

I was invited to speak at the conference "Innovation in mind" over the topic "Lean + Agile = innovation?", this document is a description of what I said or was planning to say, I have added a few extra pictures to this document compared to the presentation that I did.

I forgot to give the audience a short description of myself, so I have added it in this presentation.

Below is the table of content:

- Rune Hvalsøe – short introduction
- Creative vs. Innovation
- Organizational dysfunction
- Will Lean/Agile help?
- How can Agile help?
- The customer
- Q & A
- Thanks to Vinnova & Andreas Larsson
- Reference – I have borrowed a couple of pictures from various sites, this page include a list of where I borrowed the pictures.



Name: Rune Hvalsøe, Nationality: Danish, born 1965, MSc in Engineering.

Twitter: @runehvalsoe

Blog: [agileblog.danskerne.se](http://agileblog.danskerne.se)

Mail: [rune@danskerne.se](mailto:rune@danskerne.se)

Working at Sony since 2011 as a section manager.

I have been working at IBM and Nokia as SW developer and Agile coach etc. – details can be found at my LinkedIn profile: <http://se.linkedin.com/pub/rune-hvals%C3%B8e/0/149/7a7/>

I have been working with Agile since 2008, driving implementation of Agile at Nokia and Sony.

I have been working with Agile as a developer, a Scrum Master, a Product Owner, an Agile coach, a section manager and have seen most of the challenges that we face in striving to become better at improving our way of working.

Most of my Agile experience is with Scrum, Kanban, I also have some experience with Lean.

I enjoy to speak about Agile at request, feel free to contact me if you would like me to speak about Agile/Scrum.

## Creative vs. Innovation

Before we proceed, I would like to give my definition of innovation and creativity:

- Creativity – ideas
- Innovation – delivery of new ideas.

I started to show a couple of super important innovations in my daily life – I have diabetes and need insulin – so the insulin pen is of very high value to me.

Another innovation is when someone combined cargo shipment with kite-surfing – the project was launched December 17 2007 – it has a 160 m<sup>2</sup> computer controlled sail, below is a picture of the ship with the sail.



Another innovation was implemented at Sony, the battery time has for a long time been a problem with Smartphone's – a small group of engineers was asked to find a solution to this problem and came up with the "Super stamina" implementation. The idea is that the phone is going into a special sleep mode when the screen is turned off, all network traffic is disabled, it is still possible to receive normal calls and SMS, but IP-calling like Skype is not possible with this mode active.

This innovation increased the battery time with almost 100% - a very important innovation for the end consumer!

Below is a screen shot of the Super stamina mode setting:



Most innovations happen over time, i.e. you gradually develop your product - Radical innovations involve finding an entirely new way to do things. As such they are often risky and difficult to implement. Most larger organizations and most managers are poor at radical innovation. If you had been making **LP records** then you could have introduced incremental innovations in your design and marketing. However if this was your strategy then a radical innovation, the CD, would eventually kill you. The **CD manufacturer** could similarly introduce various incremental improvements. Once again a radical innovation, **music streaming**, would make your offering obsolete. So we need to constantly look for incremental innovations and radical innovations. We need to develop creativity and turn it quickly into innovation. Combining kite-surfing with cargo shipment is to me another example of radical innovation.

How do we create radical innovation?

- I believe that you have to first secure an environment where people create incremental innovation, you need to have teams who are playful and constantly seek new ways to do things, you need people who understand their domain and constantly seek out to try new ideas – I will discuss playfulness later in this presentation.
- I believe that you need input from the world around you, you need to feed the people working in the teams with new ideas, with new techniques coming up, some people

constantly seek new ideas others need a little help to get new input – One way to get new input, could be to regularly invite speakers to your company and to have employees participate in conferences outside the company – this will feed people with new ideas – what you could do afterwards is to have a creative session afterwards (I will discuss creative sessions later), where you explore the new stuff you learned and see if it could be used in your domain – remember that you need to have an open mind when it comes to seeking information and when you explore the options afterwards – it is very important that you stimulate your employees and give them the option to follow their heart - think about the kite-surfing and cargo shipment, I don't know how the idea was created, but I would guess that someone working with cargo shipment happened to either discuss kite-surfing with someone or they was having kite-surfing as a hobby themselves, they must have had a feeling of the power that can be collected from kite-surfing.

Innovation is hard for many companies – especially big companies is facing a serious challenge, I hope that I through this presentation can create insight and ideas of how to overcome the challenges that many of us face.

Why is innovation so important? **Seeing is believing!**



Seeing is believing, you can talk about an idea over and over again – but many people don't listen to what you say, they want to see that it works before they are willing to try it.

Most of us don't learn well from abstraction, we learn from example, we don't learn well in isolation.

