

# Cover Your Testing Bases Using the Agile Testing Quadrants



## Agile Denver

Lisa Crispin

*With Material from Janet Gregory*

# Introduction

---

- Tester on agile teams since 2000
- Current team:
  - Has delighted customers
  - Delivers value to production every 2 weeks
  - Drives development with tests, examples
  - 100% regression test automation



Copyright 2009: Lisa Crispin



# Agenda

---

- How many complete all testing tasks by end of your iteration?
- We'll look at the Quadrants to help us know:
  - What tests do we need to do? When?
  - Who does them? What tools to use?
- Planning your Testing Strategy
- References



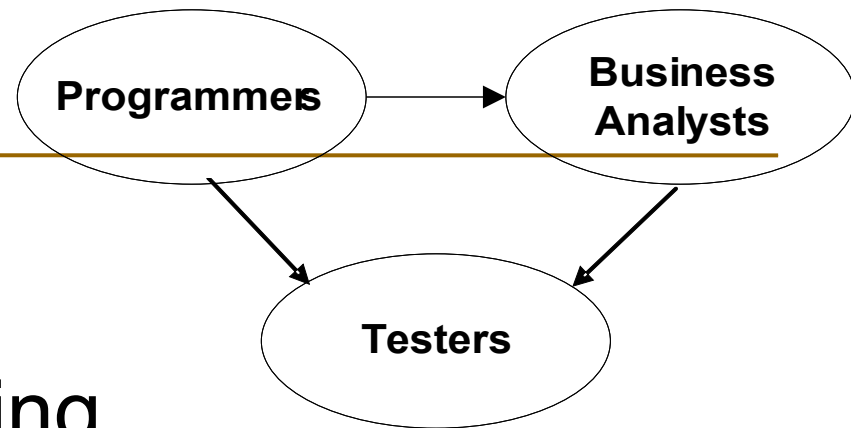
# Why Do We Test?

---

- To find bugs
- Make sure system is reliable, usable
- To learn about the application
- Guide coding
- Feedback to future stories
- Check for doneness
- Manage technical debt



# Traditional Testing



## Phased / Gated Projects

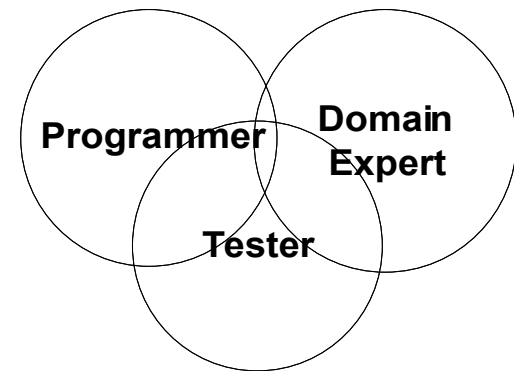
- Testers test – after coding
- Works from requirements documents
- Can be a fight to be involved up front
- Organizational culture – silos
- Defect tracking instead of conversations
- Different language than developers

