T U M

I N S T I T U T F Ü R I N F O R M A T I K

Result of the Tool Questionnaire

Markus Herrmannsdoerfer and Stefano Merenda

TUM-I0929
November 09

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-11-I0929-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

About this Document

The purpose of the central project is to provide fundamental ingredients for the SPES project as a whole. Due to the huge size of the project, we decided in work package 1 to distribute a questionnaire to all industrial project partners. The main goal of this questionnaire was to get an overview over the situation and the needs of model-based development in industry today. In this questionnaire, we concentrate on questions regarding technologies and tools to develop embedded systems. Although these questions focus on technical issues, we believe that the results also reflect the methods used today. However, questioning about the methods is more difficult than asking for the tools.

According to the results, the most important issues today are that tools and technologies are considerably heterogeneous, that model-based development is not yet used to its full extent, and that existing tools are not yet fully appropriate for efficient model-based development. The most important need is the integration of existing tools for seamless model-based development. The results provide a valuable input to work package 1 to ensure that the most important needs of the industry are taken into account.

Outline. We explain the design of the questionnaire in terms of the underlying research questions in Section 1. In Section 2, we present the results for each question asked by the questionnaire. We provide an overall answer to the posed research questions in Section 3. In Section 4, we discuss threats to the results' validity, before we conclude in Section 5. Section A of the appendix includes the distributed version of the questionnaire.

Acknowledgments. We are thankful to Martin Feilkas, Florian Hölzl and Wolfgang Schwitzer for helping us to develop the questionnaire. Furthermore, we like to thank Martin Leucker and Bernhard Schätz for feedback on earlier versions of the questionnaire. We are also grateful to Judith Thyssen and Daniel Ratiu for helpful suggestions to improve this document. Finally, we like to thank Sabine Nunnenmacher from Fraunhofer IESE and Liliana Guzman from TU Kaiserslautern for an external review of the document.

Contents

1 Design

In this section, we provide an overview of the design of the questionnaire. Section 1.1 presents the research questions underlying the questionnaire. Section 1.2 outlines how the questionnaire was developed and distributed. Section 1.3 defines the different types of questions used in the questionnaire.

1.1 Research questions

We conducted the questionnaire to answer the following research questions:

- **RQ.1: What is the status quo of the tools currently used in industry?** From the tools currently used in industry, we want to draw conclusions about the methods used for the model-based development of embedded systems.
- **RQ.2: What are the requirements for a next generation of industrial tools?** We want to derive requirements for a next generation of industrial tools. We plan to take these requirements into account for the tool architecture which should be developed as part of work package 1.4 of the central project.

1.2 Execution

To answer these research questions, we developed a questionnaire which consists of 25 questions (see Section A). The first version of the questionnaire was presented at the kickoff for work package 1 on March 19th, 2009. We got some feedback to improve the questionnaire which we integrated into the second version. The questionnaire was distributed via mailing list to all industry partners of the SPES project on April 23rd, 2009. Until the deadline on May 8th, 2009, we unfortunately did not receive many filled out questionnaires. That's why we directly contacted all industry partners on May 27th, 2009, and asked them again to fill out the questionnaire. Until July 31th, 2009, we altogether received 24 completely filled out questionnaires.

1.3 Types of Questions

In the questionnaire, we asked three different types of questions. We present each type of questions together with the diagram presentation of its results in Section 2:

- **One possible answer.** In these questions, we provided a number of answers from which the respondents should choose one. The results to these questions are represented as a pie chart showing the distribution over the possible answers. An example for such a diagram is depicted in Figure 1. The diagram always shows all possible answers to the question, even if one answer was not chosen at all. However, some respondents did not provide an answer to some questions. For missing answers, we added the additional category *no answer*.

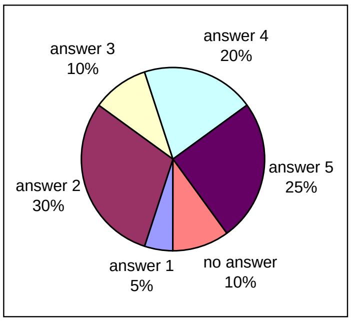


Figure 1: One possible answer

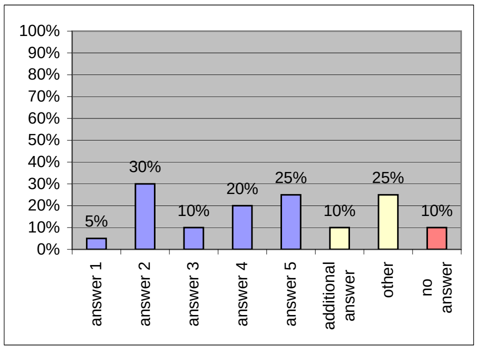


Figure 2: Multiple possible answers

- **Multiple possible answers.** In these questions, we provided a number of answers from which the respondents could choose multiple ones. Furthermore, we included a category *other* through which the respondents could name additional answers. The results to these questions are represented as a column chart showing a column for each answer. The column quantifies the fraction of the respondents which have chosen the answer. An example for such a diagram is depicted in Figure 2. The diagram always shows all the predefined answers to the question in *blue*. Additional answers are depicted in *yellow*. In case at least two respondents named the same additional answer, we created a separate column. The other additional answers which were named only once are summarized through the *other* column, and mentioned in the text. Respondents which neither choose a predefined answer nor named an additional answer are subsumed under the column *no answer*. This column is always colored in *red*.
- **One possible answer for different aspects.** In these questions, the respondents should choose one answer for different predefined aspects. The answers are the same for each aspect. The results to these questions are represented as a stacked column chart showing a column for each aspect. The column depicts the distribution of the results over all answers. An example for such a diagram is shown in Figure 3. Respondents which have not given an answer to a certain aspect are summarized through the category *no answer*. In some questions, the respondents could name additional aspects. However, we only received very few additional aspects, and thus mentioned them only in the text.

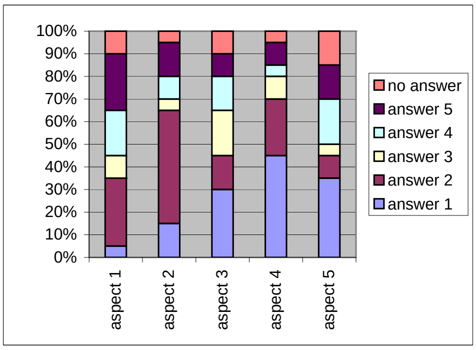


Figure 3: One possible answer for different aspects

2 Results

In this section, we present the results of each question together with a short interpretation. The questions are divided into the following categories:

- **Context.** This category asks questions about the project in whose context the questionnaire was filled out (see Section 2.1).
- **Target Platform.** This category concentrates on questions about the target platform onto which the developed embedded system is deployed (see Section 2.2).
- **Overview of Today's Tools.** This category asks questions about the status quo of tools currently used in industry (see Section 2.3).
- **Deficits of Today's Tools.** This category concentrates on questions about problems with the current tools and about requirements for future tools (see Section 2.4).

