

# Agile Testing Practices



## What You Need to Know for Agile Testing Success

Lisa Crispin and Janet Gregory

# Topics: “Life of a Tester in an Iteration”

- How the tester role has changed
- Why and how testing must “keep up”
- Continual improvement



# Agenda – Two-week Iteration

---

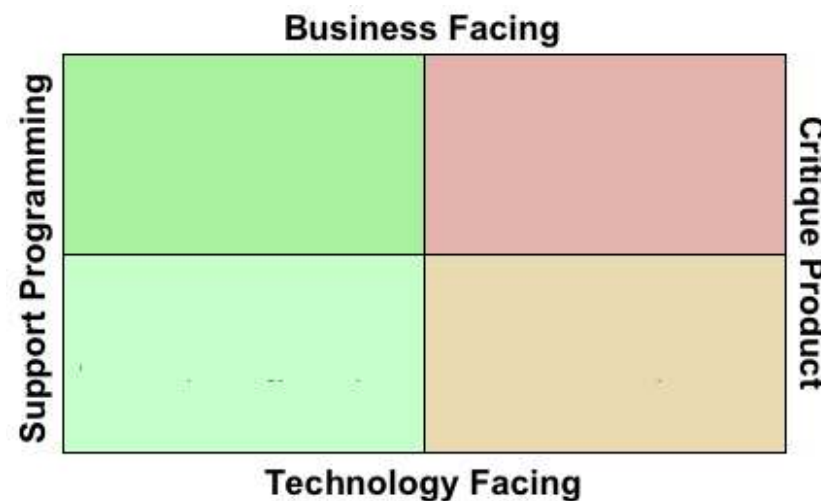
1. Preparing for the iteration (5 min)
2. Laying the foundation, first day or two (15 min)
3. Getting traction, first few days (10 min)
4. Testing/Coding, bulk of iteration (15 min)
5. Wrapping up, last day (5 min)
6. Retrospective, summary (10 min)
7. Q&A: (30 min)



# Jargon

---

- Customer = business expert, product owner, anyone on “customer team”
- Developer test = tests written by developer to assist coding: unit, integration
- Customer test = business-facing tests demonstrating requirements met



# Preparing for Iteration - Stories

---

- Be proactive
- Help customer achieve 'advance clarity' on stories
  - Customers speak with "one voice"
  - Testable stories
  - Story size



# Preparing for Iteration - Tests

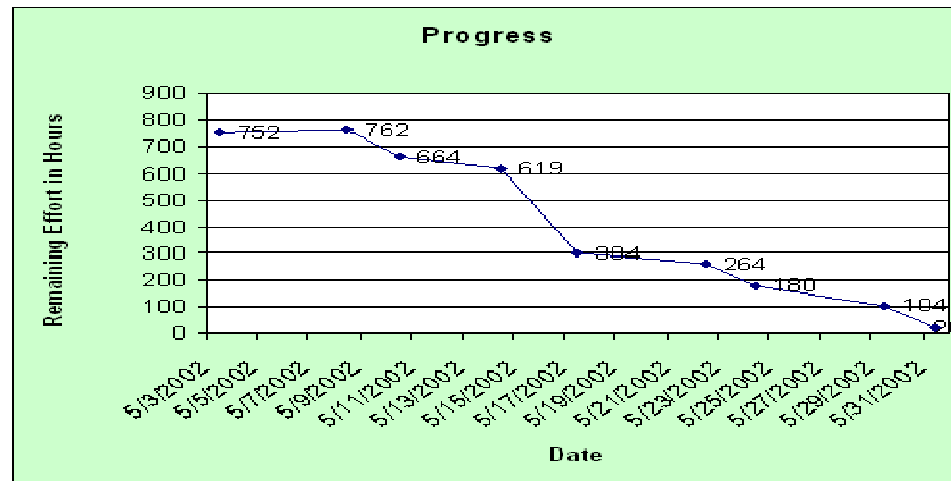
---

- Ask customer to supply & explain examples
- If resources permit:
  - Collaborate with customer to write a couple of high level tests for each story
  - If complex theme, brainstorm in advance with developer team
- If resources don't permit
  - Plan to do these day 1 & 2