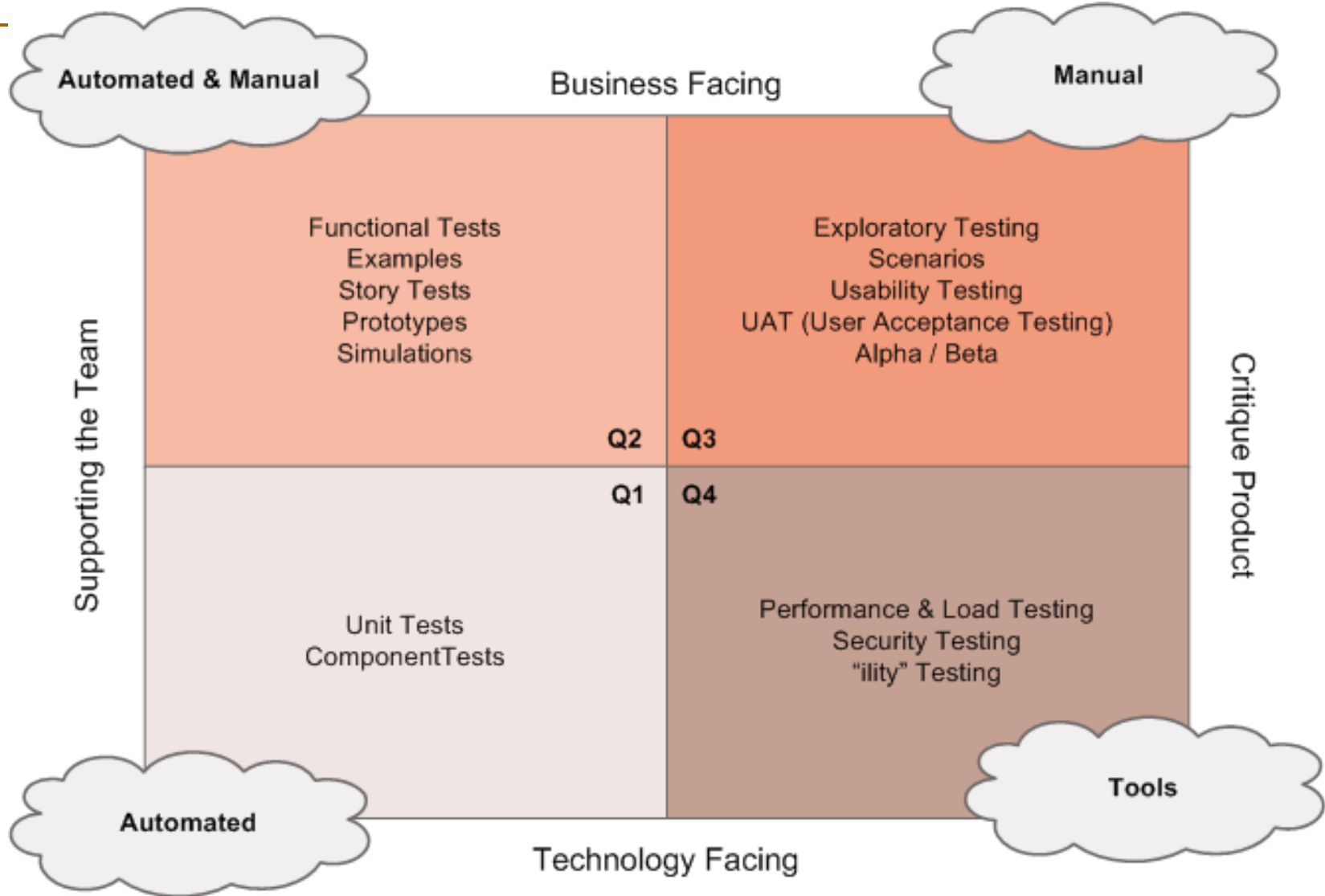


# Agile Testing

---

- Principles / Values
  - Communication, collaboration, interactions
- Everybody tests, is test-infected
- Whole team responsible for quality
- Tester's role in team
  - Help uncover hidden assumptions
  - Provide feedback
  - Elicit and clarify requirements
  - Drive development with examples





# The Quadrants .....

---

- Can be used as a communication tool
  - To the project team
  - To management
  - To explain testing in a common language
  - Tests drive coding
- Emphasize whole-team responsibility
  - Focus on collaboration
  - Whole team participation