Remark: Questions 1 and 2 asked for the name and the affiliation of the respondent, respectively. Since we want to keep the result of the questionnaire fully anonymous, their answers are left out here.

2.1 Context (Questions 3-6)

The questionnaire should be filled out from the perspective of a certain project in which the respondent works. The following questions ask for some information about the project and the respondent's role in the project.

- **Q.3: How many persons comprises the project you are working in?** (One possible answer)

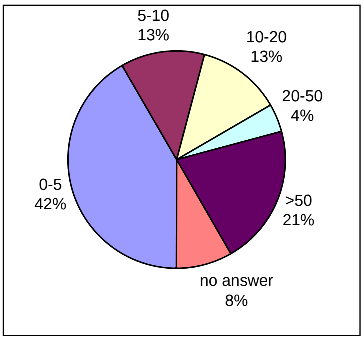
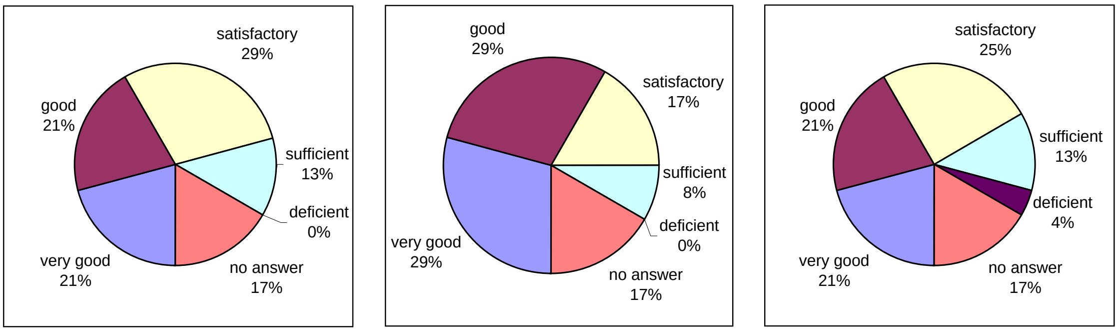


Figure 4: Project size (Q.3)

Figure 4 shows the distribution of the project size over the predefined categories. The filled out questionnaires cover the whole range of these categories. Small projects dominate the results of the questionnaire, as more than 50% of the answers are from projects with a maximum of 10 people. Unfortunately, 8% of the respondents have provided *no answer* to this question.



(a) Quality (b) Time (c) Cost

Figure 5: Project success (Q.4)

- **Q.4: How do you assess the success of the project?** (One possible answer)

  Figure 5 shows the respondent's assessment of the project success with respect to three criteria: achievement of planed quality (see Figure 5(a)), compliance with planed time (see Figure 5(b)), and compliance with planed cost (see Figure 5(c)). Even though the results of the three criteria are similar to each other, we believe that the separation lead to more honest answers. 66% to 71% of the respondents assess the project success from very good to satisfactory in terms of quality, time, and cost. However, 13% to 17% think that the project success is only sufficient or even deficient. Again, the remaining 17% of the respondents have provided *an answer* to this question.

- **Q.5: In what domain are you active?** (Multiple possible answers)

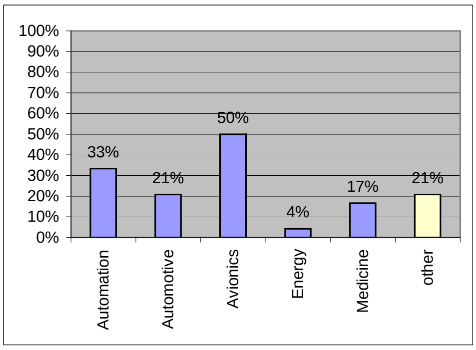


Figure 6: Domain (Q.5)

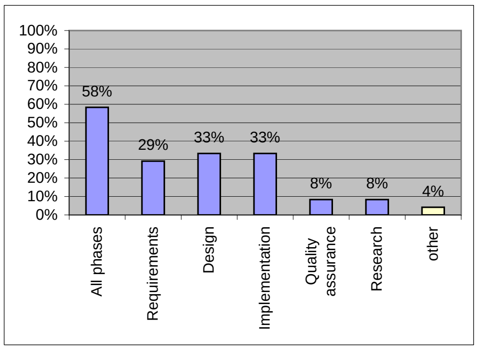


Figure 7: Process phase (Q.6)

Figure 6 shows the distribution of the respondents over the application domains of the SPES project. As we got answers from all the predefined domains, the result represents all their requirements. However, the energy domain is slightly underrepresented with only 4% of the respondents. 21% of the respondents also provided *other* domains which include customer-specific embedded systems, development tools for embedded systems, manufacturing systems and plant engineering, process engineering, and transportation.

- **Q.6: In which phase of the development process are you active?** (Multiple possible answers)

Figure 7 shows the distribution of the respondents over typical phases of a development process. More than half of the respondents are involved in all phases. Furthermore, quality assurance and research are underrepresented compared to the phases requirements elicitation, design and implementation. The *other* answer is DO178B Planing and Certification Process.

2.2 Target Platform (Questions 7-10)

The following questions focus on the target platform on which the developed embedded system is deployed. We have identified the four aspects programming language, operating system, bus system, and middleware which characterize the target platform. In the following, we present the accumulated answers to questions about each of these aspects.

- **Q.7: In what programming language is the developed system written (possibly generated from models)?** (Multiple possible answers)

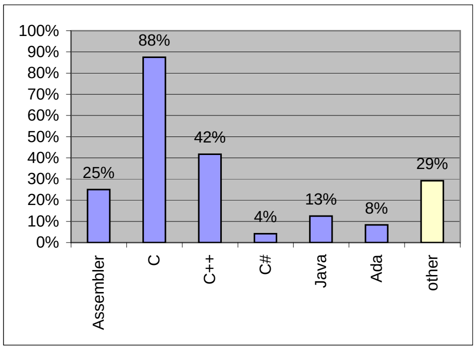


Figure 8: Programming language (Q.7)

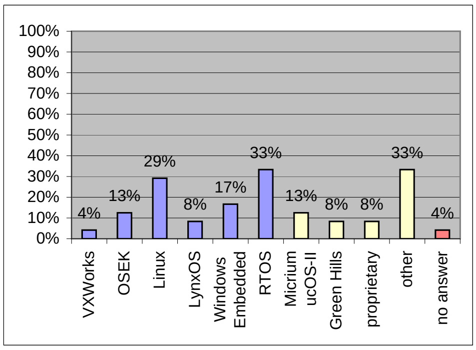


Figure 9: Operating system (Q.8)

Figure 8 shows the distribution of the answers over the typical programming languages used for the implementation of embedded systems. Low-level languages are still in wide use for the implementation of embedded systems. With 88%, C is the most widely used programming language. Its object-oriented variant C++ is used for the implementation of 42% of the systems. Moreover, 25% of the embedded systems are partly implemented in Assembler. High-level languages like Java, C# and Ada are not yet widely used for the implementation of embedded systems. 29% of the respondents also use *other* programming languages like Delphi, Pascal, PCS7, Python, Step7, Structured Text, TPU3 Microcode, VHDL, and “not specified”. The high number of different programming languages shows how heterogeneous the development of embedded systems is.

