



keynotes • tutorials • talks • openspace

pocket GUIDE

University of Porto Faculty of Engineering, 25-26 June | www.agilept.org

pocket **GUIDE**

Program Overview

Day I

Friday, 25 June

08:15 - 08:40	Registration	@ I-105
08:40 - 09:00	Conference Opening	@ I-105
09:00 - 10:30	Keynote. <i>"Improving the Quality and Productivity of Backlogs Through Envisioning"</i>	@ B002
10:30 - 11:00	Coffee Break	@ I-105
11:00 - 12:30	Talks Tutorial Open Space	@ B002 @ B003 @ I-105
12:30 - 13:30	Lunch	~ B002
13:30 - 15:00	Talks Tutorial Open Space	@ B002 @ B003 @ I-105
15:00 - 17:15	World Cup 2010. Portugal - Brazil	@ I-105
17:15 - 18:45	Talks Tutorial Open Space	@ B002 @ B003 @ I-105
18:45 - 19:00	Retrospective	@ I-105
19:00 - 22:30	Social Event	

Program Overview

Day 2

Saturday, 26 June		
08:30 - 09:00	Registration	@ I-105
09:00 - 10:30	Keynote. <i>"Big Ball of Mud: Is This the Best that Agile can Do?"</i>	@ B002
10:30 - 11:00	Coffee Break	@ I-105
11:00 - 12:30	Talks	@ B002
	Tutorial	@ B003
	Open Space	@ I-105
12:30 - 14:00	Lunch	~ B002
14:00 - 15:30	Keynote. <i>"Agile Software Development in the Large"</i>	@ B002
15:35 - 17:05	Talks	@ B002
	Tutorial	@ B003
	Open Space	@ I-105
17:05 - 17:30	Coffee Break	@ I-105
17:30 - 18:30	Panel. <i>"Beyond Agile"</i>	@ B002
18:30 - 19:00	Closing Session	@ I-105

Silver Sponsor



Bronze Sponsors

Microsoft®

ThoughtWorks®

Friend Sponsors



Supporters



Table of Contents

Welcome to Agile Portugal 2010	7
Keynotes.....	9
Talks	17
Tutorials.....	29
Open Space.....	35
Retrospective	36
Panel.....	37
Social Events	39
Notes	42
Internet Connectivity.....	52
Useful Information.....	53



Welcome to Agile Portugal 2010

Welcome to the first Portuguese Conference on Agile Methodologies, which is proudly hosted by the Faculty of Engineering from University of Porto (FEUP).

Agile Portugal 2010 will gather international professionals, writers, researchers and users to share knowledge and experiences on many topics related to agile methodologies and agile philosophy. Rather than focusing on a single methodology, it will be a forum for all Agile development approaches.

On the end of the first day, the Social Event will take the conference out into the streetscape, as participants travel along the city of Porto, becoming acquainted with its culture. Casual and spontaneous discussions will allow the users to share their experiences, concerns and challenges with both novices and experts.

We hope these type of activities will reflect the inner nature of Agile Methodologies and its thriving community.

We wish you a good conference!

Ademar Aguiar | Joseph Yoder | Nuno Flores
Filipe Figueiredo Correia | Hugo Sereno Ferreira



Keynotes

Three inspiring keynotes from our invited speakers will be strong points of the conference. They are aimed to provide us three different perspectives and insights that can challenge our thinking and how we decide, our interests, and our passions.

Keynote: “Improving the Quality and Productivity of Backlogs Through Envisioning”, Dave Thomas

Friday, 25 June, 9:00 - 10:30 @ Main Conference Room (B002)

Good backlogs are essential to the flow of work through agile teams. Hence they have a clear impact on productivity and quality. While simple in principle, we find that backlogs are often of highly variable quality. Lumpy backlogs stress the team resulting in poor estimates, more rework and poorer quality. Bloated and lumpy backlogs place excessive demands on product owners and business analysts making it hard for them to keep up with the development team. In large organizations, there is often a tendency to use independent backlogs to manage maximize portfolios of features while paying lip service to any dependencies and common architecture. This situation becomes even more problematic when product owners, business analysts, developers, and testers are remotely distributed from each other.