# Preparing for Iteration - Visibility

- If you don't already have them
  - ❑ Consider a task board or similar system of tracking tasks
  - ❑ Consider a burn down chart
  - ❑ Brainstorm with development team about test strategies, automation approach, tools



# Laying the Foundation – Iteration Planning

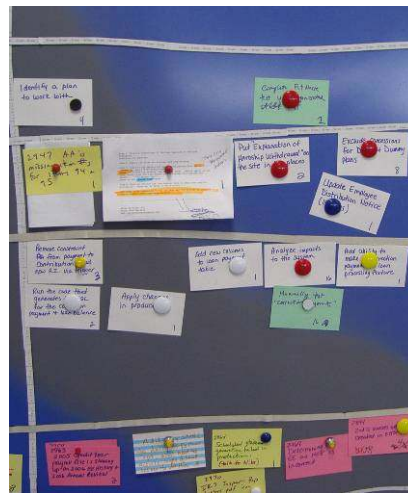
- Product owner participation
  - Or have them explain stories ahead of time
- Product owner prioritization of stories
  - Put them on task / story board in priority order





# Laying the Foundation – Estimation

- Estimate test tasks
  - ❑ Concurrent with development tasks
  - ❑ Get team input for both
  - ❑ Coding tasks include developer test time  
(or make separate card for developer tests)
  - ❑ Consider coding time vs. customer test time



Copyright 2007: Lisa Crispin and Janet Gregory



# Laying the Foundation - Commitment

---

- Commit conservatively
  - Bringing in a story beats dropping a story
  - Must deliver the work to which you commit
  - Remember:
    - the whole team responsible for testing
    - **NO** story is done until tested!



# Laying the Foundation - Collaboration

---

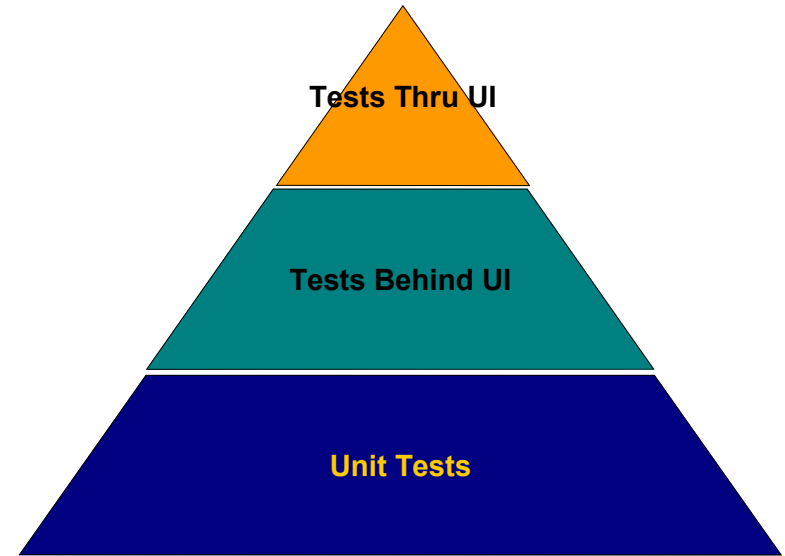
- Collaborate with customer
  - Elicit requirements
  - Find examples
  - Ask open-ended questions
  - Draw on the whiteboard
  - For complex stories, whole team may need to participate
  - Get the 'big picture'
- Consider different points of view: business, programmer, end user
  - Are there conflicts?



# Getting Traction – Preparation

---

- Prepare for testing
- Automation strategy
- Test environments
- Test data
- 3<sup>rd</sup> party involvement?
- What's in scope?
- How might story impact larger application / system?



# Getting Traction – High Level Tests

---

- Write high level test cases
  - Examples of desired and undesirable behavior
  - Use variety of tools:
    - lists, bullets, tables, wiki, or anything easy to access and maintain (probably not Word!)
    - Should give ‘big picture’
- Review with customer as needed
- Review high level tests with programmers
  - Do they see the ‘big picture’?