- **Q.8: Which operating systems are running on the developed system?** (Multiple possible answers)

Figure 9 shows the distribution of the answers over the typical operating systems which are running on the embedded systems. The most widely used operating systems are real-time operating systems (RTOS) with 33%, Linux with 29%, and Windows Embedded with 17%. 33% of the respondents employ *other* operating systems like Embedded SW, MACOS (Thales), Microware OS-9 RTOS, Simatic, Simatic TDC, Sysgo (Pike OS), Timer

Interrupt Driven Proprietary Software Sequenzer, TinyOS, Windows XP, and a simple scheduler. Again, the extremely huge number of about 20 different operating systems shows how heterogeneous the development of embedded systems is.

- **Q.9: Which bus systems do you use on the developed system?** (Multiple possible answers)

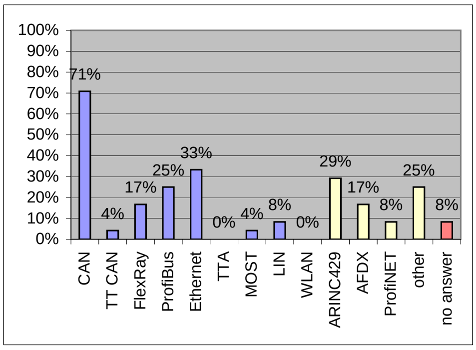


Figure 10: Bus system (Q.9)

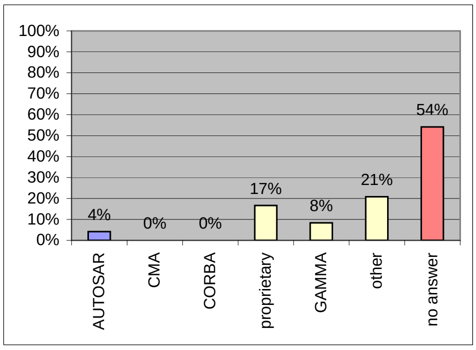


Figure 11: Middleware (Q.10)

Figure 10 shows the share of typical bus systems used for the development of embedded systems in the answers. With 71% of the answers, CAN is the most widespread bus system. Ethernet which is used for 33% of the embedded systems becomes more and more popular. Similar to the other aspects of the target platform, a very heterogeneous picture is presented. *Other* bus systems include EtherCAT, I2C, Interbus, MIL-STD-1553, proprietary, S5, S7, and "not specified".

- **Q.10: Which middleware do you use on the developed system?** (Multiple possible answers)

  Figure 11 shows the distribution of the answers over typical middlewares used for the development of embedded systems. Apparently, no middleware has been widely established in industry up to now. Over half of the respondents do not use a middleware at all, as they have provided *no answer* to that question. The respondents named a lot of *other* middlewares like ACE, ARINC653, COM, DCOM, DDS, TAO, and "not specified", which is again an indicator for the heterogeneity of embedded systems development.

2.3 Overview of Today's Tools (Questions 11-21)

This section provides an overview of the tool landscape currently used in industry. The questions are subdivided into the following three categories:

- **Nature of Tools.** This category concentrates on questions about the tools currently used in industry. It is subdivided into the following two categories:
  - **Vertical Tools.** Vertical tools focus on a special development phase, like e. g., requirements elicitation or design (see Section 2.3.1).
  - **Horizontal Tools.** Horizontal tools are necessary for all development phases, like version control systems or change management tools (see Section 2.3.2).

- **Availability of Tools.** This category concentrates on the questions whether the needed tools are available on the market and what the industry does in order to achieve the wanted tool functionality (see Section 2.3.3).

2.3.1 Vertical Tools (Questions 11-13, 17-18)

This section presents the results for questions about tools which are specific to a certain development phase. We were interested in tools used in the typical development phases requirements management, modeling, and quality assurance.

- **Q.11: What tools are used for requirements elicitation and management?** (Multiple possible answers)

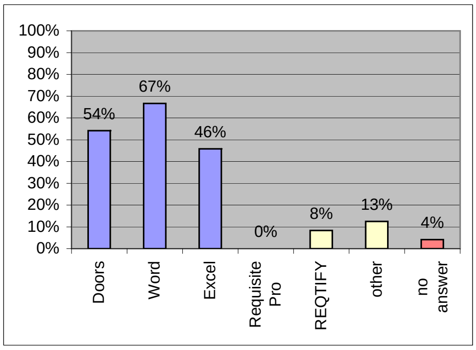


Figure 12: Requirements tool (Q.11)

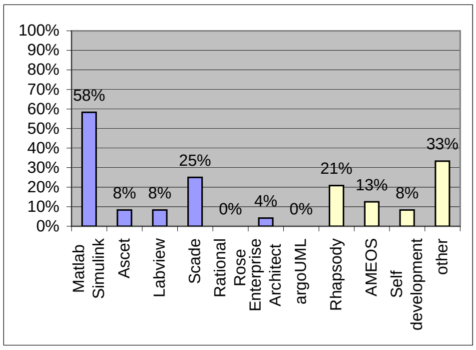


Figure 13: Modeling tool (Q.12)

Figure 12 shows the distribution of the answers over typical tools for requirements elicitation and management. Even though not a tool tailored for requirements management, Microsoft Word and Microsoft Excel are still the most widespread tools with a share of 67% and 46%, respectively. However, Doors is used as a specific requirements management tool in more than 50% of the answers. 13% of the respondents use *other* tools like STS, radCASE, and “not specified”.

- **Q.12: What modeling tools are used?** (Multiple possible answers)

Figure 13 shows the share of typical modeling tools in the answers. With a usage rate of 58%, Matlab Simulink is the most widespread modeling tool, followed by SCADE with 25%. The UML and SysML tool Rhapsody is only on the third position, with a share of 21%. Besides Rhapsody, there are other UML tools used in industry like AMEOS and Enterprise Architect. 33% of the respondents named *other* modeling tools like MID Innovator, NX, PCS7 Engineering, radCase, SiGraph, Solid Edge, TAU, Teamwork, UML, X32 BlueRiver, and “not specified”. The high number of employed modeling tools shows how heterogeneous the tool landscape for modeling is.

