3.44)	1	37

TABLE IV
Motivation to use processes (Likert scale: 1=not relevant at all, . . . , 5=strongly relevant, Presentation: mode (mean), dn=don't know, n=variable).

	DEV	dn	n	TST	dn	n
Regulations and (external) standards	4 (3.34)		38	4 (2.81)		37
Better PM integration	4 (3.43)	1	38	4 (3.59)		37
Better tool support	4 (3.03)	1	39	4 (3.39)		36
Subject to contract	4 (3.11)	1	39	4 (2.88)	2	36
Support for distributed, collaborative work	4 (3.26)		39	4 (3.31)	1	36
Early deviation detection	4 (3.59)		39	5 (4.08)		37
Measurement/progress tracking	4 (3.95)		40	5 (4.08)		37
Flexibile staff allocation	4 (2.74)		39	3 (2.78)		36
Increased efficiency	4 (3.77)	1	40	4 (3.70)		37
Increased effectiveness	4 (3.65)		40	4 (3.59)		37
Better plan-ability (cost, time, etc.)	4 (3.90)		40	4 (4.08)		37

B. RQ_1: Expectations

The first part of the study addresses the expectation of the participants regarding the use of project management, development, and test processes. The questionnaire addresses these points by asking for expectations for development and test processes, and by relating them to project management.

Table III summarizes the general expectations and shows that respondents generally expect improvements regarding communication, product quality, and cost reduction.

In order to determine drivers that motivate the use of processes, participants were asked for a rating of influencing factors and how they actually influence the selection and use of processes. Table IV summarizes this rating and shows that especially the opportunity for an early detection of planning deviations, for measurement, and progress tracking is considered highly relevant for TST, while flexible staff allocation is considered neutral. Furthermore, Table IV shows a uniform picture as all other influencing factors are rated equally relevant.

TABLE V
EXPECTED DRAWBACKS OF PROCESS USE (LIKERT SCALE: 1=NOT RELEVANT AT ALL, ..., 5=STRONGLY RELEVANT, PRESENTATION: MODE (MEAN), DN=DON'T KNOW, N=VARIABLE).

	DEV	dn	n	**TST**	dn	n
Increased cost	2 (2.59)	1	40	2 (2.70)		37
Increased administration	4 (3.00)		39	4 (3.00)		36
Increased communication	3 (2.85)		40	3 (2.97)		37
Reduced project speed	2 (2.50)	2	40	2 (2.58)		36
Reduced flexibility	4 (2.93)		40	2 (2.49)		37
Reduced efficiency	2 (2.23)		40	2 (2.33)	1	37
Reduced effectiveness	2 (2.10)		40	2 (2.22)	1	37

TABLE VI
EXPECTATIONS REGARDING TOOL SUPPORT (SINGLE SELECT, N_{DEV}=39, N_{TST}=37).

We'd like to have...	**DEV**	**TST**
one integrated tool for everything	10	10
one integration platform	17	16
individual tools in combination	10	10
Other	2	1

However, deploying a process of any kind—especially if triggered externally—is considered critical, as new or different processes might affect projects or the whole organization. Therefore, we asked for potential drawbacks, which are shown in Table V. The data shows for DEV as well as for TST that increased administrative effort is expected, and that respondents expect a reduction of flexibility for DEV. However, respondents do neither expect increased project cost nor impacts on project performance (in terms of "speed", efficiency, and effectiveness).

Finally, since tools can provide support when deploying new/different processes, we asked the participants for their expectations regarding appropriate tools to support the development and test processes. Table VI presents the results and shows that for DEV as well as for TST respondents prefer integration platforms in which different tools can be integrated to create a flexible project environment. However, 10 respondents prefer a (free) combination of individual tools to support DEV and TST. Together with the integration platform, more than the half of all respondents opts against standardized and (fully) integrated tool solutions, but prefers individual and flexible solutions.

C. RQ_2: State of the Practice

In the following, we present the state of practice as determined by the responses. In Sect. IV-C1, we present the reported state of implementing software development processes and, respectively, in Sect. IV-C2 for test processes. In Sect. IV-C3, we provide an integrated perspective, which includes the (overall) project management and practices regarding process selection. Finally, as we expect there might be a gap between expectations and actual implementations, we are interested in the respective companies' perception of improvement processes, which we present in Sect. IV-D.

