# A Large-Scale Technology Evaluation Study: Effects of Model-based Analysis and Testing

Michael Kläs, Thomas Bauer
Fraunhofer Institute for Experimental Software Engineering
Kaiserslautern, Germany
{michael.klaes, thomas.bauer}@iese.fraunhofer.de

Andreas Dereani
Daimler AG
Sindelfingen, Germany
andreas.dereani@daimler.com

Thomas Söderqvist
Volvo Group Trucks Technology
Advanced Technology & Research, Gothenburg, Sweden
thomas.soderqvist@volvo.com

Philipp Helle
Airbus Group Innovations
Hamburg, Germany
philipp.helle@airbus.com

***Abstract*—Besides model-based development, model-based quality assurance and the tighter integration of static and dynamic quality assurance activities are becoming increasingly relevant in the development of software-intensive systems. Thus, this paper reports on an empirical study aimed at investigating the promises regarding quality improvements and cost savings. The evaluation comprises data from 13 industry case studies conducted during a three-year large-scale research project in the transportation domain (automotive, avionics, rail system). During the evaluation, we identified major goals and strategies associated with (integrated) model-based analysis and testing and evaluated the improvements achieved. The aggregated results indicate an average cost reduction of between 29% and 34% for verification and validation and of between 22% and 32% for defect removal. Compared with these cost savings, improvements regarding test coverage (~8%), number of remaining defects (~13%), and time to market (~8%) appear less noticeable.**

***Index Terms*—Empirical study, embedded software quality assurance, multiple case study, GQM+Strategies, quantitative technology evaluation, model-based testing, internal baselines.**

## I. Introduction

The effectiveness and efficiency of *quality assurance* (QA) processes play a crucial role in the development of software and software-intensive systems. They impact not only the quality of the final product, but also the development and maintenance costs as well as the time to market [35].

Especially in the transportation domain a trend towards model-based development and QA can be observed [7][25]. Innovative model-based analysis and testing methods and tools have been developed and QA patterns have been proposed that describe how to effectively combine these techniques [36][34]. An important question for companies is therefore how and to which extent they can benefit from introducing such innovative techniques into their QA processes.

One can obtain this knowledge by piloting promising new technologies in individual case studies in the planned development environment [21][33]. Still, such an investigation consumes significant resources for selecting and understanding the new technologies and for setting up and conducting a study to evaluate them. Thus, practitioners expect some advance information about whether a certain set of new technologies has shown to be applicable to the planned improvement goals and what magnitude of improvement can usually be expected [8].

However, such generalized overview data is currently largely missing for up-to-date integrated *model-based analysis and testing* (MBAT) technologies in the transportation domain, nor can it be acquired by means of a meta-analysis of existing industrial case studies due to the limited number of published studies [29] and the diverging improvement objectives and measures underlying these studies, which limits the possibility to aggregate individual results [20].

We address this gap by presenting an empirical study conducted in a large-scale research project with 38 partners, including 14 companies with individual use cases for MBAT technologies and thus the potential for such case studies.

The research followed a multiple case study approach [23], which allowed us to consolidate initial individual improvement goals, align the different evaluation endeavors of the companies, and in a final step, reasonably aggregate the reported improvement results. In brief, this paper provides answers to the following three research questions:

***RQ1:*** *What are the goals underlying the introduction of (integrated) model-based testing and analysis in the transportation domain and how are they planned to be achieved?*

***RQ2:*** *How can key goals be quantified and evaluated in a large-sale study making use of multiple case studies?*

***RQ3:*** *What are the improvements achieved by (integrated) model-based testing and analysis?*

Keeping these questions in mind, the remaining part of the paper is structured as follows: *Sec. II* gives an overview of existing studies on the effects of MBAT in industrial practice. *Sec. III* provides some background information on the applied innovative MBAT technologies and the study context in general. *Sec. IV* presents the consolidated goals and the strategies

for achieving them with MBAT technologies. *Sec. V* explains the approach for quantifying the improvement goals and assuring that the reported results can be reasonably combined. *Sec. VI* gives an overview of the individual case studies and provides three examples of the use of MBAT technology in specific case studies. *Sec. VII* presents the study results, which indicate notable improvements from applying the new MBAT technologies, and *Sec. VIII* discusses threats to validity. *Sec. IX* closes the paper with a summary of the key contributions and their applicability in practice.

## II. Related Work

In this work, we aim at assessing the effects of *MBAT technologies*, which *are concrete QA approaches comprising a set of model-based QA techniques and the corresponding tool chain in an integrated set-up*. They are derived from the MBAT methodology *implementing a defined combination pattern for analysis, verification, and testing considering dedicated formal artifacts* (called analysis and test models) that should enable more effective and highly automated QA [30].

QA approaches have a measurable impact on software quality, which has been investigated in various studies. Nevertheless, the number of empirical studies is rather small compared to the total number of articles published in this area [29].

One limitation we see in existing work on the empirical evaluation of QA techniques, and which we address, is the artificial context of the evaluation. Most studies are conducted as controlled experiments in academic environments with small software applications. This conclusion is underlined by the reviews by Briand [6], Neto et al. [29], and Juristo et al. [16].

Neto et al. [29] state that only 5% of the 85 relevant publications related to *model-based testing* (MBT) report industrial evaluations, including subjective experience, which were not collected systematically. Briand highlights the challenges by measuring the effectiveness of QA by introducing faults into the system application that is being checked [6]. Furthermore, the size of the software application is usually too small to show its industrial applicability. Neto et al. reported application sizes of less than 7.000 lines of code [29].

The scope of the empirical studies is also usually restricted to one or a few QA techniques that are applied in isolation at a defined process stage. In their famous study, Juristo et al. [17], for example, compared the effectiveness of functional testing, structural testing, and code inspection considering relevant context factors. One conclusion was that equivalence partitioning and branch coverage are equally effective.

However, the combination of techniques as investigated in this paper, and as already highlighted by Briand in 2007 as a future research topic [6], is covered by a very limited number of empirical studies. A recent example is provided by Elberzhager et al. [12], who investigated the systematic combination of inspection and functional testing. Their case study results indicate up to 34% test effort reduction compared to the unfocused and unsystematic application of the two techniques.

A work strongly related to ours in providing an industry-focused high-level view on the impact of model-based QA techniques is presented by Binder, who conducted a large-scale survey on the effects of MBT [5]. This was the starting point for a first holistic assessment of MBT techniques, considering various test modeling notations, test case selection techniques, tools, and application domains. The results showed that, on average, MBT reduced the number of slipped defects by around 60%, manual test effort by 15% and test time by 30%. Nevertheless, problems were also reported, such as the growing complexity of test models and updating of test models, which were mentioned by 15%, respectively 11%, of the respondents.

Unlike Binder, our work does not use the retrospective view of a survey but applies a multiple case study design to measure and evaluate the observed improvement. Moreover, we consider not only MBT, but also model-based analysis.

