Integrating Software Engineering Technologies For Embedded Systems Development

Rini van Solingen
Project manager of the MOOSE project

CMG, The Netherlands, rini.van.solingen@cmg.com

Abstract. In order to solve the ever-increasing gap between required productivity and quality of embedded software on the one hand, and the industrial capabilities in fulfilling these requirements on the other hand, several European organisations decided to tackle this problem together. By forming the MOOSE consortium ("Software Engineering Methodologies for Embedded Systems"), they started an effort for integrating the available tools, techniques, methods and processes involved in embedded software development. The main starting point of MOOSE is that the available technologies for embedded software development are too fragmented; time, effort and money can be saved by supporting a better integration over development phases and between technologies. This paper presents the starting points of the MOOSE project and introduces some first concepts for achieving the ambitious goals set by the consortium.

1 Introduction

With the ever increasing penetration of embedded systems in society and the related increase in investments by industry to develop such systems, the investments in embedded software engineering technologies increases as well. Forecasts [2] predict continuing increase in embedded systems specific technologies. This paper uses the term 'technology' for any method, technique, process or tool that can be used to support a certain development activity.

The market for software engineering technologies is largely fragmented. There is no clear market leader, and there is no supplier present that fully supports the whole development chain of embedded products. There are different suppliers for requirements engineering technologies, different vendors for design technologies, etc. Most dominant is the sales of software tools. Tools imply to support or be supported with a methodology. Some suppliers provide methodologies with their tools, while others support generic methodologies, such as UML (OMG's Unified Modeling Language) or MOF (MOG's Meta Object Facility).

As the market of technologies is fragmented, so are the technologies themselves. Technologies in general are stand-alone solutions for specific problems. As embedded software development is collection of complex and technical problems, several technologies are applied in parallel [3]. It does not take very much time or experience to observe that this lack of integration is a cause for problems too. Technologies are used separately but depend on each other, interfaces are not defined, inconsistencies occur, etc. Despite this loss of quality due to lack of integration, there is moreover a loss of time, effort and money due to duplication, redundancy and cost of non-quality. As time pressure is prominent in many market domains where embedded systems are sold, this indicates a potential gain in time-to-market. Time-to-market gains in embedded systems development cause high potential revenue gains due to earlier market introduction and therefore deeper market penetration. Industrial companies therefore support initiatives for time-to-market reduction strongly. This is the starting point of the MOOSE project.

2 MOOSE

In 2001 several companies joined their efforts in establishing a consortium for increasing integration in embedded systems development. A project proposal has been set-up, which was rewarded with the ITEA label within the ITEA programme [4]. The MOOSE project runs from March 2002 until February 2004, consists of 13 organisations in 3 European countries, with a total effort of more than 100 person-years and a budget over 15 million Euro.

MOOSE Partners

The MOOSE partners in those three European countries are:

Finland:	
Nokia	Application partner
Solid	Application partner
University of Oulu	Technology partner and exploitation partner
VTT Electronics	Technology partner and exploitation partner
Netherlands:	
ASML	Application partner
CMG	Application partner and exploitation partner
Océ	Application partner
Philips	Technology partner and application partner
Technical University Delft	Technology partner
Spain:	
Datapixel	Application partner
ESI	Technology partner and exploitation partner
SQS	Technology partner
Team Arteche	Application partner

Three roles are distinguished for the partners:

- Application partner: will use the results of the project within their organisation for embedded software development projects, and will provide input to the project by

giving feedback on practicality and validity of project results. The application partners also aim to support the technology partners by providing requirements that help in optimising the industrial relevance of the project results.
- Technology partner: will bring knowledge and expertise on software engineering methodologies to the project and facilitate the validation activities at the industrial partners
- Exploitation partner: will market and promote outcomes of the project after its finalisation

MOOSE Objectives

The MOOSE project aims at increasing embedded software development productivity and product quality. By making development more productive and increasing product quality, development cycles become shorter and cost (and time) of non-quality drops, also causing shorter development cycles. Dependent on the specific market situation, embedded systems suppliers can decide to use the available time for earlier market introduction, additional development, or establishing cost reductions. Looking at the current situation in most embedded system markets, earlier market introduction will be the most frequently selected option.