- **Q.13: How large is the fraction of code generated from models?** (One possible answer)

Figure 14 shows the distribution of the fraction over predefined categories. The ultimate goal of model-based systems engineering is to generate 100% of the target code. The

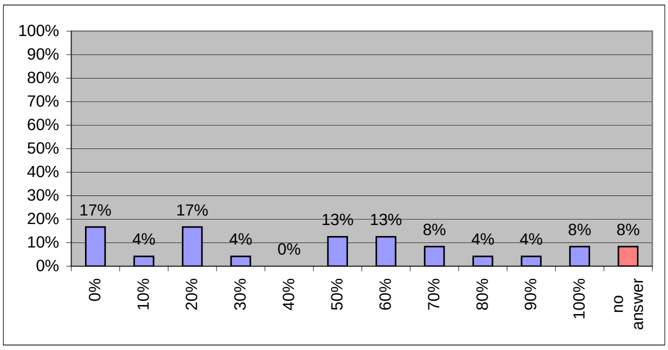


Figure 14: Generated code (Q.13)

result of this question shows us that the industry is currently still far away from that goal: Only 16% of the respondents generate from 80% to 100% of the code. 34% generate from 40% to 70% of the code, and 42% generate only 0% to 30% of the code. An interesting point is that there are already some companies that claim 100% code generation in certain projects.

- **Q.17: Which methods do you use for quality assurance?** (Multiple possible answers)

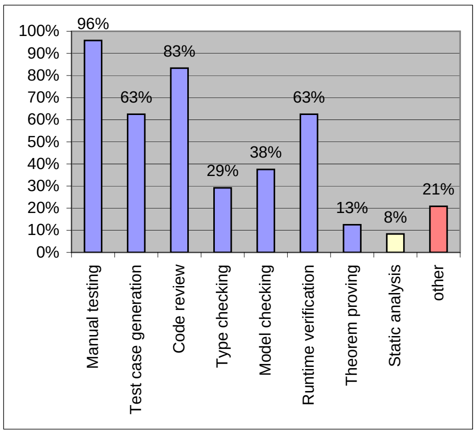


Figure 15: Quality assurance method (Q.17)

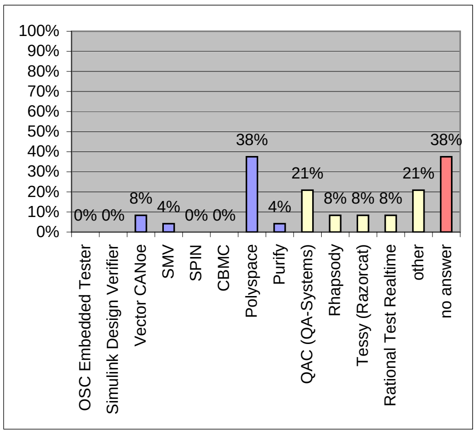


Figure 16: Quality assurance tool (Q.18)

Figure 15 shows the share of different methods for quality assurance in all the answers. Nearly every project performs manual tests and code reviews for quality assurance (96% and 83%, respectively). But also more formal methods like test case generation and runtime verification are adopted in 63% of the projects. The respondents also named *other* methods like Airplane testing, Automated testing, Design Review, Model Review, Requirements Review, Certification (especially RTCA/DO-178B), System simulation, and System testing.

- **Q.18: Which tools do you use for quality assurance?** (Multiple possible answers)

Figure 16 shows the distribution of the answers over typical tools for quality assurance. With a share of 38%, Polyspace is the most popular quality assurance tool. Although

Matlab Simulink itself is widespread, the Simulink Design Verifier is not used in a lot of cases. 38% of the respondents have provided *no answer*, and thus do not seem to use a tool for quality assurance at all. As a consequence, there seems to be either not enough or inappropriate tool support for quality assurance. 21% of the respondents named *other* tools like Automatic Test Sequenzer, FTI test environment, PC-lint, TAU, TechSat ADS-2 Test-Benches, and XLINT. Again, the high number of tools used for quality assurance provides evidence for the heterogeneity of the tool landscape in industry.

2.3.2 Horizontal Tools (Questions 14-16, 19)

This section presents the results for questions about tools which are independent of a certain development phase. We were interested in cross-cutting tools like tool platforms as well as tools for configuration management, change management, and process support.

- **Q.14: Which tool platforms are used?** (Multiple possible answers)

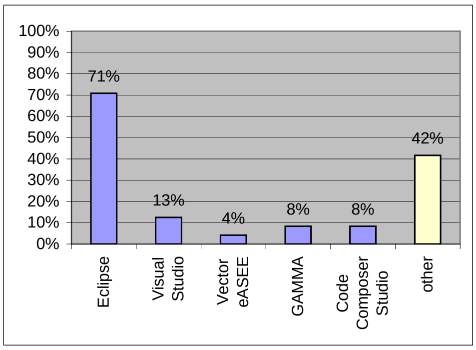


Figure 17: Tool platform (Q.14)

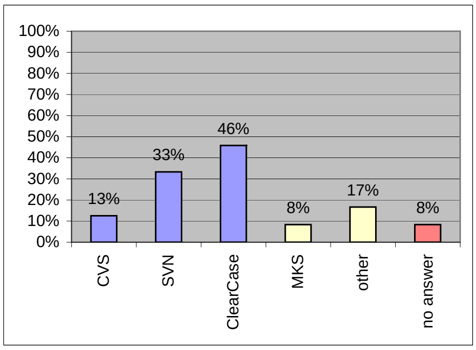


Figure 18: Config. management tool (Q.15)

Figure 17 shows the share of typical tool platforms in the answers to the questionnaire. With a share of 71%, the open source tool platform Eclipse is the clear winner of this question. The respondents also named *other* tool platforms like Borland C-Builder, Borland C++, comos, CrossCompilers DIE, gcc, Green Hills, HAWK, radCASE MULTI, Redcase, Simatic Manager, TeamCenter, X32 BlueRiver, XiBase9, and “not specified”. Consequently, the huge number of utilized tool platforms shows how heterogeneous the tooling landscape is today.

- **Q.15: What tools are used for configuration and version management?** (Multiple possible answers)

Figure 18 shows the distribution of the respondents over tools for configuration and version management. With a share of 46%, the commercial tool ClearCase is the most widespread configuration management tool. But also the non commercial and open source alternatives CVS and SVN are widely used (13% and 33%, respectively). The younger SVN is more widely used than the older CVS. 17% of the respondents also use *other* configuration management tools like Proflow, PVCS, Simatic-proprietary, SmarTeam, and SourceSafe.

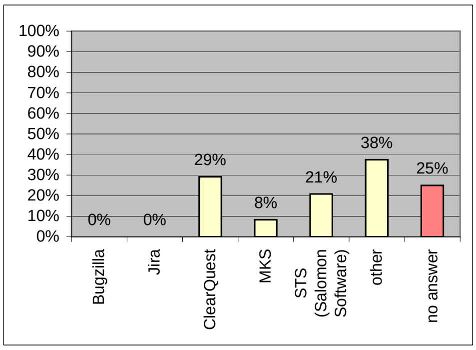


Figure 19: Change management tool (Q.16)

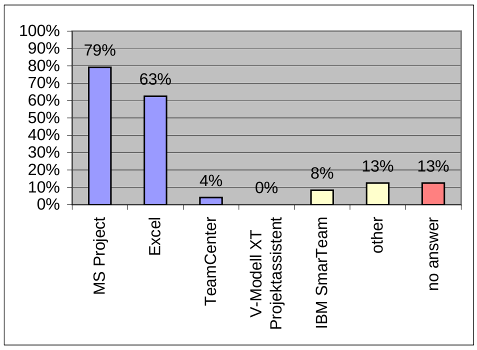


Figure 20: Process support tool (Q.19)

- **Q.16: What tools are used for change management?** (Multiple possible answers)

  Figure 19 shows the share of different tools for change management in the answers. ClearQuest (29%) and STS (21%) are the most prefered change management tools. Open-source tools like Bugzilla or Jira seem to be not used at all in industry. 38% of the respondents use *other* tools like Mantis, Microsoft Excel, proprietary, radCASE compare, SAP PLM, Serena TeamTrack, TESSY, and "not specified". Interestingly, Microsoft Excel is used as a tool for change management, even though not designed for change management. Again, the huge number of mentioned tools shows the heterogeneity of the tool landscape.

