

What is EVISOFT?

Evidence based Improvement of SOFTWARE engineering - EVISOFT - is a project that runs from 2006 through 2010 and is partially funded by the Research Council of Norway. Three research institutions, SINTEF, the Norwegian University of Science and Technology and the Simula Research Laboratory together with 10 large and medium sized Norwegian ICT companies cooperate on how to find the best methods and technologies to provide software with the right quality, within a given timeframe and at the right price.

EVISOFT focuses on:

- agile and evolutionary methods
- model based development
- estimation and risk management
- component based development

The target Audience for

EVISOFT is small and medium-sized software-intensive companies, research and development departments in large companies, universities and colleges and consulting companies that develop and offer SPI services.

Deliverables from EVISOFT

will be: scientific and popular science articles, experience reports, a web-page and this newsletter.

From the editor

Tor Stålhane, NTNU

We're back! Maybe not with a vengeance, but at least with a new issue of Improve. We start with an interview with Hakan Erdogmus with the title "Agile development - trends and future" by Geir Hanssen. Next, we have an article about the open space session on agile topics arranged during the CAISE conference in Trondheim written by yours truly and accompanied by a cute photo of some of the enthusiastic participants. Last but obviously not least, we present an article on evolutionary product development by Tor Fægri and Geir Hanssen.

The super 14 has come and gone and so has the Bledisloe Cup. Despite their well-deserved win at the Auzzi Stadium, the Wallabies lost at Eden Park and the cup stays in New Zealand. Well, well - but next year.... just you wait and see! The Rugby World Cup, taking place in the UK and France, is drawing near. I have tickets for the match between the All Blacks and Scotland in Edinburgh. See some of you there?

Hakan Erdogmus on agile development, trends and future

Improve met Hakan Erdogmus at last years ISESE conference in Rio de Janeiro. He is the new Editor in Chief of IEEE Software. Hakan has paid a lot of interest in the development of agile methods.

By Geir Hanssen, SINTEF

Improve: As a start - a bit about you: where do you work right now and what is your position?

Hakan: I work for the National Research Council Canada. It's the main research body of the Federal Government and our mandate is applied research and technology transfer. At NRC, I work for the Institute for Information Technology as a Senior Research Officer, and I am part of a small research group on software engineering. I also have an adjunct position with the University of Calgary where I teach

Improve: So, what are your current research interests?

Hakan: They follow three threads that are inter-related in many ways. The first is software economics, the second thread agile methods. The third, that I am focusing more and more on nowadays, is in-process measurement. But the context I favour for issues like software economics and in-process measurement related to agile methods - that's where my passion is.

Improve: Why are you motivated to work on these topics?

Hakan: I was introduced to agile methods around 2000 by Laurie Williams (a interview with Laurie Williams was presented in Improve 1-2007, ed. note). She was concluding her PhD thesis on analyzing her pair programming study data from an economic perspective, and we collaborated on that analysis. This grabbed my attention. Later she got me in touch with another colleague, John Favaro. John and I did an analysis looking at extreme programming, using financial analysis techniques to try to bring out the value of flexibility embedded in agile practices. And then we got hooked up with Kent Beck and ran a workshop at the first conference on Extreme Programming - the

North American one in Raleigh/North Carolina, which Laurie organized. The exposure changed not only the way I write code, but also how I approach research. My background is in formal methods. So, I always had to specify to death - formally. The agile paradigm basically sobered me up in terms of how I approached things and also liberated me a lot. And when I thought more about it, I realized there was something really important in there.

Improve: How is this a better approach for the industry?

Hakan: There are two reasons for this: The first one is that it's the only process improvement paradigm that I know that has wide grassroots support. I mean, programmers don't go by themselves and try to implement CMMI or similar processes. CMMI is at a different level than XP of course, but even PSP (Personal Software Process) has not been successful in terms of its adoption. But XP was something that I actually sought out and actively pursued as a developer. And I think that's unique. It shows that there is something in there that is compatible with how people want to do their work, and that's important. The second aspect: I never believed in this quest for repeatability and predictability in process improvement. It always sounded like it was a psychological thing - because humans need certainty and stability. So, that quest was driven by human needs rather than business needs. And I think agile methods bring a lot of emphasis on results and end value. And the scope is smaller. So, you work with smaller constraints. Plus I think its promise of delivering results and value is pretty compelling versus the promise of delivering repeatability and predictability, which cannot be end goals by themselves.



