



Information Technology for European Advancement

Task 1.2 - Inventory of Methods and Tools(D.1.2.1)

Version 01 - Public

Edited by Erik Luit

Software Development Process for Real-Time Embedded Software Systems (DESS)

ITEA COMPETENCES involved:

- 1) Complex Systems Engineering**
- 2) Communications**
- 3) Distributed Information and services**

November 2000

Table of Contents

Table of Contents	1
Purpose of this document	2
WP1.2 partners	2
1. Introduction	3
2. Conclusions	3
Appendix A. Questionnaire	5
Appendix A. Questionnaire	5
Appendix B. Manual of the questions database	8
Appendix C. Description of the Tools and Methods Database	14
Appendix D. Categorized Tool List	17
Appendix E. Reported Tool Shortcomings	22
Appendix F. Combined Methodology Questionnaire	25

Purpose of this document

The purpose of Task 1.2 is to make an inventory of existing methodologies and tools. The purpose of this document is to describe how this is achieved: the actual information is represented in database form, available on the DESS website.

The rationale for this approach is given in the Introduction section of this document. The rest of the document consists of Appendices providing insights about this database from various angles. For example, the Questionnaire section shows how information from the partners was acquired, an Appendix like the Description of the Tools and Methods Database explains the structure of the database, the categorized Tool List contains the tools that are assessed.

The Categorized Tool List Appendix is an example of the amount of information gathered in this Task: around 80 tools for which the partners have given the responses to the lists of questions (these list differing per tool type).

The actual information obtained in this Task is therefore in the database, where it is much more accessible than on paper.

Separate attention is given to the information about methodology, which proved, so far, less amenable to the database approach.

WP1.2 partners

All ITEA-DESS partners participate in WP1.2.

1. Introduction

The goal of WP1.2 is to obtain an inventory of existing methodologies and tools. This inventory is necessary to assess the position and the shortcomings of the consortium with respect to the state of the art. The work-package provides a starting and evaluation point for defining the DESS methodology and tooling. As input, the domain analysis of WP1.1 is required. WP1.2 is needed as input for all other WPs at a general level.

An inventory of tools was obtained via the construction of a questionnaire in the form of a database, which was distributed to the DESS partners. The questionnaire is included as Appendix A. The database was distributed together with a manual, included as Appendix B. The returned filled-in questionnaire databases were integrated to allow access to all data with database queries. This integrated database is part of this deliverable and is available on the DESS web site. A short description of the relevant database objects and their function is given in Appendix C. A tool inventory (presented in Appendix D) and a list of shortcomings (see Appendix E) were obtained from the integrated database. However, because the questions were open-ended, the usefulness of this “qualitative” database is somewhat limited. Also, the questions were not very detailed. Therefore, a new “quantitative” database is constructed in which much more detailed questions are asked. The answers in this database are recorded such that it is simple to analyse the data in an automated way.

The database was also meant to obtain an overview of methodologies as practised by the partners. However, because the methodology questions were based on the V-model, many partners could not fit their answers into this framework. Therefore, a new methodology questionnaire was made. Because the questions overlapped strongly with a questionnaire produced by WP5 (Software Development Process), it was decided to combine the two. The questionnaire is included in Appendix F and the first responses obtained are included in Appendix G.

2. Conclusions

At a meta-level, the database approach proved practicable as far as tools were concerned, as shown by, for example, the return on a question like presented in the Appendix Reported Tool Shortcomings. As reported already in the Introduction, the methodology issue is more elusive, but some results have been obtained (Appendix G). Also, in the process of carrying out this Task a need for more quantitative information became apparent and ways have been developed to deal with this - development of the corresponding database is work in progress, to be complete in the next phase.

Interaction with other DESS partners has been very useful, both in the sense that a quite comprehensive set of tools and tool requirements has been assessed and in the sense that Task 1.1 provided precise information to enable the incorporation of DESS specific requirements in the database to be provided by the next phase of Task 1.2. The Tasks and work packages addressing specific DESS capabilities are receiving input from Task 1.2 as well as providing feedback, e.g., as in the case of WP5.

At the level of the direct question this task addresses, the inventory of existing tools and methods, the information is in the database, questions can be formulated as queries to this database. General conclusions are the following

ITEA

- General-purpose tooling is available on the market (e.g., Rational Rose for modeling tools).
- Tooling for DESS-specific constraints as described in D1.1.2 for Task 1.2 is of much more specialist nature, often not quite as desired, sometimes specifically developed by or for a company that needs it, often non-existent.

Issues here are

- . adaptation/extension of existing tools;
- . specification or development of missing tools;
- . integration of specialist tools with existing general purpose tooling to achieve complete support for a methodology;

Especially this last issue requires detailed, quantitative information about tools.

- Process Management is, in the light of the large tool sets used, a very important but alas still quite underdeveloped competence.
- Methodology is often implicit, often tool-driven.

The last three points listed as needing attention are fortunately represented as aims in this project.

Appendix A. Questionnaire

Question	Comments
General Questionnaire	
1 Information about the company	
1 Who is the contact person at your company for general questions about tools and methods?	Please enter address, telephone number and email address
2 What is your application domain?	Please give a short description of your application domain (products and their characteristics typical for your company)
Tool Questionnaire (questions asked for each tool entered in the database by partner)	
2 General questions about the tool	
1 Who is the producer of the tool?	Please also supply web addresses of the producer.
2 Is the tool part of a suite?	If so, please name the other components.
3 On which methodology is the tool based?	E.g. UML, OMT or none
4 What is the functionality of the tool?	Please enter a short description of the tool.
3 Questions about the tool specific for your company	
1 Which version of the tool are you using?	
2 Is this the latest version of the tool available?	If not, what is the latest version?
3 Are you considering installing a new version shortly?	If so, which version?
4 What is the application area for which you use the tool?	Is there a particular type of product for which you use the tool?
5 Does the tool have a wider area of applicability?	If so: please describe the area and aspects of the tool you do not use.
6 Are there aspects of your products that are not well supported by the tool?	Which? E.g., timing constraints
7 Are there complementary or other tools that do support these aspects?	For the example of timing constraints, this could be e.g. Objectime

