



ITEA
INFORMATION TECHNOLOGY
FOR EUROPEAN ADVANCEMENT

Agile software development of embedded systems

AGILE

Newsletter #1/2007



EXTENDED AGILE TEAM: PICTURE FROM THE CLOSING SEMINAR OF AGILE PROJECT HELD IN JANUARY 15-17, 2007, ROVANIEMI, FINLAND

You have in your hand the final newsletter from AGILE project. In this issue, we present you a cross-cutting overview of the results from the project. Companies and research partners have invested 171 person years between 1.4.2004 and 31.12.2006 in developing solutions and approaches that enable the agile software development of embedded systems.

The partners have executed a large number of industrial trials demonstrating the validity of the proposed solutions. Altogether close to 2000 engineers and managers were involved in 68 in-

dustrial trial projects from 17 European companies. As you will soon discover, 73 % of the pilot projects were either considered successful or very successful. Overall, we find that AGILE has provided the necessary proof that agile software development technologies with some necessary extensions and tool support are applicable also to complex systems engineering domain and have the potential to yield significant benefits to software-intensive industry in Europe. We have developed an interactive web portal for sharing the results of the project. Please visit <http://www.houseofagile.org> for details.

Agile solutions need yet to be scaled up to meet the needs large multi site and multi technology projects. Also, we concluded that new solutions need to be extended to business and innovation domain. This work now continues in Flexible Global Product Development & Integration (FLEXI) project. FLEXI-ITEA project was launched April 1st, 2007. You are welcome to visit our web site at <http://www.flexi-itea2.org>.

Agile greetings
Pekka Abrahamsson, VTT
Ko Dooms, Philips

AGILE

CONTENTS:

- XP2007 was a success!.....2
- Speeding up embedded software development.....3
- Playing With the Big Guys.....7
- ATO: a Tool Environment supporting Agile Model-Driven Development8
- AgileREQ 3.0 by SQS, S.A.9
- Development of an effort prediction model for an agile development10
- Enabling Agile Documentation11



Call for sponsors and papers

9th international conference on Agile processes and eXtreme programming in software engineering will be organized in Limerick, Ireland, 10-14, June 2008.

Deadline for all submissions is Jan-6th, 2008.

For opportunities and contributions visit <http://www.xp2008.org>

XP 2007 was a success!

Alberto Sillitti, Free University of Bozen-Bolzano
Giancarlo Succi, Free University of Bozen-Bolzano

The eighth edition of the International Conference on Agile Processes in Software Engineering and eXtreme Programming (<http://www.xp2007.org>) came back to Italy in a beautiful location. The conference took place on the lake Como from the 18th to the 22nd of June and it was a success.

The agile approach has become mainstream in the software industry in recent years. It is able to deliver business value early in the project lifetime and to successfully deal with changing requirements. It focuses on the delivery of running, tested versions of the system at a constant pace, featuring a continuous interaction with customers, and paying extreme attention to the human component of software development. A rapidly growing scientific and practical evidence show many quality gains including increased productivity, less defects and increased customer satisfaction.

XP200x conferences attract every year an increasing number of participants from all over the world and now are a de facto reference point of the international community interested not only in Extreme Programming (XP) but in all the Agile Methods. This year the conference offered the opportunity to people coming from the industry and the academia to meet, share ideas, exchange experiences, and enjoy one of the most beautiful locations in Italy.

According to the tradition of the conference, several activities took place including: research papers ses-



sions, workshops, tutorials, panels, and activities.

Research papers, workshops, tutorials, panels, and activities had been carefully reviewed and only the best ones appeared in the conference. The review process was tight and involved a board of more than twenty international experts with an industrial and an academic background.

The conference proposed very interesting keynotes:

- Piergiorgio Grossi, Operating Manager, Ferrari Racing Team gave the opening keynote, speaking on "The GranPrix starts at 2 o'clock: a race to race Software Development eXPerience".
- Kent Beck, the founder of XP and the most prominent scientist and technologist of Agile Methods in the world, will give the closing keynote on "Ease at Work."

- Another keynote was given by Olivier Lafontan, Managing Director, Agile Enablement Services, Exoftware, speaking on "Concept to Cash&Agile, and beyond."

The XP2007 conference proceedings are published in the Springer Lecture Notes in Computer Science (Springer LNCS). The book is available online. You can find information about it at <http://www.springeronline.com/978-3-540-73100-9> or access the online version at <http://www.springerlink.com/content/978-3-540-73100-9/> For more detailed information please refer to the following web site: <http://www.xp2007.org>

The 9th edition of the International Conference on Agile Processes in Software Engineering and eXtreme Programming (<http://www.xp2008.org>) will take place in Limerick (Ireland) from the 10th to the 14th of June 2008.

Speeding up embedded software development^(*)

Application of agile processes in complex systems development projects

Author: Pekka Abrahamsson, VTT (Technical Research Centre of Finland)



Cost-efficiency drives embedded software development. Rapid increase of software in embedded devices is challenging the European software industry. The ITEA AGILE project shows that the application of agile software development methods and processes can offer an up to 70% reduction in lead time and costs in a wide range of different industry sectors. The research in AGILE produced four new methods and 12 new tools to facilitate the uptake of agile development in embedded settings.

EMBEDDED SOFTWARE FACES CHALLENGES

Recent statistics show the amount of software is growing faster than Moore's law in terms of kilobytes of code in an embedded device. At the same time, we expect to see the number of embedded devices increase by ten-fold in a typical household within the next few years. Software development productivity, however, has not increased over the past decade. For this reason, one of the most obvious solutions for software companies is to transfer a significant portion of their software design and development to countries with lower labour costs.

There is a great deal of overhead in embedded software development. Only some 7 to 10 % of the features implemented in systems are often or always used; more than 60 % of features implemented are rarely or never used. Yet, each of the features still has equally to be defined, implemented, verified, tested and deployed. This causes significant overhead in the development process. Features lack priorities with most still considered as a number one priority. This challenges development further as software chang-

es come from many sources and are inevitable. In 3G network specification for example, more than 5,000 accepted change requests have been recorded to date.