Hakan Erdogmus

Improve: How should we move on?

Hakan: There are two things that I think are important and actually present interesting opportunities for researchers, and one of them is test-driven development (TDD). First of all, TDD is one of those things that everybody agrees is a good thing to do, but they all say that they're not doing it. There was a recent survey by Cutter Consortium, - a large survey of Cutter clients looking at practices that impact a project's success. TDD was rated top practice over reviews, inspections and other things, but is the least applied one. And that raises an interesting question - why is it at the same time important and appealing, but difficult to apply? I think there is a need to understand TDD better. And also, it is a really broad practice. It can be applied at multiple levels, using multiple frameworks, in multiple contexts and so on. So, the landscape is so big that you can tackle so many different aspects of it. The other point is that people always think of TDD as a quality practice. My hypothesis is that TDD is predominantly a productivity practice, rather than a quality practice. Of course, if testing is simply optional, any testing practice will tend to be a quality practice and will incur a productivity penalty, because then you will be comparing it with a non-testing practice. I'd like to explore that hypothesis - that TDD is a productivity practice - and to be able to do this, I think we have to understand it better, and to understand it better, we have to measure it better. So, nowadays I'm very much interested in measuring it, in terms of its efficiency, but also in terms of the underlying dynamics.

Improve: In more general - based on your recent research - what advice do you have to the industry today?

Hakan: If they're interested in adopting

agile methods they have to start somewhere, and the most obvious place to start is, of course, incremental and iterative development, for instance with a process like Scrum. That's one. And I think step two is TDD, because it promises the best return on investment - if successful. And it has to be tried early, because it's very difficult to introduce it later. It becomes kind of the weakest link. And if it's not going to be successful, you want to identify it as a showstopper and maybe forget about it.

Improve: The last question, how do you think agile methods will shape or mature in the future? Is the idea complete or will it reshape?

Hakan: It's definitely not complete. In the start you had the small-processes and big-processes, and there was a separation between them. And I think since their introduction, agile methods have been moving towards the middle, and the agile proponents realized that there were some gaps. So, they started to borrow ideas. The big-process people realized that there were some constraints, and they have been trying to make their processes smaller. I think there has been some reconciliation happening but the shift is towards the agile end of the spectrum though, rather than towards the big-process end of the spectrum. People no longer think that these two are completely separate, irreconcilable paradigms. I believe it's going to keep moving in that direction and it's going to go probably a little bit too far towards the big process part and then swing back a little bit more. It's going to become mainstream. I am pretty sure about that. It is already, I think, at the level of crossing the chasm - if it has not already crossed the chasm. And it is possible that the buzzword "agile" will ultimately just disappear but the individual practices or the essence of it will be there.

If you want to read more about Hakan's thoughts about the evolution of agile software development, watch for his column in the November/December 2007 issue of IEEE Software at www.computer.org/software. The editor column is freely available from the website.

Open space and agile development

During the CAISE'07 conference, there was an open space session on agile development. The event had 26 enthusiastic participants from academia, research institutes and industry.

By Tor Stålhane, NTNU

An open space event is easy to organize and run (www.openspaceworld.org/). You start by collecting a group of enthusiastic persons and put up a large sheet of paper or use a large - and I mean *large* - whiteboard. Divide it first into a set of time slot columns of for instance 45 minutes each and then horizontally into rows. Participants can then mark a cell with their favourite topic and the place where the topic will be discussed. Have plenty of working space available or preferably, plenty of small meeting rooms. It is up to the participants to choose which sessions they want to participate in.

The open space process is simple and is guided by four principles and one rule:

- Those who show up are the right persons.
- What happened is the only thing that could have happened.
- Whenever it starts is the right time for starting.
- It's over when it's over.
- When you get tired of a topic go and do something else.

The space available in this excellent but alas, small newsletter does not allow us to go into details of all the 15 fascinating topics that were discussed. Instead, we will bring you a selection of highlights related to SPI - How can retrospectives have a larger effect, How to measure the effect of Scrum and How to improve estimation skills. By the way, a retrospective is just a fancy word for Postmortem analysis (PMA). It seems that the latter term has a rather unscientific ring to it.

How can retrospectives have a larger effect

One of the important topics that were discussed was whether really hard, people-related issues such as problems in internal collaboration were brought up, and if they were not, how could these problems be brought up, discussed and handled? Related to this problem it is also important to know how many of the problems brought up during a retrospective are related to competence and communication. Another point that is definitely worth considering is to relate the problems identified in a retrospective to the personal development plans in the company. By doing this, we could improve each person's skills *and* attack important problems observed in the projects at the same time.