ITEA

Question	Comments
8 Have you previously considered alternatives for this tool or are you currently considering alternative tools?	What were the reasons (not) to choose the alternatives? If there are tools you consider particularly interesting, mention these here.
9 Who is the contact person for this tool within your company?	Please enter address, telephone number and particular Email address

Method Questionnaire (questions asked for each method entered by partner)

4 General questions about the method

- | | |
|--|------------------------------------|
| 1 Do you use a partyicular version (flavor) of the method? | |
| 2 Please characterize the method. | A short description of the method. |

5 Questions about the method specific for your company

- | | |
|---|--|
| 1 What is the application area for which you use the method? | Is there a particular type of product for which you use the method? |
| 2 Does the method have a wider area of applicability? | If so: please describe the area and aspects of the method you do not use. |
| 3 Are there aspects of your products that are not well supported by the method? | Which? E.g., timing constraints |
| 4 Are there complementary or other methods that do support these aspects? | For the example of timing constraints, this could be e.g. ROOM |
| 5 Have you previously considered alternatives for this method or are you currently considering alternative methods? | What were the reasons (not) to choose the alternatives?
If there are methods you consider particularly interesting, mention these here. |
| 6 Who is the contact person for this particular method within your company? | Please enter address, telephone number and email address |

ITEA

Question

Comments

V-Model

6 tools, functionality and methods used in V model phases

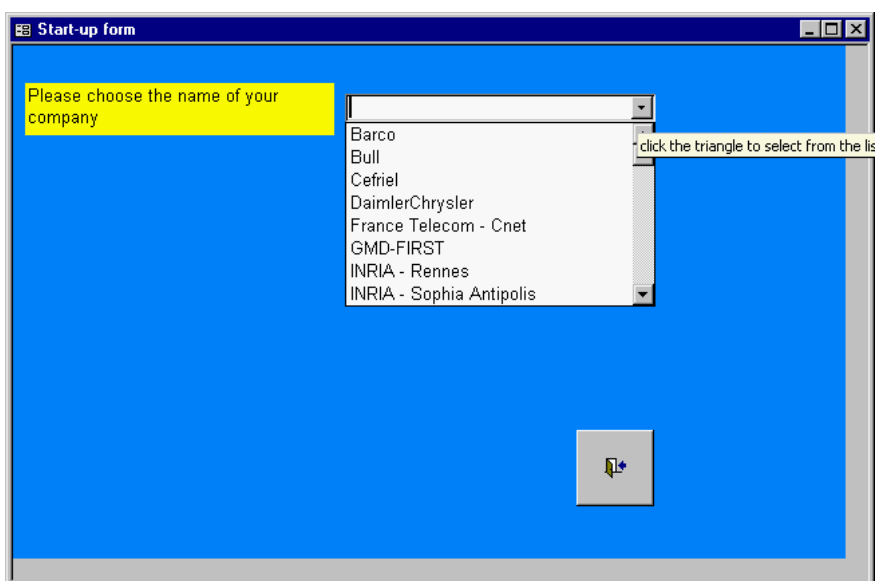
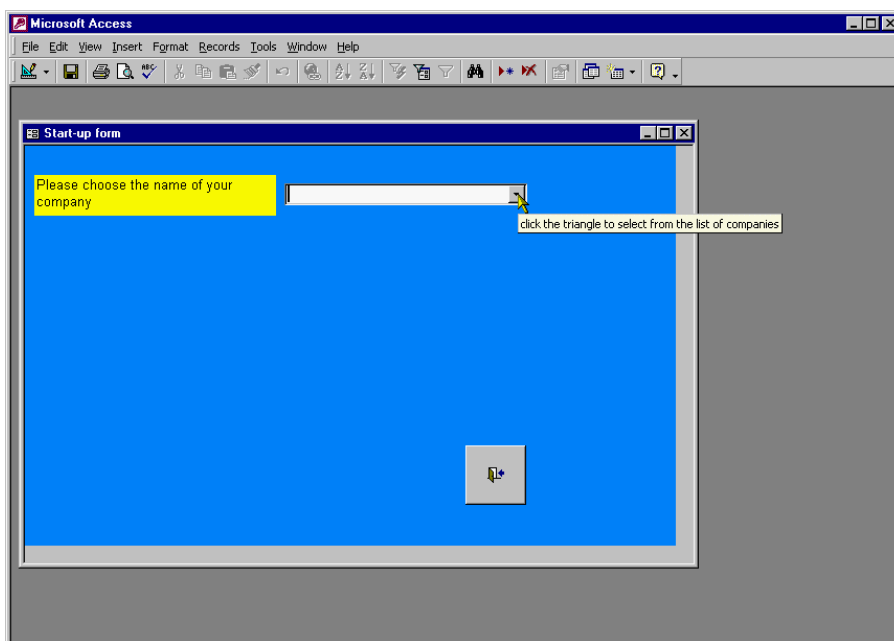
1	Preparation phase	E.g., OMT - class diagrams, event trace diagrams, state transition diagrams Cool:Jex - local model check, global model check UML - class diagrams, collaboration diagrams Rational Rose - standard only
2	User Requirements phase	Same as for Preparation phase
3	Software Requirements phase	Same as for Preparation phase
4	Architectural Design phase	Same as for Preparation phase
5	Detailed Design phase	Same as for Preparation phase
6	Coding phase	Same as for Preparation phase
7	Unit tests phase	Same as for Preparation phase
8	Integration Tests phase	Same as for Preparation phase
9	System Tests phase	Same as for Preparation phase
10	Acceptance test phase	Same as for Preparation phase
11	Transfer phase	Same as for Preparation phase
12	Operations and Maintenance phase	Same as for Preparation phase

Appendix B. Manual of the questions database

ITEA DESS initial questionnaire (Easiest is to make a printout of this)

The questionnaire is implemented in MS-Access 97. If you use Access 2000, you can convert the database; this presents no problem for us. The use of the database should be straightforward: buttons are clearly labeled and tool tips are displayed where it seemed useful.

