

QWAN

Quality Without A Name



eXperience Agile! courses & workshops 2008

Contents

Introduction.....	3
Course overview.....	4
Coaching and Mentoring.....	6
Agile Awareness.....	8
Practical Scrum for new Masters.....	9
eXperience Agile.....	10
Unit Testing Masterclass.....	12
eXperience Refactoring.....	14
Lean Software Development.....	16
Systems Thinking.....	18
Mastering Projects Workshop.....	20
Agile Implementation.....	21
Retrospectives Workshop.....	21
Half day workshops.....	22
Trainers.....	23

Introduction

Welcome to our courses & workshops brochure!

We (Marc, Rob, and Willem) help customers to get more value from software development. We recently decided to join forces, and sell software development projects, mentoring and training under a joint label. *QWAN (Quality Without A Name)*^{*} is an initiative of pragmatic practitioners, who have joined forces to deliver courses and mentoring on effective software development. *Quality without A Name* means that we strive for something elusive, which we can't word exactly.

We're interested in advancing the state of our practice and that of our customers. To that end, we investigate new practices, apply them in the real world, and reflect. We share our experiences, good and bad, in the form of articles, blogs, (practitioner's) conference appearances, courses and hands-on mentoring.

Topics we specialize in are agile software development (lean, scrum, extreme programming, test driven development), open space, and systems thinking (seeing and improving the whole as well as the parts).

Open enrollment courses - We offer eXperience Agile, Unit Testing Masterclass, and eXperience Refactoring as open enrollment courses.

In-company courses and workshops - All courses in this brochure can be run in-company, for groups of 8 or more. For smaller groups, we offer a combination of hands on coaching and mentoring, supplemented with presentations and exercises where needed.

Custom-made training - If you have specific needs that are not covered by a course or workshop in this brochure, we can create a bespoke training programme for you. To that end, we combine existing workshops or develop new course modules for you.

Coaching and mentoring - In addition to courses and workshops, we also provide hands on coaching and mentoring.

Change facilitation - After training and mentoring, there always remain issues that are specific to your organisation. We facilitate Open Space events, to let your organisation address pertinent issues and facilitate effective change.

We'd love to hear your feedback about this brochure and our offerings:

- What do you appreciate?
- What can we improve?
- Is there anything that puzzles you?
- Did you learn new things you didn't know before?
- Any (educational) wishes you still have?

