ITEA
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

Agile

INNOVATION REPORT

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 changes 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 €10,000 to €45,000 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 fail[1]. 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

1. *http://www.sei.cmu.edu/sema/*

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 2005[2] 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 gaps

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 agile 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 Succi 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.

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*

INNOVATION REPORT

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 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 report[3]. 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 Softab 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.

3. Gartner Dataquest, January 2004

INNOVATION REPORT

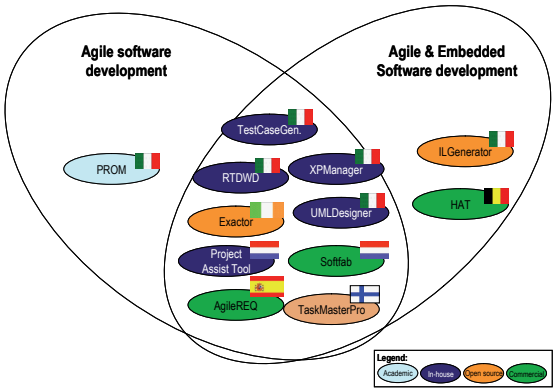

Figure 1: *Agile and embedded software development tooling produced by AGILE*

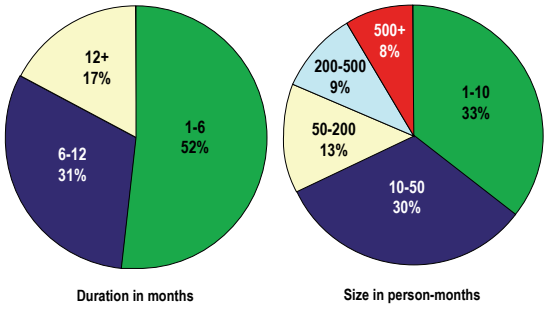

Figure 2: *The characteristics of agile software development trials in embedded systems development*

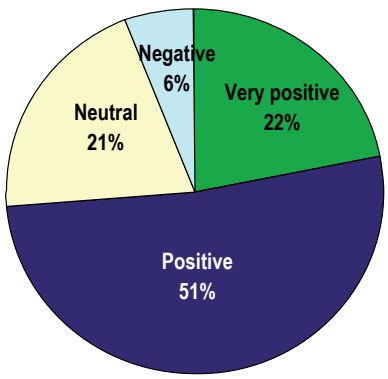

Figure 3: *Adoption of agile solutions in the embedded domain, the ITEA AGILE Pilot trials*

Figure 1 shows the tools that have been developed to facilitate the use of agile methods in embedded software development settings.

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 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 environ-ments. 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.

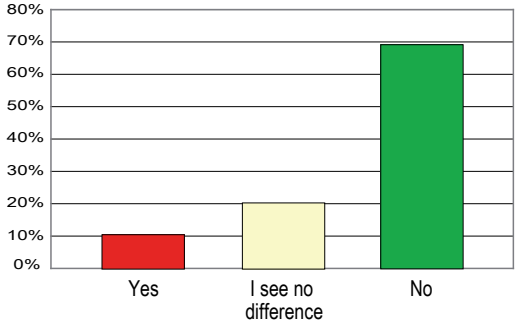

Figure 4*: Satisfaction in agile and iterative processes in Nokia networks*

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.

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 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.

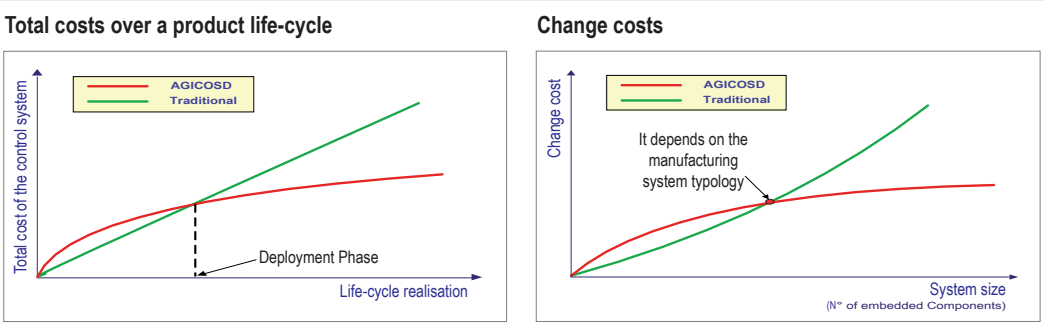

Figure 5*: Impact of agile methods in product development costs at Engisud, Italy*

INNOVATION REPORT

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.

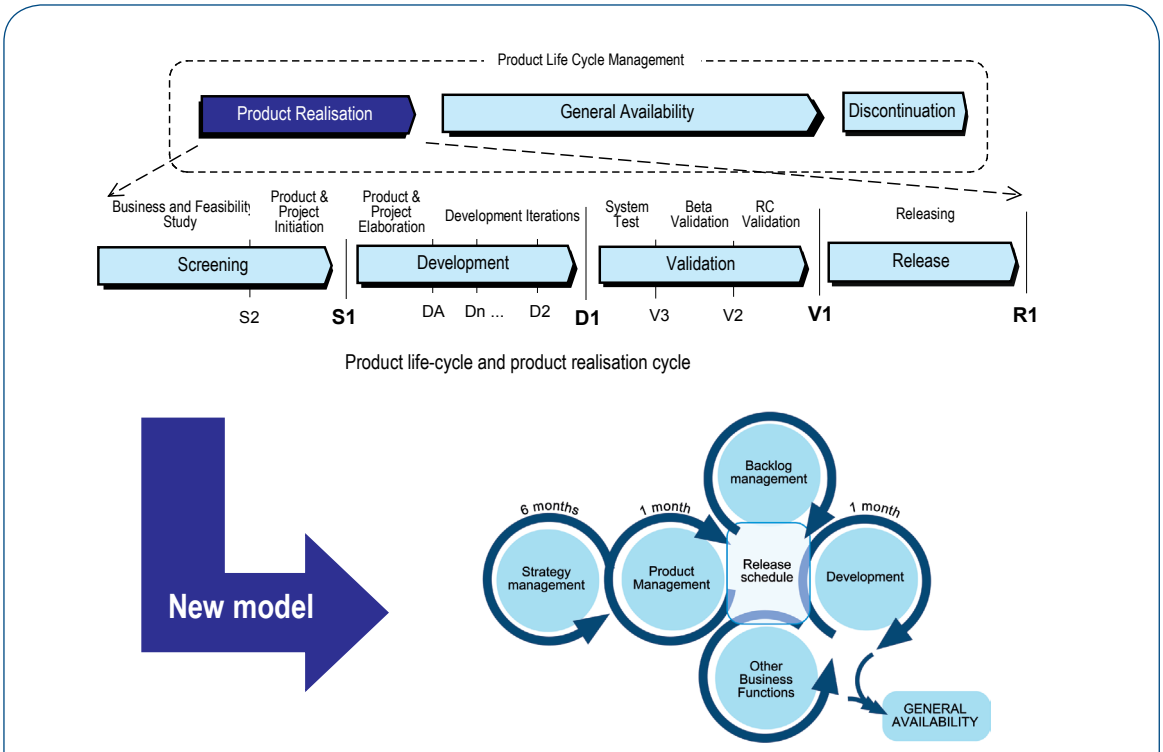

Figure 6*: The change from plan-driven to agile-based production model in F-Secure Finland*

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 Vlaminck (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 Vlaminck (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.