To start, simply double-click (open) the initialquestions.mdb file. The start-up screen will automatically be displayed.



First select your company name: click on the triangle as indicated in the pictures above and scroll to your company.

ITEA

Once you have done that, four buttons appear:

- Tools in use at <your company>
- Methods in use at <your company>
- General questions
- V Model questions

The screenshot shows a window titled "Start-up form" with a blue background. At the top left, a yellow box contains the text "Please choose the name of your company". To its right is a dropdown menu showing "T.U. Eindhoven". Below this are four buttons: "Enter Tools in use at T.U. Eindhoven", "Enter Methods in use at T.U. Eindhoven", "Fill in General Questionnaire", and "Fill in V Model Questionnaire". A small "Add Tool" button with a plus sign is located at the bottom right.

The last two buttons will start questionnaires, whereas the first two result in the appearance of a form on which you can fill in tools respectively methods in use at your company. These latter forms do not start a questionnaire; these forms are only used to collect the names of the tools (methods) used at your company. You can enter the names of several tools in sequence using the "Add Tool" button. Please use the "Quit" button to close the form when you are finished! It is always possible to add another tool or method later on via the "Tools in use at" button on the startup screen.

The screenshot shows a window titled "Tools in Use" with a blue background. At the top left, a yellow box contains the text "Please enter the name of a tool in use at your company". To its right is an empty text input field. Below this are two buttons: "Add Tool" and a small "Add Tool" button with a plus sign.

After you entered one or more tools (methods) and quit the form, an additional button appears on the start-up screen: to fill in tools (methods) questionnaires.

ITEA

Start-up form

Please choose the name of your company

T.U. Eindhoven

Enter Tools in use at T.U. Eindhoven

Enter Methods in use at T.U. Eindhoven

Fill in General Questionnaire

Fill in V Model Questionnaire

Fill in Tool Questionnaires

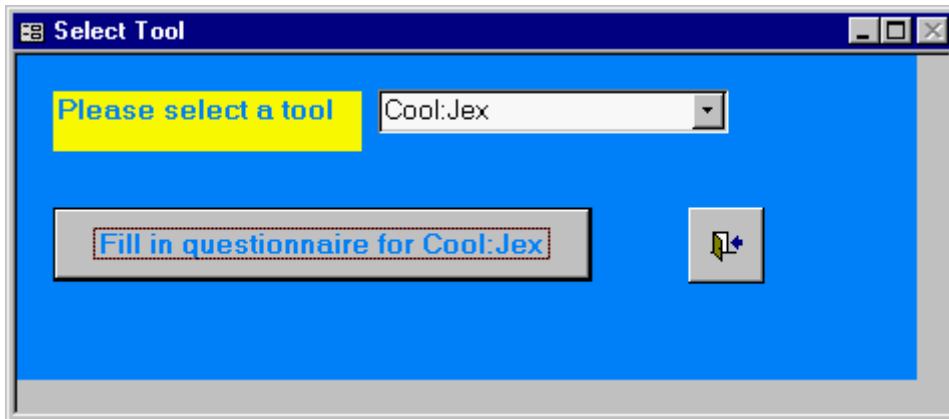
When these buttons are pressed, a form to select a tool respectively method will pop up.