## III. Study Background

Our large-scale empirical study was conducted in the European research project MBAT, which stands for combined model-based analysis and testing [27]. In the project, research partners, tool vendors, and industrial use case providers from 39 organizations and eight countries jointly investigated and developed QA techniques and the corresponding tool platforms for safety-related software-intensive systems from different transportation domains, i.e., automotive, avionics, and rail systems. A *use case* represents the context in which the technologies should be applied. It states the setting and the problems to be addressed and provides the opportunity to get quantitative feedback by conducting a corresponding case study. The addressed use cases covered different steps of the system and software QA processes as well as different quality properties, such as functional correctness, time behavior, and compliance with standards, such as ISO 26262 [15] for passenger cars and DO-178C [10] for airborne systems.

The MBAT project addressed two major topics: (1) the improvement of the QA approaches regarding their effectiveness and efficiency through the systematic combination of model-based QA techniques and (2) the integration and interoperability of the corresponding tools on the technical side.

On the conceptual side, the MBAT methodology was developed as a set of QA approaches that combine different model-based analysis, verification, and test techniques [30]. According to a bottom-up strategy, use-case-specific solutions were generalized towards a set of best practices for the model-based QA of software-intensive technical systems. A set of patterns for the usage and instantiation of the combined QA approaches was developed, representing appropriate solutions for defined application domains, quality properties, and process stages [14]. One concrete instantiation is the integrated quality assurance framework (InQA), which enables the stepwise optimization of QA planning and control, taking into account heterogeneous quality objectives such as the coverage of requirements and architectural models and the detection of the most critical and frequent defects [11].

To solve technical challenges, a project-specific *reference technology platform* (MBAT RTP) was developed, which implements new, combined QA approaches by integrating the corresponding analysis, verification, and testing tools with requirements management systems, bug trackers, and architec-

tural modeling tools [4]. MBAT RTP uses and extends the *Open Services for Lifecycle Collaboration platform* (OSLC) [32], which provides a generic infrastructure for creating tool chains for software lifecycle activities with exchangeable tool components. For each use case, a tailored version of MBAT RTP was developed, such as the integration of safety analysis, software design, and simulation tools by Kacimi et al. [18].

In order to get quantitative evaluation results for the improvements that can be achieved by using MBAT technologies, in the first stage the different goals and strategies of the use case providers were consolidated in a common framework and abstract measures were defined. In each case study, MBAT-based solutions were implemented to achieve a certain subset of goals relevant for the respective use case, and individual data collection procedures were defined. Finally, the data collected in the case studies were aggregated to obtain more general statements about the achievement of the consolidated set of improvement goals.

## IV. Improvement Goals and Strategies

This section first provides an overview of the approach for the identification and consolidation of the goals and strategies of the various use case providers and then presents the resulting *Goals+Strategies graphs* for all five high-level MBAT goals.

### A. Approach – Identify and Consolidate Individual Goals

In order to allow aggregating the results of the different case studies later on, the first task was to get a joint understanding of the relevant improvement goals and strategies. This does not necessarily mean that all case studies had to have the same improvement goals. Rather, each study defined its own goals and strategies in a common taxonomy.

The *GQM⁺Strategies method* [2] is one way of describing such a taxonomy. It allows defining business and improvement goals as well as contributing strategies on different levels of an organization, and modeling their relationships.

We use the key concepts of this approach, which is usually applied to align goals and strategies in a single company [3], in a large-scale research project as initially motivated and illustrated in a previous paper [23]:

1) We identified five high-level goals based on the project proposal. Moreover, we extracted possible sub-goals and strategies in order to achieve them, provided they are named (even implicitly) in the document. We considered this as a good starting point because all use case providers cooperated in the creation of the project proposal, meaning that all mentioned goals represent at least a first common view of the partners.

2) Based on the extracted goals and strategies, a measurement expert collaborated with MBAT experts on developing an initial Goals+Strategies graph for each high-level goal. An advantage of using Goals+Strategies graphs is that they indicate gaps and inconsistencies in the relationships between the modeled elements (e.g., goals without strategies for achieving them or sub-goals not contributing to a high-level goal). Using this information, the team complemented missing goals and strategies and consolidated overlapping goals and strategies to complete an initial version of the Goals+Strategies graphs.

TABLE I. Consolidated MBAT Goals and Strategies

| *ID* | *Goal/Strategy Description* |
|---|---|
| **G1** | **Reduce overall V&V costs** |
| S1 | Select and apply appropriate MBAT technologies in projects |
| G1.1[L] | Reduce costs for checking boundary values |
| S1.1a | Replace boundary value testing at software and/or functional level (partially) with (cheaper) boundary analysis |
| S1.1b | Introduce a more cost-efficient boundary values analysis approach (e.g., with improved tool support and interoperability between tools) |
| G1.2 | Reduce costs for T&A model development |
| S1.2a | Use T&A models that can be reused for subsequent V&V activities |
| S1.2b | Use T&A models that can be (partially) reused in several projects |
| S1.2c | Use efficient support for T&A model development and maintenance (MBAT guidelines, tools, etc.) |
| G1.3 | Reduce costs for testing |
| S1.3 | Automatically generate test cases from T&A models and static analysis results (Once such a model is available, the costs for generating V&V cases are minimized.) |
| G1.4 | Reduce costs for avoidable V&V (inefficient combination of T&A) |
| S1.4a | Smart combination of T&A (e.g., if a technique has already discovered a fault, other techniques do not have to target this fault anymore.) |
| S1.4b[L] | Avoid redundant parts of V&V activities (e.g., detect redundant test cases) |
| S1.4c | Replace parts of V&V activities with more cost-efficient V&V activities (e.g. replace certain test activities with static analysis or vice versa) |
| G1.5 | Reduce V&V costs for product variants |
| S1.5 | Re-enforce a product lines concept with support for efficient V&V of variants (e.g., by reusing V&V efforts across variants and optimizing its distribution based on variant information, as well as V&V automation). |
| G1.6[L] | Reduce costs for checking that generated code matches implementation model |
| S1.6 | Demonstrate that automatically generated code matches implementation model (e.g., SCADE code generator guarantees this compliance certified SIL3/4) |
| G1.7 | Reduce costs for static analysis |
| S1.7 | Introduce a more cost-efficient static analysis approach (e.g., one that reduces false warnings from analysis tools due to hints from testing or narrows the search space through additional information) |
| G1.8[L] | Reduce V&V costs resulting from changes in requirements |
| S1.8a | Use impact analysis to identify affected V&V artifacts |
| S1.8b | Make iterative static analysis more cost-efficient (e.g., by proving the identity of previously identified issues) |
| **G2** | **Reduce overall defect costs** |
| S2.1 | Find defects earlier during development through model-based T&A technology |
| G2.1.1 | Reduce defects slipped through construction stage |
| S2.1.1 | Increase usage of formal analysis and simulation methods |
| G2.1.2 | Increase effectiveness of MBAT-improved V&V after construction |
| S2.1.2a | Apply virtual integration testing |
| S2.1.2b | Apply new model-based testing to improve quality of tests |
| S2.1.2c[L] | Improve effectiveness of test model (e.g., by ensuring its correct generation from requirements by traceability analysis) |
| G2.1.3 | Reduce rework resulting from incomplete, instable, and erroneous requirements |
| S2.1.3 | Improve robustness of requirements through early validation |
| S2.2 | Improve fault location capabilities by applying MBAT technology |
| G2.2.1 | Reduce costs for defects found during construction stage |
| S2.2.1 | Improve fault location capabilities of analysis methods (e.g., support for detailed analysis of wrong behavior and failure situation) |
| G2.2.2 | Reduce costs for defects found during MBAT-improved V&V after construction |
| S2.2.2 | Improve fault location capabilities through model-based T&A technology (e.g., use impact analysis and provide traceability to identify affected construction artifacts such as requirements and code) |
| G2.2.3[L] | Reduce costs for defects found after MBAT-improved V&V |
| S2.2.2 | *see previous definition* |