AGILE APPROACHES ARE NEEDED

Standards-based software process improvement has been seen as a way to overcome these challenges. Capability maturity model integration (CMMI) is an example of a standard that is often used in software companies. According to the Software Engineering Institute, data shows this type of improvement is time consuming, often taking more than three to five years. Also, it is quite expensive, i.e. generally €0,000 to €5,000 for a single engineer (Jones 1999) [1]. Yet, only 0.21 % of about 1,000 studies in the IEEE database are able to show any type of return on investment (van Solingen 2004) [2]. Finally, more than 70 % of these software process improvement projects typically fail¹. Standards-based improvement approaches work better in a stable environment. Embedded systems markets in telecommunications, consumer electronics, automotive and even avionics industry sectors are highly volatile. The interactive digital television (IDTV) market is a good representative of modern technology that shows promise in consumer electronics business. Market volumes are currently low and new system must be released every 12 months. There is little room for time-consuming process-improvement projects that do not deliver significant business value within a few months. Incremental improvement is not sufficient in global competition. Rather, an innovative leap or novel approach is required in which improvements in terms of cost, quality and time-to-market exceed 50 %. Agile software solutions (Abrahamsson et al. 2002) [3] have shown strong promise in business-application and web-development environments. Less has been known its applicability in other fields.

DEVELOPING AND VALIDATING NEW AGILE METHODS

Agile software development processes and methods place emphasis on working software and stakeholder interaction. Agile methods have become increasingly popular in the field of software engineering within a few years. The focus in agile methods is on cost-efficiency, quality and time-to-market aspects. There is concrete evidence that agile methods operate well in pure software development projects. Gartner data from 2005² shows that one company in seven already uses agile methods in its R&D while 50 % of companies are aware of the methods. The applicability of these methods to stringent hardware-bound software development had not previously been investigated.

The research efforts in the AGILE project were directed into five different avenues:

1. Identification of the knowledge gap in existing solutions and current practice;
2. Performing early industrial trials to identify gaps between existing methods and embedded practice;
3. Development of new agile methods, tools and guidelines for embedded software development;
4. Carrying out a set of validation trials for new methods, tools and guidelines; and
5. Packaging the approaches and lessons learnt in a Wiki-based agile software development framework of embedded systems.

EARLY TRIAL RESULTS: KNOWLEDGE AND PRACTICE GAP

A survey in 35 projects in the AGILE consortium performed in late 2004 revealed that 60 % did not report any use of available agile practices in their respective projects (Salo and Abrahamsson 2007) [4].

About 80 % were not aware of an existing ag-

^{*} Reprinted from ITEA Innovation Reports, ITEA Project Agile, March 2007, http://www.itea2.org/innovation_reports

1. <http://www.sei.cmu.edu/sema/>

2. Corporate IT Leads The Second Wave Of Agile Adoption, 30 November 2005, Forrester Research, <http://www.forrester.com/Research/Document/Excerpt/0,7211,38334,00.html>

ile method called Scrum (Schwaber and Beedle 2002) [5] at the time. Despite the rate of unawareness, general belief was that the practices would be useful if applied. As an example, 77 % of respondents having experience with Scrum practices found them beneficial. At the time, however, very little experience had been collected. This served as a starting point for deeper understanding of the applicability of agile approaches in different embedded industrial domains. While appealing, little or no experience had been collected in the majority of the embedded software domains. It was soon discovered that, even if companies considered in the study were using agile approaches to develop their products, the organisation and the planning of the work is deeply affected by the approaches used in plan-based companies (Sillitti and Succì 2006) [6].

The early trials proved many of the agile practices are applicable but need strong adaptation depending on certification issues (Wils et al. 2006) [7], deep hardware dependency (Wils et al. 2006) [8] and company culture (Still 2006) [9]. It was discovered that software standards – such as CMMI – can be addressed using agile methods (Kähkönen and Abrahamsson 2004) [10] but that agile metrics were mostly missing (Sillitti A. et al. 2004) [11]. A key enabler for agile development was found to lie in proper tooling for the development. It is evident that current software development and management tooling is not optimised for agile production. Even well-known project-management tools are likely to fall short. The two-to-six-week production cycle, product backlog management and automation pose requirements that current tools are not handling very well. There is a significant opportunity for tool developers to fill this gap.

NEW AGILE METHODS INTRODUCED DEAL WITH PROCESS, DOCUMENTATION AND IMPROVEMENT ISSUES

To meet the gap identified in current embedded software development practice and existing solutions, research in AGILE developed four new methods, 12 software tools and a set of industry-relevant guidelines for the facilitation of agile software development in embedded settings.

The four methods developed are:

1. Mobile-D™ (Abrahamsson et al. 2004) [12], which has been tested in 16 mobile software development projects and optimised for a team of fewer than 20 people. It is described in detail in a pattern format and is fully downloadable from the house-of-agile (www.houseofagile.org);
2. RaPiD7 (Kylmäkoski, 2006) [13], which deals with authoring documents in a workshop format and is gaining wide acceptance. Its aim is to improve communication

3. Gartner Dataquest, January 2004

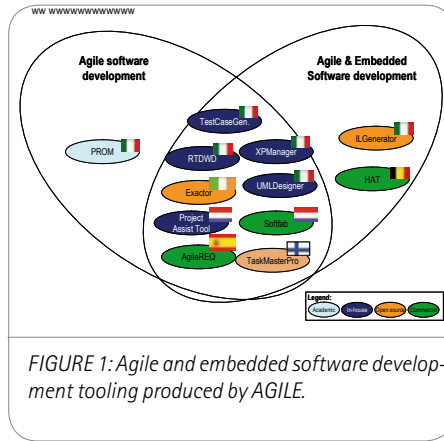


FIGURE 1: Agile and embedded software development tooling produced by AGILE.

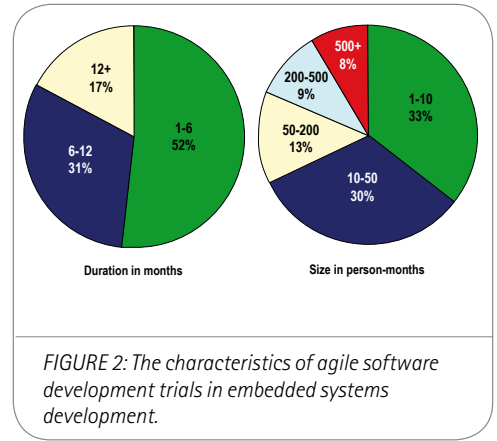


FIGURE 2: The characteristics of agile software development trials in embedded systems development.

by authoring all important documentation in a set of pre-defined, prepared and planned workshops. This will reduce calendar time and misunderstandings. The need for an extra inspection of a particular document is reduced;

3. The post-iteration improvement approach (Salo and Abrahamsson 2006) [14], which enables agile teams to improve their performance in a systematic fashion; and, finally,
4. The EDDY-process model for the mobile telecommunications industry (Kähkönen 2005) [15].