1) Development Processes: We are interested in which software processes are applied in practice and how the participants perceive these processes. In Table VII, we present the list of mentioned software processes.

TABLE VII
USED SOFTWARE PROCESSES (YES/NO, MULTIPLE SELECTIONS, N=37, PERCENTAGE RELATIVE TO MENTIONS: DEV=93).

Software Process	**QTY**	**%**
Rational Unified Process (pure)	0	0.00
Rational Unified Process (modified)	6	6.45
Open Unified Process	0	0.00
V-Modell XT (reference model, pure)	6	6.45
V-Modell XT (modified)	15	16.13
V-Modell 97	5	5.38
W-Modell	2	2.15
Hermes	0	0.00
Extreme Programming (XP)	9	9.68
Scrum	25	26.88
Kanban	9	9.68
Crystal family	1	1.08
Feature Driven Development (FDD)	2	2.15
None	2	2.15
Other	11	11.83

Respondents gave 93 mentions of which the V-Modell family and Scrum make about the half of all mentions. Only two respondents state to not apply any process, and 11 mention to apply other than the listed ones. Furthermore, our data shows that respondents combine 2-3 different development models on average, which indicates a process ecosystem highly specialized to the respective company contexts. Therefore, we are also interested in the perception of the different software processes, which is shown in Fig. 2. The figure shows that the respondents are generally satisfied with their software processes (general disagreement with the given statements).

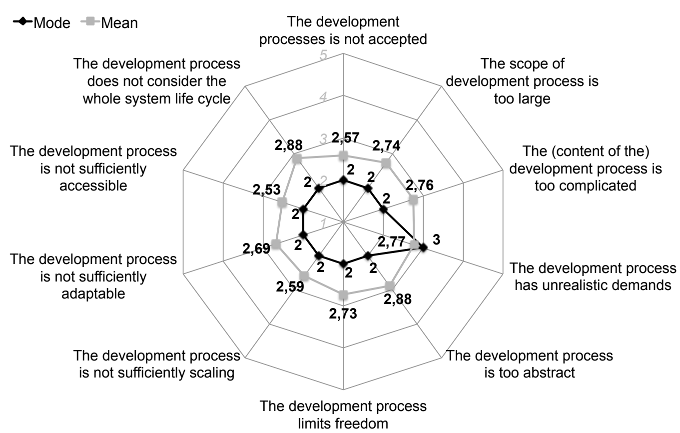

Fig. 2. Perception of the development process used (Likert scale: 1=completely disagree, ..., 5=completely agree).

2) Test Processes: Table VIII shows the survey's outcomes on the practically applied test processes. The table shows the majority of the respondents using the ISTQB Fundamental Test

Process[3], yet, about 30% using no standardized test process, i.e., following own test approaches (e.g., test-first or risk-based testing) or no (formal) test process at all (3 extra comments).

TABLE VIII
USED TEST PROCESSES (YES/NO, MULTIPLE SELECTIONS, N=26, PERCENTAGE RELATIVE TO MENTIONS: TST=27).

Test Process	**QTY**	%
Fundamental Test Process (ISTQB)	16	59.26
ISO 29119	2	7.41
TMAP/TMAPnext	1	3.70
Other	8	29.63

3) Integration in the Company/Project Context: After having determined kind and frequency of the used development- and test processes, we provide an integrated view in the following. At first, Table IX shows how the development and test processes are integrated with a project management standard. The table shows that PMI-based standards, Prince2, and the V-Modell XT are the most frequently used PM methods for integrating the different processes. However, the categories "None" and "Other" together show that about one third of the respondents do not follow a standardized approach (respondents mention, inter alia, ISO 15504, PDCA, Agile PM, or company-specific standards). Furthermore, the responses indicate that multiple frameworks are adopted, e.g., Prince2 and V-Modell XT, or IPMA and DIN/ISO 69901.

TABLE IX
INTEGRATION OF PROCESSES AND PROJECT MANAGEMENT (YES/NO, N_{DEV}=37, N_{TST}=36, PERCENTAGE RELATIVE TO MENTIONS: DEV=51, TST=50).