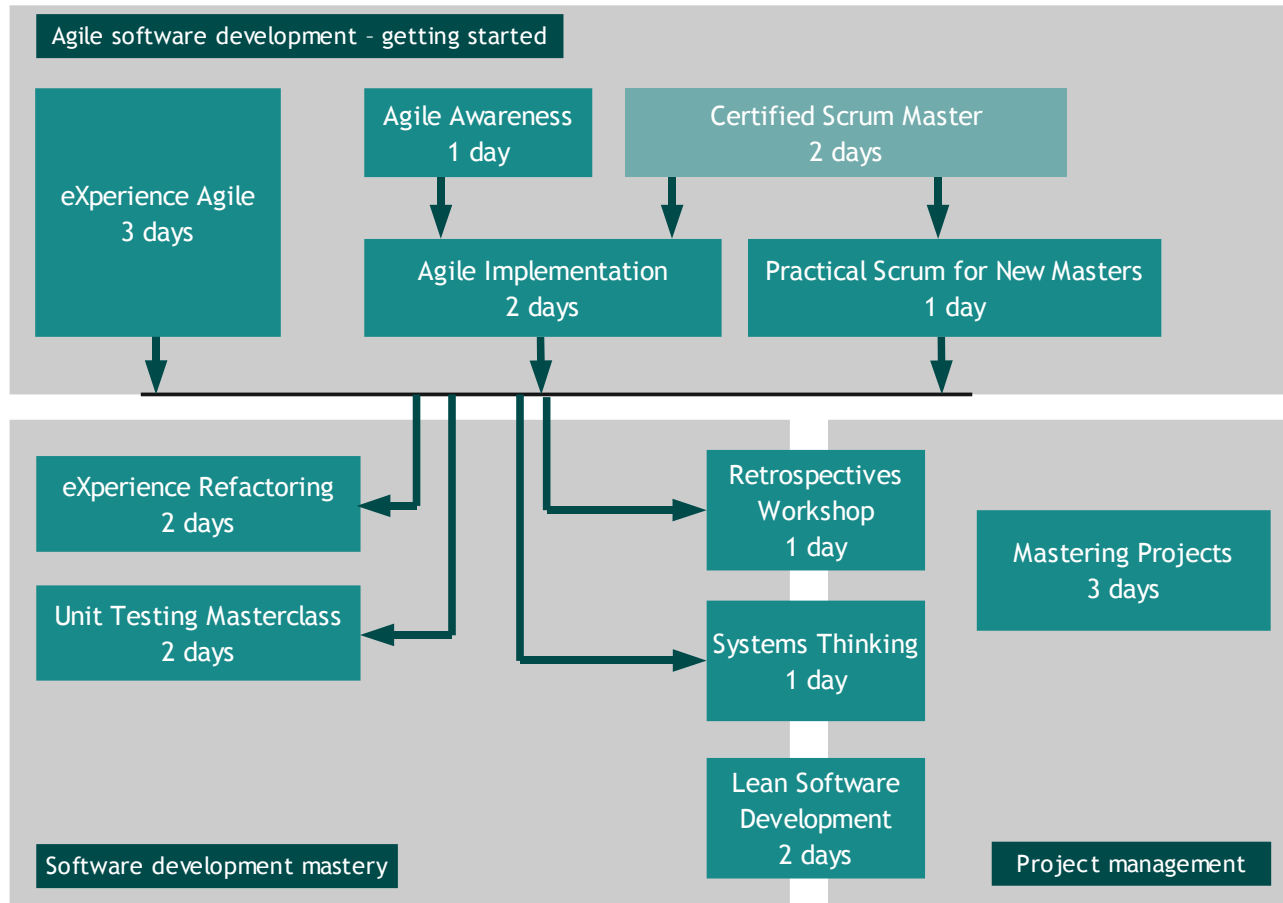
We hope to see you and your colleagues at one of our courses!

Best regards,

Marc, Willem & Rob - www.qwan.it

* The name *Quality Without a Name* was inspired by Christopher Alexander's work on architecture and patterns. We highly recommend his books *The Timeless Way of Building* and *A Pattern Language*.

Course overview



If you want to be the first to know about upcoming courses and conferences, subscribe to:
www.qwan.it/newsletter



What participants have said about our courses and workshops:

"Very good and fun course. Nice introduction to a new culture of software development."

"Three very fun and educational days, Good combination of theory/presentation and hands-on exercises."

"Very educational (as I expected); I can definitely apply this in my project"

"The practices of TDD and implementation of a small project in three iterations which involved almost all practices of XP where very good for me to get real experience."

"This was the first training where I had:
- no problem to stay awake
- no problem with the afternoon "dip"
- no problem to stick to the subject
- no problem completing the exercises."

"Awesome! Very valuable tool to understand causal effects."

"Well run, good materials. Clear and useful content."



Coaching and Mentoring

We offer coaching and mentoring of teams and individuals in agile development practices. Coaching and mentoring can be a highly effective follow up to a training course. We help people to adapt swiftly, so they can work in new ways effectively. We also support our clients in tailoring practices to the context of their day-to-day work.

Change is a hard skill

Several things can go wrong when you try to transition to a different way of working. For instance you encounter resistance from the organization, the development process becomes a boring routine, or the team doesn't go beyond 'agile by the book', so that continuous improvement never happens.

We help you change successfully by offering mentoring in e.g.:

- getting started as a Scrum Master;
- adopting Scrum, eXtreme Programming, or Lean Software Development;
- iterative planning and estimating;
- leading retrospectives;
- discovering what mix of practices works for your team;
- adapting practices to your needs.