When I started to work at Sony, I had a couple of ideas that I brought with me from my past work with Agile, one idea was to use “4+4” planning (description can be found here <http://agileblog.danskerne.se/#post18>) to overcome the challenges we had with meeting deadlines, dependencies, risk etc. – I talked about it for several months, most of the time when I described the principle and ideas, people said that it was interesting and would be worth trying, but nothing happened; I finally implemented it in my own section, it was now visible to others that it was working, my own developers was very skeptical before we did it (to them

it was yet another meeting and potentially a pull toward the old waterfall model, I was equally skeptical before I tried it out the first time), but after we did it, the developers, SM and PO saw the benefit of the work and was now going out to convince others in our company to try it out, and yet nothing happened – I realized that you also have to feel the pain and get someone to push for it, before it happens, so when another team was in trouble with their projects, they gave the “4+4” planning a try and it was amazing to hear all the positive feedback from them – it is weird that we behave like this, but I see it as a result from time that we live in – the Internet spam age, we are constantly bombarded with new information, new ideas, someone is constantly trying to sell us new stuff – so we develop a shield to protect ourselves.

## Organizational dysfunction

The first challenge that we face is the way that our organizations works, they are not created for the world we live in today – at least not as I see it.

I started to work at Nokia in 1999 when Nokia was growing and growing, I was hired as a freelance developer to work on N3310.

We were a team of about 80 people working in the low-end segment (S30), including management, release management, error management, test managers, project managers, core SW developers (creating device drivers, call stack, low level SW etc.) and UI SW – at the beginning we also had marketing and HW in our team – Nokia was selling between 50-100 million low-end phones per year, so we did an amazing job, later on the high-end phones (S40) started to bloom, the organization was changed slightly, so that each physical product was handled by their own SW team, HW and marketing was moved to their own departments, so our group (S30) was reduced from 80 to 50 people, but still we was creating all SW in the phones that we was selling. We was struggling slightly with HW and marketing, but the SW organization was working fantastic, we were small companies within the company, and we were super effective, even though someone claimed that there was an overhead in having people doing part of the same SW in multiple products.

I was moving to the S40 organization in 2009 when I returned from a 2 year expat contract in China, moving the S30 organization from Copenhagen to Beijing.

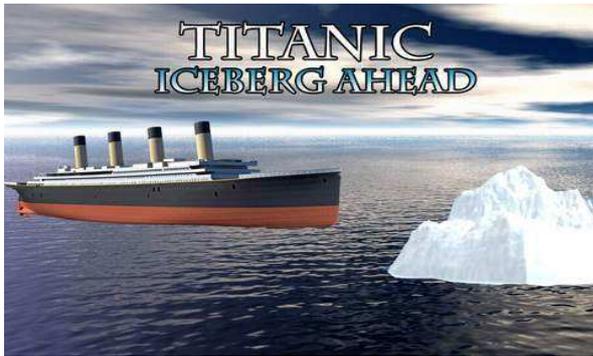
When I started to work in S40 in 2009, I was shocked – they had moved away from the organization build around the physical products and instead tried to become more effective through building competence centers, i.e. all SW for SIM cards was done by one team etc. – this was on the paper more effective, however the employees lost the connection to the products. When I started to work for S40, I saw a silo culture, developers was responsible for their area, their section was given a certain budget to handle their domain and they were fighting to protect their domain – Nokia was facing a big threat as the dominating mobile manufactory, iPhone was released 2 years before and Nokia had missed 2 important innovations in the mobile industry, the touch screen and the dual SIM card.

Mark Parker (CEO at Nike, one of the most successfully innovative companies today) said: “Companies fall apart when their model is so successful that it stifles thinking that challenges it....” ([full story](#))

I personally believe that Nokia made a big mistake when they moved away from the organization focused on the products, it is my experience that having a simple and clear goal is very important and having a feeling of creating a final product together, you could say that the section who was responsible for the messaging also had a common goal, but they are constantly developing a product to be used in a huge number of products, they don't get the feeling that this physical product is something they created; I still remember when one of the developers wrote about an error correction that he made, the description was a little extreme,

but was making the challenge very visible. The developer did a small change in the software, it took him less than one hour to identify and implement the fix, however it took him 1 week of work to finish the task – he had to deliver his fix to so many SW branches and for each branch he had to download the software, add his fix and verify that it was working on that branch, it was very de-motivating for the developers to work in an environment that was 95% bureaucratic and less than 5% software development – bureaucratic is one way to kill playfulness.

It's fantastic to be in a big company – you have a solid economic foundation and you don't need to constantly think if you will have money for tomorrow or next month, but being a big company is also a threat – I like the analogy with Titanic as the big company with all the safe feeling, but when they face a challenge that require that they act fast, they are not able to change fast enough – it doesn't have to be an iceberg, it could also be an opportunity. I believe that part of the problem is that they feel safe, like Mark Parker said: “Companies fall apart when their model is so successful that it stifles thinking that challenges it...”, makes me think about when Nokia considered to explore touch screen or create dual SIM phones...



