



Testing Techniques for Agile Testers

Belgium, 2009

With Material from Lisa Crispin

Janet Gregory
DragonFire Inc.

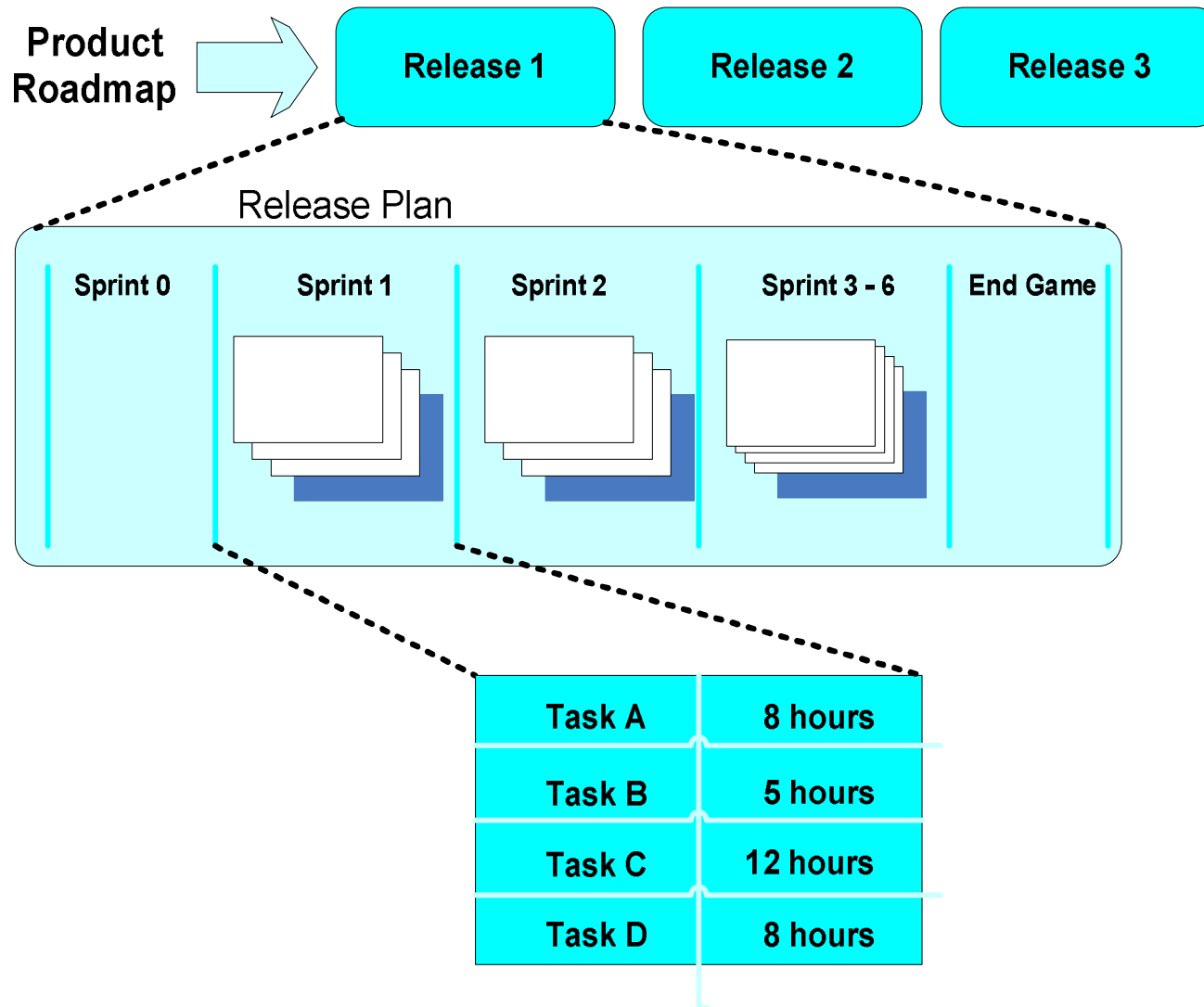


Topics

- Test Planning - Levels
- Testing and Collaboration
- The Agile Testing Quadrants
- Automation
- Planning Your Test Strategy
- References

