

AGILE PROJECT MANAGEMENT IN PRACTICE

COURSE CONTENTS

Summary:

Agile Project Management In Practice examines the actors that may deviate a project from the planned schedule and provides tools to understand the process and the corrective measures to take.

This course enumerates the list of events related to scope, time, cost and quality from an agile perspective.

Estimated duration:

- 20 hours.

Objectives:

- **Move towards simplicity** during the entire software development process.
- **Correctly prioritize** software tasks.
- **Identify factors** that affect project execution and project viability.
- Assume the **role of communication** in the team.
- Acquire techniques to **correct a late project**.
- Improve **productivity and commitment** of the team.
- Detect **quality patterns** in the course of the project.
- **Identify common marketing practices** while evaluating software.

After this course, you will be able to:

- Manage the **factors that drive uncertainty**.
- **Anticipate delays** with enough time to start corrective measures.
- **Assume changes to requirements** at any stage of the project.
- Avoid common **project management antipatterns**.
- Optimize the **work environment**.

AIMED AT

This course is focused on **project managers** and **software architects** with at least five years of experience with software development projects.

Requirements:

Experience developing systems with Java or .NET is recommended but not required. This course does not include labs with any concrete technology.

CONTENTS

Day 1	1	2	Introduction
			Requirements
Day 2	3	4	Communication
			Schedule
Day 3	5	6	Work environment
			Quality
Day 4	7	8	Antipatterns
			FUD

Module 1: Introduction

Presentation of the course.
Contents and module distribution.
The four variables of software development: cost, time, quality and scope.

Module 2: Functional requirements.

Prioritizing tasks: Late Prioritization and the Moscow Rule.
Solve the Main Business Problem.
Domain Driven Design.
The Reason It All Exists: code and added value.
The cost of each new feature: The Simplest Thing That Could Possibly Work and YAGNI.
Jakob's Law of Internet User Experience.
Suicide Requirements.
Integrator vs Orchestrator. Buy, don't build.
Comparative: cost of change with Waterfall, Prototype and Continuous Integration.

Module 3: Communication

The Mythical Man-Month: relationship between efficiency and size of the team, with and without communication.
How to handle the communication overhead.
Communication peers: clients, analysts and developers.
Noise in your office.

Module 4: Schedule

The waterfall model and the Brooks estimate.
Feedback cycles with Prototypes, Timeboxing and XP.
Introduction to Scrum.
Brooks Law.
No silver bullet, with examples.
Time estimates: Parkinson Law, Student Syndrome.
Consequences of sharing resources.
Comparison: early and late start.
Introduction to Critical Path and Critical Chain.

Module 5: Work environment

Flow and interruptions.
The phone.
"Nothing is done here between 9 a 5": brain time vs body time.
Working overtime: Spanish Management Theory and Personal Chemistry. Workaholics.
Costs associated to turnover.
Space organization.
DeMarco and Lister study of development productivity ratios.
Pair programming.
Choosing your tools: Diffusion of Innovation, Crossing the Chasm, Reflexive Markets.
Automation. Where and why.
Roles in a development team: The Peter Principle.
Bureaucracy Control.

Module 6: Quality

Why OOP: the cost of error.
Patterns and their relation to quality. Good and bad practices.
Product vs project oriented.
Continuous Integration from a quality perspective.
Technical Debt and Technical Inflation.
Software Entropy.
No Broken Window.
Refactoring and Once and Only Once. Cost associated to refactoring.
The Simplest Thing That Could Possibly Work from a quality perspective.
How to achieve continuous training in a team.

Process Oriented software.

Tests: unit tests, integration tests, regression tests, profiling, load tests.

Module 7: Project Management Antipatterns.

Truck Number.

Second system effect.

The Way Things Have Always Been Done.

Reinvent the Wheel and the Not Invented Here syndrome.

The Boat Anchor.

Golden Hammer.

Ivory Tower Architect.

Net Negative Producing Programmers.

The Grand Old Duke of York.

Module 8: FUD tips and tricks.

Argumentum ad hominem.

Associative Fallacy.

Ad Novitatem and Ad Antiquitatem.

Legitimate uses of Ipse Dixit.

Argumentum ad Populum.

Argumentum ad Nauseam.

Red herring.

Correlation and Causality.

The course includes approximately one practical lab per module.

ABOUT

Extrema is committed to **technological consulting, software development and advanced technical training**. We adapt years of software development experience to design our advanced training courses.

For more information contact training@extrema-sistemas.com