	DEV	%	**TST**	%
PMI/PMP/PMBoK	8	15.69	8	16.00
Prince2	8	15.69	7	14.00
MSP	0	0.00	0	0.00
IPMA	4	7.84	4	8.00
V-Modell XT (as PM method)	10	19.61	11	22.00
DIN/ISO 69901	5	9.80	4	8.00
None of these	11	21.57	12	24.00
Other	5	9.80	4	8.00

The number of available approaches and the options for combinations call for defining company standards, rules for process selection, and definition of project-specific processes. In Table X, we provide an overview of the degree of standardization of processes and the responsibilities regarding the definition of the project-specific process. The data shows that the individual process selection on a per-project base seems to be the normal case—especially for test processes.

Furthermore, Table XI shows how the actual adoption of the selected processes is done in the project context. The data indicates that a project manager usually tailors the process at the beginning of the project. Furthermore, 21% of the respondents conduct a situation-specific method/practice selection during the project. On the other hand, we see 23% of the respondents conducting the tailoring without specific rules based on their own expertise, and 12% performing no tailoring at all. That is, about one third of the respondents do not tailor the project-specific process in a systematic and/or reproducible manner.

[3]For further information, see: http://istqbexamcertification.com/what-is-fundamental-test-process-in-software-testing/

TABLE X
PRACTICE OF SELECTING PROJECT-SPECIFIC PROCESSES (YES/NO, N_{DEV}=34, N_{TST}=37).

The project process is...	**DEV**	%	**TST**	%
standardized for all projects	12	35.29	8	21.62
project-specific, but centrally selected	7	20.29	8	21.62
individually selected per project	15	44.12	21	56.76

TABLE XI
PRACTICE OF TAILORING PROJECT-SPECIFIC PROCESSES (MULTIPLE SELECT, YES/NO, N=37, PERCENTAGE RELATIVE TO MENTIONS: DEV=52).

The project-specific process tailoring is...	**QTY**	%
performed during the project (situation-specific)	11	21.15
performed by the project manager in the beginning	21	40.38
performed according to specific tailoring rules	12	23.08
supported by a tool	2	3.85
The process is not tailored at all.	6	11.54

D. RQ_3: Process Improvement

As we expected gaps between the general expectations and the actual implementation of the different processes, we are also interested into the respondents' awareness of these gaps, and if companies implement appropriate software process improvement (SPI) strategies to initiate and steer improvements. Furthermore, we are interested in elaborating whether companies conduct SPI in a systematic manner.

Therefore, the first question aims at revealing if companies implement an improvement strategy and how this strategy is embedded into the organization and project context. Table XII shows that only 19% of the respondents state that their company sets up a continuous improvement program, about 60% only sporadically conduct SPI initiatives, and 27% do not implement any SPI program at all.

TABLE XII
IS YOUR SPI PROCESS INTEGRATED WITH ORGANIZATION AND PROJECT MANAGEMENT (SINGLE SELECT, YES/NO, N=37).

	QTY	%
Continous improvement (in terms of CMMI level 5)	7	18.92
Sporadically conducted	10	27.03
independent from project management	10	27.03
SPI is not implemented at all	10	27.03

Furthermore, we are interested in knowing whether SPI programs aim at achieving specific goals, e.g., certification. Table XIII shows that about 40% of the companies conduct SPI

TABLE XIII
Does your SPI process address certification (single select, yes/no, n=36).

The SPI activities addresses...	**QTY**	%
a CMMI certification	6	16.67
an ISO 9001 certification	3	8.33
an ISO 15504 certification	5	13.89
Our SPI program does not aim at any certification	14	38.89
Other	8	22.22

TABLE XIV
Handling of lessons learned from projects (single select, yes/no, n=36).

Lessons learned are collected...	**QTY**	%
systematically, by an own unit/department	6	16.67
systematically, by appointing external consultants	1	2.78
only sporadically	9	25.00
rarely or through individual employees only	13	36.11
Not at all	2	5.56
Other	5	13.89

to achieve a certification, whereas the rest either does not aim at specific certifications or conducts SPI for other purposes.