Benefits

We have helped our customers reduce time to market, reduce defects, deliver software in difficult circumstances (limited budgets, unclear requirements), and increase the fun with which teams, their managers and their customers work.

Approach

Our approach is characterized by:

facilitating - we create space where people can safely experiment and learn

responsibility - everyone takes responsibility for their own learning

experiential - people learn best by doing and experiencing

systems view - let people see the whole together, to prevent suboptimization

continuous reflection and improvement - the foundation for better performance

collaboration, process, and results focus - we take responsibility for the results of our joint effort; we can implement what we recommend and keep track of both the ball and the goal.

Courses

Agile Awareness

Everybody seems to be doing or being 'agile', everyone in his own way. What is it all about? Scrum is hot, XP is not, or is it? What exactly is 'agile'? Which methods and processes are there? What does agile mean in a broader organizational context? Is agile the ultimate goal or is there more?

In this workshop, we provide an overview of agile software development, from underlying values and principles to the concrete, well known practices and processes. We will show the role agile can play for developing organizations. You will experience an iterative process and a real retrospective to round up the workshop.

Benefits

After participating in this workshop, you will:

- have an overview of what agile software development is about
- know the most important methods that fall under the agile umbrella
- see agile development and agile transitions in a broader context
- have experienced planning iteratively and holding a retrospective

Intended Audience

This workshop is intended for developers, analysts, architects, testers, team leads, project managers, IT managers, IT consultants.

Programme

- Introduction to agile: manifesto, values, principles
- Benefits, success stories, failure modes
- Overview of Scrum, eXtreme Programming, and Lean Software Development
- Simulation of iterative planning
- Cultural patterns of software development organizations
- Retrospective - an essential practice to establish a culture of continuous reflection and improvement.

Practical Information

Duration: 1 day, 9:00-17:00

Group size: 6-30 participants

Open enrollment fee: € 525 per participant (exclusive of VAT)

In-company training fee: € 4000 (exclusive of VAT)

Practical Scrum for new Masters

A Japanese proverb says: “Better than a thousand days of diligent study is one day with a great teacher.”

This course is designed for recent graduates of Certified Scrum Master courses, containing practical advice to make the first few months of being a Scrum Master easier. Patterns, best practices and more to improve being a Scrum Master, extracted from personal experience, thousands of posts to popular mailing lists and hundreds of conference presentations.

Benefits

Certified Scrum Training is designed to teach the basic understanding of Scrum. This additional training is designed to supplement that theoretical knowledge with hands-on experience reports and useful situational leadership. Giving Scrum Master confidence and the necessary tools to be more effective immediately.

- Common mistakes can be avoided, accelerating the adoption of Scrum
- Situational Leadership skills boost the Scrum Masters confidence
- Key skills are substituted with knowledge learned by delivering products utilising Scrum
- Offers a forum to answer questions with experienced mentors

Intended audience

This course is designed for recent graduates of Certified Scrum Master courses containing practical advice to make the first few months of being a Scrum Master easier.

Programme

In this practical workshop we will discuss and develop core skills needed to be a Scrum Master:

- patterns for common problems and issues
- using burndown charts
- increasing visibility
- retrospectives
- selforganised teams
- what do I do know?

Practical information

Duration: 1 day, 9:00-17:00

Group size: 6-14 participants

Open enrollment fee: € 525 per participant (exclusive of VAT)

In-company training fee: € 4000 (exclusive of VAT)

We organize this course together with David Hoehn.

eXperience Agile

Agile Software Development requires a fresh perspective, a new frame of mind, a paradigm shift. The eXperience Agile training course lets the participants experience what it is like to work in an agile way. Experience agile values, principles and practices in three days through presentations, workshops, discussions and the execution of a small software development project.

We keep this course directly relevant, applicable and valuable to everyday practice. Therefore we develop this course in an agile fashion - we use feedback from each instance of the course to improve the next.

