



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG



# **Optimization of Test Data in a Resource Limited Environment in Automotive Domain**

Bachelor of Science Thesis in Software Engineering and Management

**MATTIAS LANDKVIST**  
**SANJA COLAK**

Department of Computer Science and Engineering  
UNIVERSITY OF GOTHENBURG  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2018



The Author grants to University of Gothenburg and Chalmers University of Technology the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let University of Gothenburg and Chalmers University of Technology store the Work electronically and make it accessible on the Internet.

**Design Science Research: Optimization of Test Data in a Resource Limited Environment in Automotive Domain**

MATTIAS LANDKVIST  
SANJA COLAK

© MATTIAS LANDKVIST, August 2018.

© SANJA COLAK, August 2018.

Supervisor: IVICA CRNKOVIC

Examiner: MICHEL CHAUDRON

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering  
UNIVERSITY OF GOTHENBURG  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2018

# Optimization of Test Data in a Resource Limited Environment in Automotive Domain

Mattias Landkvist

Software Engineering and Management Department  
University of Gothenburg  
Gothenburg, Sweden  
mattias.landkvist@gmail.com

Sanja Colak

Software Engineering and Management Department  
University of Gothenburg  
Gothenburg, Sweden  
guscolsa@student.gu.se

**Abstract - During vehicle software testing the amount of data collected is ever increasing. With bandwidth limited measurement systems, comes the need for highly optimized test setups to be able to collect as much useful data during testing as possible. In this paper, our goal is to optimize the measurement setup creation in order to ensure that collected data is optimized and usable for further analysis. The resulting approach, identification of signal requirement prioritization principles and highlighting the data set optimization principles supported with a script to automate identified principles, improves overall quality of the measurement setup.**

*Key words – Bandwidth, common measurement setup, ECU, development ECU, measurement, measurement setup, optimization, optimization principles, resource limited environment, signals, signal requirements, software calibration*

## I. INTRODUCTION

Nowadays, automotive innovation is mainly found in electronics and software [1]. Manufacturers produce more and more vehicle variants to cater the requirements of the customers [2]. Another driver of increased complexity is the ever-growing complexity of the legislation [3]. In response to all this, the automotive manufacturers trend toward reusing the same powertrains in several vehicle applications in the hopes of reducing development time and costs. This is achieved by developing a generic software for the powertrains which is possible to calibrate to make it fit a particular application [2]. As functions are developed and calibrated, data is collected to analyze the system behavior. The behavior is analyzed by measuring system signals and statistically evaluating the behavior using the collected measurement data [4].

This thesis work is performed at the Diagnostics and Dependability section within the Powertrain Controls and Calibration Department of Volvo Car Group. The department is responsible for the development and calibration of the powertrain software used in Volvo vehicles. The section is responsible for the development of diagnostic and safety related functionality within the complete software package. Within the section the goal is to shorten the development time by optimizing the measurement data collection during software testing in

vehicles. The approach is to create a common measurement setup in such a way that as many functions and systems as possible can be analyzed and calibrated with the data collected during a single software test performed in vehicles. Tests are carried out by function developers during the software function development phase. Later in the project, during the calibration phase, the testing is normally carried out by calibration engineers within a calibration team, which is led by a Calibration Leader (CL). In most cases, testing is performed in prototype vehicles, containing all new technical content introduced with the project. Normally, tests are executed in various environments such as laboratories, wind tunnels, test tracks and public roads across the world. This is done to stress test the mechatronic system in conditions that the vehicles can be exposed to during the product lifetime.

The measurement setup defines what signals will be measured and recorded during a test. All calibration engineers within the development team of a project contribute to this setup. The end goal is to reduce the total number of tests and number of physical vehicles and prototypes used for measurement data collection during software development and calibration.

The main problem found in the current practice, is the lack of signal prioritization during measurement setup creation. Today, signal selection is done in an unstructured way and with minimal regards to what the optimal signal selection should be. This leads to data collected during testing being incomplete and the full potential of reducing vehicle needs and number of tests cannot be achieved.

### A. Research questions

The focus of this thesis work is to identify principles which can be used to optimize the measurement setup used by a development team in a vehicle project. The measurement setup is used by the entire team when testing software. This enables the team to share measured data by passively measuring data, required by others in the team, in the background when collecting the data they need for their own software testing. In the company, this setup is named “Common measurement

setup”. See Fig. 1 below for a general overview of setting up and using the Common Measurement Setup.

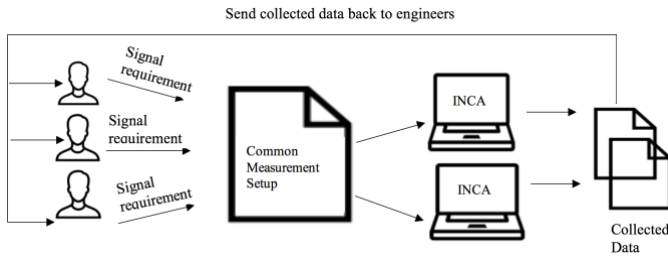


Fig. 1: General overview of creating and using the Common Measurement Setup

As the measurement environment is resource limited in terms of bandwidth, some means have to be identified that will help the calibration team create a common measurement setup for data collection by utilizing the available bandwidth in the most optimal way. The measurement environment consists of development Electronic Control Units (ECUs), measurement hardware developed by ETAS GmbH. [5], and the ETAS INCA measurement software installed on a laptop, which enables engineers to collect data from parameters within the software. The development ECU is a specific type of control unit, which differs compared to the production version of the same ECU, by having additional hardware added to enable measurement with the ETAS hardware and software package. Typically, this type of ECU is delivered during development by the same ECU supplier that later delivers the production ECUs when development has been finished and production started. This hardware has a limited capacity and the bandwidth available does not support measurement of all available parameters concurrently. Simple overview of measurement environment showed in Fig. 2 below.



Fig. 2: Simplified overview of the measurement environment

Our research is driven by the following main research question:

RQ 1: What are the possible optimization means for creating the measurement setup to ensure optimal data collection during testing with limited data collection resources?

Our approach here is: a) to prioritize the signal requirements, and b) to optimize the signal set that must be collected for the tests. This leads to two questions:

RQ 2: How to prioritize requirements on signals in a simple and efficient way?

RQ 3: How to reduce the input signals data set to optimize the resource utilization?

The goal of addressing these questions should result in a more efficient test process, but it can decrease the accuracy or the quality in general of the collected data during testing. For this reason, we need to validate the proposed approach. We state a question related to the test results:

RQ 4: What benefits and disadvantages do the end users perceive of applying the identified optimization principles?