- **Q.19: What tools do you use for process support?** (Multiple possible answers)

  Figure 20 shows the distribution of the respondents over different tools for process support. Microsoft Project and Excel are mainly used for process management – with a share of 79% and 63%, respectively. However, both tools allow the developers only to define the process in an informal way, and thus do not allow to operationalize it. Besides, there is a set of other alternatives, but they are barely used. 13% of the respondents named *other* tools like ClearQuest, PVCS, SAP PLM, STS (Salomon Software), TeamTrack, and "not specified".

2.3.3 Availability of Tools (Questions 20-21)

In this section, we concentrate on the questions whether the needed tools are available on the market, and what the industry does in order to achieve the wanted tool functionality.

- **Q.20: Do you think that you use the best tools available on the market?** (One possible answer for different aspects)

  Figure 21 shows the distribution of the respondents' rating with respect to the different activities in which tool support is required. When focusing on vertical tools, the respondents are more convinced of the tools used for modeling and implementation than those for requirements management and verification. When focusing on horizontal tools, the respondents are more convinced of the tools used for configuration management than those for change management and process support. On average, only about 15% of the respondents think that they use the best tool available on the market. A reason for this

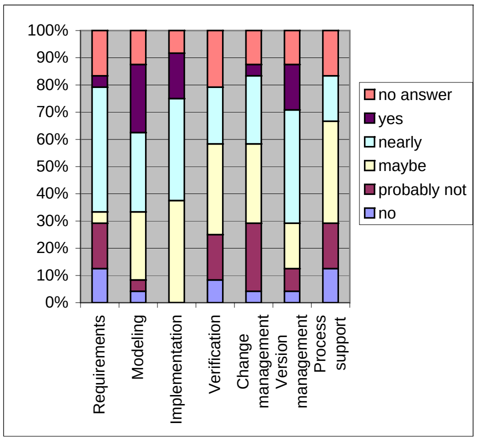


Figure 21: Best tools (Q.20)

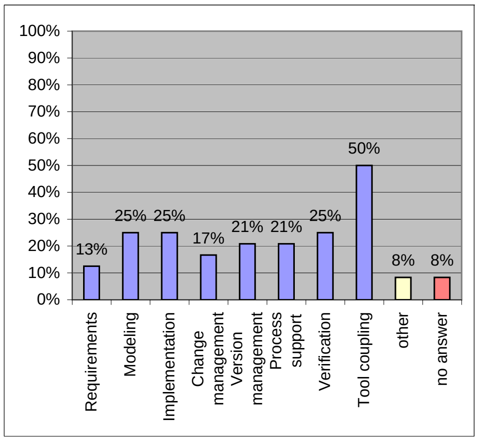


Figure 22: Own tools (Q.21)

may be the high barriers to switch to another tool because of the high migration costs. Sometimes, it may be even impossible to switch to another product, as the better tool may not integrate well with the rest of the used tools. Two respondents named *other* activities like code generation and simulation.

- **Q.21: For which tasks do you use tools especially developed for you?** (Multiple possible answers)

  Figure 22 shows the share of different activities for which tools were especially developed. These answers show two points: On the one hand, in each area there is a significant rate of proprietary tools. This may indicate that available tools do not fulfill the requirements of the industry. One the other hand, 50% of the respondents use handcrafted tool couplers to integrate different tools. This demonstrates the importance of a seamless integration of tools. 8% of the respondents named *other* activities like build management and data exchange, for which tools were especially developed.

2.4 Deficits of Today's Tools (Questions 22-25)

This section focuses on the identified deficits of today's tools. We concentrate on the missing features and their impact, as well as the current and desired integration of the used tools.

- **Q.22: What features do you miss in the tools you use today?** (Multiple possible answers)

  Figure 23 shows the distribution of the missing features over the respondents. With the share of 71%, tool integration is the major missing feature in daily work, followed by process support with 42%. Even process support is related to tool integration, since process support cannot be fully operationalized, unless all the used tools are integrated. 25% of the respondents named *other* missing features like Architecture modeling and assessment, Certification, Lifecycle Management, Modularization means, Parallel model-based development per feature, and Stability.

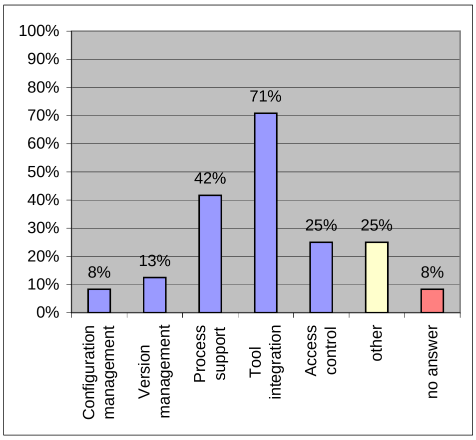


Figure 23: Missing features (Q.22)

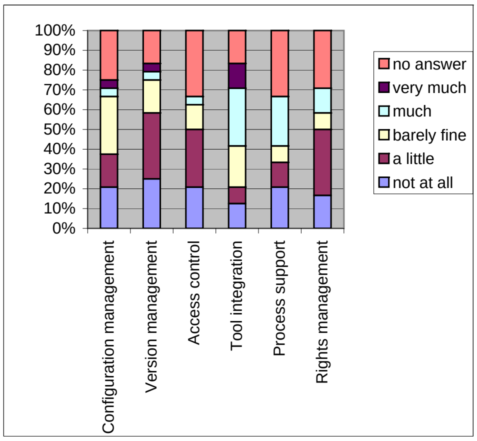


Figure 24: Impact of missing features (Q.23)

- **Q.23: How far is your work restricted by the missing features?** (One possible answer for different aspects)

  Figure 24 shows the distribution of the respondents' rating with respect to the different activities in which tool support is required. Again, the major restrictions are caused by the missing integration of tools: More than 40% of the respondents voted that the missing tool integration restricts their work much or even very much. Unfortunately, in average more than 20% of the respondents have provided *no answer* to this question. A respondent also named Lifecycle Management as another missing feature not mentioned by the questionnaire.