The Titanic ship is the illustration when you create departments within the company who are responsible for the device driver, another for the call stack, another for the message application etc – you can put it together and creates products – the problem is that the different departments have responsibility of an area NOT a product! And when everyone is responsible for the progress – no-one is

responsible... ☹

I see the way Nokia changed from small companies within the company to one big company with small factories (silos, each one responsible for part of the engine) as a transformation from small boats with a huge responsiveness to one big Titanic ship.

Nokia went Agile in 2009 – and did manage to get some momentum out of this, but not enough to get the benefit from having groups focusing on products with time to innovate...



Another way to organize a company is to have small companies within the company, with a loose connection between them. And Yes – it is much more difficult to control a pool of boats than to navigate a huge company – but the results are so much better when you have a bunch of small companies within the company – you can still have the

service organs to support the boats and you can have a CEO who is in the match official boat – giving directions etc.

You may ask yourself – what does organization have to do with Agile and innovation?

To me, teams is critical to the organization, a high performing team is so much better than the sum of the individual – you often see the cliché “1+1=3”, but when you either observe or participate in a high performing team, you will see that this is often the truth.

If you have playful teams, you have super creative teams and potentially a lot of innovation within your organization – but teams need space to explore creative ideas – I will talk about the book “Slack” later in this presentation.

To me, Agile mean teams and risk management, I will get back to the teams part later on; Agile and risk management may not seem obviously if you are not use to Agile way of working, but the principle in Agile is to identify any risk that you see and execute it as soon as possible, the reason why we do this is that we want to fail early rather than late, often risk is something that you don't know well, i.e. a new technical area or a big complex task, which is often the characteristic of radical innovation, so if you are use to work with Agile, you get use to implementing new ideas in unknown domain and learn to feel “safe” when you do this....

Agile is not just another project management tool, you can explain the basic principle of Agile in about 5 minutes, but it take most people 3-6 month to get use to use it – Agile is a mindset change and can help organizations and teams a long way towards better performance, more innovation, but it is not a silver bullet, you cannot just say you want to work according to Agile and then expect everything to happen, Agile is very much about empowering the teams, letting go of the control, which is very hard for most managers – if you don't empower the teams and work to transform the organization, you will get a little of the good stuff, but you will not release the true power of Agile working and you will not get high performing teams who create products and are innovative at the same time.

## Will Lean/Agile help?



I guess you by now know that I am a strong believer in Agile, I have seen how much we can accomplish with Agile, but I have also seen several cases where the power of Agile has not been released and for those who manage to open Pandora's box, it is a never ending story to work with the teams and the organization, Agile is not just something you buy or decide to use and then everything is perfect, Agile is a mindset change.

Let's try to look at how we can prevent innovation and prevent high performing teams, not that we want to achieve this, but it is often easier to think destructive ☺

## How to prevent high performing teams and innovation?



*Unhealthy internal competition* and *protection obsession*, when management is fighting internally about resources and domain, they create a destructive environment, you start to see teams who work counter-productive to protect their domain, you start to see blame-game, i.e. when a team receive an error report, they try to push it to another team, rather than working together to find the solution.

*Bureaucratic* is a fantastic effective way to demoralize people! Remember the example with the developer who used 95% of his time to do work that he did not like, the way the work was organized was ineffective and frustrating.

I remember another example from my past, a developer had an idea to improve a feature to give a much better user experience, he estimated that it would take him 2 hours to implement the change, but before he was allowed to start working on the change, 6 managers was discussing for 1 hour if he was allowed to use his time on this.

*Control*, Waterfall project management is build on the illusion that you have control, if you want to have strong control, i.e. if you create a blue print for every feature that you want the team to do, and you focus on getting the implementation exactly like the blue print, you will kill all creative ideas – some will claim that this is fine if the team is building the control system for an airplane or something else where following standard is very important, but even here you sometimes need people to think out of the box, and question if we are doing the right thing – and if we do software to be used by ordinary end consumers, it is even more important that you question the blue print and try to see it from the end consumers perspective.

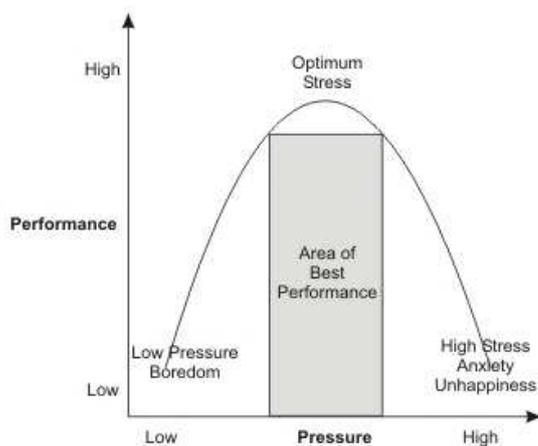
When the focus is on control, people knows exactly what you expect from them, they know that there is no room for variations, you focus on staying on track, report the progress and focus on the next event in the plan, the risk is that you become very mechanic in your behavior if there is no empowerment and engagement.