### B. Definitions

To keep consistency in the terminology, we present terms used in this paper and explanation of what the words refer to. The terms and their definitions are:

- *Bandwidth* – In this report bandwidth is the available communication capacity between the ECU and measurement system used for collecting data.
- *Common measurement setup* – A measurement setup which is created and used by multiple teams in a project for data collection during vehicle testing.
- *Development ECU* – An ECU specifically used during development of software. It has increased capacity for software flashing, calibration and measurement compared to a production ECU.
- *ECU* – Electronic Control Unit.
- *Measurement* – Process of collecting real-time data values of signals from the development ECUs.
- *Measurement environment* – The system used for data collection during testing. Typically, this consists of a development ECU hardware, ETAS ES59x/ES69x measurement module, a laptop and the ETAS INCA software installed in the laptop.
- *Measurement setup* – Created experiment in ETAS INCA software used for setting up what data is to be collected during measurement.
- *Optimization* – Refers to optimization of signals selected in the measurement setup when taking signal priorities and resource availability into account. Ensure usefulness of collected data.
- *Optimization principles* – Principles used to achieve the best possible measurement setup with regards to usability of collected data.
- *Resource limited environment* – Refers to the limitation of bandwidth available for transmitting measured signal data from the ECU to the measurement software. Typically, the limitation is per sample rate and not a total bandwidth dynamically allocated to all available measurement sample rates. Note: this may differ

between projects, as some projects and ECUs do support dynamic allocation of the available bandwidth.

- *Signal* – A variable that represents information. This usually relates to transfer, processing or storage. In this report, a signal represents a measurement parameter containing a value for a specific software parameter in the ECU. It can be measured with the measurement system.
- *Software calibration* – Refers to setting up the generic software in the ECU so that the system performs optimally in the current application.
- *Signal requirements* – Refers to various lists containing names of required signals and sample rates to use during measurement for each signal. Sample rate requirements define the slowest allowed sample rate to use. The lists can be of varying file formats.

### C. Outline of the Paper

Section II discusses related work to our research problem and solutions applied by other researchers. Section III refers to the practical problems on which this study is based within Volvo Car Group. Proposed solution is explained in section IV as well as clearly stated scientific and technical contribution of this research. Section V describes in detail what methodology is used to conduct this study and how the same is applied. Section VI presents the results received from the evaluation of the proposed solution. Section VII discusses the research work. Section VIII concludes this research and potential future work on the topic.

## II. RELATED WORK

Identifying, analyzing and resolving requirement conflicts is a very active research field in recent years [6]. Chentouf identified seven types of possible requirement conflicts, where one of the possible conflicts is that of incompatible requirements [7]. Incompatible requirements occur if two requirements are either ambiguous, incompatible or contradictory. An example of this type of conflict is if the same agent, in this case our measurement system, is required to perform the same operation on the same object, but at two different frequencies. The various techniques for resolving requirement conflicts are classified in three main categories by Aldekhail, Chikh and Ziani [6]. These are manual, automatic and general frameworks. In manual techniques, stakeholders and software engineers discuss and analyze requirements in order to detect conflicts and resolve those. Automatic techniques rely on tools to identify and manage requirements. General frameworks are techniques that cannot be categorized as either of the two as they can be a mix of both.

Promising techniques for prioritization of requirements are identified by Qiao Ma [8], which can be quickly automated by providing simple rules of grouping requirements. Examples are MoSCoW, proposed by Dai Clegg [9], and Planning Game, which are used within Agile software development. These techniques were found less difficult for the test persons to understand and achieved higher level of confidence from the

users, when compared to others. The main reason why these techniques were found inferior to others proposed, is that they result in less reliable end results and are less fault-tolerant. In our study we chose to work on these simpler techniques which would provide a lower threshold of acceptance for the target audience. This is the reason why the prioritization principles identified are based on MoSCoW and not on one of the more advanced techniques recommended.

## III. PRACTICAL PROBLEM DOMAIN AND SCOPE

### A. Problem Domain

The section where the research work is conducted has introduced new working methods over the last years in order to speed up testing and calibration of software. The process (see Fig. 3), and the tool chain developed within the section is Common Data Eval (CDE). The main goal of the process is to allow all calibration engineers in a development project to define standard evaluation reports that are used for analysis during calibration and function development once measurement data has been collected during testing. Data is continuously uploaded to a shared drive by the various engineers involved in the project. The evaluation reports are generated automatically by the CDE tool chain once all measurement data is available on the shared storage location. In order to ensure that the tool can successfully generate the required reports, a signal requirement specification is generated as a requirement on the measurement setup to be used during testing. Once the data has been evaluated, a new iteration of defining evaluation reports begins, based on the learnings from the last development iteration. The end result is a decrease of total number of tests required to complete a calibration and function development iteration. This is made possible by enabling the calibration engineers to base their calibration and function development on tests not only conducted by themselves, but also by using data collected in the background by other calibration engineers during testing.

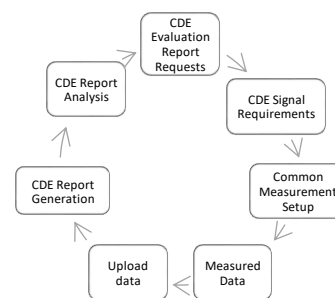


Fig. 3: CDE Process used within the company

The standard process (see Fig. 4), during a project start, is that the CL begins with a measurement setup from a related development project. The CL adds the signals required according to the CDE signal requirements specification into this setup.



Each of the calibration engineers provides a measurement requirements specification to the CL, which defines what signals need to be measured during testing. Finally, the CL adds more general signals to collect during testing, such as signals representing information on general driving conditions. Examples of these are vehicle speed, ambient temperature and engine temperature. The signal requirement specifications provided by CDE and the calibration engineers are of varying file formats.

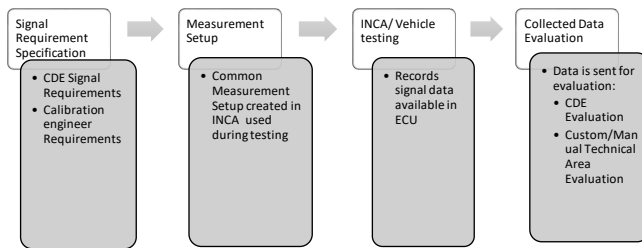


Fig. 4: Measurement process

An important technical aspect of the measurement software used by Volvo Cars, ETAS INCA [5], is that it only allows one sample rate per selected signal to record during measurement. If one signal is added more than once, only one sample rate of the signal will actually be recorded during measurement. Therefore, the signal cannot be duplicated during measurement, and it is not possible to record a signal in more than one sample rate.

#### B. Problem Statement

The measurement environment is limited in available bandwidth it can provide.

An identified problem is that due to this limitation, not all required signals can be included in the setup. There is currently no prioritization methodology or tool set in place to support the CL or the team in selecting the most important signals. The end result is that in many cases the final measurement setup used is lacking vital data for CDE evaluation.

A second problem identified is that the final measurement setup used has redundant data. This occurs when the measurement setup is based on a setup created for a previous project but not cleaned up before adding all the new signal requirements. The end result is that the setup includes signals of little or no value in the current project. This is a waste of measurement resource.

