

- Home
- News
- Local info
- Registration
- Program
- Organization

Call for submissions

Program

[Printable version of the complete program](#)

Monday, 31 May 2010: PhD Symposium

09:00-16:00	PhD Symposium
-------------	---------------

Tuesday, 1 June 2010: Workshop / Tutorial Day

08:30-10:00	Jeremy Lightsmith, Jon Jagger Intro to Test Driven Design (TDD) & Pairing	Sumeet Moghe, Rixt Wiersma Facilitating dialogue in situations of conflict		Ana Moreno, Agustin Yague Dealing with usability in an agile domain	Yves Hanouille The Leadership Game	Chris Matts A gentle introduction to Feature Injection	Pierluigi Pugliese, Mike Sutton Stuck in a rut? Improve Your Solutions!	Erik Lundh Making Sense of Lean and Agile with Getting Things Done (GTD)
10:30-12:00	Intro to Test Driven Design (TDD) & Pairing	Tomi Juhola How to relieve the agile adoption pain?	Tom Perry Treating Impedimentia	Jeff Patton User Story Mapping	The Leadership Game	A gentle introduction to Feature Injection	Gwyn Morfey Stealthholders, Shapeshifters and Sergeants: Working With "Hard" Clients	Jutta Eckstein Agile Software Development for Distributed Teams
13:00-14:30	Pete Tansey Agile Project Risk Assessment	Andreas Larsson Welcome to the Trust Spiral! Going up?	Geir Amsjø Scrum Master group therapy	Sumeet Moghe, Chirag Doshi The Distributed Agile Game	Yaser Ghanam, Frank Maurer The Second XP Workshop on Agile Product Line Engineering	Jon Gunnar Wold, Ram Yoga How UX can save your agile project from failure	Amanda Laucher Functional Programming with XP	Mary Poppendieck Lean in a Nutshell
15:00-16:30	Johannes Brodwall, Niklas Bjørnerstedt Agile release strategies	Sami Honkonen, Arto Eskelinen GROWing effective coaching questions	André Dhondt How do I measure up: the Agile Skills Project	The Distributed Agile Game	The Second XP Workshop on Agile Product Line Engineering	How UX can save your agile project from failure	Ola Ellnestam, Daniel Brolund Large-scale refactorings using the Mikado Method	Joshua Kerievsky System Metaphor Revisited: The Lost XP Practice

[Detailed overview of morning sessions.](#) [Detailed overview of afternoon sessions.](#)

18:00 Conference Reception

Wednesday, 2 June 2010: Main Conference

08:30-10:00	Conference Opening and Keynote by Scott Page: Leveraging Diversity in Parallel: Perspective, Heuristics, and Oracles
-------------	--

Conference Opening and Keynote by Scott Page: Leveraging Diversity in Parallel: Perspective, Heuristics, and Oracles.

Leveraging Diversity in Parallel: Perspective, Heuristics, and Oracles. **Scott Page (University of Michigan Ann Arbor)**. (Keynote, 60 min) [\[Link\]](#)

Scott Page is the Leonid Hurwicz Collegiate Professor of Complex Systems, Political Science, and Economics at the University of Michigan Ann Arbor and a member of the external faculty at the famous think tank Sante Fe Institute. He is an acclaimed speaker, researcher, educator, consultant, and author. Both of his excellent books "The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies" and "Complex Adaptive Systems: An Introduction to Computational Model of Social Life" (co-authored by John H. Miller) have important implications for collaborative software development. We asked Scott to illuminate us about the role of diversity in activities that we frequently perform in software development: problem solving, decision making, and prediction. For a discussion of the relationship between Scott's theories and software development, see the editorial in IEEEESW on Diversity and Software Development (<http://www.computer.org/portal/web/csdl/doi/10.1109/MS.2009.62>).

10:00-10:30	Break					
10:30-12:00	<table border="1" style="width: 100%;"> <tr> <td>Software craft</td> <td>Foundations of Agile</td> <td>Collaboration</td> <td>Software Design</td> <td>User Stories</td> </tr> </table>	Software craft	Foundations of Agile	Collaboration	Software Design	User Stories
Software craft	Foundations of Agile	Collaboration	Software Design	User Stories		

Software craft. Session chair: Jessica Hildrum (Objectware)

Growing and Fostering Software Craftsmanship. **Cory Foy (Cory Foy, LLC)**. (Invited talk, 45 min) [\[Link\]](#)

Software Craftsmanship is sometimes dismissed as being only for individual developers and/or boutique software shops. However, the practices and principles of craftsmanship are just as vital and valid to large organizations as individuals. In this talk, Cory will cover practices and approaches for adopting craftsmanship in your organization. We'll cover specific exercises, organizational challenges, and how to influence both organizational and team-level acceptance.
 Bio: Cory Foy is an agile developer, consultant and coach with a passion for looking at the entire system within an organization. His background consists of highly technical positions in Java, Ruby, .NET and C#, including working for Microsoft as a Premier Field Engineer debugging critical enterprise applications in .NET and C#, developing mobile applications using J2ME and Objective-C and building client-side applications for financial transfer using C#. Cory has also been a coach leading agile transitions with large distributed teams and a frequent speaker at conferences including the SQE Agile Development Practices conference and the Software Craftsmanship North America conference. He's also very passionate about the development community, helping start or run user groups in Florida, North Carolina and Missouri, and serving as the Global Community Liaison for the Scrum Alliance.
 Cory Foy <http://www.coryfoy.com> http://twitter.com/cory_foy

Agile fails without Craftsmanship. **Tore Vestues (Bekk)**. (Lightning talk, 10 min) [\[Link\]](#)

With the craftsmanship movement, focus has shifted from viewing the developer as a production unit in software engineering to viewing the developer as a craftsman with responsibility and individual skills. Agile with its lightweight methodologies can't afford heavy processes to control production unit developers. In order to be lightweight an agile process needs a structure of responsible craftsmen that takes responsibility for the quality by itself.

Agile self-help through Communities of Practice. **Pierluigi Pugliese (Connexxo)**. (Lightning talk, 10 min) [\[Link\]](#)

Communities of Practice are a more and more used way to capture tacit knowledge and promote cross-functional and cross-project learning. Even though such a tool suits perfectly the needs of an agile and continuously learning organisation, it is unfortunately seldom used by agile coaches. This presentation is intended to promote Communities of Practice, describing how they can be created and maintained and what benefits can be obtained.

Beauty is in simplicity. **Jørn Ølmheim (Statoil)**. (Lightning talk, 10 min) [Link]

Will expand on the topic of my article from 97things every programmer should know: http://programmer.97things.oreilly.com/wiki/index.php/Beauty_Is_in_Simplicity. The talk will be a short discussion of what I think is beautiful code, and include some examples.

We are trained to avoid responsibility. How to find a way to be responsible? **Sergey Dmitriev (Making Waves)**. (Lightning talk, 10 min) [Link]

Some years ago I met Christopher Avery (PhD, <http://christopheravery.com/>) on Agile conference and listened to one of his talks about the Responsibility Process. Through thousands of years human brain have been trained to avoid responsibility, we have build an subconscious and automatic process to guard us against it. It's always somebody else's fault, right? It's the circumstances :), not us, right? In this talk, I'll try to present The Responsibility Process (TM) itself as well as my personal experiences about its practical usage. Ill show you a way to become more responsible!

Foundations of Agile. Session chair: Letizia Jaccheri (NTNU)

Towards and Understanding of the Conceptual Underpinnings of Agile Development Methodologies. **Sridhar Nerur (University of Texas at Arlington), Alan Cannon (University of Texas at Arlington), VenuGopal Balijepally (Prairie View A&M University) and Philip L. Bond (University of Texas at Arlington)**. (Invited talk, 30 min) [Link]

While the growing popularity of agile development methodologies (ADM) is undeniable, there has been little systematic exploration of its intellectual foundation. Such an effort would be an important first step in understanding this paradigm's underlying premises. This understanding, in turn, would be invaluable in our assessment of current practices as well as in our efforts to advance the field of software engineering. Drawing on a variety of sources, both within and outside the discipline, we argue that the concepts underpinning ADM are by no means novel. In the tradition of General Systems Theory (GST), this paper advocates a transdisciplinary examination of ADM to extend the intellectual boundaries of software development. This is particularly important as the field moves beyond instrumental processes aimed at satisfying mere technical considerations.

Agile Software Development Methods: A Comparative Review. **Pekka Abrahamsson (University of Helsinki), Nilay Oza (VTT Technical Research Centre) and Mikko T Siponen (University of Oulu)**. (Invited talk, 30 min) [Link]

Although agile software development methods have caught the attention of software engineers and researchers worldwide, scientific research still remains quite scarce. The aim of this study is to order and make sense of the different agile approaches that have been proposed. This comparative review is performed from the standpoint of using the following features as the analytical lenses: project management support, life-cycle coverage, type of practical guidance, adaptability in actual use, type of research objectives and existence of empirical evidence. The results show that agile software development methods cover, without offering any rationale, different phases of the software development life-cycle and that most of these methods fail to provide adequate project management support. Moreover, quite a few methods continue to offer little concrete guidance on how to use their solutions or how to adapt them in different development situations. Empirical evidence after ten years of application remains quite limited. Based on the results, new directions on agile methods are outlined.

From exotic to mainstream: A 10-year odyssey from Internet speed to boundary spanning with Scrum. **Richard Baskerville (Georgia State University), Jan Pries-Heje (Roskilde University) and Sabine Madsen (Roskilde University)**. (Invited talk, 30 min) [Link]

Based on four empirical studies conducted over a 10-year time period from 1999 to 2008 we investigate how local software processes interact with global changes in the software development context. In 1999 companies were developing software at high speed in a desperate rush to be first-to-market. In 2001 a new high speed/quick results development process had become established practice. In 2003 changes in the market created the need for a more balanced view on speed and quality, and in 2008 companies were successfully combining agile and plan driven approaches to achieve the benefits of both. The studies reveal a two-stage pattern in which dramatic changes in the market causes disruption of established practices, experimentation, and process adaptations followed by consolidation of lessons learnt into a new (and once again mature) software development process. Limitations, implications, and areas for future research are discussed.

Collaboration. Session chair: Xiaofeng Wang (Lero)

What Agile Teams Can Learn From Sports Coaching. **Pdraig Brennan (Ericsson)**. (Experience report, 30 min) [Link]

Experience from numerous agile teams in the transition to agile methods in the Ericsson Operation and Maintenance has shown that teamwork and collaboration are crucial to achieving success. This finding motivated a review of how sports coaches work to build effective teams. The review highlighted certain characteristics that sports coaches deem fundamental to success. This report details these findings and how we use these characteristics to helping teams improve.

Communication in Context: a Stimulus-Response account of Agile Team Interactions. **Nik Nailah Binti Abdullah (National Institute of Informatics), Helen Sharp (The Open University) and Shinichi Honiden (National Institute of Informatics)**. (Short research paper, 15 min) [Link]

Previous research has indicated that work artefacts commonly used by agile teams capture progress information, while functional aspects such as requirements are developed and sustained through the team's social interactions and communication channels. This paper reports an initial empirical study to investigate the relationship between agile work artefacts and communication during stand-up meetings and pair programming sessions, specifically focusing on gathering and clarifying requirements. Using Bateson's communication theory, we found that the work artefacts, and other individuals form an external event system which supports Agile teams during the gathering and clarifying of requirements. Using this communication theory together with Cancey's situated cognition, we predict that if the two do not exist together throughout the interactions, then teams members will form disconnected actions together.

Effective teamwork. **Viktor Gulliksen (University of Oslo)**. (Lightning talk, 10 min) [Link]

This lightning talk is about the challenges of effective teamwork. I will cover themes like communication and coordination within and across teams, and talk about my experience of using a team radar to assess and improve teamwork in an agile software project.