The extensive amount of lessons learnt is built in to the house-of-agile but also in scientific works such as of Kylmäkoski (2006) [13] and Salo (2006) [16]. In her work, Salo points out that a good method can receive as many as 200 negative evaluations but also about 400 positive views. Teams can implement improvement actions at their own discretion, which is a radical improvement in comparison with the existing paradigm of process improvement.

FIT-FOR-PURPOSE TOOLS PLAY AN IMPORTANT ROLE

Tools that foster agile capabilities, such as closer interaction between software- and hardware-centred development teams at a much earlier stage of the design process, are gaining market acceptance according to a Gartner report³. The report predicts these so-called 'electronic system level' (ESL) tools are the next big change for the design-tools-automation market. Gartner sees ESL tools having a market of up to \$500 million by 2008.

The AGILE project produced 12 tools to support agile software development of embedded systems. Four of these tackle problems related to project management. Two of the tools developed will enter commercial markets in about 12 months:

1. HAT, developed by E2S in collaboration with KU Leuven and Barco Avionics, supports model-driven development in an agile environment. Modelling an application

instead of coding brings developers and customers to a higher abstraction level. This makes it easier to make changes, investigate alternatives, derive variants of an existing model and visualise changes. The actual implementation is then performed through model transformations and generators. The behaviour of the unified modelling language (UML) editor is driven by a profile containing a domain-specific embedded context. The system is in trial use with very promising results.

2. Softfab, developed by Philips, is a fully automated test management and build tool that provides a very intuitive and efficient interface for all the test script execution. The tool can be operated via a web-based user interface and is not tied to any particular test suite. Softfab produces standardised reports and is easy to set up. The Softfab produces all status reports needed by a single automated action. The tool was tested in several companies in the AGILE project. Based on the very encouraging empirical results, a commercial version of the tool is being developed.

VALIDATION OF DEVELOPED SOLUTIONS

73 % of agile and embedded industrial trials are considered successful. AGILE studied the use of agile methods in embedded software development in 68 industrial trials involving more than 1,800 engineers in 17 European companies over a 2.5-year period. Figure 2 shows the characteristics of the industrial trials.

The empirical body of evidence in attempts to combine agile and embedded software development is significant. A slight majority of the trials lasted less than six months; 17 % of the trials lasted over a year. About one third consumed less than ten person-months of effort; another one third consumed between ten and 50 person-months. Evidence was also collected in the use of agile solutions in mega-sized projects – i.e. 8 % of trial evidence comes from projects of more than

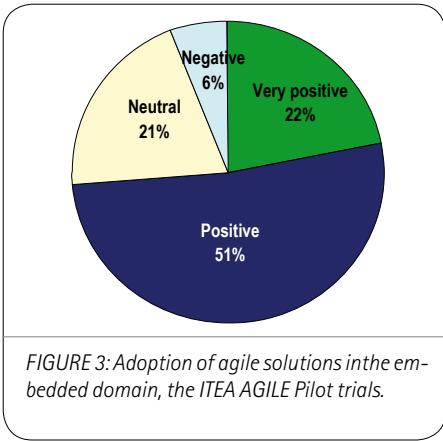


FIGURE 3: Adoption of agile solutions in the embedded domain, the ITEA AGILE Pilot trials.

500 person-months. The empirical findings are even more significant as 73 % of trial findings resulted either in a positive or a very positive outcome (see Figure 3).

The technical environments ranged from pure Java to mobile Symbian operating platform specific languages and to SDL/C telecommunications software programming tools environments. Trial targets were in line with newly developed agile methods, tools and guidelines, and involved a set of specific agile practices such as continuous integration and test-driven development, full blown methods such as Scrum or Mobile-D™ and the development tools.

BUSINESS OPPORTUNITIES

Several business opportunities have been identified: Empirically-proven cost savings, high employee satisfaction coupled with total cost savings over the product life cycle, the tool development and wider agile adoption in the companies. Each of these business opportunities are introduced in the following subsections.

SIGNIFICANT COST SAVINGS POSSIBLE

A great deal of the effort has been out into analysing the impact of agile development on embedded software R&D. The pilot results have been very positive. The Philips' Modena pilot project worked on a 500KLOC digital rights management (DRM) system and the benchmark results show that a 17-person team developed software 8 times faster and 3.5 times better in terms of defects injected than an industry average. The customer satisfaction in the Philips' pilot was 4.9 on a 5-point scale. The Modena team used individual agile practices in combination. F-Secure – an anti-virus company from Finland – achieved 70 % reduction in lead time and costs in developing software for a mobile firewall service. F-Secure used Mobile-D™ from VTT as its development method in the trial.

Would you go back to the old way of working?

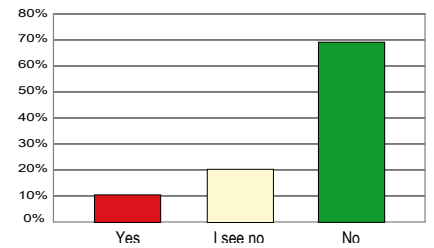


FIGURE 4: Satisfaction in agile and iterative processes in Nokia networks.