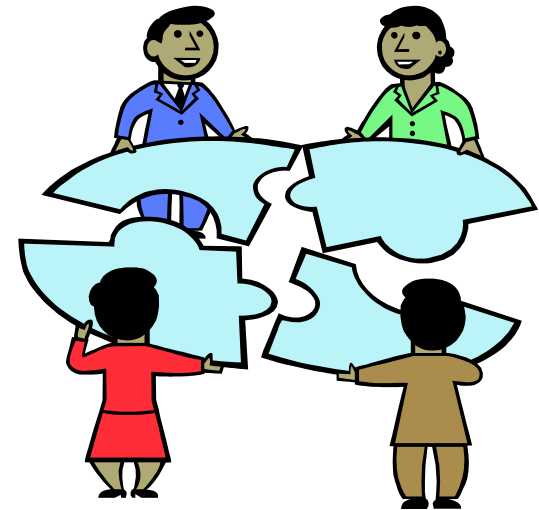


# Use to define 'Doneness'

---

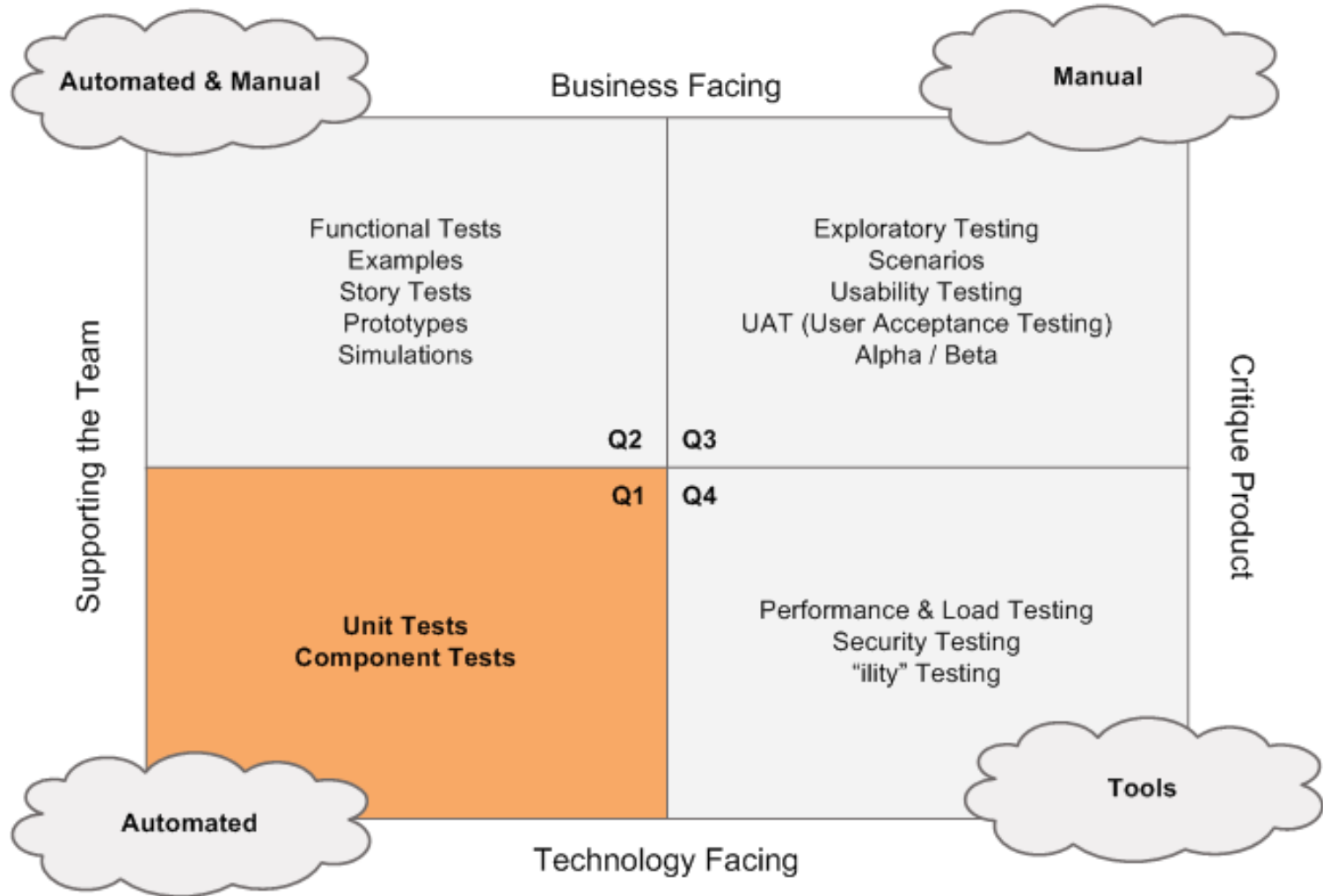
## Defining “doneness” for release readiness

- No story is done until tested
- Customer requirements captured as passing tests
- Automated regression tests
- Delivers value
- “Doneness” in all quadrants



# Quadrant 1

## Agile Testing Quadrants



# Types of Tests in Q1

---

- **Unit Tests**
  - Tests developer intent - program design
  - Tests a small piece of code
  - Makes sure it does what it should
- **Component Tests**
  - Tests architect intent – system design
  - Tests that components work together correctly

Technology-facing Tests that Support the Team



# Reasons / Benefits for Q1 Testing

---

- Focus on internal code quality
- Go faster, do more
  - Safety net, refactoring
- Builds quality in, design for testability
- Instant, continual feedback
- TDD increases confidence in design
- Foundation for regression test automation

