Universidade de Aveiro Departamento de Economia, Gestão e Engenharia e Industrial
Ano 2009

João Miguel Resende Marques

DESENVOLVIMENTO DO SOFTWARE DE SIMULAÇÃO "*GWB*"

"*GWB*" SIMULATION SOFTWARE DEVELOPMENT

Universidade de Aveiro Departamento de Economia, Gestão e Engenharia e Industrial
Ano 2009

João Miguel Resende Marques

DESENVOLVIMENTO DO SOFTWARE DE SIMULAÇÃO "*GWB*"

"*GWB*" SIMULATION SOFTWARE DEVELOPMENT

Relatório de Projecto apresentado à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia e Gestão Industrial, realizada sob a orientação científica do Doutor Henrique Diz, Professor Catedrático do Departamento de Economia, Gestão e Engenharia Industrial da Universidade de Aveiro.

Dedicado à minha familia, amigos e namorada pelo apoio demonstrado durante este percurso universitário.

o júri

presidente

Prof. Doutor Joaquim José Borges Gouveia
Professor Catedrático da Universidade de Aveiro

Prof. Doutor José António de Vasconcelos Ferreira
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor Henrique Manuel Morais Diz
Professor Catedrático da Universidade de Aveiro

agradecimentos

A elaboração deste projecto de mestrado só foi possível graças à colaboração de diversas pessoas a quem passo a agradecer:

Ao meu orientador Prof. Doutor Henrique Diz ao qual agradeço a disponibilidade na orientação do projecto.

Ao Dott. Ing. Marcello Bubba a oportunidade de integração na Robert Bosch – Belgium e por todo o conhecimento, informação, e carinho demonstrado durante o período de estágio. Também um palavra de apreço a todos os colaboradores do departamento de investigação e de engenharia pelo apoio prestado.

À Prof. Leonor Teixeira, Prof. Ana Moura e Prof. Ana Xambre pela orientação, sugestões e criticas efectuadas em momentos chave do desenvolvimento do projecto.

A todos os amigos espalhados pelo mundo, em especial ao Ing. Andreu Berna e ao Eng. Tiago Sampaio, pela ajuda e apoio durante a estadia na Bélgica.

À Doutora Diana Pereira pela espontaneidade revelada na prestação das suas competências profissionais assim como pelas sugestões e conselhos transmitidos.

Um especial obrigado a minha família e irmão por todo o esforço e compreensão.

palavras-chave

Software, ciclo de vida de um software, testes de software, validação, limpa pára-brisas, simulação

resumo

Nas últimas décadas, a evolução da indústria alcançou uma extrema importância na nossa vida pessoal e também no comportamento empresarial. O conjunto de ferramentas existentes nos computadores representam um papel fundamental na comunicação, nas estratégias, nas decisões, nos sistemas de análise de processos das nossas empresas, entre outras. Os software são normalmente concebidos para permitir aos os seres humanos realizar tarefas para as quais o cérebro humano não é capaz, tais como: manipulação de grandes quantidades de informação, realização de cálculos complexos, e controlar simultâneamente muitos processos.

Este projecto final foi desenvolvido exactamente sobre a disciplina de software. O presente documento mostra como uma empresa multinacional desenvolve internamente um novo software modular. Algumas técnicas serão investigadas e aplicadas a um problema real existente na empresa Robert Bosch-Bélgica, pertencente ao sector automóvel e principal responsável mundial pela produção de limpa pará-vidros. Aqui se encontrará uma sugestão para uma metodologia do ciclo de vida de um software e será explicado passo a passo todos os aspectos deste processo, desde a criação até ao desenvolvimento desta nova ferramenta de cálculo de apoio à decisão para o design. Esta metodologia foi aplicada à empresa permitindo assim a criação de um manual para o software de simulação chamado "*GWB*".
Mais precisamente, este projecto descreve a fase de testes, definido como a validação. Esta fase inclui um planeamento e execução de testes do software. Estes resultados foram analisados e comparados com as medições reais. Com base em conhecimentos anteriores, foi conseguido melhorar a precisão do software quer em parâmetros de produção quer em parâmetros de comportamento real dos limpa pará-brisas.

Com a criação do manual e do melhoramento do software foram alcançados os dois objectivos principais envolvidos neste estágio. Este trabalho contribuiu significativamente para o desenvolvimento do software de simulação da Robert Bosch, no entanto, é sugerido um conjunto de acções futuras. Estas têm como objectivo ajudar no desenvolvimento do "*GWB*" para uma implementação adequada no processo de produção de limpa pará-brisas.

keywords

Software, software life cycle, software testing, validation, wiper blades, simulation

abstract

In the last decades the evolution of software industry has reached an extreme importance in our personal daily life and also in the companies behavior. The existing sets of tools represent a vital role in our company's communications, strategies, decisions supports, systems and process analysis, among others.
Software is typically designed to enable humans to perform tasks which the human brain is not well capable, such as: handling large amounts of information, performing complex calculations, and controlling many simultaneous processes.

This final project was developed based under the subject software. The present document shows how a multinational enterprise develops internally a new modulate software. Some techniques will be investigated and applied to a real life problem existing in the successful *Robert Bosch-Belgium* company at the automobile industry world responsible for the production of wiper blades.
Here you will find a suggestion of a software life cycle methodology and an explanation step by step of all the aspects of this process from the creation to the development of a new calculation tool for design decision support. This methodology was applied to the company thus enabling to create a handbook for the simulation software called "*GWB"*.
More precisely this project describes the testing phase, defined as validation. This phase contains the planning and execution of software tests. These results were analyzed and compared with real measurements. Based on previous knowledge, was able to improve the accuracy of the software either in production parameters or on parameters of actual behavior.

With the creation of the manual and the software improvement two main objectives involved in this internship were accomplished. This work contributed significantly to the development of the Robert Bosch simulation software. However, it is suggested several future actions. To assist in the development of the "*GWB*" tool for a proper implementation in the wiper blade production process.

General Index

List of Figures

List of Tables

List of Definitions and Symbols

2D – Two Dimensions

3D – Tree Dimensions

Aerotwin – Commercial name for a *"GWB"*

AK – It comes from the German "äußeren Kreis". It means outer circle

ALK – Static contact force. It is the integral of the screen-rubber contact tip pressure at each normal cross section. The term "ALK" comes from the German term "AuflageKraft"

Bekaert – Supplier for material for S2 vertebras

CAE – Computer Aided Engineering

CAD – Computer Aided Design, software used for the geometry and kinematics definition of all components of the system

CDW – Supplier for the material to S1 vertebras

CFD – Computational Fluid Dynamics

CLP2 – Customer Orders Logistics and Material Planning

Cp – Capability process

ED-WS/EAB – Electronic Devices – Wiper System/ Advance wiper blade and arm engineering, Tienen

ED-WS/EAB1 – Electronic Devices – Wiper System/Engineering & Application wiper Blades – Platform development and application wiper blades

ED-WS/EAB2 - Electronic Devices – Wiper System/Engineering & Application wiper Blades – Platform development flat blades

ED-WS/EAB3 – Electronic Devices – Wiper System/Engineering & Application wiper Blades – Release planning & testing, Benchmarking

ED-WS/EAB4 – Electronic Devices – Wiper System/Engineering & Application wiper Blades, Research and development department

ED-WS/EAB5 – Electronic Devices – Wiper System/Engineering & Application wiper Blades, Application, series supervision

ED-WS/EAB6 – Electronic Devices – Wiper System/Engineering & Application wiper Blades – Rubber Development

ED-WS/ENG – Electronic Devices – Wiper System/Engineer Department in Germany

EPDM – Type of rubber

FEM – Finite Element Method

FeS – VVertebra, from the German term "Federschiene"

Fx – Type of rubber

"GWB" – From the German term "Gelenkfreies Wischblatt ", it is a wiper blade with a metallic vertebra ("FeS") having a continuous varying curvature, which behaves as a flexural spring

HSQ – High Speed Wiping Quality

H – Type of rubber

IC – Inner circle

IK – It comes from the German "Innenkreis". It means inner circle

IEEE – Institute of Electrical and Electronics Engineers

ISO – International Organization for Standardization

L – Total length of the vertebra

MOE5 – Production Department Aerotwin Wiperblades

Mycrona – Multi-Sensor Measurement Systems

OC – Outer circle

OEM – Original Equipment Manufacturer

RBBE – Robert Bosch-Belgium

RBBE/QMM – Quality management department

RBBE/PUQ – Purchasing quality control

SL – Side Lock: type of adapter

SPC – Statistical process control

TEF 1.3 – Technical Engineering Function Department

TL – Top Lock: type of adapter

WSQ – Wiping quality

1. Introduction

The present document was developed during the 8 months internship made in Robert Bosch – Tienen, Belgium under the Master's degree in Industrial Engineering and Management of Aveiro University. The plant where the author was insert is the worldwide responsible for the research and production of wiper blades in the automotive industry.
In the current chapter will be describe, first in section 1.1, the two main project objectives which concerns literature research for future application as well as a more practical activity perform during the software development process. In section 1.2, it's explained the structure of the whole project aiming to guide easily the reading of this document. Here it will be also found short chapters descriptions.

1.1 Project Objectives

The internship and the project developed had two different purposes. The first, aim was to establish the structure of the handbook to the specific simulation software for different user locations. This research required involves software[1] engineering discipline as well as software life cycle method. The present document will also be describe and analyze the process of software development as well as the methodology applied in the software tests.
The second was carrying out the validation test of wiper blade simulation software that was in a critical point of its development. This mission was to determine appropriated values for a set of parameters as well as to discover design tolerance values for several attributes into the wiper blades behavior, to implement them in the wiper blade simulation software.

[1] Consist of not just code in machine-readable form but also the documentation that is an intrinsic component of every project. (Schach, 2008)

1.2 Project Organization

Besides this chapter, the project is divided into three more relevant parts.

In chapter 2, some literature about the approach to be followed about software development is analyzed. In this chapter is summarized and presented the appearance of software engineering and its consequent development. During the study of this issue was discovered the existence of models to aid software development. Therefore, it is studied and suggested a standard software development model that has similarities between the other models. Here are demonstrated some of the more normal dilemmas regarding this type of project. Also in this chapter, a review is made on the subject of test engineering describing the main grounds and procedures of this subject.

The other important part of this document is chapter 3. This chapter refers to the practical performed during the internship concerning "*GWB*" simulation software development. After presented why the software is required, it's described the need for test validation and it's displayed the test plan that was followed in order to monitor the accuracy of the results obtained by the software prototype. Further, is shown the consequent results and analyzed in accordance with existing assumptions from past. There will be demonstrated conclusions and default parameter values as well as design tolerance values will be suggested to include in the software prototype code.

In the end of the document (chapter 4), are explained the main conclusions of the project and are shown solutions for the future development of the project.

2. Literature review

The following work was developed based on the knowledge's acquired under the subject of Software Engineering. This discipline aims to the "*production of a fault*[2]*-free software, delivered on time and within the budget, that satisfies the client's*[3] *needs*", affirms Schach (2008).
As a result, the development of software has to be explored. In this way, we intent to explore the software development from creation of the concept until the final end of the process. During the course, the product goes through a series of steps that will be stated further. The software development needs tools and techniques to design, maintain or test it, and requires high specialized software professionals. According to Schach (2008), "*the description of these steps that should be performed when building a software product*" is what we call software life cycle. In this chapter will be elucidate some basic definitions concerning the software life cycle.
In our technological world, several software with different purposes are being used by companies thus the correct software creation is nowadays a way to generate competitive advantages towards our competitors. To ensure the quality all the software products undergoes tests. Software testing is also a discipline of software engineering, therefore will also be examined fundamental rules, principles and processes that should be taken into account when performing software testing. On this, validation and verification play an important and fundamental role and as such will be subject to investigation.

2.1 History

Despite many software success stories, an unacceptable large proportion of software products are still being delivered late, over budget, and with residual faults. With the ambition of being at the same level of the other engineering's disciplines, in 1967 NATO study group invented the term "*Software Engineering*". The demand that building software is similar to other engineering tasks was endorsed in the 1968 NATO Software Engineering Conference held in Garmisch, Germany. "*A conclusion of the conferees was that the software engineering should use the philosophies and paradigms of the*

[2] A fault in the code is the consequence of a mistake (Schach, 2008).
[3] The individual who wants a product to be built (Schach, 2008).

established engineering disciplines to solve what they termed the software crisis, namely, that the quality of software generally was unacceptably low and that deadlines and budgets were not being met" alleges Schach (2008).
There is a study made by the Standish Group research that says 9236 software development projects completed in 2004 shows that exist problems to create successfully software's.

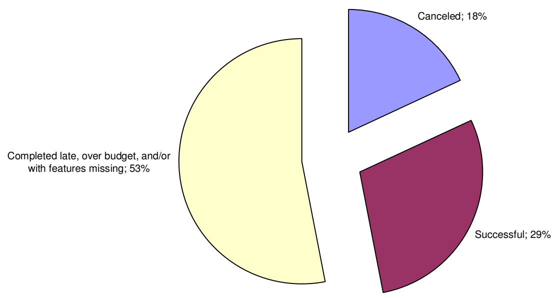


Figure 1: "The outcomes of over 9000 development projects completed in 2004" (Schach, 2008).

It's clear (Figure 1) that only a small percentage of software development is delivered on time, within budget, fault free, and meeting its client's needs.

2.2 Software life cycle

The software life cycle is a group of steps that we should follow when building a software product, but with the developments made so far several software life cycle models were generated. The difference between the software life cycle models existing consists of the number of phases within each one. Mainly these software life cycle phases may not be carried out by all the companies exactly as specified in the following model for a couple of reasons: as when time and costs are taken into account by the project manager[4]. These phases, or the name of the phases, may also change from one organization to another, but the following phases are close enough to most practices.

[4] An experienced professional who accepts responsibility for planning, monitoring, and controlling projects with respect to schedule, budget, deliverables, customer satisfaction, technical standards, and system quality (Whitten *et al.,* 2009)

2.2.1 Software life cycle phases

In the next six sections of the present chapter it will be presented the main definitions and objectives for each software life cycle phases. In generally, the several software life cycle models pass through the following six phases that we describe below:

2.2.1.1 Requirement phase

The software development process normally starts when a client appears in a software development organization with a problem. According with Schach (2008), "*the requirements workflow is for development organizations to determine the client's needs*", for Abran and Moore (2004) "*software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem*". Thus, is established the project scope and goals, which means that the domain[5] and the objective of the project should be defined. The relationship between the client and the developers[6] in this phase can be crucial to the success of the software, because the "*client's description of the desired product may be vague, unreasonable, contradictory, or simply impossible to achieve*", alleges Schach (2008). Thus, the developer's task is to detect exactly what the client needs are and to find out what constraints exist in it. The following arguments are admitted by Schach (2008) as the main constraints:

- Deadline: The client may desire a delivery deadline that is difficult to or even impossible to meet. Not carrying out the deadline can and will probably have costs associated;
- Reliability: The author referenced defines reliability as "*how often the product fails […] and how bad the effects of that failure[7] can be*". The author expose the following example: the client requires that the software must be 99% of the time operational or the mean time between failures must be at least 4 months. Another way to describe reliability is as a variety of other constraints request by the client;
- Cost: This constrain is always a big problem between the client and the developer. What happens is that after the specifications are completed, the client asks the developer to name their price for completing the project. If the client accepts the

[5] The specific environment in which the target software product is to operate (Schach, 2008)
[6] Are the members of a team responsible for building a software product (Schach, 2008)
[7] The observed incorrect behavior of the software product as consequence of the fault (Schach, 2008)

price the development process will continue. If not, the process it's probably followed by negotiations between them.

A large percentage of software product requirements can carry out inadequately. This may happen because the "*client may not truly understand what is going on in his or her own organization*" affirms Schach (2008). This author also affirms that "*what normally happens is that the client frequently asks for the wrong product*" due to his low literate computer skills.

2.2.1.2 Analysis phase

Schach (2008) claims that "*the aim of the analysis workflow is to analyze and refine the requirements to achieve the detailed understanding of the requirements essential for developing a software product correctly and maintaining it easily*".
In this phase, the architecture of the product is settled. Therefore, the same author suggests the product is divided in smaller portions relatively independent and each with its own facts and applications. These decomposed components are developed repeatedly as required until the objective and goal defined in the requirements phase are achieved. The language used at the moment should be clear to the client and to the developer. Thus should be explained in a natural (human) language to better client understanding.

After finishing the decomposed component and done the inspection of the requirements, a software project management plan (SPMP) must be draw up. Schach (2008) admits that the "*SPMP reflects the separate workflows of the development process and shows which members of the development organization are involved in each task, as well as the deadline for completing each task*". This plan can be different from one organization another and includes high or low detailed views of the project.
The analysis phase may reveal the need to revise the business scope or project goals. As result, the requirements should be update or adjust to the real need of the client.

2.2.1.3 Design Phase

The design phase can be described as a technical set due to this fact it should show how the product "is to do it". Actually the aim of the design workflow is to "*refine artifacts*[8] *of the analysis workflow until the material is in a form that can be implemented by the programmers*" alleges Schach (2008). In this process, the design team decides the internal structure and organization of the product.

Still remains two important items to be considered in this phase. The first has to do with the proper selection of the algorithms used in the model. The next has to do with the need to report constantly the design decisions made by the design team. Schach (2008) defend that this occurs for two main reasons:

- Making a continuous record of the decisions can prevent the design team of forgetting all the process and it becomes easier to backtrack the code in case of having to update or correct it;
- A good documented code simplifies future improvements (postdelivery maintenance). In the ideal scenario which in practice is difficult to achieve, "*the design of the product should be open-ended, meaning that the future enhancements can be done by adding new classes or replacing existing classes without affecting the design as a whole*" according to Schach (2008). Thus, the design team has to build up a structure that can be extended in different ways without the need of total redesign.

2.2.1.4 Implementation phase

Schach (2008) affirms that during the implementation phase, the software will be configured and installed (integration). Whitten *et al.* (2009), agrees with this statement and adds that all software components to be constructed and installed must be also tested as well as the whole software to guarantee that the software meets the client's requirements and specifications (acceptance testing).

Later, the implementation must be processed. In case of exist previous software data, it must be converted and introduced into the new software product.

Whitten *et al.* (2009) alleges also that to complete the correct implementation of software products into companies, a transition plan should be execute from the older to the new

---

[8] Is a constituent component of a software product (Schach, 2008)

software product and that this plan must include training to properly use the system by end-user[9].

2.2.1.5 Postdelivery maintenance phase

The definition of maintenance has change with times. In the past, the software product developments were viewed as composed in two separated parts, says Schach (2008). This first part is the development and the second part is the maintenance. By IEEE definition, [IEEE 610.12, 1990], any change to the software after installation on the client's computer and acceptance by the client constitutes maintenance, whether to fix a residual fault or extend the functionality. Schach (2008) alleges that this concept is named by development-then-maintenance.

What happens in this approach is that for the identical fault depending on the time it has been correct, before or after installation, it constituted develop or maintenance, respectively. Thus, there is no difference whatsoever between the two activities, but they were considered different things. For this reason this model now is considered unrealistic.

Following the [ISO/IEC 12207], 1995 definition, "*maintenance is the process that occurs when software undergoes modifications to code and associated documentation due to a problem or the need for improvement or adaptation*" before or after installation of the software product. The IEEE later on adopted this definition of maintenance.

According to Schach (2008) postdelivery maintenance can be defined as any change to the software after it has been delivered and installed on the client's computer referring to [IEEE 610.12, 1990]. Thus, for this author postdelivery maintenance is a subset of maintenance.

The already implemented software product is rarely perfect. After the installation of the software product the end-user will find defect[10] or will discover requirements, design and implementation flaws. Therefore there is the need of continuous improvement of any software product until it becomes as much perfect as it can.

Schach (2008) defends that a good software product undergoes definitely postdelivery maintenance and only a bad software product doesn't suffer this process. This occurs,

---

[9] Is the person or persons on whose behalf the client has commissioned the product and who will utilize the software (Schach, 2008)

[10] Is a generic term that refers to a fault, failure, or error (Schach, 2008)

essentially, because a software product is a model that reflects the real world that insists to change constantly.

It is now accepted that postdelivery maintenance is vital aspect of the software life-cycle process. And the following studies (Figure 2, Figure 3) of software product cost resume that fact (study in Schach (2008)).

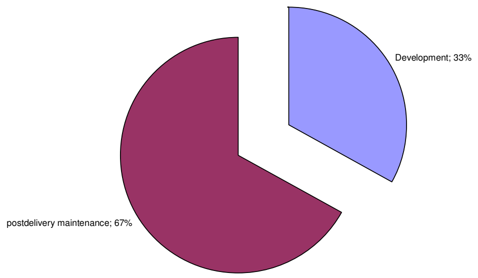


Figure 2: "Approximate average cost percentages of development and postdelivery maintenance between 1976 and 1981" (Schach, 2008).

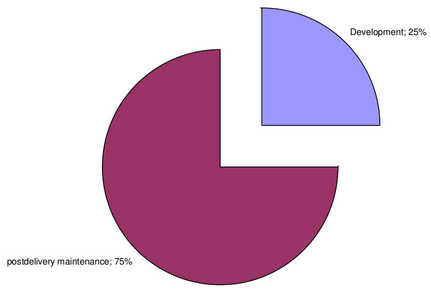


Figure 3: "Approximate average cost percentages of development and postdelivery maintenance between 1992 and 1998" (Schach, 2008).

The conclusion of this study (Figure 2, Figure 3) is that more money is spent on the postdelivery maintenance that on all other software life-cycle phases. Thus, should be carried out efforts to minimize the high cost associated to this phase.

The main problem that occurs in the software product development is the lack of documentation; refer already in the design phase. The lack of documentation is exactly what generates this high cost associated.

Thus, in order to improve or correct any software code or specifications in the requirements, analysis or design phase it should be ensured that in all phases there is documentation of support.

Most project manager prefers to deliver the software product on time than wasting the developer's time on documentation. This decision sooner or later it will be recognized as wrong, bringing extremely high cost that normally they can't support easily.

2.2.1.6 Retirement

This final phase of the software life-cycle occurs when after years of postdelivery maintenance the software product no longer is cost effective.

Schach (2008) present reasons that can justify the decision of the project manager of building a new software product:

- The proposed changes to the software are so extreme that the total redesign is more expensive that building a new software product from scratch;
- There already many changes to the software product that even a small change of a component can have a drastic effect on the functionality of the product as a whole;
- The lack of documentation existing increases the risk of future faults;
- The hardware on which the product runs is to be replaced.

2.2.2 No separate phase for planning, testing and documentation

In this chapter will be described why it shouldn't exist an independent planning, testing and documentation phase during software development projects.

2.2.2.1 No planning phase

In such a complex project as software development is very complicated to have planning phase at the very beginning of the project. Until its known all the details to be develop it can't drawn up an accurate detailed plan.

Therefore in the software development process there are three activities concerning the project planning according to Schach (2008):

- Initially a preliminary planning takes place for managing the requirement and analysis phase;
- As said before, in the analysis phase should be drawn up the software project management plan (SPMP). This must be as much detailed as it can and including budget, staffing requirements, and schedule;
- In the meanwhile as the project goes by all the life-cycle phases, the SPMP must be monitor to ensure that any deviation is updated.

In conclusion, the planning of such a product should be always done in parallel to the developing of the product.

2.2.2.2 No testing phase

The reason for the absence of a testing phase is that checking the software product when it is delivered and installed loses the specificity about the error[11]. This means that check after delivered is too late. To justify it we can see the next example: if there is a fault in the requirement, this fault will have been carried forward into the design.
But there are times when the testing process is done exhaustively. The authors Vermesan and Coenen (1999), Schach (2008), and Sokolowski and Banks (2009), all concord that the testing process occurs more often in the end of each phase, verification, and mainly done before the product is delivered and installed on the client's computer, validation. These two issues will be studied futher on in chapter 2.3.3.

To ensure that the delivered software product meets the client needs and that the product has been built correctly in every way, all software developing organizations should contain an independent group responsible for it. This group is called software quality assurance (SQA) group.