Based on previously published studies (e.g., [14], [28]), we expected companies being reluctant toward normative SPI programs (certifiable against external norms). Therefore, we included questions on the general perception of SPI programs in the questionnaire. Figure 3 illustrates the outcomes showing that respondents admit that an SPI program could positively affect product quality and help to learn about strengths and weaknesses. However, standard SPI programs are considered too demanding and poorly tailorable to the respective company context.

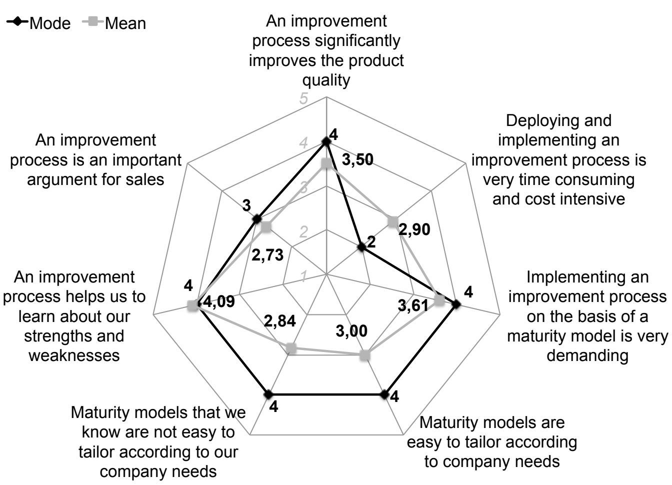

Fig. 3. Respondents' perception of software process improvement programs (mode and mean values on a 5-point Likert scale).

In order to learn about strengths and weaknesses and to improve software development, companies need to know about their performance and, furthermore, need to implement some kind of knowledge management to make lessons learned available thus enabling project teams to avoid errors and to use proven practices. In Table XIV, we provide the outcomes regarding the management of lessons learned in past projects and if such lessons are managed in a systematic way. The table shows only 19% of the responding companies implementing a systematic knowledge management. The rest manages lessons learned—if at all—only sporadically or individually.

Another aspect of enabling companies/projects to improve, e.g., performance and quality, is to collect (performance) data and to make this data available by providing aggregation and analyses. Therefore, we asked the participants for the currently implemented approach to measure SPI-related performance data. Table XV provides the results and shows that more than the half of the responding companies states to not collect and analyze performance data at all.

TABLE XV
Implementation of measurement activities to collect SPI-related performance data (single select, yes/no, n=35).

Data collection and analysis are:	**QTY**	%
Manual data input, manual analyses and aggregation	12	34.29
Manual data input, automatic analyses and aggregation	3	8.57
Automatic data input, analyses and aggregation	2	5.71
We do not collect and analyze SPI performance data	18	51.43

E. RQ_4: Patterns of Expectations

To elaborate patterns, we performed a correlation analysis by computing Kendall's τ. The correlation analysis yields in 261 positive correlations and 54 negative correlations. We analyzed the resulting correlation matrix manually and could reveal correlations that were in tune with our expectations, such as the company size correlating with the motivation respondents see in the implementation of software processes as a means to support quality assurance activities. However, it was not possible for us to manually identify candidates or clusters that would justify a deeper analysis in the paper at hands.

Therefore, we manually inspected randomly selected correlating pairs of variables with striking negative or positive τ values. We took those variables in a second step and analyzed whether they had also positive correlations to other variables and could identify overlaps. For example, in the expected disadvantages of using a development process, variable 48 showed several overlaps, such as *reduced efficiency* (v48) with *reduced performance* (v44) or with *reduced effectiveness* (v49) having the following correlations:

$$v48 \xrightarrow{+0.55} v44$$

$$v48 \xrightarrow{+0.86} v49$$

Table XVI summarizes all positively and negatively correlated variables (including the τ value and a brief description of the variable) resulting from the procedure described above. For example, for v48 as well as v68, we see that expected disadvantages regarding efficiency reduction (DEV) and increased

TABLE XVI
SELECTED CORRELATIONS FROM THE RESULT SET (INCLUDING KENDALL'S τ AND A BRIEF DESCRIPTION OF THE INDIVIDUAL VARIABLES).