How to measure the effect of SCRUM

One of the nice things with Scrum is that it is quite inexpensive to introduce and use. Thus there is little pressure from management on providing data to show that it pays off. Even so, there are several factors that a project manager or the company management want to keep track of -



Open Space participants at Caise 2007 – Trondheim, Norway

agile or not. Some important factors that were brought up in the open space session were product quality, delivery precision, development productivity and customer and developer satisfaction. To quote Jean-Baptiste Alphonse Karr (1808 – 1880): “The more things change, the more they remain the same”.

The discussion brought forward a set of interesting suggestions for metric opportunities. The most promising were the following: the number of open issues and received problem reports in the backlog, the number of checked-in changes over time and the time used to go from dry-run to final delivery. All these suggestions fulfil the basic requirements for software metrics – they make sense, they are simple to measure and they are useful for developers and management alike.

How to improve estimation skills

Part of the discussions on this theme was related to general estimation problems while the rest was related to agile specific problems. Some important general problems that were discussed were the (1) resolution of the estimates – estimating in person hours, person days or person weeks, (2) that the experts who do the estimates estimate as if the job should be done by themselves and not by somebody else who might have a different skill level and (3) that we too often forget extra-project activities like training and documentation. On the agile side, Pekka Abra-

hamsson presented the latest and greatest in estimation techniques – planning poker (www.planningpoker.com/). It is simple, it is fun and it works – or at least the zealots claim that it does.

The best improvement advice to surface during the discussion was to start by categorizing your task and compare estimates and real resource consumption for each category. In this way you can see which task categories that are overestimated – yes, it *does* happen, which are underestimated and those where the estimates were just right. This will help you to identify where your troubles come from and thus help you to improve your estimation. Unfortunately, in order to learn, you have to make the mistakes first.

At the end of the open space day we ran a Postmortem on the Open Space session. Among the positive points to remember are that it was fun, the participants got lots of new ideas and they gained a better understanding of the other participants’ ideas. Among the negative points raised, we should improve timekeeping and get better at keeping the discussions focused on the assigned theme.

All in all it was a successful event. The weather was nice, the environment was nice, the lunch was nice and I found out why Pekka Abrahamsson had become so interested in Scrum. It turned out that he had been a rugby player in his younger days and although he didn’t say so, he *must* have been a prop.

A case study on evolutionary product development

By Tor E. Fægri and Geir K. Hanssen, SINTEF ICT

Funded by the SPIKE project, we conducted a case study on evolutionary product development using EVO. The full citation is: T.E. Fægri and G.K. Hanssen, “Collaboration, process control and fragility in evolutionary product development”, *IEEE Software*, 2007. **24**(3): pp. 96-104.

In the study we report upon the experiences of a software company in their transition from a plan-driven software process to EVO, an agile evolutionary process. We describe some of the organizational implications for the supplier including customer collaboration, process control and psychological comfort. We introduce the concept “process fragility” in our discussion of critical process attributes and explain why this attribute should be taken into consideration. Our findings are primarily based on qualitative analysis of successive interviews with developers, product management and involved customers. Evolutionary product development improves stakeholder engagement and developer responsiveness over plan-based processes but requires cross-organizational discipline and involved

Puzzle Corner

Let me first clear up a slight misunderstanding. Someone has complained about the solution to the marble problem. He claims that the solution will not work if you open a box marked with red or blue except if you look at all the marbles. He is of course right – I assumed that it was possible to see all the marbles. It would be a bit unfair to let the box with both red and blue marbles contain only red ones, except for a single blue one at the bottom. What we can learn from this is “Don’t loose your marbles”.

The coin problem in last issue’s puzzle received quite a lot of solutions – all of them right. Seems like I have to make them more difficult in the future. To recap: what is the smallest amount you can pay if there are only two types of coins in circulation – one with the value of 33 and the other with a value of 60, when there is an unlimited amount of coins available for both buyer and seller?