Select Tool

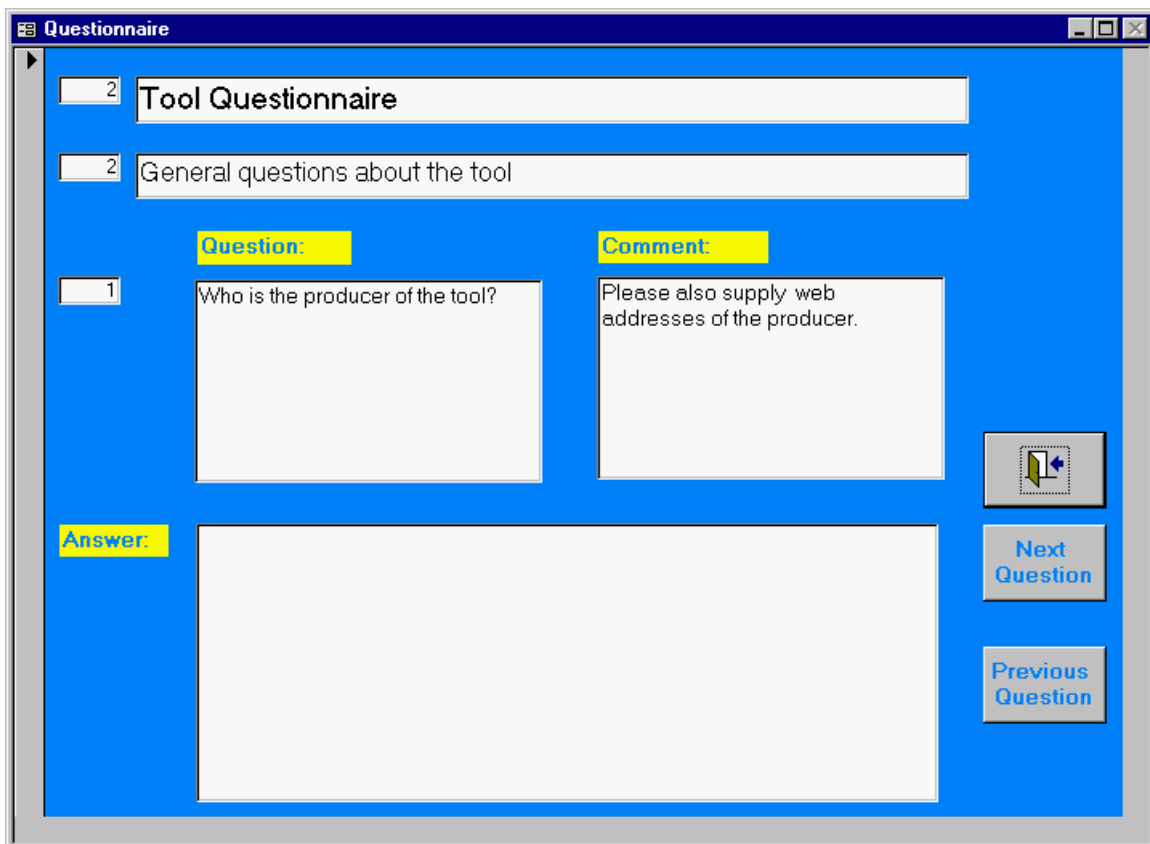
Please select a tool

The tool (method) can be selected again by clicking on the triangle and scrolling to the appropriate tool. After selecting a tool or method, a button to open the questionnaire for this tool or method becomes visible.

ITEA



When you click this button, the questionnaire form pops up.



You can fill in the questionnaires in any order you like. Individual answers can be 64000 characters long, so it is unlikely that you run out of space. There are buttons to go to the next or to the previous question. You can use the quit button at any time and resume answering the questions at a later time. Answers to questions can be modified afterwards. After you close the questionnaire with the “Quit” button, please also close the “Select Tool (Method)” questionnaire with the “Quit” button unless you want to fill out more questionnaires.

As to the V Model Questionnaire, after having made a general inventory of tools and methods, we want to investigate their deployment in more detail. Often, one tool (or method) is applied at several stages in the development process. Usually, at each stage only a specific part of the functionality is used. The purpose of the V Model Questionnaire is to register this information. The (slightly extended) V Model is shown on the last page of this document. The V Model is only used as a structure to order which aspects of which tools (or methods) are used at each stage of the development

ITEA

process – it does not necessarily mean that the V Model is adopted in practice.

We would like you to adopt the following format for this information at each entry of the V Model:

<Method1> - <aspect1>, <aspect2>,

<Tool1> - <aspect1>, <aspect2>,....

<Tool2> - < aspect1>, <aspect2>,....

<Method2> - <aspect1>, <aspect2>,

<Tool3> - <aspect1>, <aspect2>,....

<Tool4> - < aspect1>, <aspect2>,....

Where Tool1 and Tool2 are based on Method 1, while Tool3 and Tool4 are based on Method 2.

What we would like to see is the functionality of the method that you actually use, e.g. the use of class diagrams. For the tools, you do not have to mention that you use e.g. the class browser (this follows from the aspect “class diagrams” of the method). Only mention which functionality of the tool you use that is not implied by the method. If the tool enforces e.g. model checks automatically, this does not need to be mentioned either. So, for example (this example is also included in the comment with each V-model question):

OMT - class diagrams, event trace diagrams, state transition diagrams

Cool:Jex - local model check, global model check

UML - class diagrams, collaboration diagrams

Rational Rose – standard only

In this (academic) example, the tool Cool:Jex is used for OMT and the optional model checks are used in addition to the standard features of Cool:Jex. In the case of UML, only the standard options of Rational Rose are used. The model checks are enforced by Rational Rose itself, so they do not need to be mentioned.

In the exceptional case where more than one different development trajectory is used at your company, please make a copy of the database to store the information for each trajectory.

If you feel that the format of the present questionnaire did not enable you to enter some essential information, please let us know when you return the database.

Please return the databases by November 15th 1999.

Best regards,

Erik Luit

Ruurd Kuiper

Practical steps

Put the file Initialquestions.mdb on your hard disk (preferably not on a laptop because of the ensuing screen limitations).

Fill in the questionnaires (use Quit and Stop buttons to save the information entered!). *Please make sure that all questionnaires are completed.*

Send back the file Initialquestions.mdb to us at: E.J.Luit@tue.nl

If you need more information, please contact us at the same email address.