| *ID* | *Goal/Strategy Description* |
|---|---|
| **G3** | **Reduce total cost of ownership for development platform** |
| S3 | Make use of MBAT RTP and tools |
| G3.1 | Reduce tool license and support costs |
| S3.1 | Simplify integration and communication between tools through RTP (e.g., by the means of shared information about environment, scheduling, compile process, shared assumptions and verification results, and defined extension points) |
| G3.2 | Reduce costs for platform certification |
| S3.2a | Simplify certification through standardized solutions and RTP interfaces |
| S3.2b[L] | Support incremental certification by using pre-certified components |
| G3.3[L] | Reduce costs for training |
| S3.3 | Reduce necessary trainings through standardized solutions and RTP interfaces |
| **G4** | **Provide high product quality in the face of increased complexity** |
| S4 | Select and apply appropriate MBAT technologies |
| G4.1 | Increase coverage of test criteria |
| S4.1a | Apply model-based V&V technologies (V&V cases can be generated, executed, and evaluated automatically, and test coverage can be assured) |
| S4.1b[L] | Use requirements-based coverage information to complete test case generation |
| S4.1c[L] | Integrate analysis and test tools so that they can exchange their findings to improve analysis results |
| G4.2 | Reduce number of defects detected after MBAT-improved V&V |
| S4.1a | *see previous definition* |
| S4.2 | Apply effective combination of A&T (e.g., techniques are giving each other hints on where to find defects) |
| G4.3 | Use technology that is applicable / scales for systems with increased complexity |
| S4.3 | Use T&A models to abstract from system details (leads to reduced complexity for each abstraction level) |
| **G5** | **Reduce time to market for embedded systems products** |
| S5.1 | Provide higher automation of the analysis & test process |
| G5.1 | Reduce time needed for V&V after construction |
| S5.1a | Apply efficient combination of T&A technologies (e.g., techniques are giving each other hints where to find defects and tools exchange data about findings) |
| S4.1a,b[L],c[L] | *see previous definitions* |
| S5.2 | Strongly foster an early start and high quality of V&V activities |
| G5.2[L] | Reduce time for localize and correct defects after MBAT improved V&V |
| S2.1, S2.2.3 | *see previous definitions* |
| G5.3[L] | Ensure test coverage for well-defined subclasses of quality criteria |
| S5.3.1 | Employ formal analysis methods |

V&V: Verification and validation; T&A test and analysis; RTP: Reference technology platform; [L]: Goal or strategy with relevance only for a limited number of case studies

For each identified goal, an abstract measure was also defined to give an idea of how this goal could be quantified.

3) A survey involving all use case providers (n=14) was conducted with a response rate of ~80%. The survey contained the initially consolidated Goals+Strategies graphs including abstract measures, giving the use case providers the option to rate the goals and strategies regarding their importance for their use case. Moreover, they could extend the graphs with additional goals and strategies and make remarks about existing elements.

4) The survey analysis allowed us to identify 21 elements, i.e., goals or strategies, that were deemed relevant only by very few or none of the use case providers and 16 new elements that had to be added to the initial graphs. Moreover, we reformulated ambiguous phrases in the description of various goals and strategies. The survey analysis also revealed that, on average, each use case provider considered half of the 28 consolidated goals as relevant in their context.

### *B. RQ1 Result – Consolidated Goals+Strategies Graphs*

Table I provides an overview of the five high-level goals (G1 to G5) with their corresponding refinement into sub-goals and strategies. The ordering and the numbering schema indicate their relationships. Goals (shaded) and strategies (white) that have only limited relevance for the use case providers are marked with [L]. The assumption underlying the graph is that each sub-goal contributes to the achievement of its high-level goal, e.g., *reducing costs for test and analysis model development* (G1.2) helps to *reduce overall verification and validation costs* (G1). The sub-goal itself can be achieved in different ways. For example, models can be reused by different QA activities (S1.2a) and in further projects (S1.2b), or model development and maintenance can be better supported by tools, guidelines, etc. (S1.3c). Some strategies apply for several goals in the graph, in which case they are referenced by their ID.

## V. Quantification and Evaluation Approach

The previously consolidated set of improvement goals and abstract measures provides a frame for all case studies. In order to conduct the individual case studies, the abstract measures of the improvement goals that are relevant in the specific case study were operationalized in the first step. This means, in particular, that *individual measurement plans* were developed following the well-known Goal/Question/Metric (GQM) paradigm [1]. A measurement plan structures and refines an improvement goal, concretizes it via questions, and finally quantifies it with measures, including corresponding measurement scales. For each measure that needs to be collected (*base measure*), it defines who, when, and how the data have to be gathered, and for each *derived measure*, it describes the calculation rules for deriving it based upon other measures. For illustration purposes, Figure 1 provides a condensed version of such a measurement plan.

In order to help the use case providers define appropriate measurement plans, they were supported by measurement experts with a two-day workshop on goal-oriented measurement, a moderated user group, individual support via email and phone, as well as a review of their final measurement plans.

It is important to note that the previously specified abstract measures with predefined measurement scales and units assure that, although improvement goals can have different operationalizations in different studies, the improvement results can be combined on a level that abstracts from case-study-specific measurement units and terminology. Moreover, each abstract measure is defined in a relative way, e.g., *% reduction of costs caused by defects detected after construction stage*. This allows the use case providers to keep sensitive cost and defect numbers internal and report on relative improvements.

However, in order to measure such relative improvements, an individual *baseline* has to be defined for each case study. A baseline represents the state before the new MBAT technology is applied. Baselines can be defined based on data collected in earlier projects, a second project running in parallel, a previous release or sprint of the case study project, or by splitting the investigated system into two parts: one where the current technology is applied and one where the new MBAT technology is

| **Improvement Goal** | G1.3: Reduce cost for testing |
|---|---|
| **Applied Strategies** | S1.3: Automatically generate test cases from T&A models and static analysis results |
| **Measurement Goal** | Analyze the *test processs* (object) for the purpose of *evaluation* regarding cost (quality focus) from the viewpoint of a *project manager* in the context of *Use Case -xxx -in the automative domain* (context). |
| **Reported Improvement** | % Decrease of average test effort per requirement with MBAT technology |
| **Baseline** | Measurement data from project -xxx- |

Q1 Which level of improvement was obtained with respect to the quality in focus?