Var.	τ	Variable description
v48: Expected disadvantages of using a development process (DEV): reduced efficiency		
v40	-0.55	Expected adv. of DEV: increased efficiency
v44	**+0.55**	Expected disadv. of DEV: reduced performance
v49	**+0.86**	Expected disadv. of DEV: reduced effectiveness
v68	**+0.66**	Expected disadv. of TST: more expensive
v69	**+0.65**	Expected disadv. of TST: reduced performance
v72	**+0.52**	Expected disadv. of TST: increased communication
v73	**+0.66**	Expected disadv. of TST: reduced efficiency
v74	**+0.52**	Expected disadv. of TST: reduced effectiveness
v158	-0.63	Maturity models are easy to tailor to company
v68: Expected disadvantages of using a test process (TST): more expensive		
v16	-0.52	DEV reduces project cost
v17	-0.52	DEV increases product quality
v40	-0.58	Expected adv. of DEV: increased efficiency
v43	**+0.75**	Expected disadv. of DEV: more expensive
v44	**+0.64**	Expected disadv. of DEV: reduced performance
v47	**+0.53**	Expected disadv. of DEV: reduced flexibility
v48	**+0.66**	Expected disadv. of DEV: reduced efficiency
v69	**+0.74**	Expected disadv. of TST: reduced performance
v72	**+0.51**	Expected disadv. of TST: reduced flexibility
v73	**+0.59**	Expected disadv. of TST: reduced efficiency
v158	-0.74	Maturity models are easy to tailor to company
v159	**+0.60**	Known maturity models are not easy to tailor

cost (TST) are positively correlated with several other expected disadvantages, and improvements regarding product quality or cost reduction are negatively correlated with v48 and v68. Furthermore, expected negative impacts of using development processes also relates to expected disadvantages of using test processes and vice versa.

Although we are not able to distill accurate patterns from our data, we see tendencies in selected clusters. The exemplary selection in Table XVI, for instance, suggests a general negative attitude toward (standardized or company-wide) processes as respondents seem to perceive those processes as a limitation of freedom generating extra cost and reducing performance.

The results of the correlation analysis provide a valuable input for the adaptation of our instrument in the future. However, we are aware that the kind of pair-wise correlations (also given our data) does not yet allow for their reliable usage for infering a theory whereby we rely for this theory on the results of our descriptive analysis only (see also the next section). The results from our correlation analysis is disclosed with our raw data[1].

V. WORKING THEORY FOR FUTURE REPLICATIONS

One major goal of this work is to synthesise previously conducted surveys into one project to be replicated over the next years until having a certain saturation in our underlying theory. The first step in this endeavour thereby consisted in the survey round described in this paper, having more a curiosity-driven character. Based on our results, we generate a working theory (of expectations) which we use to calibrate our instruments over the next replications in dependency on the extent to which we will be able to support or reject our expectations.

We use the term *theory* in the sense of a social theory according to Popper [29] and refer to a set of falsifiable and testable hypotheses. Those hypotheses are based on the results described in the previous sections and, due to the currently still low number of respondents, reasoning by argument. We design our initial theory by formulating 4 hypotheses:

- H_1: There is no standard among software processes used within and among different software companies.
- H_2: Software companies opt for integration of processes and tools, but do not yet master it.
- H_3: Software companies do not have established a systematic software process improvement program.
- H_4: Software companies reject a software process improvement program that is build on the principle of standardization.

The hypotheses are based on the results from our descriptive statistics that shows certain tendencies. Those include (not limited to) that the field of software processes used is extremely diverse with no candidate that would indicate to a preferred standard. Our results indicate that software development is characterized by individualism overruling the possibilities and the desires for standardization. This could also be a reason why SPI programs as such are not or—if at all—reluctantly implemented by the respondents to our survey, and that participating companies do not yet have integrated company-wide approaches. This holds at least for those programs that rely on standardized norms.

Concluding, we believe that the overall tone of perception in our field can be best described by an adaptation of the IDIC principle "Software development is still driven by **I**nfinite **D**iversity in **I**nfinite **C**ombinations."