# Testing & Coding Time

---

- Focus on one story at a time
- Use tests to drive development
- Lots of examples!
- Whiteboard discussions
- Ask questions
- Not a conduit between programmers and customers
- But do facilitate, encourage discussions



# When Coding Starts

---

- Write detailed test cases from examples
  - Make them executable if possible
- Write simple, happy path tests to automate
  - Share them with the programmers
- Once simple tests passes, write more test cases
  - Start with happy path
  - Then move towards edge cases
- Identify possible exploratory scenarios



# Testing/Coding: Whole Team Approach

---

- Anyone can pick up a testing task
- Run automated regression tests regularly
- Ideally on continuous build
- Minimally on daily build
- If automated tests are not available, engage the whole team in manual testing





# Testing/Coding: Don't sit and wait!

---

- Is any testable part of a story ready?
  - Test with behind-the-GUI tool such as FIT?
  - Or other harness to bypass GUI
- Pair with programmers
  - Test together before check-in
  - Show them issues
  - Ask questions
  - Bugs found here are cheap and easy to fix



# Testing/Coding: Collaboration

---

- Pair with customers
  - Show customer early
  - Does delivered feature meet expectation?
- Other team members, e.g.
  - Work with DBA to understand database
  - Work with sys admin to get better messaging from continual build



# Wrap it up: Last Day

---

- Whole team responsible for delivering tested code
- Show customers what you delivered
  - Could be on first day of next iteration
- This should not be a long or difficult day!
- Standup in afternoon if needed
- Release to staging environment
- Plan production release



# “Keeping Up”

---

- Ensure stories are testable
- Estimate test tasks with development tasks
  - The team understands the effort
- Keep track of “big picture”
  - Consider how each story affects rest of application
  - Does it affect other systems?
  - Don’t forget end to end tests
- Collaborate with customer to determine minimum success criteria



# “Keeping Up” - Automate

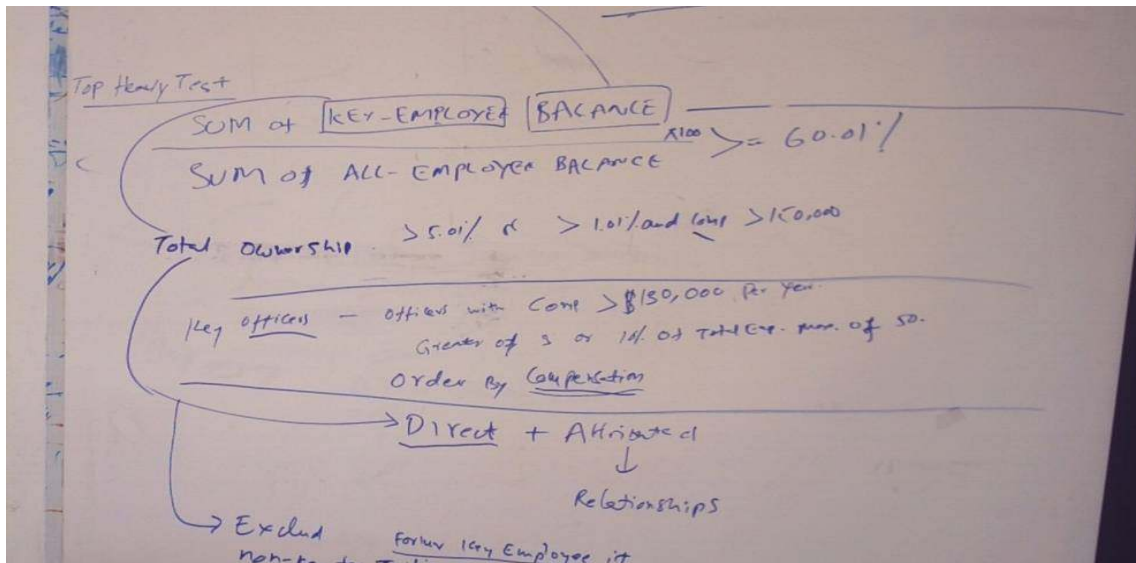
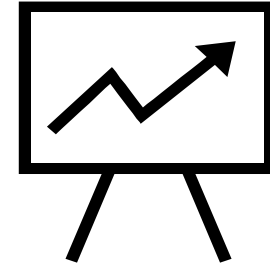
---