Experience shows that a challenge in improving software processes often lies in the reluctance of the development teams to use the proposed solutions. Indeed, if concrete gains are sought, developers are required to find new processes, methods and tools that are efficient and fit for the purpose. If this is not achieved, the methods will not remain in use but will be shelved. In Nokia Networks, a centralised support programme was established with only a few people to support the deployment of agile methods in practice. Product programmes were offered the possibility to try out agile and iterative processes at their own discretion – in other

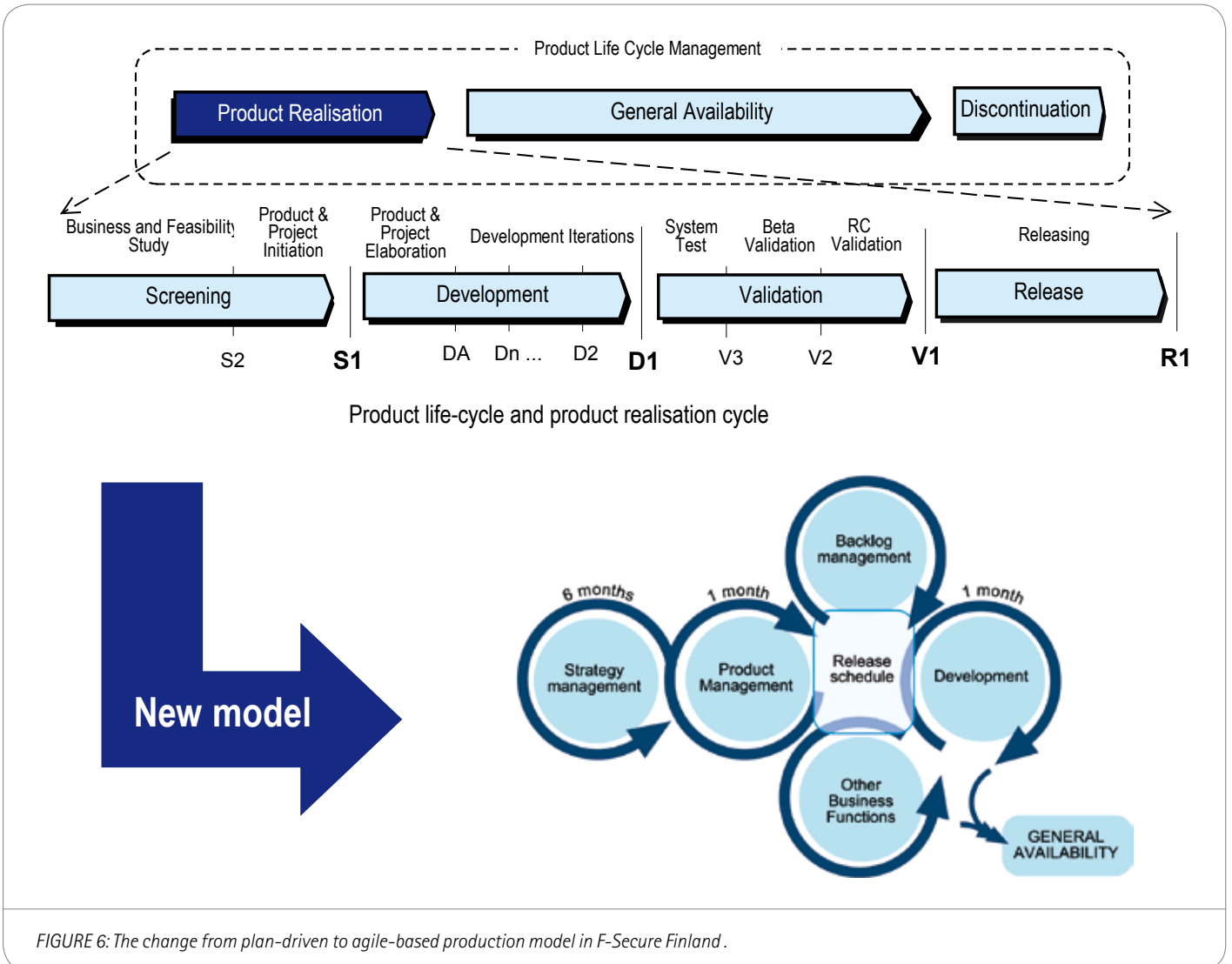


FIGURE 6: The change from plan-driven to agile-based production model in F-Secure Finland.

words, no process push was enforced. As a result, two out of three Nokia Network business units are currently employing agile methods as their primary method to develop software. One reason for success has been widespread acceptance by developers of the new methods. Over 70 % of the developers perceive agile methods as either useful or very useful. Similarly, about 70 % of the developers are reluctant to return to their old ways of working (Figure 4). The most commonly adopted agile method in Nokia Networks has been Scrum, which provides a strict time-limited project-management framework for software development.

Use of agile methods in hardware-bound software development is not straightforward and may require a great deal of customisation to be successful. Engisud (Italy) operating in the industrial automation systems business sector devised a fully operational agile control software development (AGICOSD) methodology with tooling support to enable agile development of manufacturing systems. Figure 5 shows data obtained from five Engisud trials. This shows that change and reconfiguration costs, as well as total change costs, are significantly lower over time using AGICOSD.

WIDER AGILE ADOPTION

A change from traditional development to an agile development model in R&D is likely to have an impact beyond the immediate software R&D setting. There are several reasons for this. Agile development teams operate in time-boxed two-to-six-week development cycles. Product requirements are prioritised and reprioritised based on strategy and market needs, and are not frozen until the latest possible stage of development. This is very challenging both technically and operationally. All other business functions from business management to technical writers are influenced by the time-limited development. One solution to this is to restructure the whole organisation or major parts of it to meet the needs of agile development. Based on the positive agile pilot results, F-Secure launched a company-wide reorganisation of its development and business operations. As a result, a new operational model was introduced (Figure 6), where a decision to divert from milestone-based development to a continuous development mode was initiated.

Wider agile adoption is also demonstrated in other companies such as Barco, Engisud, Philips and British Telecom. An important avenue of influence has been upcoming standards, namely IEEE 1648 and DO-178C.

THE FUTURE: EMBEDDED AGILE INSTITUTE TO FOSTER AGILE PROCESSES ADOPTION

AGILE project members are seeking an opportunity to establish an Embedded Agile Institute in Europe to foster adoption of agile processes in European software-intensive companies. As a service, AGILE members are launching an interactive House-of-Agile web portal (<http://www.houseofagile.org>) to disseminate their results beyond the consortium. It is a Wiki-based solution that enables the community to update the contents.