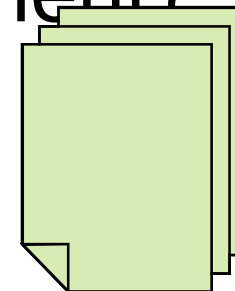


Levels of Planning – Product, Release, Iteration



Release Planning ... Test Planning

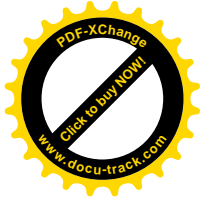
- Consider:
 - Scope / Features
 - Priorities
 - Risks
- Budget time:
 - For building test infrastructure
 - For finding test tools
- What do you need to document?
 - Consider its usefulness



Iteration Planning ... Test Planning

- Consider:
 - Doneness for each feature
 - Priorities – which story to detail first
 - Risks
- Budget time:
 - For defect fixing & retesting
- What do you need to document?
 - At minimum, the acceptance test(s)





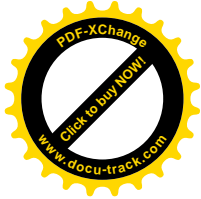
The Importance of Collaboration

- The Star Experiment
 - Gerald Weinberg and Donald Gause,
Exploring Requirements
- What is the right answer?



How Many Points?

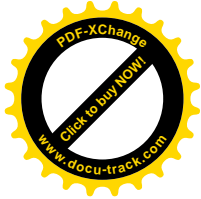




Collaboration

- Collaboration means ...
 - ▣ working together
- Communication means
 - ▣ sharing ideas, information, decisions & solutions
- Tools for collaboration
 - ▣ Feedback
 - ▣ Interaction
 - ▣ Visibility - allows people to talk about it.





Collaboration Discussion

1. Split into teams, and discuss
2. Present your findings

TWO QUESTIONS....

1. How does your teams communicate?
2. What are the collaboration tools you use?



What are the Agile Testing Quadrants?

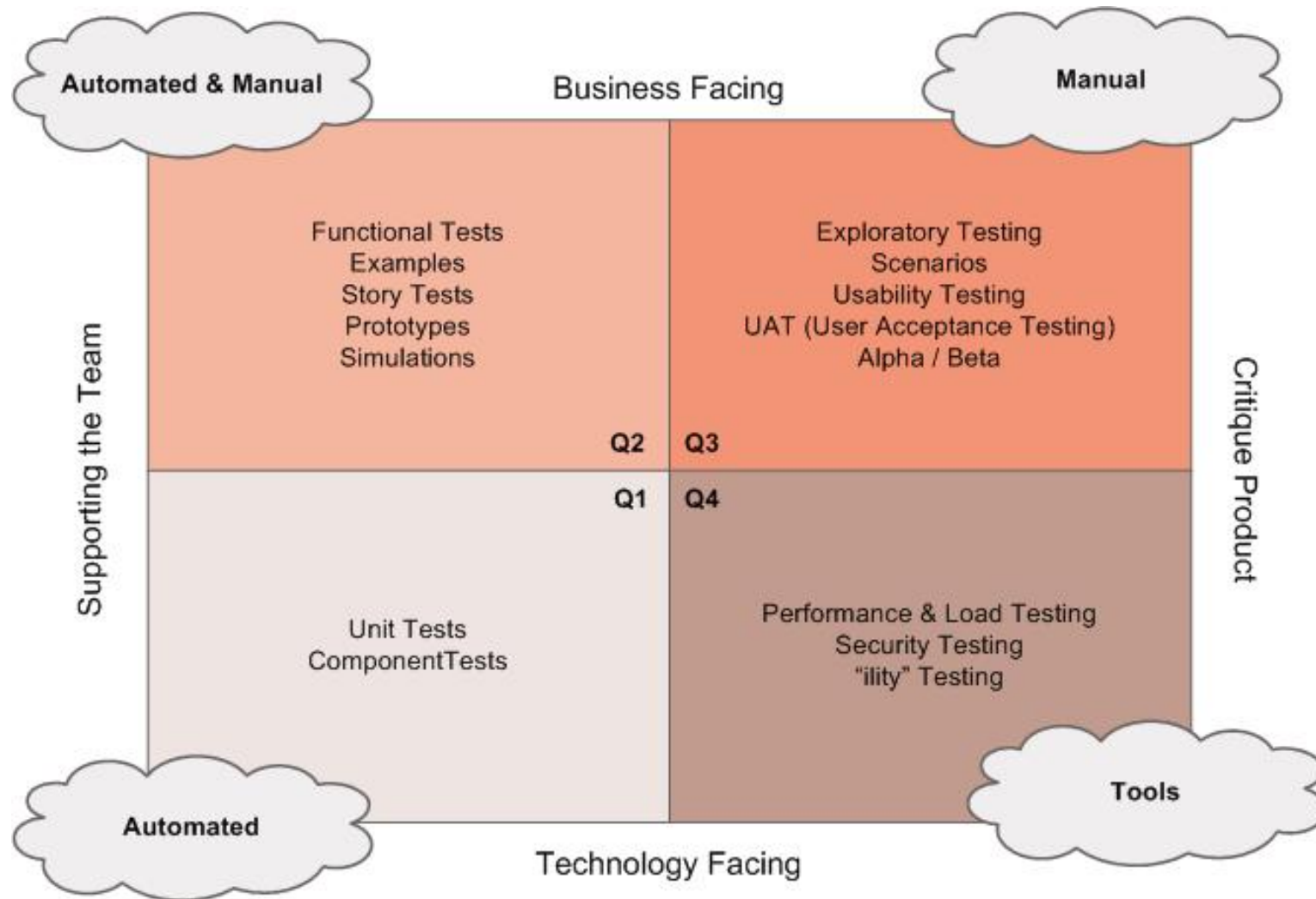
- Method for classifying tests
- Looks at the purpose of the tests
 - not the when
- Tests may cross boundaries
- Tests that are created to support the team,
 - often turn into regression tests