- Automate all regression tests
  - Consider collaborative approaches such as FIT / FitNesse
  - Consider ROI
  - Automate at lowest level possible
- Automate time consuming tasks
  - Calculations, parsing files...
- Make time for exploratory testing
  - Involve customer team if possible



# “Keeping Up” - Focus

- Keep the team’s focus
  - Big visible charts
  - Focus on one story at a time
- Use lots of examples
- And whiteboard discussions



# Team Retrospective - Continuous Improvement

- What's in your way
- Solve problems as a team
- Review last iteration's "stop/start/continue"
  - Did we improve on important items?
- What should we start doing?
- What should we stop doing?
- What should we continue doing?
  - Continue until it's a habit



# Team Retrospective – Analyze & Action

---

- Is there one single limiting factor?
  - something holding you back
  - something causing a blown iteration
  - Anything causing other big problems
- Action items
  - Focus on 1 or 2 things to improve
  - Use big visible charts
  - Create task cards for future iterations
  - Need new resources?
    - E.g.. new team member with particular expertise





# The Agile Tester Role - Review

---

- Get moving! Be proactive!
  - Don't sit and wait for things to come to you
- Who does what testing?
  - Understand the “Whole Team” approach
- Collaboration is key
  - Customers/product owners/business experts
  - Developers
  - Other team members