And an old saying goes “you get what you measure”.

*Ego* – If you have Bosses who reject ideas and kill innovation through fear and ridicule.

*No organization culture for innovation*, think about it for a second, does your organization encourage innovation, if you have a new idea, you might even have proven that it works, whom do you turn to, to get it into production or how do you move forward? Will you have to fight for your idea or does the organization have a process for innovations? Imagine that you think you have a great innovation and you realize that you have to fight for it (which is sometimes good, as it will only allow the best ideas to pass) and at the same time you have to take care of you normal job...

*100% busy* all the time, a very common problem as I see it, there is a lot of research about this challenge, a classic research is the Yerkes-Dodson law, it show that performance decrease if pressure is too high.



The Inverted-U relationship between pressure and performance

The figure is different for each individual and has another shape and position for simple tasks.

I believe the 100% busy challenge is closely related to the *Task switching* challenge.

I had an eye opener experience in Beijing in 2009 when I introduced Kan-Ban way of working for our maintenance at Nokia,

a description can be found here: <http://agileblog.danskerne.se/#post13> – we found that task switching was killing our performance, we removed the task switching and increased productivity with 70%.

I think that we can easily agree that we should not allocate ourselves or others 100% if we want a place where people are creative and innovative (implement creative ideas), I often encourage people to use the Pomodoro technique to be more effective – a good idea would then be to allocate creative time and potentially time to implement the creative ideas, I use to call Pomodoro “one man scrum”.

The main question was “Will Lean/Agile help?”

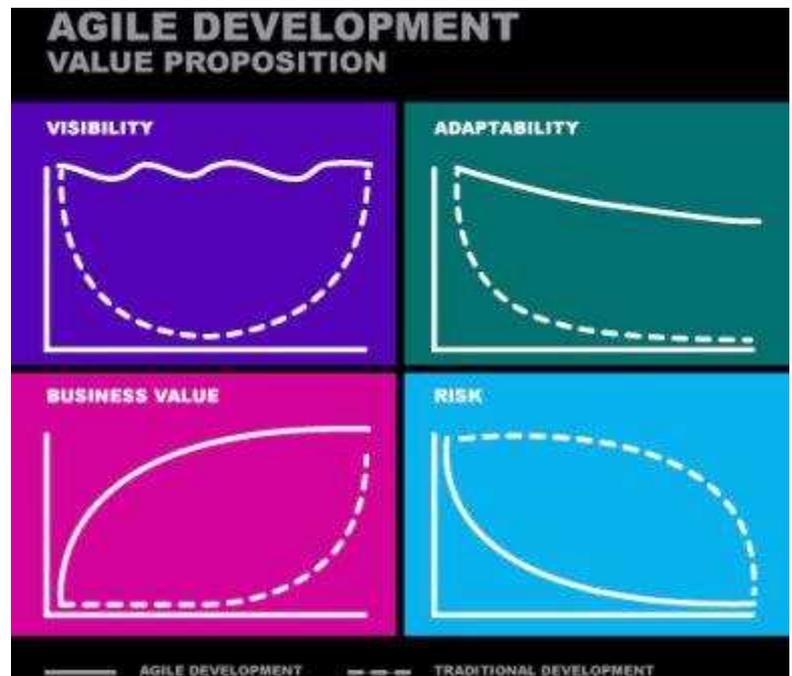
I would answer it with a huge Yes!



## How can Agile help?

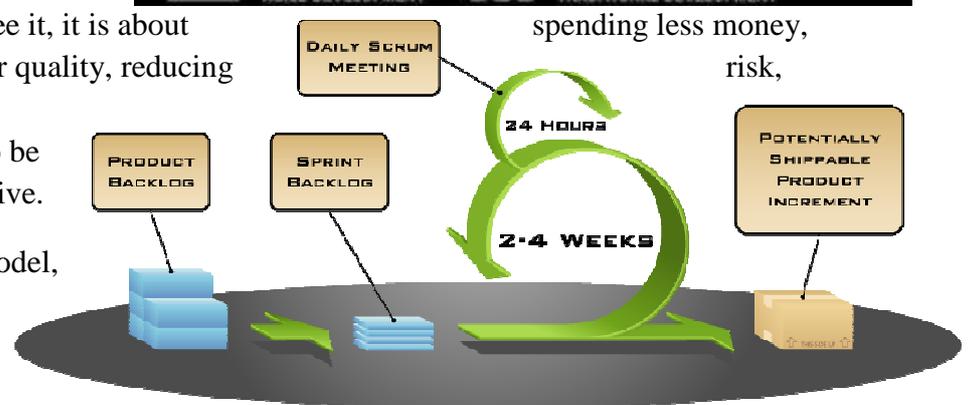
Let me quote Israel Gat, formerly of BMC Software “To succeed with Agile, management’s need for results must be greater than their need for control.”