| ID | Derived Measure | Calculation | Unit | Time | Collector |
|---|---|---|---|---|---|
| %DecA_ TEpReq | % Decrease of average test effort per requirement with MBAT technology | = (1-A_TEpReq_M /A_TEpReq_B) *100% | % (rational) | Milestones M24 & M33 | Use case provider |
| A_TEpReq_ M | Average test effort per requirement with MBAT technology | = TE_M/ #Req_M | PH (rational) | After case study with MBAT | Use case provider |
| A_TEpReq_ B | Average test effort per requirement with baseline technology | = TE_B/ #Req_B | PH (rational) | After baseline case study | Use case provider |
| TE_M | Test effort with MBAT technology | = TE_TCD_M +TE_TE_M +TEx_Tev_M | PH (rational) | After case study with MBAT | Use case provider |
| TE_B | Test effort with baseline technology | = TE_TCD_B +TE_TE_B +TEx_Tev_B | PH (rational) | After baseline case study | Use case provider |

| ID | Base Measure | Collection | Unit | Time | Provider |
|---|---|---|---|---|---|
| TE_TCD_M | Test effort for test case design with MBAT technology | Effort Sheet | PH (rational) | After test case design | Project manager |
| TE_Tex_M | Test effort for test execution with MBAT technology | Effort Sheet | PH (rational) | After test execution | Project manager |
| TE_Tev_M | Test effort for test evaluation with MBAT technology | Effort Sheet | PH (rational) | After test evaluation | Project manager |
| #Req_M | #Requirements with MBAT technology | Extract from DOORS/MKS | (integer) | Before test case design | Use case provider |
| TE_TCD_B | Test effort for test case design with baseline technology | Effort Sheet | PH (rational) | After test case design | Project manager |
| TE_Tex_B | Test effort for test execution with baseline technology | Effort Sheet | PH (rational) | After test execution | Project manager |
| TE_Tev_B | Test effort for test evaluation with baseline technology | Effort Sheet | PH (rational) | After test evaluation | Project manager |
| #Req_B | #Requirements with baseline technology | Extract from DOORS/MKS | (integer) | Before test case design | Use case provider |

Q2 Which factors differentiate the baseline from the MBAT case?

| | | | | | |
|---|---|---|---|---|---|
| Skill_M | Skill level of tester for MBAT technology | Internal Questionnair | (ordinal) | After MBAT case study | Tester |
| Skill_B | Skill level of tester for baseline technology | Internal Questionnair | (ordinal) | After baseline case study | Tester |
| … | … | … | … | … | … |

Fig. 1. Condensed and anonymized example of a measurement plan.

applied. The advantages and limitations of the different ways for collecting baseline data are discussed by Kläs et al. [23].

The overall evaluation endeavor was planned for and conducted in two iterations. During the first evaluation round in 2013, the use case providers had the opportunity to get to know the evaluation approach, test their data collection procedures, and identify confounding factors that might limit the validity of their study results. Based on this experience, they planned the final evaluation round, which took place in 2014 and whose results are reported in this paper.

Because it became obvious that not each case study could collect data on all improvement goals considered relevant by the study provider, the option was given to evaluate improvement goals not only based on measurement data but also based on expert judgment. In order to distinguish these two kinds of data collection, two reporting forms were provided.

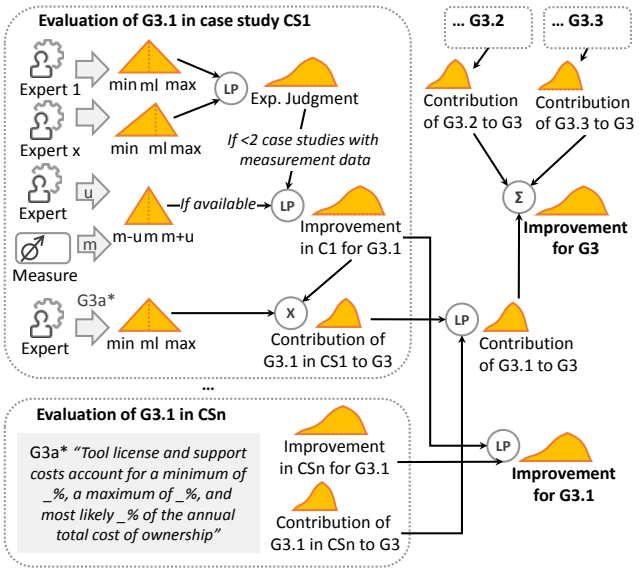


Fig. 2. Example of data aggregation for G3.1 as sub-goal of G3.

The *M Form* allows reporting for each improvement goal: data on the measured improvements (m), observed confounding factors, ratings on the quality of the collected data, and how well the underlying improvement strategy was implemented. It also allows a final judgment on the remaining uncertainty in the measurement results (u), e.g., ±10%.

The *E Form* allows reporting expert opinion on the observed improvement for each goal that is considered relevant but is not measured in the study. Moreover, the form collects data required to determine the contributions of the sub-goals to the high-level goals. The uncertainty in the expert opinions is captured by asking for minimum (min), maximum (max), and most likely (ml) values. These values are then used to define a triangular probability distribution, which is a well-known technique in quantitative risk management [28].

This enables the combination of measurement and expert data via *linear pooling* (LP) [9] for sub-goals without or with sparse measurement results. Linear pooling is a common technique used to combine, e.g., expert estimates made with uncertainty [24], and, when applied without dedicated weights, can be considered as a kind of simple average between the underlying probability distributions. Figure 2 illustrates how the different pieces of information represented as probability distributions are combined using Monte Carlo simulation [22] as implemented in the Palisade @Risk add-on for Excel.

## VI. Case Study Overview

This section first gives a brief overview of all the cases studies and then describes in more detail the context and the concrete improvement strategies in three case studies, which can be considered to be representative of the remaining ones.

The 14 case studies for which improvement goals were operationalized via measurement plans cover the transportation domain, with several studies each for automotive, avionics, and rail systems. Moreover, they cover 12 different companies and a variety of domain-typical context factors and technologies. We are not allowed to characterize all individual case studies