VI. CONCLUSION

Our overall goal is to develop an instrument to continuously monitor the state of practice in global software development to investigate the degree of professionalism and systematization of software development. We focus on the perceived importance of the two general software engineering process areas project- and quality management and their implementation in practice to draw a big picture of strengths and weaknesses to, eventually, derive concrete improvement proposals.

To this end, we developed a questionnaire which was to a certain extent grounded in previously conducted isolated studies. We conducted our initial survey in context of the German software industry. The paper at hand presents the results from this resulting *3ProcSurvey*, which served as public test run of the questionnaire to collect initial data, to create a baseline, and to lay the foundation for future research by serving the development of working hypotheses.

So far, our results indicate that the necessity of a systematic development approach seems well recognized. Project management, development, and test processes are considered important for project success, and the respondents see considerable improvement potential for these processes. Furthermore, the results indicate that companies still seem to use a "method zoo" comprising different development and test processes, which are partially integrated with project management and test standards. About 12% of the companies state to use non-standardized company-specific development processes, and approx. 30% state to use non-standardized company-specific test approaches. Furthermore, about 30% of the respondents mention to not having an integration of development- and test processes with standardized project management approaches. Method selection, composition, and integration in the company's process ecosystem do often not follow standardized procedures; these activities are oftentimes left to the project managers who decide on a per-project basis (more than 60%). Finally, although our respondents mention huge improvement potentials, systematic improvement activities seem to be scarcely implemented. For instance, more than 60% of the respondents mention to manage lessons learned only sporadically or on an individual basis, and more than 50% mention that they do not collect SPI-related performance data at all.

In summary, our results obtained so far indicate that the importance and potential of using standardized processes is well recognized. However, as soon as projects are started, only immediate project-related problems seem to count and long-term perspectives and goals appear to become irrelevant. Furthermore, our findings point to a general reluctance toward any standardization attempt: standardized methods are considered costly, inappropriate for the actual context, and they are meant to hamper flexibility and performance. However, advantages that are meant to come with standardized and balanced methods seem to be neglected. Especially in the context of GSD, this needs to be considered a serious project risk as a diversity of used methods implies a mixture of concepts and terminology thus paving the way for misunderstanding and conflicts.

A. Impact & Implications

Our survey provided a first step in synthesizing previously conducted, more isolated studies. Based on our data, we could further infer a first set of falsifiable hypotheses which we use to calibrate our instruments. Our goal is to reach a point where our current theory patterns have the required saturation that allows us leaving the level of conventional wisdom that currently dominates the publication landscape in software engineering.

We expect our results to positively impact the possibilities of practitioners and researchers. Practitioners can already use our results to analyze their own current situation with respect to more general trends. As we disclosed our raw data and the analysis results, researchers can already build their own investigations based on our data, and they can analyze the relation to existing evidence in more detail. Disclosing our data and proposing a working theory allows us to plan replications necessary to reach our long-term goal.

B. Relation to Existing Evidence

As mentioned in Sect. II, a variety of survey-based research is available addressing manifold topics of software engineering. In the context of the present paper, some studies report findings, which we (although our findings need to be regarded tentative) can support: Vijayasarathy and Butler [5] study the use of software processes. Their study—mainly in the context of the United States—shows a variety of different development approaches used and combined with each other (also supported by, e.g., Komus et al. [18], Cusumano et al. [6], and Jones [21]). Furthermore, they analyze correlations and find, among others, that agile methods are preferably used in organizations with moderate revenues and small number of employees, and one-team-projects with small team sizes. Our study draws a similar picture in which we find a variety of software processes in use. However, we were not studying correlations between process use and, e.g., project size and risk, but add another perspective: Do companies explicitly opt for applying multiple development approaches and who makes the decisions? Our findings show that in the majority of companies' development processes are not standardized and the actual selection is performed by the project managers on a per-project basis.

Another aspect often investigated is the question for why companies do not implement SPI programs (e.g., Staples et al. [28]). In this context, among others, Méndez Fernández and Wagner [14] support partitioners' reluctance toward so-called normative improvement approaches (in particular in the area of requirements engineering). Our results draw a similar picture, as only 19% of the responding companies state to have implemented a continuous SPI program. Moreover, 36% of the responding companies manage their lessons learned on individual initiatives only (consistent with Komus et al. [18]: when deploying agile methods, 68% is based on individual employees[4]), and approx. 51% mention to not collect SPI-related performance data. That is, our study supports the generally perceived reluctance toward normative SPI approaches and indicates that improvement activities are driven more by chance than by strategy.