ITEA Title: DESS

ITEA Title: DESS

Preparation

User Requirements

Software Requirements

Architectural Design

Detailed Design

Coding

Unit Tests

Integration Tests

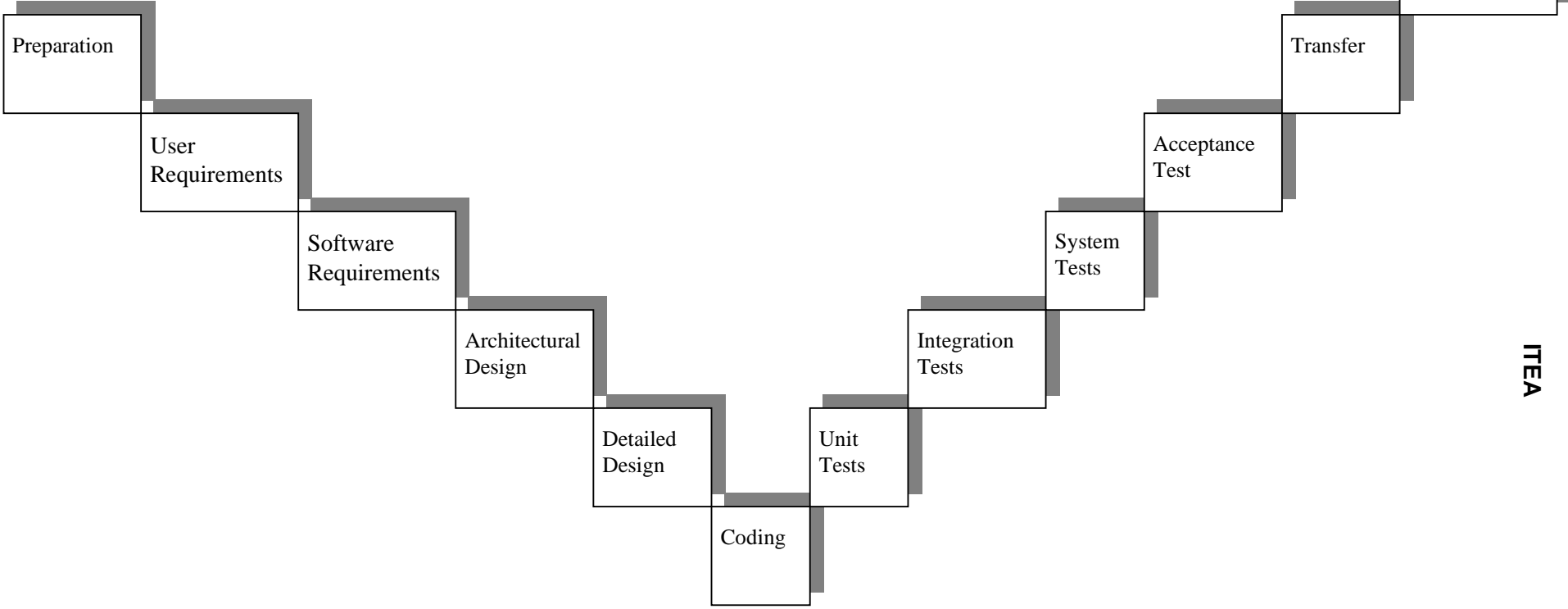
System Tests

Acceptance Test

Transfer

Operations and Maintenance

ITEA



Appendix C. Description of the Tools and Methods Database

This document gives a description of the structure of the database per July 2000 insofar this is relevant to retrieve interesting data from the database. This means that e.g. the code to merge the databases from the different companies is not described. Some remnants of a previous incarnation of the database are still left – these may come in handy for the next version. These are not described either. Relations between the fields of different tables are obvious – related fields have the same name in all tables.

Tables

Table	Fields					
tblCompanies	Company Id	Company				
tblTools	ToolId	ToolName	Tool TypeId			
tblToolTypes	ToolType Id	ToolType Name				
TblCompany Tools	Company ToolId	ToolId	Company Id			
tblMethods	MethodId	Method Name	Method TypeId			
tblCompany Methods	Company MethodId	MethodId	Company Id			
tblCategories	Category Number	Category	Category-HeaderId			
tblCategory Headers	Category HeaderId	Category-Header				
tblQuestions	QID	Question	Comment	Category Number	Question Number	Tool TypeId
tblInitial Answers	QID	ToolId	MethodId	Company Id	Answer	

Comments:

The ToolTypeId and MethodId are not (yet) used. Both are set to 1 (unassigned) for all tools and methods.

The CompanyToolId and CompanyMethodId are the link between the tools and methods in use at a company and the company.

Tools mentioned as alternatives are not yet entered in a table: they can only be found in the answers to the question about alternative tools.

ITEA

Queries

Query	Description
qryAlternativeTools	Picks out the answers to the question that asks for alternative tools
qryAnswersVModel	Ditto for answers to the V Model questions (with phases)
qryGeneralAnswers	Ditto for the answers to the general questions – query for rptInitialAnswers
qryInitialQuestions	The form frmInitQuestions is based on this query
qryMethods	Returns all methods in use by partners – used for frmMethodsInUse
qryMethodsAnswers	Returns all answers given to the methods questions – query for rptInitialAnswers
qryMissingFunctionality	Returns the answers to the question about missing functionality and the tool concerned
qryToolsDescriptions	Returns all descriptions given for the different tools in use
qryTools	Returns a list of all tools and their type Ids
qryToolsAnswers	Returns all answers given to the tools questions – query for rptInitialAnswers
qryToolsatCompanies	Answers to the tool description question (with tool and company name)
qryToolsOfCompany	Tool name, company name, ToolId and CompanyId
qryVModelAnswers	Answers to V Model questions with question and category number – query for rptInitialAnswers
qryUpdInitialQuestions	Can be used to modify or add questions (not finished?)

ITEA

Forms

Form	Description
frmAddCategory	To enter company names into the database - called from frmMainMenuAdmin
frmCompanies	Form to display all questions and to collect the answers. Based on qryInitialQuestions- called from frmStart
frmInitQuestions	Form to display all questions and to collect the answers. Based on qryInitialQuestions - called from frmStart
frmMainMenuAdmin	Form to add questions, companies, tools etc (incomplete). Invokes forms: frmCompanies, frmTools, frmToolTypes, frmCompanyTools (not implemented yet), frmRemoveCategory, frmAddCategory, frmUpdateQuestions
frmMergeDatabases	To merge the databases of partners
frmMethodsInUse	To enter methods in use prior to filling in the questions on methods – called from frmStart
frmMethodsQuestionnaire	Select a method to answer questions about – called from frmStart
frmRemoveCategories	To remove a category – called from frmMainMenuAdmin
frmStart	Startup form for answering questions – invokes frmToolsInUse, frmMethodsInUse, frmInitQuestions, frmToolsQuestionnaire and frmMethodsQuestionnaire
frmTools	To enter tools – called from frmMainMenuAdmin
frmToolsInUse	To enter tools in use prior to answering questions – called from frmStart
frmToolsQuestionnaire	Select a tool to answer questions about – called from frmStart – invokes frmInitQuestions
frmToolTypes	To add tool types – called from frmMainMenuAdmin
frmUpdateQuestions	To add/modify the questions – based on qryUpdInitialQuestions – called by frmMainMenuAdmin

Reports

There are essentially two reports in the database: rptQuestions and rptInitialAnswers. RptQuestions shows all the questions and the accompanying comments. RptInitialAnswers does not (yet) work for the merged version of the database: it requires an outer loop to iterate over the companies. RptInitialAnswers is based on four subforms: rptGeneralAnswers (based on qryGeneralAnswers), subrptVModelAnswers (based on qryVModelAnswers), subrptToolsAnswers (based on qryToolsAnswers) and subrptMethodsAnswers (based on qryMethodsAnswers).