A third problem identified is that sampling rate requirements can be overwritten. The problem occurs when a specific signal is included in multiple requirement sources but with different sampling rate requirements.

## IV. SOLUTION

The solution is related to how to effectively prioritize requirements on signals and how to optimize data sets used in a measurement setup, as described in detail below.

### A. Signals requirements prioritization

#### 1) Priority between requirements

The technique used for prioritization is MoSCoW [8], which is a common technique within the Agile software development. Each calibration engineer creates and classifies their signal requirements. They can provide more than one signal requirement file, depending on how many functions they are responsible for developing and calibrating. The final approval of all requirement classifications is done at the review meeting led by the CL. All the calibration engineers in the project are present during the review meeting. No calibration engineer is allowed to classify their signals as “*Must*”, as the “*Must*” category is reserved for signals required by the CDE and general driving conditions defined by the CL. Each requirements source is classified according to:

*M – Must.* All signals required by the CDE signal requirements specification and general signals which record the general driving conditions of the vehicle. The latter are added by the CL. No calibration engineers are allowed to add signal requirements in this classification.

*S – Should.* Signals required for measurement and analysis of newly developed SW functions or HW which is new in the vehicle project.

*C – Could.* Signals required for measurement and analysis of software and hardware systems that are already in production. Typically, these can be of interest if the team want to analyze if there is room for improvement of already released software.

*W – Won’t.* Signals in the current measurement setup, but not required by any stakeholder. No involved stakeholder actively adds signals into this classification.

#### 2) Requirement conflict identification

The signal requirement sources are analyzed and signals that exist in multiple sources but with different sampling rate requirements are identified. This enables the CL to start a dialog with the affected stakeholders in order to identify the most appropriate sample rate to use in the final measurement setup.

### B. Data set optimization principles

#### 1) Identification of missing signals

The current measurement setup is compared to all signal requirement specifications and a report containing what signals are missing is created. In this way the CL can either add the

missing signals to the measurement setup or contact the owner of the signal requirement specification not fully covered and discuss the level of importance of the missing signals.

2) *Identification of non-required signals*

The measurement setup is compared to all signal requirement specifications and a report containing what signals are included in the current setup, but not required by any specification, is created. This enables the CL to identify what signals can be removed from the current setup in order to reduce the measurement capacity utilization.

3) *Resource utilization*

The measurement setup is analyzed for the current utilization level of the measurement environment. This enables the CL to identify how much available measurement resource exists for lower priority signals to be added. The intention is to have the CL only add “*Must*” signals to the measurement setup. “*Should*” signals are only added if the utilization view shows that there is remaining available measurement resource capacity after all “*Must*” signals have been added. In a similar fashion, signals classified as “*Could*” are added in the measurement setup if the utilization view shows remaining available resource after all “*Should*” signals have been added.

“Priority between requirements” is introduced as a best practice in the department and training material was developed. In order to support “Requirement conflict identification” and “Data set optimization principles”, the MSS script was developed. The technical contribution of the work is to understand domain requirements, and based on that, in communication with the stakeholders, identify possible optimization principles. Then design and implement a system that will provide an optimized selection of data. From the

research point of view, the work addresses a specific optimization problem within resource (bandwidth) limitations.

V. RESEARCH METHODOLOGY

A. Research design

This research project adopts design science research methodology [10]. Design science research is most suitable for this work because it is a problem-solving oriented method [10]. In our case this is identification of principles for the optimization of the measurement setup in a resource limited environment and the development of a script for supporting the optimization work.

The study was conducted iteratively and incrementally, and it was evaluated by applying two evaluation methods, Expert Evaluation and Technical Experiment [11].

The research process was done iteratively in six phases as shown in Fig. 5.

In Step 1 (Problem identification) we identified the research problem, defined the practical problem and scope of the research. This step helped us find the research focus and to understand why the addressed problem needed to be researched.

Step 2 (Objectives of a solution) defined the objectives of a solution for the existing problem. We aimed to optimize the test data collection by applying identified optimization principles when creating the measurement setup.

In Step 3 (Design & Develop) we designed and developed the artefacts. The script implemented the identified principles that were possible to translate to an algorithm for this particular problem domain.

In Step 4 (Demonstration) we demonstrated our work to the stakeholders within Volvo Car Group. The demonstration was performed by creating a measurement setup using the artifacts created during the progress of this research. After each

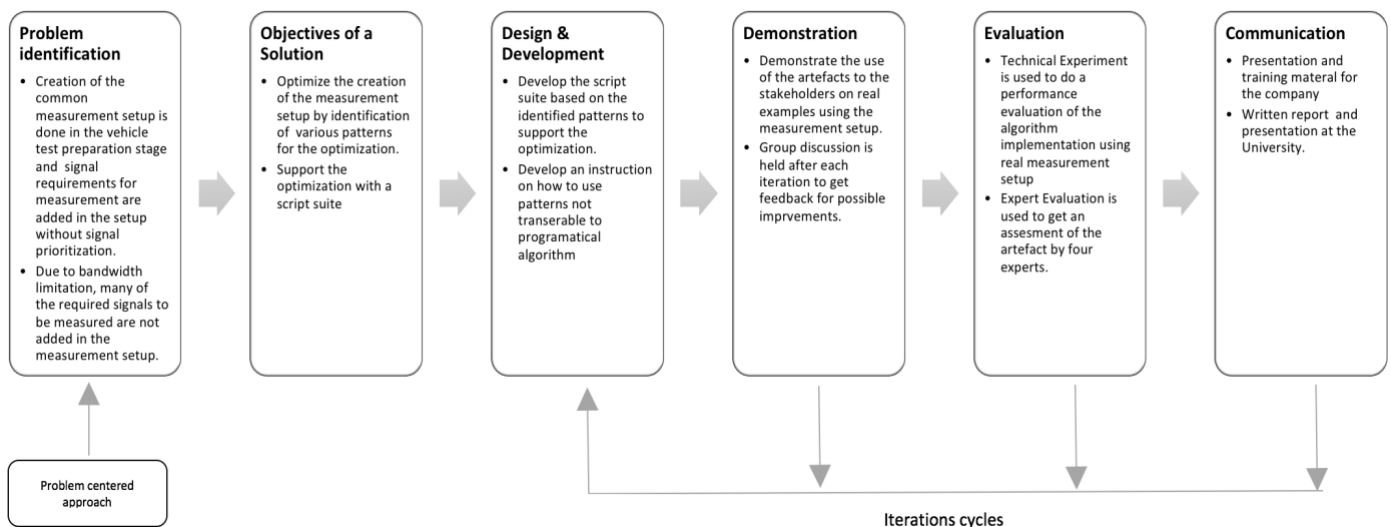


Fig. 5: Design science research process used in this study

demonstration session, a feedback discussion was held. The number of participants varied between the various demonstration sessions.

In Step 5 (Evaluation) the artefacts were evaluated using Technical Experiments and Expert Evaluations, further described in subsection V.B “Evaluation” below.