In order to work towards this aim, the MOOSE project has a set of project goals. These goals are to:
- Integrate systems and software engineering, requirements engineering, product architecture design and analysis, software development and testing, product quality and software process improvement methodologies into one common framework and supporting tools for the embedded domain. This framework will enable the structuring, management and evaluation of embedded software projects in order to guarantee the attainment of its objectives.
- Develop new or enhance existing systems and software engineering methodologies to be integrated in this framework, with the aim to improve product quality and software development productivity through optimised integration and interfacing. Product quality (a) will be defined clearly and measurable and (b) has to be guaranteed by co-ordinated process adaptation, product architecture design and quality assuring measures.
- Validate the operational strength of the framework by extensive usage in different types of embedded software projects (e.g. automotive, telecom, consumer electronics).

MOOSE Project Structure

For project management purposes the project has been subdivided into four related but different workpackages. Each workpackage has a workpackage manager, and several partners in the consortium cooperate in several different tasks in these workpackages. The four workpackages of the MOOSE project are:
- Workpackage 1: Framework Development and Validation. In this work package a common framework for integrating several systems and software engineering

methodologies for embedded systems will be developed and validated. The framework will provide a means for each embedded software development project to select the most beneficial set of software engineering methodologies in order to achieve its goals with respect to cost-effectiveness, time-to-market and product quality. As such it is used for self-evaluation of projects, or decision making for technology-update and improvement.

- Workpackage 2: Systems and Software Requirements Engineering. This workpackage focuses on tailoring, customising and combining existing systems and software requirements engineering and quality techniques, methods and tools to complex embedded systems and software development. The question "how systems requirements engineering relates to software requirements engineering and how these should co-operate over which kind of interface" is a central theme in this workpackage.
- Workpackage 3: Embedded Product Architectures. This work package focuses on tailoring, customising and combining existing product architecture definition, design and development methods and techniques into embedded systems development, and to interactions between systems and software requirements engineering and improvement management within the framework.
- Workpackage 4: Improvement Management. This work package focuses on customising and combining existing improvement methods and techniques to embedded systems development. Typical examples of these methods and techniques are: Software design or code measurement, Product quality assessment and Process capability assessment. Product and process assessments are basic tools for improvement management, which only makes sense when baselines exist to compare to. Measurement is needed to make these comparisons. Specific emphasis is put on the software implementation and testing tasks, as they are highly relevant for embedded systems development.

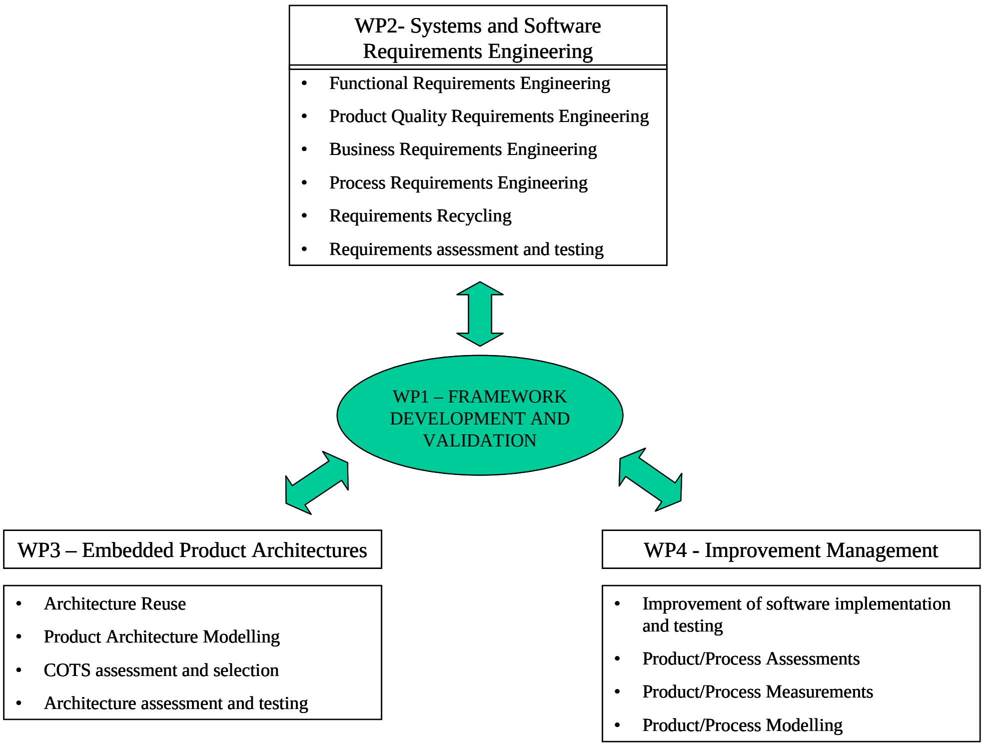

Fig. 1. Workpackage organisation of the MOOSE project

The project will focus on three main topics within the lifecycle of an embedded system: requirements engineering, product architecture, and improvement management. Testing is an integral part of all three work packages as testing of for example requirements is dealt with in WP2, while testing of architectures is dealt with in WP3. Software implementation (coding, commenting, etc.) is being addressed in WP4: improvement management. This work package focuses to improvement of development processes, and as coding is one of the most dominant processes within embedded software development, it is dealt with fully in WP4, together with the testing aspects of software implementation.