Benefits

If you participate in eXperience Agile, you will:

- Learn planning, programming and testing techniques hands-on
- Understand the way Scrum and eXtreme Programming are complimentary
- Experience the role testing and test driven development play in eXtreme Programming
- Understand the relation between agile values, principles and practices
- See the value of eXtreme Programming
- Have fun! We take pride in creating a fun and effective learning environment, therefore we mix presentations with simulation games, exercises and hands-on development.
- Know how to deliver projects on-time and within budget, without compromising on quality

Intended Audience

Programmers and technical project leaders who seek a basic understanding of Agile principles and eXtreme Programming .

Prerequisite knowledge and experience:

- Object orientation
- Experience in a programming language with Object/Class features (Java, C#, C++, VB(.Net), Ruby, Python, Smalltalk)
- Some experience working on software development projects

Programme

eXperience Agile consists of three intensive days of training:

The first day is introductory, with an overview of eXtreme Programming values, principles and practices, participants playing a simulation of planning in eXtreme Programming and an explanation plus hands-on demonstration of Test Driven Development.

Day two starts with participants doing Test Driven Development exercises in pairs, followed by an introduction to Agile Project Management with Scrum and an interactive part on writing User Stories (the eXtreme way of handling Requirements).

Day three is the hands-on day. In three iterations we develop (plan, test, code, refactor, accept) new features to an existing application. Participants are coached in applying what they have learned - delivering high quality software that is fit for purpose in an eXtremely short timespan, and improve their process while working.

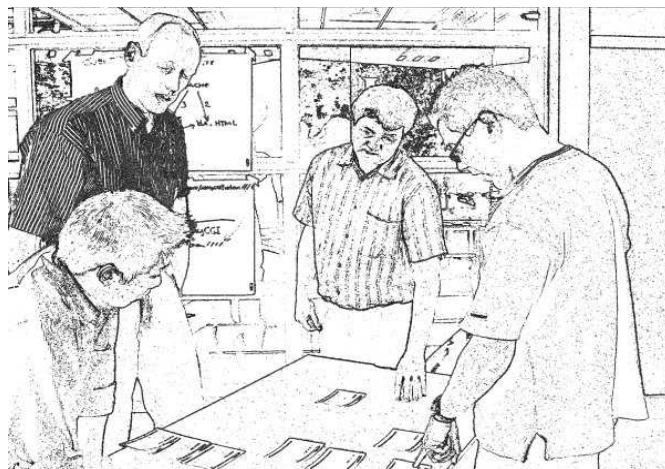
Practical Information

Duration: 3 days, 9:00-17:00

Group size: 8-14 participants

Open enrollment fee: € 1575 per participant (exclusive of VAT)

In-company training fee: € 12000 (exclusive of VAT)



Unit Testing Masterclass

In this two day, in depth practical course, you will master different unit testing techniques, like mocking and unit testing in legacy code.

Benefits

When a team masters unit testing, they and their customers will benefit from a faster development cycle with fewer defects and software that has a better design. This makes it easier to adapt to changing customer requirements.

By participating in this masterclass, you will:

- gain a better understanding of what a unit test is
- learn techniques and guidelines for writing and maintaining tests
- test code in isolation with test driven development and mocking
- experience various styles of mocking
- apply a number of techniques to break unwanted dependencies
- have some experience adding tests to legacy code
- know how to evaluate and improve your unit tests
- understand system dynamics of unit testing and design debt
- have fun! - we take pride in creating a fun and effective learning environment, mixing presentations with exercises and hands-on development.

Intended audience

Software developers, architects, technical team leads, Scrum masters, and agile coaches who seek to become masters of unit testing, so that they are able to deliver better code faster.

Prerequisites

- Knowledge of object oriented concepts
- Experience in a programming language with Object/Class features, like Java, C# or Ruby
- Experience working on software development projects
- Basic knowledge of automated testing

Programme

The course consists of a mix of short presentations, live programming demonstrations, and lots of programming exercises.

Course Introduction

Round of introductions; set expectations & collect issues; presentation *Why unit testing?* to see the short term and long term effects of unit testing from a systemic perspective.