How to Engage Subcontractors in Scrum Projects?. **Jakub Rudzki (Solita Oy)**. (Lightning talk, 10 min) [Link]

This talk presents a few practices used for building a working Scrum team with subcontractors. The practices are based on experiences gained in working with subcontractors. The emphasis of the talk is put on personal contact, equality, and fun as elements engaging all team members into a Scrum project. The practices are presented at general level with a few specific examples. The content of the talk is based on a book chapter "Considering Subcontractors in Distributed Scrum Teams" that will be presented at workshop: <http://xp2010.agilealliance.org/node/5467>

Understanding the Importance of Trust in Distributed Agile Projects: A Practical Perspective. **Siva Dorairaj, James Noble and Petra Malik (Victoria University of Wellington)**. (Short research paper, 15 min) [Link]

Agile methods rely on face-to-face communication but are being used in distributed projects. We have conducted grounded theory research with Agile practitioners to collate the strategies they use to overcome the challenges in distributed projects. In this paper, we argue that trust is one of the key factors in determining the success or failure of distributed Agile projects, and describe how trust can be generated and sustained by increasing effective communication and understanding cultural differences.

Software Design. Session chair: Angela Martin (University of Waikato)

Thawing the 'Design Winter'. **Michael Feathers (Object Mentor)**. (Invited talk, 45 min) [Link]

We've been practicing 'emergent design' for years within the agile community. Some teams have been successful at it and other teams missed the subtlety. They just wrote code with tests paying little attention to design. When ideas become subtle they are lost. The first generation knows that they can do what is not discussed. The second generation doesn't even know that there is anything to discuss. In this talk, Michael Feathers will talk about his experiences assessing the state of design in various teams around the world and efforts teams have made to reintroduce a sense of urgency about design. He will also discuss ways that we as a community can start to communicate about design in much more direct ways which leave space for subtlety but enough prescription to highlight its importance.

Bio: Michael Feathers is a consultant with Object Mentor. He balances his time between working with, training and coaching various teams around the world. Publically, Michael developed Cppunit, the initial port of JUnit to C++, and FitC++, a C++ port of the FIT integrated-test framework. Michael is also the author of the book 'Working Effectively with Legacy Code' (Prentice Hall 2004).

Stakeholder Engagement in the Evolutionary Design of an API. **Ken Power (Cisco Systems)**. (Experience report, 30 min) [Link]

Iterative development of non-GUI software in general, and APIs in particular, in an agile context presents a number of challenges, not the least of which is effective engagement with people outside the development teams as the API evolves. We use two strategies to overcome these challenges. The first strategy is effective stakeholder engagement. This paper describes how we identify stakeholders in our product's API, who those stakeholders are, how we engage with them, and how we incorporate feedback on a continuous basis. The second strategy is the development of an API Test Client Application that allows many different stakeholders to use the product directly as it evolves. The Test Client Application evolves in parallel, iteration by iteration, with the main product. The experiences described here can be of benefit to anyone developing an API product with multiple consumers, for example an in-process library, an out-of-process service, a Web Service, or a service in Service-Oriented Architectures.

Structuring Complexity Issues for Efficient Realization of Agile Business Requirements in Distributed Environments. **Richard Mordinyi, Eva Kühn, and Alexander Schatten(Vienna University of Technology)**. (Short research paper, 15 min) [Link]

One of the ideas of agile software development is to respond to changes rather than following a plan. Constantly changing businesses result in changing requirements, to be handled in the development process. Therefore, it is essential that the underlying software architecture is capable of managing agile business processes. However, criticism on agile software development states that it lacks paying attention to architectural and design issues and therefore is bound to engender suboptimal design-decisions. We propose an architectural framework, that by explicitly distinguishing computational, coordinational, organizational, distributional, and communicational models offers a high degree of flexibility regarding architectural and design changes. The framework strength is facilitated by a) combining the characteristics and properties of architectural styles captured in a simple API, and b) offering a predefined architectural structure to the developer of distributed applications to cope with complexities of distributed environments. The benefit of our approach is a clear architectural design with minimal mutual effects of the models with respect to changes, accompanied by an efficient realization of new business requirements.

User Stories. Session chair: Tore Dybå (SINTEF)

Auto-tagging Emails with User Stories Using Project Context. **S. M. Sohan, Michael M. Richter and Frank Maurer (University of Calgary)**. (Research paper, 30 min) [Link]

In distributed agile teams, people often use email as a knowledge sharing tool to clarify the project requirements (aka user stories). Knowledge about the project included in these emails is easily lost when recipients leave the project or delete emails for various reasons. However, the knowledge contained in the emails may be needed for useful purposes such as re-engineering software, changing vendor and so on. But, it is difficult to relate texts such as emails to certain topics because the relation is not explicit. In this paper, we present and evaluate a technique for automatically relating emails with user stories based on their text and context similarity. Agile project management tools can use this technique to automatically build a knowledge base that is otherwise costly to produce and maintain.

Prototypes are forever Evolving from a prototype project to a full-featured system. **Hugo Corbucci, Mariana V. Bravo, Alexandre Freire da Silva, and Fernando Freire da Silva (Agilbits)**. (Experience report, 30 min) [Link]

Prototypes are a well known, widely accepted development practice but, if not carefully evolved, they can become a nightmare to maintain. This paper presents the experience of a four person agile team who successfully grew a prototyped system to a full-featured application without any clear transition in the project. The paper describes how the project started with a very simple prototyping goal, evolved through iterations and spikes to a partly working system and transformed, in the end, in a complete application widely tested and refactored.

A Systematic and Lightweight Method to Identify Dependencies Between User Stories. **Arturo Gomez (BTH), Gema Rueda (BTH) and Pedro P. Alarcón (Universidad Politecnica de Madrid).** (Short research paper, 15 min) [\[Link\]](#)

The order in which user stories are implemented can have a significant influence on the overall development cost. The total cost of developing a system is non commutative because of dependencies between user stories. This paper presents a systematic and lightweight method to identify dependencies between user stories, aiding in the reduction of their impact on the overall project cost. Initial architecture models of the software product are suggested to identify dependencies. Using the method proposed does not add extra load to the project and reinforces the value of the architecture, facilitates the planning and improves the response to changes.

12:00-13:30	Lunch			
13:30-15:00	Lean/Kanban	Scaling	Code Quality	Panel: Is Agile Research Dead in the Water? User Experience

Lean/Kanban. Session chair: Jessica Hildrum (Objectware)

The Five Habits of Successful Lean Development. **Mary Poppendieck (Poppendieck).** (Invited talk, 50 min) [\[Link\]](#)

Lean principles have been around for quite a while now, long enough for companies to try them out and see how well they work. So what does success look like? It turns out that in successful lean development organizations, people routinely dedicate their intelligence and creativity to helping the company become and remain successful. Because of this, the companies themselves are adaptive - they respond to changing market conditions and opportunities faster than their competitors. This talk will be about what these successful lean companies do to energize their workplace.

Bio: Mary Poppendieck has been in the Information Technology industry for thirty years. She has managed solutions in software development, supply chain management, manufacturing operations, and new product development. She spearheaded the implementation of a Just-in-Time system in a video tape manufacturing plant, resulting in dramatic improvements in the plant's performance. Mary's team leadership skills were honed in 3M, where new product development is a core competency. Her teams commercialized products with embedded software three times faster than normal, partnering with small suppliers in the process. A popular writer and speaker, Mary's classes apply lean principles to Software Development problems and offer a fresh perspective on software development processes. Her book *Lean Software Development: An Agile Toolkit* was awarded the Software Development Productivity Award in 2004. *Implementing Lean Software Development: From Concept to Cash*, was published in 2006, and *Leading Lean Software Development* was published in November, 2009.

Done Considered Harmful. **Marcus Ahnve (Valtech).** (Lightning talk, 10 min) [\[Link\]](#)

We are terribly concerned about being done with things. But are we really ever done with things? Done is an ambiguous term. Why else would we need a Definition of done? Definitions of done normally includes quite a few activities, but why are they concealed in a definition? Sometimes "Done" is at the same time so desirable and ambiguous that teams have two or even three versions of it - "Done Done" and "Done Done Done". Teams should discard the notion and need of being "done" and instead find the real states that their tasks have and make them visible.

Scrumban: from Scrum to Kanban in 10 easy steps! **Tor Hovland (Powel).** (Lightning talk, 10 min) [\[Link\]](#)

For many projects it is difficult to make the Scrum process fit. For example, it may be impossible to focus on a month's worth of pre-planned work without substantial interruptions. Kanban takes a different approach, and Scrumban has been proposed as a way to combine Scrum with Kanban. We will talk about how Kanban can be better for your team, and how to go from Scrum to Kanban in 10 easy steps!

Google Culture Ideal for Agile & Kanban?. **Susan Hunter (Google).** (Lightning talk, 10 min) [\[Link\]](#)

I'd like to share some key aspects of Google culture that make it very receptive and an ideal fit for lean and agile practices. Also some aspects that initially seem a barrier. Focus will be on principles such as: transparency; collaboration; agility; openness to embracing change; innovation; ambition; questioning of the status quo. We are about to introduce kanban and mix it up with agile, scrum & lean practices. By March 2010 I will have much to share on how we have soared and how we have crashed.

Improving software quality with Kanban. **Ketil Jensen (Miles).** (Lightning talk, 10 min) [\[Link\]](#)

Bugs is the number one waste in software development. In fact, it has been so since the early days of software development. Finding and fixing bugs late in the software development process is hard, time consuming and expensive. Developer focus is moved away from building features and testers get fed up by having to report errors that should have been detected earlier in the process. In this talk, I will focus on how we can use Kanban in combination with solid engineering principles like ATDD and TDD can help improve the quality of the software delivered.

Scaling. Session chair: Letizia Jaccheri (NTNU)

Tech Challenges in a Large-Scale Agile Project. **Harald Søvik and Morten Forfang (Computas).** (Experience report, 30 min) [\[Link\]](#)

A five year, 25 man java project effort, that started with a waterfall-like methodology and that adopted Scrum after less than a year, has been concluded. We present three key technical challenges, briefly analyze their consequences and discuss the solutions we adopted. Firstly, we discuss how we modularized our architecture, module delineation principles, coupling and the trade-offs of abstraction. Then we discuss testing environments, their automation and relation to branches and the effect on customer involvement and feedback. Finally we discuss the benefits and disadvantages of representing domain knowledge declaratively. For all three challenges we discuss how the project's agility was affected.

Transitioning a Large Organisation: Adopting TDD. **Pdraig Brennan (Ericsson).** (Experience report, 30 min) [\[Link\]](#)

Test-Driven Development (TDD) is promoted as a powerful technique for combining software design, testing, and coding to increase reliability and productivity. However the transition to TDD is not always easy. Is it worth the effort and what can really be gained from it? This report describes a useful transition strategy based on different TDD styles and identifies some key elements required for each style. It then identifies the main differences found on the code and designs that developed using these TDD styles. The differences are striking in their consistency and provide a strong indication that TDD is well worth the effort.

Architected Agile Solutions for Software-Reliant Systems. **Barry Boehm (University of Southern California), Jo Ann Lane (University of Southern California), Supannika Koolmanojwong (University of Southern California) and Richard Turner (Stevens Institute of Technology).** (Invited talk, 30 min) [\[Link\]](#)

Systems are becoming increasingly reliant on software due to needs for rapid fielding of "70% capabilities," interoperability, net-centricity, and rapid adaptation to change. The latter need has led to increased interest in agile methods of software development, in which teams rely on shared tacit interpersonal knowledge rather than explicit documented knowledge. However, such systems often need to be scaled up to higher level of performance and assurance, requiring stronger architectural support. Several organizations have recently transformed themselves by developing successful combinations of agility and architecture that can scale to projects of up to 100 personnel. This chapter identifies a set of key principles for such architected agile solutions for software-reliant systems, provides guidance for how much architecting is enough, and illustrates the key principles with several case studies.

Code Quality. Session chair: Xiaofeng Wang (Lero)

The Limited Red Society. **Joshua Kerievsky (Industrial Logic).** (Invited talk, 40 min) [\[Link\]](#)

You've heard about limiting WIP (Work-In-Progress) but how good are you at limiting red time? Red time is when you have compilation errors and/or failing tests. A growing group of practitioners have learned how to effectively reduce red time while test-driving and refactoring code. To understand how to limit red time, it helps to visualize it. In this talk, I will analyze live programming sessions using graphs that clearly visualize red time. You'll learn what programming processes help or hurt our ability to limit red time and you'll gain an appreciation for the visual cues that can help make you a better programmer and fellow member of the Limited Red Society.