2.2.2.3 No documentation phase

As refer already in some chapters before, the documentation of a software development product must be always complete, correct and up to date. Thus, it is impossible that should be a separate documentation phase.
According to Schach (2008), the reasons for this fact can be enumerate in the following list:

---

[11] Is the amount by which a result is incorrect (Schach, 2008)

- There is a constant turnover in the software personal so only with a good documentation the project can survive to the changes;
- Is almost impossible to develop a specific phase unless the previous one is complete, correct and up to date documented;
- Unless there is documentation to show how a software product is supposed to behave, it's impossible to perform tests to check if the product is working correctly;
- To perform an accurate maintenance it's necessary to have documentation that describes precisely what the current version of the product does.

These three activities (planning, testing and documentation phases) must be done continuously over the project to ensure that all the information is correct and up to date. These processes will facilitate following the changes of the development over the time.

2.3 Software Testing

In order to help performing test and validation phases within project objectives, was also reviewed some literature concerning this subject. Thus in the following section of this chapter, it will be described concepts, principles and objectives of software testing. The complete understanding of these issues allows carrying out a higher quality software testing.

2.3.1 Assumptions

The quality assessment process is responsible for ensure the performance of a software product and this process is related with the idea of quality checking, tests and metrics. However, Cardeñosa J., in Vermesan and Coenen (1999), affirms that the validation and verification process are associated to quality assessment but as well as to software development and that those process are in charge of controlling the early phases.

In the IEEE standards concerning quality assurance is explicit that there must be a validation and verification process but almost always are applied to a product and not to prototypes alleges Cardeñosa, J. in Vermesan and Coenen (1999).

In order to clarify, this project doesn't concern to quality assurance because the validation and verification processes performed were related to a software prototype.

Following this concept will be presented a review of concepts and definitions of software testing and in particular the validation and verification processes.

2.3.2 Software fundamentals

Abran and Moore (2004) affirm that software engineers can perform tests in different ways. For example, they can test only a module of the software, or group of them or they can test the whole system. Each of these procedures is called levels and is a fact that none of them is more important than the other.

The same authors define these stages as unit, integration and system testing, respectively. The first level verifies the functioning in isolation of software pieces which are separately testable and its process is specify in the IEEE standards for software unit testing. The integration testing verifies the relation between different modules of the software while the third level is regard to the behavior of the entire system.

The validation and verification phases are critical elements of the software development process and they represent the ultimate review of specification, design, and code generation.

Before executing the testing, the software engineers must be sensible to a number of software testing principles. Pressman (2001) and Abran and Moore (2004) define the following principles:

- Test selection criteria – the existence of a selection criterion is a means of deciding whether a set of test cases is adequate;
- Testing effectiveness – the effectiveness of a test set can only be evaluated according to their objective;
- Testing for defect identification – during the tests for the identification of specific defects, only the test that causes the system to fail is considered successful. This is quite different from testing to demonstrate that the software meets its specifications, or other desired properties, in which case testing is successful if no failures are observed;
- The oracle problem – An oracle is any (human or mechanical) agent which decides if the program is operating correctly in a given test. The automation of an oracle can be difficult and expensive;
- Theoretical and practical limitations of testing – performing tests can only expose bugs but never expose their omission;

- The problem of infeasible paths – Infeasible paths are a significant problem in path-oriented testing because they cannot be exercised by any input data;
- Testability – Both authors agree that testability can have more than one definition. They think that testability means how adequately a particular set of tests will cover the product. But Abran and Moore (2004) assert that testability also means that if the software is defective, probably the test will detect the failure, on the other hand Pressman (2001) testability is also defined as how easily a tool can be checked and repaired. This author also refers that testability occurs as result of a good design. In conclusion testability represents the capability of the tool to perform a good test.

Accordantly with Pressman (2001), the objective of testing software is related to find or not find errors in the software. More precisely, this means the objectives can follow the next set of rules:

- Testing is a process of executing a program with the intent of finding an error;
- A good test case is one that has a high probability of finding an as-yet-undiscovered error;
- A successful test is one that uncovers an as-yet-undiscovered error.

This perspective of the testing objectives is not accepted by other authors as being too restricted. Abran and Moore (2004) defend that testing is conducted considering a specific objective and with varying degrees of precision. Thus, testing can be aimed at verifying different properties. Test cases can be designed to check that the functional specifications are correctly implemented as well as to check others test properties as performance, reliability, and usability, among many others.
Therefore, this last authors identity the next set of potential test objectives:

- Acceptance/qualification testing: this test checks the system behavior against the customer's requirements;
- Installation testing: the software may be check upon installation in the target environment;
- Alpha and beta testing: a trial version can be given to a set of potencial end-users to perform a testing process;
- Conformance testing/Functional testing/Correctness testing: conformance testing is attempt at validating the behavior of the tested software conforms to its specifications;

- Reliability achievement and evaluation: this test by helping to identify the faults are a way to increase reliability;
- Regression testing: according to [IEEE610.12-90], regression testing is the "*selective retesting of a system or component to verify that modifications have not caused unintended effects*". This testing can be perform at each of the test levels;
- Performance testing: as example, these tests have the goal to verify the capacity and response time;
- Stress testing: these stress testing takes a software to its maximum design load;
- Back-to-back testing: the same test is analyzed in two implemented versions in a software product, and the results are compared;
- Recovery testing: Recovery testing can be defined as the capability of a software to restart after a disaster;
- Configuration testing: configuration testing verifies the software under various configurations concerning the type of users;
- Usability testing: is the assessment of learning and ease of handling software for the end-users;
- Test-driven development: Test-driven development is an independent check that the software has correctly implemented the requirements.

Accordantly to Abran and Moore (2004) the process of testing includes different concepts, strategies, techniques in order to guide carefully the testing teams since the planinng towards its termination.

This series of actions are managed by people thus they should be receptive to failures during this process. Software managements should be able to organized correctly all tools, techniques and software profissionals by using them to plan, documented, perform and complete the testing process in a cost effective way.

Several rules can be described about the activities of software testing. According to Abran e Moore (2004) these rules relate to the planning that must take into account the enviroment, the time, the effort and the tools available. The plan drawn at this time will also consider the level of testing and the consequent choice of test cases.

Regarding the implementation and evaluation of testing should always be performed and documented as simple and clear as possible allowing anyone to understand and follow the process without being a software expert.

Although implicit, any error or fault found in testing should be documented, analyzed and corrected.

2.3.3 Validation and verification

Validation and verification are specific testing processes. Therefore, before plan and performed these process, all the software testing principles and fundamentals should be assimilated by the software project manager.
It´s also important to refer once more that these two process, on one hand, are "*an essential part of the global quality process of the organization*" (Vermesan and Coenen (1999)), and on the other hand, the same author defend, that they should be adopted in a "*development methodology*". This thought is also reported by Sokolowski and Banks (2009) when they say that "*these terms* [validation and verification] *have meaning in a general quality management context as well as in the specific modeling and simulation context*".
Different definitions were found while researching literature. But the authors followed will report submitted sufficiently general opinion.

As said in 2.2.2.2, verification process should be performed "*during the different phases of the system development so as to check the absence of errors*" alleges Cardeñosa, J. in Vermesan and Coenen (1999). Further, this author defines verification as "*the set of activities aiming at checking that the system (products and sub-products) is adapted to the system requirements*". This view is shared by Sokolowski and Banks (2009) when they say that, in modeling and simulation point of view, verification is "*the process of determining if an implemented model is consistent with its specification*". The IEEE describes verification as "*the process of evaluating a system to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase*".
Thus, verification should be able to answer questions as:

- Does the generated code implements the conceptual model?
- Does the objectives of the model are achieved by the conceptual model?
- Does the executable model present results when it is needed and in the desired format?

Referring the some authors of the previous chapter, validation *"is the process of determining the degree to which the model is an accurate representation of the*

simuland[12]" (Sokolowski and Banks, 2009). And by IEEE 1990, validation "*is defined as the process of evaluating a system to determine whether it satisfies the specified requirements*".
The main objective of this phase is being capable of solving the next questions:

- Is the conceptual model a description of the simuland?
- How similar are the results produced by the executable model to the simuland behaviuor?
- What degree of confidence that the results have?

Already apprehended the definitions and objectives of verification and validation, and when they should be performed, its important now to mention how we should do it. In the research made it was discovered that there are many methods. These varieties of methods are implicitly related to the amount of different simulation projects available.
A set of methods and techniques were chosen and will be listed and described next. According to Sokolowski and Banks (2009) the methods are grouped in four categories: informal, static, dynamic, and formal.

- Informal

The author identifies informal verification and validation methods as more qualitative than quantitative. Therefore, this analysis is composed of a few mathematical elements are then preferably based on subjective assessments. Sokolowski and Banks (2009) also classify two informal methods:
The first one is the inspection and can be defined as a method of comparison the project artifacts. Here it's organized teams of developers and testers that examine manually the artifacts. Normally the teams are divided by specific roles and based on their knowledge and experience it's identified, assessed and registered potential faults.
The second method is called face validation. The same author refers face validation as a method "*that compares simuland behavior to model result*". The observers may be potential users or subject experts and their process is to "*compare the behavior of the simuland as reflected in the simulation results with their knowledge of the behavior of the actual simuland under the same conditions, and judge whether the former is acceptably accurate*" (Sokolowski and Banks, 2009)

[12] It is the object, process, or phenomenon to be simulated. (Sokolowski and Banks, 2009)

- Static

Citing Sokolowski and Banks (2009) "*static verification and validations methods involve assessment of the model's accuracy on the basis of characteristics of the model and executable model that can be determined without execution of a simulation*". This method is usually performed by developers and other technical experts, thus, while using this method, it is common to examine the programming language code.
Other two examples of methods are presented: First, "*data analysis is a verification method that compares data definitions and operations in the conceptual model to those in the executable model*" (Sokolowski and Banks, 2009).
The second example is the cause-effect graphing. This validation method compares the connection between cause and effect in the simuland and in the conceptual model.

- Dynamic

As the name implies, this method assess the effectiveness of the results by executing the program model. This test method involve, normally, numerical results and data, therefore, it's an objective and quantitative method. The same author also enumerates two dynamic methods: sensitivity analysis and predictive validation.
Sokolowski and Banks (2009) define sensitivity analysis as "*a validation method that compares magnitude and variability behavior to magnitude and variability in the model results*". Sargent (2007) also calls to this method: "*parameter variability*". And he describes it as method that "*consists of changing the values of the input and internal parameters of a model to determine the effect upon the model's behavior or output*". The authors then add that the variability shown in the model results should be close to the ones seen in the simuland, and if not it should be corrected to do it.
Both authors allege that the second method is a method of forecasting. The outcomes of the models are direct compared with the real simuland behavior. To perform this method is necessary to have already historical and experimental data.

- Formal

"*Formal verification and validation methods employ mathematical proofs of correctness to establish model characteristics*" allege Sokolowski and Banks (2009). This author affirms that aplly in pratice this method is quite difficult due to the fact that the models are very complex. However, two examples of formal methods are inductive asserttions and predicate calculus.

The first, is described as a method that compares the programming language code of the model to descritions of the simuland. Thus this verification method use mainly mathematic induction.

The predicate calculus is a validation and comparison method. "*One procedure of predicate calculus is the proving of arguments, which can demonstrate that one set of properties of the object in question, if true, together imply additional proprities. The goal of the method is that by describing properties of the simuland and conceptual model using predicate calculus, it can be possible to prove that the two are consistent*." (Sokolowski and Banks, 2009)

3. Project

This chapter is reserved for the most practical part carried out during the internship. Here will be explained throughout the journey of the testing stage. It can be found here a description of the objectives of the tests as well as a detailed explanation of the reasons for the development of this software. Later will be presented practical results, the conclusions and corrections at the end of the testing process.

3.1 Objectives and explanations

The internship made and the consequent project done for the Robert Bosch – Belgium was divided into two main objectives. A important contribute to this work was the 6 months task as wipeability simulation assessement and wiper blade theoritical design specialist for comercial cars in colaboration with application engineering department.
During the first six months of work, the author of this document was required to carried out research on software developement in order to structure the *"GWB"* software handbook. Thus, after debating with the project manager, it was decided and put into practice the new organization of the handbook. This task, involved research of a variety of internal and external documents developed by different professionals and departments, and, in consequence, was correctly compiled in the *"GWB"* handbook. As note, some of the chapters of the handbook are used further is this document in order to help explaining the test environment.

The second purpose, in collaboration with the company, had to do with a more practical component, carrying out the validation phase of the "*GWB*" simulation software prototype.
It will be presented how the development team organized this process to achieve its objectives. The plan contained three essential functions to be insightful: - it started by analysing the vertebra bending shape of the wiper blades - then it was analyzed the wiper blade contact force simulation by comparison to the values measured experimentally. Last, it was studied the production parameters tolerance as well as the behaviour parameter tolerances. With the analysis of these three components, is intended to obtain a set of default values in order to be placed on the "*GWB*" simulation software prototype code.

3.2 Introducution to "*GWB"* software

The "*GWB"* tools software has been created with purpose of improving the quality and efficiency of the wipeability assessment and vertebra shape design simulation process for Aerotwin wiper blades (*"GWB"*) design. The tool has been developed with a multidisciplinary simulation approach to ensure robust design, extended static and dynamic properties and "*modularized*" to ensure its differentiated usage for advanced user location, low cost user locations and production user locations.

The modulization of the new software is one of the important changes from the current tools. This aspect of "*GWB"* tools allows any user to have access to a diverse of modules: data management, screen analysis, vertebra robust shape design for wipeability, extended vibration assessment and wipeability check of existing beams. This software is a friendly user tool so no advanced engineering knowledge is required.

"GWB" tools was developed in a way that the main production process parameters and operative field parameters will be statistically taken into account into *"GWB"* design procedure (Bubba, 2008f). It means that the linearization of the *"GWB"* tools uses the previously knowledge of the production procedures to achieve a robust design (see Figure 21).

The future software will have modules as show in the Figure 4.

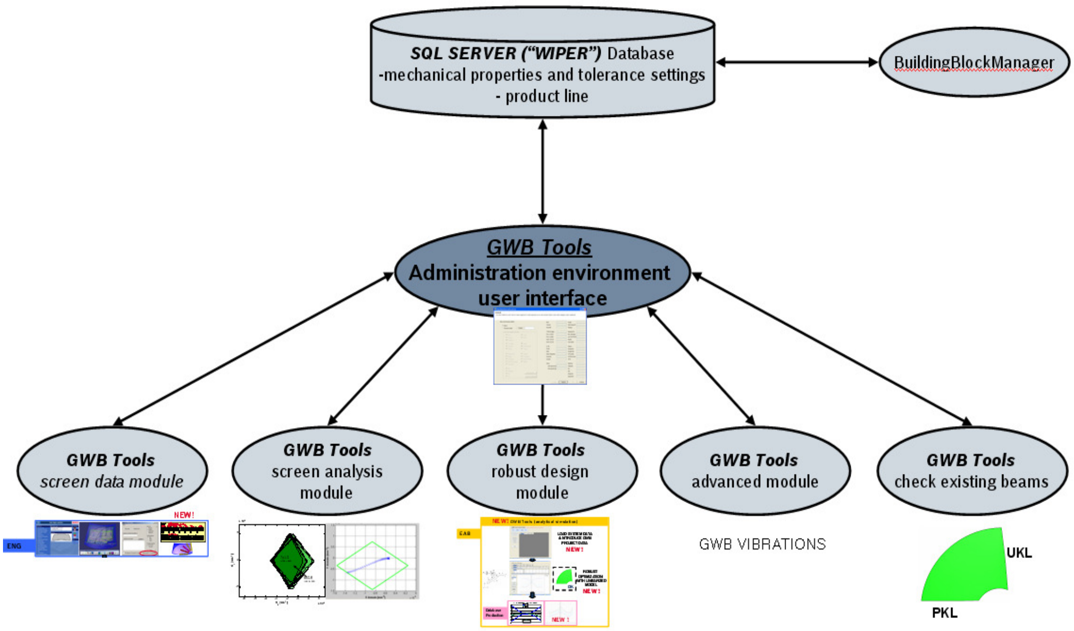


Figure 4: "GWB" tools workflow overview

As a multifunction tool, the *"GWB"* tools incorporate different areas of study as we enumerate and see in Figure 5:

- Wiper system simulation: enhance data exchange with system development engineering ED-WS/ENG according to ED-WS/ENG standards CAE wiper systems (Bubba, 2008f);
- Production process and operative conditions: main production process parameters and operative field parameters will be statistically taken into account into *"GWB"* design procedure, and process and field simulations parameters will be fitted accordingly to production and testing control procedures (Bubba, 2008f);
- 2D FEM static: *"GWB"* analytical simulation happens throughout the rubber profile static (steady state) properties: ALK (distance, attack angle), α2 (distance, attack angle), contact point (distance, attack angle) (Bubba, 2008f);
- 2D FEM dynamic: integration of impedance rubber characteristics from Flip-Over Noise data for *"GWB"* vibration and transient analyses (Bubba, 2008f);
- 3D FEM static model interface: improvement interface exchange data between wiper analytical simulation and FEM 3D (Bubba, 2008f);
- CFD aerodynamics loads: improvement interface exchange data between fluent output and analytical simulation program (Bubba, 2008f).

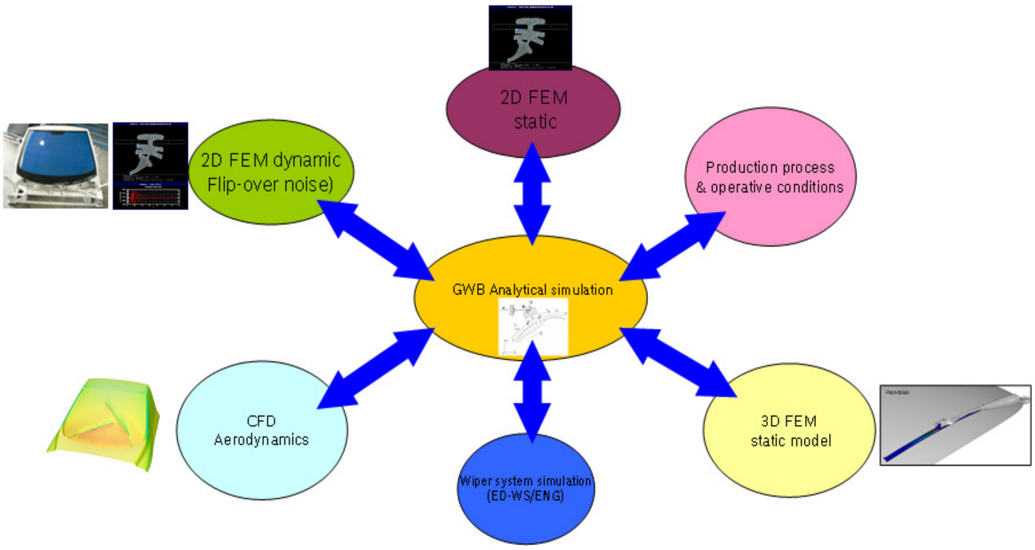


Figure 5: Goal of the "GWB" Tools software: Integration and enhancement of multidisciplinary simulation with focus on analytical simulation (Bubba, 2008f).

Current status analytical simulation:

1. Integration of screen analysis:

- OEM and internal customers receive radius ranges thresholds for screen design and the first wipeability assessments;
- An alternative wipeability prediction method (4th order) has been developed and implemented into a prototype algorithm. The method can be considered as a valid tool for the first approximation wipeability assessments;

2. Integration of “Robust design”:

- A prototype algorithm has been developed for the integration of “robust design” procedure into the current *“GWB”* analytical algorithm. A comprehensive experimental validation has to be done before its implementation into organization.

3. Integration of attack angle and torsion:

- The current *“GWB”* analytical algorithm is not able to take attack angles and *“GWB”* torsion into account. A new screen macro with screen attack angle information has been developed with the support of ED-WS/ENG2. New rubber profile characteristics interface has been developed. A first prototype algorithm able to consider attack angles (but not yet torsion) in steady state simulations is in advanced state of development.

4. Integration of vibration and transient simulation into current 2D static algorithm:

- The current *“GWB”* analytical algorithm is not able to take vibration and transient behaviour into account. 2D vibration equations are in advanced state of development.

5. *“GWB”* software modulization:

- The current *“GWB”* analytical algorithm is not “modular”: it is only meant for advanced users with specialist knowledge.

As the first stage of the life cycle of the software, it is then presented the requirements of the software

Requirements:

1. Necessity for screen analysis and wipeability criteria:

- OEM customers and windscreen suppliers need windscreen tolerance methods for screen design and wipeability assessments at early stage of development;
- Internal customers need simplified windscreen geometrical tolerance methods to cope with wipeability assessment.

2. Necessity for "robust design" procedure:

- Internal customer requires simulation methodologies able to identify the effects of production tolerances onto Aerotwin functionality (example: theoretical effect of adapter "clamping angle" comes from ALK project);
- OEM customers require suppliers with robust production processes which cannot be fulfilled without a profound theoretical comprehension;
- The request of OEM customers for screen tolerances methodologies for their screen design procedures has become stronger. It is more and more clear that those companies able to fulfil this requirement will get a competitive advantage;
- OEM customers requires short development time, this cannot be reached throughout a robust design procedure which allows the lowest testing activity.

3. Necessity for enhanced simulation accuracy:

- The simulation of customer's specification at different environmental conditions (example: different temperatures, grade of rubber ageing, different friction conditions) cannot accurately be always guaranteed with appropriate accuracy by the current analytical *"GWB"* simulation algorithm. This is mainly due to the fact that attack angles and *"GWB"* torsion are not taken into account into the current algorithm. The lack of attack angle and torsion mechanism into simulation causes inaccuracies (comparison to 3D FEM model) which become not negligible in combination with higher rubber profile stiffness (example: at low temperatures and low friction).

4. Necessity for vibration and transient simulation in analytical algorithm:

- Although wiper system is subjected to many dynamic loads, the knowledge about vibration and stability under lateral and torsional friction loads is weak. In case of customer's specific demands, "quick" solutions have to be found (example of investigation of "fish tailing" effect in multi-body software), with risk of not being able to deeply understand phenomena;
- The development of an analytical formulation offers to the possibility to better understand the phenomena and to achieve very low computational time.

5. Necessity for *"GWB"* software modulization:

- In the next years Bosch wiper blade development activities will be more and more extended to different business locations: Belgium, Hungary, China, Korea. Each business location will differ in the typology of applications (economy/ business/ executive/ premium) therefore there is no need (it would be even not justifiable from the business point of view) that each business location develops the same level of engineering specialization and performs the same level of simulation activities. Therefore there is a need to customize ("modulization") the *"GWB"* simulation capabilities in function of the typology of the end user.

In order to ensure a better and organized development of the client requirements, it's applied the second phase of the software life cycle: analysis phase. It was possible to sub-cluster these disciplines into more specific activities. Each of them was or is being developed by different departments and specialists. The following explanations clarify precisely what should be done about each discipline:

Analysis phase:

1. Integration with system design:

- Enhance data exchange with system development engineering ED-WS/ENG, according to ED-WS/ENG standards CAE Wiper systems;
- Include attack angles and arm load data in standard exchange data procedure;
- Include arm load variation and system tolerance data.

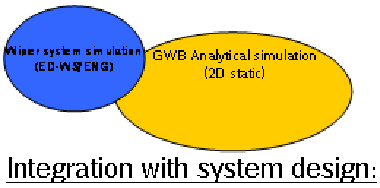


Figure 6: "GWB" analytical simulation integration with system design

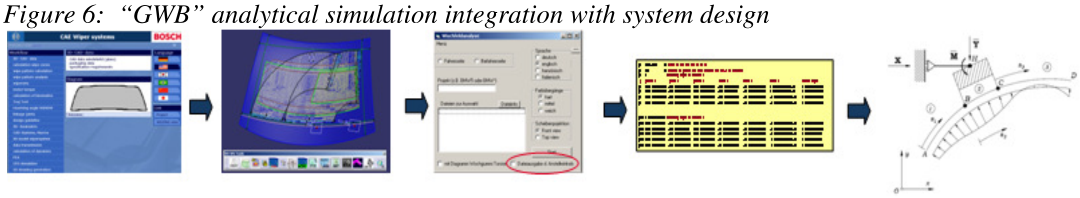


Figure 7: Example of the "GWB" analytical simulation integration with system design

2. Integration with production process and operative field conditions:

- Main production process parameters and main operative field parameters will be statistically taken into account into *"GWB"* design procedure. Process and field simulation parameters will be fitted accordingly to production and testing control procedures.