Dag Belnes from the Pharos company in Oslo presented the best solution, which runs as follows: a payment will consist of A 33-coins and B 60-coins where A and B are integers. A positive value means that you pay while a negative value means that you receive. Thus, a transaction can be written as $A*33 + B*60$ or $3(A*11 + B*20)$. The smallest value is $N*3$, given by finding the smallest integer $N > 0$ that satisfy the equation $N = (A*11 + B*20)$. A quick search shows that we can get $N = 1$ by choosing $A = 11$ and $B = -6$. Thus, the smallest value that can be paid is 3.

Now, on to the new puzzle. This puzzle contains two parts – one, or rather two, easy ones and one hard on. First the easy part: We start with a string of numbers 87654321. Insert five add operands so that the sum becomes 99. When you have finished this first part of the warm up session, take the same string and insert six add operands to make the sum equal to 99.

Done that? OK – here’s the difficult part: Prove that the two solutions are unique. Have fun.

At last I have a small trick question for all of you. Be careful – the trick might not be where you think it is. The question is as follows: “You have a box of matches, a candle and a kerosene lamp. What should you light first?”

As always, the solution should be sent to stalhane@idi.ntnu.no. A Halvdan and a book await the lucky winner.

management. Our findings may constitute important advice to other product companies considering evolutionary development practices.

Upcoming events

ASWEC 2008, 26-28 March

The 19th Australian Software Engineering Conference in Perth, Australia. Deadline 15 Oct. 2007

ICSE 2008, 10-18 May

The 30th International Conference on Software Engineering in Leipzig, Germany! Deadline: Research Papers and Experience Tracks: 14 Sep .2007

XP 2008, 10-14 June

The 9th International Conference on eXtreme Programming and Agile Processes in Software Engineering, in Limerick, Ireland. Deadline: 6th of Jan. 2008

The Chef Recommends

Tor Stålhane recommends some easily digestible, but highly “notorious” reading for your spare time!

Starter

Are you tired of looking at the usual maps? Then Google maps may be good idea (maps.google.no/). You can see the map, a satellite photo or a combination. The quality varies quite a lot but for instance the Potsdam area, where the EuroSPI 2007 will take place, is excellently covered. By the way, so is Sydney.

Main course

You may have noted an increasingly unison choir praising all that is labeled agile. We never learn do we – some years ago object-orientation should cure all our pains, before that it was structured programming – strictly no GOTOs – and now we have agile methods. I don’t know about you folks, but I tend to get suspicious when the answer to all objections is a Kent Beck quote. Amen.

There are, however, dissenting voices out there, even one that has presented his message at an agile conference – Gerold

Keefer. It was probably received with the same enthusiasm as when the pope received the news of Luther’s 95 theses nailed to the door of the Castel Church of Wittenberg. Here are his main points. Keefer’s that is, not Luther’s:

- The proof of the pudding for XP – the Chrysler Comprehensive Compensation system – C3 for short – started in 1995, went agile in March 1996 and was killed in February 2000 without meeting its requirements. After this, XP was retired as a development method at Daimler-Chrysler. This seems to be common knowledge in the XP community but it is not com-il-faux to talk about it.
- The idea to embrace change looks nice. However, what about co-projects and sub-projects? In addition, a sound system cannot be based on unbound and constant transitions as this will undermine its conceptual integrity.
- What about non-functional requirements such as reliability and security? It is hard to see how to apply the “write the tests first” approach here. As a matter of fact – considerations for non-functional requirements seem to be totally missing from the XP community.

OK. I guess you got the idea. If you haven’t, you should do two things (1) draw the hype curve and indicate where XP is on that curve now. Then try to indicate where it will be next year. (2) Read G. Keefer’s article:

www.agilealliance.org/system/article/file/945/file.pdf

Dessert

After such a heavy main course, you can probably make do with a light dessert. You probably expected more haikus. Unfortunately I cannot deliver. My list of haikus is in Trondheim and I’m on sabbatical at Chalmers Technical University in Gothenburg. So – instead of a haiku I will give you a quote from the local rag:

If a man says to a girl “I dig our handbag” you can be sure that it is just a pick-up line. If he really liked handbags, he would rather talk to Orlando Bloom.

Free Subscription to the the Improve Newsletter

If you want to keep up to date on software process improvement, you can sign up for a free subscription to this quarterly newsletter by registering on www.sintef.no/improve or sending a mail to improve@sintef.no. If you want to stop receiving this free newsletter send a mail to improve@sintef.no

Chief editor: Nils Brede Moe, SINTEF
Supreme chief editor: Tor Stålhane, NTNU
Associate editor: Torgeir Dingsøy, SINTEF/NTNU