Bio: Joshua Kerievsky leads Industrial Logic, a fourteen-year-old company that guides organizations in successful agile transitions. He has more than twenty years of experience in the software field, is an expert and early pioneer in agile management and Extreme Programming, and is a prolific author of eLearning albums that help companies around the world scale agility faster. Joshua's 2004 bestselling book, *Refactoring to Patterns*, won a Jolt Cola award. His pioneering work in agile processes has helped popularize readiness assessments, story-test-driven development, and agile chartering.

Agilifying a Legacy Code Base in Two Weeks. **Håvard Stranden (Conceptos Consulting).** (Lightning talk, 10 min) [\[Link\]](#)

This lightning talk will share my experiences from an effort to create a more agile code base from legacy code with no test coverage in two weeks. I'll share what worked, what didn't work, and the importance of being pragmatic rather than dogmatic.

Introducing Agile Methods in a Large Software Development Team: The Impact on the Code. **Mary Giblin (Athlone Institute of Technology), Pdraig Brennan (Ericsson) and Chris Exton (University of Limerick).** (Research paper, 30 min) [\[Link\]](#)

The adoption of agile methods of software development has gained momentum within the software industry. NW Soft Solutions Ltd. (a pseudonym) is a large software development unit that develops large-scale network centric software solutions. NW Soft Solutions Ltd decided to adopt an agile development methodology. In this case study, we use object-oriented metrics to evaluate and characterise the source code of an application produced by a team using agile methods. We compare the results obtained from the source code produced using agile methods with the results for source code produced for a similar type of application by the same team using a more traditional methodology. The contrast is stark. This case study shows that agile methods have guided the developers to produce code that manifests better quality and maintainability characteristics.

Agile Methods may Discourage Quality. **Morten Forfang (Computas).** (Lightning talk, 10 min) [\[Link\]](#)

Agile methods, and in particular scrum, have a very strong focus on productivity and immediate business value. The developers are encouraged to complete their tasks in due time, and this may lead to quick and dirty solutions. Thus, a quality assurance system is required to ensure that technical quality is within required bounds. This system may be implemented through pair programming, code review, design review etc. It is crucial that whatever QA systems one choose, it has strong enough authority to inflict a decrease in focus factor, or even a postponed deadline.

Panel: Is Agile Research Dead in the Water?.

Is Agile Research Dead in the Water?. **Frank Maurer (Host) - University of Calgary**
participants: Pekka Abrahamson, Angela Martin, Steven Fraser, Michael Feathers, Torgeir Dingsøy, Jutta Eckstein. (Panel, 90 min) [\[Link\]](#)

The goal of the panel is to discuss if there is a point in researching agile methods and, if so, what this point is. It asks how researchers contribute to the agile community and where their impact is. Process/Mechanics. Conferences on agile methods are running for more than 10 years now. In research papers, we are observing a trend towards more sound research and deeper empirical studies. At the same time, agile methods are flourishing in industry. Nevertheless, it seems that the two realms - industry and academia - are quite separate and not well aligned. The panel will discuss where innovation is created, what research contributes to innovation, which research methods are appropriate in fast moving domains and where research impacts industrial practice.

User Experience. Session chair: Rachel Davies (Agile Experience Ltd)

Making Agile Relevant for Organizations Seeking Good Product. Jeff Patton (independent consultant). (Invited talk, 45 min) [Link]

Over the last 10 years, in organizations mired in heavy process and paralyzing analysis and design phases, we've seen agile approaches get delivery of working software moving again. But, it turns out that successful delivery doesn't result in successful products. And, for most organizations building software, it was the successful product they'd wanted all along. In this short talk Jeff Patton explores process where the primary goal is a successful product. He'll discuss the values and principles common in healthy product organizations that aren't the primary focus of agile development. You'll hear examples of innovative practice emerging from companies that subscribe to agile approaches while remaining true to their organization's values and driving towards the primary goal of delivering products people want.

Bio: Jeff Patton has designed and built software for the past 15 years on a wide variety of products from on-line aircraft parts ordering to electronic medical records. Jeff has focused on Agile approaches since working on an early Extreme Programming team in 2000. In particular Jeff has specialized in the application of user experience design practice to improve Agile requirements, planning, and ultimately the products delivered. Jeff currently works as an independent consultant, agile process coach, product design process coach, and instructor. Current articles, essays, and presentations on variety of topics in Agile product development can be found at www.AgileProductDesign.com and in Alistair Cockburn's Crystal Clear. Jeff is founder and list moderator of the agile-usability Yahoo discussion group, a columnist with StickyMinds.com and IEEE Software, a Certified Scrum Trainer, and winner of the Agile Alliance's 2007 Gordon Pask Award for contributions to Agile Development.

Design and Development in the "Agile Room": Trialing Scrum at a Digital Agency. Katerina Tzanidou (Cimex Media) and Jennifer Ferreira (The Open University). (Experience report, 30 min) [Link]

Scrum was trialed at Cimex -- a Digital Media Agency in the UK. Our insights centre in particular around the close interactions between the designers and developer working in the same room, and how the design roles were played out in the Scrum context. The lessons learned from this experience are now presented to our clients as a case study in order to make them more aware of the benefits of running an Agile project. Since this trial, we have adapted our current practice to include more immediate forms of communication. We now have hands-on experience integrating design with Scrum and can say that it was an enjoyable, bonding experience for the team.

Values and Assumptions Shaping Agile Development and User Experience Design in Practice. Jennifer Ferreira, Helen Sharp and Hugh Robinson (The Open University). (Short research paper, 15 min) [Link]

Previous discussions concerned with combining User Experience (UX) design and Agile development have either focused on integrating them as two separate processes, or on incorporating techniques from UX design into an Agile context. There is still no rigorous academic view on the nature of practice in this area. Further, the many and varied settings in which Agile developers and UX designers work together, and how those settings shape their work, remain largely unexplored. We conducted two field studies to address this. The results suggest that the values and assumptions of decision-makers external to the teams shape UX/Agile practice. The current focus on processes and techniques falls short of providing the insight necessary to improve practice. Instead, our results indicate that improvement requires further explication of contextual values.

15:00-15:30	Break and Poster Session				
15:30-17:00	Kanban	Adoption	Testing: Automation	Stories	User Interface

Kanban. Session chair: Juhani Iivari (University of Oulu)

From a timebox tangle to a more flexible flow. Jørn Ola Birkeland (Bekk). (Experience report, 30 min) [Link]

Flow-based software development (FSD, a.k.a. lean, pull-based, or kanban) is attractive in certain types of software development projects, e.g. maintenance projects. This experience report shows one project team's attempt at moving from a timebox-based development process (scrum) to a flow-based process

From Chaos to Kanban, via Scrum. Kevin Rutherford (Rutherford Software), Paul Shannon (Codeweavers), Craig Judson (Codeweavers), Neil Kidd (Codeweavers). (Experience report, 30 min) [Link]

Since late 2007 the software development teams at Codeweavers UK have been incrementally improving their ability to deliver motor finance and insurance web services. This two-year journey has taken the company from chaos to kanban-style single-piece flow, including Scrum briefly along the way. This paper charts that journey, showing the benefits gained from a simple "inspect and adapt" cycle in which the teams tackled their biggest problem at each stage.

Kanban at an Insurance Company (Are You Sure?). Olav Maassen (QNH AD&S) and Jasper Sonneveld (ASR Insurance). (Experience report, 30 min) [Link]

ASR Insurance, one of the top 3 insurance companies in the Netherlands is transitioning their IT maintenance and operations from a more traditional approach to Kanban. They started small with 1 team and slowly increased to 7 teams to gain experience. Due to the positive results they are now in the middle of transitioning 200 people to their new environment. This experience report highlights the experiences gained by the first implementing during the initial phase of the Kanban implementation. This report reflects both the practicing perspective and the coaching perspective.

Adoption. Session chair: Helen Sharp (Open University)

Exploring Defect Data, Quality and Engagement during Agile Transformation at a Large Multisite Organization. Kirsi Korhonen (Nokia Siemens Networks). (Research paper, 30 min) [Link]

Agile methods promise improvements in code quality, but getting the full benefits is claimed to take years. A lack of visibility to improvements in the early stages of agile transformation can result in motivation decrease among the people in the organization, and thus slow down the agile transformation progress. In this study we analyzed defect data in a large multisite organization during the first six months of agile transformation. Defect data was compared to the results of a survey on agile transformation experiences and perceptions, which was conducted in the organization six months after starting the agile transformation. According to the results, improvements were visible in the defect data, but less than 25% of the people in the organization felt that quality had improved. Further study revealed that a realistic perception of the positive changes in the defect data coincided with positive emotional engagement in agile transformation.

Adoption of Software Engineering Process Innovations: The Case of Agile Software Development Methodologies. Mali Senapathi (Auckland University of Technology). (Short research paper, 15 min) [Link]

Though agile methodologies have gained widespread acceptance in the past decade, there are still a number of potential adopters who are yet to join the critical mass. Some of these adopters need assurance of whether agile software methodologies will continue to be the dominant software process technology of the 2010s and beyond. This study applies the Software Engineering Process Technologies Adoption Grid proposed by Fichman & Kemerer [3] to evaluate the adoption trajectory of agile software development (ASD) methodologies. The study concludes that ASD methodologies will continue to be the dominant software process technology of the 2010's, and adopted by more business organizations.

Applying SCRUM in an OSS Development Process: an Empirical Evaluation. Luigi Lavazza, Sandro Morasca, Davide Taibi, Davide Tosi (University of Insubria). (Research paper, 30 min) [Link]

Open Source Software development often resembles Agile models. In this paper, we report about our experience in using SCRUM for the development of an Open Source Software Java tool. With this work, we aim at answering the following research questions: 1) is it possible to switch successfully to the SCRUM methodology in an ongoing Open Source Software development process? 2) is it possible to apply SCRUM when the developers are geographically distributed? 3) does SCRUM help improve the quality of the product and the productivity of the process? We answer to these questions by identifying a set of measures and by comparing the data we collected before and after the introduction of SCRUM. The results seem to show that SCRUM can be introduced and used in an ongoing geographically distributed Open Source Software process and that it helps control the development process better.

Improving Responsiveness, Bug Detection, and Delays in a Bureaucratic Setting: A Longitudinal Empirical IID Adoption Case Study. Caryna Pinheiro, Frank Maurer and Jonathan Sillito (University of Calgary). (Short research paper, 15 min) [Link]

This paper empirically studies a group of projects in a large bureaucratic government agency that adopted iterative and incremental development (IID). We found that a project that followed IID since inception provided substantially better bug-fixing responsiveness and found bugs earlier in the development lifecycle than existing projects that migrated to IID. IID practices also supported managerial decisions that lead to on-time & on-budget delivery.

Testing: Automation. Session chair: Lasse Koskela (Reaktor Innovation)

Test Automation at enterprise scale. Mark Streibeck (Google). (Invited talk, 45 min) [Link]

Agile Methods brought a (renewed) focus on testing and test automation to software development. And with it came a whole range of testing tools and technologies: continuous integration systems, mocking libraries, dependency injection frameworks... Almost all of these tools are focused on single teams or a group of teams. There are very few (if any) tools out there that work on an enterprise scale. Most larger development organizations either use these tools as islands in individual departments and/or try to scale them to their limits to reach as many as possible teams. But the requirements and opportunities of a test automation infrastructure in large organizations are not just solved by scaling up the existing systems. Over the last years, Google completely redeveloped its testing infrastructure. By using Google's cloud infrastructure, we developed systems that makes testing fast and almost transparent. Developers are able to verify their changes across Google's entire code base before submitting their changes. Furthermore, by automatically storing test results and other code metrics, we give developers more insight into their code and tests.