Agile is a mindset change more than anything else – i.e. you can more or less explain Scrum in 5-10 minutes – the same goes for more or less any other Agile discipline.



Agile has a lot of value as I see it, it is about becoming faster, create higher quality, reducing delivering more value to the customer etc. – but it can also be used to become more innovative.

In the traditional Waterfall model, if we have an idea, we probably have to make a decision and discuss if we should start a 10M € project, it will require that you are ready to fight for your idea and push to get the funding.



COPYRIGHT © 2005 MOUNTAIN DOG SOFTWARE

A very well known project is when Xerox created graphical interface to computers, it is not famous because Xerox did it, but because Steve Jobs bought it and used it in Apple computers. I don’t know how the graphical interface project was started, but Agile was not around before 1980, so I assume that it was a traditional project with a huge funding, according to my understanding (which I have from the biography of Steve Jobs – fantastic book!), Xerox had create the project to implement graphical interface to computers, but they did not intend to use it, so it was potentially a very expensive innovation investment that was not giving any return, they managed to get some or all their cost back through Apple.



Often when you read or talk about innovation, you get the comment “Creativity costs less but Innovation requires huge investment with respect to time, efforts, money, manpower, energy, space etc etc.”.

Agile is perfect in the way that it can be used to test your assumptions, it is more or less build into the model. An Agile principle is to reduce risk and execute any task that has a high risk

first – often this type of task is a “spike”, i.e. a short investigation or prototype of the idea – the risk reducing principle is “Failing fast, failing cheap” (if it doesn’t work) – and we do not use Agile because we want to fail – The purpose of test is to learn, if you test often and early, you will create a fast growing learning curve

“Failing quickly is actually not that bad,” says CEO Jeff Immelt, a 31-year veteran of General Electric. “Failing slowly is deadly.”

With Agile we often prototype ideas and check if it is working, we might need a few man-weeks to verify that it works, but it is much cheaper than “traditional” innovation projects.

In the last section we looked at the following elements:

- *Unhealthy internal competition and protection obsession,*
- *Bureaucratic*
- *Control,*
- *Ego*
- *No organization culture for innovation*
- *100% busy all the time*

I would like to shortly discuss and address each item with focus on how Agile will help, or if there are things that we should have in mind.

*Unhealthy internal competition and protection obsession:* There is no simple way to cure an unhealthy culture, not even Agile, but if you implement Agile in the organization, you build an environment with strong collaboration.

Collaboration is super important when we talk about creative ideas and innovation,

*Bureaucratic, ego and Control,* is all closely connected – one of the most important principles in Agile is empowerment! You only get strong engagement if you empower the teams! I use to think that the only person, who should be responsible for the backlog, is the PO (Product Owner), however I have realized that you get a much better setup if you encourage the team to add stories to the backlog.

I have a team in my section, they have a PO who is not allocated 100% to work with the team and the backlog, it would be more correct to say that she is working 20% with the team and the backlog and the rest of the time is used for other work – this means that the team including the PO cannot work as “Agile by the book”, but instead the PO is the sales agent for the team and the backlog, the PO has a lot of knowledge about the topic and can help to guide the team; the team develop the Sketch application on the Sony Mobile phones, the first version can be admired on Sony Xperia Z Ultra.

The Sketch team is creating at least 50% of the stories on the backlog – they add stories

- During the sprint, when they discuss stories they come up with new ideas which they add to the backlog

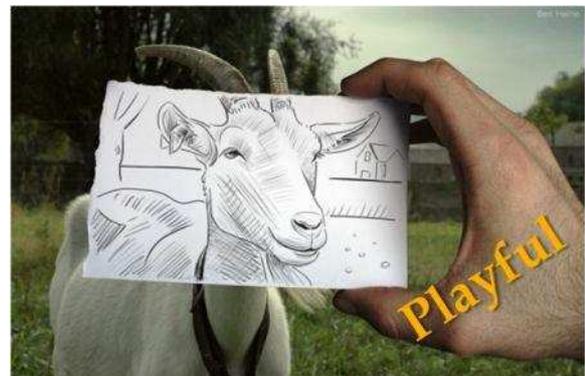
- During “backlog grooming” – where they estimate and discuss stories on the backlog, again they get new ideas as they discuss existing ideas
- During creative workshops – they discuss visions, future plans and create stories during a creative workshop – very much like the “Creative week workshop” that I described here: <http://agileblog.danskerne.se/#post26>



The picture above is from our “Creative week workshop”.