.... Let's look at the picture!



The Agile Testing Quadrants

Introduced by Brian Marick



The Quadrants

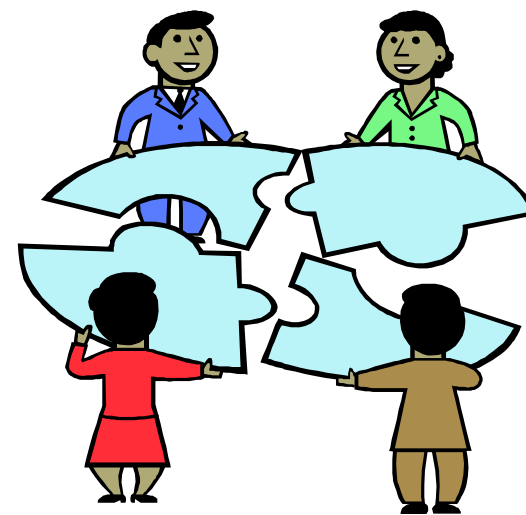
- Can be used as a communication tool
 - ▣ To the project team
 - ▣ To management
 - ▣ To explain testing in a common language
- Emphasize whole-team responsibility
 - ▣ Focus on collaboration
 - ▣ Whole team participation



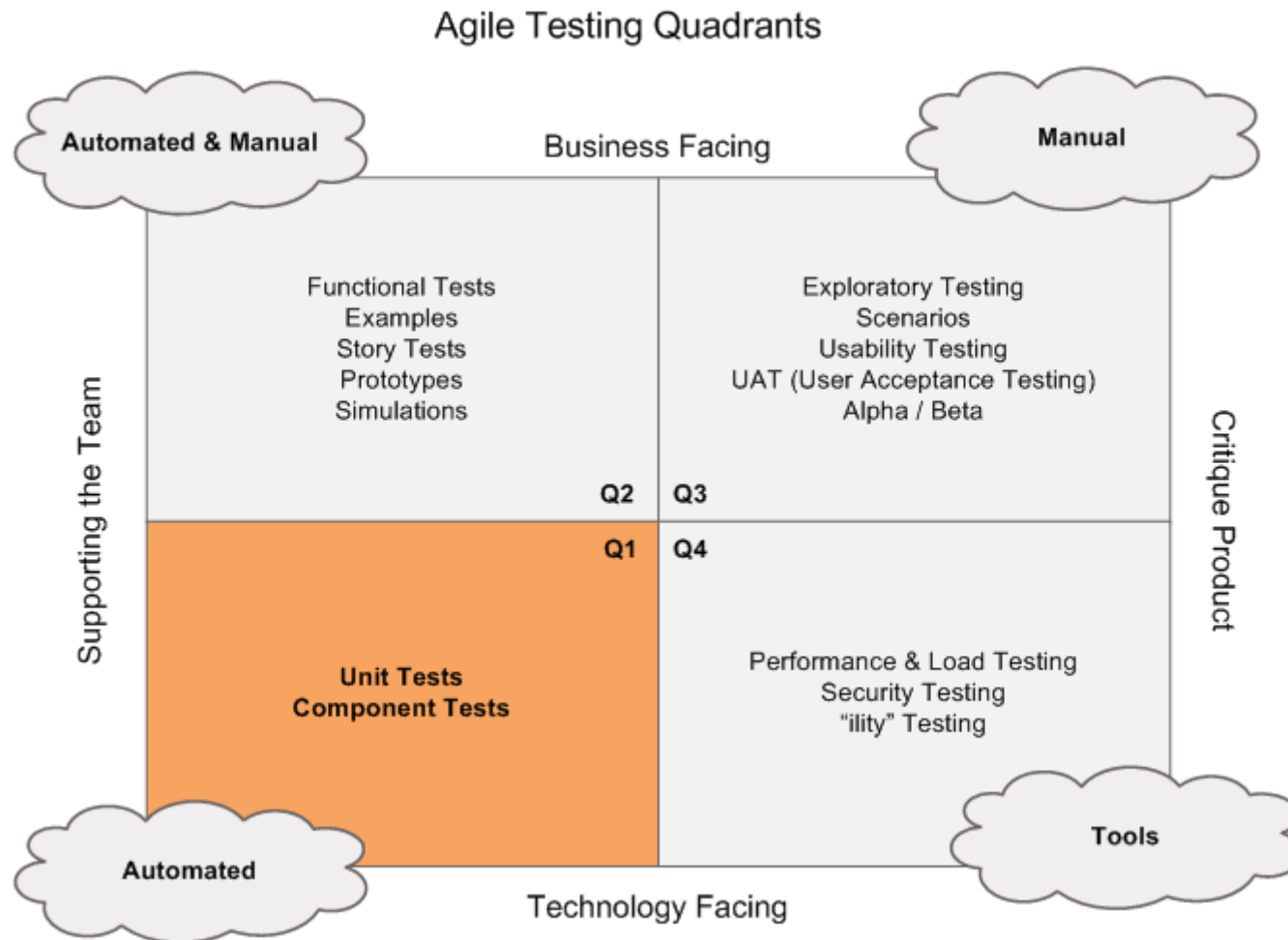
Use to define 'Doneness'

Defining "doneness" for release readiness

- ❑ No story is done until tested
- ❑ Customer needs captured as passing tests
- ❑ Automated regression tests
- ❑ Delivers value
- ❑ "Doneness" in all quadrants



Quadrant 1



Benefits of TDD

- Focus on internal code quality
- Builds testability into code
- Increases confidence in design
- Provides instant feedback to developers