REFERENCES

- [1] Jones, C. (1999). The economics of software process improvements. Elements of software process assessment & improvement. K. E. Emam and N. H. Madhavji. Los Alamitos, CA, IEEE Computer Society: 133-149.
- [2] Van Solingen, R. (2004). 'Measuring the ROI of Software Process Improvement.' IEEE Software 21(3): 32-38.
- [3] Abrahamsson, P., O. Salo, J. Ronkainen and J. Warsta (2002). Agile software development methods: Review and Analysis. Espoo, Finland, Technical Research Centre of Finland, VTT Publications 478, <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>.
- [4] Salo, O. and P. Abrahamsson (2007). A Survey of the Use of Extreme Programming and Scrum in European Embedded Software Development Organizations. submitted.
- [5] Schwaber, K. and M. Beedle (2002). Agile Software Development With Scrum. Upper Saddle River, NJ, Prentice-Hall.
- [6] Sillitti, A. and G. Succi (2006). "The Role of Plan-Based Approaches in Organizing Agile Companies." Cutter IT journal 19(2): 14-19.
- [7] Wils, A., T. Holvoet and K. De Vlamincq (2006). Timing driven architectural adaptation. International Conference on Distributed Applications and Interoperable Systems (DAIS) 2006, Bologna, Italy.
- [8] Wils, A., S. Van Baelen, T. Holvoet and K. De Vlamincq (2006). Agility in the Avionics Software World. XP2006, Oulu, Finland, Springer.
- [9] Still, J. (2006). "First company-wide agile R&D process established." AGILE-ITEA Newsletter 2(1): 6-7.
- [10] Kähkönen, T. (2005). Agile Methods for Large Organizations – Building Communities of Practice. Agile Development Conference, Salt Lake City, Utah, USA.
- [11] Sillitti A., Russo B., Zuliani P. and Succi G. (2004). Deploying, Updating, and Managing Tools for Collecting Software Metrics. QUantitative TEchniques for SoftWare Agile Processes (QUTE-SWAP) at ACM SIGSOFT2004/FSE-12, Newport Beach, CA, USA.
- [12] Abrahamsson, P., A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen and O. Salo (2004). Mobile-D: An Agile Approach for Mobile Application Development. OOPSLA 2004, Vancouver, Canada.
- [13] Kylmäkoski, R. (2006). RaPID7: A Collaborative Method for the Planning Activities in Software Engineering - Industrial Experiments. Software Engineering. Tampere, Tampere University Of Technology.
- [14] Salo, O. and P. Abrahamsson (2006). "An Iterative Improvement Process for Agile Software Development, Software Process Improvement and Practice." Software Process - Improvement and Practice 12(1): 81-100.
- [15] Kähkönen, T. and P. Abrahamsson (2004). Achieving CMMI Level 2 with Enhanced eXtreme Programming Approach. submitted to a conference.
- [16] Salo, O. (2006). Enabling Software Process Improvement in Agile Software Development Teams and Organisations. Department of Information Processing Science. Oulu, Finland, University of Oulu.

New: Agile in
embedded training
www.houseofagile.org



Brian Hanly
Exoftware



Alberto Sillitti
U. Bozen-Bolzano



Santiago Estela
SQS



Jari Still
F-Secure



Ko Dooms
Philips

Playing With the Big Guys

By Boyan Angelov
Nemetschek Ltd

Imagine you are some SME company with good background producing software in different fields, but you have the need to make some research and don't have the necessary resources in terms of knowledge and funding. Then you will be in a similar situation as Nemetschek Ltd was in the beginning of 2003. Fortunately, there are things you can do.

Since the dawn of its creation, the European Union has established a lot of programs for funding of research work. This has been identified as a key challenge for turning Europe into the "most dynamic competitive knowledge-based economy in the world". Nowadays everyone has heard of the most famous Framework programmes (FP7 being the current one), providing instruments for financing of projects in almost all scientific areas with budget of over 50 billion EUR for seven year period. But have you ever tried to prepare a successful project that will actually get funded? Our company, Nemetschek Ltd, has participated into lots of such efforts, some successful, many not. The competition, especially since FP5, is fierce, getting accepted into a project consortium is not easy, managing the project is tough and the rules are restrictive.

Luckily, if you are some SME company that wants to participate into research specifically in the field of software intensive systems, you have other option too – the EUREKA cluster programme called ITEA (Information Technology for European Advancement). Although writing a good ITEA project proposal is not easier than in FP for example, the chances of getting this proposal accepted are generally higher. That is a combination of several factors:

erally higher. That is a combination of several factors:

- Partners funding – probably the biggest difference to note. There is no fixed amount of money that can be distributed among projects, so if you have a good proposal, it will not be cut out because "the budget limit is reached". Instead, the budget of the project is just a frame, and each partner has to apply for funding from his country Public Authority. This means that some partners will get funded easier and some might not get funded at all. So here comes another difference – partners flexibility. Partners can leave ITEA project and partners can join virtually at any time during the project lifecycle as long as this is approved by ITEA and does not compromise the consortium quality and project chances of success.
- Quality of proposals – the project proposals are usually initiated by some big R&D and Industry players, which already have solid knowledge about the project area and ITEA rules and how a successful project can be carried out;
- Less competition – with the projects being between 50 and 300 man years in terms of effort, the number of proposals competing is much smaller;
- Two-step procedure – each ITEA call involves a two step procedure for evaluation of proposals, so there is a chance for making the proposal stronger between the steps and increasing the chances to win;

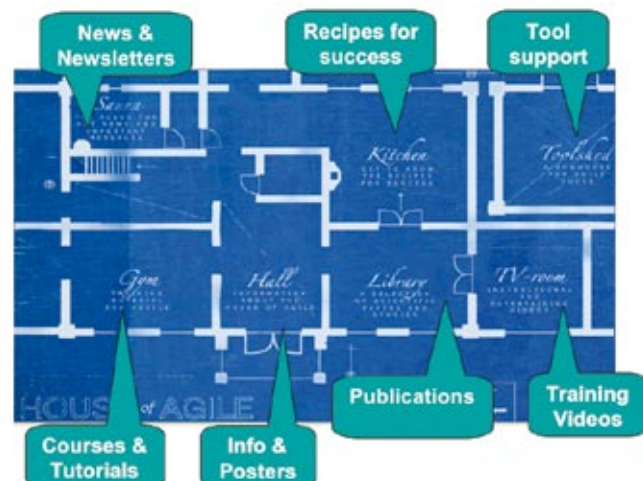
We at Nemetschek were lucky to get invited into the ITEA project "AGILE" by one of the R&D partners, who knew us from previous projects. We joined the consortium since the Project outline preparation and have stayed with the project until its end in January 2007. It has been a great challenge, because the funding rules for such projects in Bulgaria have not been established and it took 2 years of effort to establish them together with the Public authority and finally get the funding in 2005. The ITEA projects flexibility and the partners' quality allowed us to stay with the consortium and finally be able to contribute to the work. The chance we had to work together with the Big Guys – VTT, Philips, Nokia, and the rest of the partners is a real benefit from such projects and is worth all the effort put into winning the proposal, winning of the funding, all the work done and the money spent. So being into an ITEA project for us means an opportunity to do research together with some big names in Europe and even get some funding for it. Using such opportunities is what can make the difference for SME companies on the global competitive market today.