C. Limitations

Our study is an exploratory first survey run and, thus, it has limitations: The questionnaire is the result of a merge of different input material (e.g., [13], [26], and the Iqnite study), which was enriched by further questions. Furthermore, the presented data results from the survey initially conducted in Germany, which aimed at testing the developed questionnaire and at creating a reference data set to develop working hypotheses and, eventually, a theory. Therefore, we still are far from being able to generalize. Yet, we could provide a first snapshot and show candidate variables to be used to develop proper hypotheses. Furthermore, at the current stage, the result set is rather small and diverse with varying n, and it remains unclear if, for instance, the study population is representative, which limits the result set's testability for, e.g., significance right from the beginning. The study at hand is therefore the first

[4]Komus et al. [18] do not provide information whether the activities around the deployment of agile methods follow a formal SPI program. However, their results show that the majority of companies deploys agile methods on a per-project basis and driven by individual employees. It remains unclear if this is a more curiosity-driven approach or a planned deployment strategy as proposed by, e.g., Münch et al. [30].

step toward an instrument for continuous monitoring, which, however, requires independent replication.

D. Future Work

The present study is the first step and, thus, future work requires the improvement of the instrument (based on the initial findings) and its independent replication. In order to prepare the replication, the questionnaire as such is currently under revision to reflect the learnings from the first run on, e.g., improvement of questions and questionnaire structure, development of working hypotheses, and further alignment of questions and hypotheses. Furthermore, partner networks are in the formation process to allow for international replication, which is a necessary step to enhance the data set and to allow for an in-depth analysis.

As the questionnaire is based on previously conducted studies, a further activity is the (initial) analysis of trends. For instance, in [31], we initially studied the process-related parts of the IOSE-W^2 result set and our data to investigate the use of software processes over time (6-year period), and found a trend toward using agile methods and, notably, their integration with traditional processes; at the same time, the Rational Unified Process almost extincted.

Since we aim at providing such trend analyses, we continuously adapt our instrument presented and disclosed with the paper at hands and cordially invite researchers and practitioners to participate in future replications.

Acknowledgement

We want to thank Annette Kossmann and Frank Simon for their collaboration on conducting the first survey round. We further thank BITKOM and GI for opening their distribution channels for this project, and Antonio Vetrò and Oliver Linssen for fruitful discussions during the analysis.

References

[1] R. Sangwan, N. Mullick, and D. J. Paulish, *Global Software Development Handbook (Applied Software Engineering)*. Auerbach Publishers Inc., 2006.

[2] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "Distance, dependencies, and delay in a global collaboration," in *Conference on Computer Supported Cooperative Work*. ACM Press, 2000, pp. 319–328.

[3] M. Nordio, H. C. Estler, B. Meyer, J. Tschannen, C. Ghezzi, and E. D. Nitto, "How do distribution and time zones affect software development? a case study on communication," in *International Conference on Global Software Engineering*. IEEE, 2011, pp. 176–184.

[4] D. Smite and C. Wohlin, "A whisper of evidence in global software engineering," *IEEE Software*, vol. 28, no. 4, pp. 15–18, 2011.

[5] L. Vijayasarathy and C. Butler, "Choice of software development methodologies - do project, team and organizational characteristics matter?" *IEEE Software*, no. 99, p. 1ff., 2015.

[6] M. Cusumano, A. MacCormack, C. F. Kemmer, and B. Crandall, "Software development worldwide: The state of the practice," *IEEE Software*, vol. 20, no. 6, pp. 28–34, 2003.

[7] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *International Requirements Engineering Conference*. IEEE, 2013, pp. 125–134.

[8] T. Roehm, R. Tiarks, R. Koschke, and W. Maalej, "How do professional developers comprehend software?" in *International Conference on Software Engineering*. IEEE, 2012, pp. 255–265.