Introduction to Test Driven Development

Experience the basics of test driven development; learn what makes a *unit* test.

Responsibility Driven Design with Mocking

Learn and experience the use of mocking techniques to write better isolated unit tests and to arrive at a more loosely coupled design; learn about the difference between interaction based testing and state based testing.

Mocking Styles

Learn and experience different ways of creating mock objects; learn the difference between mocks, stubs, fakes, and dummies and when to apply them.

Getting Your Tests In: Breaking Dependencies in Code

Learn how to break dependencies in a responsible way; learn a number of refactorings that enable you to add tests to existing code without unit tests.

Story Testing with RSpec

Get a fresh perspective on functional/integration/unit testing; RSpec is a behaviour driven development framework that can be used both for writing story tests and writing executable behaviour specifications for objects.

Closing Retrospective

Reflect on learning experience, focus on bringing lessons learned to daily practice.

We can adapt the programme if desired when we run the course in-house. Some alternative course components are:

Unit Testing Clinic

Learn how to solve specific problems from daily practice

Participants bring in examples of unit tests they have trouble with or code that is hard to unit test. We will discuss the issues, provide suggestions for improvement, and, whenever possible, help to make it work directly.

Right Sizing Your Unit Tests - what makes a good test?

By examining several code examples, learn to answer questions like: what is the right size of scope for a unit test? how does the design of production code affect test code and vice versa?

Practical information

Duration: 2 days, 9:00-17:00

Group size: 8-14 participants

Open enrollment fee: € 1050 per participant (exclusive of VAT)

In-company training fee: € 8000 (exclusive of VAT)

eXperience Refactoring

In this two day training course, you will learn how to improve the design of software step by step, while continuing to deliver value to your customers.

Everyone wants software that 'just works', that is a joy to maintain, and where you can add new features easily. In practice however, there are always bugs to fix and customers to please. Corners are cut and design debt accumulates, ultimately resulting in *legacy code* - code without tests.

You would like to improve the software in small steps using *refactoring* - which means improving the design of existing code without changing its behaviour. There is a Catch-22 here: to refactor safely, you need automated tests to ensure the code behaves the same as before. To add tests, you first have to make it modular by refactoring.

There still is hope! Through presentations, demonstrations, exercises, and a representative case, you will learn to identify smells, break dependencies, refactor, and add tests in messy legacy code, so that you can add new features with more ease and confidence.

The techniques taught in this course are also useful in *greenfield* projects, to prevent the software from turning into legacy code. They help you to recover quickly from mistakes and learn from them.

This course helps you get started and offers effective practices and courage to persevere.

Benefits

If you participate in eXperience Refactoring, you will:

- learn how to recognize specific code smells and tackle them with refactoring
- apply a number of techniques to break unwanted dependencies in code
- know how to start making changes and adding tests simply and responsibly
- be able to find *seams* in code and use them to break dependencies
- understand the system dynamics of design debt
- have fun! - we take pride in creating a fun and effective learning environment, mixing presentations with exercises and hands-on development.
- experience prioritizing, planning, and executing a large refactoring in small steps with a team, on real, representative legacy code.

Intended Audience

Software developers, architects, technical team leads, Scrum Masters, and agile coaches who seek to augment their refactoring knowledge, so that they are able to deliver clean(er) code more often.

Prerequisite knowledge and experience:

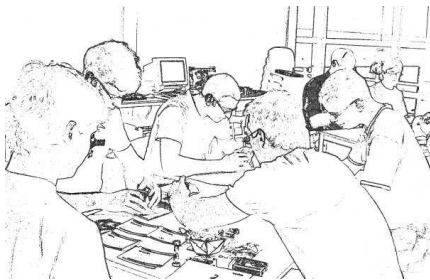
- Object orientation
- Experience in a programming language with Object/Class features, for example Java, C#, C++, VB(.Net), Ruby, Python, Smalltalk
- Some experience working on software development projects
- Basic knowledge of automated unit testing

Programme

eXperience Refactoring consists of two intensive days of training.