TABLE II. MAPPING BETWEEN CASE STUDIES AND GOALS

| Anonym. Study ID | CS262 | CS371 | CS405 | CS546 | CS559 | CS562 | CS563 | CS612 | CS671 | CS808 | CS815 | CS823 | CS940 | CS964 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G1.1 | PE | M | - | - | - | - | - | P | - | P | - | - | - | M |
| G1.2 | PE | - | - | - | - | - | - | - | - | - | - | - | - | M |
| G1.3 | M | M | PE | M | M | ME | PE | ME | ME | P | M | M | PE | M |
| G1.4 | M | - | - | M | - | - | - | E | M | - | - | - | PE | ME |
| G1.5 | - | - | - | - | E | - | - | - | - | - | - | - | - | PE |
| G1.6 | - | - | - | PE | - | - | - | E | - | - | - | - | - | P |
| G1.7 | PE | P | ME | - | - | E | - | E | M | P | - | - | - | - |
| G1.8 | PE | M | - | - | M | - | - | - | - | - | PE | M | - | E |
| G2.1.1 | PE | M | M | - | E | - | PE | PE | - | P | M | - | - | PE |
| G2.1.2 | PE | - | - | PE | M | - | PE | - | - | P | E | E | - | PE |
| G2.1.3 | - | M | E | - | M | PE | - | PE | M | - | - | E | - | PE |
| G2.2.1 | - | - | E | - | - | PE | - | - | - | - | M | - | PE | E |
| G2.2.2 | - | E | - | - | - | - | - | - | - | - | PE | - | - | E |
| G2.2.3 | - | - | - | - | - | ME | - | - | - | - | PE | - | - | - |
| G3.1 | PE | - | - | PE | - | - | - | - | - | - | - | - | - | PE |
| G3.2 | PE | - | - | PE | E | - | - | - | - | P | - | - | - | - |
| G3.3 | PE | - | - | - | E | - | PE | - | - | - | - | - | - | - |
| G4.1 | - | - | PE | - | - | - | PE | ME | M | P | - | M | PE | PE |
| G4.2 | - | PE | - | - | - | - | PE | - | M | P | - | E | - | PE |
| G4.3 | - | PE | - | - | - | - | - | - | PE | P | - | - | - | - |
| G5.1 | PE | PE | - | - | - | - | - | E | - | P | PE | E | PE | PE |
| G5.2 | - | E | - | - | - | - | - | - | - | P | - | E | - | E |
| G5.3 | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

P: Measurement plan defined but not evaluated; PE: Measurement plan defined but only expert opinion; E: Expert opinion; M: Measurement data; ME: Measurement data and additional expert opinion

regarding these factors since this information is considered too sensitive and might allow attributing results to specific companies. Rather, we abstract from specific techniques and situations by reporting aggregated results, which are assumed to *characterize the improvements that are possible with the help of MBAT technologies in the transportation domain in general*. Thus, the overall baseline for the reported improvements is the status quo in the considered domain. We are aware that this necessary compromise inhibits variation analysis, which would usually be applied to explain differences between the cases.

### *A. Daimler – Turn Indicator*

The system in the Daimler turn indicator case study implements the functionality of an automotive exterior light controller. The system consists of about 14 electronic control units (ECUs) and realize indicator functionality as well as the passing, fog, and brake light. Communication between ECUs is realized using CAN or LIN bus systems. Thus, the exterior light control system is a distributed hard real-time system, featuring high complexity due to continuous data exchange between ECUs. The entire system requirements are written as textual requirements in IBM Rational DOORS. The implementation of the example device under test has been realized as a TargetLink model containing a total of 81 TargetLink blocks.

The MBAT technology applied here reuses V&V artifacts created during model-based development with Matlab/Simulink (S1.2a). The smart combination of artifacts from preliminary steps, especially static analysis activities, helps to improve the overall system quality [19]. Figure 3 depicts the workflow and the corresponding tools. First, a static code analysis is performed by using the AbsInt tools Astrée, WCET, and StackAnalyzer (S1.7), which was pre-configured with the results from the design model analysis (BTC EmbeddedTester), such as assertions regarding variable values (S1.1b). In the next step, the static analysis is executed and the results are processed by EmbeddedTester to find potential test cases of interest (S1.4a). In the last step, EmbeddedTester generates test cases by using the context of the model (S1.3). These test cases are executed against the device under test and the results are manually evaluated by an engineer. The different tools were coupled using the OSLC technology (S3.1).

Overall, the conducted case study showed the expected workflow optimization, as the C-code of the turn indicator can be statically analyzed in the EmbeddedTester tool. Due to the use of OSLC as a tool interoperability protocol, this approach can be easily extended to other development artifacts and tools.

The evaluation and measurement were planned in cooperation with the developers of the turn indicator systems. The required baseline was determined in cooperation with the actual system developers. The results showed that the reuse of artifacts from development and QA activities helped to reduce time and focus the verification of a device under test.

### *B. Volvo – Brake-by-Wire*

The brake-by-wire case study dealt with verifying the models and the code for a brake-by-wire system in a number of analysis and testing scenarios. The system is developed in-house at Volvo for use as a research object and has been employed in different research projects to study various aspects. The system contains five control units, one per wheel and one central unit. The models describing the system were developed using EAST-ADL (Architecture Description Language) for modeling the structure, and Simulink and TargetLink for modeling and coding the behavior in accordance with AUTOSAR specifications [31].

In the study, eleven analysis and testing scenarios were considered, each describing certain verification objectives and success criteria, such as the systematic combination of QA approaches (S1.4a) considering model checking (S2.1.1), test case generation (S1.3), test execution, and static code analysis (S1.7), or better support for creating more formal and consistent QA artifacts (S2.1.3) [26].

In order to implement the strategies, the applied technologies reuse V&V artifacts created in the model-based development steps. The system description models, which were created by using EAST-ADL, Matlab/Simulink and TargetLink tools, are translated into intermediate models that can be processed

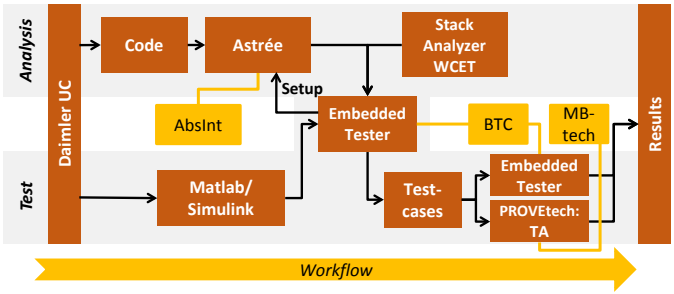


Fig. 3. Integrated analysis and test approach at the Daimler case study.

by the other QA tools used in the approach. Requirements are translated from natural language into formal notation using the OFFIS PatternEditor and are associated with model elements in the OFFIS RTP link tool to assure traceability and consistency. The impact of changes is analyzed with a dedicated impact analysis tool and affected elements and requirements are pointed out. The generated source code is statically analyzed using BTC Embedded Tester and AbsInt Astrée to find problems that are otherwise detected in later test phases. Models are checked and used to generate abstract test cases with the ViTAL tool from Mälardalen University, then handed over and further refined by the Enea Farkle tool, which has access to more concrete system-specific information. Interoperability of the tools in the chain was achieved by the OSLC technology via the MBAT RTP tool adapters.

The case study was conducted by applying the tool chains to the previously defined usage scenarios. The measurements were made by utilizing suitable sequences of the tool chain applications to measure absolute values of the various base measures. Based on the measurements results, it was then assessed which improvements were achieved in the QA processes with the applied MBAT technologies. As a baseline, a snapshot of the current development process of the brake-by-wire system was used. Since the brake-by-wire system is a small-scale project, the measurement results contain some uncertainties regarding their validity for large-scale serial production development projects. However, the results are promising and would in many cases likely be of benefit also in product development.

### *C. Airbus Group – EO/IR Sensor for UAV*

The system considered in the Airbus Group case study is an electro-optical/infrared (EO/IR) sensor for Unmanned Aerial Vehicles (UAVs) to enable wide-area ground and maritime surveillance. The EO/IR sensor system includes two main sensors, a daylight CCD camera and an IR sensor. EO/IR has been adopted by a wide range of UAVs as a payload. All system requirements are written and managed in IBM Rational DOORS. An executable SysML specification model has been implemented in IBM Rational Rhapsody. Sepp.med MBTSuite was selected as the tool for test case generation (S1.3) and IBM Rational Quality Manager for test case management. These engineering tools have been integrated into an OSLC-conformant tool chain by using the IBM Jazz-based technology platform (S3.1).