In Step 6 (Communication) we held a presentation within Volvo Car Group focusing on the findings of the research. During the presentation a quick overview of the created training material was presented as well. The research study was also presented at the University of Gothenburg, where the work was defended.

### 1) *Measurement Setup Support (MSS)*

MSS was designed and developed iteratively using the Python programming language. The script was presented and demonstrated to the stakeholders on regular intervals. We devised the script to automate the principles identified which could be translated to an algorithm. As the measurement setup was a .exp file, a format used by ETAS INCA, the INCA COM API was used in Python to access information on the current measurement setup. This was also the case for many of the signal requirement sources which were also in the same file format. The script consists of two main functionalities. The functionalities are: Analyze and Cleanup.

#### *Analyze*

The main purpose of this functionality was to assist the measurement setup creator to find missing signals in the setup. This script reads all files containing signal requirements provided by the different calibration engineers. The requirements are compared to the current measurement setup and a report is generated to the script user. The report contains a list of signals that are missing in the setup but required in the requirement specification files. The report also specifies which signals can be removed from the current setup as those signals are not required by any of the requirement specification files. The script also identifies requirement conflicts when a signal has more than one requirement on sample rate. If such conflicts are identified, these conflicts are highlighted in the report.

#### *Cleanup*

The main purpose of this functionality was to support the creator of the measurement setup to remove all non-required signals in the current measurement setup. It performs automatic removal of these signals.

The final result of the script was presented as a report with statistic of the measurement setup based on the introduced principles.

For the pseudocode of MSS, see Appendix C.

## *B. Evaluation*

K.Peppers et. al identified eight main methods of evaluation method types when conducting design science research evaluation [11]. In this work we chose to make use of the Prototype, Technical Experiment and Expert Evaluation methods in order to achieve answers to our research questions. Initially, a prototype was created implementing the identified principles which were possible to translate into an algorithm.

The prototype consisted of a script, MSS. Principles we could not implement in MSS were translated into an instructional guide for the engineers. The resulting prototype script and instructional guide were evaluated in a Technical Experiment. In the Technical Experiment the CLs were given two tasks. The first task was to create a common measurement setup using the current working methods. The second task was to use the created instructional guide and MSS in order to create a second measurement setup.

Both created measurement setups were evaluated according to the following criteria:

- Must priority signals included in the measurement setup
- Should priority signals included in the measurement setup
- Could priority signals included in the measurement setup
- Won't signals included in the measurement setup
- Non-valid signals included in the requirements
- Must priority signals missing in the measurement setup
- Should priority signals missing in the measurement setup
- Could priority signals missing in the measurement setup

The results from using both methods were compared in order to identify whether the introduced instructional guide and the script resulted in a higher quality measurement setup. Higher quality was defined as having a greater share of the required signals included in the measurement setup while at the same time reducing the number of non-required signals included. Far greater importance was given to signal inclusion share of very high and high classified signals than low important signals.

Three CLs with previous calibration leader experience were asked to conduct both tasks. The number of possible participants was limited as the evaluation was conducted during a period of expeditions, which are test trips to various climates performed by the engineering teams and thereby make them unavailable in their normal geographical location. By comparing objective aspects between the two methods, an unbiased result was expected. Once the Technical Experiment evaluation had been completed, an Expert Evaluation was conducted. Semi-structured interviews with open-ended questions with each CL, that had participated in the Technical Experiment, were conducted. A list of questions was created based on the data collected from the Technical Experiment. The Expert Evaluation aided us to get an overview of the artefact usefulness and possible improvements in the future. This also gave us an indication of expected acceptance level within the organization of the instructional guide and script introduced.



### C. Validity Threats

#### 1) Internal validity

One internal validity threat identified was a result of the evaluation being conducted during an expedition period. This led to a limitation of available roles that could be used for evaluating the proposed technique and MSS from different stakeholder perspectives. Additionally, there was the threat of selection bias as the participants in the evaluation were not selected randomly but were selected by us. As the selected candidates in the evaluation phase were also part of the problem identification phase, there is a risk that the proposed result of this research is too heavily biased towards the preferences of the selected practitioners for the evaluation. A third internal validity threat identified that could affect the research results is the risk of human error. Initial requirements source classification is done by the calibration engineer. The CL can review all “Should” and “Could” sources in order to decide if they belong in this classification or not. As all prioritization classifications are done by humans and based on their experience and knowledge, this is a potential risk to both the end result and also the conclusions of this research as the classification may end up not being optimal for the project.

#### 2) External validity

As the study was conducted within only one company, it was not proven that the results were applicable and valid in other companies. Another external validity threat identified is that MSS was developed using the ETAS INCA COM API and therefore includes the capabilities and possibilities allowed within this API. Other companies may use other measurement software and MSS would have to be modified according to what is allowed in their equivalent API. During this research, other measurement software and their API capabilities were not studied.

## VI. RESULTS

The first research results consist of data given from the Technical Experiment conducted in three projects at the section. The first step was to follow the created instructional guide for the projects in order to classify all requirements according to the decided priority scheme. The structure uses one folder for each of the requirements classifications. These are “MUST”, “SHOULD” and “COULD”. See Fig. 12 for an example of the structure used. Only the CL was allowed to prioritize and classify signal requirements during the experiment. The only requirement from us was that CDE signal requirements (*required\_signals.csv*) and general driving condition signal requirements (*General.exp*) had to be classified as “MUST”. This in order to secure that CDE evaluation can be successfully completed when analyzing collected measurement data.

Once the priority task was completed, the existing measurement setup in each project was analyzed by MSS, which provided us with data on the current state of the measurement setups used in the projects. The following step was to create an

updated measurement setup for each of the projects, by applying the optimization principles introduced in this research. The updated measurement setups were analyzed using MSS and the results of the new measurement setups compared to the original ones can be seen in Fig. 6, Fig. 7 and Fig. 8 below. The “Before optimization” bars represent the original measurement setups that are currently used in the projects, while the “After optimization” bars represent the new updated measurement setups created by using the optimization principles introduced in this research. The histogram charts contain information about the numbers of signals in each of the categories used during analysis as defined in “V.B. Evaluation” above. Our focus during evaluation was mainly on “Must” signals.

The expectations were if there were missing “Must” signals in the original measurement setup, these would be added. If there was no free measurement resource available, the result would be that resource would be available by removing “Won’t”, “Could” and “Should” signals, in that particular order. In a similar fashion “Should” signals would be added at the expense of “Won’t” and “Could” signals. “Could” signals would only be added if there was remaining measurement resource available at the end of adding all “Must” and “Should” signals and also removing all “Won’t” signals.