http://en.wikipedia.org/wiki/Lisbon_Strategy
http://cordis.europa.eu/fp7/home_en.html
<http://www.itea-office.org/>

<http://www.agile-itea.org/>

House of Agile

A community dedicated for agile embedded knowledge, training and services



ATO: a Tool Environment supporting Agile Model-Driven Development

The Agile Alliance Manifesto states that working software should be valued over comprehensive documentation. This has led to the neglect of modelling activities and software models in many Agile software development approaches, since they consider software models to be part of the development documentation.

Agile Model-Driven Development (MDD) tries to apply the Agile values on software modelling activities. When code could be (semi-)automatically generated from software models, the software models can be considered as working software themselves. Such approach achieves a leverage of the implementation platform from programming languages to modelling languages.

TOOL SUPPORT FOR AGILE MDD

An adequate tool environment to support code generation is primordial for a successful application of Agile MDD. Existing tools are often restricted to predefined code generation schemes that cannot easily be extended or modified. On the other hand, more advanced tools offer code generation possibilities using non-trivial scripting languages that have a huge learning curve.

The ATO (Analysis and Transformation of Objects) tool environment developed by E2S support the realisation of an Agile MDD approach by offering a UML environment for the specification of models, the transformation of models, and the generation of code, documents, and tests. Its key characteristic is that it uses UML's own OCL (Object Constraint Language) in order to define the transformation and generation rules. One can navigate through the whole model and the OCL expressions that are contained within it, using the same OCL language for navigation and selection of the elements, and for defining the transformation and generation rules that have to be applied. The possibility to define transformations and generations external to the tool is crucial for Avionics applications, as it simplifies the certification of the generated software.

Model transformations, which create a tree structure of dependent models, can be used

to simplify the code generator by performing preparatory transformation actions prior to the actual code generation.

ATO incorporates an OCL debugger, a transformation debugger and a code generation debugger.

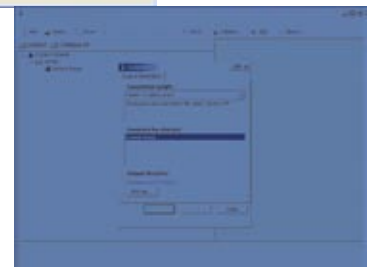
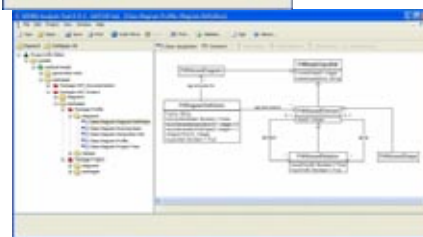
The transformation and generation rules are stored into an ATO model profile that can be reused on other models. The profile can define a domain specific context, consisting of

- A selection of the different model elements and diagrams that may be used within the domain.
- The graphical representation of the model elements and stereotypes.
- Definition of constraints on the model that apply for the whole domain.
- Transformers and generators to produce models, tests, documents and code from a given model. Currently, ATO support the transformation and generation starting from class diagrams (including full support for run-time OCL constraints), state machine diagrams, sequence diagrams, and package diagrams.

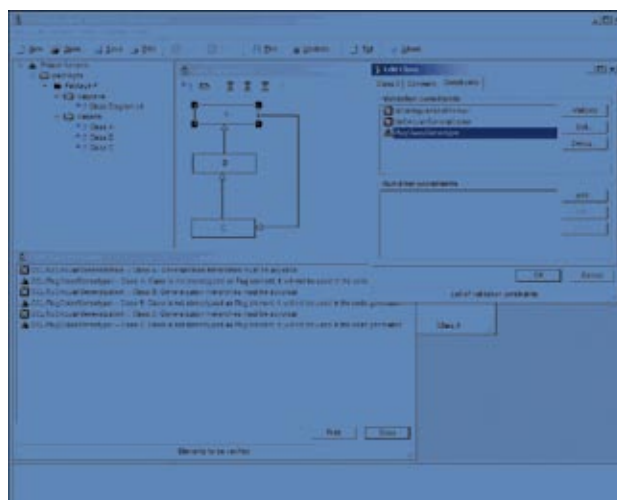
APPLICATION VS. PROFILE DEVELOPERS

The definition of a profile becomes the responsibility of the domain specialist, who defines the development process and the quality of the development.

The application developer then starts UML modelling with a selected domain profile, which now drives the behaviour of the UML modeling tool. Its validation function makes sure the model conforms to the domain constraints. After completing a model, the application developer then selects the required transformation(s) and code gener-



ator to produce the code corresponding with the model and its constraints.



ATO has been used within the AGILE project to create several profiles, containing generators to Java, Ada and Delphi Pascal.

For more information, contact:
Stefan Van Baelen
Stefan.VanBaelen@cs.kuleuven.be
<http://www.cs.kuleuven.be/~stefanv>

Michel Huybrechts
mhu@e2s.be, <http://www.e2s.be>

Stijn Rammeloo
Stijn.Rammeloo@barco.com
<http://www.barco.com>

AgileREQ 3.0 by SQS, S.A.

AgileREQ is the solution developed by SQS, S.A. to help with the requirements management in Agile development projects. Agile methodologies are specially appropriated for projects and environments where changes are likely to happen and AgileREQ has been designed to make its implementation easier and to increase its benefits.

The Agile development methodologies have widely proved their efficiency to avoid problems in the requirements' management. Their strategy, based on short iterations, provide a great opportunity to access promptly and easily to the most important information. AgileREQ takes advantage of this to provide a certain amount of metrics, which helps with the decision-making related to the requirements' management.