Technology-facing Tests that Support the Team



# Example Tools for Q1

---

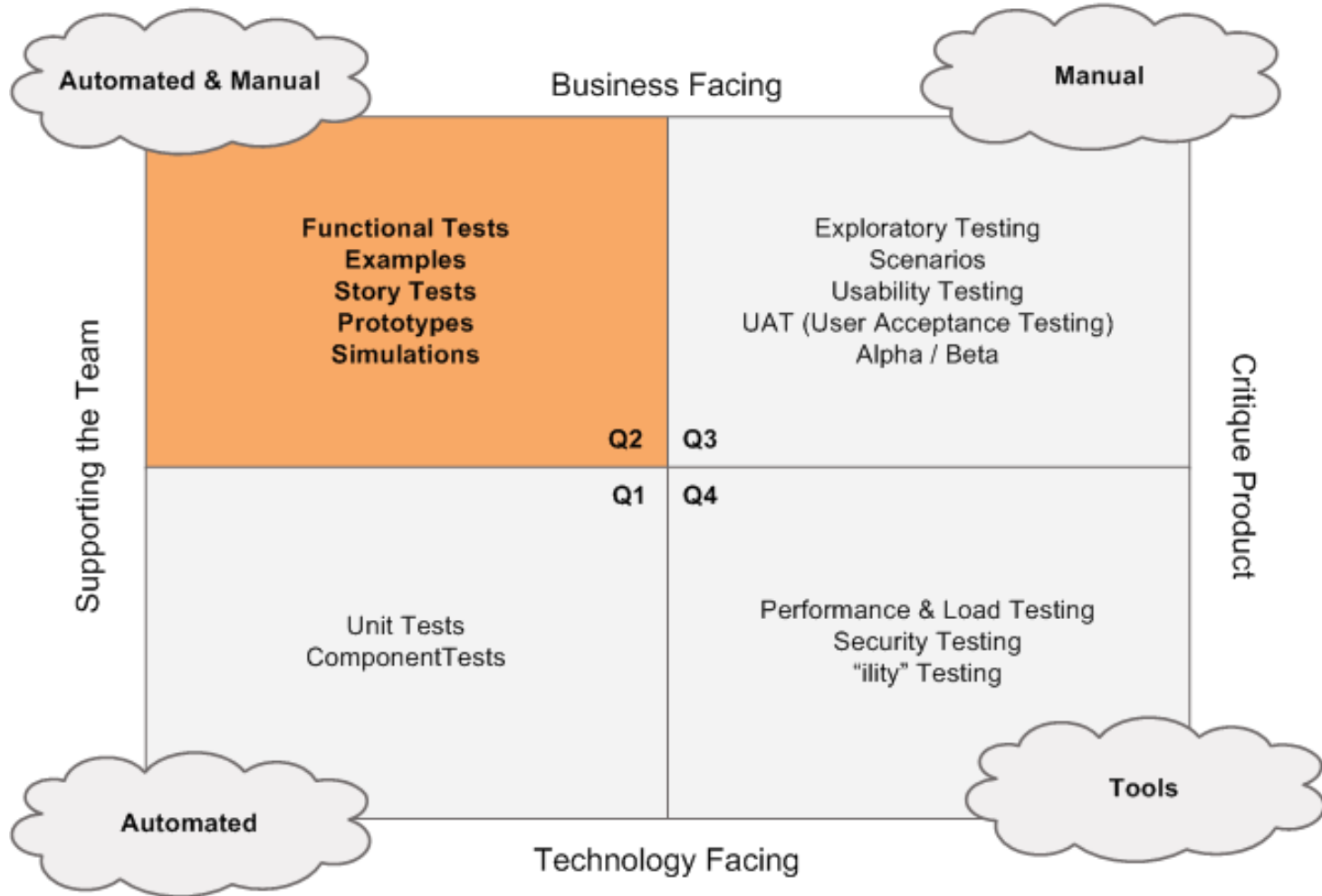
- Source code management, build tools
  - SVN, Hudson, CruiseControl, ant
- IDEs
  - Eclipse, IntelliJ, NetBeans
- Unit test tools
  - Xunit
  - BDD – NBehave, RSpec, easyB
  - Mocking tools – EasyMock, JMock

Technology-facing Tests that Support the Team



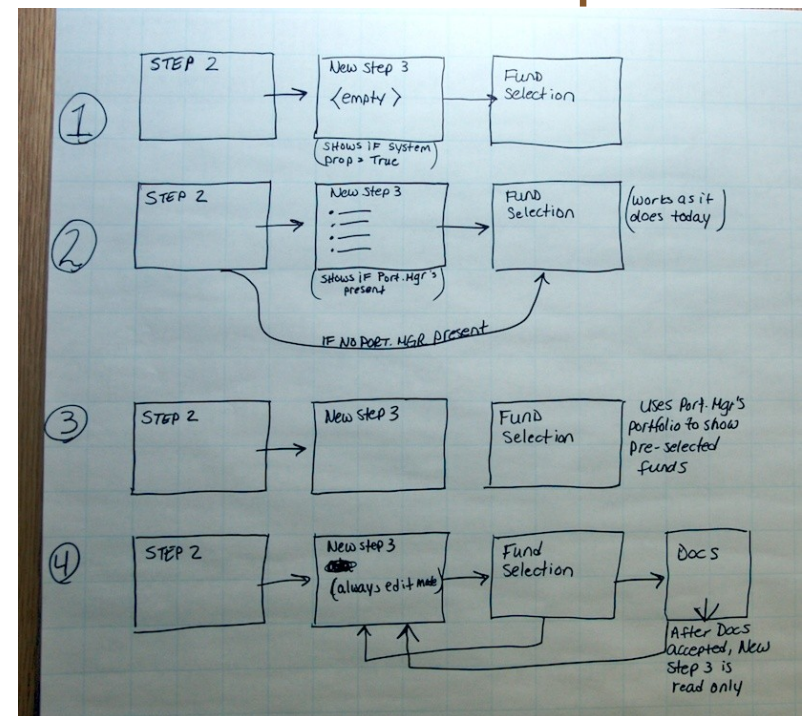
# Quadrant 2

## Agile Testing Quadrants



# Types of Tests in Q2

- Elicit requirements, examples
- Tests to drive coding
  - allows developers to code until the tests pass
- User Experience
  - wire frames
  - Mock-ups / prototypes
- Pair Testing