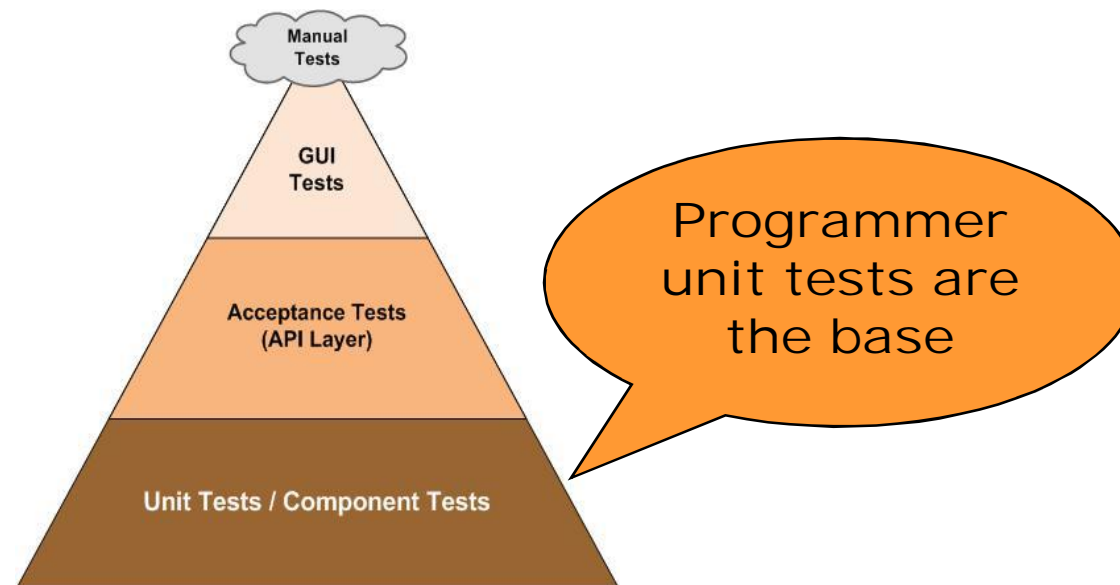


Technology-facing Tests that Support the Team



Other Benefits of Q1 Testing

- Start of building quality into the product
- Provides refactoring support
- Forms the foundation of automation suite

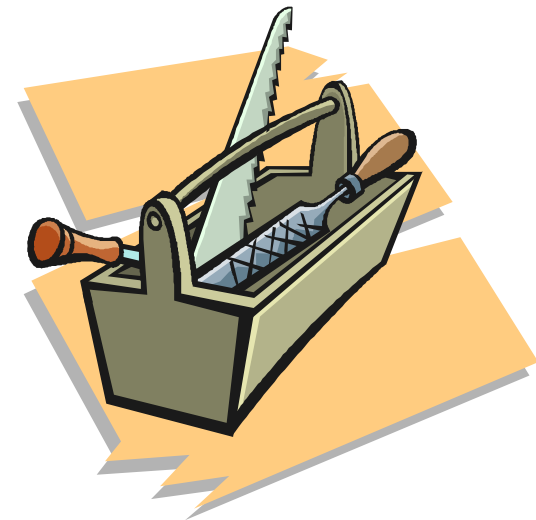


Technology-facing Tests that Support the Team



Q1 Toolkit

- Source code management
- Integrated development environment
 - compile, debug, build GUI, refactor
- Build tools
 - eg. CruiseControl, Hudson
- Unit test tools
 - xUnit
 - Mocking tools
- All needed to enable collaboration



Technology-facing Tests that Support the Team



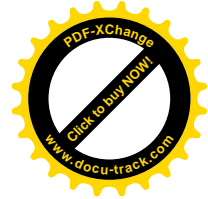


Discussion

What can testers do to help the developers?