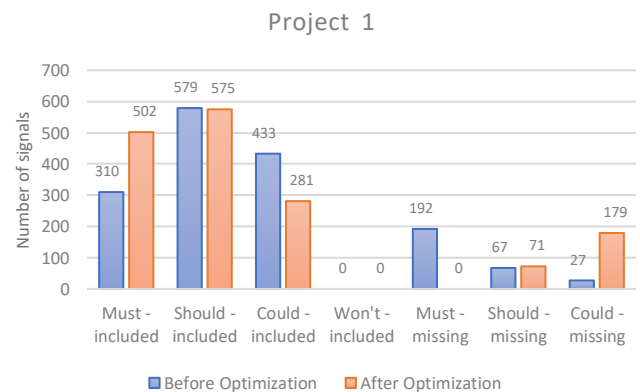


Fig. 6: State of the measurement setup for Project 1 before and after applying optimization principles

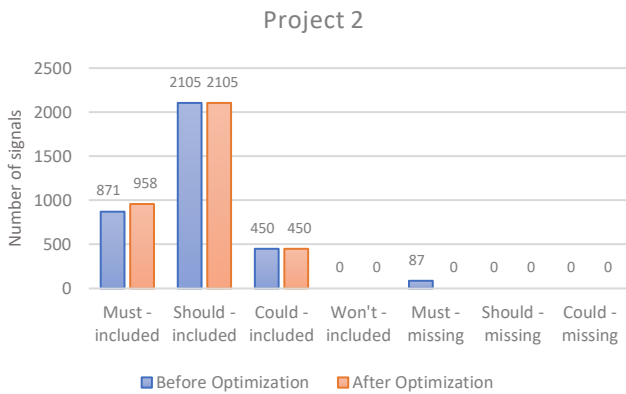


Fig. 7: State of the measurement setup for Project 2 before and after applying optimization principles

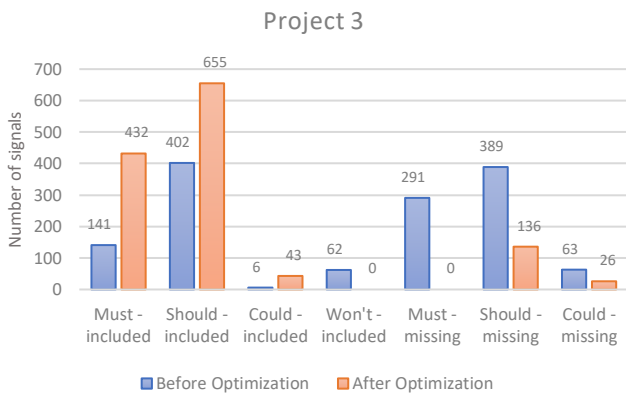


Fig. 8: State of the measurement setup for Project 3 before and after applying optimization principles

In Project 1, the total number of signals which had the “Must” classification was 502 while only 310 were included. A clear rebalance of available measurement resource was noted in the results of the optimized measurement setup. All signals classified as “Must” were included, as expected. As a result of adding these additional signals to the measurement setup, 4 signals with “Should” classification and 152 signals with “Could” classification were removed. They were removed in order to make enough bandwidth available for the higher priority signals to be included. Unexpectedly, 281 signals with “Could” classification remained in the optimized measurement setup even though 71 signals with “Should” classification remained missing. Initial expectations were that “Should” signals would be prioritized over all “Could” signals. Therefore, the expectation was that as long as there were “Should” signals missing, no “Could” signals would be included in the optimized measurement setup. After further analysis, two reasons for this behavior were found. Firstly, some signals of the “Could” classification were also included in some “Must” or “Should”

classified sources. For this reason, the signals were in fact treated as “Must” signals when being added to the measurement setup, but at the same time they were also included in the “Could” statistics as well. The second reason was that some of the “Should” signals that were missing, were required to be measured in sample rates that were fully utilized after “Must” signals had been added. It was found that some of the “Could” signals that remained after optimization had requirements to be measured in sample rates which were not fully utilized. For this reason, there was resource available to include these “Could” signals in the optimized measurement setup, but at the same time no resource available to include some of the “Should” signals.

It is clearly shown that Project 2 had the same issue identified in Project 1. In Project 2, 87 “Must” signals were not included in the original measurement setup. A different result was achieved through optimization when compared to the results of Project 1. As the available measurement resource was not fully utilized when using the current working method for creating the measurement setup, no signals amongst the required ones were omitted in the final result. It was noted that all 87 signals of the “Must” classification that were missing, were added into the final measurement setup. For this particular test case, prioritization was not required, but the use of MSS ensured that no signals were omitted when creating the measurement setup.

Project 3 showed the greatest problem as more than two thirds of the required “Must” classified signals were missing in the setup. We also see that in the original measurement setup used in Project 3, 62 included signals were not required at all (“Won’t”). This is a symptom of basing the measurement setup on a measurement setup used in previous projects without updating it according to the new requirements. This negatively affected the measurement data, where instead of having an optimized measurement setup, available resource was wasted on collecting non-required data. A clear improvement was noted by applying the suggested optimization principles and using MSS. All “Must” classified signals, 432 in total, were included in the optimized measurement setup. As in Project 1, it was noted in Project 3 that some signals classified as “Could” remained in the optimized measurement setup despite some signals classified as “Should” remained missing. After further analysis, it was found that the reason for this pattern in Project 3 was the first reason mentioned in the results of Project 1, all included “Could” classified signals were also required by “Must” signal requirement sources. Finally, the waste of including many non-required signals was resolved by removing all of those signals from the optimized measurement setups, as can be seen in the histograms.

After each test, resource utilization was analyzed and summarized in Fig. 9, Fig. 10 and Fig. 11 below. The results were extracted using ETAS INCA measurement SW as the API used by MSS did not include access to resource utilization data.

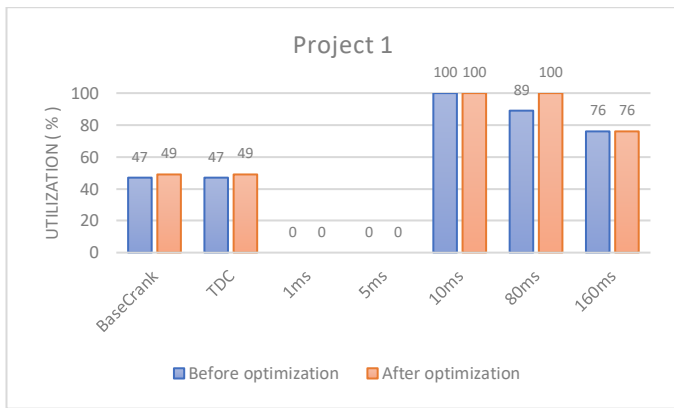


Fig. 9 Project 1 resource utilization before and after applying optimization principles in %