AGILEREQ EVOLUTION

AgileREQ V1 was based on SQS' experience in the management of Agile projects and was introduced both in business and academic forums; the feed-

back received led to the evolution of the tool to the current AgileREQ 3.0. This new version includes an improved data structure, which makes the data export and later processing easier, a more complete set of reports and graphics, which are more simply generated and more visual now, and new metrics related to errors, which are collected and processed in full detail.

FUNCTIONALITY

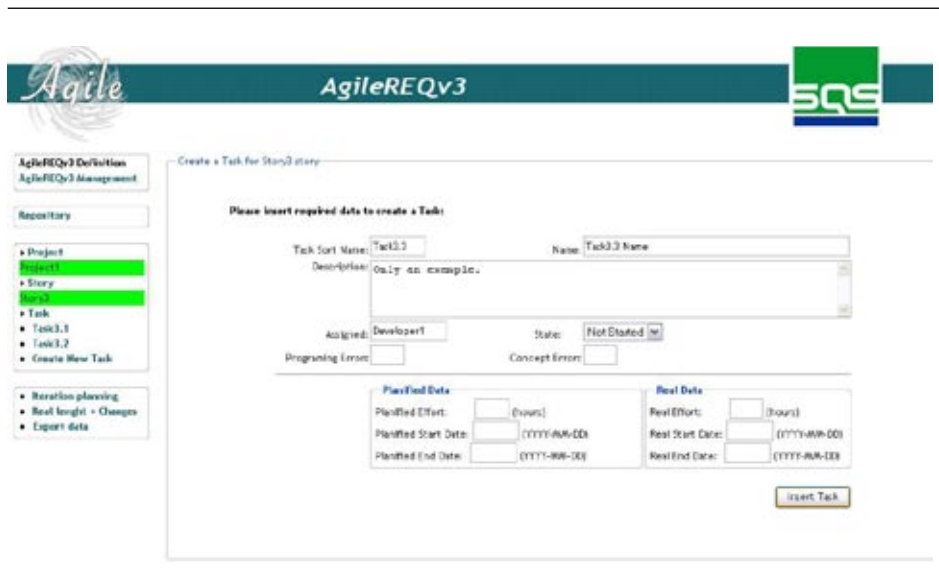
The main function of AgileREQ is the gathering of requirements which are collected as User Stories, prioritized by the customer in order to configure every iteration, and later divided into task by the developers. Once the scope of the present iteration has been defined, AgileREQ makes it easier the monitoring of the state of User Stories and Tasks. The system gathers information and generates a diversity of metrics that allow the efficient management of software development projects. AgileREQ also provides the possibility to modify requirements, efforts and almost every detail at any

moment, recording the different versions in the project's historical.

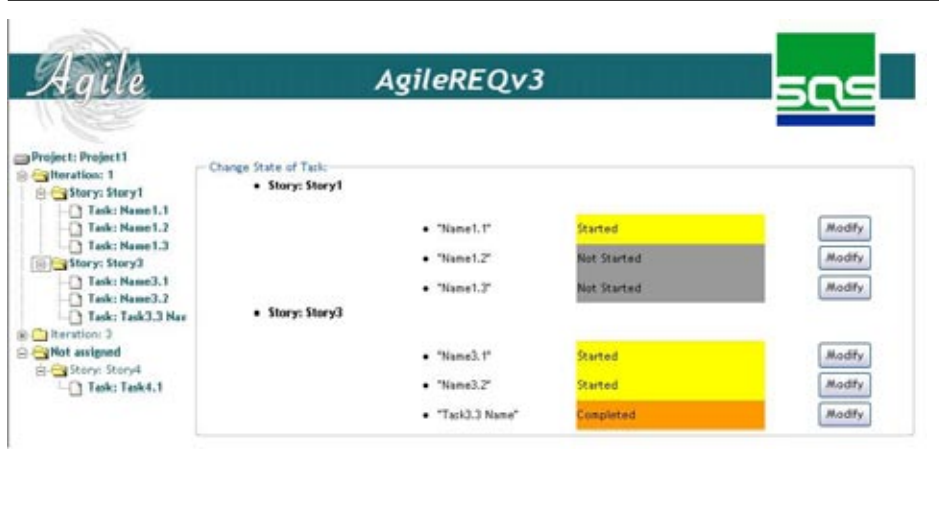
Finally, the tool allows the creation of a wide variety of reports, such as User Stories and Tasks reports as well as errors and efforts reports. The tool also allows exporting to Excel the data related to a specific project.

Having developed AgileREQ as a Web tool, makes this tool suitable for being used by the actors involved in a development process.

To sum up, AgileREQ 3.0 is an agile tool appropriate to monitor tasks and control the work flow and the information generated in a process that maintains the agility and handling easiness of the previous versions but strengthening its functionality and features. Represents, consequently, an inestimable help in the agile development processes, leading in its execution, and making easier the gathering and generation of metrics, so necessary to take the right decisions when leading a project.



Santiago Estela
SQS



Development of an effort prediction model for an agile development project at Engisud

The objective of the study was to develop a model for effort prediction in an agile, industrial development environment setting. The Free University of Bolzano and Engisud propose a new methodology for data collection and effort prediction designed to support iterative development and based on the automated collection of data. Such data are collected in a non-invasive way and can be used to build effort prediction models in an incremental way. The IDEFO overview of the methodology is shown in Figure 1. For such models, the development effort is the dependent variable or network output and design/code metrics are the predictor/independent variables.

CASE STUDY AT ENGISUD

We collected data for a project that aims at creating an Agile project management tool written in Java (Extreme Project manager). The data collection period used for model building extends from June 16 to September 28 2005. The total time spent in the development activity collected by the automated tools is 104.7 hours. Most of the development has been carried out by one developer who

worked on 45 Java classes with a total of 18290 lines of code (LOC). Table 1 gives a descriptive statistics of the collected data. As predictor metrics we use Weighted Methods per Class (WMC), Lack of Cohesion Of Methods (LCOM), Depth of Inheritance Tree (DIT) design metrics of the well-known Chidamber and Kemerer set of design metrics and the Number Of Attributes of a class (NOA) as additional size-related metric. The dependent variable is coding effort inside the IDE (Eclipse3.0) measured in hours.

Several prediction models have been inspected to solve the problem. The method that best fitted is based on Neural Networks.

Software engineers and researchers often argue against the use of Neural Networks for modeling development effort as such models are seen as black boxes and do not reveal the underlying relationship for example between effort and design metrics. However, in this specific application a partial interpretation of the data is possible and it is under investigation.