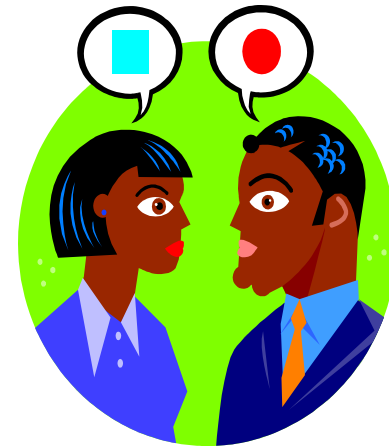
Business-facing Tests that Support the Team



# Reasons / Benefits for Q2 Testing

---

- Know what to code, value needed
- Help customers achieve advance clarity
- Help collaborate
- Executable tests = quick feedback
- Focus is external quality
- Know when we're done



Business-facing Tests that Support the Team



# Toolkit – Turning Examples into Tests

- Fit/FitNesse
  - ATDD
  - Promotes collaboration, communication

Build Employees Fixture												
userid	dob	doh	doe	dot	directOwnerPct	lookbackTotalOwnerPct	lookbackAnnualComp	annualComp	deferral	eligibleComp	match	addl
1001	01-01-1950	01-01-1993	01-01-1994	null	0	0	101500.00	102500.00	16000.00	102500.00	16000.00	true
1002	01-01-1960	01-01-1993	01-01-1994	null	4	3	102500.00	102500.00	13000.00	102500.00	13000.00	true
1003	01-01-1960	01-01-1993	01-01-1994	null	5.01	5.01	30000.00	30000.00	7500.00	30000.00	7500.00	true
1004	01-01-1960	01-01-1993	01-01-1994	null	10	10	20000.00	30000.00	3000.00	30000.00	3000.00	true
1005	01-01-1960	01-01-1993	01-01-1994	null	8	0	40000.00	40000.00	8000.00	40000.00	8000.00	true
1006	01-01-1960	01-01-1993	01-01-1994	null	5.01	0	150000.00	150000.00	13000.00	150000.00	13000.00	true
1007	01-01-1960	01-01-1993	01-01-1994	null	0	0	100000.00	100000.00	0	100000.00	0	true
1008	01-01-1960	01-01-1993	01-01-1994	null	0	0	40000.00	50000.00	3000.00	50000.00	3000.00	true

## OPERATE ON INPUT BY RUNNING ADP TEST

Operate Adp Test Fixture
operatel
true

## MAKE ASSERTIONS ABOUT ADP TEST RESULTS

Check Employee Fixture				
userid	isHce?	isEligible?	adr?	acr?
1001	true	true	12.682927	15.61
1002	true	true	12.682927	12.68
1003	true	true	25.00	25
1004	true	true	10.00	10
1005	true	true	20.00	20
1006	true	true	8.666667	8.67
1007	true	true	0	0
1008	false	true	6	6