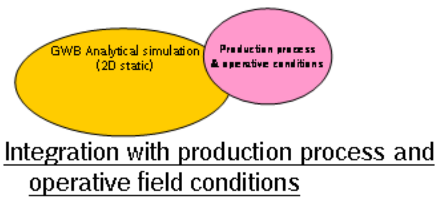


Figure 8: "GWB" analytical simulation integration with production process and operative field conditions

3. Integration with rubber profile design (FEM static) for *"GWB"* simulation enhancement with attack angles and torsion (2D → 3D):

- Further integration with *"GWB"* analytical simulation throughout additional rubber profile functions:
    a. ALK(distance, attack angle);
    b. α2(distance, attack angle);
    c. Contact point(distance, attack angle;

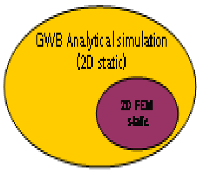


Figure 9: "GWB" analytical simulation integration with rubber profile design (static)

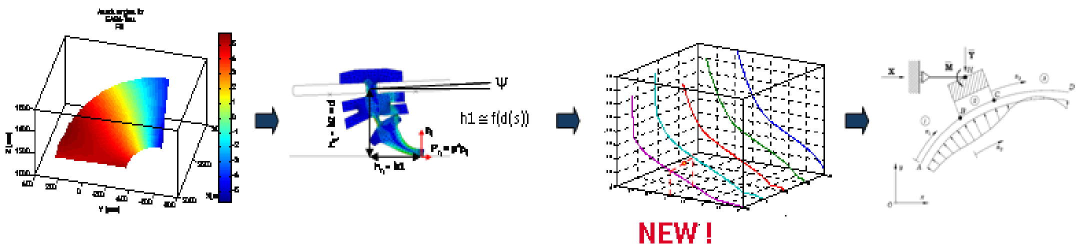


Figure 10: Example of the "GWB" analytical simulation integration with rubber profile design (static)

4. Integration with rubber profile design (FEM dynamic) for *"GWB"* transient simulation:

- Integration of vibration and transient analysis;
- Transient rubber profile behaviour within *"GWB"* analytical simulation will be taken into account throughout the rubber profile dynamic properties determined in 2D FEM dynamic.

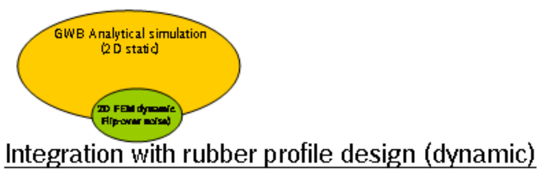


Figure 11: "GWB" analytical simulation integration with rubber profile design (dynamic)

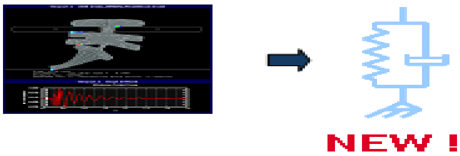


Figure 12: Example of the "GWB" analytical simulation: rubber impedances calculated with FEM (dynamic)

5. Integration with 3D FEM:

- Improvement interface exchange data between Wiper Analytical simulation and FEM 3D.

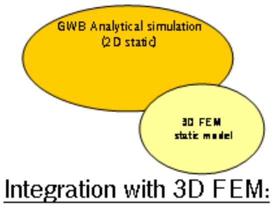


Figure 13: "GWB" analytical simulation integration with 3D FEM

6. Integration with Aerodynamic design:

- Improvement interface exchange data between Fluent output and analytical simulation program.

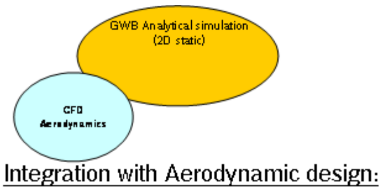


Figure 14: "GWB" analytical simulation integration with Aerodynamic design

7. Enhancement of wipeability assessment with extended static simulation capabilities (with attack angles and torsion: 2D → 3D and determination of vibrational modes):

- Improve wipeability assessment accuracy with the integration of attack angles and flex-torsion mechanism (warping);
- New output: wipeability assessment on the whole wiping pattern ("Continuous" wiping assessment);
- Externsion to vibrational modes during contact.

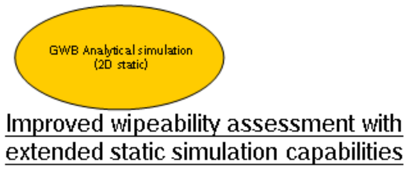


Figure 15: "GWB" analytical simulation integration with improved wipeability assessment with extended static simulation capabilities

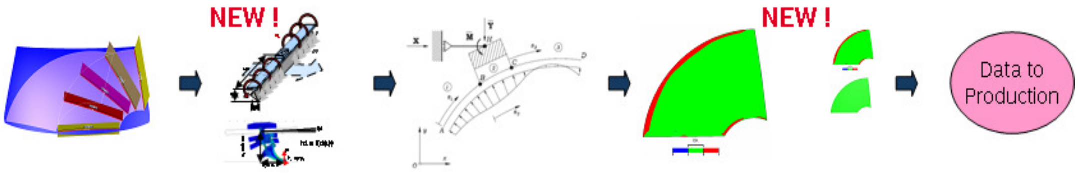


Figure 16: Example 1 of the "GWB" analytical simulation integration with improved wipeability assessment with extended static capabilities

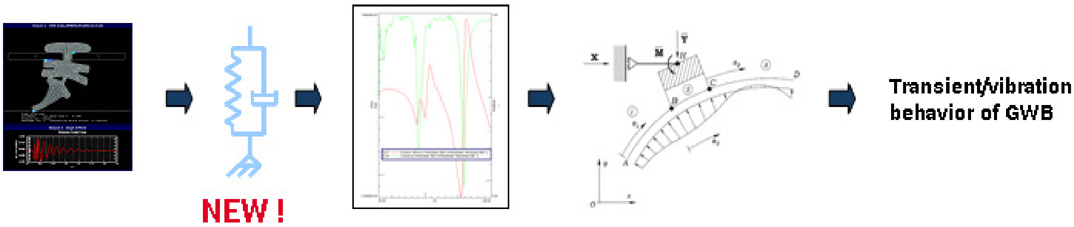


Figure 17: Example 2 of the "GWB" analytical simulation integration with improved wipeability assessment with extended static capabilities

8. Modular capabilities in function of the end user

- Modulization of simulation software for advanced user locations, low cost user locations, production user locations.

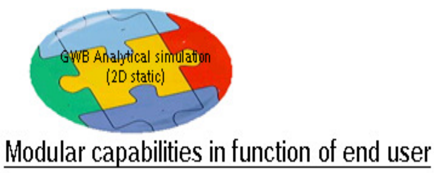


Figure 18: Modular capabilities in function of the end user

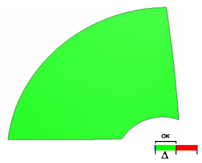
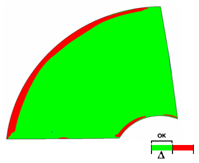
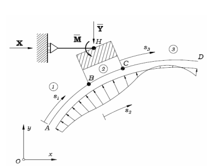
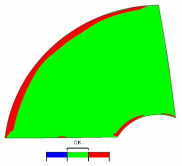
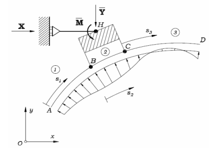
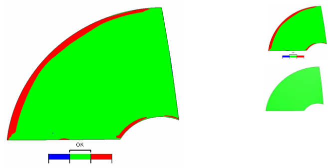
	SIMULATION PORTFOLIO	Low cost AA Baukasten APPLICATION		Customised APPLICATION		Advanced APPLICATION	
		local plant	RBBE	local plant	RBBE	local plant	RBBE
	Wipeability assessment						
ANALYTICAL: SCREEN ANALYSIS TOOL	 BASED ON SCREEN CURVATURE WITH LARGE TOLERANCES (ACCEPTANCE OF NOT-WIPED AREAS)	X					
ANALYTICAL: SCREEN ANALYSIS TOOL	 BASED ON SCREEN CURVATURE (NOT WIPED AREAS NOT ALLOWED)			X			X
	Wipeability assessment and Vertebra design						
ANALYTICAL: SCREEN ANALYSIS TOOL 	 BASED ON ALK SIMULATION (1 OPERATIVE CONDITION)			X	X		
ANALYTICAL: SCREEN ANALYSIS TOOL 	 BASED ON ALK SIMULATION (MULTIPLE OPERATIVE CONDITION)						X

Figure 19: Example of modular capabilities in function of the end user

At beginning of the project was elaborated the SPMP. In Figure 20 it's possible to realize that the software creation was planned to take no more than 3 years. At this moment it is a fact that the software won't be completed released in the next year. However, there is the strong possibility to release a smaller version in the end of 2010.

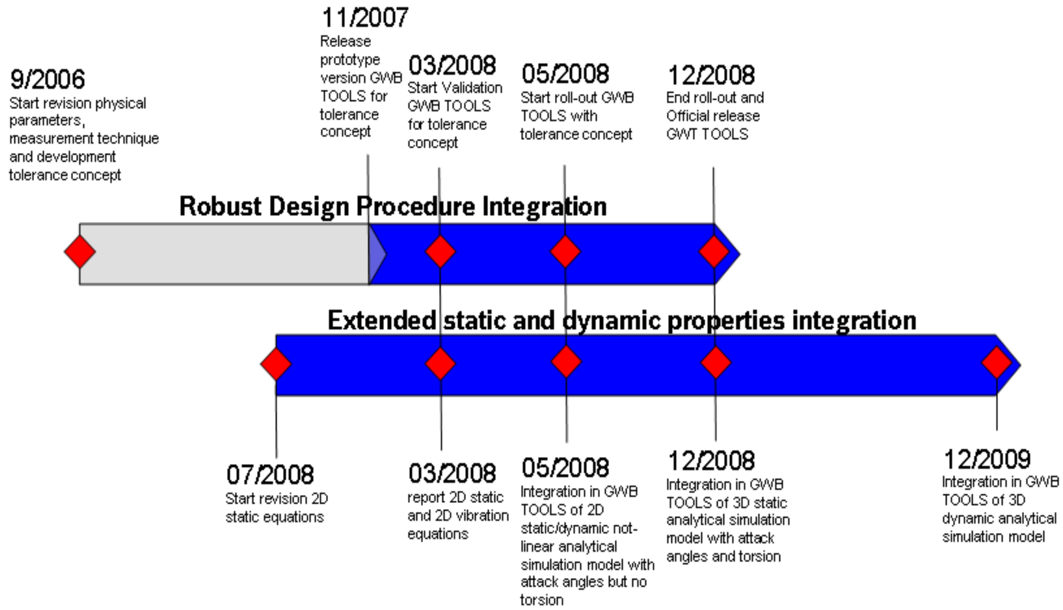


Figure 20: Original timeline for "GWB" tools project according to Bubba (2008f).

Robust design module:

From all the disciplines and the studies required by the client for the *"GWB"* simulation software, for only one of them was performed software tests. This attribute concerns the robust design procedure.
In order to clarify better this procedure it now will be explained this process.

One of the most important new advantages of the *"GWB"* tools software is the linearized approach in the simulation of the contributes of the standard deviations of field and production parameters onto contact force deviations (σ_{ALK}, see Figure 21). The linearized approach is on the basis of the robust design of the vertebra shape used in *"GWB"* Tools: each parameter is, within the typical ALK design space (see Figure 22) quite independent from each other and the overall variance of the effects can be linearly combined.

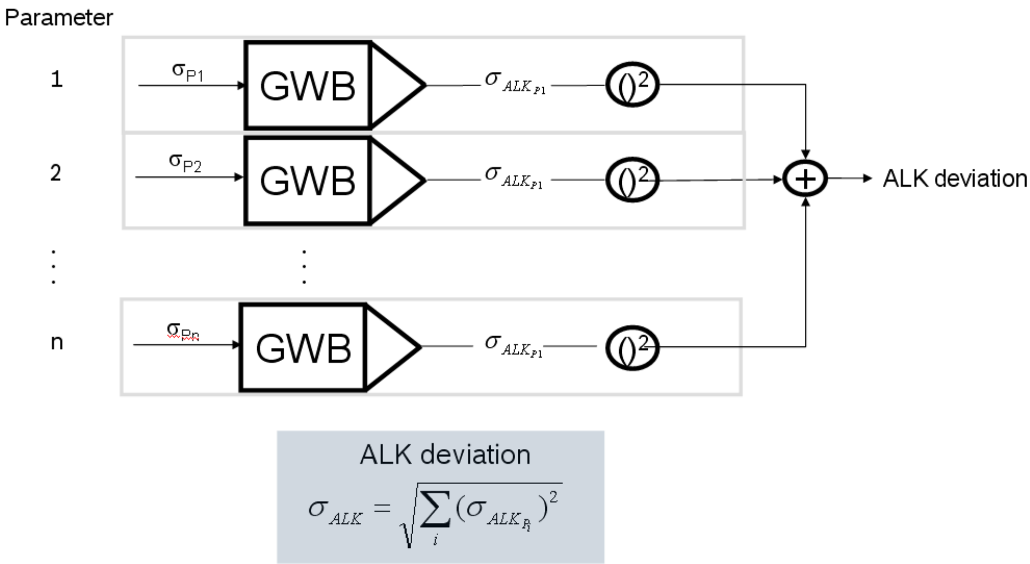


Figure 21: Robust design integration (SPC ALK linearized system)

The variation of each field and process parameter will cause a deviation of ALK into the ALK design space (defined in Figure 22 and still under investigation in Bubba (2007e), RBBE ED-WS/EAB4 (2007c), Vinckenroye (2009).

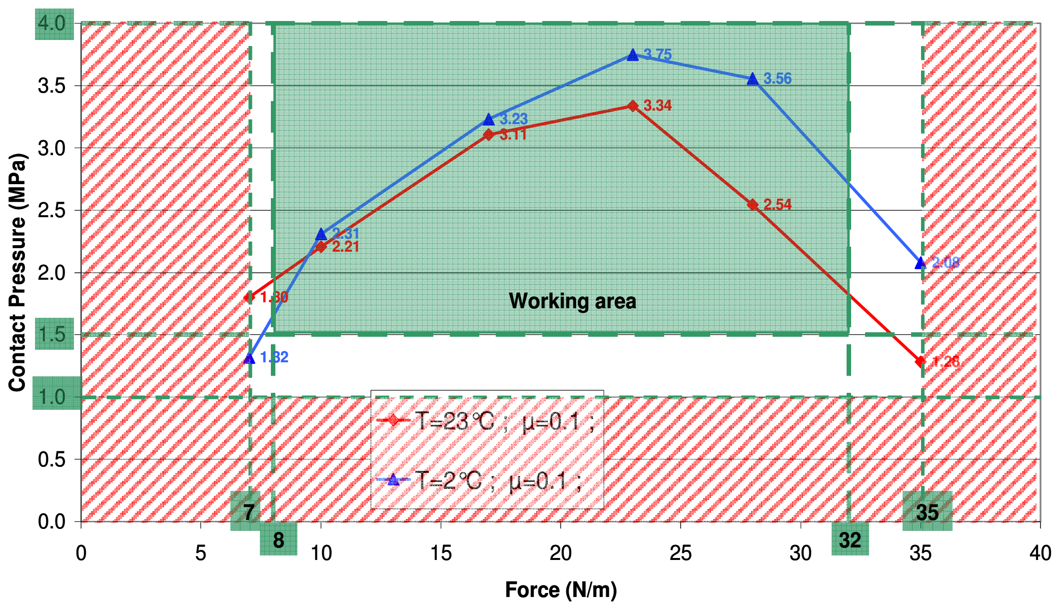


Figure 22: Upper and lower limits of wipeability ALK

In order to guarantee wipeability, all ALK deviations should remain within the allowed design space. In order to fulfil this requirement, two options are possible:

1. For each possible parameter deviation, the correspondent ALK on the windscreen sections has to be calculated and has to lie be within the design space thresholds (see example in Figure 23). In order words, each optimization loop for the vertebra optimization has hundreds of function calls (calculations) that have to be performed for each section in order to check if all possible ALK deviations are still within design space. The final robust vertebra shape will be that one that fulfils all the design conditions within the design space thresholds. Today this is not applied due to the enormous amount of calculation demanded;
2. Design space at nominal design conditions can be reduced (see Figure 24) by taking into account the ALK variances (linearly combined): robust design is met by calculating the optimal vertebra shape in nominal condition which lies within the reduced design space. In other words, each optimization loop runs in nominal conditions only and does one single function call per section. This is the new robust design methodology that will be used within *"GWB"* Tools.

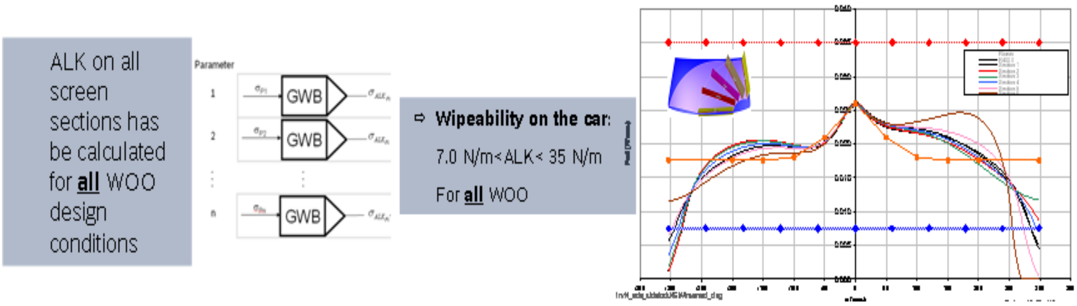


Figure 23: Current situation of the ALK simulation

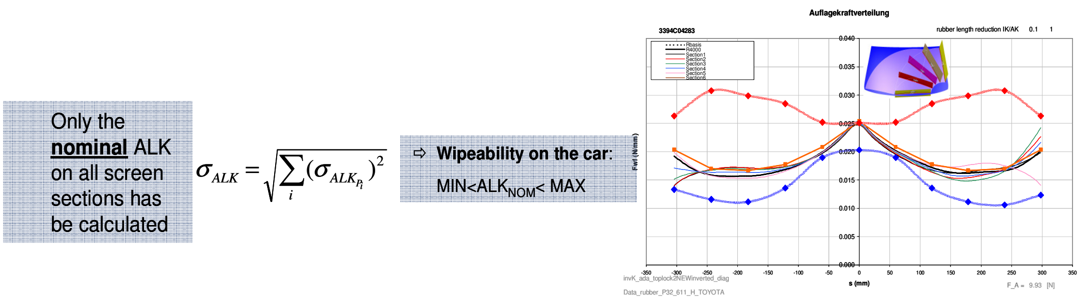


Figure 24: Prospected situation of the ALK simulation

Once the robust design optimization has been performed, according to the new methodology, we can automatically calculate (Figure 25) the contribute of all production parameters for ALK SPC (Statistical process control) in production.

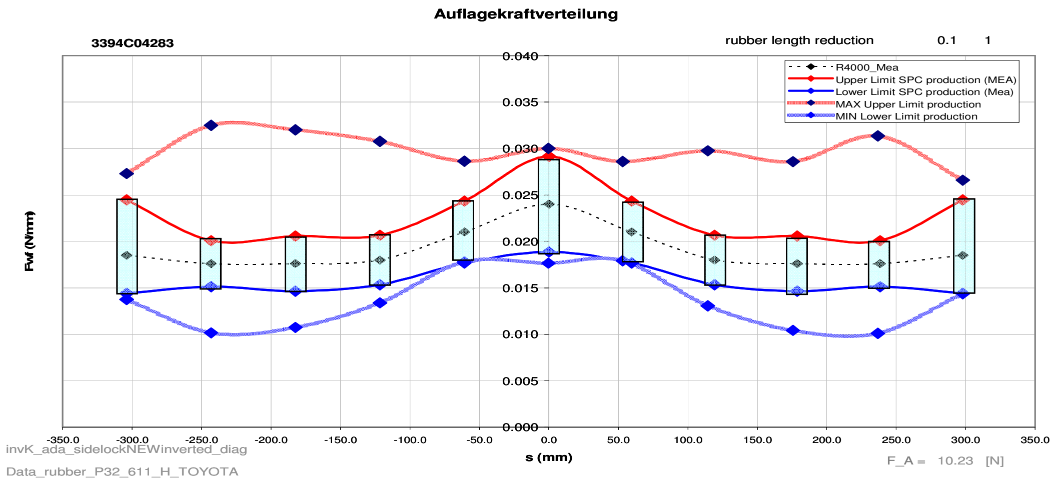


Figure 25: SPC ALK for production parameters only (control on limit number of points)

The possibility to automatically calculate the ALK SPC will allow a drastic modification of the current design workflow. Nowadays indeed, the ALK SPC is empirically defined at the really end of a long testing activity of the application engineering department (see Figure 26) and only a few of possible main parameter variances are taken into account during testing.

The new methodology (see prospected solution in Figure 27) has the potential to theoretically predict all main ALK variances and thus reduce (target is "one single verification loop") the demanded amount of testing and trial and error loops.

The introduction of this software will cause changes in the structure of communication between departments as well as changes in the procedures used.

The current scenario and the new can be seen in the following figures.

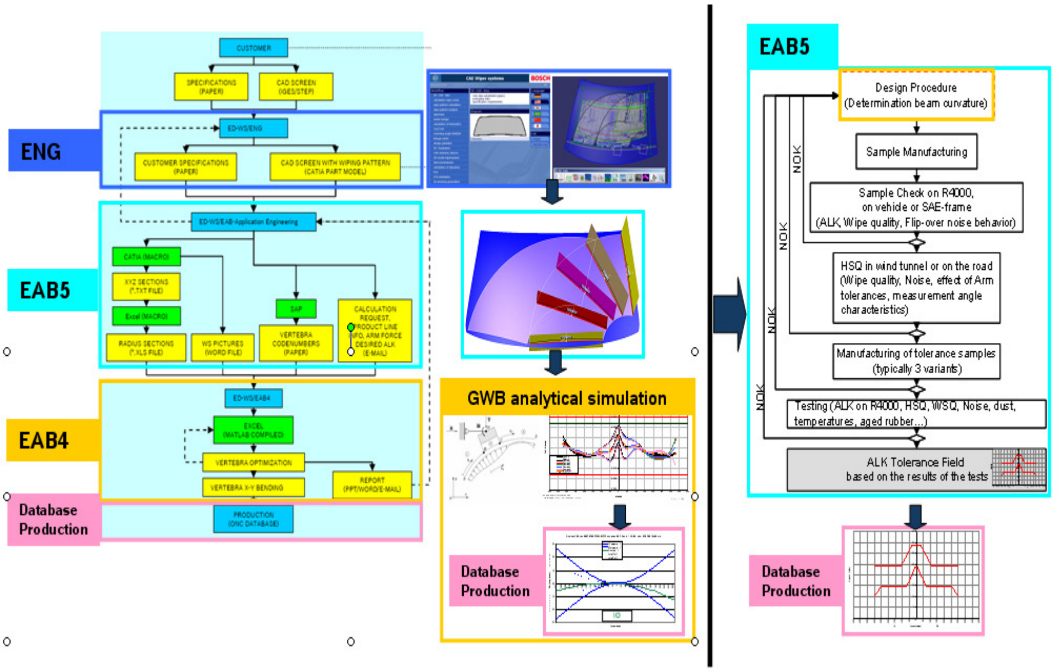


Figure 26: Current design workflow overview

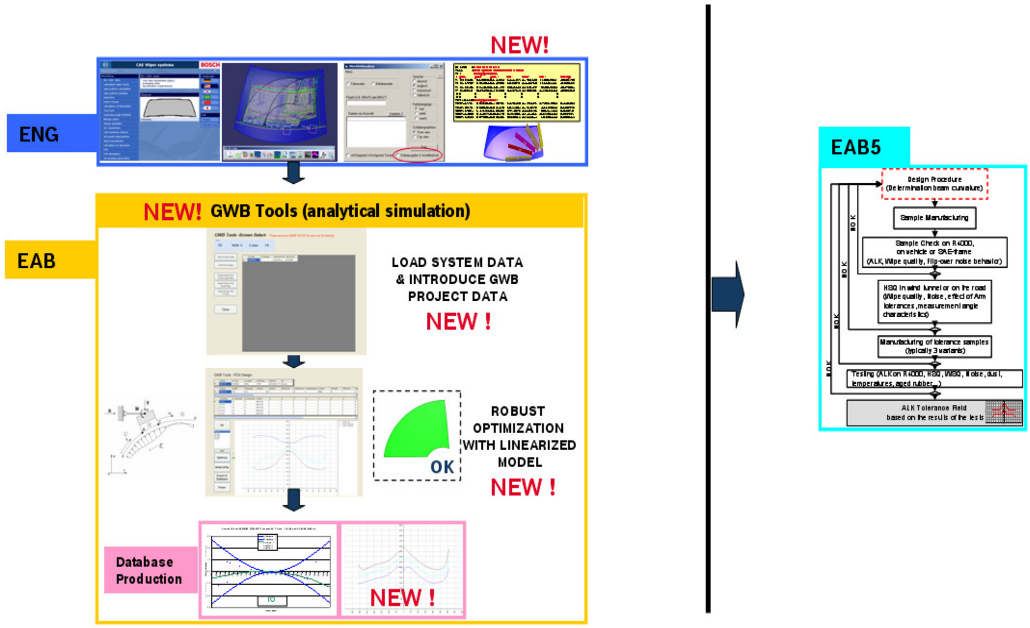


Figure 27: Prospered design workflow overview

3.3 Validation test process

In software development process its expected and desired to build a fault free software product. To achieve this complicated mission it should always be tests in every phase of the development instead of a separate testing phase. Normally in this way it's possible to save time and money.