Bio: Mark Striebeck is an engineering manager at Google where he is responsible for developer testing infrastructure, tools and adoption. In his 20% time he works in an internal user group which tries to further the adoption of agile practices. He has been working for more than 10 years in the software industry in a variety of engineering and management positions. Since discovering XP and agile development methodologies 5 years ago, he has become actively engaged in the agile community. He constantly tries to put new ideas and agile approaches to work. The great variety of projects and individuals at Google give plenty of opportunity for this. Striebeck is a frequent speaker at Agile and other conferences. He holds two master's degrees in mathematics and computer science from the University of Hannover and Brunel University, London.

Automated Acceptance Testing of High Capacity Network Gateway. Ran Nyman, Ismo Aro, Roland Wagner (Nokia Siemens Network). (Experience report, 30 min) [Link]

In this presentation we will explore how agile acceptance testing is applied in testing high capacity network gateway. We will demonstrate how the organisation managed to grow agile acceptance testing from two co-located teams to 20+ multi-site team setup and how acceptance test driven development is applied to complex network protocol testing. We will also cover how the initial ideas that we had of agile acceptance testing evolved during product development. At the end of paper we give

recommendations to future projects using agile acceptance testing based on feedback that we have collected from our first customer trials. Testing from two co-located teams to 20+ multi-site team setup. We will also cover how the initial ideas that we had of agile acceptance testing evolved during product development. At the end of presentation we give recommendations to future projects using agile acceptance testing based on feedback that we have collected from our first customer trials

An Automated Approach for Acceptance Web Test Case Modeling and Executing. **Felipe M. Besson, Delano M. Beder and Marcos L. Chaim (University of Sao Paulo).** (Short research paper, 15 min) [\[Link\]](#)

This paper proposes an approach for modeling and executing acceptance web test cases and describes a suite of tools to support it. The main objective is to assist the use of Acceptance Test-Driven Development (ATDD) in web applications by providing mechanisms to support customer-developer communication and by helping test case creation. Initially, the set of web pages and relations (links) associated with a user story is modeled. Functional test possibilities involving these relations are automatically summarized in a graph, being each path of the graph a user story testing scenario. Once a testing scenario is accepted by the customer, a testing script is automatically created. A web testing framework then executes the script, triggering the ATDD process.

Stories. Session chair: Patrick Kua (Thoughtworks)

Who killed the digital nomad?. **Kjetil Moløkken-Østvold (Conceptos Consulting).** (Lightning talk, 10 min) [\[Link\]](#)

In the late nineties, a paradigm shift facilitated by enhanced technology was said to liberate individual workers. Powerful laptops, ubiquitous high-speed wireless connections, modern work-patterns and the war for talent all seemed to create a new modern work-force. The rise of the digital nomad was imminent. Yet, a few years later, the digital nomad is dead. This is especially apparent in the software industry, the very industry that helped create the digital nomad. Who actually killed the digital nomad? This entertaining and thought provocative talk will provide you with the answer.

Lessons from a Software Engineering Dojo: The MSE at Carnegie Mellon University. **Michael Keeling (Carnegie Mellon University).** (Lightning talk, 10 min) [\[Link\]](#)

The Software Craftsmanship movement advocates that the best way to learn how to create software is to practice under a knowledgeable master. For 20 years, the Master of Software Engineering (MSE) program at Carnegie Mellon University has done just that. At the center of the MSE curriculum is the studio project, a mentored, long term (12-16 month duration) "live lab" where students apply classroom lessons while building software for a real (often commercial) client. The MSE is a great example of how mentoring and the apprenticeship model can be practically applied in industry.

A scrum master's struggles in agile estimation and reporting. **Terje Brasethvik (Bouvet).** (Lightning talk, 10 min) [\[Link\]](#)

Project sponsors or managers like the idea of high performing agile teams and bureaucratic, Åless development processes, but they are rather less agile when exercising budget limits, monitoring cost and delivery dates. However, with the backlog being the most flexible element of the project, Åi what happens to delivery dates and budgets? Developers enjoy breaking down their own tasks and getting the final say in how many hours of hard labour a task requires. However, with the budget burning down faster than the backlog and with the managers watching, how agile are they likely to be?

Technical Debt is Good. **Olve Maudal (TANDBERG).** (Lightning talk, 10 min) [\[Link\]](#)

With borrowed money you can do something sooner than you might otherwise. If you are willing to incur technical debt you can release a product earlier than you might otherwise.

Security Extensions for Agile Methods. **Torstein Nicolaysen (NTNU).** (Lightning talk, 10 min) [\[Link\]](#)

A disturbing trend is showing. More and more projects are done using Agile methods, and more and more software is connected to the Internet. This increases the risk of attack. No Agile method has techniques or practices for tackling software security activities in an Agile fashion, but somehow software security needs to be prioritized like any other software quality attribute. A suggestion on how to mitigate the problems encountered were a part of a student specialization project. The presentation contains the essence of this research as well as a walk-through of the new techniques.

Dealing with Navigation and Interaction Requirement Changes in a TDD-Based Web Engineering Approach. **Juan Burella (Universidad de Buenos Aires), Gustavo Rossi (UNLP), Esteban Robles Luna (UNLP) and Julián Grigera (UNLP).** (Short research paper, 15 min) [\[Link\]](#)

Web applications are well suited to be developed with agile methods. However, as they tend to change too fast, special care must be put in change management, both to satisfy customers and reduce developers load. In this paper we discuss how we deal with navigation and interaction requirements changes in a novel test-driven development approach for Web applications. We present it by indicating how it resembles and differs from "conventional" TDD, and showing how changes can be treated as "first class" objects, allowing us to automate the application changes and also to adaptively prune the test suite.

An Ongoing Story about Agile Adoption. **Tommi Juhola (Lindorff Finland Oy).** (Lightning talk, 10 min) [\[Link\]](#)

This is a story about one major project and the people contributing to it. The means of adoption, the high amount of communication and the feelings involved in adoption will be presented. The main offering of this talk are the experiences, instructions and small hints. These will help you on your own journey.

Kano-weighting: Prioritizing your user stories. **Sergey Dmitriev (Making Waves).** (Lightning talk, 10 min) [\[Link\]](#)

This talk will give you a short introduction to a combination technique to prioritize user stories that I named kano-weighting. I have been using it for a couple of years and want to share my experiences. It combines two often used techniques for story prioritization: kano analysis and story weighting in a single light-weighted approach.

User Interface. Session chair: Sridhar Nerur (University of Texas)

Agile Interaction Design and Test-Driven Development of User Interfaces - A Literature Review. **Theodore D. Hellmann, Ali Hosseini-Khayat and Frank Maurer (University of Calgary).** (Invited talk, 30 min) [\[Link\]](#)

This chapter describes the development of GUI-based applications, from usability engineering and prototyping to acceptance test-driven development, in an agile context. An overview of current agile interaction design practices will be presented, including a thorough analysis of the current role of prototyping and current attempts to facilitate test-driven development of GUI systems, as presented in academic and industrial literature. Traditional usability engineering approaches show that if user input is taken into consideration early in the development process by repeatedly conducting usability tests on low-fidelity prototypes of the GUI system, the final version of the GUI will be both more usable and less likely to require revision. The major risk associated with test-driven development of GUIs is the high likelihood of change in the target GUI, which can make test development unnecessarily expensive and time consuming. A unification of these styles of development will be presented, along with a prediction of how this process can be used to simplify creating test-able GUI-based applications by agile teams.

Launchpad's Quest For A Better And Agile User Interface. **Martin Albisetti (Canonical).** (Experience report, 30 min) [\[Link\]](#)

Introducing change into an established team with an aging code base is always a challenge. This paper is about my experience in shifting the Launchpad teams' focus to a more user-friendly mindset, evolving the existing processes to encourage constant improvements in the user interface and teaching developers to see their work from the users' perspective. We went from nobody owning the user interface to the developers feeling strong ownership over it, and I describe the ups and downs we went through as a team, what worked, what didn't, and what we could have done better.

Making GUI Testing Productive and Agile. **Emily Bache (eLabs).** (Lightning talk, 10 min) [\[Link\]](#)

Automated testing through the GUI has a bad reputation, and many people recommend that tests should instead go via an API just underneath it. But is that just because today's tools are not up to it? The heart of the problem is one of coupling - Tests that use the GUI break when it changes, and it tends to change often. PyUseCase, and tools like it, insert a layer of abstraction between your tests and the GUI, and enables them to be written in a high level domain language. I'd like to demonstrate how and why this can make testing through the GUI both productive and agile.

Usability testing in Agile Land. **Eli Toftoy-Andersen (Steria).** (Lightning talk, 10 min) [\[Link\]](#)

Agile methodology, test driven develop and beautiful code is not enough. The UX police demands mandatory usability testing in agile project to ensure better user experience.

UI Prototyping in Agile Software Projects. **Jonas Folless (Cappgemini).** (Lightning talk, 10 min) [\[Link\]](#)

We have several techniques to become more agile when writing code. We apply design patterns and use practices like pair programming and test driven development. But how can we get more agile in the way we design the user experience of our applications? In this lightning talk I will give an introduction to UI prototyping and how it can help your team become more agile in the way you work with user interface design. (This session was given in Norwegian during the Smidig2009 conference in Oslo: <http://smidig2009.no/talks/111>)

18:30-23:30

Conference banquet and keynote by Bjørn Alterhaug and John Pål Inderberg: **Improvisation: Between Panic and Boredom. Perspectives on teamwork, dialogue and presence in music and other contexts**

Conference banquet and keynote by Bjørn Alterhaug and John Pål Inderberg: **Improvisation: Between Panic and Boredom. Perspectives on teamwork, dialogue and presence in music and other contexts**

Improvisation: Between Panic and Boredom. Perspectives on teamwork, dialogue and presence in music and other contexts - Bjørn Alterhaug (NTNU) and John Pål Inderberg (NTNU). (Keynote,) [\[Link\]](#)

Improvisation is facilitating good communication. We see this clearly in the interaction among musicians, but other professions might learn something from the fundamental jazz mind-set: creating together, taking risks, and not being overly concerned about making mistakes. People are generally not very aware of improvisation, and improvisation receives little or no attention in education. The one-sided focus on language-based knowledge and rationality often overshadows important emotional and aesthetic aspects of human interaction. Today, technological innovations, pressure of time, detailed administrative procedures, and increasingly rigid management systems often get in the way of direct dialogue and interaction between professionals and ordinary people. In all skill-based professions, e.g. psychiatrists, physicians, pilots, musicians, etc., people perform better when deep, internalised knowledge is put into action through active dialogue and interaction: improvisation. The two key components of any meeting are the ability to listen and to be creative together with other people. Models from improvised music may contribute to a new understanding of the interactive aspects of different communities of practice.

Thursday, 3 June 2010: Main Conference

08:30-10:00 Keynote by David Anderson: Catalyzing Lean - Building a Limited WIP Society in Your Organization

Keynote by David Anderson: Catalyzing Lean - Building a Limited WIP Society in Your Organization.

Catalyzing Lean - Building a Limited WIP Society in your Organization. **David Anderson.** (Keynote, 60 min) [\[Link\]](#)

For at least a decade the agile community has understood that there was value in Lean Thinking and that methods such as Extreme Programming could be interpreted and explained using Lean concepts. However, institutionalized adoption of Lean, with its culture of continuous improvement ("kaizen"), has been slow. There is no recognized Lean software development lifecycle or project management methodology in common use today. The Lean idea of flow is widely accepted but the tools and techniques required to manage and optimize it have gained little traction, until recently. The introduction of kanban systems that visualize work and specifically limit the quantity of work-in-progress (WIP) have generated a significant improvement in the adoption of Lean concepts, techniques and tools. This growth is rapid and global and echoes the growth rate of XP almost a decade earlier. Unlike XP adoption is happening in a wide range of businesses from conservative Northern European insurance firms that failed to embrace agile methods, to startups in emerging markets such as Cambodia. This talk, citing real examples from around the World, will look at why and how choosing to limit WIP can create a cultural evolution within your organization that will lead to a leaner future.