# More Example Q2 Tools

---

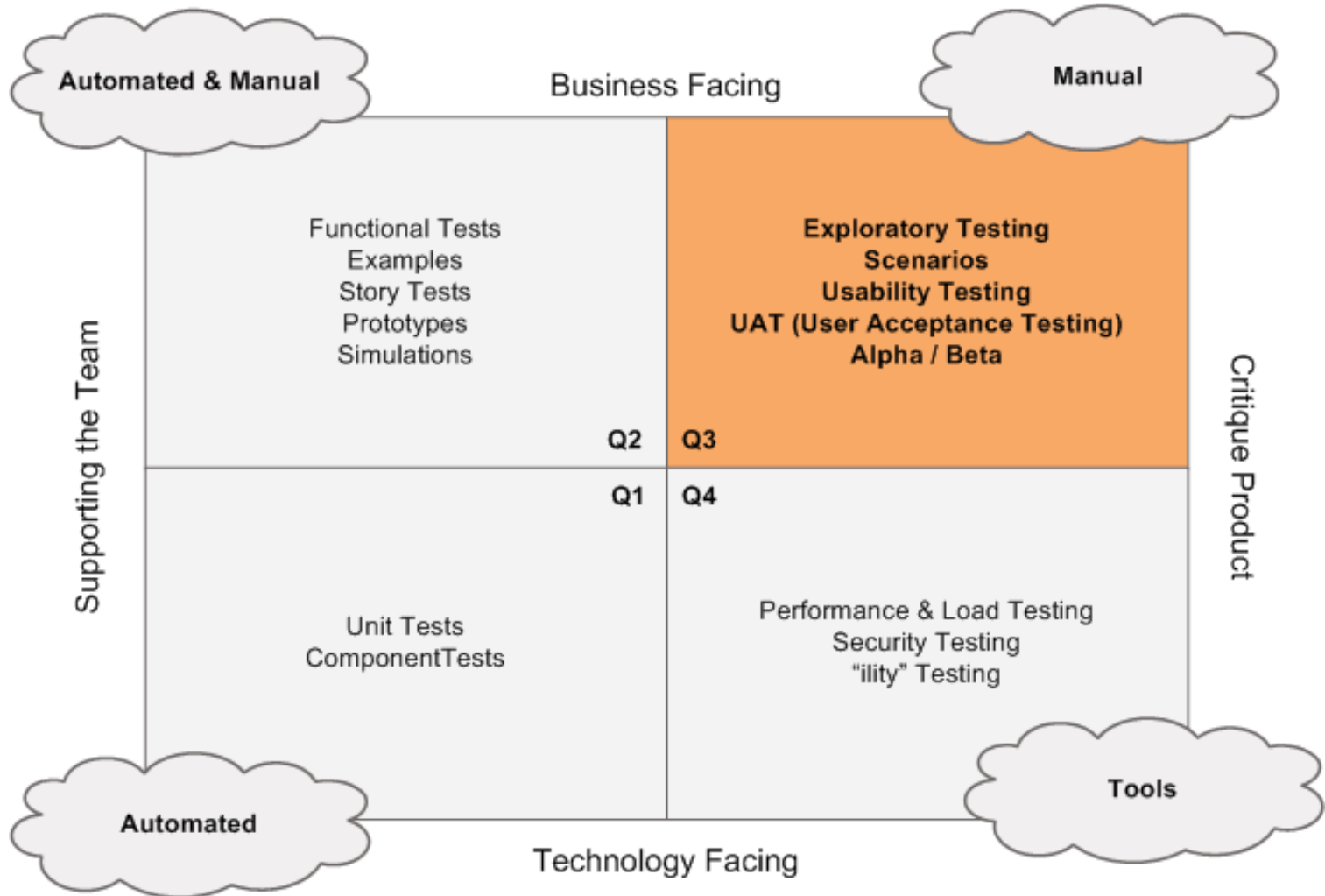
- Paper and whiteboards
  - Prototypes, Wizard of Oz testing, mockups
- BDD – RSpec, Cucumber, Rasta, easyB, Twist
- GUI test tools – simultaneous with coding
  - Thin slice approach
    - Watir/WatirCraft & related frameworks
    - Selenium RC

Business-facing Tests that Support the Team



# Quadrant 3

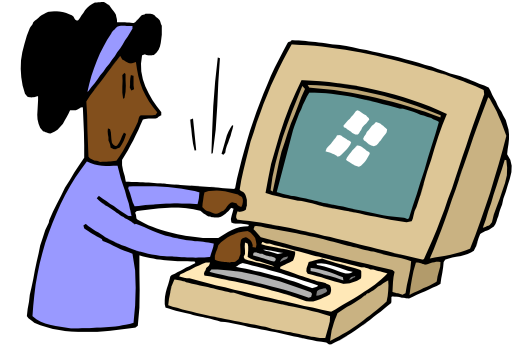
## Agile Testing Quadrants



# Types of Testing in Q3

---

- **Exploratory Testing**
  - Learning → Q1 & 2 tests
  - Collaborate with customers
  - Developer-tester collaboration
- **Usability testing**
  - understand end users, personas
  - who will be using the system, how
- **Demos, reviews**
  - During and after coding



Business-facing Tests that Critique the Product



# Reasons / Benefits for Q3 Testing

---

- Feedback to Quadrants One and Two
- Evaluate product – is value there?
- Recreate actual user experiences
- Learn as you test
- Context
  - What works for your situation
  - “It depends”
  - A tool, not a rule
- Constructive, critical thinking



Business-facing Tests that Critique the Product



# Example Tools for Q3

---

- Your brain, eyes, ears, fingers, gut...
- Use Q1 and Q2 tools to set up data, scenarios
- Leave the important work for your manual testing