# Questions?

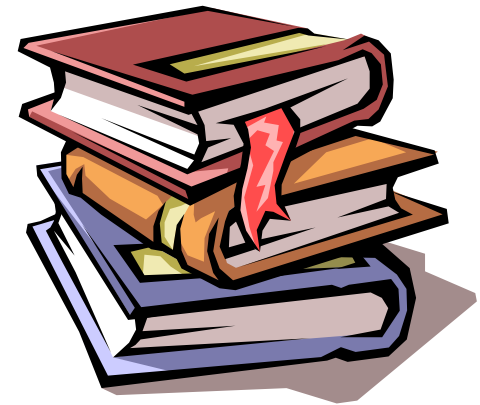
---



# Agile Testing Resources

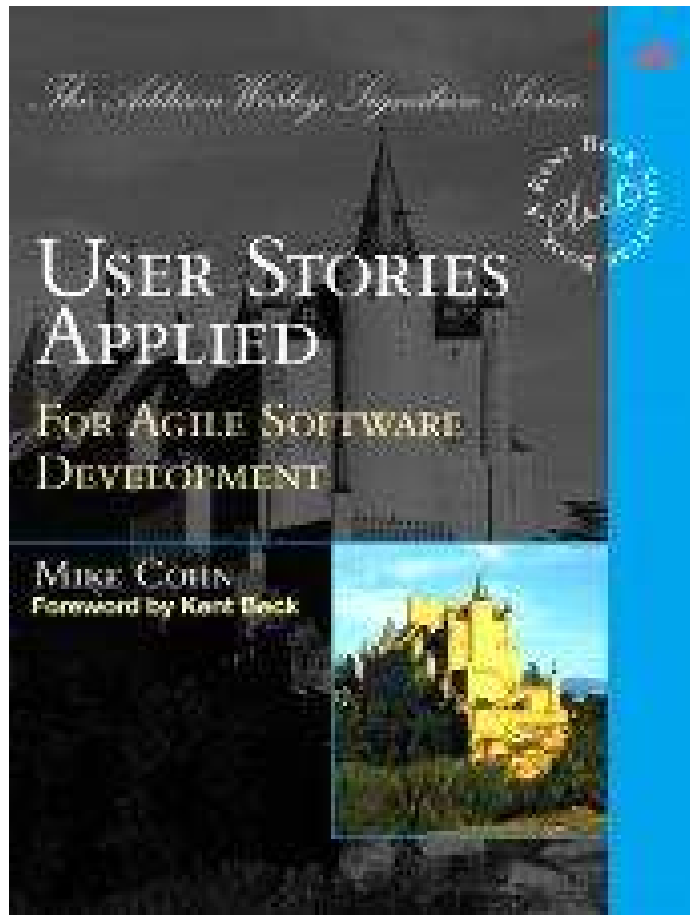
---

- [lisa.crispin.home.att.net](http://lisa.crispin.home.att.net)
- [www.agilealliance.org](http://www.agilealliance.org)
- [www.testing.com](http://www.testing.com)
- [agile-testing@yahoogroups.com](mailto:agile-testing@yahoogroups.com)
- [www.fitnessse.org](http://www.fitnessse.org)
- [webtest.canoo.com](http://webtest.canoo.com)
- [fit.c2.com](http://fit.c2.com)