Technology-facing Tests that Support the Team

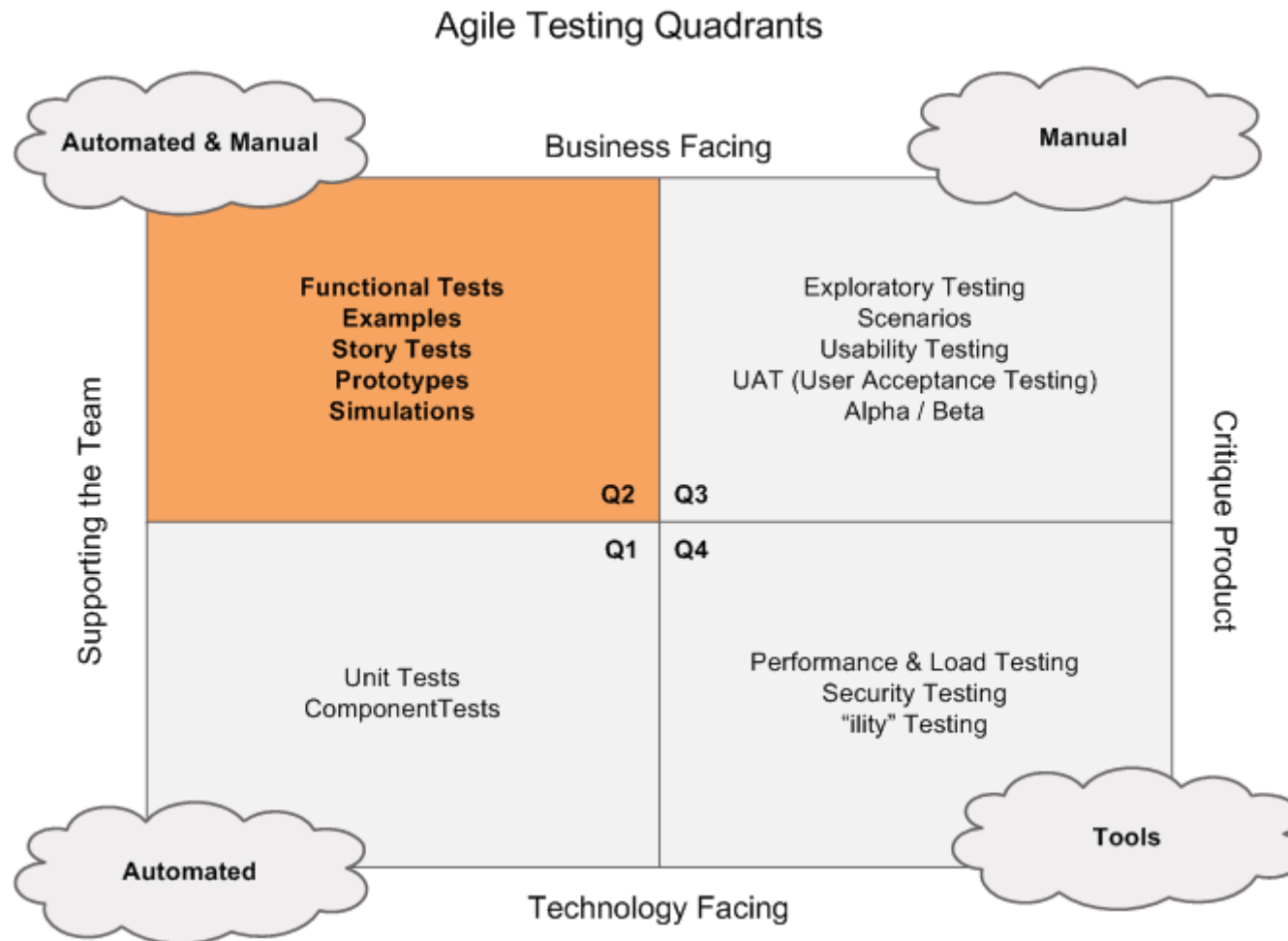




Helping developers



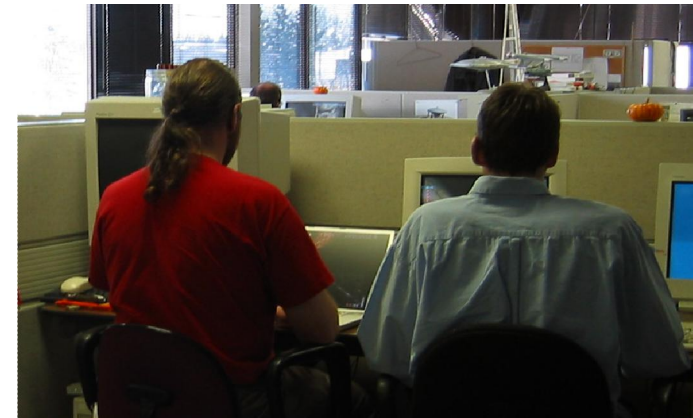
Quadrant 2



Quadrant 2 Tests

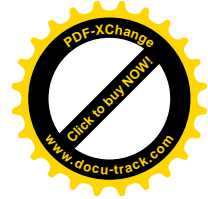
Business-facing tests that support the team

- Acceptance tests
- Examples
 - Used to elicit requirements
- User experience prototypes
 - wire frames, mock-ups
- Pair testing – “Show me”