- **Q.24: How far is your tool chain integrated?** (One possible answer for different aspects)

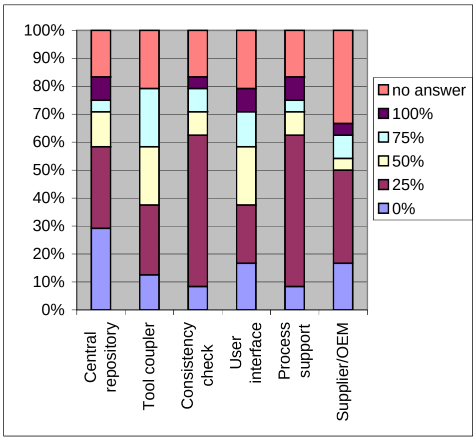


Figure 25: Current tool integration (Q.24)

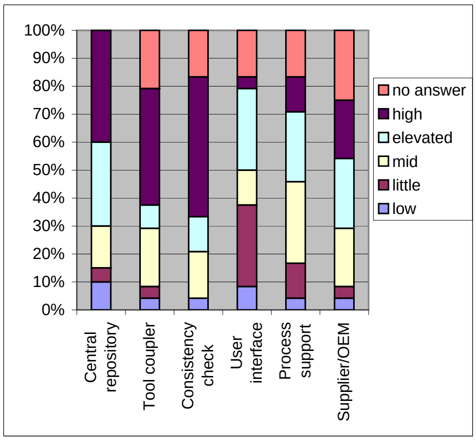


Figure 26: Desired tool integration (Q.25)

Figure 25 shows the distribution of the respondents over the degree of integration for different aspects of tool integration. Although there is a high demand for tool integration, tool chains currently used in industry are not deeply integrated. In average, more than

50% of the respondents are able to integrate less than 25% of the tools. Most tool chains are integrated by means of tool couplers, and even have an integrated user interface. However, a deeper integration of the tools through a central repository with integrated consistency management and process support is not very widespread.

- **Q.25: How would you prioritize the different kinds of integration?** (One possible answer for different aspects)

Figure 26 shows the distribution of the respondents over the importance of integration for different aspects of tool integration. A central repository, tool couplers and consistency checks are the most important aspects for tool integration. An integration of the user interface is not seen to be relevant. One respondent also named Offline-PC-Simulation as another important aspect for tool integration.

3 Discussion

In the previous section, we provided the detailed results of the questionnaire. Based on these numbers, we now discuss the results in terms of the research questions posed in Section 1. Research question 1 is discussed in Section 3.1, and research question 2 in Section 3.2.

3.1 What is the status quo of the tools currently used in industry?

When looking at the current status of development tools in industry, we can make the following observations:

- **Heterogeneity of target platform.** The target platform onto which the embedded system is deployed is affected by a considerable heterogeneity. As the results of questions Q.7 to Q.10 show, the target platform varies a lot from project to project in terms of both programming language, bus system, operating system, and middleware. It is important to note that this heterogeneity is not only between different companies, but also within a single company. As a consequence, the development tools are also affected by heterogeneity, as each of them is usually specialized for a certain target platform.
- **Heterogeneity of tools.** There is also an enormous heterogeneity in the tools that are used for the model-based development of embedded systems. This holds for both vertical and horizontal tools. The results show that vertical tools vary a lot for the activities of requirements management (Q.11), modeling (Q.12), and quality assurance (Q.18). Horizontal tools vary a lot in terms of the tool platform (Q.14) as well as for the activities of version (Q.15) and change management (Q.16). Again, this heterogeneity is not only between different companies, but also within a single company. The heterogeneity hampers the integration of the tools, as the different tools are usually based on different technologies.
- **Weakly defined models.** Tools which are based on models without a precise meaning are still widely used in industry. For requirements management, the industry still relies a lot on Microsoft Word and Excel which are not completely adequate for modeling requirements (Q.12). The situation is much better for system modeling with a high usage of modeling tools such as Matlab Simulink and Scade (Q.12). For process support, a lot of tool support is based on Microsoft Project and Excel which do not allow for the rigorous definition of a process (Q.19). Models with a precise meaning can be better analyzed as well as operationalized. Consequently, there is still much opportunity for improvement to benefit from the promises of model-based development.
- **Sporadic usage of model-based development.** The tool chains used in practice do not yet provide the full benefits of model-based development. Even though the industry uses a lot of advanced modeling tools like Matlab Simulink and Scade, a lot of code is still handwritten instead of generating it from the models (Q.13). For quality assurance, the industry still uses a lot of manual techniques like manual testing and code review (Q.17). Automatic techniques for model-based quality assurance like model checking and theorem proving are not yet widely adopted in industry. This may be due to the fact that

appropriate tools are not available on the market, or the respondents cannot use them due to other restrictions, like e. g., the missing interoperability with the current tool chain.

- **Inability to use best-in-class tools.** A lot of respondents to the questionnaire are not able to use the best tools available on the market. More specifically, the results show that only 15% of the respondents think that they use the best tool (Q.20, average over all process phases). A reason for this may be the high barriers to switch to another tool because of the high migration costs. For instance, migration effort is necessary for migrating existing data as well as for training developers to use the new tool. Sometimes, it may be even impossible to switch to another tool, as the better tool may not integrate well with the existing tool chain.
- **High rate of proprietary tools.** The industry makes use of a lot of proprietary tools which are especially developed for them. In particular, the results to question Q.21 show us that proprietary tools are developed for all process phases. Although a huge number of development tools are available, the market does not seem to provide solutions satisfying the requirements. This may be also due to the missing customizability of existing tools to the individual requirements. In particular, most effort is spent on the integration of existing tools which is very important for seamless model-based development.

3.2 What are the requirements for a next generation of industrial tools?

When discussing the major needs of future development tools, we can make the following observations:

- **Importance of tool integration.** A deep integration of the development tools is seen as the major issue. Most effort for proprietary tool development is spent on coupling of existing tools (Q.21). This is consistent with the result that tool integration is the most missing feature of today's tools (Q.22). Moreover, the missing integration of the tools heavily restricts the respondents in their daily work (Q.23). Currently, only very few tool chains are fully integrated on different levels (Q.24). However, tool integration is considered as important on nearly all levels of integration (Q.25).
- **Importance of process support.** Support of the development process is seen as an important feature which is not fully satisfied by current tools. Currently, the respondents mainly use Microsoft Process and Excel for process support (Q.19), but at the same time do not think that they use the best tools on the market (Q.20). However, they do not invest too much effort into the development of individual process support (Q.21). This is probably due to the fact that good process support also requires tool integration which is also missing today. After tool integration, the respondents see process support as the most important missing feature (Q.22). The missing process support restricts a lot of respondents in their daily work (Q.23). Today, only very few tool chains are fully integrated by means of process support (Q.24), but however more integration is desired (Q.25).
- **Back end integration more important than front end integration.** Integrating the back end is seen to be much more important than integrating the front end. In particular, back end integration techniques like a central repository, tool couplers and consistency

checks are considered as highly important, whereas integration of the user interface is not as important (Q.23). This seems to be the right prioritization, since without an integrated back end an integration of the front end cannot provide significant advantages.