# Customer Team

---



*User Stories  
Applied*

by Mike Cohn

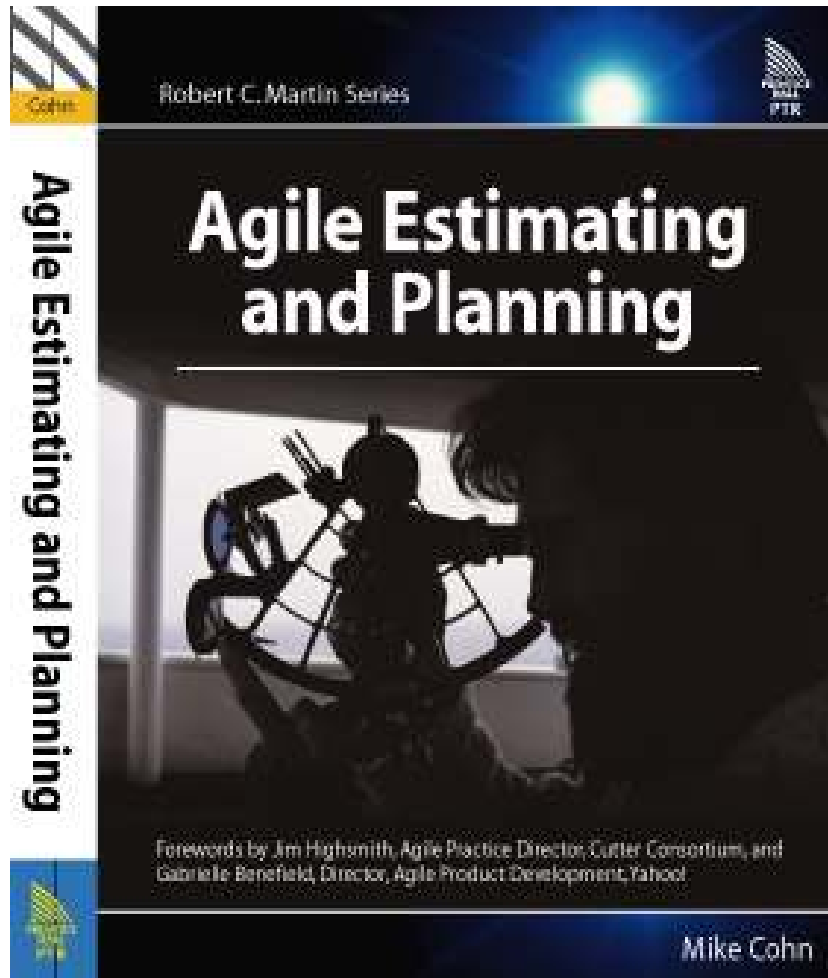
Available on  
Amazon



Copyright 2007: Lisa Crispin and Janet Gregory



# Estimating



*Agile Estimating  
and Planning*

By Mike Cohn

Available on  
Amazon

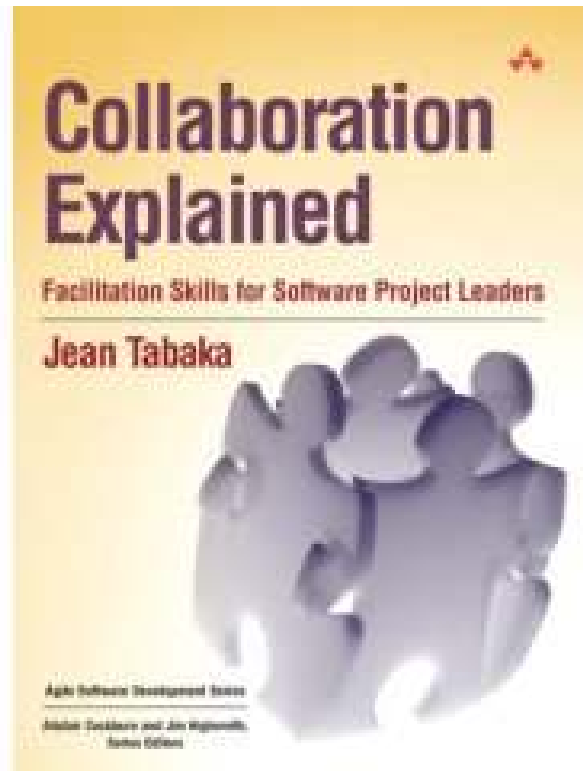


Copyright 2007: Lisa Crispin and Janet Gregory



# Collaboration

---



*Collaboration Explained :  
Facilitation Skills for  
Software Project Leaders*

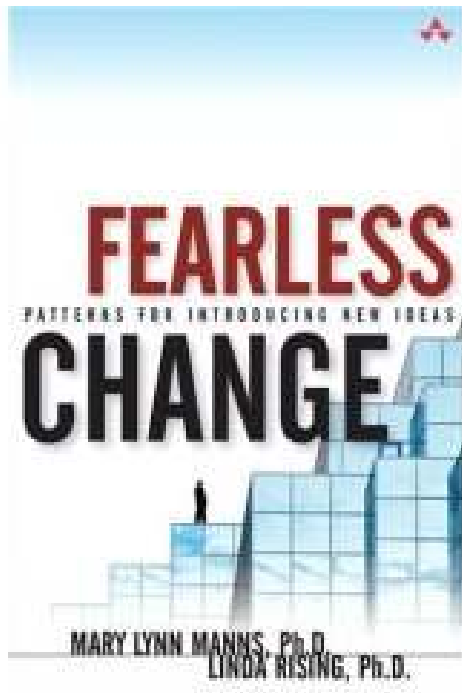
By Jean Tabaka

Available on Amazon



# Implementing Change

---



*Fearless Change: Patterns  
for introducing new ideas*