The first day is introductory, with an overview of refactorings, code smells, and how to break dependencies by finding seams in the code. We will do this through a mix of presentations, demonstrations and small exercises. You will also learn about the systemic causes and effects of design debt and how refactoring in small steps can help you break out of a vicious cycle.

Day two is the hands-on day. In two parts we will plan and execute a large refactoring on a real-world legacy system. Participants are coached in applying what they have learnt: working on a large, messy legacy code base without getting lost, using the vocabulary of smells, refactorings, seams, and dependencies to co-ordinate, plan and execute a large refactoring with a team. And do so safely, even though in the beginning no automated tests are present.



Practical information

Duration: 2 days, 9:00-17:00

Group size: 8-14 participants

Open enrollment fee: € 1050 per participant (exclusive of VAT)

In-company training fee: € 8000 (exclusive of VAT)

Lean Software Development

Competitive advantage through continuous reflection and improvement

Lean is a way of thinking that originates from manufacturing and production processes. Thanks to the sustained success of companies like Toyota and Zara, interest for lean is growing.

Lean thinking helps you to deliver increasingly better products faster. The principles of lean can be summarized as:

- respect for people
- relentless elimination of waste from processes

Waste is every activity that doesn't directly add value for the customer. Lean means continuous reflection and continuous improvement. It also means seeing people as the key factor and continuously developing people.

In this course, you will learn principles and practices of lean software development. You will also practice with a number of concrete conceptual tools, so that you can apply these in your daily work.

Intended Audience

This course is aimed at people involved in software development, in particular project managers and IT managers.

We expect participants to have some experience with software projects, as a stakeholder, manager or developer.

Benefits

After participating in this course, you will be able to:

- explain what the principles and background of lean software development are and how lean relates e.g. to other agile processes
- determine where and when lean thinking and lean practices add most value within your organisation
- apply a number of lean tools and use them to run projects (and your organization as a whole) more effectively

Programme

Principles and background of lean software development

Applying the Toyota Way principles to software development

Systemic approach to (IT) organizations:

- local vs. global optimization
- lean and learning organizations: single, double, triple loop learning
- effect of measurement systems in organizations, like reward and accounting systems

Position of lean software development with respect to other processes, methods, frameworks, and approaches, like Scrum, eXtreme Programming, DSDM/Atern, RUP, best practices and CMM.

Practices and tools:

- detect and eliminate waste in development processes
- set based development
- identify and optimize the whole value stream using value stream mapping
- pull systems and one piece flow: effects of push systems and batch size
- process visualization, e.g. visualizing progress and work in process
- role of testing and reviews
- retrospectives
- continuous reflection and improvement

Lean software development in practice

Introducing of lean in an organization

This interactive course consists of presentations, assignments, exercises, and simulations. There will be ample room for questions and discussions. The proportion of presentations to exercises and simulations is 50% / 50%

Practical Information

Duration: 2 days, 9:00-17:00

Group size: 6-20 participants

Open enrollment fee: € 1050 per participant (exclusive of VAT)

In-company training fee: € 8000 (exclusive of VAT)

Systems Thinking

Agile software development methodologies prescribe some odd practices that counter conventional wisdom. Even some of the names sound funny and can hardly be taken seriously.

However, Agile methods have a common basis consisting of mechanisms that have proved to be effective in several industries (not only software development). While several software development methodologies describe a recipe for practices that worked in a specific context, the exact same recipe is not guaranteed to work in all circumstances.

We introduce our approach to process development, which integrates retrospectives, systems thinking, and value stream mapping to effect substantial change in small, controllable steps, and prevents local (sub)optimization.

Understanding why your process works the way it works enables you to monitor the extent to which a particular practice is effective in its context. It also helps to organize incremental process improvement in such a way that your organization can grow and develop its own way of developing software.