David J. Anderson, leads a management consulting firm focused on improving performance of technology companies. He has been in software development for 26 years and has managed teams on agile software development projects at Sprint, Motorola, Microsoft and Corbis. David is credited with the first implementation of a kanban process for software development in 2005. David was a founder of the agile movement through his involvement in the creation of Feature Driven Development. He was also a founder of the APLN, a founding signatory of the Declaration of Interdependence, and a founding member of the Lean Software and Systems Consortium. He moderates several online communities for lean/agile development. He is the author of the book "Agile Management for Software Engineering - Applying the Theory of Constraints for Business Results". Most recently, David has been focused on creating a synergy of the CMMI model for organizational maturity with Agile and Lean methods through projects with Microsoft and the SEI. He is based in Seattle, Washington

10:00-10:30 Break and Poster Session

10:30-12:00 Panel: Collaboration in an Agile World

Distributed	Refactoring	Personal Development	Product owner
-------------	-------------	----------------------	---------------

Panel: Collaboration in an Agile World.

Collaboration in an Agile World. **Panel Impresario: Steven Fraser, Director - Cisco Research Center.**

Panelists: Bjørn Alterhaug, Professor, Norwegian University of Science and Technology, David Anderson, President David J. Anderson & Associates Management Consulting for Knowledge Workers, Diana Larsen, Partner FutureWorks Consulting, Scott Page, Leonid Hurwicz Collegiate Professor of Complex Systems, Political Science, and Economics, University of Michigan Ann Arbor. (Panel, 90 min) [\[Link\]](#)

Collaboration, the art of working together on a joint intellectual effort, is an essential part of system and product development. Collaboration is generally learned on the job rather than by academic training. This panel will review challenges, including: organization, geography, culture (norms, beliefs, shared vocabulary), and communication (both verbal and non-verbal) - and the mechanisms now available to ameliorate. This panel will bring together a set of experts to discuss, debate and propose collaboration strategies for the 21st century.

Distributed. Session chair: Patrick Kua (Thoughtworks)

Distributed Meetings in Distributed Teams. **Marc Bless (WEINMANN medical technology).** (Experience report, 30 min) [\[Link\]](#)

Emails and phone calls are insufficient and most inefficient for distributed communication and meetings. It is impossible to let a team improve by these means. This experience report presents how we dealt with the lack of communication in a distributed Scrum Team in three locations and two different time zones. We created a virtually collocated team by using video conferencing systems even for daily meetings and remote desktop systems for handling documents and tools. With such systems all necessary meetings and practices can be executed. It is even possible to support highly collaborative sessions like planning poker, pair programming, and retrospectives. The necessary budget to set up such systems is outweighed by the obtained effectiveness and efficiency of team communication.

Energy project story, from waterfall to distributed Agile. **Tomáš Tureček, Roman Smířák, Tomáš Malík, Petr Boháček (Tieto Czech).** (Experience report, 30 min) [\[Link\]](#)

Our team helps Tieto teams distributed over the entire world to set up effective way of working applying Agile, Lean, Kanban and Global sourcing principles. In May 2009 we have faced our biggest challenge. Energy sector product, 30 people in Norway, Sweden and Finland with waterfall way of working organized into regular 6 sub-applications wanted to achieve quite challenging targets. To grow to 70 people by setting up another teams in Czech Republic, to transfer 15 years legacy system knowledge and to be up and running with this all in 6 months while maintaining the production for the clients at full speed. Paper describes how we fought and won over the challenges by basing the service transfer on Agile.

Geographically Distributed Team - a Success Story. **Morten Moen (Powel).** (Lightning talk, 10 min) [\[Link\]](#)

Learn from the experience of a team that has been working geographically distributed for over a year. Learn the importance of solving communication challenges using good practices and an including work atmosphere. Discover freely available web tools that will help you solve most of the daily scrum tasks.

Lessons Learnt working with a Remote Scrum Team in Bangalore. **Ole Dahlström (Projectplace International).** (Lightning talk, 10 min) [\[Link\]](#)

In this session I will talk about the challenges and opportunities we see working with a remote team in Bangalore, India. Scrum and XP are ideal when working with a remote team, and my talk will be about what we have learnt running a development department offshore.

Architecture-Centric Development in Globally Distributed Projects. **Joachim Sauer (C1 WPS GmbH).** (Lightning talk, 10 min) [\[Link\]](#)

In this talk architecture-centric development is proposed as a means to strengthen the cohesion of distributed teams and to tackle challenges due to geographical and temporal distances and the clash of different cultures. A shared software architecture serves as blueprint for all activities in the development process and ties them together. Architecture-centric development thus provides a plan for task allocation, facilitates the cooperation of globally distributed developers, and enables continuous integration reaching across distributed teams.

Refactoring. Session chair: Johannes Brodwall (Steria)

Code should be Improved, not Thrown Away! **Karianne Berg (Objectware).** (Lightning talk, 10 min) [\[Link\]](#)

Tangled architecture. Unreadable code. Lack of tests and documentation. These are just some of the factors that makes it hard to maintain and add new features to old code bases. Is the best solution to this to throw away all the old code and start over? I think this rarely is the case, and I will tell you why in this presentation.

Old code should be laid to rest, not refactored to death. **Ole-Marius Moe-Helgesen (Itverket).** (Lightning talk, 10 min) [\[Link\]](#)

In our world of agile development, continuous improvement and refactoring we've been taught how to deal with legacy code: Refactor to support testing, write a test suite and THEN start to change it. The other option is what we did before: throw away the old and start again. The problem is that we can't release it until we've completed all the features of the old system. There is a third option though, and that is to combine agile practices with a strategy of replacing all the old code. I'll share some experiences doing both and explain why I think we should let old code die a peaceful death.

The Effects of Gravity on Software Development. **Craig Judson (Codeweavers).** (Lightning talk, 10 min) [\[Link\]](#)

From infancy, we use metaphors to make sense of abstract concepts. The metaphors we choose have huge cultural significance, and bring with them both a rich vocabulary for discussing the target concept, and a bag of related concepts. And if we are not careful, we can sometimes believe those related concepts to be true of the real situation, forgetting we are in the land of the metaphor. So it is with some architectural patterns -- notably those that make use of layers. This session will shine a light on some assumptions we make about software when we forget we are in the land of metaphors.

Extreme Product Line Engineering - Refactoring for Variability: A Test-Driven Approach. **Yaser Ghanam and Frank Maurer (University of Calgary).** (Research paper, 30 min) [\[Link\]](#)

Software product lines - families of similar but not identical software products - need to address the issue of feature variability. That is, a single feature might require various implementations for different customers. Also, features might need optional extensions that are needed by some but not all products. Software product line engineering manages variability by conducting a thorough domain analysis upfront during the planning phases. However, upfront, heavyweight planning approaches are not well-aligned with the values of minimalistic practices like XP where bottom-up, incremental development is common. In this paper, we introduce a bottom-up, test-driven approach to introduce variability to systems by reactively refactoring existing code. We support our approach with an eclipse plug-in to automate the refactoring process. We evaluate our approach by a case study to determine the feasibility and practicality of the approach.

Extending Refactoring Guidelines to Perform Client and Test Code Adaptation. **Wafa Basit, Fakhar Lodhi and Usman Bhatti (National University of computer and emerging sciences, Pakistan).** (Research paper, 30 min) [\[Link\]](#)

Refactoring is a disciplined process of applying structural transformations in the code such that the program is improved in terms of quality and its external behavior is preserved. Refactoring includes evaluation of its preconditions, execution of its mechanics and corrective actions required to retain the behavior of the program. These transformations affect various locations throughout a program which includes its clients and unit tests. Due to the complex dependencies involved within the program, preservation of program behavior often becomes nontrivial. The guidelines on refactoring by Fowler lack precision and leave opportunities for developers to err. In this paper, we analyze and present an exhaustive categorization of refactoring guidelines based on their impact on production and test code together. In addition, we present extended refactoring guidelines that adapt the clients and unit tests to keep it syntactically and semantically aligned with the refactored code.

Personal Development. Session chair: Lars Arne Skår (Miles)

To Run Fast you Need to Be in Balance! **Thommy Bommen (Computas).** (Lightning talk, 10 min) [\[Link\]](#)

In our world today it is important to get things done and be fast about it. This often lead to people running too fast while being off balance.

I will in this talk inspire you to grab hold of your own life and get in balance. I will show examples where it hurts and you as a developer would be off balance if you do not act. I will make use of tai-chi to exemplify my experience and invite you to see a minute of being in balance.

This talk is about you!

Great Scrum Masters. Geir Amsj  (Spitia). (Lightning talk, 10 min) [\[Link\]](#)

More than 60.000 persons are Certified Scrum Masters, but there still is a long way to go before this role is well understood. I have met some frustrated Scrum Masters recently, and I strongly believe it is vital for Agile success that this role is better understood both by management, the team and the Scrum Masters themselves. In this talk I will share my thoughts and experiences from working with several organisations trying to adopt Scrum.

Being an Agile Manager. Kristin Wulff (Kantega). (Lightning talk, 10 min) [\[Link\]](#)

I am the manager of a group consisting of two scrum teams. This last year we have improved ourselves. We have managed to turn a chaotic workday with a lot of stupid errors, into a quiet workday where the tasks are solved in a good manner. This has resulted in better ROI, and more pleased customers and employees. So what are the next steps? It seems that the quiet workday is too quiet. There is too little inspiration, pressure and motivation to do an excellent job. Is this due to the nature of the tasks, the process or my leadership? This next six months I will experiment to find out.

Transaction Analysis – An aid for improving personal interaction in agile teams?. Finn Prytz (Dnv). (Lightning talk, 10 min) [\[Link\]](#)

A brief presentation of the basic Transaction Analysis concepts of ego-states and games, traditional areas of application, and a warning of oversimplifying a complex theory from a 10 minutes speech. A few attempts of simple Transaction Analysis to typical interactions between agile team members. How can Transaction Analysis help team members create more powerful teams from insights in their own personal behavior and communication style.

Being able to Code does not make you a Good Developer! . Rune Sundling (Objectware). (Lightning talk, 10 min) [\[Link\]](#)

Developers need to know how to create and maintain code, use an analytical approach to problems, and be aware of a range of technologies, patterns and best practices. But a good developer needs to focus on a lot more than that! I will take a closer look at why personal responsibility, cooperation, communication and personal traits are important. The lightning talk will focus on knowledge easily implemented in practice, not hard to follow abstract terms.

Using Code Review to Increase Quality and Strengthen the Team. Eirik Wahl (Powel). (Lightning talk, 10 min) [\[Link\]](#)

The talk will discuss how a team can conduct regular code reviews to increase the code quality, share knowledge, and minimize "code ownership". A practical, time-saving tool (and how to use it correctly) will be presented: <http://www.atlassian.com/software/crucible/> (The speaker is not in any way affiliated with Atlassian.) This 10 minutes will get you started with code review as a standard end-of-sprint activity, and save you time and money!

Change Requires Drive-Driven Personality. Marc Bless (WEINMANN medical technology). (Lightning talk, 10 min) [\[Link\]](#)

Change requires people with the ability to drive that change. Many people have to play Scrum Master, Product Owner, and other agile roles. They do their very best and yet keep standing. They inspect but do not adapt. These people do not have the required driving skill as they miss one major inner force: drive-driven personality. Drive-Driven Personality is needed to push changing the world!

Product owner. Session chair: Anders Haugeto (Iterate)

An Ideal Customer – A Grounded Theory of Requirements Elicitation, Communication and Acceptance on Agile Projects. Angela Martin (University of Waikato), Robert Biddle (Carleton University) and James Noble (Victoria University of Wellington). (Invited talk, 30 min) [\[Link\]](#)

This chapter explores the reality of the customer role – a critical, complex, and demanding role on agile teams. Despite initial difficulties, customers love agile development and would not do it any other way, but they also encountered many difficulties in their day-to-day work. In this chapter we describe the practices that have emerged to ensure the role works effectively and sustainably, and how the role has evolved from an individual to a team. We hope customers will find this chapter helpful in performing their role, and programmers will find it useful to understand the complexities of customer's role on the project.