The main result of this experimentation is that it seems possible to create a model for the prediction of the effort for an iteration given the effort of the previous ones and some code metrics. The reliability of the prediction seems to increase as the time goes by and more data are available.

Once integrated in a development tool, such results may help agile teams to improve their estimates and control better their development process.

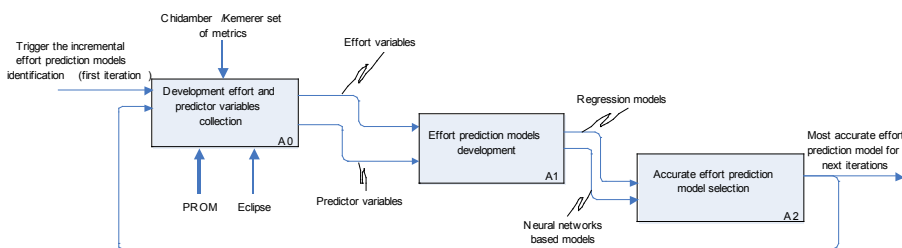


Figure 1: IDEFO diagram of the effort prediction models selection methodology.

Metric	WMC	LCOM	DIT	NOA	Effort (h)
Mean	9.2	0.5	2.8	17.4	2.3
Std	±9.6	±0.4	±1.0	±20.9	±4.3

Table 1: Descriptive statistics of data at a class level.

Authors:



Giovanni Aiello
(giovanni.aiello@eng.it)



Marco Alessi
(marco.alessi@eng.it)



Alberto Sillitti
(Alberto.Sillitti@unibz.it)



Giancarlo Succi
(Giancarlo.Succi@unibz.it)

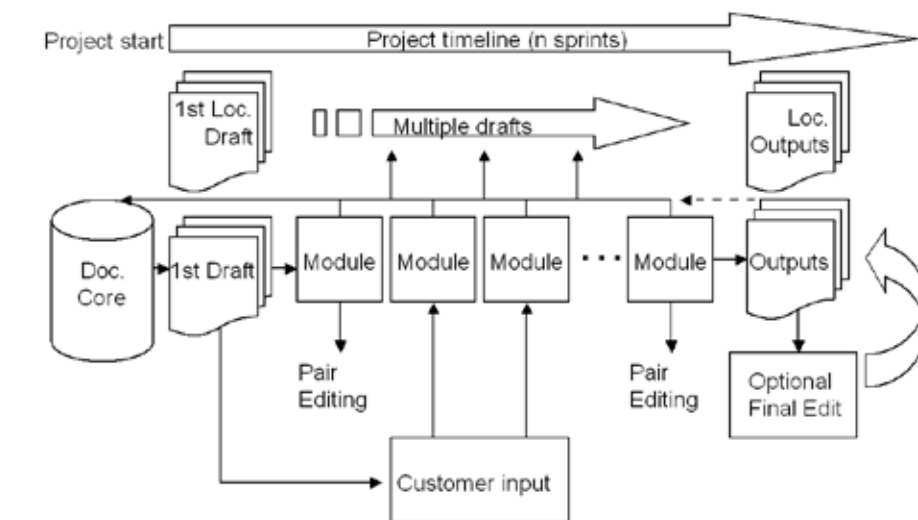
Enabling Agile Documentation

After our first foray into the agile world with localization, we decided to tackle the other traditional artisan-type component in software products, documentation. Surprisingly, a lot of the process and philosophy outlined in our previous article (ITEA Agile Newsletter 2/2006) applied to the documentation process as well. However, making technical writing agile requires us to reevaluate what the key competency in both writing and translating software documentation is, so that we can then by way of automation, and elimination highlight the core of the writing task to be done iteratively by writers (**bolded in the text**).

Let's take a look at a standard tech writing waterfall workflow.

1. Reading the specs
2. Project refamiliarization
 - a. Fixing open bugs/defects
 - b. Defining change scope
3. **Development team interaction**
 - a. **Documenting new features**
 - b. **Facilitating the validation**
4. Screenshots
5. DTP
6. Building Online Help
7. **Testing Online Help**
8. **Fixing online help**
9. Rebuilding
10. Translation
 - a. Instructing the translator
 - b. Translators' kit
 - c. Editing 100% matches
 - d. Editing fuzzy matches
 - e. **Translation**
 - f. **Validation**
11. Localized screenshots
12. Translation DTP
13. Building translated online help
14. **Testing translated Online Help**
15. **Translation bug fixing**
16. Rebuilding translated versions

Step 1. Since we do not rely on specs in an iterative process, the first step can most often be eliminated.



Step 2 in an agile environment is usually very brief, since the product documentation should also improve constantly on a sprint basis, which eliminates a build-up of bugs for 2a.

In step 3, we find one of the competencies where the writer really adds value to the product. Human interaction and creating new material are what we should use people for, hence it is bolded.

Step 4 and 11 is a lot of work, admittedly it can help the writer to familiarize with a new product, but from an iterative project standpoint, as a repetitive task it should be automated, especially once you support more than just the language in which you write in.

Step 5 and 12. Desktop publishing is time consuming. Using XML-based documentation allows you to separate the majority of the formatting and layout tasks to a style sheet.

Steps 6, 9, 13 and 16. Those who have worked with MS Word based Online Help tools know that generating the help itself is extremely time consuming and error prone since it relies heavily on font and style information. This should be automated, especially in an agile environment where generating English and the localized help files should be a per sprint effort.

Steps 7 and 8. Testing and bug fixing is one area again where the human interaction is crucial. We

can automate checkers for some help functionality, but the human element is priceless.

Steps 10, 14 and 15. The key element is the artisanship of the translator, not the file handling or computer aided translation maintenance, which can be done automatically when needed. By using XML-based source material and tools that clearly guide the translator, we can allow the translator to the majority of their time to generate added value by translating new material.



Mika Pehkonen
Documentation and Localization Manager
F-Secure Corporation
mika.pehkonen@f-secure.com



AGILE

FACTS:

AGILE - Agile development of embedded systems

21 partners, 8 countries, 171 person years

1.4.2004 - 31.12.2006

<http://www.agile-itea.org>

Project coordinator: Pekka Abrahamsson, VTT

(pekka.abrahamsson@vtt.fi)