4 Threats to Validity

We are aware that our results can be influenced by various threats to validity. They can be divided into threats to internal and external validity. In Section 4.1, we list threats to internal validity which deal with the impact of the questionnaire's design on the results. In Section 4.2, we mention threats to external validity which deal with the representativeness of results.

4.1 Internal Validity

The results might be influenced by the design we chose for the questionnaire. Thereby, the results might be affected by the way we posed the questions and arranged them.

- **Implicit assumptions in questions.** In the questions, we make implicit assumptions about what is the (optimal) scenario of model-based development. For example, we assume that model-based development should provide full code generation (Q.13), and that full tool integration is the optimal solution (questions Q.24 to Q.26). But there may be completely different solutions which we do not take into account, and thus they are not represented by the questionnaire. However, these assumptions are in line with the paradigm and the promises of model-based development.
- **Priming through predefined answers.** The selection of the possible answers may influence the result of the questionnaire. When filling out the questionnaire, an important answer may be simply forgotten, because it is not explicitly mentioned. To mitigate this threat, we provided a field to specify *other* answers. This field was explicitly highlighted by an underline which makes it difficult to miss this field.
- **Different understanding of terms.** The respondents may have a different understanding of the terms we used in the questions. Examples may be terms like "tool platform" (Q.14), or "tool integration" (questions Q.24 to Q.26). Consequently, we may misinterpret their answers, and thus come to a wrong conclusion. To mitigate this threat, we tried to find commonly understood terms as far as possible.

4.2 External Validity

The results might be influenced by the fact that we questioned only a subset of the possible developers of embedded systems. Thereby, the results might not be representative for model-based development of embedded systems in general.

- **Number of respondents.** We only received 24 filled out questionnaires which poses a great threat to the results' validity. However, these questionnaires were filled out in various different kinds of contexts: the results represent all project sizes (Q.3), different levels of project success (Q.4), all domains (Q.5), and all process phases (Q.6).
- **Distribution of respondents.** The respondents were not equally distributed over all SPES companies, taking their size into account. That means that we received more filled out questionnaires from smaller companies, but fewer from bigger others. So we cannot really assume that the distribution of different tools and technologies (questions Q.7 to

Q.19) represents the real situation in industry. However, we can still derive that the tools and technologies used for the development of embedded systems are quite heterogeneous today.

- **Locality of respondents.** All the questionnaires were filled out by respondents located in Germany. Consequently, this threatens the transferability of our results to model-based development in other countries. However, we also had respondents working in companies that act globally, as well as in globally distributed projects.

5 Conclusion

We have developed a questionnaire to better assess the current situation of model-based development of embedded systems in industry. We have distributed this questionnaire to all industry partners of the whole SPES project, and have received a reasonable number of answers. In this paper, we have presented as well as discussed the results of the answers in terms of two research questions.

First, we were interested in the most important issues of model-based development in industry today. We found a considerable heterogeneity in both the development tools as well as the target platform onto which the embedded system is deployed. This heterogeneity makes it difficult to seamlessly integrate both the tools and methods for model-based development. The results also show that model-based development is not yet used to its full extent in industry. In some development phases, only weakly defined models are used which makes it impossible to employ advanced techniques for model-based quality assurance. Furthermore, the existing tools and methods are still far away from being appropriate for efficient model-based development. That's why the industry has to develop more appropriate tools themselves which results in significant efforts. Additionally, high migration costs from one tool to another hamper the usage of the best tools on the market.

Second, we wanted to identify the most important needs for model-based development in the future. According to the results, the integration of tools and methods is considered as the most promising way to more efficient model-based development. The integration of the tools is not only important on the level of the models, but also on the level of the process. However, seamless integration on the higher levels (e. g., the process) requires integration on the lower levels (e. g., the models).

In work package 1 of the central project, we try to address all the issues and needs which we have identified through the questionnaire. Work package 1.1 develops a modeling theory which allows to rigorously define all models created during a development process as well as their dependencies. Based on the modeling theory, work package 1.2 defines a modeling language which allows to model an embedded system on different abstraction layers. This modeling language integrates different views onto an embedded system, and can be tailored to a certain domain. Work package 1.3 develops methods to assure the quality of the models created with this modeling language. Finally, work package 1.4 provides concepts for a tool architecture for integrated model-based development. This tool architecture provides horizontal tools as an infrastructure, and can be easily extended by vertical tools.

A Questionnaire (German)

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

Lehrstuhl IV: Software & Systems Engineering
Prof. Dr. Dr. h.c. Manfred Broy

SPES 2020

Fragebogen Werkzeuge & Technologien

Ziel: Dieser Fragebogen soll einen Überblick über die aktuell im industriellen Umfeld eingesetzten Programmiersprachen, Werkzeuge, Betriebssysteme und Technologien zur Entwicklung von eingebetteten Systemen schaffen. Dies soll unter anderem dabei helfen, Anforderungen für eine erste Version der **SPES 2020**-Werkzeugarchitektur abzuleiten.

Vorgehen: Der Fragebogen soll bei allen Projektpartnern verteilt werden. Insbesondere von Interesse sind natürlich die Abteilungen, die Serienentwicklung betreiben. Basierend auf den Ergebnissen aus dem Fragebogen sollen kurze Interviews geführt werden, um die Anforderungen an die Werkzeugarchitektur weiter zu detaillieren. Abschließend werden die Ergebnisse des Fragebogens und der Interviews in anonymisierter Form allen Teilnehmern zur Verfügung gestellt.

1. **In welchem Unternehmen sind Sie tätig?** (Diese Angabe ist freiwillig)

   ____

2. **Wie sind Ihre Kontaktdaten?** (Diese Angabe ist freiwillig, wird aber für die Interviews benötigt)

   ____

Füllen Sie die Fragen bitte immer im Kontext **eines** Projektes aus, in dem Sie arbeiten. Falls Sie in mehreren Projekten arbeiten, können Sie gerne mehrere Fragebögen ausfüllen.

3. **Wie viele Personen umfasst das Projekt, in dem Sie arbeiten?**

0-5	5-10	10-20	20-50	> 50
☐	☐	☐	☐	☐

4. **Wie schätzen Sie den Erfolg des Projektes ein?**

	schlecht		mittel		sehr gut
Erreichen der geplanten Qualität	☐	☐	☐	☐	☐
Einhaltung des geplanten Termins	☐	☐	☐	☐	☐
Einhaltung der geplanten Kosten	☐	☐	☐	☐	☐

5. **In welcher Domäne sind Sie tätig?** (Mehrfachnennungen möglich)

   ☐ Automatisierungstechnik ☐ Automotive ☐ Avionik
   ☐ Energie ☐ Medizin
   Sonstige: ______________________________

6. **In welcher Phase des Entwicklungsprozesses sind Sie tätig?** (Mehrfachnennungen möglich)
   - ☐ Alle Phasen
   - ☐ Anforderungsanalyse
   - ☐ Entwurf
   - ☐ Implementierung
   - ☐ Qualitätssicherung
   - ☐ Forschung