So, during all the time, the product should be checked constantly and precisely. Though there are two periods where an intense procedure of tests should be always done: in the end of each phase (verification), and before the product is delivered to the client, validation.

In fact what its showed next is all the test procedure for *"GWB"* software prototype validation process before the deliver to the client. At this moment, is mandatory to check the software in all ways possible in order to make a good advance to achieve the fault free software.

Goal of the validation process is to assess the accuracy of the simulation algorithm as compared to real measurements for:

- Vertebra bending shape as compared to real measurements;
- Contact force simulation (ALK) as compared to real measurements on R4000;
- Contact force design tolerances as compared to field performances.

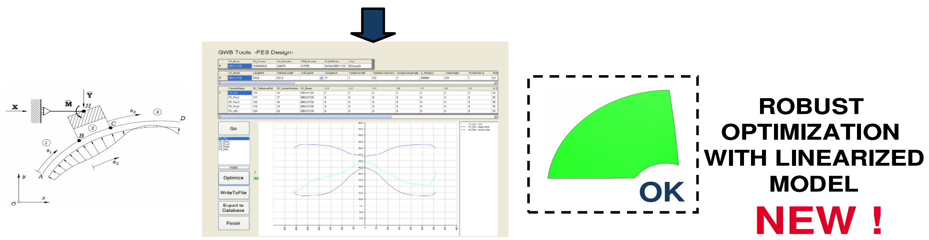


Figure 28: "GWB" tools basic procedure

From the theoretical point of view, the validation activity has to cover the possible product line combinations and this in compliance with the different possible field conditions.

In order to reduce the number of testing activities and their consequent costs, a design of experiments, DOE test matrix, was built (Table 1). The DOE matrix has the ambition of represent the current product line differentiation and the different field conditions.

	L (total lenth FeS)	FeS type	Screen type	adapter type	GWB type
Codenumber 1_FeS 3394C08323_ZsB 3398.132.594	1	1	2	1	1
Codenumber 2_FeS 3394C08321_ZsB 3398.132.595	2	2	1	1	1
Codenumber 3_FeS 3394C08319_ZsB 3398.132.596	2	2	2	1	1
Codenumber 4_FeS 3394C08321_ZsB 3398.132.597	2	2	1	2	1
Codenumber 5_FeS 3394C08317_ZsB 3398.132.598	2	2	2	2	1
Codenumber 6_FeS 3394C08395_ZsB 3398.132.599	3	3	2	2	1
Codenumber 7_FeS 3394C08377_ZsB 3398.132.600	1	1	2	4	2
Codenumber 8_FeS 3394C08379_ZsB 3398.132.601	4	2	2	4	2

Table 1: DOE Test Matrix

Toward, to complete the previous information, two additional small tables were built as legend for Table 1. These allow seeing which are the product line differentiations between all the test beams, Table 2, and the default arm forces and rubbers used for the current product line, see Table 3

	1	2	3	4
L (total lenth FeS)	400	600	750	700
FeS type	6x0,8	7x0,9	7x1	
Screen type	Flat Screen	car (DC W204/Ford S-max/Golf V)		
adapter type	TL	SL		S2
GWB type	GWB2S1	GWB2S2		
Average arm force	17N/m			
ALK target	default/within Area			
rubber type	33.611 EPDM			
spoiler type	standard			
Production FeS for each codenumber	100 stuks			
FeS to be numbered	100 stuks			
FeS to be scanned on Microna	all			
Production ZsB for each codenumber	~ 50 (2 matten)			
ZsB to be complete assembled and numbered (record FeS numbers)	30			
ZsB to be measured on ALK	30			
ZsB to be measured on WQS tests	30			
ZsB to be measured on HSQ tests	30			
Time Period shift	1 month			

Table 2: Legend for DOE Test Matrix (Table 1)

	ALK HELP			
	1	2	3	4
L Bouwkast reference lenth FeS	400	600	750	700
weight complete assempby [kg]	0,091	0,123	0,153	0,151
ALK NOMINAL [N]	7,69	11,41	14,25	13,38
ALK +10% [N]	8,37	12,43	15,53	14,57
ALK -10%-1N [N]	6,01	9,39	11,98	11,19

	ALK HELP (for repeated tests)			
	1	2	3	4
L Bouwkast reference lenth FeS)	400	600	750	700
FeS type	6x0,8	7x0,9	**7x1**	(7x0.9)
rubber default	33.611 EPDM	33.611 EPDM	**33.611 EPDM**	33.611 EPDM
rubber optional (standard productie)	32.613 H	32.613 H	**32.613 H**	32.613 H
rubber optional (standard productie)	37.613 FX	37.613 FX	**37.613 FX**	37.613 FX

	WSQ HELPHELP			
	1	2	3	4
L (total lenth FeS)	400	600	750	700
weight complete assempby [kg]	0	0	0	0
Arm Force NOMINAL [N]	6,80	10,20	12,75	11,90
Arm Force +10% [N]	7,48	11,22	14,03	13,09
Arm Force -10%-1N [N]	5,12	8,18	10,48	9,71

	HSQ HELPHELP			
	1	2	3	4
L (total lenth FeS)	400	600	750	700
weight complete assempby [kg]	0	0	0	0
Arm Force NOMINAL [N]	6,80	10,20	12,75	11,90
Arm Force +10% [N]	7,48	11,22	14,03	13,09
Arm Force -10%	6,12	9,18	11,48	10,71

Table 3: Legend for DOE Test Matrix (Table 1)

It is clear that it is not possible to monitor all possible influencing parameters during a validation activity. A limited choice had to be defined. Thanks to the extensive sensitivity study performed in the past (see Bubba, 2007c) it was possible to rank the major influencing parameters onto ALK and all of these have been monitored (Table 4).

D	DIRECTLY	Measureable/ Simulated
ND	NOT-DIRECTLY	Measureable/ Simulated
NA	Not available	

	Parameter Name	Responsible	Estimation Technique (Simulation)		Validation technique (Experimental)	
			Sim	Used technique	Exp	Used technique
deviations	Lower Limit	EAB6	D	FEM	ND	b. m. on a car (Toyota) not on a probe
	Upper Limit	EAB6	D	FEM	ND	b. m. on a car (Toyota) not on a probe
	FIELD P.: Screen StdDev	EAB4	D	analytical	D	Image scan technique for screen (but not statistic study done yet)
	FIELD P. : Arm force system StdDev (wiping angle)	ENG	D	analytical	NA	b.m. on full system not available
	FIELD P.: Arm force StdDev (spring, friction torque)	ENG	D	analytical	NA	b.m. on full system not available
	FIELD P.: Arm force system StdDev (HSQ)	ENG	D	CFD	NA	Not available
	FIELD P.: Spoiler force distribution StdDev (HSQ)	EAB4	D	CFD	NA	Not available
	FIELD P.: Adapter Friction system StdDev (wip. Angle, HSQ)	EAB2	NA	sytem input loading unknown	D	friction torque measurement available but input loading unknown
	FIELD P.: rubber profile stiffness StdDev (friction)	EAB4	D	FEM	ND	ALK/distance curve on GWB probe
	FIELD P.: rubber profile stiffness StdDev (attack angle)	EAB4	D	FEM	ND	ALK/distance curve on GWB probe
	GWB Pr. P.: Bending StdDev	TEF1.3 / MOE5	D	analytical	D	Microna
	GWB Pr. P.: clamping angle StdDev (assembly process)	TEF1.3 / MOE5	D	analytical	ND	ALK
	GWB Pr. P.: FeS thickness StdDev	EAB2	D	analytical	NA	not monitored by QMM
	GWB Pr. P.: Rubber Lenght IK/AK StdDev	EAB2	D	analytical	D	Length Sensor but no statistic studies
	GWB Pr. P.: Measurement StdDev	TEF1.3	D	analytical	ND	ALK
	GWB Pr. P.: FeS E-mod., residual stress StdDev	EAB2	D	analytical	D	S-N curve, but no statistics available at all
	GWB Pr. P.: FeS Width StdDev	EAB2	D	analytical	NA	not monitored by QMM
Nominal	Clamping angle ***	EAB4	D	analytical	ND	ALK
	Adapter stiffness ***	EAB4	D	analytical	ND	ALK
	Arm force ***	EAB4	D	analytical	ND	indirect with ALK Sensors
	Rubber profile stiffness friction (20°, μ=0.4, attack angle=0°) at 17 N/m ***	EAB4	D	analytical	ND	ALK/distance curve on GWB probe
	Weight ***	EAB4	D	analytical	D	weight balance
	Spoiler stiffness ***	EAB4	D	analytical	ND	ALK/distance curve on GWB probe

Table 4: Validation technique methods used for each parameter

Considering the endless possibilities of combinations, a plan had to be done to guide carefully all the validation life activity. As result the Table 5 was built as a test plan. For the validation activity, two stages were chosen (T1 and T2) and each of them starts with the production of all the components of the wiper blades, the measurements of the components, the assembly of them and finally the tests.

(green)	done
(orange)	ongoing
(grey)	not applicable

	L (total lenth FeS)	FeS type	Screen type	adapter type	GWB type
Codenumber 1 _FeS 3394C08323 _ZsB 3398.132.594	1	1	2	1	1
Codenumber 2 _FeS 3394C08321 _ZsB 3398.132.595	2	2	1	1	1
Codenumber 3 _FeS 3394C08319 _ZsB 3398.132.596	2	2	2	1	1
Codenumber 4 _FeS 3394C08321 _ZsB 3398.132.597	2	2	1	2	1
Codenumber 5 _FeS 3394C08317 _ZsB 3398.132.598	2	2	2	2	1
Codenumber 6 _FeS 3394C08395 _ZsB 3398.132.599	3	3	2	2	1
Codenumber 7 _FeS 3394C08377 _ZsB 3398.132.600	1	1	2	4	2
Codenumber 8 _FeS 3394C08379 _ZsB 3398.132.601	4	2	2	4	2

T1 (ZsB ESM)	01-FeS production and numbering	02-Microna metingen (X-Y all FeS)	03-ZsB assembly and numbering (coupled to FeS numbers)	04-ALK on R4000 (NOM / +10% / -10%-1N arm force) at T=20°	05-WSQ Wiping quality test (evaluation according to macro)	06-HSQ test	07-repeat ALK on R4000 measurement at T=20° (EPDM / H / FX)	08-repeat ALK on R4000 at T= -20°/-10°/0°/10°/20° (EPDM / H / FX)	09-repeat ALK on R4000 at very lowfriction coefficients at T=20° (EPDM / H / FX)	10-repeat ALK on R4000 with spoiler removed at T=20° (EPDM)	11-ALK on R4000 (NOM / +10% / -10%-1N arm force) at T=20° FeS NOT-coated (EPDM)
Codenumber 1	x	x	x	x	x		x				
Codenumber 2	x	x	x	x	x		x			x	x
Codenumber 3	x	x	x	x	x	x	x			x	x
Codenumber 4	x	x	x	x	x		x				
Codenumber 5	x	x	x	x	x		x				
Codenumber 6	x	x	x	x	x	x	x			x	x
Codenumber 7	x	x	x	x	x		x				
Codenumber 8	x	x	x	x	x	x	x			x	

T2 (ZsB Pr)	01-FeS production and numbering	02-Microna metingen (X-Y all FeS)	03-ZsB assembly and numbering (coupled to FeS numbers)	04-ALK on R4000 (NOM / +10% / -10%-1N arm force) at T=20°	07-repeat ALK on R4000 measurement at T=20° (EPDM / H / FX)	10-repeat ALK on R4000 with spoiler removed at T=20° (EPDM)	11-ALK on R4000 (NOM / +10% / -10%-1N arm force) at T=20° FeS NOT-coated (EPDM)
Codenumber 1							
Codenumber 2	x	x	x	x	x	x	x
Codenumber 3	x	x	x	x	x	x	x
Codenumber 4							
Codenumber 5							
Codenumber 6	x	x	x	x	x	x	x
Codenumber 7							
Codenumber 8	x	x	x	x	x	x	

Table 5: Test Matrix

3.3.1 Vertebra bending shape as compared to real measurements

The vertebra is the spring element of the *"GWB"* and its theoretical shape is designed to achieve the desired contact force onto the screen. Vertebra shape deviations from theoretical values results in contact force deviations onto the screen. Goal of this chapter is to measure the current RBBE vertebra shape standard deviation in order to determine the theoretical value of the parameters, $\sigma_k \times \sqrt{\lambda_0}$ (see Bubba, 2007c), to be used in the *"GWB"* Tool simulation.

Therefore Mycrona measurements have been performed for all examined code numbers. For each code number a minimum of 60 vertebras (equivalent of 30 *"GWB"*) has been scanned with Mycrona equipment (TEF 1.3 responsibility).

3.3.1.1 Results

Mycrona data are first rigidly corrected so that they are centred at the theoretical zero s coordinate and their tangent at s=0 is put to zero (see theory in Braun (2009a), Braun (2009b), Braun (2009c) and Braun (2009d)).

The Figure 29 and Figure 30 display the shape deviation of the measured vertebra from the mean and it's respectively standard deviation from the mean by the using of two different theories. The rest of the other code number results can be consulted in Annex A.

Formula $\sigma_h(x) \equiv \sigma_k \sqrt{\lambda} \times dw$

From the shape deviation to the mean we can calculate the measured standard deviation and compare to the theoretical curve (see theory in Braun (2009a), Braun (2009b), Braun (2009c) and Braun (2009d)).

The measured standard deviation is compared to two different theoretical curves (the $\sigma_k \times \sqrt{\lambda_{0.}}$ and $\sigma_k \times \sqrt{\lambda_{0\,simp.}}$). The theoretical curve (the first) better takes into account the boundary conditions of the vertebra at s=0, the second is a more simplified expression.

By analysing the Figure 29 and Figure 30 as well as the other results in Annex A, it seems that the variation of the shape depends on the length of the vertebra. The longer the vertebra the largest the variation of the vertebra seems to be. From this analysis we observe the deviation is similar in inner circle and outer circle.

We can also visualize that in almost of the cases the behaviour of the deviation is symmetric to axis x. But we see that exist 3 cases of an asymmetric behaviour. The C08321_2 (4), C08395, C08377. In the first two cases there are some deviations that represent 50% more bent in inner and outer circle then the symmetric range from the mean. In the last case, C08377, the same seems to happen but in the opposite direction, it means there are high deviations that represent 50% less bent.

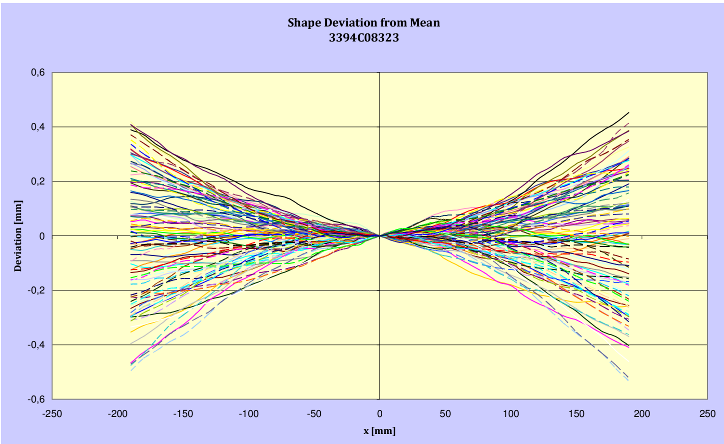


Figure 29: Shape deviation and Standard deviation from the Mean for C08323

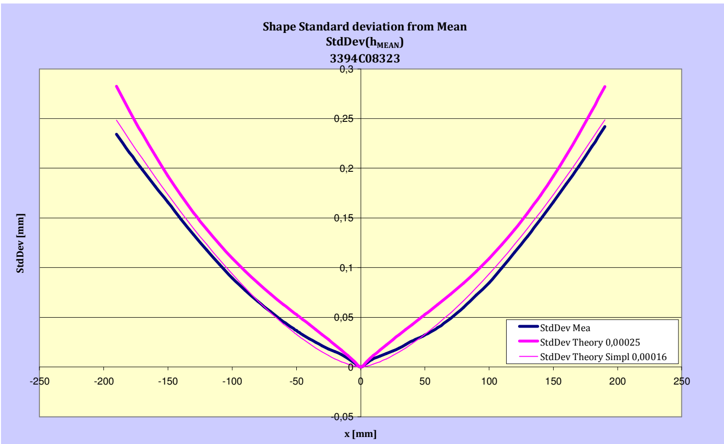


Figure 30: Standard deviation from the Mean for C08323

(Consult Annex A to see the other code numbers)

3.3.1.2 Vertebra bending StdDev determination

Table 6 and Figure 31 give an overview of the fitting parameter values used to fit the measured standard deviation. As we can see there exists a good fitting value both for CDW and Bekaert materials (≈0.00025). The fitting is not accurate at higher lengths. Here a higher value seems necessary (≈0.0004). The reason for the higher values is at the time of this report not clear.

	L (total lenth FeS)	FeS type	Screen type	adapter type	GWB type	Bending StdDeV (Theory) GWB TOOLS 2009	Bending StdDeV (Simpl. Theory) GWB TOOLS 2009
Codenumber 1_FeS 3394C08323_ZsB 3398.132.594	1	1	2	1	1	2,5E-04	1,6E-04
Codenumber 2_FeS 3394C08321_ZsB 3398.132.595	2	2	1	1	1	2,5E-04	1,6E-04
Codenumber 3_FeS 3394C08319_ZsB 3398.132.596	2	2	2	1	1	2,5E-04	1,6E-04
Codenumber 4_FeS 3394C08321_ZsB 3398.132.597	2	2	1	2	1	2,5E-04	1,6E-04
Codenumber 5_FeS 3394C08317_ZsB 3398.132.598	2	2	2	2	1	2,5E-04	1,6E-04
Codenumber 6_FeS 3394C08395_ZsB 3398.132.599	3	3	2	2	1	4,0E-04	3,1E-04
Codenumber 7_FeS 3394C08377_ZsB 3398.132.600	1	1	2	4	2	2,5E-04	1,6E-04
Codenumber 8_FeS 3394C08379_ZsB 3398.132.601	4	2	2	4	2	4,0E-04	3,1E-04

Table 6: Bending Fit Parameter Variation

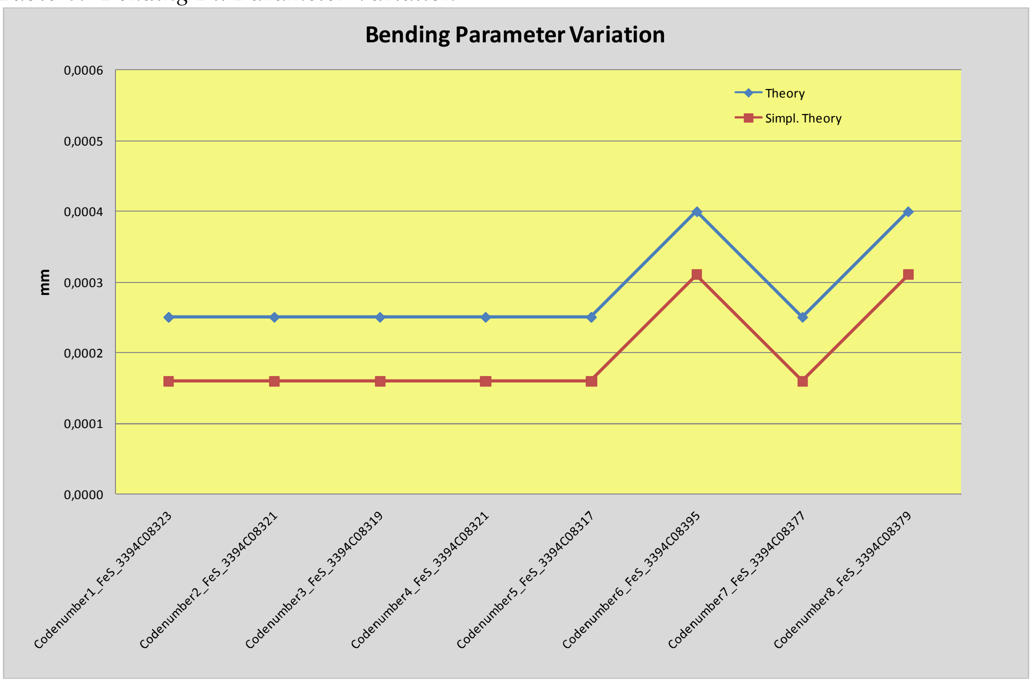


Figure 31: Bending Parameter Variation

3.3.2 ALK simulation as compared to real measurements on R4000

The purpose of this sub-chapter is to analyse if experimentally measured values of ALK are close to the *"GWB"* simulation.

In order to better visualize the experimentally measured values of ALK and the simulations values, was compile for each beam a couple of plots that synthesize all the information for tree levels of arm force (nominal, minimum, maximum arm force).

3.3.2.1 Results

For a better understanding it was chosen to simulate with *"GWB"* tools two different curves. The first is nominal ALK and the second is the mean of 30 measurements. This two curves plus the curve of the real measurements ALK values are compile together in the following figures for each beam at each level of arm force.