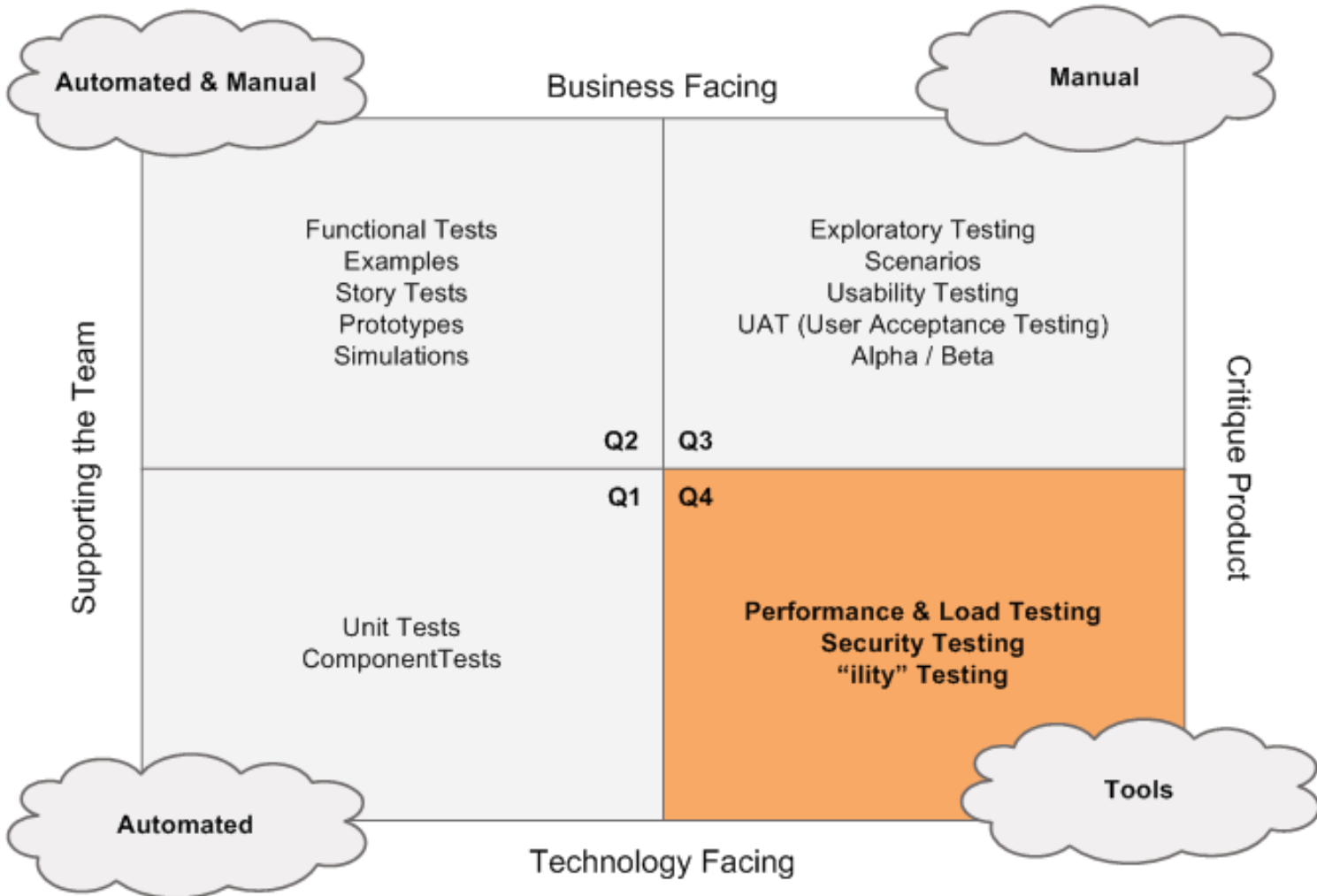


Business-facing Tests that Critique the Product



# Quadrant 4

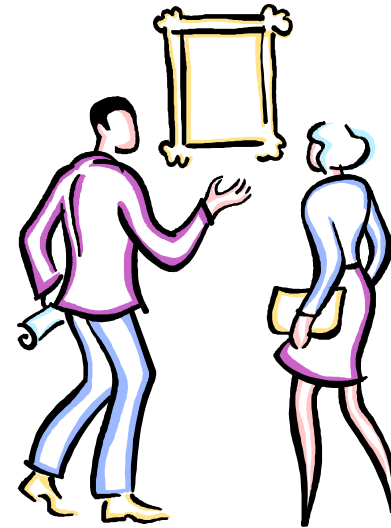
## Agile Testing Quadrants



# Types of Tests in Q4

---

- Nonfunctional or parafunctional tests
- “ility” testing
- Performance, scalability, stress, load
- Memory management
- Security testing
  - Roles & permissions
  - System ‘hacking’
- Data migration
- Infrastructure Testing