Representing the Product Owner. Ingunn Moen (Kantega). (Lightning talk, 10 min) [\[Link\]](#)

When the customer is not at the site, the Product Owner of the team is but a proxy that tries to translate the customers needs to detailed functionality. To make sure the customer is happy, the Product Owner needs to discuss solutions, and needs to get questions answered quickly on behalf of the developing team. We have collected and tried out various ways to bridge the gap between the team and the remote customer, who is the real Product Owner. This talk gives some examples on what went well, and what can be done better.

The Product Backlog hinders Value Creation. Trond Wing rd (Steria). (Lightning talk, 10 min) [\[Link\]](#)

The Product Backlog is considered a cornerstone of agile software development. However, the Product Backlog can impede and sometimes even prevent good problem understanding, creativity, agility and ultimately spoil our attempts at creating something valuable. In this lightning talk we'll view the reasons for this deficiency, and sketch out some alternatives.

The top six Technical Practices every Product Owner must know about. Mikael Boman (Citerus). (Lightning talk, 10 min) [\[Link\]](#)

As a Product Owner, you need to understand the value of the technical practices your development team do use to understand the value of spending time improving how the team uses them. Knowing more about this will also enable you to demand that they start using the technical practices you deem most valuable to your product. In this talk, I will present the most important software development practices that you as a Product Owner should know about, with focus on the value they bring and not the specifics on how they are implemented.

Product and Release Planning Practices for Extreme Programming. Gert van Valkenhoef (University of Groningen), Tommi Tervonen(University of Groningen), Bert de Brock (University of Groningen) and Douwe Postmus (University Medical Center Groningen). (Short research paper, 15 min) [\[Link\]](#)

Extreme Programming (XP) is an agile software development methodology defined through a set of practices and values. Although the value of XP is well-established through various real-life case studies, it lacks practices for project management. In order to enable XP for larger projects, we provide the rolling forecast practice to support product planning, and an optimization model to assist in release planning. We briefly evaluate the new practices with a real-life case study.

12:00-13:30	Lunch				
13:30-15:00	Culture and Organization	Collaboration	Testing: New approaches	Learning and Improvement	Developer Interaction

Culture and Organization. Session chair: Rachel Davies (Agile Experience Ltd)

Diagnosing Organizational Culture: Can Agile work in My Company?. Diana Larsen (FutureWorks Consulting). (Invited talk, 45 min) [\[Link\]](#)

Choosing your approach to a transition to Agile software development hinges more on your organization's culture than on industry, marketplace, economy, or type of product. But how can you tell whether your culture will support adopting Agile? Will it succeed at your company? Will it provide the results you want? In this talk, Diana Larsen will describe one way to diagnose your organizational culture, determine how you'll measure success, and, working with culture, identify the next steps to getting the results you want. Bio: Diana Larsen chairs the Agile Alliance board of directors. As a principal consultant at FutureWorks Consulting LLC (<http://futureworksconsulting.com>), she sparks workplaces where productive, resilient teams focus on frequent delivery of high value software. Diana co-authored Agile Retrospectives: Making Good Teams Great! and co-founded the annual Agile Open Northwest conference.

Organizational Culture and the Deployment of Agile Methods: The Competing Values Model View. Juhani Iivari and Netta Iivari (University of Oulu). (Invited talk, 30 min) [\[Link\]](#)

A number of researchers have identified organizational culture as a factor that potentially affects the deployment of agile systems development methods. Inspired by the study of Iivari and Huisman (2007), which focused on the deployment of traditional systems development methods, the present paper proposes a number of hypotheses about the influence of organizational culture on the deployment of agile methods.

A Journey towards Structured Chaos - Destination: Google. Susan Hunter (Google). (Lightning talk, 10 min) [\[Link\]](#)

A presentation that compares and contrasts 3 unique experiences across 3 distinct organizations implementing Agile. The presentation will illustrate the constant variables of context, culture and collaboration and how they interrelate when implementing agile. From a large traditional financial company in Dallas: Nurturing a newly formed collaborative team in a competitive environment to a competitive hierarchical advertising company in New York: Navigating land-mines of anti-scrum sentiment; to Google: Giving structure to chaos with the freedom to create options.

Collaboration. Session chair: Angela Martin (University of Waikato)

Three 'C's of agile practice: collaboration, co-ordination and communication. Helen Sharp and Hugh Robinson (The Open University). (Invited talk, 30 min) [\[Link\]](#)

The importance of collaboration, co-ordination and communication in agile teams is often discussed and rarely disputed. These activities are supported through various practices including pairing, customer collaboration, stand-ups and the planning game. However the mechanisms used to support these activities are sometimes more difficult to pin down. We have been studying agile teams for over a decade, and have found that story cards and the Wall are central to an agile team's activity, and the information they hold and convey is crucial for supporting the team's collaboration and co-ordination activity. However the information captured by these usually physical artefacts pertains mainly to progress rather than to functional dependencies. This latter information is fundamental to any software development, and in a non-agile environment is usually contained in detailed documentation not generally produced in an agile team. Instead, this information resides in their communication and social practices. In this chapter we discuss these three 'C's of agile development and what we know about how they are supported through story cards and the Wall.

Agile Undercover: When Customers Don't Collaborate. Rashina Hoda, James Noble and Stuart Marshall (Victoria University of Wellington). (Research paper, 30 min) [\[Link\]](#)

Customer collaboration is vital to agile projects. Through a Grounded Theory study of New Zealand and Indian Agile teams we discovered that lack of customer involvement was causing problems in gathering and clarifying requirements, loss of productivity, and business loss. "Agile Undercover" allows development teams to practice Agile despite insufficient or ineffective customer involvement. We present the causes and consequences of lack of customer involvement on Agile projects and describe the Agile Undercover strategies used to overcome them.

Using Team Radar for Assessing and Improving Agile Teamwork. Sigurd Ringbakken (Acando). (Lightning talk, 10 min) [\[Link\]](#)

Presentation of experience with the use of a team radar as an instrument for improving agile teams. The instrument and the key findings will be presented.

"Condi" - a tool to include the customer into the project. **Steinar Line (Kantega)**. (Lightning talk, 10 min) [\[Link\]](#)

Kantega has for more than a decade developed web applications using Open Aksess, a open source CMS, for customers with differering needs. The general rule is that customers have strong opinions of design and functionality. Also customers tend to always be very busy. The recent year Kantega has developed Condi, a tool allowing for easy deployment of latest builds of a web application out to a prototype server. By allowing the customer access to this server and using functionality in Condi to deploy the latest build, the customer may inspect what is developed so far and give instant feedback.

Testing: New approaches. Session chair: Johannes Brodwall (Steria)

Generate Characterization Tests for Legacy Code from existing Integration Tests. **Jonas Follesø (Cappgemini)**. (Lightning talk, 10 min) [\[Link\]](#)

A characterization test is a means to describe (characterize) the actual behaviour of an existing piece of software, and therefore protect existing behaviour of legacy code against unintended changes via automated testing. In this talk I will share my experience from a project where we automatically generated fast running characterization tests based on existing slow running integration tests. The characterization tests helped us do a major redesign of a core calculation module in an insurance system without breaking existing functionality.

The Safety Net of Functional Web Testing. **Ole Gunnar Borstad (Cappgemini)**. (Lightning talk, 10 min) [\[Link\]](#)

Code maintainability is an important enabler for agile software development. Working with legacy code is challenging because its maintainability suffers from high coupling and lack of unit tests. By introducing automated functional testing, we can create a safety net that will detect any breaking changes to existing functionality. This allows developers to be more confident. We will demonstrate a test framework that allows us to specify application behaviour by programmatically interacting with the user interface of a web application.

Prove it and Ship it!. **Anders Sveen (Cappgemini)**. (Lightning talk, 10 min) [\[Link\]](#)

Even though many projects are doing what the call agile; standups, backlogs and some automated testing, they are not reaping the full benefits. Proving you can adapt to circumstances by shipping new versions rapidly will be the true test. And it doesn't matter if you're doing agile or not: Getting to that point will expose problems in your process and help you achieve your goal of delivering quality software that solves real problems for real customers. Prove that you're doing it right and ship it!

Security Testing in Agile Web Application Development - A Case Study Using the EAST Methodology. **Gencer Erdogan (CERN), Per Håkon Meland (Sintef) and Derek Mathieson (CERN)**. (Research paper, 30 min) [\[Link\]](#)

There is a need for improved security testing methodologies specialized for Web applications and their agile development environment. The number of web application vulnerabilities is drastically increasing, while security testing tends to be given a low priority. In this paper, we analyze and compare Agile Security Testing with two other common methodologies for Web application security testing, and then present an extension of this methodology. We present a case study showing how our Extended Agile Security Testing (EAST) performs compared to a more ad hoc approach used within an organization. Our working hypothesis is that the detection of vulnerabilities in Web applications will be significantly more efficient when using a structured security testing methodology specialized for Web applications, compared to existing ad hoc ways of performing security tests. Our results show a clear indication that our hypothesis is on the right track.

Continuous Selective Testing. **Bastian Steinert, Michael Haupt, Robert Krahn and Robert Hirschfeld (University of Potsdam)**. (Research paper, 30 min) [\[Link\]](#)

A manual and explicit activity, the frequent selection and execution of tests requires considerable discipline. Our approach automatically derives a subset of tests based on actual modifications to the code base at hand, then continuously executes them transparently in the background, and so supports developers in instantly assessing the effect of their coding activities with respect to the overall set of unit tests to be passed. We apply techniques of selective regression testing, mainly relying on dynamic analysis. By taking advantage of the internal program representation available in IDEs, we do not need to rely on expensive comparisons of different program versions to detect modified code entities.

Learning and Improvement. Session chair: Tore Dybå (SINTEF)

Put it to the Test: Using Lightweight Experiments to Improve Team Processes. **Michael Keeling (Institute for Software Research Carnegie Mellon University)**. (Experience report, 30 min) [\[Link\]](#)

Experimentation is one way to gain insight into how processes perform for a team, but industry teams rarely do experiments, fearing that such educational excursions will incur extra costs and cause schedule overruns. When facing a stalemate concerning the use of pair programming one industry-like, academic team constructing a commercial-grade web application, performed a lightweight experiment comparing pair programming and programming alone using Fagan inspection. Through the experiment, the team learned that pair programming was not only faster than programming alone, but also required less effort and produced code of more predictable quality. Conducting the experiment required only eight hours of effort over six weeks (a mere 0.5% of the total effort during that time frame) and afforded crucial information for choosing the best practices for the team. As demonstrated by this experience, lightweight experimentation is cost effective and does not threaten project schedules.

Feedback and Retrospectives in Agile Projects. **Rolf Knutsen (Know IT Objectnet)**. (Lightning talk, 10 min) [\[Link\]](#)

Measuring use of retrospectives in teams. Why you should measure this and our findings? We found that teams are reluctant to perform retrospectives - why is that? How we motivate our projects to do retrospectives? How we help our projects to do retrospectives.

Tough Projects and the Kubler Ross Grief Cycle. **Mark Needham (ThoughtWorks)**. (Lightning talk, 10 min) [\[Link\]](#)

As Agile practitioners there will always be projects where key decisions have been made before we arrive and while we may disagree with them we still have to deliver. The way people react in this situation seems to be quite similar to the type of reaction we have when confronted by grief. We will examine a particular client case study where key decisions on both architecture and team composition were made before the team arrived and compare the reaction of team members with the Kubler Ross grief cycle. We will also look at some ways that we can handle this type of situation more effectively.

What agile teams can learn from rescue dogs. **Olav Maassen (QNH)**. (Lightning talk, 10 min) [\[Link\]](#)

Whenever there's a disaster, earthquake, mud slide and people get lost, rescue dogs are summoned to help and find these people. Within 24/48 hours rescue dogs from all over the world are operating in that area. Timeliness is really a matter of life and death. People from different parts of the world, who haven't worked together before have to cooperate quickly and effectively with each other. This talk will provide a quick view into how the International Rescuedog Organization tries to solve this problems and what I think we in agile teams can learn from this approach.