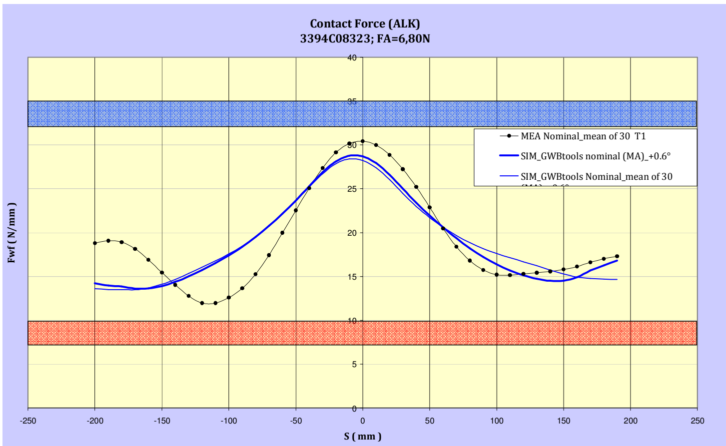


Figure 32: ALK for Nominal force for C08323

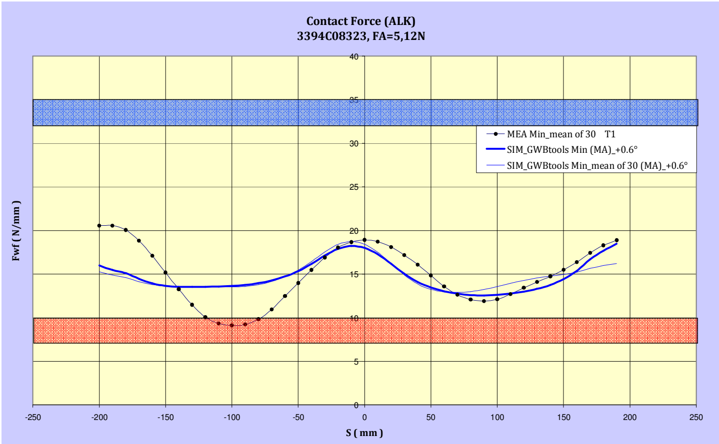


Figure 33: ALK for Minimal force for C08323

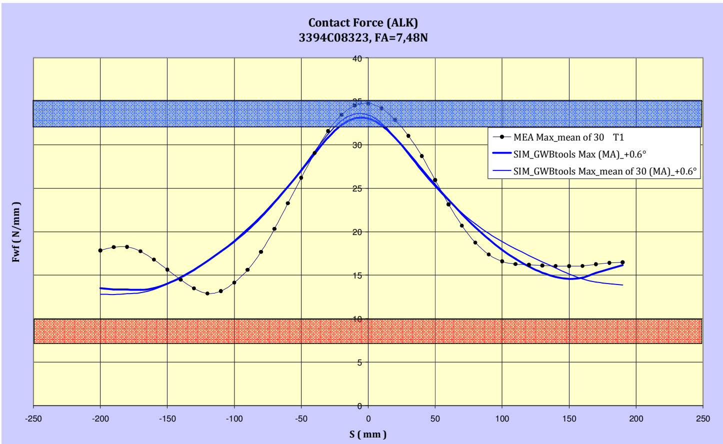


Figure 34: ALK for Maximum force for C08323

Here is only present C08323 case, but the rest of the result can be consulted in the Annex B – 1.

At this point the behaviour of the several parameters should be understand, thus some combinations of fit parameters values were done and synthesize.

For all beams it was changed each parameter fit number, while the rest of the parameters were left constant, in order to verify the changes into ALK.

In Figure 35 we see exactly the sensitivity of those behaviours on ALK after changing some parameters values. Here is only present C08323 case, but the rest of the result can be consulted in the Annex B – 2.

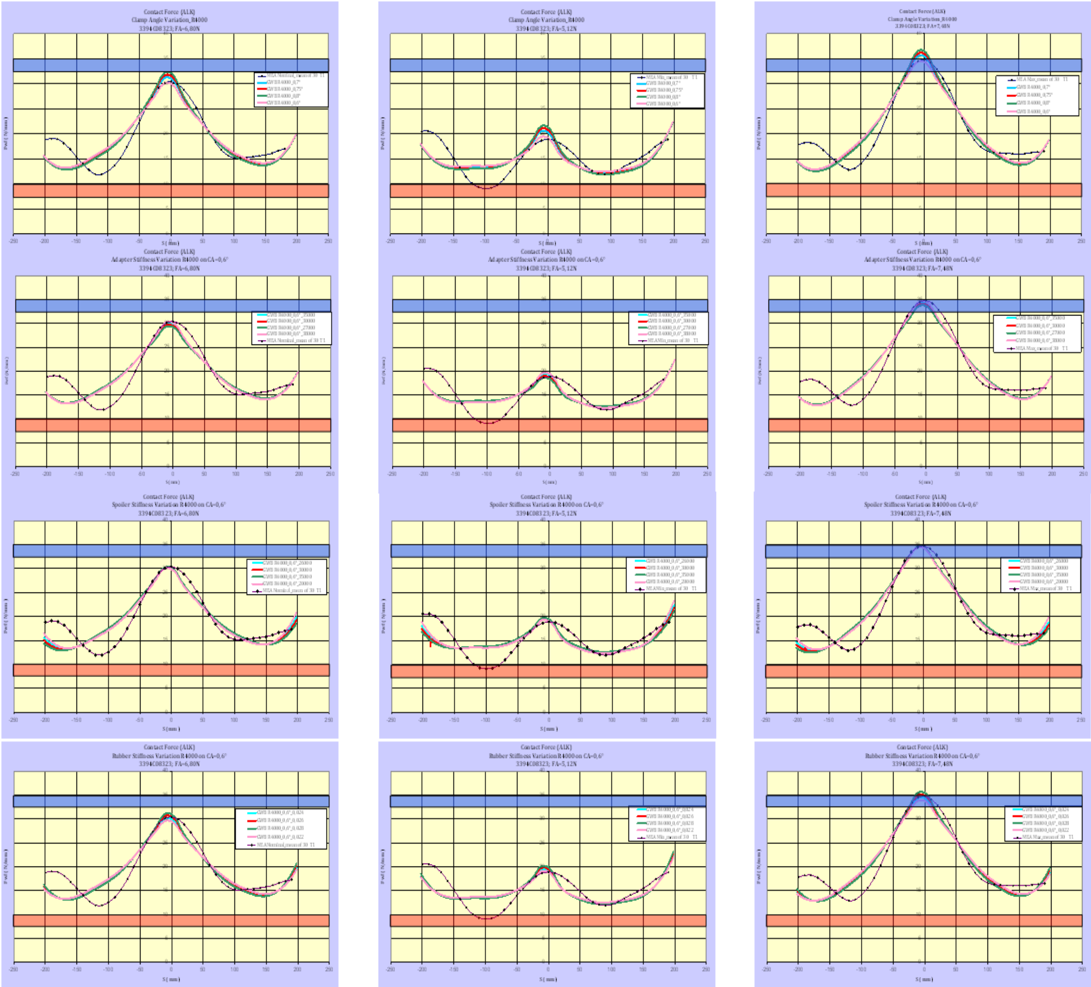


Figure 35: Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08323

3.3.2.2 Clamping angle determination

As showed in the Figure 35 (see sensitivity analyses in Bubba (2007c)) the modification of the clamping angle results in a variation of the ALK especially on the peak. The cross comparison of the 8 code numbers has showed that there is a much higher ALK sensitivity due to clamping angle deviations than due to adapter stiffness, rubber stiffness, spoiler stiffness.

Therefore it was possible to fix the adapter stiffness, the rubber stiffness and the spoiler stiffness to their theoretical values (adapter stiffness from FEM, rubber stiffness from Bubba, (2007d) and spoiler stiffness from FEM) and determine the best fitting clamping angle.

- Best fitting choice of clamping angle**: 0.6°**

3.3.2.3 Adapter stiffness determination

As the parameter before, the variation of this parameter reflects ALK changes in the peak. The behaviour of this parameter during the change of the adapter stiffness demonstrates very small variation of ALK in the peak for all the arm forces.
Likewise the previous parameter, there is a good fitting number for the samples wiper blades thus it's suggested the following number:

- Best choice of adapter stiffness fit number (from FEM)**: 38000 Nmm/rad**

3.3.2.4 Rubber profile stiffness determination

Once more the variation of one parameter reflects changes along the beam length. Anyhow, the results presented conclude that the previous fit number for the rubber profile stiffness parameter, 0.024 N/mm/mm, is still robust.
Thus is proposed to use the same fit number for the rubber stiffness parameter.

- Best choice of rubber profile stiffness fit number**: 0.024 N/mm/mm**

3.3.2.5 Spoiler stiffness determination

It's clear that this parameter has a different behaviour than the other remain parameters studied until this moment.
The modification of the spoiler stiffness has very small deviations only in inner and outer circle. Since the peak area is considered the most important area and the inner and outer circle area are not so critic, it's suggested continuing using the same fit value used in the past.

➢ Best choice of spoiler stiffness fit number(from FEM)**: 26000 N/mm^2**

3.3.2.6 Overview of the parameters fit number

In the Table 7 it is shown the parameters fit values suggested for all each beam.

	L (total lenth FeS)	FeS type	Screen type	adapter type	GWB type	Bending StdDeV (Theory)	Bending StdDeV (Simpl. Theory)	Clamping Angle [°]	Adapter Stiffness [Nmm/rad]	Rubber Stiffness [N/mm/mm]	Spoiler Stiffness [Nmm^2]
						GWB TOOLS 2009		GWB TOOLS 2009			
Codenumber 1_FeS 3394C08323_ZsB 3398.132.594	1	1	2	1	1	2,5E-04	1,6E-04	0,6	38000	0,024	26000
Codenumber 2_FeS 3394C08321_ZsB 3398.132.595	2	2	1	1	1	2,5E-04	1,6E-04	0,5-0,6	38000	0,024	26000
Codenumber 3_FeS 3394C08319_ZsB 3398.132.596	2	2	2	1	1	2,5E-04	1,6E-04	0,5-0,6	38000	0,024	26000
Codenumber 4_FeS 3394C08321_ZsB 3398.132.597	2	2	1	2	1	2,5E-04	1,6E-04	0,6	38000	0,024	26000
Codenumber 5_FeS 3394C08317_ZsB 3398.132.598	2	2	2	2	1	2,5E-04	1,6E-04	0,6	38000	0,024	26000
Codenumber 6_FeS 3394C08395_ZsB 3398.132.599	3	3	2	2	1	4,0E-04	3,1E-04	0,6	38000	0,024	26000
Codenumber 7_FeS 3394C08377_ZsB 3398.132.600	1	1	2	4	2	2,5E-04	1,6E-04	NA	NA	NA	NA
Codenumber 8_FeS 3394C08379_ZsB 3398.132.601	4	2	2	4	2	4,0E-04	3,1E-04	NA	NA	NA	NA

Table 7: Overview of all the parameter fit number for all the beams

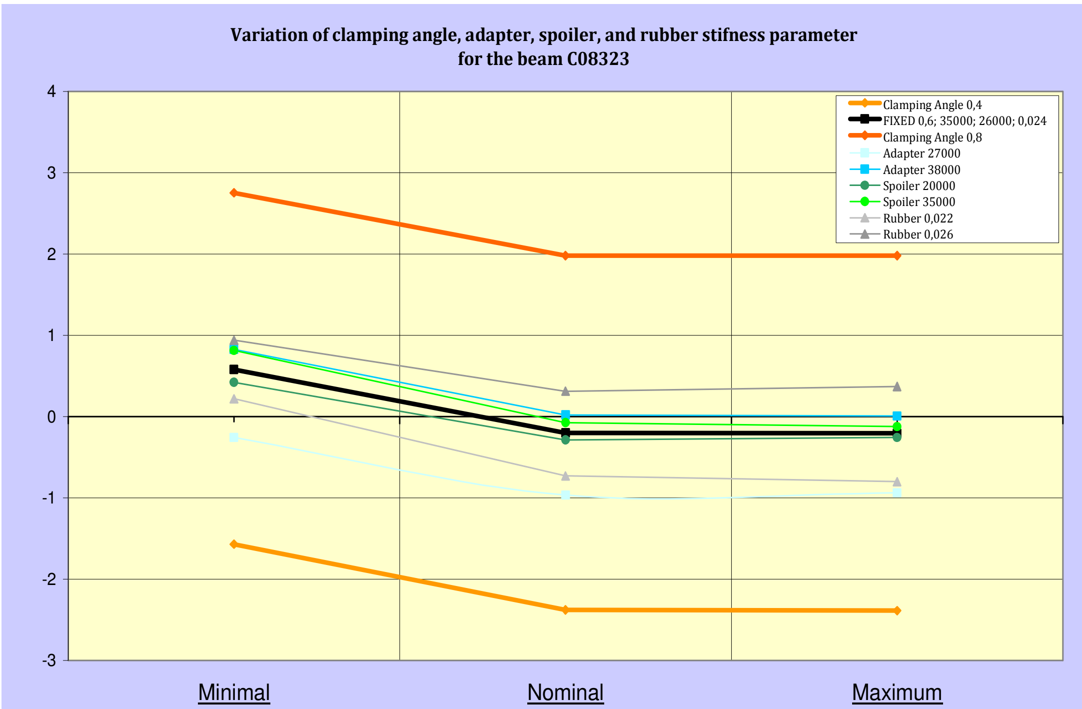


Figure 36: Overview of the variation in the peak of all parameters for different arm forces for the C08323

After this analysis of ALK variation due to changes in the fit parameters values, it's shown in Figure 36 only for one case, the C08323 vertebra, the variation of ALK peak values on the tree levels of force.
The main conclusion of the previous figure is that the clamping angle seems to be the parameter more responsible for big changes in ALK.

At this time is still not possible to simulate correctly the behaviour for S2 beams. For now *"GWB"* tools does not take into account the vertebra notch at the adapter area as has been done for Matlab non-linear program. Therefore the results for S2 beams are at this moment poor and inconclusive.

3.3.3 Contact force design tolerances as compared to field performances

Extensive sensitivity studies have been performed in the last years in order to understand the influence of process and functional parameter deviations onto ALK (see Braun (2009a), Braun (2009b), Braun (2009c), Braun (2009d); and the sensitivity resume in Figure 37, Figure 38, and Figure 39). Based on these studies, the most relevant process and functional parameters have been selected and further considered for the next robust studies.

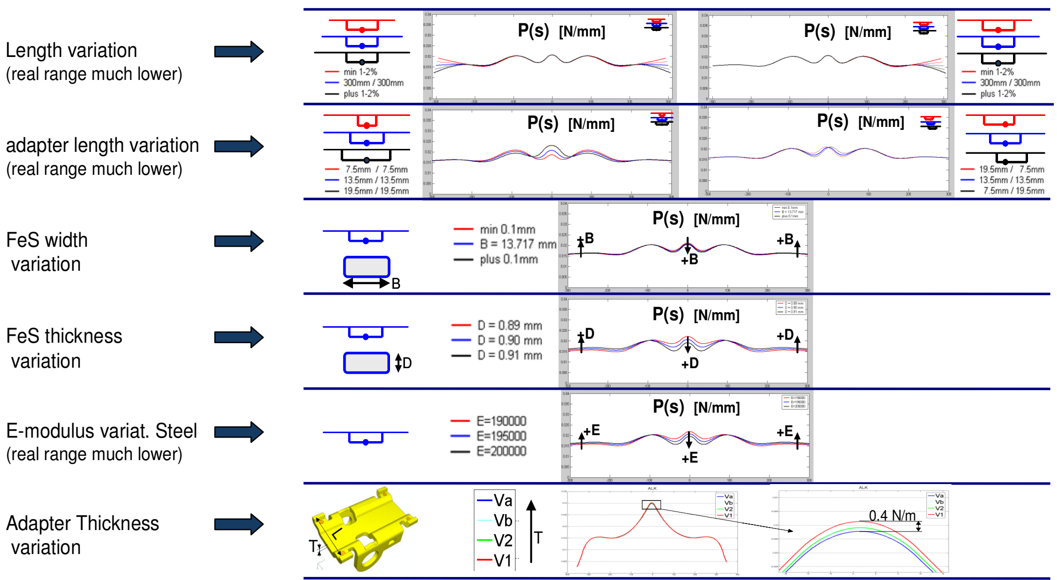


Figure 37: Sensitivity analysis of several parameters – Summary (1)

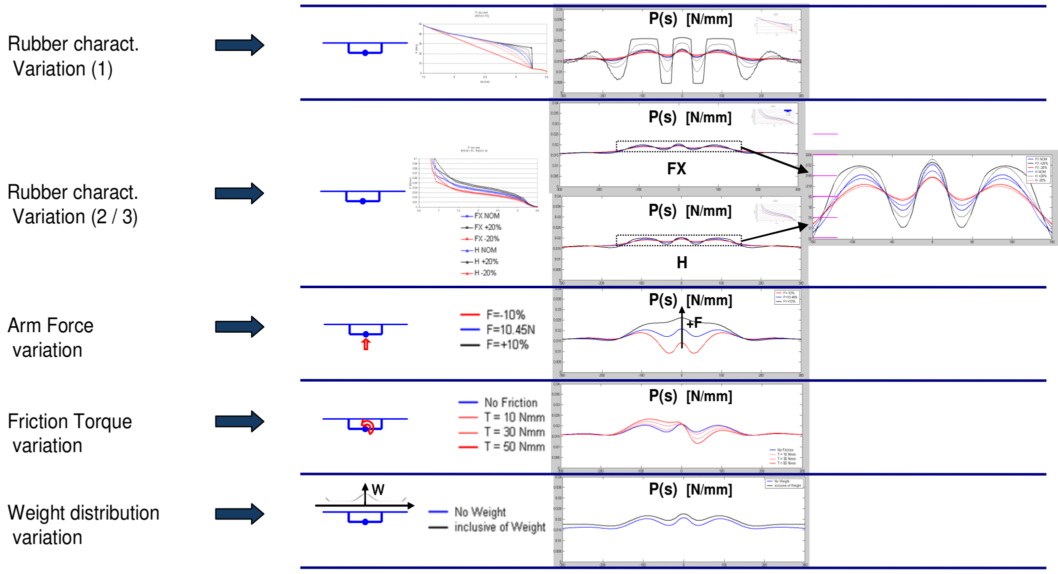


Figure 38: Sensitivity analysis of several parameters – Summary (2)

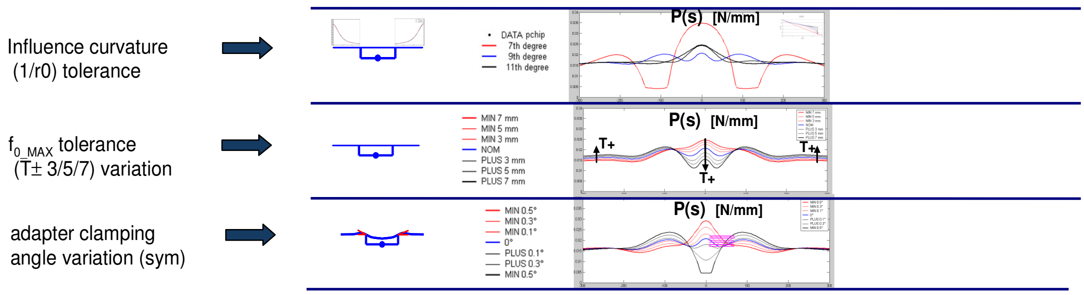


Figure 39: Sensitivity analysis of several parameters – Summary (3)

Afterward of this sensitivity analysis of several parameters occur the thought that not all the parameters are equal relevant into ALK. Concerning the preponderance of the parameters was examined the weight parameters into ALK along the beam length.

Succeeding it was able to design the Figure 40 and Figure 41 that shows exactly the analysis of influence of the parameters along the *"GWB"* length for lower and upper limits respectively.

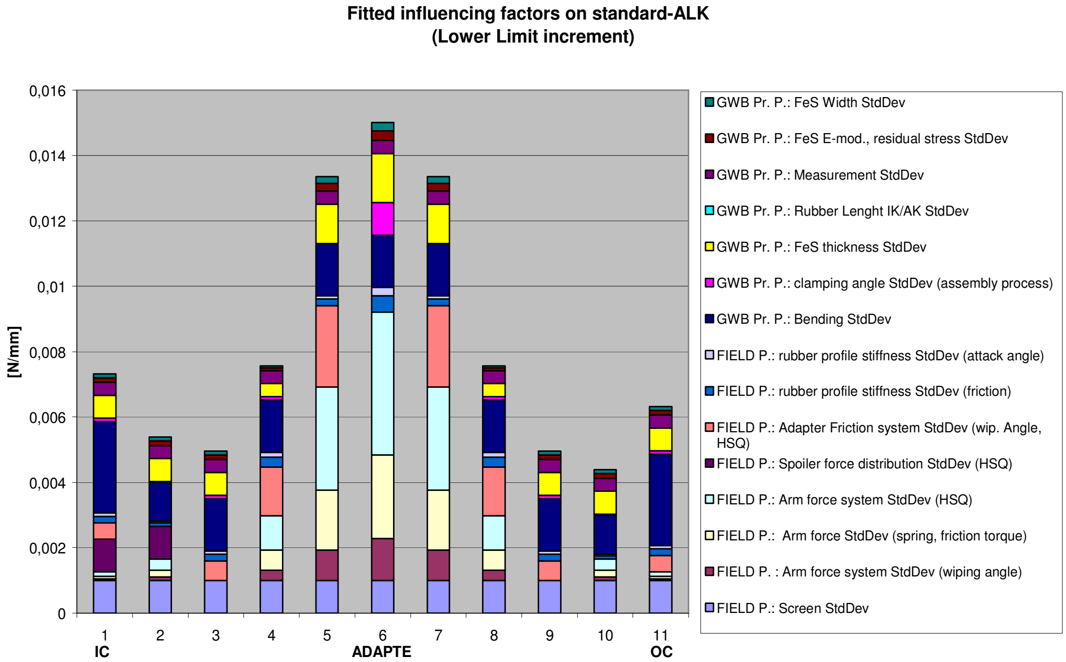


Figure 40: Main influence parameters along "GWB" length for the lower limit increment

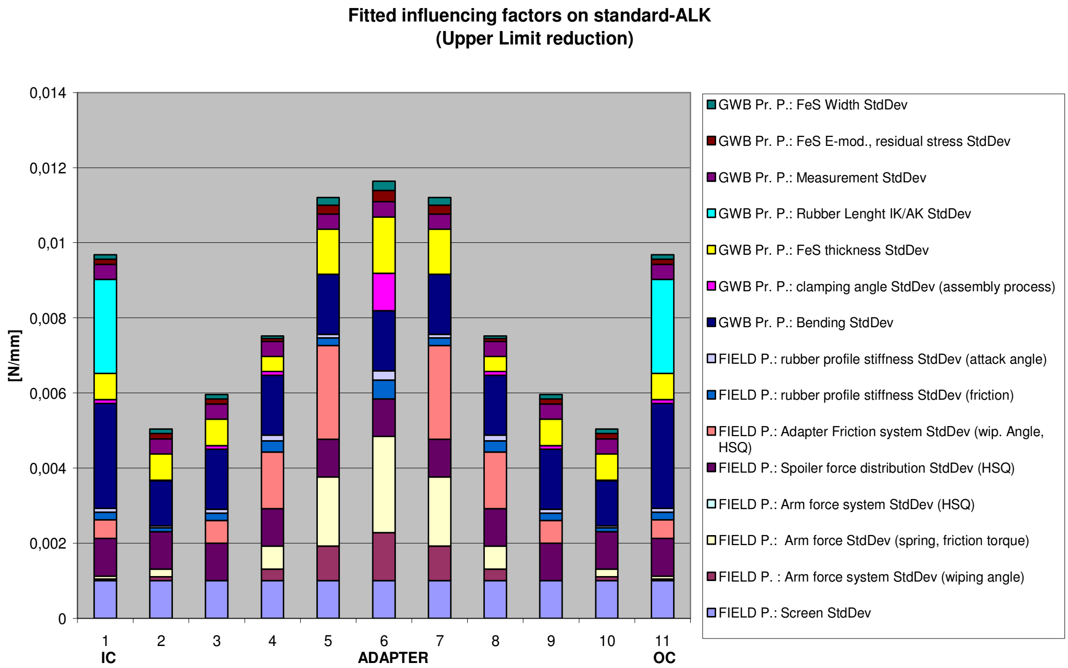


Figure 41: Main influence parameters along "GWB" length for the upper limit reduction

These influenced parameters can be qualified in two distinct ways: as system fields parameters and as *"GWB"* production parameters. Considering this approach it's shown in Figure 42 and Figure 43 which are the most influence parameters concept along the *"GWB"* length for lower and upper limits.