Benefits

After participating in this course you will:

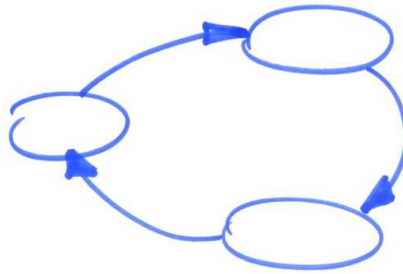
- view software organizations from different, systemic perspectives
- find root causes for all kinds of persistent, messy team problems
- to really solve your problems, instead of fighting fire after fire
- understand why different software organization work and can work and why they sometimes stop working effectively
- adapt your change initiatives to the specific organizational context
- understand how change happens in individuals and organizations, and what this means for steering a change process

Intended audience

Project leaders, managers, agile coaches, Scrum Masters who have tried one or more agile approaches or want to try some, and who want to understand their way of working better and learn new ways from both their colleagues and the presenters.

Programme

- Introductory presentation, explaining how systems thinking, value stream mapping, and retrospectives reinforce each other and deliver effective, manageable change.
- Introduction to systems thinking with a story from the presenters. Followed by hands-on systems thinking by the participants the problems they bring in.
- Value stream mapping explanation, with an example from the presenters. Exercise on one or more value streams from participants.
- Explanation of retrospectives and how to use them for ongoing process improvement. The course itself will be used as an example to retrospect on.



Practical information

Duration: 1 day, 9:00-17:00

Group size: 6-20 participants

Open enrollment fee: € 525 per participant (exclusive of VAT)

In-company training fee: € 4000 (exclusive of VAT)

Mastering Projects Workshop



Success or failure depends on successfully leveraging the human factors of each project: aligned expectations, successful communications, political support, and real motivation. Too often in our focus on managing project elements - scope, work breakdown structures, task relationships, roles, responsibilities - we lose touch with the heart of the effort, the human factors that matter the most. Without even realizing it, we wind up exhausting our resources instead of using our projects to build our long term capabilities.

This workshop teaches practical human centered design strategies that turn a project's day-to-day barriers into a foundation for success. You'll take away concrete techniques to structure the project, focus your energy, motivate your people, and align the effort to deliver real success.

Developed to work with any method - Agile or traditional, Mastering Projects is designed from best practices of organizations all over the world creating new products, new software, and new organizations. Using experiential learning, practical application, and reflective observation, this workshop will evoke profound lessons.



Benefits

You will learn to:

- Tailor your project into a natural fit with your organization.
- Turn overcommitted resources into a capable self-managing team.
- Plan without creating a tangle of tasks and responsibilities.
- Maintain focus amidst shifting goals and a whirlwind of urgencies.
- Keep efforts on track to delivery without overwhelming yourself.
- Agilely master your project in the most challenging organization.

Intended Audience

- Formal or 'accidental' project managers
- Project managers juggling multiple projects or managing projects in addition to their 'regular' job
- Executives and leaders working on complex projects and cross-functional business initiatives
- Scrum masters working in projects that don't go by the book

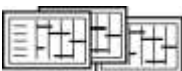


Practical Information

Duration: 3 days, 9:00-17:00

Group size: 6-20 participants

In-company training fee: please contact us for a quote



This course is facilitated by David Schmaltz and Amy Schwab from True North pgs. For more information, see www.projectelephant.eu

Agile Implementation

When you have learned the basics of agile software development through our Agile Awareness workshop or through a Scrum Master training class, you might wonder how to actually get things done and get your software out of the door when you go agile.

This two day, hands on course focuses on the down to earth side of working in an agile way: how to plan, build, test, and deploy the software. How to get it done using user stories, test driven development, automated testing, incremental design, relentless refactoring, pair programming, continuous integration.

Duration: 2 days, 9:00-17:00

Group size: 8-14 participants

Open enrollment fee: € 1050 per participant (exclusive of VAT)

In-company training fee: € 8000 (exclusive of VAT)

Retrospectives Workshop

A one day intensive, practical workshop, where you learn tools and techniques to get more out of your retrospectives.