In this talk, we describe how to create and maintain high quality backlogs. We begin by defining the properties of a good backlog. Then we discuss the practices used dur-

ing Envisioning by integrated product teams of Product Analysts, Owners, Designers, Customers, Architects, Developers and Testers. We explain how Envisioning facilitates portfolio management, improves acceptance testing, reduces business and technical risk; and integrates the voice of the customer into the backlog. We show how envisioning practices help bridge the gap between the developer, designer, analyst, and architect roles in the organization.

Dave Thomas has a wide spectrum of experience in the software industry as an engineer, professor, consultant, architect, executive and investor. Dave is founder and CEO of Bedarra Corporation; which provides virtual CTO and CEO, business mentoring and seed investment to emerging companies. Recently formed Bedarra Research Labs undertakes speculative research on applications of emerging software technologies.

He has many years of experience in structured documents including the design of laser printer controllers, early commercial applications of Tex. He has advised on the IBM B2B strategy, and is on the MS Customer Advisory Council and with OLL contributed to the SCORM elearning standard, and authoring tools. He is Chairman of Xia Systems, Online-Learning.com (OLL), and a director of Stilo/Omnimark, Bitflash, Amikanow and Synop and several other software companies.

Dave is best known as the founder and past CEO and president of Object Technology International Inc. (formerly OTI, now IBM OTI Labs) and led the commercial

introduction of object and component technology. The company is often cited as the ideal model of a software technology company.

He was also the principal visionary and architect for IBM VisualAge Smalltalk and Java tools and virtual machines including the initial work on popular multi-language Eclipse.org IDE. OTI pioneered the use of virtual machines in embedded systems with Tektronix shipping the first commercial products in 1988. He was instrumental in the establishment of IBM's Pervasive computing efforts and in particular the Java tooling.

Dave is an adjunct research professor at Carleton University, and the University Of Queensland and is widely published in the software engineering literature. He is a popular humorous albeit opinionated keynote speaker. Dave remains active in various roles within the technical community including ECOOP, AOSD, Evolve, and Agile Development Conference, Agile/XP Universe and OOPSLA Onward. He is a founding director of the Agile Alliance and most recently a founder of Open Augment Consortium. Dave writes expert columns in Otland Online in Germany, and the Journal Of Object Technology in Switzerland where he also serves on the editorial board.

Keynote: “Big Ball of Mud: Is This the Best that Agile can Do?”, Joseph Yoder

Saturday, 26 June, 9:00 - 10:30 @ Main Conference Room (B002)

It was back in '97 when Brian Foote and I first opined that: while much attention had been focused on high-level software architectural patterns, what is, in effect, the de-facto standard software architecture had seldom been discussed: the Big Ball of Mud. A Big Ball of Mud is haphazardly structured, sprawling, sloppy, duct-tape and bailing wire, spaghetti code jungle. We've all seen them. These systems show unmistakable signs of unregulated growth, and repeated, expedient repair. Information is shared promiscuously among distant elements of the system, often to the point where nearly all the important information becomes global or duplicated. The overall structure of the system may never have been well defined. If it was, it may have eroded beyond recognition. Programmers with a shred of architectural sensibility shun these quagmires. Only those who are unconcerned about architecture, and, perhaps, are comfortable with the inertia of the day-to-day chore of patching the holes in these failing dikes, are content to work on such systems. Somewhat to our astonishment, since our original statement, no one has ever undertaken to dispute this premise. Still, this approach endures and thrives. Why is this architecture so popular? Is it as bad as it seems, or might it serve as a way-station on the road to more enduring, elegant artifacts? What forces drive good programmers to build ugly systems? Can we avoid this? Should we? How can we make such systems better?

This keynote will examine the paradoxes that underlie Big Balls of Mud, what causes them, and why they are so prominent. What Agile Practices help us avoid or cope with mud? Does Agile practices such as TDD really help minimize mud? What are we doing RIGHT? What Agile Practices contribute to the problem? Encourage mud? Is Mud really the best that Agile can do? Is Agility's utilitarian focus on process rather than design its secret weapon, or its Achilles heel?