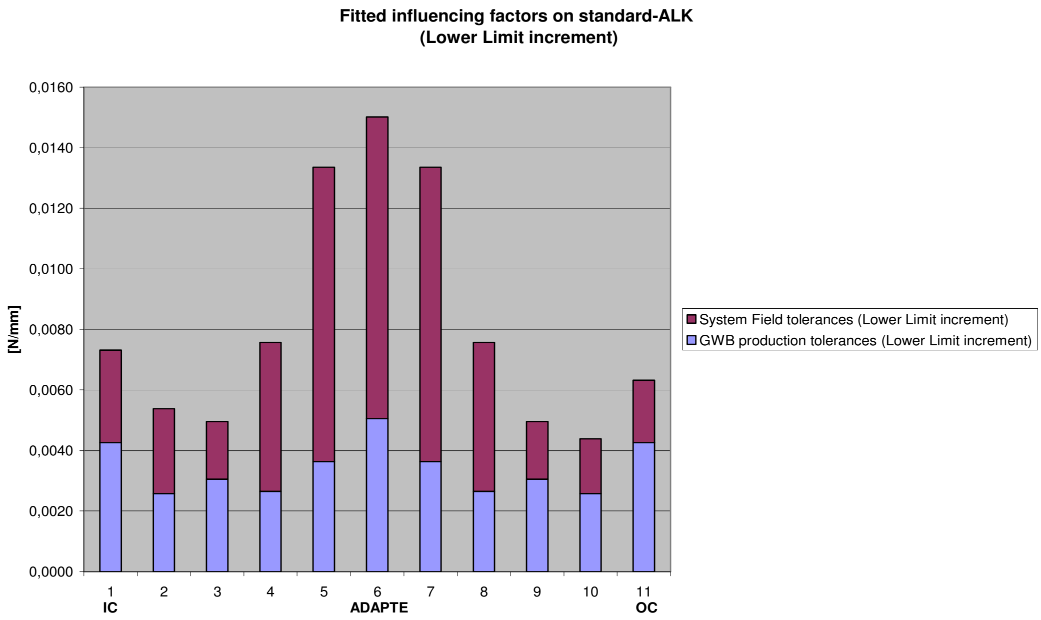


Figure 42: Field influence parameters on standard ALK along "GWB" length for lower increment

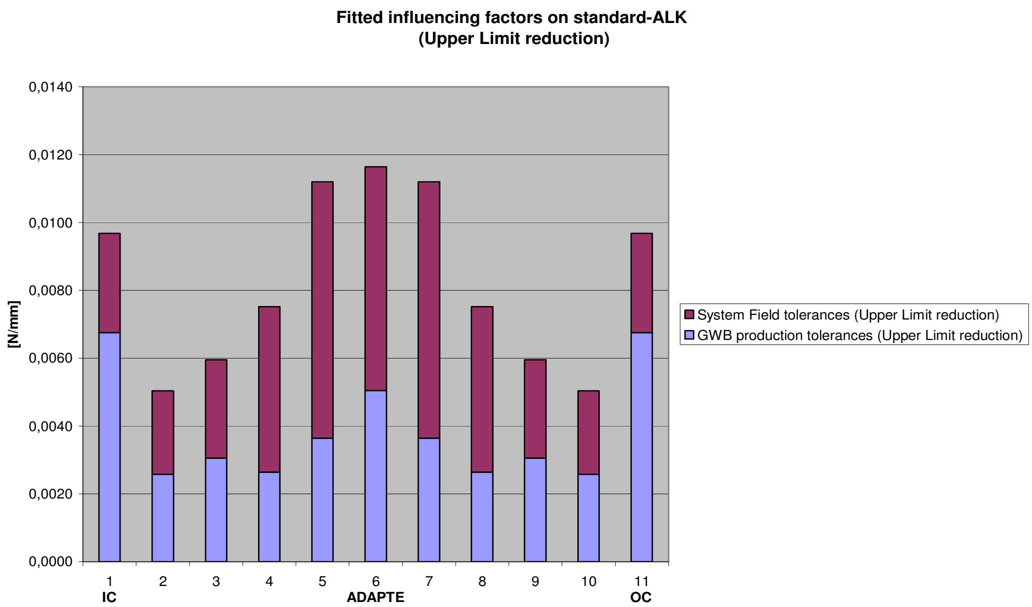


Figure 43: Field influence parameters on standard ALK along "GWB" length for upper increment

Parameter Name
Lower Limit
Upper Limit
FIELD P.: Screen StdDev
FIELD P. : Arm force system StdDev (wiping angle)
FIELD P.: Arm force StdDev (spring, friction torque)
FIELD P.: Arm force system StdDev (HSQ)
FIELD P.: Spoiler force distribution StdDev (HSQ)
FIELD P.: Adapter Friction system StdDev (wip. Angle, HS
FIELD P.: rubber profile stiffness StdDev (friction)
FIELD P.: rubber profile stiffness StdDev (attack angle)
GWB Pr. P.: Bending StdDev
GWB Pr. P.: clamping angle StdDev (assembly process)
GWB Pr. P.: FeS thickness StdDev
GWB Pr. P.: Rubber Lenght IK/AK StdDev
GWB Pr. P.: Measurement StdDev
GWB Pr. P.: FeS E-mod., residual stress StdDev
GWB Pr. P.: FeS Width StdDev

Table 8: List of the main influence parameters along "GWB"

Thus the parameters selected in Table 8 were chosen as more influent. Therefore these parameters are now objected of investigation and simulation to detect the fit parameters values.

In the next chapters is shown the theoretical studies until the moment made concerning the selected parameters areas of study.

3.3.3.1 Bending StdDev

The theoretical value of bending standard deviation σ_h can be determined based on the fitted values $\sigma_k \times \sqrt{\lambda_k}$ experimentally determined in chapter 3.3.1.

Formula 1 (simplex):

$\sigma_{h1}(x) \equiv \sigma_{k1}\sqrt{\lambda_1} \times dw_1$; See RBBE ED-WS/EAB4 (2007a), eq.GI.4.11, page 20.

Formula 2:

$\sigma_{h2}(x) \equiv \sigma_{k2}\sqrt{\lambda_2} \times dw_2$; See Braun (2009c), eq.4.8, page 13.

Based on these theoretical bending standard deviation relations we can finally estimate the theoretical ALK deviation.

When using the fitted value of 0,00025 mm (or 0,00016 mm in case of using theory 2), the ALK deviation with cp=4 is around 2N/m for a 7*1.0 vertebra (in order to calculate the contribute for other vertebra cross sections see equation 3.18 in RBBE ED-WS/EAB4 (2007a)). The resume of the bending standard deviation is show next:

			Bending StdDev
Width & Thickness	6,00 mm	0,80 mm	**1,3 N/mm**
	7,00 mm	0,90 mm	**1,8 N/mm**
	7,00 mm	1,00 mm	**2 N/mm**

Table 9: Standard deviation values for the different "GWB" wiper blades

It's also necessary refer that the ALK behaviour due to bending standard deviation is growing at the inner and outer circle thus the values for this standard deviation are a mean value along the length.

3.3.3.2 Clamping angle StdDev determination (assembly process)

The sensitivity analysis (see Bubba (2007c)) done in the past about this specific issue reflect precisely the importance of controlling this parameter.

In this study is clearly demonstrate that a variation on the assembly process, resulting in different clamping angles, culminate in an ample modification of ALK values in the adapter area (Figure 45).

Figure 44: Scheme of the clamping angle diagram

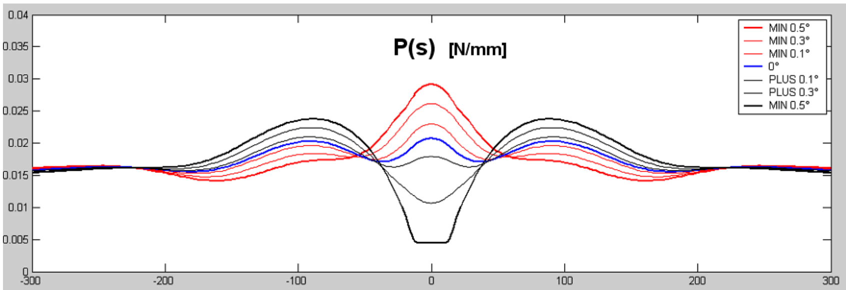


Figure 45: Sensitivity analysis of the clamping angle variation

Due to this reaction in ALK values this *"GWB"* production parameter should be better control.

At this moment the standard deviation estimated for this parameter is 0.2°, but further investigation will continue in order to come across with a more precise value and if possible to achieve a decrease of its standard deviation value.

During the assessment of the clamping angle standard deviation we know that only some other parameters can induce influence in our test results.

Thus, in Table 10 is exhibit those parameters. Some of these parameters are fixed because it's analyse beams from a small batch from the same coil, and they are inputs for the calculation and also because it is a simulation of the behaviour in R4000.

	Within same batch (T1)	Between different batches (T1 → T2) (3 months later)
FIELD P.: Screen StdDev	0 (R4000 high accuracy)	0 (R4000 high accuracy)
FIELD P. : Arm force system StdDev (wiping angle)	0 (given)	0 (given)
FIELD P.: Arm force StdDev (spring, friction torque)	0 (given)	0 (given)
FIELD P.: Arm force system StdDev (HSQ)	0 (given)	0 (given)
FIELD P.: Spoiler force distribution StdDev (HSQ)	0	0
FIELD P.: Adapter Friction system StdDev (wip. Angle, HSQ)	0 (R4000 holder)	0 (R4000 holder)
FIELD P.: rubber profile stiffness StdDev (friction)	0 (friction fixed on R4000)	0 (friction fixed on R4000)
FIELD P.: rubber profile stiffness StdDev (attack angle)	0 (R4000 holder)	0 (R4000 holder)
GWB Pr. P.: Bending StdDev (4 □)	2 N/m for 7x1.0 (EXATLY MONITORED)	2 N/m for 7x1.0 (EXATLY MONITORED)
GWB Pr. P.: clamping angle StdDev (assembly process)	to be determined	to be determined
GWB Pr. P.: FeS thickness StdDev	0	0.75 N/m (1.5 sigma = 1.5*0.5 N/m)
GWB Pr. P.: Rubber Lenght IC/OC StdDev	0	0
GWB Pr. P.: Measurement StdDev	0,0002 [N/mm]	0,0002 [N/mm]
GWB Pr. P.: FeS E-mod., residual stress StdDev	0	0.15 N/m (1.5 sigma = 1.5*0.1 N/m)
GWB Pr. P.: FeS Width StdDev	0	0.12 N/m (1.5 sigma = 1.5*0.8 N/m)

Table 10: Parameter definition and assumption during the experimental tests

During this phase was analyse ALK peak values for each beam in different arm forces and times (T1 and T2). The summary of that study is in shown in Table 11.

Before the investigation of the clamping angle standard deviation, we first had to make some assumptions of our tests. We accept that the variation of the bending can influence till 1.3 N/mm, 1.8 N/mm, 2 N/mm for 6*0.8, 7*0.9, 7*1 beams respectively, also we assume that our population is represented by 3σ as a normal distribution, and thus FeS E-modulus, FeS thickness and FeS width standard deviation of our tests are representing 1.5σ.

Knowing that each of these last parameters have their own standard deviation and their resulting ALK variation, we achieve that in our population tests these parameters together can represent around about 1 N/mm (for more details see chapters 3.3.3.3, 3.3.3.6, 3.3.3.7).

	L (total lenth FeS)	FeS type	Screen type	adapter type	GWB type	Minimum T1 Contribute on CA	Minimum T1 Delta Cl. Angle	Nominal T1 Contribute on CA	Nominal T1 Delta Cl. Angle	Maximum T1 Contribute on CA	Maximum T1 Delta Cl. Angle
Codenumber 1_FeS 3394C08323_ZsB 3398.132.594	1	1	2	1	1	-0,1	------	-0,2	------	-0,4	------
Codenumber 2_FeS 3394C08321_ZsB 3398.132.595	2	2	1	1	1	-0,6	------	-0,5	------	-0,7	------
Codenumber 3_FeS 3394C08319_ZsB 3398.132.596	2	2	2	1	1	-0,6	------	-0,4	------	-0,4	------
Codenumber 4_FeS 3394C08321_ZsB 3398.132.597	2	2	1	2	1	0,4	0,04 [°]	0,3	0,03 [°]	0,3	0,03 [°]
Codenumber 5_FeS 3394C08317_ZsB 3398.132.598	2	2	2	2	1	-0,2	------	0,3	0,03 [°]	0,6	0,06 [°]
Codenumber 6_FeS 3394C08395_ZsB 3398.132.599	3	3	2	2	1	0,8	0,08 [°]	1,4	0,14 [°]	1,5	0,15 [°]
Codenumber 7_FeS 3394C08377_ZsB 3398.132.600	1	1	2	4	2						
Codenumber 8_FeS 3394C08379_ZsB 3398.132.601	4	2	2	4	2						

	L (total lenth FeS)	FeS type	Screen type	adapter type	GWB type	Minimum T2 Contribute on CA	Minimum T2 Delta Cl. Angle	Nominal T2 Contribute on CA	Nominal T2 Delta Cl. Angle	Maximum T2 Contribute on CA	Maximum T2 Delta Cl. Angle
Codenumber 1_FeS 3394C08323_ZsB 3398.132.594	1	1	2	1	1						
Codenumber 2_FeS 3394C08321_ZsB 3398.132.595	2	2	1	1	1	-0,6	------	0,1	0,01 [°]	0,3	0,03 [°]
Codenumber 3_FeS 3394C08319_ZsB 3398.132.596	2	2	2	1	1	0,7	0,07 [°]	0,6	0,06 [°]	0,5	0,05 [°]
Codenumber 4_FeS 3394C08321_ZsB 3398.132.597	2	2	1	2	1						
Codenumber 5_FeS 3394C08317_ZsB 3398.132.598	2	2	2	2	1						
Codenumber 6_FeS 3394C08395_ZsB 3398.132.599	3	3	2	2	1	2,7	0,27 [°]	2,2	0,22 [°]		
Codenumber 7_FeS 3394C08377_ZsB 3398.132.600	1	1	2	4	2						
Codenumber 8_FeS 3394C08379_ZsB 3398.132.601	4	2	2	4	2						

Table 11: ALK peak analyse for clamping angle StdDev determination

In such a way is possible to estimate the ALK variation due to clamping angle by the difference between the measured peak results less the loss due to FeS E-modulus, FeS thickness and width deviations.

- Standard deviation for clamping angle: **0.18°**

3.3.3.3 FeS thickness StdDev determination

As we know the FeS is manufactured by one of our suppliers and after the raw material arrives to our supplier and then it is store as coils. But those coils meet the standard quality requirements?

What was found during this project was that there is no quality control about the geometry of the FeS by Bosch. It was also found out that there are no constant characteristic reports by the supplier to Bosch that information are only delivery when asked.

Because the geometry of the FeS is one important parameter, it's suggested to improve that relationship between BOSCH Group and supplier.

With the constant monitoring of the geometry we believe that it's possible to keep under control this parameter and to better understand the root causes of its possible deviations.

From theoretical sensitivity studies (see Bubba (2007c)) we know how this FeS thickness deviation influences the ALK (Figure 46, Figure 47).

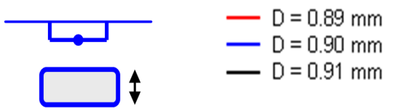


Figure 46: Scheme of the FeS thickness diagram

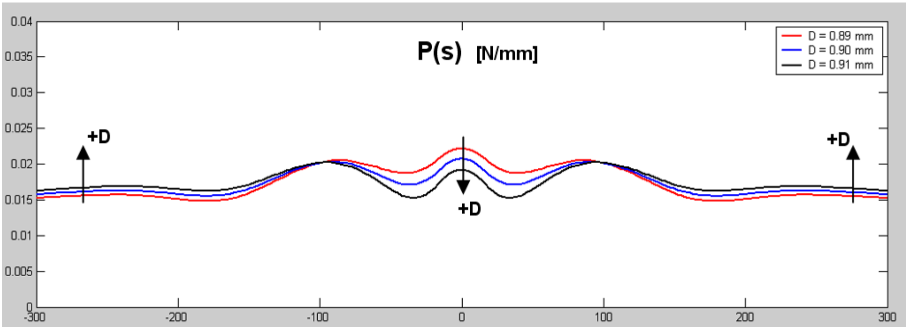


Figure 47: Sensitivity analysis of the FeS thickness variation

Thus it is important to control the FeS thickness standard deviation of the supplier because these bring us more chance to get satisfactory results in terms of ALK for Aerotwin wiper blades.

Therefore the RBBE/PUQ department has been asked to request data about this thickness issue from the FeS supplier. Unfortunately data was not available at the present time.

For the analysis of this standard deviation we had to make some pre-assumptions based on the drawing tolerances. We assumed that the drawing tolerance of +/- 0.01mm corresponds to +/- 3σ process control.

Since, from sensitivity studies, we know that every 0.01mm (3σ) reflects an ALK variation of 1.5 N/m, and if we assume that our tests represent a population of 1.5σ (80%), then we achieve a specific value for the FeS thickness:

- Standard deviation for FeS thickness: **0.75 N/m**

3.3.3.4 Rubber length IC/OC StdDev determination

From the sensitivity analyses made (see Bubba (2007c)) and resumed in Figure 48 and Figure 49 for symmetric rubber length, and Figure 50 and Figure 51 for asymmetric rubber length, it's clear that the non-control of the rubber length influence the behaviour in ALK in inner and outer circle.

The behaviour is simple: as shorter the rubber length as higher the ALK is in IC/OC. If the rubber length is bigger than the FeS length, no significant ALK deviation due to the very low bending stiffness of the rubber.

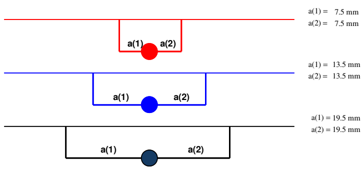


Figure 48: Scheme of the symmetric rubber length IC/OC diagram

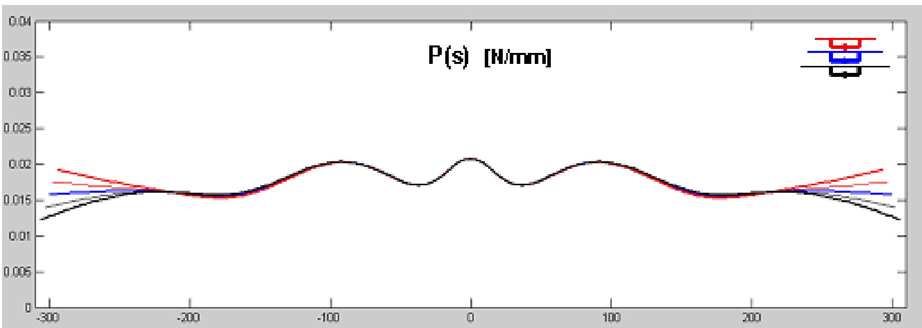


Figure 49: Sensitivity analysis of the symmetric rubber length variation

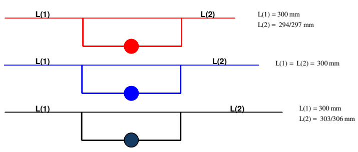


Figure 50: Scheme of the asymmetric rubber length IC/OC diagram

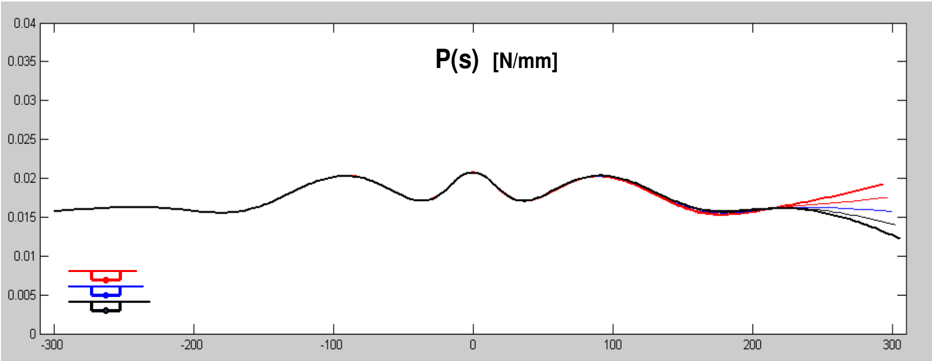


Figure 51: Sensitivity analysis of the asymmetric rubber length variation

In order to predict ALK deviations due to rubber lengths, EAB4 needed statistic lengths measurements from the rubber runners. This information was not easily available and only after meetings with different departments (EAB6, EAB5, CLP2, MOE5, QMM7) some basic information could be taken.

We took as population 3 different rubber lengths (400mm, 550mm, and 700mm) of the extruded FX rubber measured in a two years time span:

400mm, FX rubber → 3398119081; (2 M part produced);
550mm, FX rubber → 3398119331; (2.1 M parts produced);
700mm, FX rubber → 3398116957 (1.2 M parts produced).

The results given for QMM7 for those wiper blades number are displayed in the next figures. Attached to the pictures there are some statistic values of interest such as the number of samples (“Aantal”), the mean (“Gemiddelde”), the sigma (“standard deviation”), the process performance (“Ppk and Pp”).

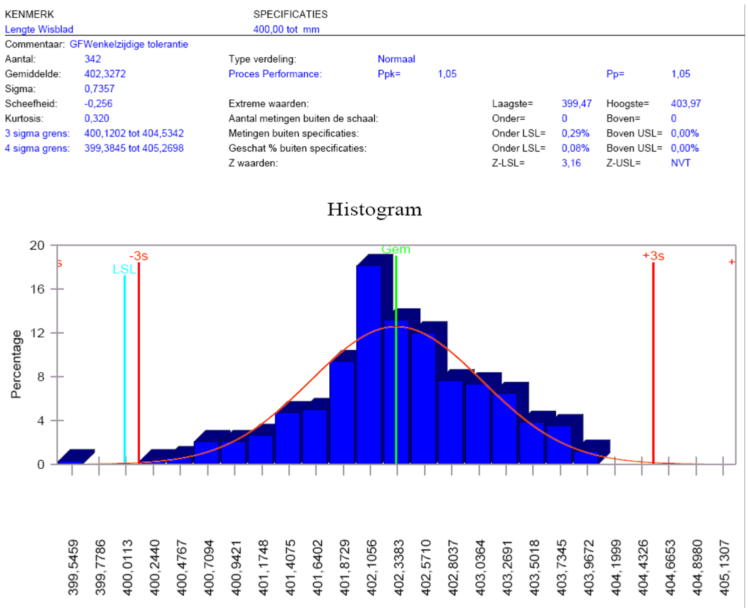


Figure 52: Statistic results for the wiper blade 3398119081(400mm)

The statistic results for the 550mm and 770mm wiper blades can be consulted in Annex C.

At this moment it's clear that, for the rubber length chosen, the distribution is not exactly normally distributed and that the Ppk and Pp values are around 1.1. Here below is the summary for the three examined lengths:

σ	3σ	L	%L
0,736	2,207	400	0,55%
1,052	3,156	550	0,57%
1,191	3,573	700	0,51%

Table 12: Summary of the three examined lengths

It was also seen that the specified tolerance fields, for the 400mm and 550 mm lengths, were always above the nominal FeS lengths (thus no significant ALK deviation when no local sticking effects happen) while for the 700 mm length the rubber may be smaller than the nominal vertebra length (thus possible increment of ALK). Thus it is not clear what were the criteria that inspired the tolerance definition and it is thus not evident to establish common rubber length deviation criteria for the whole product line. Therefore, further investigation is here necessary.

For the time being, assumptions have to be made: if we assume that the rubber length may be shorter than the FeS length up to 1.5σ (similarly to the 700 mm length) than we can consider a typical rubber length reduction of 0.25% of the nominal length (from sensitivity study we have a typical sensitivity of 0.8N/m/m for each mm of length deviation for a 7*0.9 mm FeS):

- Standard deviation for the rubber length: **0.25% * L**

3.3.3.5 Measurement equipment (R4000) StdDev determination

It is obvious that the deviation due to the measurement equipment have to be taken into account. From TEF 1.3 investigations:

- Standard deviation for Measurement equipment (R4000): **4*0.0002 N/mm**

3.3.3.6 FeS E-modulus and residual stresses StdDev determination

From sensitivity analysis (see Bubba (2007c)) the influence of E-modulus deviations onto ALK is known and it is schematically shown Figure 54. More complex is the influence of residual stresses, which is currently not taken into account and has to be further investigated.