*No organization culture for innovation*, if you want to have innovation, you need to create an environment where people can be playful, you want to have an environment where people are curious, where you often see a group of people gathering around a table to see a new idea and discuss and appreciate it.

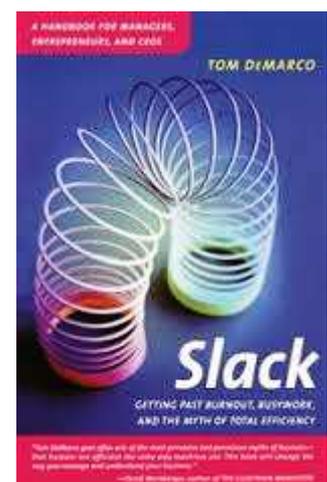
You want to have an environment where you give credit to creative ideas and innovation – where you constantly learn and improve.



We have learned that it is a good idea to have cross-functional teams, i.e. teams with different skills; we have found that a mix of developers and graphical designers is a really good combination to foster creativity.

Scrum (and Agile) is build with this in mind, after each sprint, the team has a retrospect where they identify what to improve in their way of working, this means that you build a culture where people see change and new ideas as something positive!

*100% busy* all the time - Remember people under pressure don't think faster – a quote from the book “Slack” by Tom Demarco, a fantastic book with a huge amount of good stuff.



Most of us don't learn well from abstraction, we learn from example, we don't learn well in isolation.

Great things in business are never done by one person – they're done by a team of people!

Knowledge work is by definition collaborative – if you have a high performing team, you will notice that the team work is more than the sum the individual, you often see expressions like “1+1=3”, together you can create much more than what you can achieve alone – “there is no ‘I’ in ‘team’” – as an example I mentioned that I was on a management workshop just before I did the presentation at

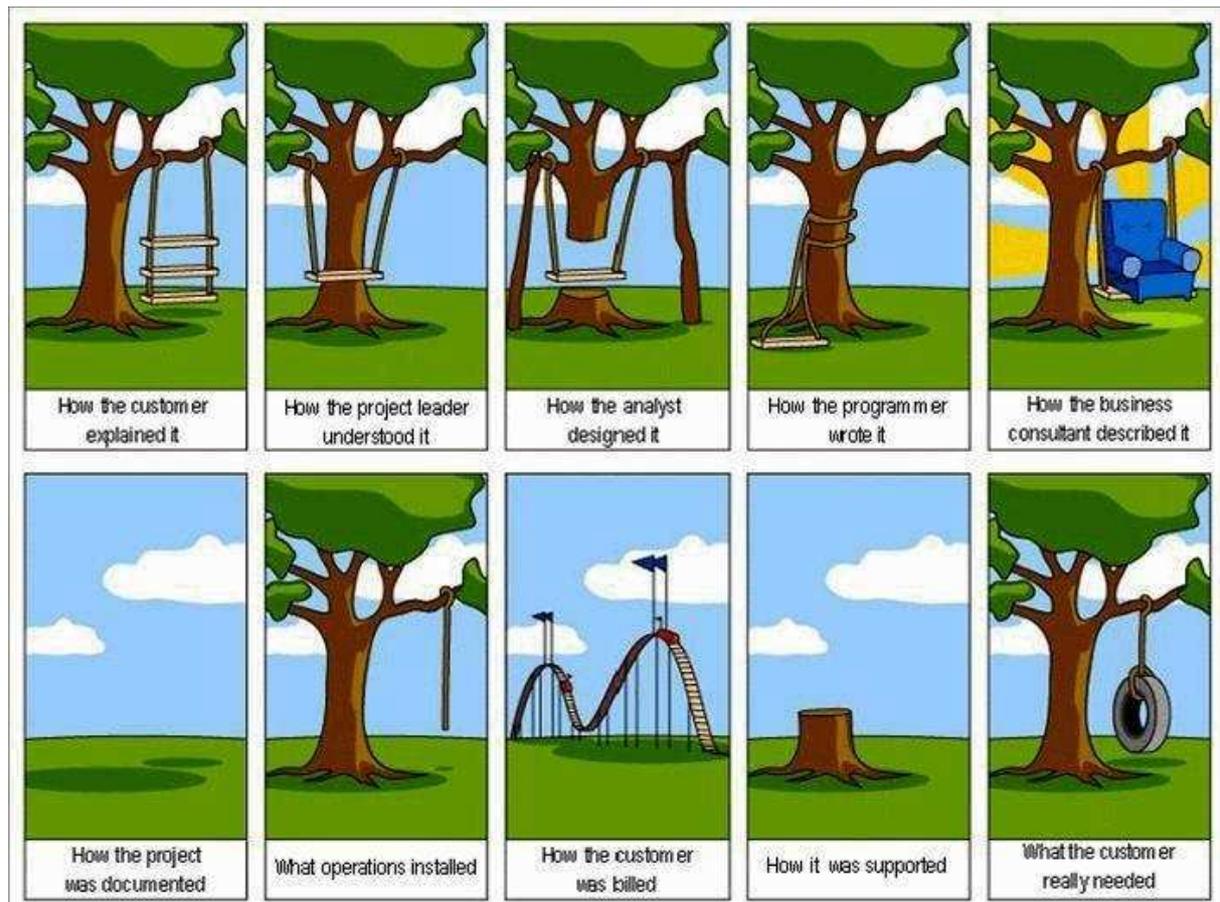
“Innovation in mind”, we were discussing how we should secure that our developers had the right level of knowledge and was inspired by new trends; we started to discuss using CoP (Community of Practice) which is a strong tool to share knowledge and solve common problems, and we ended up with a solution that we call “Tech Tribe”, inspired by “Lean Tribe” where those who have a common interest meet once a month, and listen to someone who present new ideas etc. – we decided to setup a small group of people who will drive a series of “after work meeting” and who will secure that the speakers (internal or external) feed our developers with new and interesting information about what is going on out there, and at the same time, the group will meet with our developers on a regular basis and act as CoP, where all kind of topics can be brought up.