Joseph Yoder is a founder and principle of The Refactory, Inc., a company focused on software architecture, design, implementation, consulting and mentoring on all facets of software development. Joseph has been working in the software industry since the 80's and is an international speaker and pattern author and long standing member of The Hillside Group, a group dedicated to improving the quality of software development. Joseph specializes in Object-Oriented Analysis and Design, C#, Java, Smalltalk, Patterns, Agile Methods, Adaptable Systems, Refactoring, Reuse, and Frameworks. He has mentored developers on many types of software applications. Joe is the author of many patterns including being the co-author of the Big Ball of Mud pattern, which illuminates many fallacies in the approach to software architecture. Joseph has chaired the Pattern Languages of Programming Conference (PLoP), as well as presented tutorials and talks at conferences such as AGILE, ECOOP, JAOO, OOPSLA, PLoP, and QCON.

Joe currently resides in Urbana, Illinois. He teaches Agile Methods, Design Patterns, Object Design, Refactoring,

and Testing in industrial settings and mentors developers on these concepts. He currently oversees a team of developers who have constructed many system based on enterprise architecture using the .NET environment. Other projects involve working in both the Java and .NET environments deploying Domain-Specific Languages for clients. Joe thinks software is still too hard to change. He wants do something about this and believes that with good patterns and by putting the ability to change software into the hands of the people with the knowledge to change it seems to be on promising avenue to solve this problem.

Keynote: “Agile Software Development in the Large”, Jutta Eckstein

Saturday, 26 June, 14:00 - 15:30 @ Main Conference Room (B002)

A lot of people still believe that agile software development is for small teams only. However, the agile value system and the principles behind as stated in the agile manifesto don't say anything about team or project size. Furthermore the projects I'm working on are typically large, distributed and mission-critical. Therefore, several years ago I took the challenge and tried agile software development in the large. Meanwhile I made the similar experience on many large projects: Also large and even distributed teams can benefit from a value system that is beneficial for small teams.

In this keynote I want to show how to scale agile processes to teams of 300. In fact, the same techniques are

also relevant to teams of ten or more developers, especially within large organizations.

Jutta Eckstein, a partner of IT communication, is an independent coach, consultant and trainer from Braunschweig, Germany. Her know-how in agile processes is based on over ten years experience in developing object-oriented applications. She has helped many teams and organizations all over the world to make the transition to an agile approach. She has a unique experience in applying agile processes within medium-sized to large distributed mission-critical projects. This is also the topic of her books “Agile Software Development in the Large” and “Agile Software Development with Distributed Teams”.

Besides engineering software she has been designing and teaching OT courses in industry. Having completed a course of teacher training and led many ‘train the trainer’ programs in industry, she focuses also on techniques which help teach OT and is a main lead in the pedagogical patterns project. She has presented work in her main areas at ACCU (UK), JAOO (Denmark), OOPSLA (USA), SD West, SD Best Practices (both USA), XP (Europe) and Agile (USA).



Talks

Talks are 20min presentations (+10min questions), about exciting ideas, research projects, experience reports, or lessons learned, from experienced speakers on the relevant topics for the audience of the conference.

Talk: “Agile for BAU”, Lachlan Heasman

Friday, 25 June, 11:00 - 12:30 @ Main Conference Room (B002)

This is an experience report of a life insurance team moving to agile delivery. I'll tell you about what was tried, what failed, what was tried instead. It's the story of a transition from a directed and managed support group to a self-leading, self-governing and self-correcting team. Sharing the messages from the team, the customers and the management, we will focus on people, process, experience and outcomes.

Talk: “Creating Great Agile Teams”, Mário Araújo

Friday, 25 June, 11:00 - 12:30 @ Main Conference Room (B002)

During the last 4 years I had the opportunity to take part in several agile projects. Throughout this period my role ranged from Team Member, Scrum Master and Product Owner. I worked with co-located teams and with teams distributed around the world. I dealt with junior and senior team members, with customers that were agile fans and with skeptics.

And there's still so much to learn...

In this talk I'll share a framework for coaching agile teams to success as well as simple yet pragmatic tips that will help any Team Member, Scrum Master or Product Owner be a better player, have more fun and deliver better products.

Talk: "The Role of Tools and Platforms in Agile Development", Lúcio Ferrão

Friday, 25 June, 11:00 - 12:30 @ Main Conference Room (B002)

"Individuals and interactions over processes and tools" is the first rule of the Agile Manifesto. This simple rule is so obvious that it shouldn't need to be stated. Unfortunately software development keeps getting more complex and more distant from business users.