[9] M. Niazi, D. Wilson, and D. Zowghi, "Critical success factors for software process improvement implementation: an empirical study," *Software Process: Improvement and Practice*, vol. 11, no. 2, pp. 193–211, 2006.

[10] The Standish Group. (1994 ff.) Chaos reports. [Online]. Available: http://www.standishgroup.com/

[11] J. Eveleens and C. Verhoef, "The rise and fall of the chaos report figures," *IEEE Software*, vol. 27, no. 1, pp. 30–36, 2010.

[12] R. Buschermöhle, H. Eekhoff, and B. Josko, *SUCCess and failurE of hard- and Software projectS (SUCCESS)*. BIS-Verlag der Carl von Ossietzky Universität Oldenburg, 2006.

[13] M. Fritzsche and P. Keil, "Kategorisierung etablierter vorgehensmodelle und ihre verbreitung in der deutschen software-industrie," Technische Universität München, Research Report (in German) TUM-I0717, 2007.

[14] D. M. Fernández and S. Wagner, "Naming the pain in requirements engineering: A design for a global family of surveys and first results from germany," *Information and Software Technology*, vol. 57, pp. 616–643, 2015.

[15] A. Gorla, I. Tavecchia, F. Gross, and A. Zeller, "Checking app behavior against app descriptions," in *International Conference on Software Engineering*. ACM Press, 2014, pp. 292–302.

[16] S. Khan, M. Niazi, and R. Ahmad, "Critical success factors for offshore software development outsourcing vendors: A systematic literature review," in *International Conference on Global Software Engineering*, 2009, pp. 207–216.

[17] M. Sulayman, C. Urquhart, E. Mendes, and S. Seidel, "Software process improvement success factors for small and medium web companies: A qualitative study," *Information and Software Technology*, vol. 54, no. 5, pp. 479–500, 2012.

[18] A. Komus, M. Kuberg, C. Atinc, L. Franner, F. Friedrich, T. Lang, A. Makarova, D. Reimer, and J. Pabst. (2014) Status quo agile 2014. [Online]. Available: http://www.status-quo-agile.net/

[19] F. Simon, A. Kossmann, M. Kuhrmann, and D. M. Fernández, "Wunsch oder Wirklichkeit? Professionelle Softwareentwicklung Made in Germany," *OBJEKTspektrum (in German)*, no. 1/2014, pp. 16–23, 2013.

[20] R. L. Glass, "The state of the practice of software engineering," *IEEE Software*, vol. 20, no. 6, pp. 20–21, 2003.

[21] C. Jones, "Variations in software development practices," *IEEE Software*, vol. 20, no. 6, pp. 22–27, 2003.

[22] M. Mah, "How agile projects measure up, and what this means to you," Cutter Consortium, Tech. Rep., 2008.

[23] M. Huo, J. Verner, L. Zhu, and M. Babar, "Software quality and agile methods," in *International Computer Software and Applications Conference*, 2004, pp. 520–525.

[24] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, and T. Kahkonen, "Agile software development in large organizations," *IEEE Computer*, vol. 37, no. 12, pp. 26–34, 2004.

[25] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol. 50, no. 9–10, pp. 833 – 859, 2008.

[26] M. Kuhrmann, C. Lange, and A. Schnackenburg, "A survey on the application of the V-Modell XT in german government agencies," in *Conference on European System & Software Process Improvement and Innovation*, ser. Communications in Computer and Information Science, vol. 172. Springer Berlin Heidelberg, 2011, pp. 49–60.

[27] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, 1988.

[28] M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, "An exploratory study of why organizations do not adopt cmmi," *Journal of Systems and Software*, vol. 80, no. 6, pp. 883–895, 2007.

[29] K. R. Popper, *The Logic of Scientific Discovery*. Taylor & Francis Ltd., 2004 (Reprint 2002).

[30] Jürgen Münch, Ove Armbrust, Martin Kowalczyk, and Martin Soto, *Software Process Definition and Management*. Springer, 2012.

[31] M. Kuhrmann and O. Linssen, "Welche Vorgehensmodelle nutzt Deutschland?" in *PMV 2014*, ser. Lecture Notes in Informatics, vol. P-236. Gesellschaft für Informatik (GI) e.V. (in German), 2014, pp. 17–32.