Make sure that you encourage the team to “fika” as we say in Sweden, i.e. meet during the day to have a cup of coffee and some cake, the team will build a strong relation during these sessions, and very often they will also bring up work related challenges. Remember the discussion about Yerkes-Dodson law, it is important that the team has some space to develop.

## The customer

If you don't know your customer, you don't know what quality is! A simple and yet very powerful statement that we believe in!

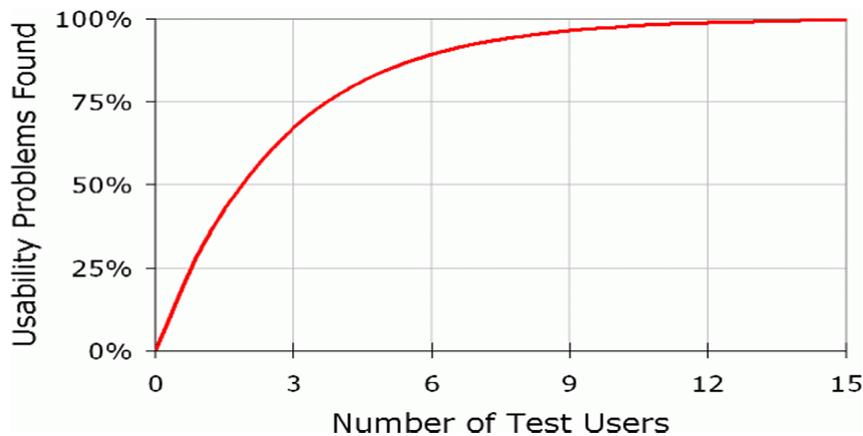


Bryan Adams - (Everything I Do) I Do It For You.

Our teams test the software in each sprint, to make sure that the quality is good, to make sure that the story is implemented like they planned, but they have one problem, they are engineers and as such they cannot think like an end consumer think – I remember that I once complained to my wife that she never returned my SMS, she said that it was because she did not notice when she received a SMS, it turned out that the sound was off when she was receiving SMS, she had tried to change it and our teenage daughter have also tried, but none of them was able to find the place where you change the sound for incoming SMS; the challenge is an Android challenge, it is not related to Sony, it is exactly the same challenge on all other Android phones – I looked at the phone, went into settings, only to discover that I could not find out where to change the SMS sound, but I am an engineer and did not give up so easy, so after playing around with phone for some time, I managed to find out that you have to enter messaging to find the setting for SMS sound.... When you have an experience like this, you start to think that it might be worth to change our way of working slightly.

We started a cooperation with LTH to test the usability of our software, Jacob Nielsen (a UX guru) have done some investigation in this area, and on his web-site you can find a description

of his findings, the graph below is from his website <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> - the recommendation is to use 5 testers, however we have decided to only use 3 testers due to time limitations but we perform the test frequently (every 4 week).