MOOS Output results

The main visible results of the MOOSE project will be:

- The framework with defined component interfaces which guarantees an effective software development approach for a specific situation and which enables future innovation and expansion. This framework is supported with a decision model for selecting appropriate software engineering methods, techniques and tools, which

will be publicly accessible through a web-based repository that is maintained and shared based on open-source principles.
- Experiences with applying the framework and its components in many industrial projects, within or maybe even outside the consortium, in order to validate its effectiveness and efficiency.
- A number of enhanced/new systems and software engineering methodologies and related tools that function as components in the framework (e.g. product assessment, process architecting, requirements engineering, etc.)

3 Embedded domain specific requirements

Though many software product and process technologies are already available, the embedded software domain puts specific demands to the application of these technologies. Dedicated research results and products are present for software architecture development and assessment, requirements engineering and validation, software process improvement, and tools to support all these technologies. However the major disadvantages of these technologies are that they do not take into account the specific needs for embedded systems and that they are applied “stand alone”, which in many cases is not very effective and leads to disappointing results.

The embedded systems industry puts specific demands to the usage of such methodologies, such as the large dependency on real-time features, limited memory storage, large impact of hardware platform technology and the related cost drivers of the hardware, etc. The existing software engineering methodologies do not distinguish the specific impacts or necessary customisation for the embedded domain, nor is it indicated how they should be used specifically for each specific area within this domain, i.e. automotive, telecom, consumer electronics, safety critical, etc. The embedded software domain puts dedicated pressure on these methodologies. Reasons for this are the high complexity of these products and the dependency in this domain on innovative highly technical solutions.

Furthermore, the embedded domain is much more driven by reliability, cost and time-to-market demands. This makes the embedded domain a specific area for which available generic methodologies need to be adapted.

4 Proposals for the MOOSE ‘integration framework’

The integration of existing software engineering technologies into one framework is required, in order to achieve an effective overall approach, which is customisable to the needs of each specific software development of embedded systems. Application of these methodologies requires mutual information exchange and co-operation along specific interfaces. The framework will arrange these interfaces and will make it possible to exchange methodologies, depending on the specific needs of the product and the organisation, as if they are components.

The framework contains: processes, methods, techniques, tools and templates. The framework also contains classification of business situation, project area, etc. that supports the decision making model. The fact is that there are too many methods or approaches available and, therefore, there is a need to reduce uncertainty in the decision making in selecting the set of methods, techniques and tools for a certain use situation. That means that support for decision making in choosing the method for a certain situation is needed.

Quite typical situation where the framework is needed is a project aiming to choose the most effective and efficient way for the development of embedded systems. The framework integrates knowledge about processes, methods, techniques, tools and templates into a decision making model that operates as outlined in the following picture.

Framework directions

In several workshops in the MOOSE consortium a rough outline of the framework has recently been constructed and is included in figure 2.