OutSystems tries to make a difference in the market stressing simplicity and customer collaboration to compete in the enterprise custom software development.

See for yourself the next generation of software development tools in order to understand how you can again focus back on the individuals and interactions instead of the tools and their complexities.

Talk: “Behave Yourself! — Taming the wild, wild multiple system integration using RSpec, Cucumber & Friends”, José Bonnet

Friday, 25 June, 13:30 - 15:00 @ Main Conference Room (B002)

PT Inovação has decided to completely review its Business Support Systems’ offer (BSS), by following the most up to date references in the area. This implied re-architecting a mostly monolithic solution into several systems that now can live on their own, each following its own evolution and commercial path. But this apparently simple decision left us with a bunch of systems to integrate, some of them in scenarios where this integration has to be done with systems from other vendors, all of this being done in typically very short time-frame.

We have therefore decided to start exploring Behavior Driven Development (BDD) techniques for multi-vendor multi-system integration testing, where we can define business valuable scenarios that cross more than one system and clearly show what’s working and what’s not. Furthermore, being a complex and repetitive task, this integration testing strongly benefits from BDD’s inherent automation background.

This work is still in the exploratory phase, and we’re eager to hear from the Agile community if this approach has been tried elsewhere and with what sort of results.

Talk: “Automating Interaction Testing with UML Sequence Diagrams — Where TDD and UML meet”, João Pascoal Faria

Friday, 25 June, 13:30 - 15:00 @ Main Conference Room (B002)

The development of computer-based UML design models for documentation only is time consuming and the result is often wrong and soon becomes outdated. But if the UML models are used also as a basis for automatic code generation and/or test generation, then the time invested can be recovered, the quality of the models can be checked and improved, and there is a good chance that they are kept up-to-date.

In this talk it will be presented an approach and supporting tools for constructing executable UML design models in an agile way. The structural/static aspects of the system are modeled through class diagrams, from which class skeletons are automatically generated using already existing tools. The novelty of the approach has to do with the dynamic modeling. The dynamic behavior of the system is incrementally specified/ modeled through sequence diagrams that act simultaneously as test scenarios. From these scenarios, unit tests (in fact, interaction tests) are automatically generated. The test code generated not only checks the results returned, but also checks that internal interactions among objects in the system occur as specified in the sequence diagrams.

To demonstrate and support the approach, it was developed a plug-in for Enterprise Architect (a leading UML modeling tool) that generates JUnit tests from test sce-

narios specified as UML sequence diagrams. It was also developed a test helper framework in AspectJ that traces the actual interactions among objects that occur at run-time, and checks if they are a supertree of the interactions specified in the sequence diagram (and coded into the test code).

This approach can be used iteratively as in test-driven development, achieving a true integration between agile modeling and TDD.

Talk: “Test-Driven Development with Visual Studio 2010”, Nuno Silva

Friday, 25 June, 13:30 - 15:00 @ Main Conference Room (B002)

This session takes a look at the features provided in Visual Studio 2010 to facilitate a test-driven development approach, as well as the integration of unit tests as part of an ALM process, taking advantage of Team Foundation Server to quickly find and fix bugs. We also take a brief look into PEX, an extension to Visual Studio made by Microsoft Research, that generates parameterized unit tests based on the code you’ve written.

Talk: “Patterns, Wikis and Agility: Where They Came From & Where They’re Going”, Ademar Aguiar

Friday, 25 June, 17:15 - 18:45 @ Main Conference Room (B002)

Although it may at first seem that software patterns, wikis and agile are completely unrelated, the fact is that they have very strong roots in common. In this talk, we

will discover together the most important factors they share, by revisiting their roots, and by observing how they have adapted along the way until today. Based on their trends, we will glaze at the crystal ball and see the direction they are all heading to.

Talk: “Collaborative Development Environments: Grasping the collective knowledge”, Nuno Flores

Friday, 25 June, 17:15 - 18:45 @ Main Conference Room (B002)

With the advent of the Web 2.0 and the evolution of the communication media, the software development teams are becoming more and more distributed. The development paradigm of the whole team in one physical space is fading with time. So far, collaborative development environments have been concerned with suitable presentation, seamless integration of tools and reliability of results. Only recently did the need for supporting the social aspects of software development became an issue. There was a need to bring the outside [the development environment] communication into the development environment, as a means to enable the needed collaboration between team members. Now, there is a real opportunity to harness knowledge that, otherwise would be kept in people's heads and that would usually have a lifespan of a few minutes during a conversation.