Adoption of Team Estimation in a Specialist Organizational Environment. **Tor Erlend Fægri (Sintef)**. (Research paper, 30 min) [\[Link\]](#)

Specialist organizational environments and lack of redundant knowledge reduce flexibility and therefore inhibits transition to agile development. This action research reports from the adoption of team estimation as a vehicle to increase redundant knowledge within a group of specialists. The group suffered from low levels of group learning legitimized by high work pressure and a specialist organizational environment. This resulted in poor planning and optimistic task estimates which contributed to increase the work pressure even higher. I framed the research as double-loop learning; I illustrate how different barriers to team estimation arose from conflicts with existing efficiency norms and then how benefits from team estimation created sufficient momentum to change practice. The results are obtained from qualitative analysis of empirical data gathered during one year of collaboration with the group. The article contributes to understanding of barriers to group learning and agile adoption in software organizations.

Developer Interaction. Session chair: Anders Haugeto (Iterate)

Is the grass greener for wiki users?. **Svein-Magnus Sørensen (IteraConsulting)**. (Lightning talk, 10 min) [\[Link\]](#)

The city administration of Oslo, Norway, has recently introduced the Atlassian Confluence enterprise wiki to be a collaborative multi-tool for addressing several of their challenges with internal project management and organizing information. In light of this effort this talk will consider some of the benefits of taking a structured approach when adopting such grass-roots knowledge management tools, and how Confluence in particular is well suited to this type of tasks due to its powerful plugin-framework.

Towards Understanding Communication Structure in Pair Programming. **Kai Stapel, Eric Knauss, Kurt Schneider and Matthias Becker (Leibniz Universität Hannover)**. (Research paper, 30 min) [\[Link\]](#)

Pair Programming has often been reported to be beneficial in software projects. To better understand where these benefits come from we evaluate the aspect of intra-pair communication. Under the assumption that the benefits stem from the information being exchanged, it is important to analyze the types of information being communicated. Based on the Goal Question Metric method we derive a set of relevant metrics and apply them in an extreme Programming class room project. Data covering a total of 22.9 hours of intra-pair communication was collected. We found that only 7% of the conversations were off-topic (e.g. private), 11% about requirements, 14% about design, and 68% about implementation details (e.g. syntax). Accordingly, a great share of the information being exchanged in Pair Programming is on a low level of abstraction. These results represent a first data point on what kind of information is communicated to what extent during Pair Programming.

Continuous Open Space. **Iija Preuß (disy Informationssysteme GmbH)**. (Lightning talk, 10 min) [\[Link\]](#)

Two basic principles of Agile organizations are self-organization and cross-functional collaboration. Open Space has a strong focus on self-organization and invitation, but in its original form is limited to special events of a few days length. At the end of the first OS at disy, the question came up: "how can we integrate this way of collaboration into our daily work?" I'd like to present our experiences with trying to use open space principles and techniques in our daily work for almost two years now - what did we do, what did work, what problems did we encounter, what did we learn?

Top five Problems you face as an Agile Developer, and how to Solve them. **Pål Fossmo (Acando)**. (Lightning talk, 10 min) [\[Link\]](#)

In all development projects, you run into problems and boring stuff. How can you solve the problems, and in the process make them more fun? In this lightning talk we will touch subjects like why using a debugger is wrong, how to help your team mates be more passionate and the importance of having a plan.

Agile team success: Multiple specialities or overlapping competencies?. **Terje Brasethvik (Bouvet)**. (Lightning talk, 10 min) [\[Link\]](#)

The talk presents challenges in orchestrating a development team with complementary skills and competencies - all of which very relevant and required for the project at hand. However, according to the team assessment instrument presented in [1], a multidisciplinary team with the seemingly perfect and required diversity in competences, is not guaranteed a high score. [1] Moe, Dingsoyr, Royrvik, "Putting Agile Teamwork to the Test - A Preliminary Instrument for Empirically Assessing and Improving Agile Software Development", in Abrahamsson, Marchesi & Maurer (Eds.): "XP 2009", LNBP 31, 2009

15:00-15:30	Break			
15:30-17:00	My Agile Suitcase	Knowledge and Innovation	Testing: TDD	Adoption

My Agile Suitcase. Session chair: Martin Heider (infomar software) and Bernd Schiffer (it-agile GmbH)

My Agile Suitcase. **Martin Heider (infomar software) and Bernd Schiffer (it-agile GmbH)**. (Pecha Kucha, 90 min) [\[Link\]](#)

Imagine you are an agile consultant or coach. You are called by the inhabitants from waterfall island, who haven't heard about agility before and want to benefit from your advice. Which practices, principles and values would you pack in your agile suitcase for providing them guidance? What would you leave at home? In this Pecha Kucha session experienced agilists will deliver insights in their agile suitcases. Their short, concise and entertaining presentations may give you hints for your own suitcase. So fasten your seatbelts and enjoy your travel with:

- Rachel Davies
- Mary Poppendieck
- Joshua Kerievsky
- Jeff Patton
- Patrick Kua

After listening to your flight attendants you will have the opportunity to work on your own agile suitcases. This may help you to travel lighter next time.

Knowledge and Innovation. Session chair: Helen Sharp (Open University)

Project Management Knowledge Areas in Agile Projects. **Rolf Knutsen (Know IT Objectnet)**. (Lightning talk, 10 min) [\[Link\]](#)

PMI, and others, have defined knowledge areas for running projects. Some of these knowledge areas are covered by agile processes, whereas other knowledge areas are not. Are these knowledge areas applicable for agile processes? How should we handle them in agile projects? This lightning talk will aim to answer these questions, and go into a bit more details regarding risk handling.

Travelling Light ... with Powerful Questions. **Deborah Hartmann Preuss**. (Lightning talk, 10 min) [\[Link\]](#)

We're busy: we rush and look for "efficiencies," short-cutting conversation by embedding answers in our questions, ex: "Will that be ready on time?" The smart people we've hired quickly learn that we don't want their opinions complicating our timelines. We unintentionally micro-manage while morale and innovation fade. *What if we learned to frame powerful questions and invite real dialogue, sharing the load of thought-leadership? What if we chose to dig for root causes?* Get ready to run your own tiny coaching experiments, guaranteed to show you unseen patterns and "aha!" moments.

Using the Wisdom of Crowds to Spark Innovation. **Mike Sutton (Wizewerx)**. (Lightning talk, 10 min) [\[Link\]](#)

Innovation is the holy grail of all commercial organisations (whether they realise it or not). Yet large enterprises are hugely unaware of the hidden potential of their large workforce in sparking innovation through the systematic and controlled use of the collective wisdom of crowds. I want to share my thinking about this and what conditions need to be in place. We will have a 5 minute audience involved experiment around this talk.

Beyond Estimation. **Ole Christian Rynning (Bekk)**. (Lightning talk, 10 min) [\[Link\]](#)

We estimate offers, budgets, systems, backlogs, queues, products, features, dependencies, releases and iterations. When are estimations useful, and more important: when is it wasteful? Is the only useful aspect of estimation planning? My wish is to renew the debate on estimation, and suggest an alternative path.

Monkey Business (Loss aversion). **Haakon Spilde (Know IT Objectnet)**. (Lightning talk, 10 min) [\[Link\]](#)

In this lightning talk I will discuss loss aversion. Loss aversion refers to the tendency for people to strongly prefer avoiding losses rather than acquiring gains. I will give some examples from studies done with moneys, before I share a few ideas on how this knowledge can be used for your benefit in Agile projects.

Future Research in Agile Systems Development: Applying Open Innovation Principles within the Agile Organization. **Kieran Conboy (National University of Ireland, Galway) and Lorraine Morgan (University of Limerick)**. (Invited talk, 30 min) [\[Link\]](#)

A particular strength of agile approaches is that they move away from 'inverted' development and intimately involve the customer in all areas of development, supposedly leading to the development of a more innovative and hence more valuable information system. However, we argue that a single customer representative is too narrow a focus to adopt and that involvement of stakeholders beyond the software development itself is still often quite weak and in some cases non-existent. In response, we argue that current thinking re-garding innovation in agile development needs to be extended to include multiple stakeholders outside the business unit. This paper explores the intra-organisational applicability and implications of open innovation in agile systems development. Additionally, it argues for a different perspective of project management that includes collaboration and knowledge-sharing with other business units, customers, partners, and other relevant stakeholders pertinent to the business success of an organisation, thus embracing open innovation principles.

Testing: TDD. Session chair: Lasse Koskela (Reaktor Innovation)

A Literature Review on Story Test Driven Development. **Shelly Park and Frank Maurer (University of Calgary)**. (Short research paper, 15 min) [\[Link\]](#)

This paper presents a literature review on story-test driven development. Our findings suggest that there are many lessons learned papers that provide anecdotal evidence about the benefits and issues related to the story test driven development. We categorized these findings into seven themes: cost, time, people, code design, testing tools, what to test and test automation. We analyzed research papers on story test driven development to find out how many of these anecdotal findings were critically examined by researchers and analyzed the gaps in between. The analysis can be used by researchers as a ground for further empirical investigation.

The case against TDD fundamentalism. **Christian Lundestad and Arnulf Krokeide (Conformit)**. (Lightning talk, 10 min) [\[Link\]](#)

In this lightning talk we argue against what we see as 'TDD fundamentalism'. We posit that practices such as "do the simplest thing that can possibly work" and "deal with one requirement at the time" are wasteful, risky and do not encourage innovative, competitive solutions. Instead we argue that adding a small up-front design phase will allow us to arrive faster at a better solution. We advocate introducing stakeholder values and product qualities as explicit parts of the design and introduce some design heuristics that we find are sensible alternatives to more fundamentalist approaches.

So you think you can test?. **Eirik Bjørnsnøs (Kantega)**. (Lightning talk, 10 min) [\[Link\]](#)

Unit testing is a widely adopted agile practice. But can we ensure that tests are executed, kept up to date or even written? This talk presents a concept web framework which integrates unit testing in a novel way. Based on the premiss that untested code can't be trusted, the it will refuse running your code if it does not have tests, the tests don't cover all your code, or your tests fail to detect bugs artificially injected into your code. By integrating unit testing, coverage analysis and mutation testing, the framework aims to help developers write higher quality software

Top five tricks for testing tricky Java code. **Johannes Brodwall (Steria)**. (Lightning talk, 10 min) [\[Link\]](#)

When testing real world applications, there are some things that are harder to test than others. But so far, I've found that all aspects of my programming can be driven by tests. In this talk, I go through my top five tips to improve JUnit tests: * Testing web applications in-process with Jetty and WebDriver * Testing ORM-code with HSQLDb * Testing business logic with Mockito * Making your JUnit tests more readable with FEST-Assert * A surprise ending! Come to this lightning talk if you want to see some simple ways to improve your JUnit tests.

Experiences Introducing TDD with BDD Concepts to a Legacy Codebase. **Anders Hammervold (Bekk)**. (Lightning talk, 10 min) [\[Link\]](#)

This lightning talk describes the experiences with introducing TDD and BDD-concepts to legacy code-bases. The examples will range from unit-test-styles as a means of introducing the concept of BDD, and touch upon BDD using Cucumber through the wire-protocol to .Net. This will lead up to what benefits and effects this had on the development of the code-base, with comments from a skeptic, a convert, and a believer.

A Quantitative Comparison of Test-first and Test-last Code in an Industrial Project. **Burak Turhan (University of Oulu), Ayse Bener (Bogazici University), Pasi Kuvaja (University of Oulu) and Markku Oivo (University of Oulu)**. (Short research paper, 15 min) [\[Link\]](#)

This paper presents a comparison of test-first and test-last development approaches on a customer account management software of a telecommunication company. While the management had plans for initiating a process that enforces test-first development over test-last, they also had concerns about the tradeoffs. Therefore, an exploratory case study with quantitative analysis is carried out on a pilot project where the code metrics and estimated manual inspection efforts of both approaches are compared. Our results indicate no statistical difference between the two approaches in terms of design (CK) metrics. On the other hand, we observe that test-last development yields significantly simpler code in terms of cyclomatic complexity and less (though not significant) manual inspection effort. Hence, our initial results indicate no superiority of test-first over test-last development in the described industrial context.