The main emphasis of the EO/IR sensor case study was on the integration of model-based testing and model-based analysis techniques into the system development process (S1.4a) [13]. Furthermore, ensuring traceability between related artifacts in all tools in a heterogeneous tool landscape was a major requirement (S2.1.2c). The main goal of the study was to demonstrate the feasibility of a model-based workflow for the development and testing of an aircraft system throughout the development lifecycle.

Measurement of the improvements provided by the MBAT technologies was done by evaluating expert interviews as well as through comparison with data collected in previous projects.

The general OSLC-based tool integration proved to be very valuable. Establishing links between engineering artifacts in different engineering tools becomes a necessity when it comes to managing the development and testing of more and more complex products. On the other hand, the work and effort needed to enable this proved that there is still work to do. It appears that a lot more effort must be spent on promoting the RTP idea in order to make OSLC-enabled tools a common sight in the commercial tool market.

The automatic test case generation using model-based testing techniques and tools proved to be a useful and sufficiently mature tool and will be deployed for the validation and verification of future products in the company.

## VII. Study Results

In the final evaluation round, we obtained data collected for 13 case studies (i.e., all except CS808) in which 13 of the 23 sub-goals were covered by measurement data and all except one goal by expert estimates. Table II summarizes which goals were evaluated based on which means in which case studies.

If at least two case studies provided measurement data on a specific sub-goal, then we only used the measurement data. If not, we also included expert opinions collected in other case studies to get a broader picture of the range of possible improvements in different environments.

Following the aggregation procedure described above, we calculated a probability distribution for each goal, which we used to determine the average improvement and the corresponding confidence interval on the confidence level CL = .90.

Figure 4 illustrates the results of the aggregation approach for sub-goal G2.1.3, where measurement data collected in three case studies indicate an average improvement of 32% with a confidence interval ranging from 20% to 44%. If the expert opinions from five other case studies are considered in addition, the expected average improvement would be only 24%, but the cone of uncertainty would be smaller, which means that there is a 93% chance that the average improvement is still within the 90% confidence interval of the measurement-based evaluation. This clearly motivates the advantage of characterizing the average improvement not only with a single value but also with an interval.

We also analyzed further information regarding data quality and the degree to which the planned improvement strategies were really implemented. These results show the expected correlation with the corresponding uncertainty estimates provided by the experts, but are not further discussed in this paper.

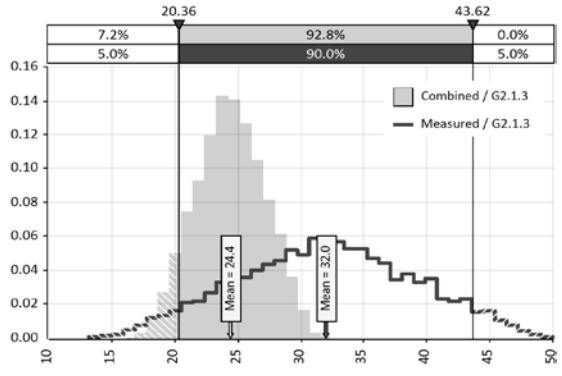

Fig. 4. Probability distributions resulting from data integration for G2.1.3.

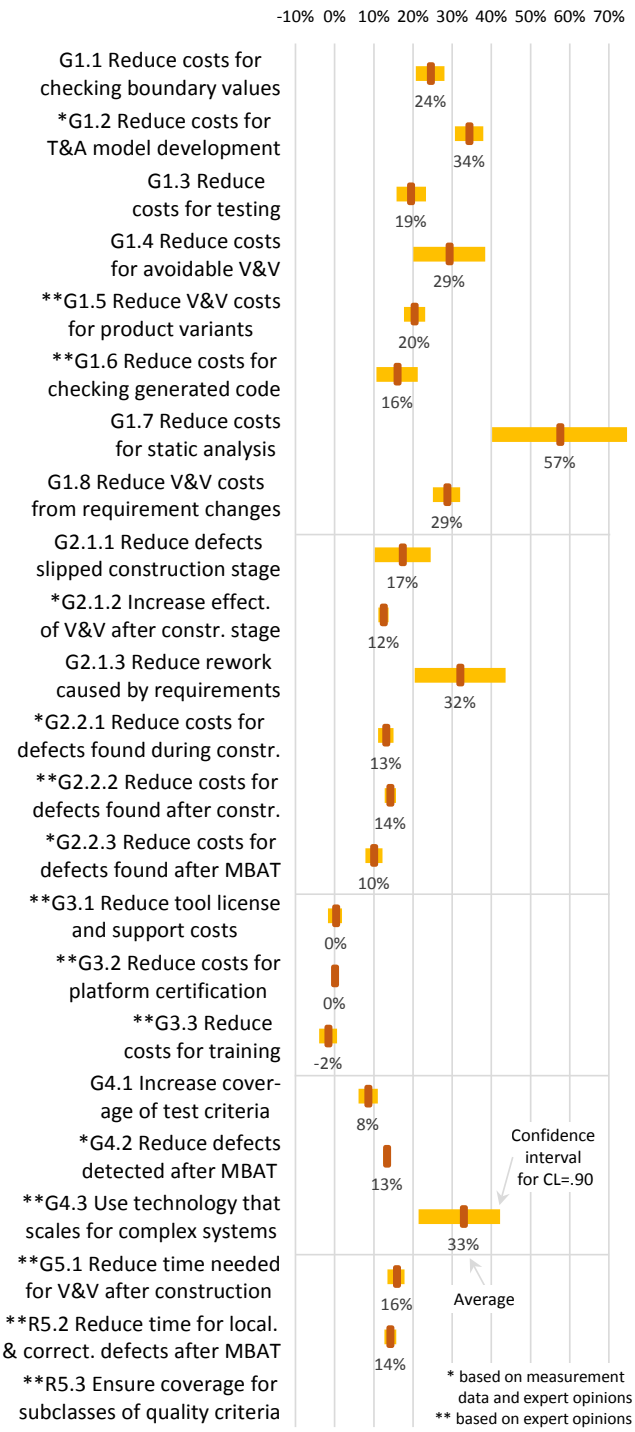


Fig. 5. Aggregated improvement results for sub-goals.

Figure 5 summarizes the aggregated improvement results for all sub-goals, indicating the kind of data used. The sub-goal results were then utilized to calculate the improvements that can be achieved for the high-level goals, which are summarized in Figure 6 and are discussed below:

*Verification and validation costs* (G1) have the potential of being reduced by an average of 32% considering the data collected in the 13 case studies. One sub-goal showing very high improvement potential is the reduction of static analysis costs (G1.7), but the results also indicate high uncertainty regarding the real average improvement caused by largely diverging results in the two underlying case studies. The sub-goal covered best by the case studies is the reduction of test costs (G1.3), which is most likely between 16% and 23% based on the measurement data collected in ten individual case studies.