**Talk: “What makes a good information radiator?”,
Lachlan Heasman**

Friday, 25 June, 17:15 - 18:45 @ Main Conference Room (B002)

What makes a good information radiator? What questions kind of questions should it answer and what questions should it raise? There is more to it than sticking cards on a wall.

Talk: “Scrum with Visual Studio”, Tiago Silva

Saturday, 26 June, 11:00 - 12:30 @ Main Conference Room (B002)

As ferramentas de alta produtividade que tenham em conta as necessidades reais dos programadores, são cada vez mais procuradas. O Scrum framework, em conjunto com a utilização adequada do Visual Studio Team System 2010, são claramente a melhor combinação que um Scrum Developer pode usar. A sessão é destinada a programadores que pretendem aplicar as boas práticas do Scrum em situações de projecto.

Talk: “Agility with uncertain tasks”, Joaquim Baptista

Saturday, 26 June, 11:00 - 12:30 @ Main Conference Room (B002)

Scrum assumes that a 2 to 4 week sprint can be planned in advance, with the work fully divided into 4 to 16 hour tasks. However, uncertain tasks cannot be estimated in practice, because they require an initial learning effort that comprises most of the actual effort.

uScrum (uncertainty Scrum) is an agile process distilled from the current practice at Altitude Software, where technical writing tasks are often uncertain. In uScrum, managers classify tasks into 4 orders of ignorance, depending on how much is unknown about them.

In 0^{oi} tasks, workers know everything needed to complete the tasks and managers have reliable estimates. In 1^{oi} tasks, workers must gather information from others to complete the tasks. Managers can estimate the total effort, but the date of completion often depends on external events. In 2^{oi} tasks, the work unfolds as workers follow a generic process and learn more about the task. Initial estimates of 2^{oi} tasks often fail when workers uncover unexpected work. In 3^{oi} tasks, workers face complex problems without any clear solution.

uScrum uses several strategies to manage uncertain tasks, leading to 4 different kinds of sprints.

In a “get things done” sprint, workers are overbooked with a mix of 0^{oi} and 1^{oi} tasks, and then micro-manage their tasks to overcome obstacles. At the end of the sprint, workers completed some subset of the scheduled work. In a “try with fallback” sprint, workers advance 2^{oi} tasks while using 0^{oi} and 1^{oi} tasks to fill any slack. At the end of the sprint, workers should have learned how to complete the 2^{oi} task or, otherwise, why the task is difficult. In a “just try” sprint, workers advance a single 2^{oi} task, hoping to understand the task better. In a “cannot go back” sprint, workers try to advance on complex tasks (2^{oi} or 3^{oi}). Managers must strive to integrate con-

tributions from workers into a shared and evolving understanding of related problems.

Although managers facing uncertain tasks cannot be certain of sprint outcomes, they can schedule sprints that balance risk against expected outcomes.

Talk: “XP Game — Introduction”, Tiago Silva

Saturday, 26 June, 11:00 - 12:30 @ Main Conference Room (B002)

XP Game is a way to go from theory to practice by simulating a real project managed with Agile concerns in mind. It enables a quick and fun way to understand concepts like velocity, story estimation and team motivation amongst others. There are no special requirements to attend this session and play the game. The goal is for each team to produce as much “work” and deliver as much business value as possible during the allotted time and iterations. This first session will introduce the concepts and explain the game to be played on the second session (tutorial).

Talk: “Simple Code — Regain Control over Software through Decremental Development”, Peter Sommerlad

Saturday, 26 June, 15:35 - 17:05 @ Main Conference Room (B002)

Developing Software using the latest complex technology is cool. Software evolution by adding features and fixing bugs can be a burden. Both lead to brittle, buggy, overly complex and poorly performing software solutions. The value of Simplicity of a software solution is

often neglected by many software people today. Making software simpler increases its value, because of lower maintenance burden and easier extendability.