Culture and Organization. Session chair: Juhani Iivari (University of Oulu)

Agile IT strategy using Balanced Scorecard. **Simen Fure Jørgensen (Iterate)**. (Lightning talk, 10 min) [\[Link\]](#)

Business executives are an important part of agile, as it is their job to define what value is for a particular organization. They point out the overall business strategy, and business value is simply something that contributes to this. However, these leaders are too busy to play the part of the product owner. Even if they were not that occupied, it doesn't seem like a good use of management time. By using Balanced Scorecard to define and follow up the IT strategy, the business executive can say "what" without saying "how", get continuous feedback and adjust the course when necessary.

Planning in Scrumproject. **Anna-Lena Bergman (Powel)**. (Lightning talk, 10 min) [\[Link\]](#)

I will do a presentation in how we do scrum planning with a common product backlog. I will talk about impediments that we meet in planning for example the difficulty to do long term planning in a product when working in scrumprojects.

Are you ready for Change?. **Anders Sveen (Capgemini)**. (Lightning talk, 10 min) [\[Link\]](#)

At the core of Agile lies the ability to adapt and change as you learn more about the system and problem domain. And even though many are doing the ceremonies of Agile they fail to acknowledge what it really means to be ready for change. In several projects I have seen things that is constantly blocking our efforts to become agile. Doing the ceremony might make you feel good, but it doesn't mean you are reaping the full benefits of Agile. In this lightning talk I will take you through some of the common mistakes I have seen, and hopefully prevent you from doing them too.

Stealth Scrum: 3 years after. **Thomas Almnes (CIBER Norway)**. (Lightning talk, 10 min) [\[Link\]](#)

Three years after sneaking in Agile methodology into one of several software development departments in the largest bank in Norway, we ask our selves if it was worth the effort?

What started out as a stealth approach has become "the way we do projects". But not without a struggle. How do you navigate an Agile project in a waterfall-centric organization? What kind of roles and persons do you need to succeed? What is not important and what are the things you ought not to leave out? With lots of experience to show to, this talk will discuss the organizational impact when adopting Agile methodology.

More Metaphors in 10 minutes. **Ken Power (Cisco Systems)**. (Lightning talk, 10 min) [\[Link\]](#)

Here are some metaphors that I find useful when describing software product development. In particular, they are useful for describing the basic principles of agile development and the lifecycle of a typical agile project. Learn why software development is like releasing a music album, producing a magazine, staging a theatre production, or producing a film. I will present some justification for each metaphor, and describe how and why I find it useful. I will describe their usefulness as metaphors for software development in general, and for agile development in particular.

Dad, Dad, Are we there yet? **Chris Matts (Emergent Behaviour)**. (Lightning talk, 10 min) [\[Link\]](#)

Much of the Agile ThoughtLeadership is focusing on getting Agile "Across the Chasm". Listening to these leaders, you would be forgiven for thinking Agile has solved all the problems. Agile practitioners know that we have a long way to go. There are a lot of problems that have not been solved. I start with a brief but inflammatory history and current assessment of Agile. This was very popular with practitioners. I then present about 10 areas where Agile is far from complete and where we need to focus more effort to find solutions.

So You Think You're Agile? **Colm O'hEocha (National university of Ireland Galway), Kieran Conboy (National University of Ireland, Galway), Xiaofeng Wang (University of Limerick)**. (Experience report, 30 min) [\[Link\]](#)

Some agile projects succeed, some fail miserably. Research shows that time does not necessarily cure such ills and there can be many complex underlying reasons. Evaluating the ways agility is supported across three supposedly agile projects reveals a myriad of organizational, human and political issues. Using a novel approach to assess agile projects from first principles, this paper outlines several key findings and recommendations beyond mere compliance to textbook methods.

Adoption. Session chair: Lars Arne Skår (Miles)

How to develop an agile developer. **Anders Haugeto**. (Lightning talk, 10 min) [\[Link\]](#)

Companies have success with agile development because of their culture. Consciously (or unconsciously), senior developers build culture continuously when teaching their co-workers about technology, process and business goals. What are they like, the senior developers that manage to guide their co-workers in the every day routine towards agility?

Flight of the Agile. **Thomas Nilsson (Responsive)**. (Lightning talk, 10 min) [\[Link\]](#)

Agile is about adapting. Adapting is about learning. Learning is about maturity. Traditional maturity models like CMMI tells us that there are different things you should focus on depending on the maturity of your organisation. This talk will present a different viewpoint on team and organisational maturity, and in particular how it can help a team or an organisation that is adopting Agile. Maybe it's not about what we do, but how we do them?

Agile Documents: Toward Successful Creation of Effective Documentation. **Mazni Omar (Universiti Utara Malaysia), Sharifah Lailee Syed Abdullah (Universiti Teknologi MARA) and Azman Yasin (Universiti Utara Malaysia)**. (Short research paper, 15 min) [\[Link\]](#)

This paper presents the initial findings from an action research study in applying Extreme Programming (XP) activities and generating agile documents. The study was carried out in a university computer centre in Malaysia. Data were collected from four software engineering (SE) teams in this centre. Simple and practical agile documents were successfully created during the software project and received positive feedbacks amongst team members and the centre management. The results of this study suggest that a right combination of personality types in a team can influence team performance. These findings offer valuable insights into aspects related to agile approach and documentation that can be explored and understood to suit an organizational culture and team personality types.

Distributed requirement handling. **Kjetil Moløkken-Østvold (Conceptos Consulting)**. (Lightning talk, 10 min) [\[Link\]](#)

Adopting some agile practices, like stand-up meetings and yellow notes, is easy. The hard part is handling requirements and estimates, especially in a distributed environment. Describing, refining, communicating and prioritizing requirements is difficult for business developers. Estimating and understanding requirements is difficult for developers. This talk describes a flexible sharing regime, implemented in a distributed in-house development environment, which succeeded in improving requirement handling and estimation.

Key concepts from Lean to bring agility to the enterprise. **Jørn Øimheim (Statoil)**. (Lightning talk, 10 min) [\[Link\]](#)

Ideas from Lean is in my opinion crucial in order to successfully introduce agility to an enterprise. Scrum and XP will work for individual projects and can scale to pretty large development organizations, but when you try to involve the entire enterprise, Lean concepts has to be part of the implementation. I will go through the key idea from Lean that I think is crucial for enterprise agility.

Anatomy of a ScrumBut. **Øyvind Kvangardsnes (Bekk)**. (Lightning talk, 10 min) [\[Link\]](#)

ScrumBut is a slightly derogatory term used for describing projects which are using Scrum but skips practice X or Y. Given the derogatory nature of the term, it is interesting to note that most Scrum projects fits this description of ScrumBut quite well. This lightning talk aims to highlight some common characteristics of a ScrumBut. Another aim is to debunk the myth that ScrumButs should be considered harmful. Adopting Scrum is quite challenging, and striving to do it by-the-book can be counter productive. Remember that Scrum is only a way to improve your process.

Introducing Agile Methods in a Large Software Development Team: The Developers Changing Perspective. **Mary Giblin (Athlone Institute of Technology), Pdraig Brennan (Ericsson) and Chris Exton (University of Limerick)**. (Short research paper, 15 min) [\[Link\]](#)

NW Soft Solutions Ltd. (a pseudonym) is a large software development unit that develops large-scale network centric software solutions. NW Soft Solutions Ltd decided to adopt an agile development methodology. Martin Fowler in his article "The New Methodology" [1], states that in his opinion "Since agile methods are so fundamentally people-oriented, it's essential that you start with a team that wants to try and work in an agile way". Using NW Soft Solutions as a case study, this paper sets out to show how the developers attitudes towards agile methods change during the adoption phase. We see a shift in focus from agile practices at a superficial level to the core values that underpin agile methods.

An agile tragedy - the agile practitioner visits the liquor store. **Thor Henning Hetland (Webstep and Cantara)**. (Lightning talk, 10 min) [\[Link\]](#)

A critical experiment to see how far different Agile "techniques" will get you in a simple real-world experiment.
 The though experiment * Observe how typical Agile development stereotypes handle the same task * Task-description * Context
 The contenders * The freshman agile practitioner * The Kanban practitioner * The peer-programming team * The agile code-nazi / code quality geek *
 The "hot-shot wannabee * The pomodoro enthusiast * The value-stream agile practitioner * "Kent Beck"

17:00-19:30	Free food buffet in the sponsor area at the conference hotel
18:00-19:00	Open Space Opening

Open Space Opening.

Open Space Opening. **Charlie Pool and Diana Larsen**. (Open Space, 1 hour) [\[Link\]](#)

Welcome to Open Space at XP2010! Open Space's self-directed learning environment will help you integrate and extend the information and experiences you've gained during the rest of XP2010. Open Space provides an opportunity to meet in self-organizing groups to share your latest ideas, challenges, questions, hopes, experiences and experiments. The Open Space format gives just enough structure to foster collaboration and take direction from those who choose to participate. Make this conference your own. Commit to attending the opening on Thursday evening, offer your ideas about important topics to explore, and attend sessions Thursday night and/or Friday.

The opening session:

During the opening session, your hosts will explain the Four Principles of and One Law of Open Space. If you have questions, volunteers and hosts will offer answers.

The Four Principles:

- The right people show up
- It starts when it starts
- It ends when it ends
- People in context determine outcomes (Whatever happens is the only thing that could have.)

The Law of Personal Mobility

If you find yourself sitting where you can't learn or contribute, move yourself to a place where you can. We ask that everyone attending any sessions at Open Space XP2010 show up for the opening. It's important. You'll listen to the presentations and take note of those you want to attend. As you listen, you may be inspired to present a session yourself. There are no deadlines or time limits: simply step up and describe your idea for a session. Your commitment to arriving at the beginning on Thursday and staying until the end on Friday will ensure we build on conversation after conversation.

[Read more](#)

19:00-19:30	Break
19:30-24:00	Open Space

Friday, 4 June 2010: Workshop / Tutorial Day

08:30-10:00	Open Space	Emily Bache Test Driven Development:	Ola Ellnestam, Tobias Anderberg A semi clinic discourse	André Dhondt Speak like a	Jutta Eckstein Facilitating Virtual	Bernd Schiffer, Marko Schulz Can you Kata?	Martin Gijzen Business oriented keyword-driven testing	Gojko Adzic Building software that matters	Nancy Van Schoenderwoert Transient State: From
-------------	------------	--	---	-------------------------------------	---	--	--	--	--

		Performing Art	about opinions, intentions and pathologies in groups	Native	Retrospectives		for testers and developers		Funding to First Iteration for Embedded Systems
10:30-12:00	Open Space	Test Driven Development: Performing Art	Mike Sutton, Olaf Lewitz Embrace Diversity: Techniques to make our differences work for us	Giorgio Natili Agile practices Vs Flash Platform developers	Facilitating Virtual Retrospectives	Can you Kata?	Business oriented keyword-driven testing for testers and developers	Building software that matters	Jørn Ola Birkeland, Unni Nyhamar Hinkel Kick-starting flow-based (lean) software development
13:00-14:30	Open Space	Hugo Corbucci Coding Dojo - Kake format	Marc Bless Tailoring Agile and Regulated Environments	Luca Grulla, Brian Blignaut The secret art of Agile Javascripting	Jenny Wong, Danilo Sato Inception Workshop: Kickstarting an Agile project	Ken Power, Kieran Conboy Enabling Open Innovation through Agile Development	Sallyann Freudenberg, Rachel Davies Trust and Culture Workshop	Cory Foy Serious Play: Product Planning and Prioritization Through Innovation Games	Darja Smite Starting up, Managing and Working with Agility in Distribution
15:00-16:30	Open Space	Coding Dojo - Kake format	Tailoring Agile and Regulated Environments	Patrick Kua Building the Testing Pipeline	Inception Workshop: Kickstarting an Agile project	Enabling Open Innovation through Agile Development	Markus Gärtner Testing Dojos	Martin Heider, Jens Coldewey Sharpen your saw - How to improve as agile coach	Starting up, Managing and Working with Agility in Distribution

Detailed overview of morning sessions. Detailed overview of afternoon sessions.