Technology-facing Tests that Critique the Product



# Reasons / Benefits for Q4 Testing

---

- “Non-functional” requirements may be higher priority than “functional”
  - Performance
  - Stability
  - Security
- Specialized expertise might be needed
  - Collaborate to transfer skills
- Makes the ‘finished’ product
- Does your application deliver the ‘right’ value

Technology-facing Tests that Critique the Product



# Example Tools for Q4

---

- Programmers can write load test harnesses using unit tests
- Open source and vendor tools
  - Jmeter, the Grinder
  - WebLoad
  - External services
- Monitoring tools
  - JConsole, Jprofiler
- Native database tools
- Specialized tools, external security audits



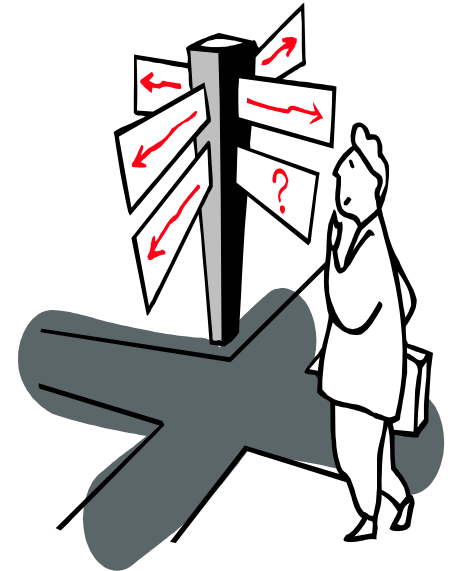
Technology-facing Tests that Critique the Product



# Planning Your Test Strategy

---

- Consider all four quadrants
  - Team plans, prioritizes
  - Work in small chunks
- Consider scope, priorities, risks
  - Limit scope so testing finished
- Tools that solve the problem
- Whole-team approach
- Guided by agile principles
- Use lessons learned to improve



# Questions?

---



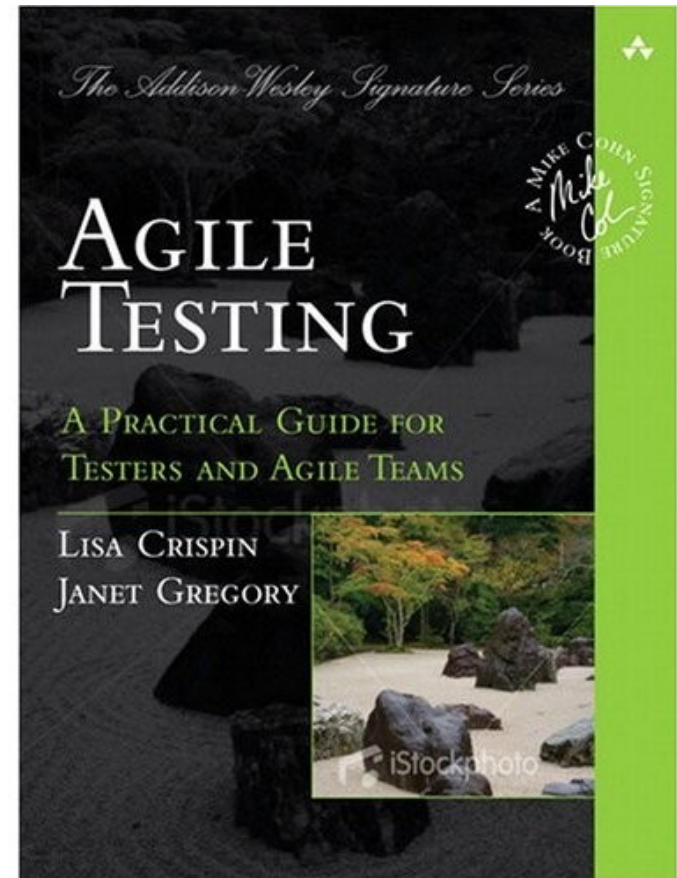
# Now Available

---

## *Agile Testing: A Practical Guide for Testers and Agile Teams*

By Lisa Crispin and Janet Gregory

[www.agiletester.ca](http://www.agiletester.ca)



Copyright © 2009: Lisa Crispin



# Some Agile Testing Resources

---

- [lisacrispin.com](http://lisacrispin.com)
- [janetgregory.ca](http://janetgregory.ca)
- [agilealliance.org](http://agilealliance.org)
- [exampler.com](http://exampler.com)
- [agile-testing@yahoogleroups.com](mailto:agile-testing@yahoogleroups.com)
- [testobsessed.com](http://testobsessed.com)
- [testingreflections.com](http://testingreflections.com)