Business-facing Tests that Support the Team





Drive Development

- Example (or Acceptance Test) Driven Dev
 - Use examples to demonstrate differences
 - Examples can become tests
 - Express as executable tests
 - Code until the tests pass

- Identify hidden assumptions

Business-facing Tests that Support the Team



Other Benefits for Q2 Testing

- Help customers achieve advance clarity
- Executable tests added to regression suite
- Focus is external quality
- Helps define when we're done a story
- Customer – developer – tester collaboration



Business-facing Tests that Support the Team



Given, When, Then - Behavioural

Given ... Preconditions

When ... Triggers, actions

Then ... Consequences, results

Simple ex. (from Gojko Adzic, Bridging the Communication Gap):

Given the customer has a balance of \$1.00 and deposits \$5.00 to his account

When an authorization code A-123 is received

Then the transaction is completed, and the new balance is \$6.00



Q2 Collaboration Toolkit

- Checklists
- Personas
- Mind maps
 - brainstorming
 - words, ideas, tasks
- Mock-ups / paper prototypes
 - User-centered design
- Flow diagrams
- Whiteboards (physical and virtual)
- Thin slice/steel thread

Business-facing Tests that Support the Team





Toolkit – Turning Examples into Tests

- Fit/FitNesse
 - Collaboration in software development
 - Takes place of regular UI
 - Take the place of UI
 - Use to set up data
 - Test different inputs faster
- xUnit frameworks
 - Works well for developers



More Tools to Turn Examples into Tests

- Behaviour-driven development tools
 - Another approach to TDD
 - Focus on examples, “should” – common language
 - easyB, jBehave
- GUI test tools
 - Test UI behaviour, system test, legacy systems
 - Some examples:
 - Selenium
 - Watir/Watij
 - WebTest



Fit Example – Failing Test

Set-up: Create a variety of users that start with the first initial, have the same name, similar email address, and unique ids.

create user with login id	frankb	first name	Frank	last name	Billian	email address	frankb@ngx.xx	phone number		roles	Quality Assurance
create user with login id	junebug	first name	Juno	last name	Williams	email address	junebug@ngx.xx	phone number		roles	Quality Assurance
create user with login id	fillpot	first name	Darien	last name	Fillpot	email address	fillpot_darien@ngx.xx	phone number		roles	Quality Assurance
create user with login id	jonero	first name	Jone	last name	Roberts	email address	Jone.Roberts@ngx.xx	phone number		roles	Operations
create user with login id	billboa	first name	Bill	last name	Bia	email address	bill_bia@ngx.xx	phone number		roles	Employee
create user with login id	june	first name	Juno	last name	Will	email address	junebug1@ngx.xx	phone number		roles	Employee

Query users - return all: no criteria set (sort by login id)

query users with login id		first name		last name	email address	phone number	roles
<i>first name</i>	<i>last name</i>	<i>login id</i>	<i>email address</i>	<i>phone number</i>	<i>roles</i>		
Idi	Administrator	administrator	idi_administrator@ngx.com	403-974 4957	Employee,Operations,Production Support,Quality Assurance		
Bill	Bia	billboa	bill_bia@ngx.xx		Employee		
Darien	Fillpot	fillpot	fillpot_darien@ngx.xx		Quality Assurance		
Frank	Billiann <i>expected</i>	frankb	frankb@ngx.xx		Quality Assurance		
	Billian <i>actual</i>						
Jone	Roberts	jonero	Jone.Roberts@ngx.xx		Operations		
Juno	Will	june	junebug1@ngx.xx		Employee		
Juno	Williams	junebug	junebug@ngx.xx		Quality Assurance		

Query users based on first name (sort by login id)

query users with login id		first name	J	last name	email address	phone number	roles
<i>first name</i>	<i>last name</i>	<i>login id</i>	<i>email address</i>	<i>phone number</i>	<i>roles</i>		
Jone	Roberts	jonero	Jone.Roberts@ngx.xx		Operations		
Juno	Will	june	junebug1@ngx.xx		Employee		
Juno	Williams	junebug	junebug@ngx.xx		Quality Assurance		