ITEA

Appendix D. Categorized Tool List

Legend

+: in use

?: considered

Used	ToolName	Description
	Project Management	
+	MS-Project	Scheduling work
?	PSN	
	Configuration Management	
+?	Clearcase	Configuration management for documentation, source code and tests
+	Configuration Scripts	Build configuration management for generating application for specific embedded platforms.
+	continuous	Version control and problem tracking
+	CVS	Source Code version management
?+	PVCS Version Manager	Project configuration management
?	Visual SouceSafe	Version control
	Documentation Support	
+	Document Template	Template and guidelines for reports
+	SoDA	Automatically generates documentation (FrameMaker or Word) from template and information from Rose, Apex or TestMate
	Requirements Management	
+	<partner 1> Traceability Tool	Ease the traceability demonstration between System requirements, Software Requirements and the associated test coverage
+?	DOORS	High configurability capability. "Easy" integration with any tool. Most of the tool vendors provide interface to it.
+?	icCONCEPT RTM	Manage project requirements. Report changes, custom report generation, version control for requirements, compare versions, on-line information system to relay information.
+?	RequisitePro	Requirement version management. Discussion group change proposal mechanism. Organize, prioritize, trace relationships and track changes to requirements. Establish and analyze traceability. Assess coverage for verification and validation. Requirement metrics capabilities.

ITEA

	Problem Tracking	
+	<partner 1>-Softpro	Software Problem report and tracking
	PVCS Tracker	Change requirement management (engineering change requests) Defects management
	BoundChecker	
	Prototyping /Interface design Tools	
?	ALTIA	
+	Vaps	Human-Machine Interface prototyping and code generation; rapid testing with interpreter
	Modeling Tools	
	<i>General Modeling Tools</i>	
+?	Matlab Simulink	Block oriented design of continuous control systems, extended with discrete aspects by finite automata (stateflow)
+?	Matlab Stateflow	State machine modeling tool
+	Scade	Graphical specification of logic; code generation (C or Ada).
+?	Statemate	State machine modeling tool; code generation (C, Ada, VHDL, Verilog).
	<i>Esterel Modeling Tools</i>	
?	Esterel Studio /GE /GS /CG	Will be used by Thomson CSF for DESS
	<i>MOOD Modeling Tools</i>	
	MME	* graphical and textual editors * automatic generation of the application documentation * automatic checking of the obtained design model * automatic generation of the corresponding MML code for application implementation
	<i>UML/OMT Modeling Tools</i>	
?	Artisan RT studio	Multi-user suite of development tools for modeling and component-based development for real-time systems. Eases validating system behavior with models.
+?	Cool:Jex	UML/OMT modeling tool; code generation, reverse engineering (tool also known as Cayenne ObjectTeam, Telelogic Tau).Shared repository. Model checking capability.
+	Objecteering/UML Modeler	UML modeling tool, code generation, reverse engineering
+	ObjectGEODE	UML, MSC and SDL modeler; analysis, design, verification and validation (simulation, code generation and testing of real-time and distributed applications)

ITEA

?	Paradigm	UML-based component modeling. Round-trip engineering, extensions to UML for requirements management and database design. Shared repository.
?	Rhapsody	UML-based. Analyze, model, design, implement and verify the behavior of embedded systems software. Complete code generation (C, C++, micro-C for micro controllers; customized to e.g. trade speed for size) and animation of design models, model checking. Support for RTOSs.
+	Rational Rose	UML Modeling; code generation and round-trip engineering
+?	Rational Rose RealTime	UML Modeling with real-time extensions high performance complete code generation, and model execution; support for RTOSs
?	Select	Considered as alternative for Rational Rose
?	Software through Pictures	Multi-user UML (+ others) analysis and design environment with a shared repository. Full lifecycle support. Syntax and semantic checks for diagrams and model. Reverse engineering, code generation.
?	System Architect	UML/SASD/.... Modeling. Multi-user.
+?	Together/J	UML modeling tool with simultaneous round-trip engineering for Java; supports patterns, components, team development. QA tracking for requirements, metrics and audits.
	Model Checker	
+	KRONOS	Inputs: a formal timed automaton and a proposition in formal modal timed logic Output: set of control states of the automaton Checks: temporal logic properties and response time
?	Esterel Studio/FVT	
	Code Generator	
+	CCG Lite	C generator for VAPS
	Utilities	
+	Makefile Generation	Generation of makefiles; automation of code compilation across platforms and projects
	Development Environment/Compiler	
+	Apex	IDE for Ada and C++ (many features)
+?	C++Builder	IDE for C++
+?	CodeWarrior	IDE for C, C++ and Java
+?	DIAB Compiler and SingleStep Debugger	C/C++ compiler + debugger
+	DSL Compiler	Compiler for DSL (Domain Specific Language)