Duration: 1 day, 9:00-17:00

Group size: 6-20 participants

Open enrollment fee: € 525 per participant (exclusive of VAT)

In-company training fee: € 4000 (exclusive of VAT)

Half day workshops

XP Game

An experiential way to learn iterative planning

The planning game is one of the more paradoxical parts of eXtreme Programming. The XP Game is a playful way to familiarize the players with velocity, story estimation, yesterday's weather and the cycle of life. Anyone can participate. This game is especially useful when a team or company starts adopting Scrum, XP, or any iterative development process.

Drawing Carousel

Experience how a pair programming team works

Working in a pair is different from working on your own: you have to articulate what you want to do, the other person has different insights, and the result will be different from what you would have done on your own. You will experience the effect of agile tools like daily stand up meetings, pair rotation, coding standards and collective code ownership. You will see improvements in the quality of your product and the productivity of your team.

Rightsizing your unit tests

Quickly learn about the costs and benefits of various testing techniques

This workshop helps programmers who have started test driven development to grasp the consequences of advanced testing techniques, and managers to understand the short term and long term consequences of trade-offs made by developers. The participants examine several code examples to learn how to answer questions such as: what is the right size or scope for a unit test? How does the design of production code influence tests and vice versa?

Who's Dropped the Ball !?

An introduction to systems thinking using causal loop diagrams.

Systems thinking is a methodology independent tool for understanding team and project dynamics and for finding effective interventions.

You will gain some insight in the nonlinear dynamics of teams and learn how to make causal loop diagrams to capture these dynamics.

Practical Information

Duration: ½ day

Group size: 8-24 participants

In-company training fee: € 2000 (exclusive of VAT)

Trainers

Rob Westgeest



After years of experience with Object Oriented Software Development with UML, several development processes and project approaches as developer, trainer and project leader, Rob worked on his first XP project in 2000. And with great success! He supports projects and people in the application of agile practices, principles and values since then. Rob develops himself and others continuously by visiting, organising and hosting workshops at conferences and user group meetings like SPA, XP Days, XP-NL and Agile Open.

Rob explains hard problems in a simple way, so the problems and the solutions are easy to fathom. He is able to let others experience what he learned quickly, and so doing guides teams around pitfalls. His enthusiasm and sense of humor makes it a great pleasure to work with Rob.

Willem van den Ende



Willem van den Ende is a Dutch eXtreme Programming pioneer. Since 1999 he guides organisations in the introduction of Agile Software development as an all-hands person: coach, developer and facilitator. Always active in the local and international community, he has served as board member of the [Agile Alliance](#), host of [systemsthinking.net](#) and the [European Agile Open](#) conferences. Willem is an appreciated workshop facilitator at practitioners' conferences like XP(Day), [Software Practice Advancement](#) and [Agile200*](#).

Willem's sharp vision, his broad knowledge, and twenty years of experience as programmer and coach enable him to adopt a very flexible and improvising attitude during workshops. He has the ability to let people see things differently.

Marc Evers



Marc works as an independent coach, trainer and consultant in the field of (agile) software development and software processes. Marc develops true learning organizations that focus on continuous reflection and improvement: *apply, inspect, adapt*.

Marc also organizes workshops and conferences on agile and lean software development, extreme programming, [systems thinking](#), theory of constraints, and [effective communication](#). Marc is co-founder of the [Agile Open](#) and [XP Days Benelux](#) conferences.

Marc knows how to combine his real-world experience with knowledge that is out there to create novel solutions. He likes to add games to his highly-rated workshops, so participants have fun and learn from experience.

QWAN

Quality Without A Name

www.qwan.it

Contact information: www.qwan.it/contact

The QWAN founders are:



Living Software B.V.

www.livingsoftware.nl

willem@livingsoftware.nl

+31 6 4130 6965



Westgeest Consultancy

www.westgeest-consultancy.com

information@westgeest-consultancy.com

+31 6 4577 6328



Piecemeal Growth

www.piecemealgrowth.nl

marc@piecemealgrowth.nl

+31 6 4455 0003

Edition: September 2008

© 2008 Living Software B.V., Westgeest Consultancy,
Piecemeal Growth