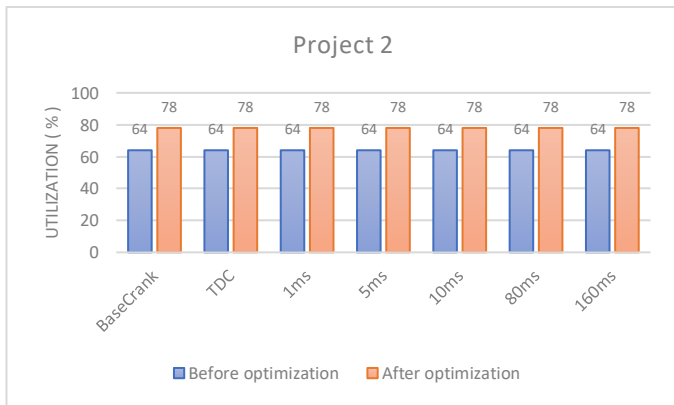


Fig. 10 Project 2 resource utilization before and after applying optimization principles in %

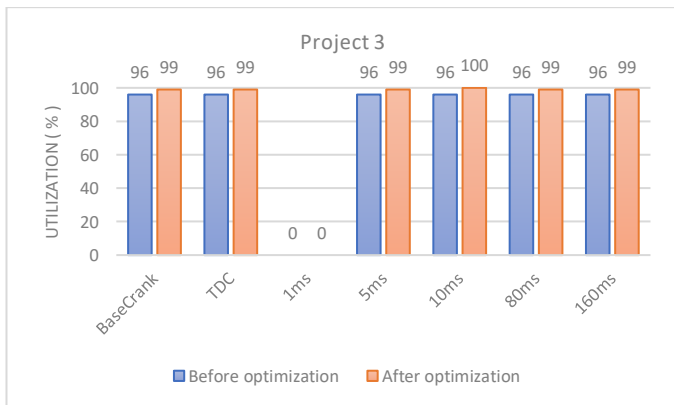


Fig. 11 Project 3 resource utilization before and after applying optimization principles in %

As can be seen in Fig. 9 above, Project 1 already had a high resource utilization in the 10ms sample rate. While most of the improvement noted when analyzing the results was a result of rebalancing available resource, it should be noted that the 80ms sample rate was previously not utilized to 100%. The introduced

techniques and MSS ensured that the available resource was used to a higher utilization degree. However, we can also see that the 1ms and 5ms sample rates are empty. This means that no signals were required to be measured in these sample rates. It indicates that the selected algorithm for adding signals to the measurement setup does not use the available resource fully. Signals that were required with the “Should” classification with a 10ms or 80ms sample rate requirement could have been added to the 1ms or 5ms sample rates.

Project 2 is different compared to the others as it is a “Generation 3” powertrain project. For this project there is a new requirement by the department on the measurement system itself, which demands the available measurement resource to be dynamically shared between all available sample rates. In older development projects, each sample rate would have a fixed allocation of the available bandwidth. For this reason, all sample rates show the same level of utilization in Project 2. As can be seen in Fig. 10 above, the available measurement is not fully utilized, which explains why all required signals were included in the optimized measurement setup.

In Project 3, the end result was a higher utilization of the available resource. As we can see in Fig. 11 above, the 10ms sample rate ended up being fully utilized with no more room for added signals. As in Project 1, there was a sample rate that was not fully utilized. The 1ms sample rate was not used at all and this sample rate could have been used to include some of the missing “Should” classified signals to get a better requirements coverage with the final measurement setup.

After further analysis, the BaseCrank sample rate was also fully utilized despite being shown at a 99% utilization level. It was not possible to add any of the missing signals required to be measured in this sample rate. The fact that it was shown at 99% instead of 100% may be a bug in the ETAS INCA SW. Another possibility is that the signals required are too large to fit in the remaining bandwidth, as measurement signals can be of varying size from 1 bit to 4 bytes in the system analyzed.

After analyzing the results, Table 1 below shows the number of signals identified as signals appearing in more than one signal requirement source with different sample rate required by the sources and this leads to signal requirement conflicts. MSS automatically adds each signal according to the first priority classification read, which may not always be “the correct” sample rate to select. The generated report has a specific section where all signals with conflicts are listed. The report also specifies the different requirements on the signal and the sources of each requirement in order to help the CL decide on the “correct” requirement for the final measurement setup. Alternatively, the CL has to setup a meeting with involved parties to discuss and resolve requirement conflicts. An example from a report can be seen in Fig. 15 below. The number of conflicts found in the three projects differ and a summary can be seen in Table 1 below.

Table 1: Total number of signals with requirement conflicts

	Signal Requirement Conflicts	Total number of required signals
<b>Project 1</b>	123	1608
<b>Project 2</b>	304	3513
<b>Project 3</b>	129	1292

Each CL participating in the Technical Experiment gave their feedback and opinion through a short interview at the end of the evaluation phase. In general, the experts interviewed were very positive with regards to the simplicity and working method for signal requirement prioritization. The reports generated by MSS were also greatly appreciated. Concern was raised with regards to adoption of the new working methods as the experts were of the opinion that it is difficult to introduce new principles in the organization. Additional features to MSS were suggested to make the script set even more useful in the future. These answers gave us insight in positive and negative aspects of the suggested optimization principles. This provided us with an Expert Evaluation. These were the interview questions and the summaries of the answers:

1. **Did you experience any benefits using MSS and the suggested work method instruction?**
  - The tool helps with better organization of the measurement setup, more utilization and it will give overview of the setup status which will save time for checking the setup.
  - The tool saves time when creating the measurement setup.
  - One of the biggest benefits is that the tool helps clean the measurement setup. Historically, we have had great difficulty removing no longer needed signals in the setup.
2. **Did you experience any disadvantages using MSS and the suggested work method instruction?**
  - It can be difficult to introduce new practice of working for all stakeholders involved. Until the methodology of signal requirement classification is accepted by all stakeholders, the tool may not be used by every CL.
  - The process adds requirements on the calibration engineers to follow a specific structure. This may seem like an added workload on the calibration engineers at first and prevent fast adoption.
3. **Would you use this method and toolset in your future work?**
  - Definitely will try to use it. It is important to get all calibration engineers and calibration leaders to follow suggested work structure.
4. **Do you see potential for improvements? If yes, what?**

- The script may be done as a tool where filtering can be added as well as automatic choice of sample rates when it comes to signal requirement conflicts.
- Automatic import of measurement visualization setups. (Not possible with the current API).
- Add drop-down GUI where the script user can select one of the conflicting requirements and add this to the final measurement setup.

## VII. DISCUSSION

In this thesis we have proposed optimization principles for measurement setup creation with an automated support for signal requirement analysis. The proposed optimization principles ensure that important signals will be measured and collected.

Our fourth research question is answered through the subjective feedback by experts on the suggested optimization principles. The main benefits from the application of the optimization principles are mostly related to reduction of time invested when preparing measurement setups. This is achieved by providing a fixed structured way of creating the setup and by automating some of the time-consuming tasks. The disadvantages with the suggested method are mostly related to the requirement prioritization itself. Since the current working method is long established within the section, switching to a new working method could take some time before all stakeholders accept it.