*Defect costs* (G2) could be reduced by an average of 22% to 32% based on data collected in 13 case studies. The uncertainty is caused particularly by the circumstance that we cannot assume a fixed defect introduction and removal profile for all environments, but have to consider a range of possible profiles. A high degree of improvement could be observed for rework caused by incomplete, instable, and erroneous requirements, which is reduced by nearly one-third on average (G2.1.3), and for the number of defects that slip through the development activities and are found later on during testing (G2.1.1), which were reduced by an average of 17% based on measurement data collected in three case studies.

*Cost of ownership for the development platform* (G3) could not be reduced regarding the data collected in five case studies. Although there was major work on a reference platform for simplifying interconnectivity and data exchange between different technologies, positive effects on licenses, support, certification, or training costs could not be quantified yet. Measurable cost savings are rather expected for the time after the project's end, when the developed RTP and technologies are exploited further by partners and external organizations.

*Quality and scalability* (G4) as a high-level goal were not evaluated; instead, the results for the individual sub-goals were considered, which indicate that MBAT technologies can reduce the remaining defects by an average of 13% (G4.2) and – if we trust the expert opinions – also scales better for more complex systems (G4.3). Interestingly, the coverage regarding applied test criteria could only be increased by 8% (G4.1), which appears low at first glance. However, further information provided by the case studies showed that their existing approaches already provide very high coverage levels, which makes it difficult and sometimes impossible to increase them further.

*The time to market* (G5) could be reduced most likely by an average of approximately 8%. However, it should be noted that the seven case studies that contributed to the relevant sub-goals provided only expert opinions (cf. Fig. 5). Therefore, we would conclude that MBAT technologies probably reduce time to market, but not by a large amount.

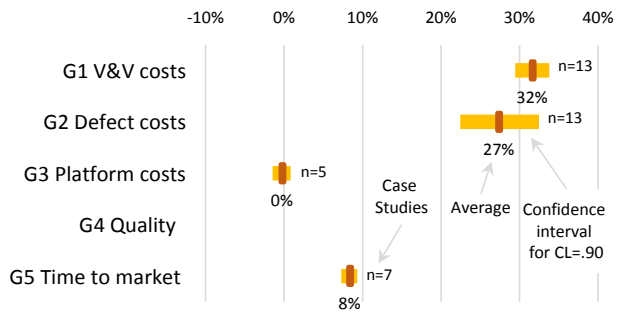


Fig. 6. Aggregated improvement results for high-level goals.

## VIII. Threats to Validity

In this multiple case study setting, we have to distinguish between threats caused by the overall evaluation design and threats that are specific for individual case studies.

An obvious threat caused by the study design is the fact that aggregated data was collected in the different case studies using individual measurement plans. However, we assured combinability by standardizing the measured concept via consolidated improvement goals and abstract measures, which were defined to be independent of the scales and units of the underlying operationalizations. For example, the reported relative reduction of test effort is independent of the unit used to measure effort and the concrete activities comprising testing in a specific company. Moreover, the reliability of data collection was supported by templates for measurement plans, individual guidance, and standardized forms for reporting results.

Another threat is that raw measurement results, such as absolute defect numbers, were not communicated outside the specific company due to confidentiality reasons and could therefore not be assessed directly with regard to validity. We tried to address this issue by including detailed questions on data quality in the reporting forms and providing the option to report an uncertainty range for the improvement results.

A potential bias that is difficult to deal with is the situation that all participants are interested in positive evaluation results to justify their participation in the research project. This may have led to the reporting of results that are too optimistic, especially in the case of pure expert estimates. However, considering the gathered data, we could find no indication for such a tendency. Rather, our cross-checks showed that the expert opinions were typically more pessimistic (85% of the cases) than the measurement data provided for the same goals.

Regarding external validity, it has to be stressed that all case studies were conducted in a realistic industrial context, including typical variations, in the transportation domain; thus, they are assumed to provide a valid summary for this context. However, it is not clear to what extent they can be transferred to other domains.

Because the case studies were organized and conducted by the individual use case providers, there are internal validity threats that are specific to certain case studies. During each evaluation round, we collected confounding factors observed in the case studies via open questions and clustered them (Figure 7). In order to better deal with the factors that were most frequently observed in the first evaluation round, specific countermeasures were proposed to the use case providers prior to the second round.

## IX. Conclusions

This paper provides a consolidated overview of the goals that companies try to achieve in the transportation domain when introducing model-based testing and analysis (MBAT) and of the strategies they intend to apply (*RQ1*). This may help other companies that are thinking about introducing MBAT technologies to formulate and structure their goals and cross-check their own strategies. Researchers may benefit from this structure because they can more easily communicate

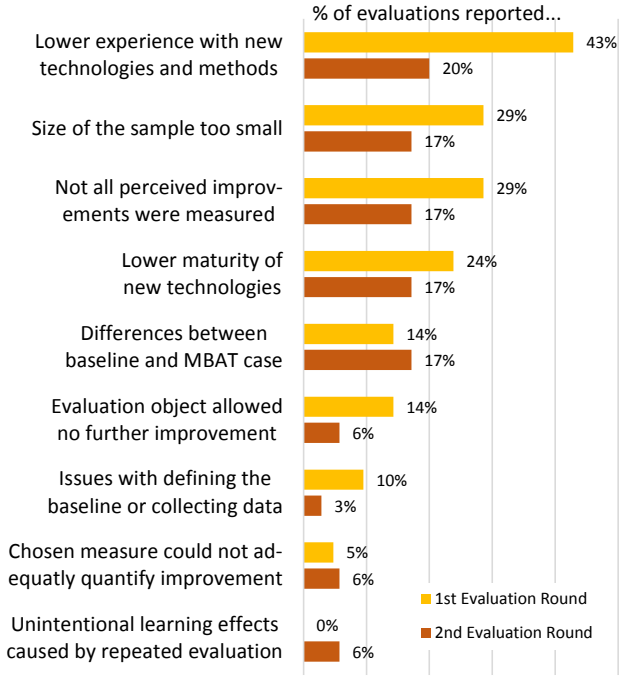


Fig. 7. Confounding factors observed in the case studies.

the contribution of their work by evaluating it regarding goals that are relevant for industry.

Moreover, this paper illustrates that it is possible to conduct quantitative technology evaluation in large-scale research projects using a multiple case study design (*RQ2*). To the best of the authors' knowledge, there are no other studies yet that report on the successful application of such rigorous quantitative evaluation via multiple case studies in a software engineering research project of comparable scale.

The reported improvement results including the calculated confidence bounds (*RQ3*) together with other information sources may help managers to make a decision for or against a more thorough investigation regarding the introduction of MBAT technologies in their development projects. Researchers working in the context of MBAT may benefit especially from the overview of the most commonly observed confounding factors, which they should be aware of when planning their own studies in industry.

While this paper provides a general overview of improvements that are possible with MBAT technologies, we see the further need for reporting individual case study research on concrete methods and tools to provide a more detailed picture and lessons learned regarding concrete techniques and settings. Moreover, we plan to publish a more detailed description of the applied evaluation approach and of our experiences to make it easier for other researchers to apply it in other settings. Study material comprising equations for cost calculations, templates, and questionnaires can be obtained from the authors on request.