Figure 53: Scheme of FeS E-modulus variation

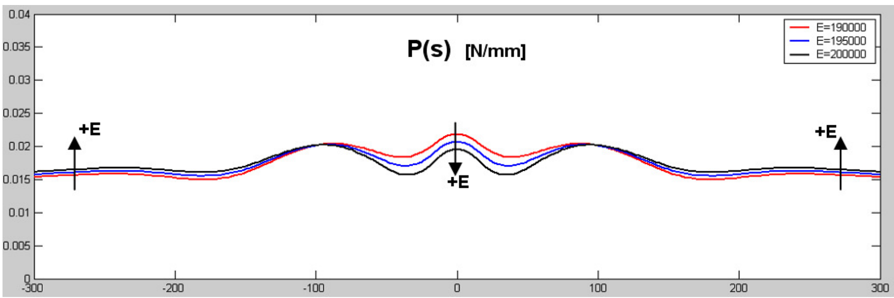


Figure 54: Sensitivity analysis of the FeS E-modulus variation

We see that the variation of this parameter produce changes of ALK also in the adapter area, thus is extremely important to take into account this variation even though being small.

Thus we know that each 1% of E-modulus variation represents a ALK variation of 0.3 N/mm. Assuming again that 1% E-modulus deviation represents 3σ population and that our population test is located at 1.5σ (80%), then the standard deviation of the FeS E-modulus is found.

- Standard deviation for FeS E-modulus: **0.15 N/mm**

3.3.3.7 FeS width StdDev determination

In the determination of the standard deviation of this parameter we found same identical problems as for the FeS thickness StdDev determination chapter (see page 57, no statistics available since the parameter is not monitored by QMM). Since we buy from a certain supplier the FeS coils, we should make sure that we control the geometry (width and thickness) of FeS. The quality control isn't done by BOSCH and quality reports from the supplier aren't imperative.

As for the FeS thickness StdDev, sensitivity studies (see Bubba (2007c)) shows which is the effect onto ALK distribution of FeS width deviations (see Figure 56).

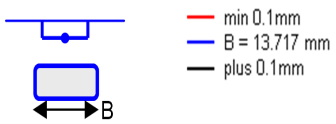


Figure 55: Scheme of FeS width variation

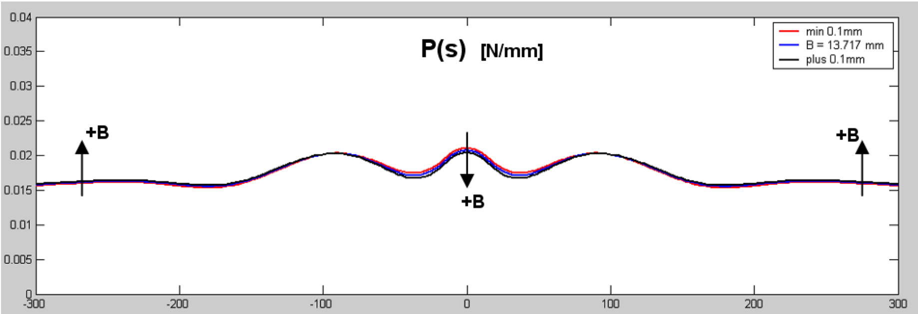


Figure 56: Sensitivity analysis of the FeS width variation

For the analysis of this standard deviation, we had thus to make some pre-assumptions based on the drawing tolerances as done for the FeS thickness. We assumed that the drawing tolerance of +/- 0.1mm corresponds to +/- 3σ process control. Since, from sensitivity studies, we know that every 0.1mm (3σ) reflects a ALK variation of 0.25 N/m, if we assume that our tests represent a population of 1.5σ (80%), then we achieve a specific value for the FeS thickness standard deviation of:

- Standard deviation for FeS width: **0.12 N/mm**

4. Final conclusions and future proposals

This final chapter mentions the main conclusions from the internship period and from the project performed. Will also be suggested some further work to the simulation software development.

4.1 Final conclusion

Succinctly, the 8 months internship and its objectives at Robert Bosch - Belgium was a success. The author inclusion in the research department of Bosch Belgium corresponded, in terms of knowledge and personal and technical competence, to what was expected by officials from the department. Concerning the objectives proposed for the internship the author helped significantly the development of several projects.

For the author, this work experience, allowed the perception of the reality of working in a multinational company. It was also tested communication skills that are paramount to success within this organization and were also tested personal skills and professional know-how techniques for carrying out several projects.

Regarding the first objective of the project, it is concluded that the study of the software life cycle was valuable and therefore was accepted by officials of EAB4 department. Thus, the suggested standard structure of software life cycle was applied to create the "*GWB*" handbook for guiding the development of the simulation software.
Related to the "*GWB*" handbook, professionals responsible for the project are of the opinion that this document is an advantage for the proper development of simulation software. The document corresponds to the request, and was clearly organized bringing a greater probability to the success of the simulation software implementation. On this matter, was only detected a problem during the initiation of this activity. This problem has to do with the delay of documentation in relation to the progress and development ever conducted to date.

The validation process of the robust design module of the simulation software prototype was also performed successfully witch concerns to the second objective of the internship.

Several parameter fit values and standard deviations values were found that enable better calculation accuracy by the robust design module of the "*GWB*" software prototype.

Despite the success in the course of the tests were discovered some problems. These failures relate primarily to errors in the programming code used. Therefore, led to the correction of those and was performing the verification of the absence of the failure.
To carry out various tasks, the author of this project was to collect very specific data in different departments of the factory. So was found another difficulty throughout the project: the communication between departments that mainly do not usually work together. This connection proved to be a long and tricky.

After completing the required activities, this project has boosted the temporarily stalled activities of the software development causing initiation of other tests and actions.

4.2 Future proposals

The first note of this sub-chapter is related to the software life cycle activity. More precisely, it is proposed to follow the steps within the "*GWB*" handbook for the continuity of the development process of simulation software. To all future work is suggested to be carried out simultaneously with all the necessary documentation so that no information is lost and for the record all procedures used.

In relation to validation tests, the suggestions can be divided into several themes. A list and brief description of these suggestions are showed next:

1. WSQ and HSQ tests:
   It is suggested that for the same set of wiper blades it must be perform the validation of wiping quality tests as well as high speed wiping quality tests.

2. T2 - Time results:
   It must be analyzed the T2 results concerning the same issues investigated in this document and also to the tests in the previous suggestion.

3. Remaining modules:
   There should be continuity of research in other areas to achieve the development of all software prototype modules. Later, must be done the testing plan and the validation test to all modules: screen data module, screen analysis module, advanced module, check existing beams module.

Finally, it shouldn't be forgotten that the wiper blades S2 behaviour appeared to suffer from a strange phenomenon during the tests. Thus it's suggested that these events are investigated and solved. If these objectives are achieved, validation tests should be repeated for S2 wiper blades.

References

AMARAL, L., VARAJÃO J. (2007), *Planeamento de sistemas de informação*, FCA, [S.L.]

ABRAN, A., MOORE, J. W. (2004) (Executive editors), *SWEBOK® Guide to the software engineering body of knowledge*, Bourque, P., Dupuis, R., Tripp, L. L. (editors). Accessed on 4 October 2009 on the website: Institute of Electrical and Electronics Engineers on http://www.computer.org/portal/web/swebok/

BARSOTTI, R. (2007), *2D static response of wiper blades*, RBBE internal document

BRAUN, H. (2009a), *Equations*, RBBE internal document, EB-09-999

BRAUN, H. (2009b), *Summary*, RBBE internal document, EB-09-999

BRAUN, H. (2009c), *Tools*, RBBE internal document, EB-09-999

BRAUN, H. (2009d), *Validation*, RBBE internal document, EB-09-999

BUBBA, M. (2007a), *2D equations wiper blade simulation model*, RBBE internal document, EB-07-181

BUBBA, M. (2007b), *"GWB" simulation 2D equations* Status performances, RBBE internal document, Folie 6586

BUBBA, M. (2007c), *"GWB" simulation sensitivity study*, RBBE internal document, Folie 6470

BUBBA, M. (2007d), *Rubber profile simulation* Internship results, RBBE internal document, EB-07-144

BUBBA, M. (2007e), *Wisbaarheid beoordeling en FeS uitlegging*, RBBE internal document, Ti-16.2-004_BUM_ver5.0

BUBBA, M. (2008f), *"GWB" simulation extension*, RBBE internal document, Folie 6747

BUBBA, M. (2008g), *"GWB" tools robust design integration*, RBBE internal document, Folie 6769

BUBBA, M. (2008h), *Matlab not-linear analytical model vs 3D FEM Abaqus model*, RBBE internal document, Folie 6665

BUBBA, M. (2008i), *Status wipeability prediction and screeen tolerancence method for OEM and screen suppliers*, RBBE internal document, EB-08-168

RBBE ED-WS/EAB4 (2004), *Wiping quality on aged rubber*, RBBE internal document, EB-04-311

RBBE ED-WS/EAB4 (2007a), *ALK tolerance concept* Abschlußbericht SES, RBBE internal document, EB-07-181

RBBE ED-WS/EAB4 (2007b), *Contact force range for good wiping quality*, RBBE internal document, Folie 4247

RBBE ED-WS/EAB4 (2007c), *Rubber profile tip pressure*, RBBE internal document, Folie 4095

RBBE ED-WS/EAB4 (2007d), *Rubber profile working angle for good wiping quality*, RBBE internal document, EB-07-099

PRESSMAN, R. S. (2001), *Software engineering A practical approach*, fifth edition, The McGraw-Hill, [S.L.]

PROFOZICH, D. M. (1998), *Managing change with business process simulation*, Prentice Hall PTR, [S.L.]

ROYCHOUDHURY, A. (2009), *Embedded systems and software validation*, Morgan Kaufmann, [S.L.]

SARGENT, R. G., (2007), *Verification and validation of simulation models*, Accessed on 5 October 2009 on the website of: INFORMS Simulation Society on http://www.informs-sim.org

SCHACH, S. R. (2008), *Object-oriented software engineering*, Internacional Edition, The McGraw-Hill, [S.L.]

SOKOLOWSKI, J. A., BANKS, C. M. (2009), *Principles of modeling and simulation A multidisciplinary approach*, John Wiley & Sons, Inc., New Jersey, United States of America

VERMESAN, A., COENEN, F. (1999), *Validation and verification of knowledge based systems Theory, tools and practice*, Kluwer Academic Publishers, Dordrecht, Holanda

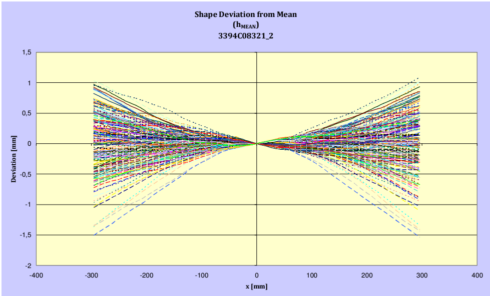
VINCKENROYE, L. V. (2009), *Contact pressure wiper rubbers*, RBBE internal document, Folie 7170

WHITTEN, J. L., BENTLY, L. D., DITTMAN, K. (2009), *System analysis & design methods*, sixth edition, The McGraw-Hill, [S.L.]

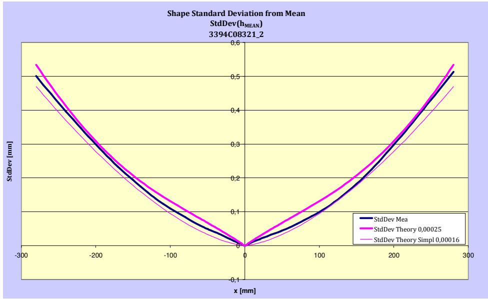
Annexes

Annexes A

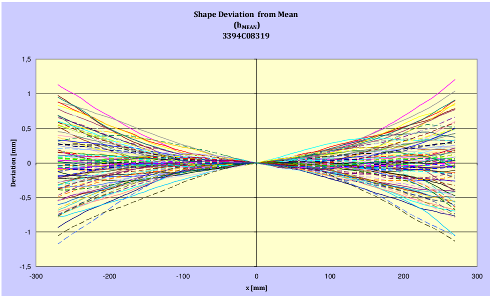
Vertebra bending shape as compared to real measurements



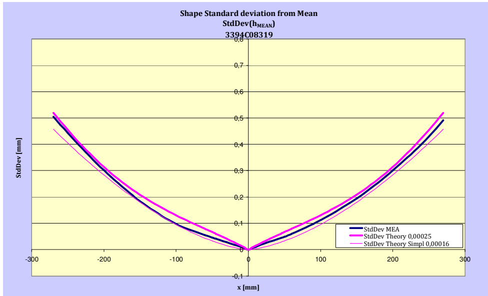
Shape deviation and Standard deviation from the Mean for C08321_2



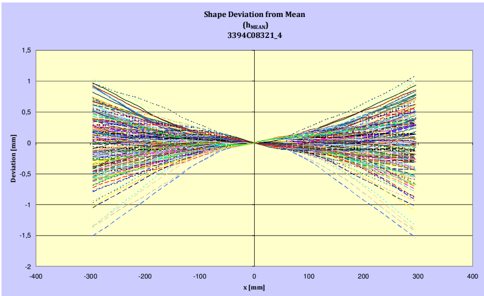
Standard deviation from the Mean for C08321_2



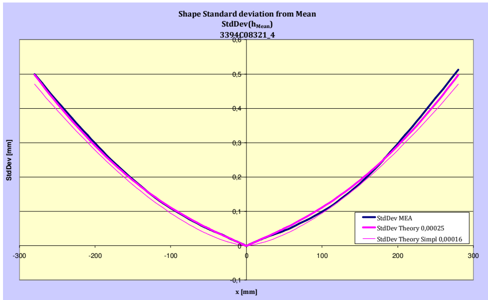
Shape deviation and Standard deviation from the Mean for C08319



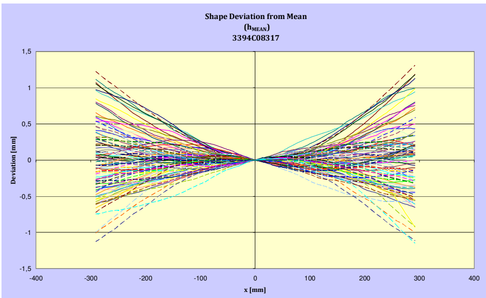
Standard deviation from the Mean for C08319



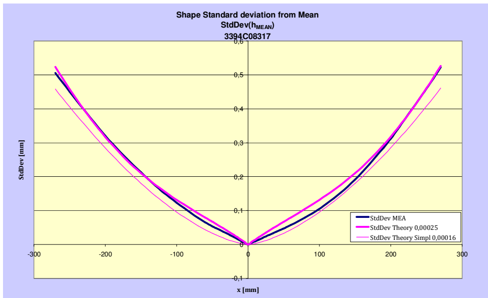
Shape deviation and Standard deviation from the Mean for C08321_4



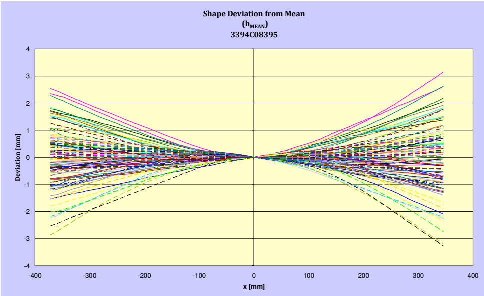
Standard deviation from the Mean for C08321_4



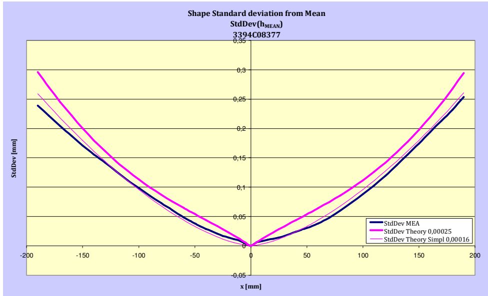
Shape deviation and Standard deviation from the Mean for C08317



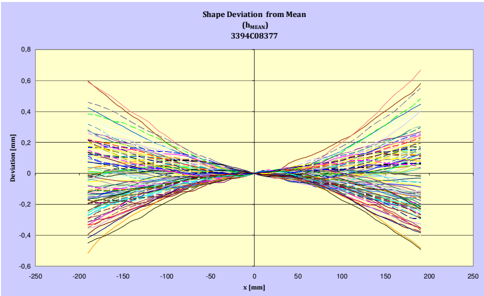
Standard deviation from the Mean for C08317



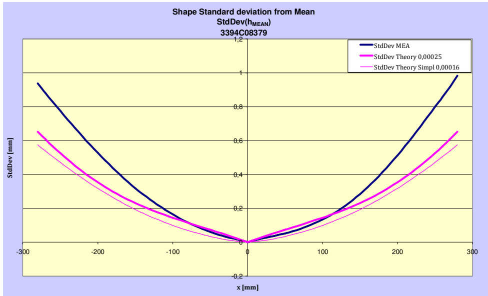
Shape deviation and Standard deviation from the Mean for C08395



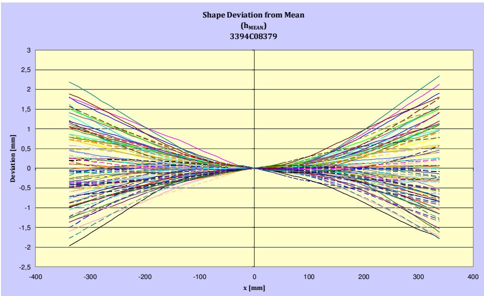
Standard deviation from the Mean for C08377



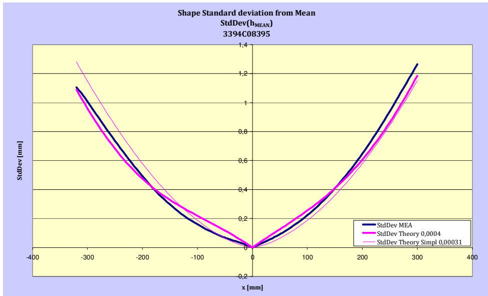
Shape deviation and Standard deviation from the Mean for C08377



Standard deviation from the Mean for C08379



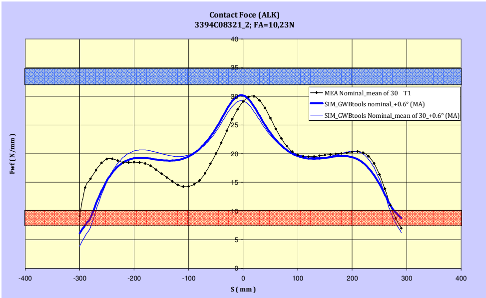
Shape deviation and Standard deviation from the Mean for C08379



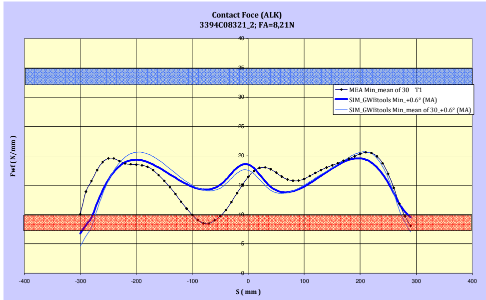
Standard deviation from the Mean for C08395

Annexes B

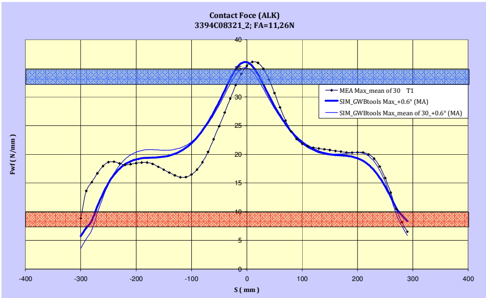
B – 1: Contact force simulation as compared to real measurements on R4000