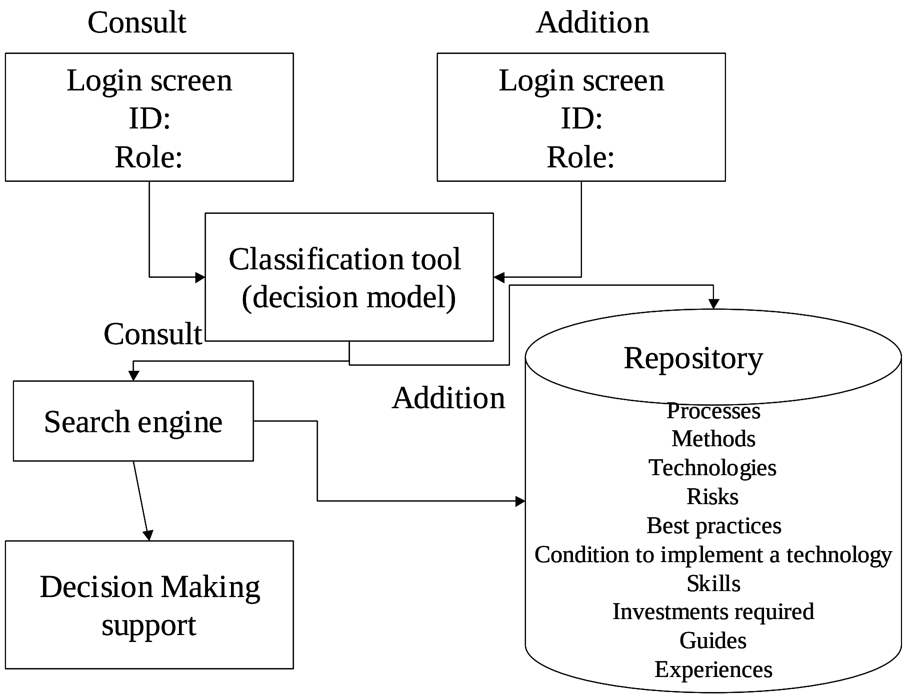

Fig. 2. First concept for MOOSE framework

The overall framework will be a web-repository including a decision model and filled with the findings from the inventory in the MOOSE project. The repository will be maintained based on open source concepts, guaranteeing knowledge from large variety of sources and continuous innovative expansion.

The web-repository is accessible in two modes: consulting (getting answers to questions) and adding (expanding the repository with results and experiences on technologies). The exact structure of the repository is currently designed but will contain commercial areas (for vendors) non-commercial areas for engineer experiences, best-practice areas, links, explanations, etc.

Furthermore there will be a classification tool included which supports in classifying embedded systems projects. The demands put on different embedded systems are varying largely, the structures of embedded software projects are also various. Grouping these projects together will have the effect that recommendations from the web-repository will be too general. Therefore, a classification mechanism will be constructed that subdivides the total set of embedded projects in several classes that share key-characteristics. A more specific focus can be provided, and projects and technologies can be better linked. There are several scenarios in which the web-repository will be used, for answering user questions such as:

- I want to know more about a certain technology. Tell me more about it.
- I have a certain ‘problem’. Which technology solves this problem?
- Which project type do I have? Please classify my embedded project?
- Which project types are there anyway? Show me the options?
- What type of embedded product do I have? Please classify my product?
- Which embedded product types are there anyway? Show me the options?
- What typical technologies fit to my project type?
- What typical technologies fit to my product type?
- What best practices are known for a specific project/product type?
- Can I contact other practitioners that have more experiences with a certain technology?

Further steps

In order to construct the framework as illustrated above, several steps are required. It would take too much time to present the full MOOSE project plan and steps in this paper. However, based on the current concepts for the framework there will be initiatives to:

- Construct a classification scheme for embedded software projects based on the inventory [1] in the companies of the industrial partners and in literature.
- Construct a list of important factors for embedded software to inventarise software engineering technologies, in order to list the most important issues for the known technologies (e.g. cost, learning time, support in multi-site development, etc.).
- Construct the web-repository based on the above concepts and fill the repository with knowledge on currently available and known technologies.
- Experiment in industrial projects with the usage of the framework during project construction

5 Conclusions

The MOOSE consortium has been formed in order to solve a challenging, complex, but highly relevant issue: integration between software engineering technologies for embedded systems. The project has started recently, but current intermediate results are promising. Especially the dominant position of industry in steering the output results of the project are expected to be a key-factor for success. Success of the project will be measured in the amount of support that can be provided to embedded software projects and that is experienced helpful by those projects.

The MOOSE project will continue to work on its objectives. If you want to keep in touch, please visit the MOOSE web-pages: http://www.mooseproject.org/

Acknowledgements

The author would like to thank all partners in the MOOSE consortium who have all provided their input to the construction of the concepts presented in this paper. Furthermore the national authorities in the three countries are thanked for their rewards in funding this project. Finally, regards go to the ITEA programme for enabling and supporting the MOOSE collaboration.

References

1. Graaf, B., Lormans, M., Toetenel, H., “Software Methodologies for Embedded Systems: An Industry Inventory”, submitted to PROFES 2002 conference.
2. IDC, IDC reports: 24809, 25176 and 24811, Http://www.idc.com/, 2001
3. Solingen, R. van, Product Focused Software Process Improvement: SPI in the embedded software domain, BETA Research Series, Nr. 32, Downloadable from: http://alexandria.tue.nl/extra2/200000702.pdf, Eindhoven University of Technology, 2000.
4. ITEA, Http://www.itea-office.org/