Our third research question aimed on how to reduce the input data set to optimize the resource utilization. The main reduction of input data which led to the greatest optimization, was to remove lower priority signals from the measurement setup. This freed up resource for the higher priority signals. The result in one of the projects showed that it was possible to free up resource by removing signals that were no longer required but still present in the measurement setup.

MoSCoW, a methodology for requirement prioritization proposed by Dai Clegg [9], proved efficient in the suggested process in the research done by Qiao Ma [8]. Furthermore, it showed that the method is self-explanatory and made it easy for the CL to decide how to do the classification of the signal requirements. This answered our second research question. The results of the thesis show that the suggested optimization principles have a positive impact on the measurement data collection by increasing the number of important signals needed to the measurement setup.

Combining the results of RQ2, RQ3 and RQ4 gives us an answer to the main research question. The MoSCoW technique can be used to improve the creation of a measurement setup with limited data collection resources. However, it is important to select a better algorithm in MSS than the one used in this research as the available resource was not fully utilized after the new measurement setup had been created. As identified in the research by Qiao Ma [8], the MoSCoW technique was easy to

explain and quickly adopted by the CLs when trying to prioritize amongst the signal requirement sources. It is also clear by looking at Table 1 above, that we found many cases of incompatible requirements in all three projects analyzed. This was one of the seven types of possible requirement conflicts identified by Chentouf [7]. The approach suggested by the researchers for identifying this was by using the automated tool, MSS, and the suggested method to resolve them was by manually discussing the incompatible requirements with the involved stakeholders. This was in line with the suggested methods by Aldekhail, Chikh and Ziani [6].

In order to increase acceptance and speed up the measurement setup creation process, certain aspects of the process can be optimized yielding positive results in this research as identified during the Expert Evaluation.

### VIII. CONCLUSION

In this paper we identified a set of principles to optimize the test data collection in a resource limited environment. We carefully identified five optimization principles and applied them to the measurement setup creation and support them by automated analysis of the created measurement setup. The principles were applied on three different projects and evaluated by three experienced practitioners. The main principle is prioritizing signal requirements. Once this has been done it is possible to identify requirement conflicts, missing signals in the current measurement setup and signals in the current measurement setup that are not required, as three other important principles. Finally, keeping track of current resource utilization is the fifth introduced principle to ensure optimal data collection.

The MoSCoW requirement prioritization technique proved an efficient prioritization technique in this research. It is simple to classify the signal requirements according to the MoSCoW structure and it enabled us to create an automated support script, MSS, for the measurement setup analysis. With the help of MSS, non-required input signals were removed from the final measurement setup, which freed up resource and enabled a more optimal use of available measurement resource. Signal requirement prioritization optimized the resource utilization as well by exchanging lower priority input signal data sets for higher priority input signal data sets.

As proven by this research, the application of the stated optimization principles when creating the measurement setup and with the aid of MSS, clear benefits for the end users were shown. Better organization of the signal requirements, simple overview of the measurement setup status and automatic cleaning of the signals not required by any stakeholders, speeds up the measurement setup creation process and ensures that collected data is more useful. However, the adoption of new practices within the organization may take some time and following the specific structure could possibly increase the workload for the engineers involved. It should be noted that this research was conducted in a specific company, using a specific

hardware and software setup for collecting test data. While this company is not alone in the industry in using this setup, not every company in the automotive industry uses the same system. The optimization principles themselves are valid regardless, but the developed script would need modification to fit the used measurement system. While working on this study, we have not seen any similar study within the industry conducting a similar research. Our main contribution is proving that a requirement prioritization technique found in the software development field is applicable with positive results in software testing within the automotive industry.

In the future, other prioritization techniques should be studied to find possible candidates for even greater improvement. During the work on this report, we discovered that the problem is in fact a bin packing problem [12]. Algorithms within the bin packing problem domain should be further researched in order to find better solutions for MSS. This may result in a faster creation process and also a higher degree of optimization. Another possible future topic was noted during this research. While everyone involved in the artefacts evaluation agreed that the suggested techniques and script would vastly improve the current situation, it was expressed with great doubt that this work method would become widespread and adopted within a short timeframe. But if such an improvement can be noted, how come it takes so long to adopt? Since the problem is solved within one specific company, how to adopt the introduced process with other companies and how to generalize it, should be further studied and better understood.

## IX. APPENDIX

## Appendix A. Example of file structure of signal requirements

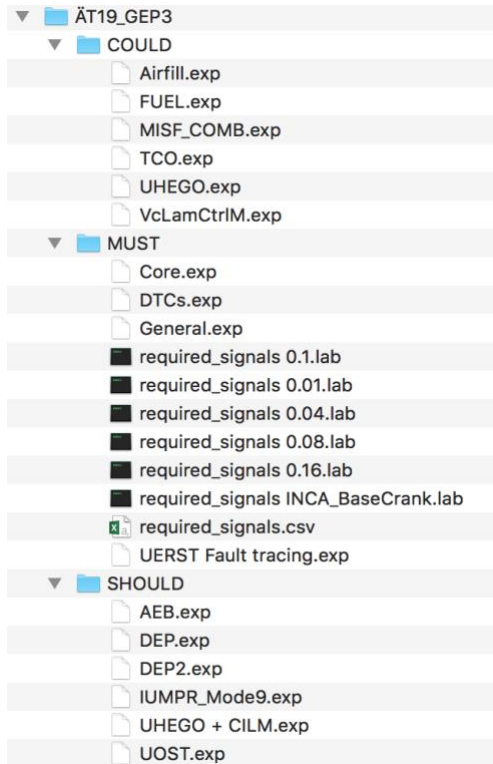


Fig. 12: Example of file structure after classification of the signal requirements

## Appendix B. Measurement Setup Support Report

The report contains following categories:

- Summary of created measurement setup
- Missing signals
- Requirement conflicts
- Non-valid requirements

Summary displays two sections: Measurement Setup Summary (*Panel Summary*) and Requirements Summary.

Measurement Setup Summary, as shown in Fig. 13, displays:

- Measurement setup name and total number of signals added into the setup. The number of total added signal number is split into two categories: *i.*) Required Signals Included *ii.*) Not Required Signal Included

Requirements Summary, as shown Fig. 13 displays:

- Classification of requirements and total number of signals in each category split into three sub categories: *i.*) Signals in the measurement setup *ii.*) Signals not in the measurement setup *iii.*) Invalid signals and *iv.*) the source of the requirement.

Figures 11-14 show the simple statistical detailed report after analyzing a measurement setup during evaluation step.



# SUMMARY

## Panel Summary

Common Panel Name	Signals - Total number	Required Signals Included	Not Required Signals Included
Transport_VEP_Gen2_180628_v9_post	1131	1131	0

## Requirements Summary

Classification	Total number	In Common Panel	Not in Common Panel	Not Valid	Sources
MUST	1458	502	0	956	
SHOULD	646	575	71	0	(S) DEP.exp (S) AEB.exp (S) UOST.exp (S) UHEGO + CILM.exp
COULD	460	281	179	0	(C) Airfill.exp (C) VcLamCtrlM.exp (C) FUEL.exp (C) UHEGO.exp

Fig. 13: An example report of the measurement setup summary

# MISSING SIGNALS

These signals are missing in the Measurement Experiment

## MUST SIGNALS

No issues found with MUST signals. All required signals included.