ITEA

+?	EDE + CrossViewPro	IDE for embedded applications + debugger
?	Esterel Studio /C	
+?	Forte for Java	IDE for Java
+?	GNU compiler	Compiling tool chain with cross compilation possibility
+?	Jbuilder	IDE for Java
?	JDK	IDE Java
?	Kawa	IDE for Java
+	mcc68k	IDE for embedded systems
+?	Processor Expert	SW and HW oriented co-design environment (component oriented)
+	PRISM+	graphically oriented (IDE) for developing and delivering embedded systems.
+	PSPECTRA	Portable environment for designing software radio applications
+	SAXO	Re-targetable C compiler dedicated to embedded applications
+	SAXO-RT	ESTEREL compiler that can process some hard real time annotations
+	Sniff+	IDE for C, C++, Java and ADA
?	Source Navigator	Considered as alternative for Sniff+
+	Tasking C166 EDE	IDE for embedded applications
+	Visual C++	IDE for C++
?	VisualCafe	IDE for Java
+	WorkShop	IDE for C/C++.
	Code Quality	
+	AdaAnalyser	Static analysis tool for Ada Coding Standard verification Metric computation Helps code walkthrough
+	AdaRepair	Automatic repair of common coding standard violations.
+?	CMT++	Complexity Measures Tool for C/C++, static analysis/metrics tool for C and C++
+?	Logiscope	Check software against programming rules and asses the overall software quality
	Test /Validation Tools	
?	AdaTest	Considered as alternative for TestMate
+?	Attol UniTest	unit testing for real time embedded systems.
+	CTC++	Test Coverage Analyzer for C/C++
+	CTE	systematic design of test cases
+	Pure Coverage	Code-coverage analysis for Visual C++, Visual Basic and Java
+?	QACenter	Various functionalities for testing: understand how applications will perform in production; plan testing process; verify application functionality;

ITEA

		validate data and test results; manage resolution of defects uncovered during testing.
+	TAXYS	Timing validation tool.
+	TESSY	Test tool for C, e.g. test case design, test data description, test execution, regression testing
	TestDirector	Planning test plans, test design and test cases. Management and execution of test sessions. Tracking anomalies arising from tests. Management of manual and automatic tests (WinRunner).
+	TestMate	Test Case Management and Generation Test automation, Coverage computation and Test report building
	Run-Time	
+	C-SMART	Ada RunTime kernel for safety critical applications
+	MARK	Ada RunTime kernel for safety critical applications
?	RAVEN	Considered as alternative for MARK
+	Text Editor	
	Emacs	
+	Word	
	Spreadsheet	
+	Excel	
	Database	
+	Access	Database (used for requirements mgmt)

Appendix E. Reported Tool Shortcomings

Answers to the question:

Are there aspects of your product that are not well supported by the tool?

Answer	ToolName
Project Management	
Tracking is not very good, assignment of multiple tasks to the same person (parallel tasks)	MS-Project
Configuration Management	
Low performance, when using clearmake, which computes some information for the share of productions by multi-users and labelizations. For the production step only we generally use gnumake.	Clearcase
code restructuring	CVS
Documentation Support	
Requirement traceability	SODA
Requirements Management	
The dynamic update of requirements is currently not possible. Extraction to document is not easy and based on complex M\$-Word macros.	<partner 1> Traceability Tool
Requirements in graphical form	DOORS
The users needs to know the internal language to extract the information needed for the traceability. The tool lacks of a friendly user interface.	DOORS
Requirements in graphical form	icCONCEPT RTM
Requirements in graphical form	RequisitePro
Problem Tracking	
Traceability to the code. What changes in the code come from what PR.	<partner 1>-Softpro
Interface with other tools	PVCS Tracker
Prototyping /Interface design Tools	
Qualification, traceability of the requirements. Therefore we are looking at QCG, DesignDoc and DOORS.	Vaps

ITEA

Modeling Tools	
<i>General Modeling Tools</i>	
Reasoning about the prospective resources (for example memory, time) needed by the designed software	Matlab Simulink
Timing analysis in the sense of deriving worst case execution times for implementations of any models on a certain hardware.	Matlab Stateflow
The verification of timeliness properties (under consideration of real execution times on the target system) is not supported by Statestate.	Statestate
Realistic visualization of prototypes	Statestate
Timing constraints	
Memory constraints	
Performance evaluation	
Timing constraints, traceability to High Level Requirement.	Scade
<i>MOOD Modeling Tools</i>	
Incomplete code generation	MME
<i>UML/OMT Modeling Tools</i>	
Real time aspects	Objectteering/UML Modeler
Our special focus is on	ObjectGEODE
- code generation with minimal footprint and - timing analysis in the sense of deriving worst case execution times for implementations of any models on a certain hardware	
These were not supported optimally in the former version. We have not yet checked possible advances in the latest version.	
Several: Timing constraint, Memory Constraint Code Generation from all diagrams	
real time aspects	Rational Rose
Timing constraints	Rational Rose
Memory constraints	
Communication protocols	
Performance evaluation	
yes (code generation is not adaptable)	Rational Rose
timing constraints, memory constraints	Rational Rose
Insufficient generation of code	Rational Rose
Insufficient for the design of real time systems (timing constraints, memory constraints, ...)	
Yes, code generation and documentation.	Rational Rose
Constraints in general	Rational Rose
Rose RealTime does not support the complete UML. It would be a desirable feature.	Rational RealTime Rose
See Rational Rose questionnaire	Together/J

ITEA

Code Generator	
Qualification (the generator is not qualified so the verification and validation phase is complex), traceability toward High Level Requirements.	CCG Lite
Development Environment/Compiler	
Yes, no support for the last C++ standard ISO proposal (RTTI, Standard C++ library, ...); memory constraints	DIAB Compiler and SingleStep Debugger
Yes The tools aims to generate C code from a description of a driver. At this time some performance issues are not solved by the tools (efficiency of the code generated and monitoring of the code generation process). Moreover, the tools is not especially designed to deal with embedded and real-time systems.	DSL Compiler
Real time and embedded aspects are not taken into account	GNU compiler
There's too many versions of the product due to some problems of reliability.	mcc68k
Verification of properties in algorithms implemented.	PSPECTRA
Most of the effort has been devoted on high level transformations to speed up CPU intensive parts of the application. The optimization of classical control code is less developed.	SAXO
Today, the estimations of the duration of the actions have to be provided manually.	SAXO-RT
Debugging, Profiling etc	Visual C++
Locating Run Time errors	Visual C++
Some of tools that we use aren't yet compliant with Workshop version 5 (ex: llogViews 3.1) and we have some linker troubles with the use of the STL (Standart Template Library). That's why the majority of the projects use the version 4.	WorkShop
Code Quality	
Bad performance on NT platform for C++ code analysis.	Logiscope
Test /Validation Tools	
The main use of Attol Unitest is the unit testing of inputs and outputs of a function. This tool isn't yet adequate for object concepts of class and inheritance. You can't inherit of the base class tests scenarios, in a derived class.	Attol UniTest
Sequence testing	CTE
Automatic software "good by construction" synthesis issue.	TAXYS
Testing of continuous signals, test script language for sequence testing	TESSY
Interface with other tools	TestDirector
No for what we use the tool for. However if you consider the whole verification and validation activity, several part are missing (covered by other tools)	TestMate