This talk explains what “simple” software means and how to approach it with “Decremental Development” even for existing (unnecessarily) complicated software.

Talk: “Evolving Software: Five powerful metaphors to explain change”, Filipe Figueiredo Correia

Saturday, 26 June, 15:35 - 17:05 @ Main Conference Room (B002)

Human thought is fundamentally metaphorical in the way it deals with concepts, so it is not surprising that metaphors are so pervasive in such an abstract field as software development. Besides being a primary tool for thought, metaphors are also priceless when we need to explain non-trivial ideas.

One of the four values of Agile Software Development is “responding to change”, an area which is particularly fruitful in metaphors. In this talk we will look at five metaphors for software evolution, and how they relate to each other: “Learning to Drive”, “Software Decay”, “Technical Debt”, “Code Smell” and “Big Ball of Mud”. We will discuss the role that metaphors play in software development, their benefits and eventual liabilities.

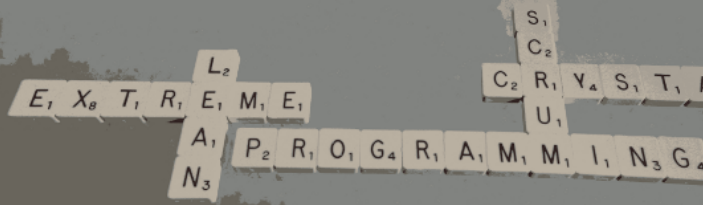
Talk: “Incomplete by Design — Thoughts on Agile Architectures”, Hugo Sereno Ferreira

Saturday, 26 June, 15:35 - 17:05 @ Main Conference Room (B002)

Despite several advances in software engineering, today’s developers still face recurrent problems in acquiring, inferring, capturing and formalizing requirements, particularly with systems where the process is highly-coupled with the stakeholders’ perspective and the requirements often change faster than the implementation.

Traditionally, once the analysis phase is finished and the implementation progresses, a strong resistance against further changes emerge, due to the mismatch between specification and implementation artifacts. Nonetheless, some domains do rely on constant adaptation of their processes to an evolving reality, since knowledge being continuously acquired lead to new insights of their own business and what support they expect from software.

Confronted with the above issues, Agile methodologies have intensified their focus on a highly iterative and incremental approach, accepting that change is, in fact, an invariant of software development. This blurs the line between designing and developing, specifying and implementing, and should we wish to harness continual change, that distinction no longer suits our purposes: design should become both the medium and outcome of action. Consequently, we are thus looking forward not just for a process to be effective and agile, but to what form should agile software take.



Tutorials

Tutorials are 90 minutes classes, taught by experts, designed to help software professionals rapidly get deep insight about agile methods, practices, techniques, and tools.

Tutorial: “Introducing Agility into an Organization or: How to become Agile”, Jutta Eckstein

Friday, 25 June, 11:00 - 12:30 @ Tutorial Room (B003)

According to Forrester Research 14% of the enterprises both in the USA and in Europe apply an agile approach and another 19% consider to get started with agile. A lot of companies want to become agile because they regard agility as a promising development approach, not only because the Standish Group is recommending agile development processes for avoiding project failures.

Yet, many teams are uncertain about how and where to start in order to become agile. Moreover, transitioning to agile often has an impact in many dimensions that are difficult to foresee.

In this tutorial, Jutta provides insights in how to get started with agile, what pitfalls to watch out for in the transition phase and as well how to ensure that the agile mindset will be preserved.

Tutorial: “Agile Refactoring for Making Systems More Adaptable”, Joseph Yoder

Friday, 25 June, 13:30 - 15:00 @ Tutorial Room (B003)

Refactoring has become a well accepted principle within the Agile community. Uncle “Bob” (Robert Martin) even states in his Agile Software Development – Refactoring and Pair Programming book: “Refactoring, which when done as part of Test-Driven Development (TDD) is my personal favorite way to spend my development time”. Agile software development promotes a process that encourages and embraces changes to requirements. Whereas Refactoring is a behavior preserving source-to-source program transformation. So how can refactoring help you make your system more adaptable? This tutorial will examine software evolution techniques such as Refactoring within the Agile Development process for making your system more adaptable as dictated by changing requirements.