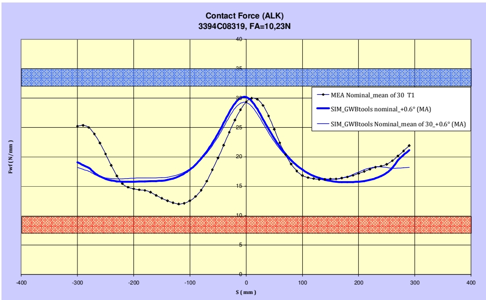
ALK for Nominal force for C08321_2



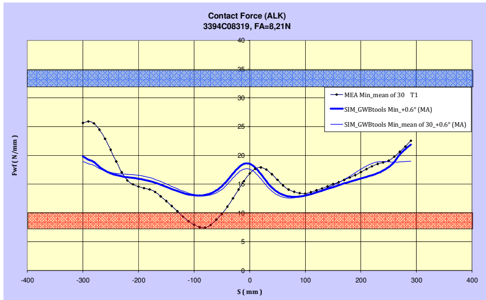
ALK for Minimal force for C08321_2



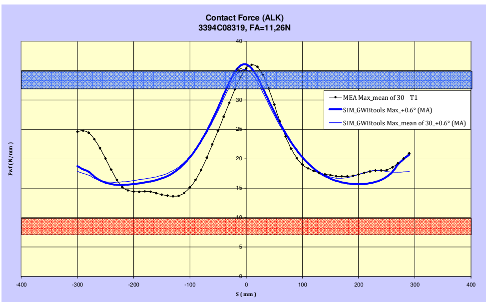
ALK for Maximum force for C08321_2



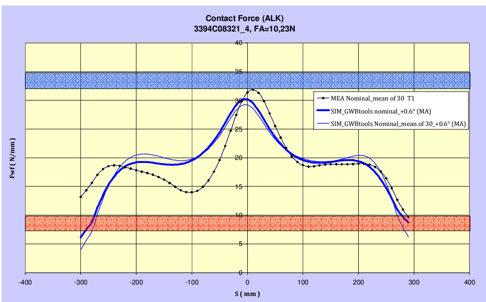
ALK for Nominal force for C08319



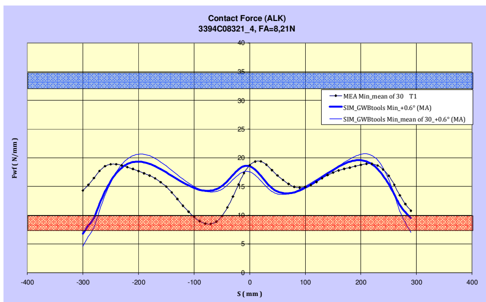
ALK for Minimal force for C08319



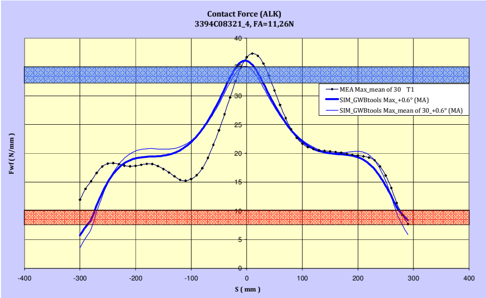
ALK for Maximum force for C08319



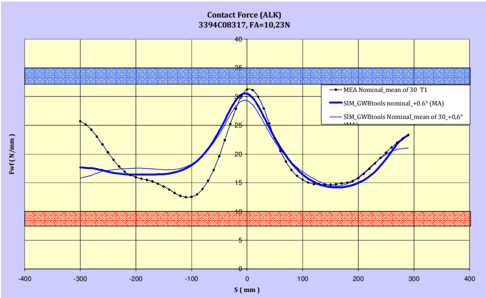
ALK for Nominal force for C08321_4



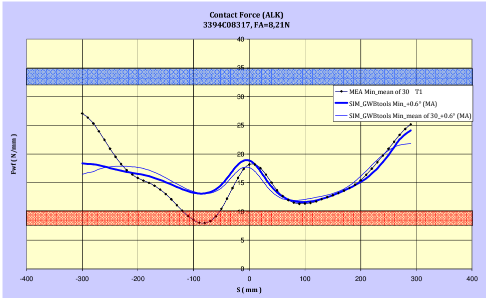
ALK for Minimal force for C08321_4



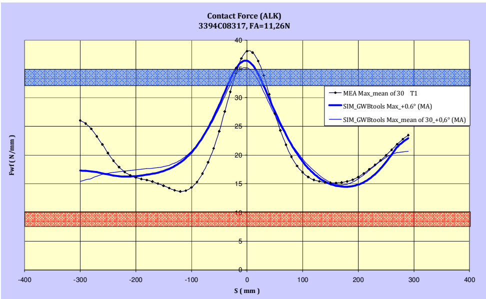
ALK for Maximum force for C08321_4



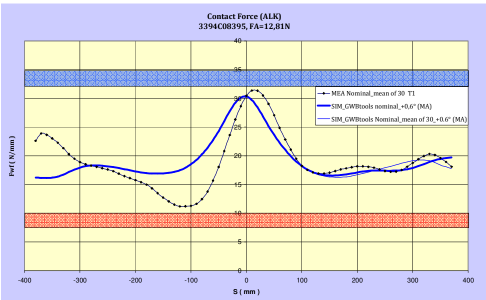
ALK for Nominal force for C08317



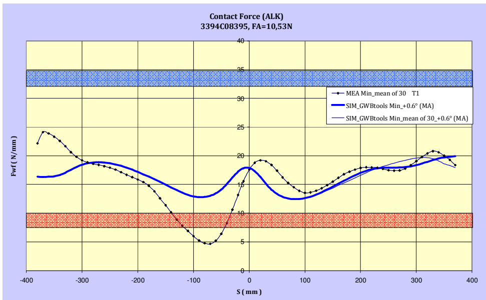
ALK for Minimal force for C08317



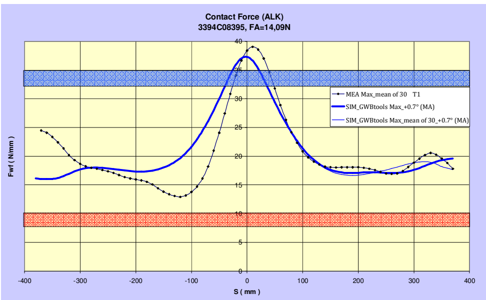
ALK for Maximum force for C08317



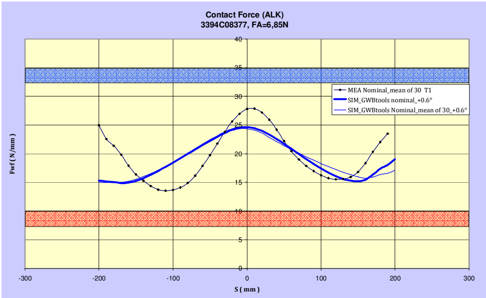
ALK for Nominal force for C08395



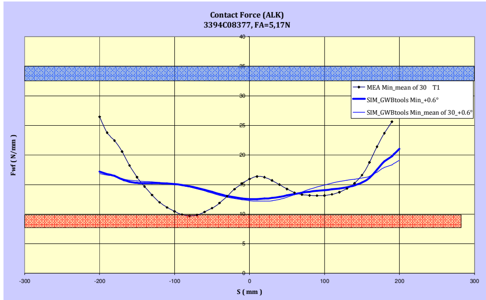
ALK for Minimal force for C08395



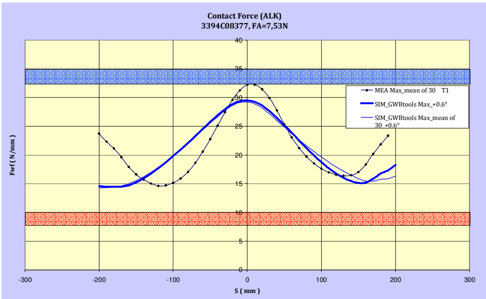
ALK for Maximum force for C08395



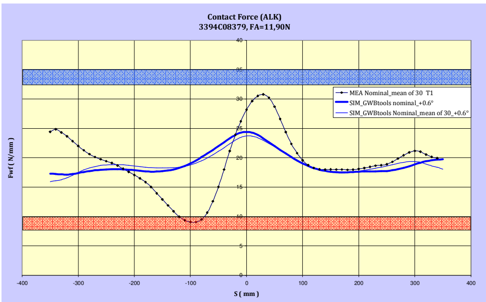
ALK for Nominal force for C08377



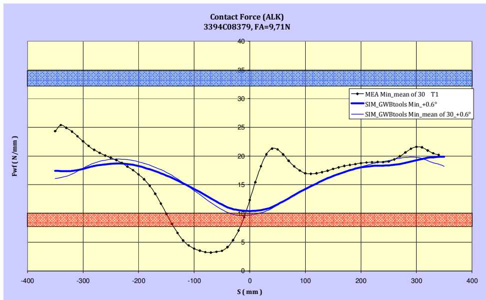
ALK for Minimal force for C08377



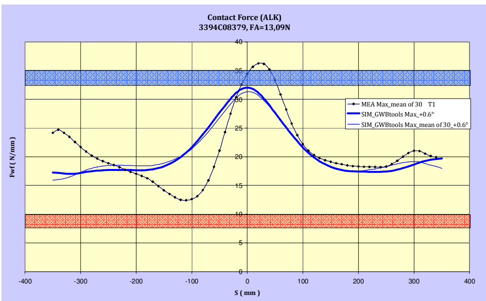
ALK for Maximum force for C08377



ALK for Nominal force for C08379



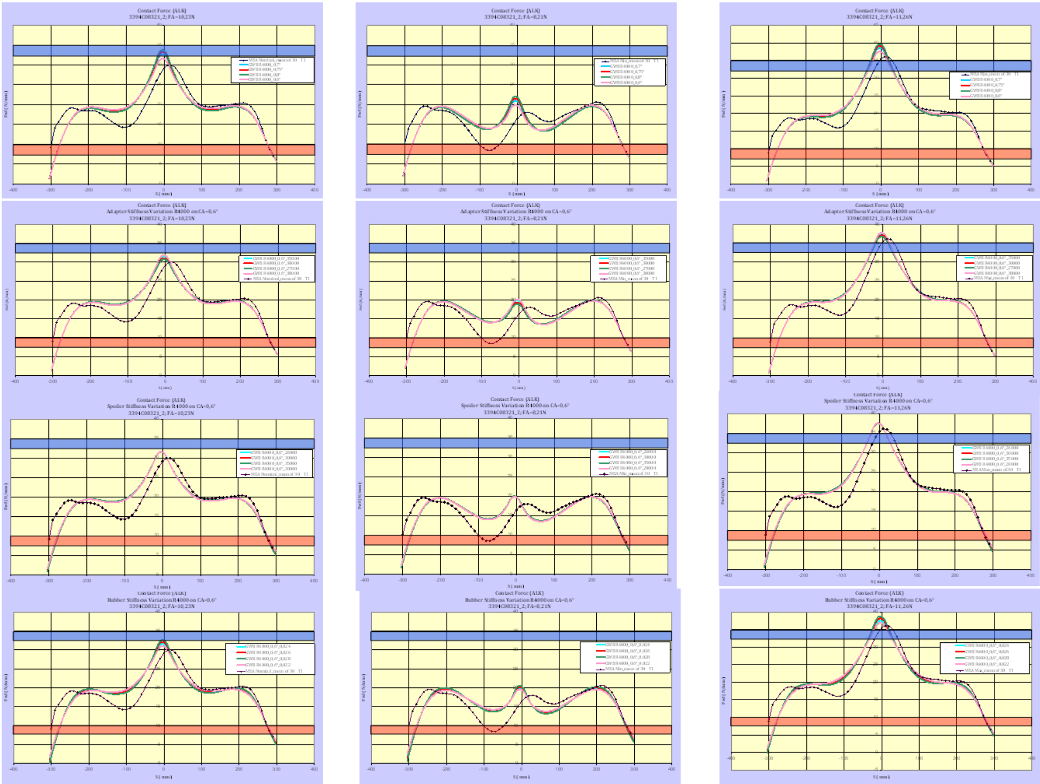
ALK for Minimal force for C08379



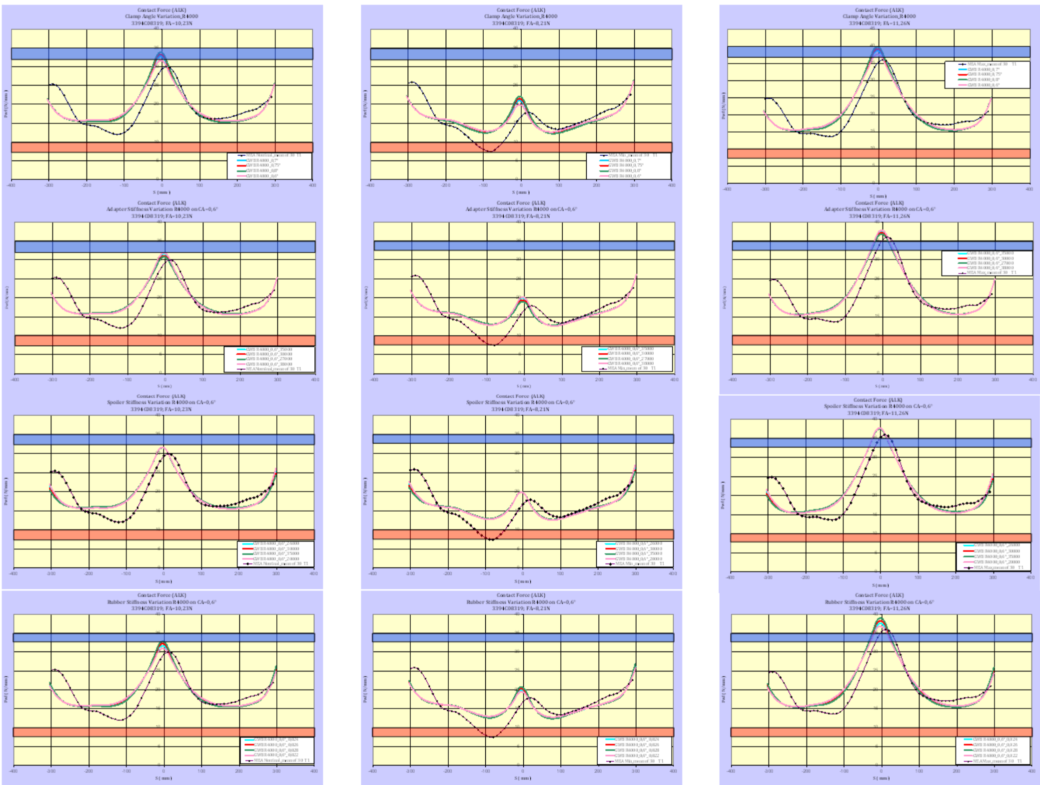
ALK for Maximum force for C08379

Annexes B

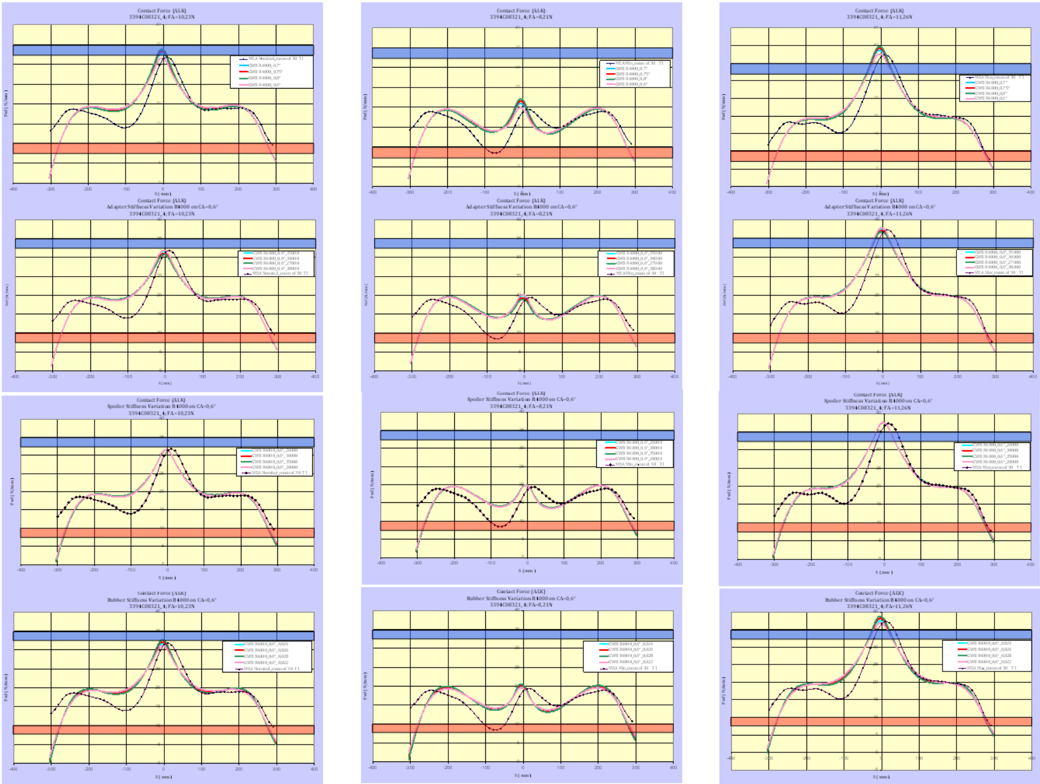
B – 2: Contact force simulation as compared to real measurements on R4000



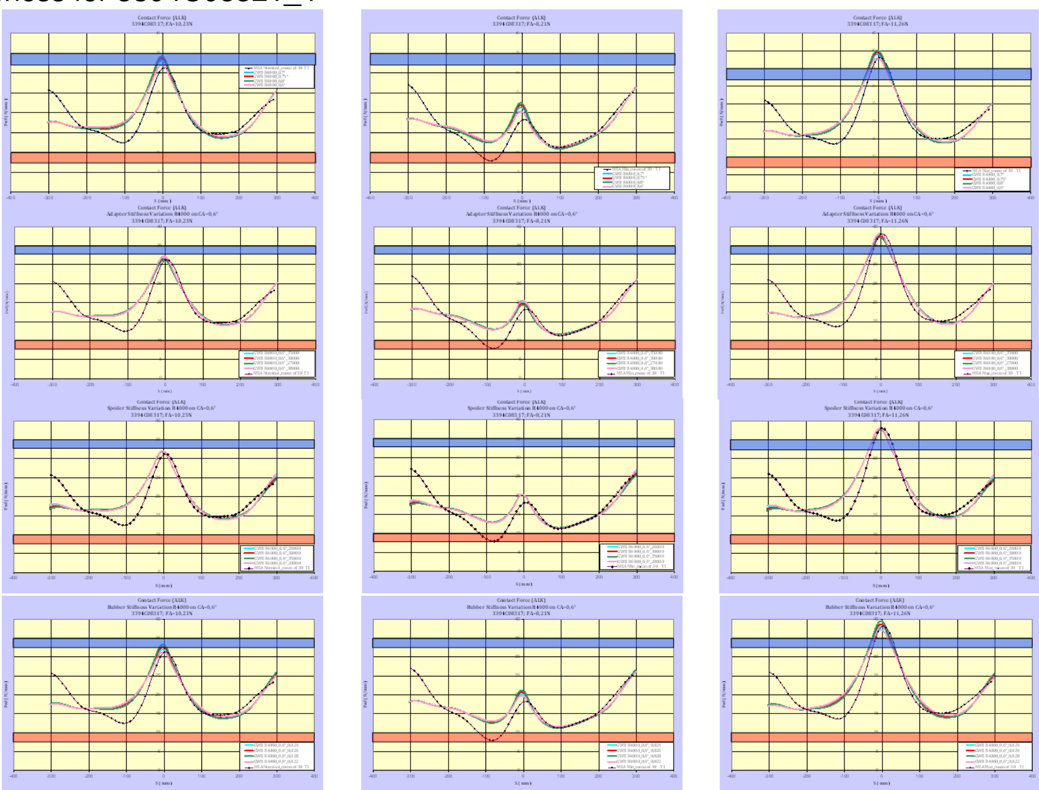
Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08321_2



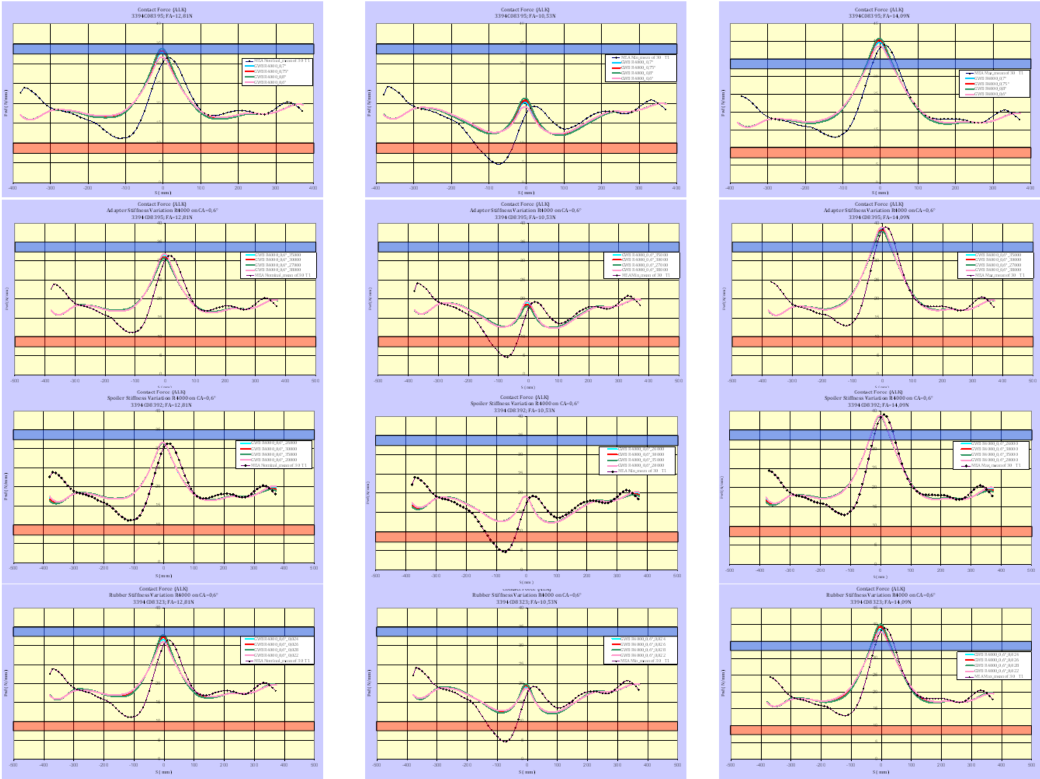
Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08319



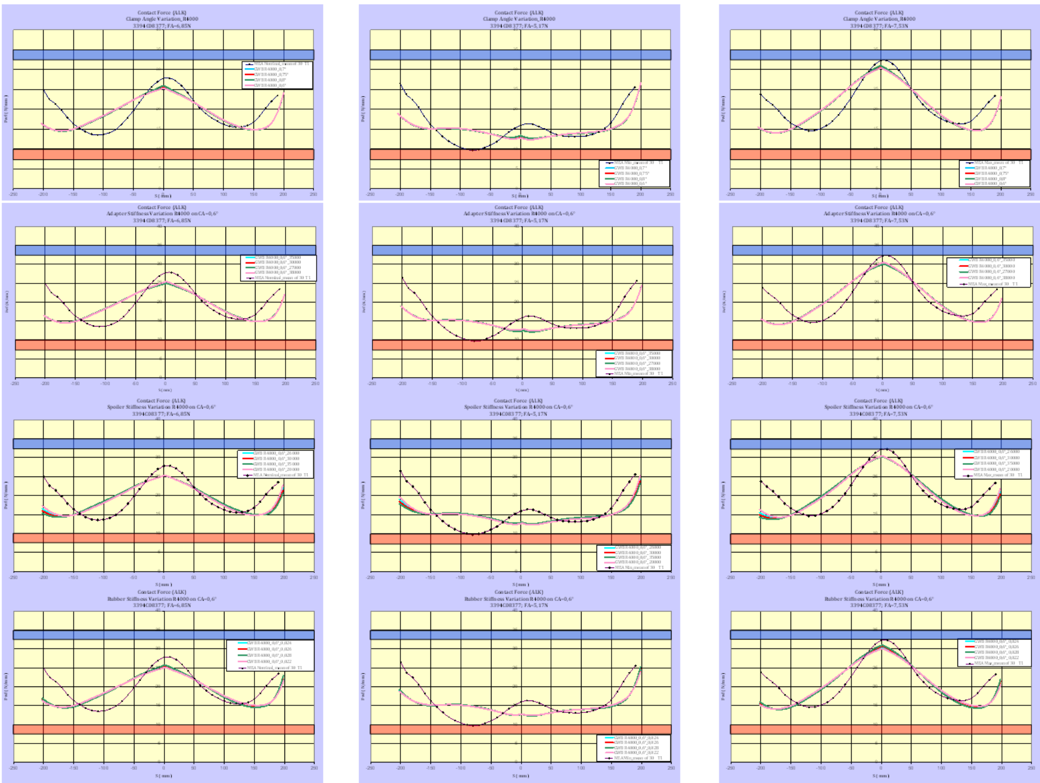
Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08321_4



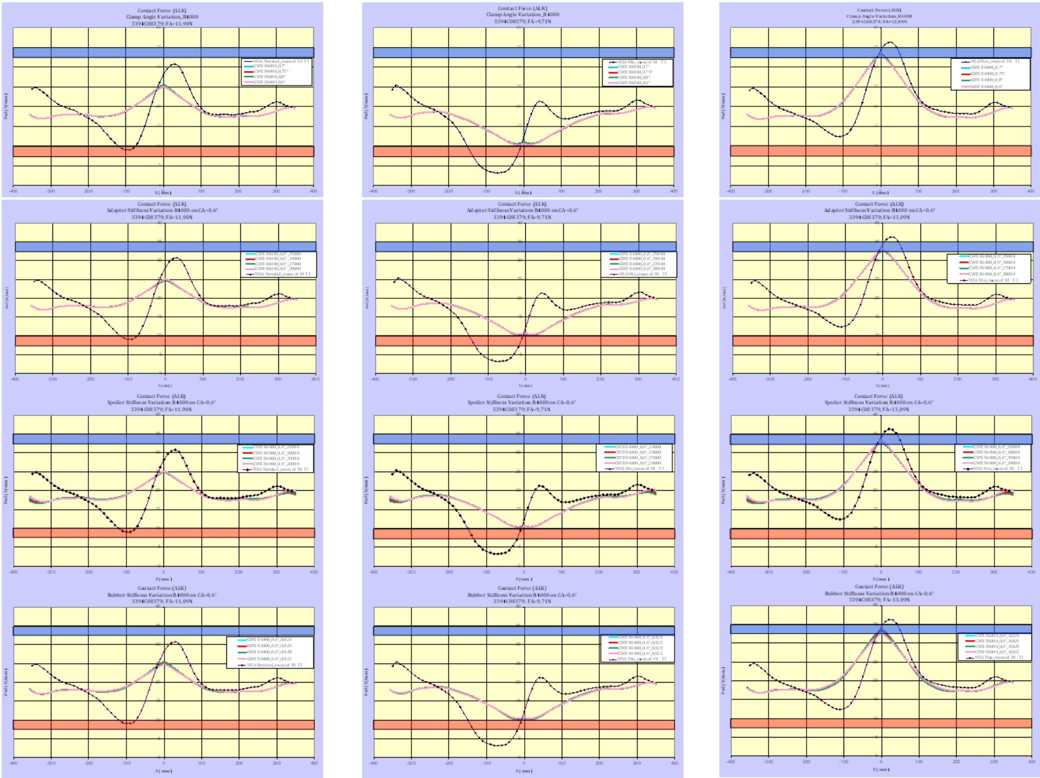
Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08317



Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08395



Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08377



Sensitivity analysis for clamping angle, adapter stiffness, spoiler stiffness, rubber stiffness for 3394C08379