## SHOULD SIGNALS

Signal	Raster time	Source
rVcDepTre_Tq_AllwdCrshEstICEDiff	10ms	(S) DEP.exp
Rsf_TPSDif_DevAccu	10ms	(S) DEP.exp
xVcDepTre_B_PropRearAxisMonExe	10ms	(S) DEP.exp
Sfi_TgtLmdFinal_bank1	10ms	(S) AEB.exp (S) UOST.exp

Fig. 14: An example of a missing signals report

## REQUIREMENT CONFLICTS

These signals with conflicting requirements

Signal	Raster time	Source
Scm_AtmoPres_Filter	80ms	(M) General.exp (M) required_signals.csv
	10ms	(M) required_signals.csv
Scm_InManiPres_Filter	80ms	(M) General.exp (M) required_signals.csv
	10ms	(M) required_signals.csv
Scm_Batt_Volt	80ms	(M) General.exp (M) required_signals.csv (S) DEP.exp
	10ms	(M) required_signals.csv

Fig. 15: An example of a requirement conflicts report

## NON VALID REQUIREMENTS

These signals do not exist in the current software release

Signal	Raster time	Source
B_TipDwn	10ms	(M) required_signals.csv
B_TipUp	10ms	(M) required_signals.csv
Dem_NvmMemory.EventStatusExt[ROXR]	10ms	(M) required_signals.csv
Dem_NvmMemory.EventStatusExt[UOXR]	10ms	(M) required_signals.csv
NHpTub	80ms	(M) required_signals.csv
NLpTub	80ms	(M) required_signals.csv
N_HP	80ms	(M) required_signals.csv
N_LP	80ms	(M) required_signals.csv

Fig. 16: An example of a non valid requirements report

## Appendix C. MSS Pseudo Code

```

algorithm measurementsupport is
  define:
    ConflictFlag cf with identifier{"MAJOR" = 2, "MINOR" = 1, "NONE" = 0}
    Classification c with identifier{"MUST" = 3, "SHOULD" = 2, "COULD" = 1, "WON'T" = 0}

    Signal s as string
    Frequency f as an instance of TimeBasedFrequency or InteruptBasedFrequency
    Requirement r (Signal s, Frequency f, Classification c)

    List L (Requirement r)
    List O (Requirement r, ConflictFlag cf)

    Function append(Requirement r) //Add r to List

  input: L
  output: O

  for (r) in L:
    //Add requirement to T if classification is not "won't", "won't" is discarded
    if r.c > "WON'T":
      O.append(r)

  for (r,cf) in O:
    //CHECK FOR REQUIREMENT IN HARDWARE SPECIFICATION
    if  $\exists(r) \in H$ :
      O(r).cf = "MAJOR" //Set conflict of r to "MAJOR" if r doesn't exist in the
      HardwareSpecification

    //TIME BASED FREQUENCY CHECK ON SAME SIGNAL
    if  $\exists(r1.f \wedge r2.f) \in \text{TimeBasedFrequency} \wedge \exists(r1.s = r2.s)$ : O:
      if r1.f <> r2.f:
        O(r1).cf = "MINOR" //Set conflict of r1 to 1 which requires a general resolution
      else:
        O(r1).cf = "NONE" //No conflict thus can be handled automatically

    //INTERUPT BASED FREQUENCY CHECK ON SAME SIGNAL:
    //Set conflict of r1 to "MINOR", requires a general resolution
    else if  $\exists(r1.f \wedge r2.f) \in \text{InteruptBasedFrequency} \wedge \exists(r1.s = r2.s)$ : O:
      if r1.f <> r2.f:
        O(r1).cf = "MINOR"
      else:
        O(r1).cf = "NONE" //No conflict thus can be handled automatically

    //SAME SIGNAL,DIFFERENT FREQUENCY TYPES:
    //Set conflict of r1 and r2 to "MAJOR", requires manual resolution
    else if  $\exists r1.f \in \text{TimeBasedFrequency}$ 
       $\wedge \exists r2.f \in \text{InteruptBasedFrequency}$ 
       $\wedge \exists(r1.s = r2.s)$ : O:
      O(r1).cf = "MAJOR"
      O(r2).cf = "MAJOR"

  return O //Return list without WON'T values and with identified signal conflicts

```

## BIBLIOGRAPHY

- [1] R. N. Charette, "This car runs on code," *IEEE Spectrum*, vol. 46, no. 3, p. 3, 2009.
- [2] O. Predelli, *On-board Diagnose III*, Essen: Haus Der Technik, 2009.
- [3] W. Dieterle, "Mechatronic systems: Automotive applications and modern design methodologies," *Annual Reviews in Control*, vol. 29, no. 2, pp. 273-277, 2005.
- [4] ETAS, "Calibrating Automotive Electronics," ETAS, [Online]. Available: [https://www.etas.com/en/products/solutions\\_calibrating\\_automotive\\_electronics.php](https://www.etas.com/en/products/solutions_calibrating_automotive_electronics.php). [Accessed 25 May 2018].
- [5] ETAS, "INCA Software Products," ETAS GmbH, [Online]. Available: [https://www.etas.com/en/products/inca\\_software\\_products-measurement\\_calibration\\_diagnostics.php](https://www.etas.com/en/products/inca_software_products-measurement_calibration_diagnostics.php). [Accessed 26 06 2018].
- [6] M. Aldekhail, C. Azzedine and D. Ziani, "Software Requirements Conflict Identification: Review and Recommendations," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 10, pp. 326-355, 2016.
- [7] Z. Chentouf, "Managing OAM&P requirement conflicts," *Journal of King Saud University - Computer and Information Sciences*, vol. 26, no. 3, pp. 296-307, 2014.
- [8] M. Qiao, "The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review," Auckland University of Technology, Auckland, 2009.
- [9] D. Clegg and R. Barker, *Case Method Fast-Track: A Rad Approach*, Boston, MA, US: Addison-Wesley Longman Publishing Co., Inc. , 1994.
- [10] A. Hevner and S. Chatterjee, "Design Science Research in Information Systems," *Integrated Series in Information Systems*, vol. 28, no. 1, pp. 75 - 105, 2004.
- [11] K. Peffers, M. Rothenberger, T. Tuunanen and R. Vaezi, "Design Science Research Evaluation," in *DESRIST'12 Proceedings of the 7th international conference on Design Science Research in Information Systems: advances in theory and practice*, Las Vegas, NV, 2012.
- [12] E. Coffman, J. Csirik, G. Galambos, S. Martello and D. Vigo, *Bin Packing Approximation Algorithms: Survey and Classification*, New York, NY: Springer, 2013.