Tutorial: “Agility and SOA – Experiences with System Landscapes under Maintenance”, Nicolai Josuttis

Friday, 25 June, 17:15 - 18:45 @ Tutorial Room (B003)

The world is growing together and more and more we have the task to realize business processes distributed over multiple teams, companies, locations, and domains. For example, mobile phone companies these day exchange data with banks and insurances, logistics companies (to ship mobile phones), phone vendors, airlines (to offer business customers both special rates and

lounge access), and so on. But, of course, we still have to be agile. We have to deliver new or modified distributed business processes as soon as possible with the best quality we can provide.

SOA is an approach that tries to deal with this requirement. And it has a meta story behind, which is pretty similar to agility. In essence, it is a value systems that takes the next revolutionary step for appropriate software development of large systems. Just like agility changes a lot by accepting that requirements change, SOA changes a lot by accepting that systems and solutions are heterogeneous. The problem is, however, that due to the size of the systems the common practices have to change to keep the agile value system alive.

In this talk I will present and discuss my experiences and lessons learned by trying to apply agile principles while maintaining large system landscapes with plenty of systems and hundreds or even thousands of developers. Topics will include: (i) Where do traditional agile practices not scale? (ii) Why are systems under maintenance different? (iii) How to refactor distributed systems? (iv) What kind of organizational structure and culture is necessary and appropriate for distributed development? (v) How to deal with distributed system tests? The value of this tutorial will be to get a consolidated first class experience report about the problems and possible solutions to be agile in our modern world of continuous development and modification of system landscapes that are getting larger and larger.

Tutorial: “CUTE GUTs for GOOD – Good Unit Tests drive Good OO Design”, Peter Sommerlad

Friday, 26 June, 11:00 - 12:30 @ Tutorial Room (B003)

While Java and .NET tend to have good tools fostering unit testing all code, C++ developers often lack the ability to unit test and refactor their code under development. At IFS we are extending Eclipse CDT with support for unit testing with the CUTE plug-in (C++ Unit Testing Easier) and several C++ refactorings we implemented.

This talk shows how to unit test with CUTE and how design improvements are driven by unit tests. It also shows how the attempt to unit test code exposes deficiencies in its design well before other uses are showing these. Especially close coupling and lack of cohesion is easily discovered and thus triggers refactoring the code. While the tools uses are for C++, the underlying concepts also apply to other languages as Java and the .NET family of languages.

The session will incorporate audience interaction by actively creating unit tests and refactoring and extending example code by the presenter under the audience’s guidance.

Tutorial: “XP Game — Playing”, Tiago Silva

Friday, 26 June, 15:35 - 17:05 @ Tutorial Room (B003)

This session follows on the previous talk on the same topic. It is the actual playing of the XP game according to

the rules that were previously explained. Teams will have 4 to 6 persons, and there will be a prize for the winning team.



Open Space

Open Space (in room I-105) goes beyond the program's pre-selected sessions, providing a place to share questions, talk to the experts, demonstrate software and techniques, and experiment with emerging Agile practices and ideas. Its schedule will be written by the participants, and modified throughout, as new topics emerge. Notes from the sessions should be recorded at the Open Space wiki.

Sessions are not meant to be formal, like the ones on the program — anyone can convene a session at the Open Space based on a passion and a commitment to make the session happen. Walk away with valuable new insights into topics that are relevant to you today.

Lightning-Talks

If you have something to share about agility to the other participants of the conference, and it requires using a video projector, you can upgrade to a lightning-talk. These are 5min presentations (+3min questions) integrated in the Open Space, devoted to inspiring ideas, fears, questions, concerns, experience reports, etc.

Retrospective

Teams that strive to improve need to accurately find and fix on-going problems. Agile retrospectives help teams to assess what went right and what went wrong on any project; iteratively and incrementally.

The retrospective by the end of the first day of Agile Portugal will help us examine what is working and what is not, thus assisting us on improving the ongoing conference. Day 2 will undoubtedly become much better thanks to everyone's feedback.

Panel

The Agile Development Process has enjoyed over a decade of acceptance in Industry and recently has grown to become the popular "buzz" word to throw around; even if it is done to show you participating with the latest accepted development process.

Agile focuses more on the people side encouraging better interactions and communications between people. Additionally Agile openly embraces changes to the system. So given all this hype and acceptance, is Agile the end? Have we finally found a process that can be used for most of our software development? Or is there life beyond Agile?

