

# Extreme Programming

## Myths & Realities

Martin Lippert

University of Hamburg & it Workplace Solutions GmbH,  
Germany

[lippert@acm.org](mailto:lippert@acm.org)

Are you

really

using

Extreme Programming  
?!?  
!!!

# What is Extreme Programming ???

Hacking ?      Something new ?  
Chaos ?  
Uncontrollable ?      Extreme ?  
**Hype ?**  
Bad for Management ?  
Structured Anarchy ?  
Fun ?  
Unplannable ?      High-Risk ?  
The Death of QS ?

# Is Extreme Programming . . .

. . . a development process  
for hackers ?

. . . chaotic ?

. . . not more than  
structured anarchy ?

# Is Extreme Programming . . .

. . . a development process  
for hackers?



. . . chaotic?

. . . not more than  
structured anarchy?

# Extreme Programming is ...

- > ... a highly disciplined development method
- > For example:
  - Test-Driven Development
  - User stories are implemented in prioritized order

**Is Extreme Programming . . .**

**. . . uncontrollable ?**

**. . . unplannable ?**

**. . . bad for  
management ?**

# Is Extreme Programming . . .

. . . uncontrollable ?

. . . unplanable ?



. . . bad for

management ?



# Extreme Programming . . .

- > . . . meets every deadline!
- > . . . gives feedback about the progress in short cycles
- > Supportive Practices:
  - Planning Game
  - On-Site Customer
  - Tracking

# Planning Game



# Planning Game

- > Customer writes stories
- > Developers estimate stories
- > Customer prioritizes and chooses stories
- > Developers realize the stories in exactly that order

# On-Site Customer

- > This is clearly one of the most problematic practices of XP
  - Customer should be available for questions and feedback
  - e. g. on-phone customer
  - frequent meetings

**Is Extreme Programming . . .**

**. . . high risk ?**

Is Extreme Programming . . .

. . . high risk?



# Extreme Programming is . . .

- > . . . risk-minimizing
- > Do first things first
  - Spike Solutions for potential problems
  - Simple Design
  - Refactoring
  - Continuous integration
  - Collective ownership

# Simple Design

- > Design for today, not for tomorrow
  - often discussed practice
- > Evolve the design !!!
  - It works !!!

"It's easy to have a complicated idea. It's very very hard to have a simple idea."

– Carver Mead



# Refactoring

- > Keeps the software soft !!!
- > This makes evolvable design possible
- > Works even for larger restructurings

# Continuous Integration

- > Move forward in small steps
- > Share code with the team immediately
- > Keep the system running

# Small Releases

- > Work in short iterations
- > Release often to the customer
  - Value for the customer
  - Feedback for the team

"Complex systems that work evolved from simple systems that worked"

- Grady Booch

# Collective Ownership

- > Collective Ownership means that everybody inside the team is responsible
- > Collective Ownership does not mean "No" Ownership !!!

# Pair Programming



# Pair Programming

- > Reviewing is good
- > Pair-Programming is reviewing all the time

"Don't be afraid of pair programming: You're not as good as you think you are, but You're not as bad as you fear."

- Ron Jeffries

**Is Extreme Programming . . .**

**. . . the death of QS ?**

**. . . the end of  
professional testing ?**

# Is Extreme Programming . . .

. . . the death of OS ?

. . . the end of  
professional testing ?





# Extreme Programming is . . .

- > . . . definitely not the death of QS !!!
- > . . . definitely not the end of professional testing !!!
- > **INSTEAD:**
  - XP demands a close collaboration between these groups
  - Let them work in one team

**Is Extreme Programming . . .**

**. . . new ?**

Is Extreme Programming ..

**YES AND NO**

# Extreme Programming is ...

- > ... not new because no single practice is new
- > It is a set of proven techniques
- > But: The key new concept is the combination of them in that way XP promotes

**Is Extreme Programming . . .**

**. . . fixed ?**

Is Extreme Programming . . .



# Extreme Programming is ...

- > ... a starting point !!!
- > Tuning and adapting the process is a central idea

" Start stupid and evolve "

– Kent Beck

# Tuning Workshops

- > Think about your development process after every iteration
- > Adapt and change your development process as necessary



# Is Extreme Programming . . .

. . . only useful for  
small teams ?

. . . unscalable to larger teams ?

Is Extreme Programming . . .

. . . only useful for  
small teams?

. . . inapplicable to larger teams?

**SOMETHING**

# Team Size

- > The standard XP process fits only for small teams
- > But: XP can be adapted to fit the needs of large teams and large organizations

# XP for large teams

- > Jutta Eckstein did an XP project with more than 150 people
  - Take a look at her book for more information on that

**Is Extreme Programming . . .**

**. . . suitable for  
everything?**

# Is Extreme Programming . . .

. . . suitable for  
everything?



# Extreme Programming . . .

- > Can be used for various kinds of projects
- > There are many different adaptations to many different situations
- > But: If the team resists in old habits and old culture . . .

# Extreme Programming

It is all about

**feedback**

and

**change**



**"Our highest priority is to  
satisfy the customer  
through early and  
continuous delivery  
of valuable software"**

**Principles behind the Agile Manifesto  
<http://www.agilealliance.org/>**

**Embrace**

the

**Change**

**Thank you  
for your attention**

Hear more, come to



June 6-10, Garmisch, Germany

<http://www.xp2004.org>

Martin Lippert

[lippert@acm.org](mailto:lippert@acm.org)

<http://www.martinlippert.com/>