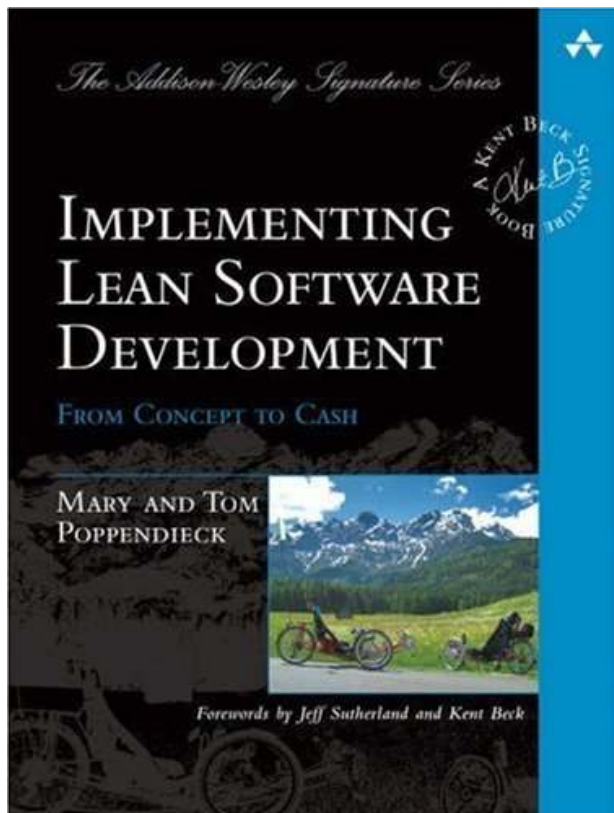
By Linda Rising and  
Mary Lynn Manns

Available on Amazon



# Lean Development

---



*Implementing Lean Software Development: From Concept to Cash*

By Mary and Tom Poppendieck

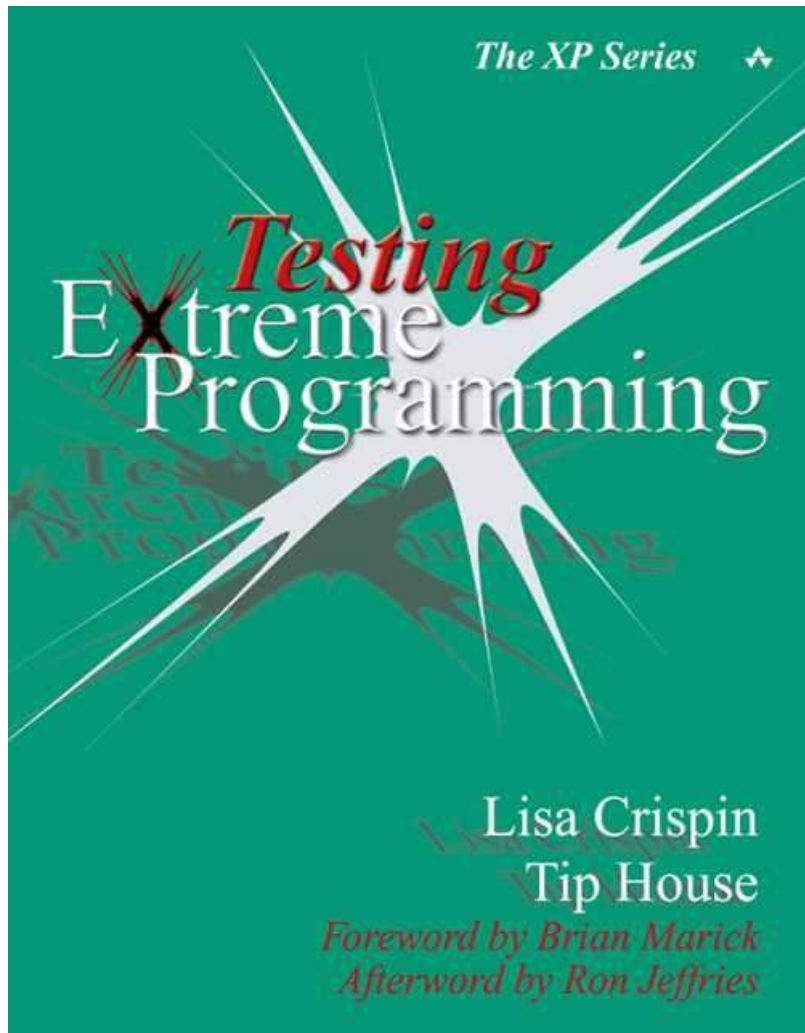
Available on Amazon





# Agile Testing Resources

---



## *Testing Extreme Programming*

By Lisa Crispin and  
Tip House

Available on  
Amazon



Copyright 2007: Lisa Crispin and Janet Gregory

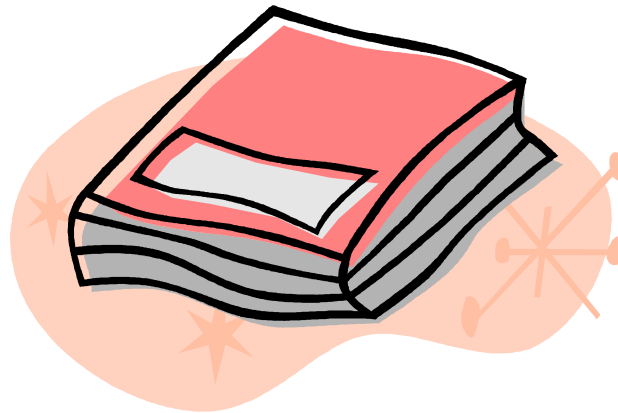


# Coming in 2008!

---

## *Agile Testing*

By Janet Gregory and Lisa Crispin



Copyright 2007: Lisa Crispin and Janet Gregory