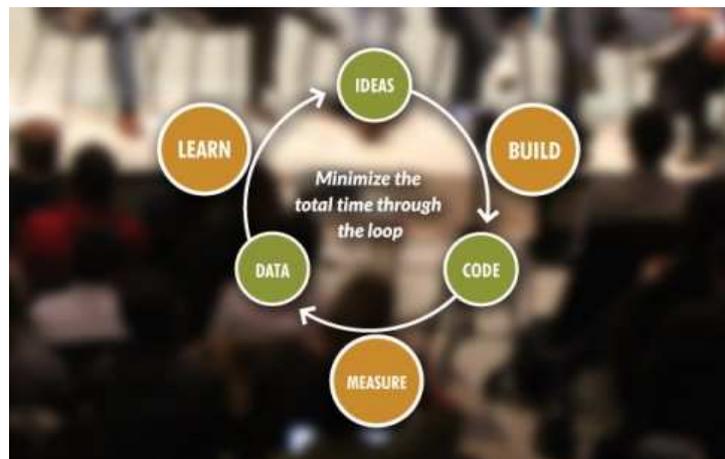
The benefit from Source: Jacob Nielsen the usability testing with LTH is not only to identify the challenges that the students help us to find, but it is also about giving the developers, the interaction designer and the PO a much wider view on the end customers behavior and knowledge. Even though the interaction designer create a good blueprint, the developers always interpret it and need all the knowledge to interpret with the end user in mind and also be able to stand up and challenge the design in the blue print.

We have seen a positive development in the usability after we started to do the frequent usability test, and the team is very positive to the input they receive from the test – a more detailed description of the test can be found in another presentation <http://agileblog.danskerne.se/#post20>

We have started to extend our search to understand the end consumer - we are planning to visit a café one day with the team, so that they can watch the customers play with their or competitors products - to learn how they use them, for what purpose and what they would like to do with the tools - this will help them to create the right ideas when they (the team) contribute to the backlog...

We have also started to implement another important principle in Agile – we call it “Closing The Loop” (CTL), it is related to the principle of getting feedback as often as possible.

We collect data from the end consumer; we analyze the data and try to get a picture of how they use our software; we change our software to reflect the new picture we get and release a new version and continue the loop.



As an example, we found out that the globe in our Album implementation (to tag where you took your picture and get an overview) was only used by 0,01% of the users, we moved the globe feature to the dash board and noticed that it was now used by up to 3% of the users.

You may ask yourself, why is it important to understand the customer to be innovative?

I hope you already have the answer to this one – you need to have some kind of understanding/judgment if your innovation can be used by a customer out there, you need to test your assumption, but without basic understanding, it will be difficult to get anywhere...

## Q & A

I do not remember all the questions or the details, this is what I remember:

1. Is it possible to run a waterfall project in iterations?
2. Do you need to create a solid architecture if you create an innovation?
3. How do you get developers to select and implement creative ideas?

*Is it possible to run a waterfall project in iterations?*

I remember that I was running out of time here, I said yes without elaborating on the topic, I believe that you will end up with Scrum if you do this; let's imagine that you want to have an iteration in your Waterfall project every 2. week to demonstrate what you have achieved at this point (a milestone), that would be more or less the same as Scrum, you would start to break up your project into tasks that can be solved in 2 week cycles, you might not have the morning meetings, but you will have something similar, but less effective, you might not have retrospective, but normally you have lesson learned in waterfall, which is exactly the same as retrospect in Scrum, you don't have sprint planning, this is done by the project manger, but I have never seen a project manager who did not consult the teams before doing the planning, so you will have a less effective sprint planning, you might not have poker-planning (story size estimation), but the project manager will ask each individual to estimate the different tasks, etc. etc. – as I see it you will have a limited version of Scrum with less advantages – so why bother and not just go for the real thing?

*Do you need to create a solid architecture if you create an innovation?*

Absolutely no – when you implement a creative idea – as a proof of concept, you don't need the final quality – it is a hack and the only purpose is to show that it is possible – you can do the real thing, but in most cases you should only prove that it is possible and create something that can demonstrate the creative idea.

*How do you get developers to select and implement creative ideas?*

The question came when I described the “creative week” that we plan, and it is actually a problem to get the developers to select team and idea, we decided to ask them to make their choice Thursday before the creative week, so we had Friday to go around and “talk/push” the developers who did not manage to select idea/team – we want to have everyone ready Monday morning when we start our creative week.

## **Thanks to**

Andreas Larsson for helping me creating the presentation – Andreas is Associate Professor, Innovation Engineering at Department of Design Sciences at Lund University, Faculty of Engineering

Vinnova (<http://www.vinnova.se/sv/>) – who have sponsored a research project which we used for inspiration

## Reference

<http://spamup08.deviantart.com/art/Seeing-is-Believing-Iris-201383559> borrowed "seeing is believing" eye.

[Cat face German Sheppard's](#)

[Yes – No – Maybe cube](#)

[Yes contacts](#)

[Agile picture](#)

[Learn – build – measure picture](#)

[Customer picture](#)

<http://jamienotter.com/2012/04/the-feeling-of-failure/> - picture borrowed, man "thinking"

<http://www.brainpickings.org/index.php/tag/flickr/> - picture borrowed with goat

[Titanic ship](#)

[Racing ships](#)

<http://www.mountangoatsoftware.com/> - Scrum picture