Appendix F. Combined Methodology Questionnaire

DESS Work Package 5 – Software development process

1. Introduction

This questionnaire is to be used in the preparatory actions for WP5 but also for WP1.2. WP5 aims at the following:

- Investigate the existing formalisms that could be used for representing the process models. Investigation should span both the formalisms currently used within the consortium (if any) and the existing standards. Among the latter are outstanding IDEF0 (or IDEF3) and RUP (the Rational Unified Process). RUP is not a standard, but it is quite easy to predict that it will become very popular.
- Investigate the existing formalisms that could be used for representing the process artifacts (i.e., specifications, design document, code, test cases, error reports, etc.). Investigation should span both the formalisms currently used within the consortium (if any) and the existing standards. Among the latter are outstanding the Entity/Relationship diagrams (in all the numerous variants) and UML.
- Produce process and product models concerning the current *actual* situation in the partners' organizations at various granularity levels. Note that it is important to model the current situations and the desired situation (i.e., the one that DESS should help to achieve). Note: the models we need are those in actual use. Models described in books that are never read are not relevant. The models should include as far as possible indications concerning tools used (especially environments, configuration management systems, etc.), constraints, strategic goals, standards adopted in production, etc.
- Investigate the existing tools and environments that are available (commercially or as research products) in order to support the development process.
- Collect information on tools currently used within the consortium and/or preferences and constraints concerning tools.

Note that the last two points have already been assessed in less detail in WP1.2.

The questionnaire is organized in two main parts: the first one aims at representing the current situation of DESS partners' development process; the second part aims at collecting the requirements for the DESS process to be defined and supported.

How to use the questionnaire

Questions are expressed in a rather free form. There are no predefined answers. In general it is possible to answer several questions by means of a few paragraphs. The goal is to understand the situation, so the important issue is that the global picture is clear, even though some question is not clearly/fully answered.

Please read all the questions before starting to answer.

The whole set of answers is expected to range from 3 to 10 pages. Additional material taken from organizations' standard documentation is welcome.

ITEA

2. Current situation

2.2 Product

- 1 What kinds of product does your organization produce?
- 2 Does your organization use a standard notation to describe products?
- 3 Is there a reference definition for products in your organization?
- 4 Does your organization have a reference description for the artifacts (requirements, specifications, design documents, etc.) produced/used throughout the whole development process?

Can you provide a few samples of these descriptions?

2.2 Process

- 2 Describe structure and phasing of the process, e.g., waterfall, V-model, iterative approaches like the spiral process (repeatedly going through phases like Analysis/Design/Coding/Testing), other, differently structured, incremental approaches. Also indicate the larger phasing of the process, i.e., the kind of larger milestones that are reached in the process, e.g., those occurring after some iterations in the iterative approach.
- 3 Indicate time allocated (percentage) for the different activities at different stages of the process (e.g. for Analysis/Design/Coding/Testing in the different stages of the spiral model).
- 4 Broad description of the main (formal) notation and tooling used in the larger phases of the process. It also should be indicated where informal notation is used.
- 5 Has your organization defined a reference process
- 6 Does your organization adapt a reference process to the specific projects depending on projects' features and needs?
- 7 Does your organization's process conform to any standard?
- 8 Has your organization been certified with respect to process quality (ISO9000, CMM, etc.)?
- 9 Own assessment of how what you consider as the best practices [see, e.g., Kruchten00] are followed. Preferably for each item a few explanatory lines as well as an assessment of current status:

Is there a prescribed process? (P - yes/no.)

Is it followed? (F - yes/partially/no.)

Is it efficient and effective? (E - satisfactory (+), moderate ([]), to be improved (-).)

- Iterative development (Example answer: P(y), F(p), E([]).)

- Explicit requirement management

- Use of Component Based architectures

- Visual modeling (e.g., class diagrams)

- Explicit software quality verification (not just correctness, also structure, documentation etc.)

- Controlling changes.

ITEA

- 10 Own assessment on current status of handling key aspects for DESS.
 - Real-Time
 - Resource constraints
 - Embedded system aspects
 - Formal methods
 - Component-Based
 - Code generation
 - Validation and Testing.
- 11 Is your organization pursuing a process improvement program? If so, following which methodology?
- 12 Does your organization carry out regularly process/product measurement?

3. Requirements for the DESS process

- 1 What are the issues of process development that you would like the DESS process to address (and thus –hopefully– improve?)
- 2 In particular, what technical and managerial activities should be supported by the DESS process?
- 3 What of the existing tools do you consider important to integrate in the DESS process support?
- 4 What are your needs in terms of cooperation and communication?
- 5 What are your needs in terms of (geographically) distributed development management?

We would also welcome a short comment on current status and desirability of formal methods in combination with automated verification.

PLEASE INDICATE WHAT YOU WOULD LIKE TO BE ADDED TO THIS QUESTIONNAIRE.
For example, if in 8 or 9 you find that further items would be useful, please add such.

Reference

Kruchten00 P. Kruchten, The Rational Unified Process, An Introduction, Addison-Wesley, 2000.