## Acknowledgment

The research leading to these results received funding from the ARTEMIS Joint Undertaking under grant agreement no. 269335 (ARTEMIS project MBAT) and from the German

Federal Ministry of Education and Research (BMBF) including the research projects ARAMiS (grant 01IS11035), SPES (grant 01IS12005E). We would also like to thank all study participants for their support and Liliana Guzmán and Sonnhild Namingha for their initial review of this paper.

## REFERENCES

[1] V. Basili, D. Rombach, "The TAME project. Towards improvement-oriented software environments," *Transactions on Software Engineering*, vol. 14, pp. 758-773, 1988.

[2] V. Basili, A. Trendowicz, M. Kowalczyk, J. Heidrich, C. Seaman, J. Münch, D. Rombach, *Aligning organizations through measurement – The GQM+Strategies approach*, Springer, 2014.

[3] V. Basili, et al., "Linking software development and business strategy through measurement," *Computer*, vol. 43, pp. 57-65, 2010.

[4] M. Biehl, J. El-khoury, F. Loiret, M. Törngren, "On the modeling and generation of service-oriented tool chains," *Software and System Modeling*, vol. 13, pp. 461-480, 2014.

[5] R. Binder, "Model-based testing user survey: Results and analysis," http://robertvbinder.com/wp-content/uploads/rvb-pdf/arts/MBT-User-Survey.pdf, 2011. Retrieved on 17 Oct 2014

[6] L. Briand, "A Critical Analysis of Empirical Research in Software Testing," in Proc. of 1st Int. Symp. on Empirical Software Engineering and Measurement, pp. 1-8, 2007.

[7] M. Broy, "Challenges in automotive software engineering", in Proc. of 28th International Conference on Software Engineering, pp. 32-42, 2006.

[8] M. Ciolkowski, *An Approach for quantitative aggregation of evidence from controlled experiments in software engineering*, Fraunhofer Verlag, Stuttgart, 2012.

[9] M. Di Bacco, V. Mocellin, "A Bayesian justification for the linear pooling of opinions," *Journal of the Italian Statistical Society*, vol. 1, pp. 325-334, 1992.

[10] DO-178C, Radio Technical Commission for Aeronautics Software, DO-178C:2011 Considerations in airborne systems and equipment certification, 2011.

[11] F. Elberzhager, T. Bauer, "Optimizing quality assurance strategies through an integrated quality assurance approach," in Proc. of 40th EuroMicro Conf. on Software Engineering and Advanced Applications, pp. 402-405, 2014

[12] F. Elberzhager, J. Münch, D. Assmann, "Analyzing the relationships between inspections and testing to provide a software testing focus," *Information and Software Technology*, vol. 56, pp. 793-806, 2014.

[13] P. Helle, W. Schamai, "Towards an integrated methodology for the development and testing of complex systems-with example," *International Journal on Advances in Systems and Measurements*, vol. 7, no. 1 and 2, pp. 129-140, 2014.

[14] W. Herzner et al. "Expressing best practices in (risk) analysis and testing of safety-critical systems using patterns," in 2nd Int. Workshop on Risk Assessment and Risk-driven Testing, in conjunction with 25th ISSRE, 2014.

[15] ISO 26262, International Standardization Organization, ISO 26262:2011 - Road vehicles – Functional safety, 2011

[16] N. Juristo, A. Moreno, S. Vegas, "Reviewing 25 years of testing technique experiments," *Empirical Software Engineering*, vol. 9, pp. 7-44, 2004.

[17] N. Juristo, et al., "Comparing the effectiveness of equivalence partitioning, branch testing and code reading by stepwise abstraction applied by subjects", in Proc. of 5th Int. Conf. on Software Testing, Verification and Validation, pp. 330-339, 2012.

[18] O. Kacimi, C. Ellen, M. Oertel, D. Sojka, "Creating a reference technology platform - Performing model-based safety analysis in a heterogeneous development environment," in Proc. of 2nd MODELSWARD, pp. 645-652, 2014.

[19] D. Kästner, U. Brockmeyer, M. Pister, S. Nenova, T. Bienmüller, A. Dereani, "Combining model-based analysis and testing," in Proc. of the Embedded Realtime Software and Systems Congress, 2014.

[20] B. Kitchenham, et al. 2010, "Systematic literature reviews in software engineering – A tertiary study," *Information and Software Technology*. vol. 52, pp. 792-805, 2010.

[21] B. Kitchenham, L. Pickard, S. Pfleeger, "Case studies for method and tool evaluation," *Software*, vol. 12, pp. 52-62, 1995.

[22] M. Kläs, A. Trendowicz, A. Wickenkamp, J. Münch, N. Kikuchi, Y. Ishigai, "The use of simulation techniques for hybrid software cost estimation and risk analysis," *Advances in computers*, Academic Press, vol. 74, pp. 115-174, 2008.

[23] M. Kläs, T. Bauer, U. Tiberi, "Beyond herding cats: aligning quantitative technology evaluation in large-scale research projects," in Proc. of 14th Int. Conf. on Product-Focused Software Process Improvement, pp. 80-92, 2013.

[24] R. Larrick, J. Soll, "Intuitions about combining opinions: Misappreciation of the averaging principle," *Management Science*, vol. 52, pp. 111-127, 2006.

[25] P. Liggesmeyer, M. Trapp, "Trends in Embedded Software Engineering," *Software*, vol. 26, pp. 19-25, 2009.

[26] R. Marinescu, M. Saadatmand, A. Bucaioni, C. Seceleanu, P. Pettersson, "A model-based testing framework for automotive embedded systems," in Proc. of 40th Euromicro Conf. on Software Eng. and Advanced Applications, pp. 38-47, 2014.

[27] MBAT project website, https://www.mbat-artemis.eu. Retrieved on 17 Oct 2014.

[28] M. Morgan, M. Henrion, *Uncertainty a guide to dealing with uncertainty in quantitative risk and policy analysis*, reprint ed. Cambridge University Press, 1992.

[29] A. Neto, R. Subramanyan, M. Vieira, G. Travassos, F. Shull, "Improving evidence about software technologies: A look at model-based testing," *Software*, vol. 25, pp. 10-13, 2008.

[30] B. Nielsen, "Towards a method for combined model-based testing and analysis," in Poc. of 2nd MODELSWARD, pp. 609-618, 2014.

[31] Official website of the AUTOSAR partnership. [Online] http://www.autosar.org/. Retrieved on 15 Jan 2015.

[32] Open Services for Lifecycle Collaboration (OSLC) web-site. [Online] http://open-services.net/. Last visited 2014-10-22

[33] P. Runeson, M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, pp. 131-164, 2009.

[34] M. Utting, B. Legeard, *Practical model-based testing: A tools approach*, Morgan Kaufmann, 2007.

[35] W. Humphrey, "The software quality challenge," *Crosstalk* – The Journal of Defense Software Eng., vol. 21, pp. 4-9, 2008.

[36] J. Zander, I. Schieferdecker, P. Mosterman, *Model-based testing for embedded systems*, CRC Press, 2011.