   Sonstige: ____________________

7. **In welcher Programmiersprache wird das entwickelte System programmiert (ggf. auch aus Modellen generiert)?** (Mehrfachnennungen möglich)
   - ☐ C
   - ☐ C++
   - ☐ Java
   - ☐ C#
   - ☐ Ada
   - ☐ Assembler

   Sonstige: ____________________

8. **Welche Betriebssysteme kommen bei dem entwickelten System zum Einsatz?** (Mehrfachnennungen möglich)
   - ☐ VXWorks
   - ☐ OSEK
   - ☐ Linux
   - ☐ LynxOS
   - ☐ Windows Embedded
   - ☐ RTOS

   Sonstige: ____________________

9. **Welche Bussysteme verwenden Sie auf dem entwickelten System?** (Mehrfachnennungen möglich)
   - ☐ CAN
   - ☐ TT CAN
   - ☐ FlexRay
   - ☐ ProfiBus
   - ☐ Ethernet
   - ☐ TTA
   - ☐ Most
   - ☐ LIN
   - ☐ WLAN

   Sonstige: ____________________

10. **Welche Middleware setzen Sie auf dem entwickelten System ein?** (Mehrfachnennungen möglich)
    - ☐ AUTOSAR
    - ☐ CMA
    - ☐ CORBA
    - ☐ Sonstige: ____________________

11. **Welche Werkzeuge werden zur Anforderungserhebung und zum Anforderungsmanagement eingesetzt?** (Mehrfachnennungen möglich)
    - ☐ Doors
    - ☐ Word
    - ☐ Excel
    - ☐ Requisite Pro

    Sonstige: ____________________

12. **Welche Modellierungswerkzeuge werden eingesetzt?** (Mehrfachnennungen möglich)
    - ☐ Matlab Simulink
    - ☐ Ascet
    - ☐ Labview
    - ☐ Scade
    - ☐ Rational Rose Realtime
    - ☐ Enterprise Architect
    - ☐ argoUML

    Sonstige: ____________________

13. **Wie groß ist bei Ihnen der Anteil der aus Modellen generierten Software?**

0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

14. **Welche Werkzeugplattformen werden eingesetzt?** (Mehrfachnennungen möglich)
☐ Eclipse ☐ Visual Studio ☐ Vector eASEE
Sonstige: ____________________

15. **Welche Werkzeuge werden zum Konfigurations- und Versionsmanagement eingesetzt?** (Mehrfachnennungen möglich)
☐ CVS ☐ SVN/Subversion ☐ ClearCase
Sonstige: ____________________

16. **Welche Werkzeuge werden zum Change Management eingesetzt?** (Mehrfachnennungen möglich)
☐ Bugzilla ☐ Jira
Sonstige: ____________________

17. **Welche Methoden zur Qualitätssicherung nutzen Sie?** (Mehrfachnennungen möglich)
☐ Manuelles Testen ☐ Testfallgenerierung ☐ Code-Review
☐ Typüberprüfung ☐ Model Checking ☐ Runtime Verification
☐ Theorem-Beweisen
Sonstige: ____________________

18. **Welche Werkzeuge zur Qualitätssicherung nutzen Sie?** (Mehrfachnennungen möglich)
☐ OSC Embedded Tester ☐ Simulink Design Verifier ☐ Vector CANoe
☐ SMV ☐ SPIN ☐ CBMC
☐ Polyspace ☐ Purify
Sonstige: ____________________

19. **Welche Werkzeuge zur Prozessunterstützung verwenden Sie?** (Mehrfachnennungen möglich)
☐ MS Project ☐ Excel ☐ TeamCenter
☐ V-Modell XT Projektassistent
Sonstige: ____________________

20. **Glauben Sie, dass Sie die besten Werkzeuge auf dem Markt benutzen?**

	nein				ja
Anforderungen	☐	☐	☐	☐	☐
Modellierung	☐	☐	☐	☐	☐
Implementierung	☐	☐	☐	☐	☐
Change Management	☐	☐	☐	☐	☐
Versionsmanagement	☐	☐	☐	☐	☐
Prozessunterstützung	☐	☐	☐	☐	☐
Verifikation	☐	☐	☐	☐	☐
Sonstiges: ____________________	☐	☐	☐	☐	☐

21. Für welche Aufgaben verwenden Sie eigens für Sie entwickelte Werkzeuge?
(Mehrfachnennungen möglich)

☐ Anforderungen ☐ Modellierung ☐ Implementierung
☐ Change Management ☐ Versionsmanagement ☐ Prozessunterstützung
☐ Verifikation ☐ Kopplung existierender Werkzeuge

Sonstige: ____

22. Welche Funktionen vermissen Sie bei Werkzeugen, die sie heute einsetzen?
(Mehrfachnennungen möglich)

☐ Konfigurationsmanagement ☐ Versionsmanagement
☐ Prozessunterstützung ☐ Integration mit anderen Werkzeugen
☐ Zugriffskontrolle und Rechteverwaltung

Sonstige: ____

23. Wie stark schränken Sie die fehlenden Funktionen in Ihrer Arbeit ein?

	gar nicht				sehr stark
Konfigurationsmanagement	☐	☐	☐	☐	☐
Versionsmanagement	☐	☐	☐	☐	☐
Zugriffskontrolle	☐	☐	☐	☐	☐
Integration mit anderen Werkzeugen	☐	☐	☐	☐	☐
Prozessunterstützung	☐	☐	☐	☐	☐
Zugriffskontrolle und Rechteverwaltung	☐	☐	☐	☐	☐
Sonstiges: ______________________	☐	☐	☐	☐	☐

24. Wie stark ist die von Ihnen verwendete Werkzeugkette integriert?

	Anteil der integrierten Werkzeuge				
	0%	25%	50%	75%	100%
Zentrale Modelldatenbank	☐	☐	☐	☐	☐
Werkzeugkoppler incl. Tracing	☐	☐	☐	☐	☐
Sicherstellung der Konsistenz zwischen Modellen	☐	☐	☐	☐	☐
Integrierte Benutzeroberfläche	☐	☐	☐	☐	☐
Integrierte Prozessunterstützung	☐	☐	☐	☐	☐
Integration der Werkzeugketten Zulieferer/OEM	☐	☐	☐	☐	☐
Sonstiges: ______________________	☐	☐	☐	☐	☐

25. Wie stark würden Sie die verschiedenen Arten der Integration priorisieren?

	Priorisierung				
	niedrig		mittel		hoch
Zentrale Modelldatenbank	☐	☐	☐	☐	☐
Werkzeugkoppler incl. Tracing	☐	☐	☐	☐	☐
Sicherstellung der Konsistenz zwischen Modellen	☐	☐	☐	☐	☐
Integrierte Benutzeroberfläche	☐	☐	☐	☐	☐
Integrierte Prozessunterstützung	☐	☐	☐	☐	☐
Integration der Toolketten von Zulieferer und OEM	☐	☐	☐	☐	☐
Sonstiges: ______________________	☐	☐	☐	☐	☐