This panel will examine the potential benefits of Agile and contrast this to what we might see as we move Beyond Agile!



Social Events

Agile Portugal 2010 provides you many opportunities for you to gather with your peers, meet your friends, make new ones, socialize and find common interests in an informal way.

Coffee Break

In the middle of talks and keynotes you can have the opportunity to grab some coffee and cookies, and get energy for the rest of the day. Coffee Breaks are always served in the Open Space Room (I-105).

Lunch

Lunch meals are served right in front of the Main Conference and Tutorial Rooms (B002/3), in exclusivity for Agile Portugal 2010.

World Cup 2010: Portugal - Brazil

This is the year of the Soccer World Cup – Portugal and Brazil belong to Group G and will face each other on June 25th. The game will be displayed in a big screen in the Open Space Room (I-105) simultaneously with an extended coffee break. A great opportunity to thrill in support of our team among the world top agile experts!



Porto by the Douro River

For those that are visiting Porto for the first time (but not only), the Social Event will provide you an intensive impression of the city and its riverfront attractions.

We will leave the conference venue (FEUP) by bus, we will do some sightseeing, and we will end the bus trip at Ribeira de Gaia.

We will then take our time to visit the Porto Wine Cellars, and taste the most important thing they hold. After that we will see Porto from the Douro river, during a “Porto World Heritage Cruise”.

After all of this, it will be dinner time. A restaurant nearby will serve us a buffet with the most important of the traditional Portuguese gastronomy:

Tromba Rija Gaia

Avenida Diogo Leite, 102 – Vila Nova de Gaia

T +351 223 743 762 | +351 968 603 750

GPS 41.137581, -8.613842

Notes

Internet Connectivity

How can I use FEUP wireless network during the conference?

1. Connect to the guest-eduroam SSID.

This is a closed network and only gives access to the FEUP hotspot information, so you must enter a credential for network access.

2. Choose Login

It's the link located on the bottom of the loaded page.

3. Enter the credentials supplied

username: agile

password: portugal2010

4. Done

You should be connected now. This service only supplies access to the internet using HTTP and HTTPS.

Useful Information

Conference Contact

If you have a problem and need to contact someone from the conference, please ask someone at the information desk. If you really need to call someone, you can use the FEUP/DEI phone..... 22 508 2134

Emergencies

Emergency number in Europe (Police, Accidents)..... 112

Taxis

Taxis close to the venue..... 22 507 3900

Touristic Information

In your participant bag you may find touristic information about Porto.

Student Volunteers

Several students from the Masters in Informatics and Computing Engineering will be helping with the organization of the conference, and there will always be at least two elements in each room and information desk. They will be more than glad to provide you assistance.

Personal Schedule

Day I

Friday, 25 June

08:15 - 08:40	Registration	@ I-105
08:40 - 09:00	Conference Opening	@ I-105
09:00 - 10:30	Keynote. <i>"Improving the Quality and Productivity of Backlogs Through Envisioning"</i>	@ B002
10:30 - 11:00	Coffee Break	@ I-105
11:00 - 12:30	_____	@ _____
12:30 - 13:30	Lunch	~ B002
13:30 - 15:00	_____	@ _____
15:00 - 17:15	World Cup 2010. Portugal - Brazil	@ I-105
17:15 - 18:45	_____	@ _____
18:45 - 19:00	Retrospective	@ I-105
19:00 - 22:30	Social Event	

Personal Schedule

Day 2

Saturday, 26 June		
08:30 - 09:00	Registration	@ I-105
09:00 - 10:30	Keynote. <i>"Big Ball of Mud: Is This the Best that Agile can Do?"</i>	@ B002
10:30 - 11:00	Coffee Break	@ I-105
11:00 - 12:30	_____	@ _____
12:30 - 14:00	Lunch	~ B002
14:00 - 15:30	Keynote. <i>"Agile Software Development in the Large"</i>	@ B002
15:35 - 17:05	_____	@ _____
17:05 - 17:30	Coffee Break	@ I-105
17:30 - 18:30	Panel. <i>"Beyond Agile"</i>	@ B002
18:30 - 19:00	Closing Session	@ I-105