Query users based on last name (sort by login id)

query users with login id		first name		last name	Bi	email address	phone number	roles
<i>first name</i>	<i>last name</i>	<i>login id</i>	<i>email address</i>	<i>phone number</i>	<i>roles</i>			
Bill	Bia	billboa	bill_bia@ngx.xx		Employee			
Frank	Billian	frankb	frankb@ngx.xx		Quality Assurance			



Fit Example – Passing Test

Set-up: Create a variety of users that start with the first initial, have the same name, similar email address, and unique ids.

create user with login id	frankb	first name	Frank	last name	Billian	email address	frankb@ngx.xx	phone number	roles	Quality Assurance
create user with login id	junebug	first name	Juno	last name	Williams	email address	junebug@ngx.xx	phone number	roles	Quality Assurance
create user with login id	fillpot	first name	Darien	last name	Fillpot	email address	fillpot_darien@ngx.xx	phone number	roles	Quality Assurance
create user with login id	jonero	first name	Jone	last name	Roberts	email address	Jone.Roberts@ngx.xx	phone number	roles	Operations
create user with login id	billboa	first name	Bill	last name	Bia	email address	bill_bia@ngx.xx	phone number	roles	Employee
create user with login id	june	first name	Juno	last name	Will	email address	junebug1@ngx.xx	phone number	roles	Employee

Query users - return all: no criteria set (sort by login id)

query users with login id	first name		last name	email address	phone number	roles
first name	last name	login id	email address	phone number	roles	
Idi	Administrator	administrator	idi_administrator@ngx.com	403-974 4957	Employee,Operations,Production Support,Quality Assurance	
Bill	Bia	billboa	bill_bia@ngx.xx		Employee	
Darien	Fillpot	fillpot	fillpot_darien@ngx.xx		Quality Assurance	
Frank	Billian	frankb	frankb@ngx.xx		Quality Assurance	
Jone	Roberts	jonero	Jone.Roberts@ngx.xx		Operations	
Juno	Will	june	junebug1@ngx.xx		Employee	
Juno	Williams	junebug	junebug@ngx.xx		Quality Assurance	

Query users based on first name (sort by login id)

query users with login id	first name		last name	email address	phone number	roles
first name	last name	login id	email address	phone number	roles	
Jone	Roberts	jonero	Jone.Roberts@ngx.xx		Operations	
Juno	Will	june	junebug1@ngx.xx		Employee	
Juno	Williams	junebug	junebug@ngx.xx		Quality Assurance	

Query users based on last name (sort by login id)

query users with login id	first name		last name	email address	phone number	roles
first name	last name	login id	email address	phone number	roles	
Bill	Bia	billboa	bill_bia@ngx.xx		Employee	
Frank	Billian	frankb	frankb@ngx.xx		Quality Assurance	



Exercise Story

- As an online shopper, I want to be able to see the cost of different shipping options when I check out, so I can choose which one I want.

Assumptions:

- User has already entered shipping address.
- API for cost calculator available, takes postal code and weight.
- The options are Ground, 2 day and Overnight.
- PO Boxes are ground only.
- Items > 20 lbs are Ground only.
- User will be able to choose different options for different items.

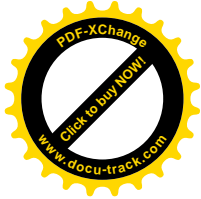




Exercise – Let's Question

1. What kind of shipping options are we talking about?
2. What costs do we have to consider?
3. Can we break this story into smaller chunks?
4. Are there multiple viewpoints? Who?
5. What would the first slice be?





Exercise - Let's draw

- **The first “slice” is:**
- 1. In your group, select a customer.
- 2. Draw the process flow(s)
- 3. Identify the steel thread or thin slice you can start developing

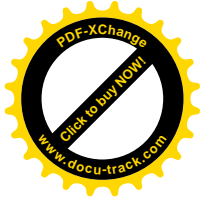




Exercise – Let's ATDD

1. Identify a high level acceptance test (s) for the first slice.
 2. For the 2nd? 3rd
-
- What did you learn?





Story Test Planning

Think of variations

Capture assumptions, thoughts....

Story test template



Exercise – Let's Automate

- Take your acceptance tests
- Create Fit type tests or Given, When, Then

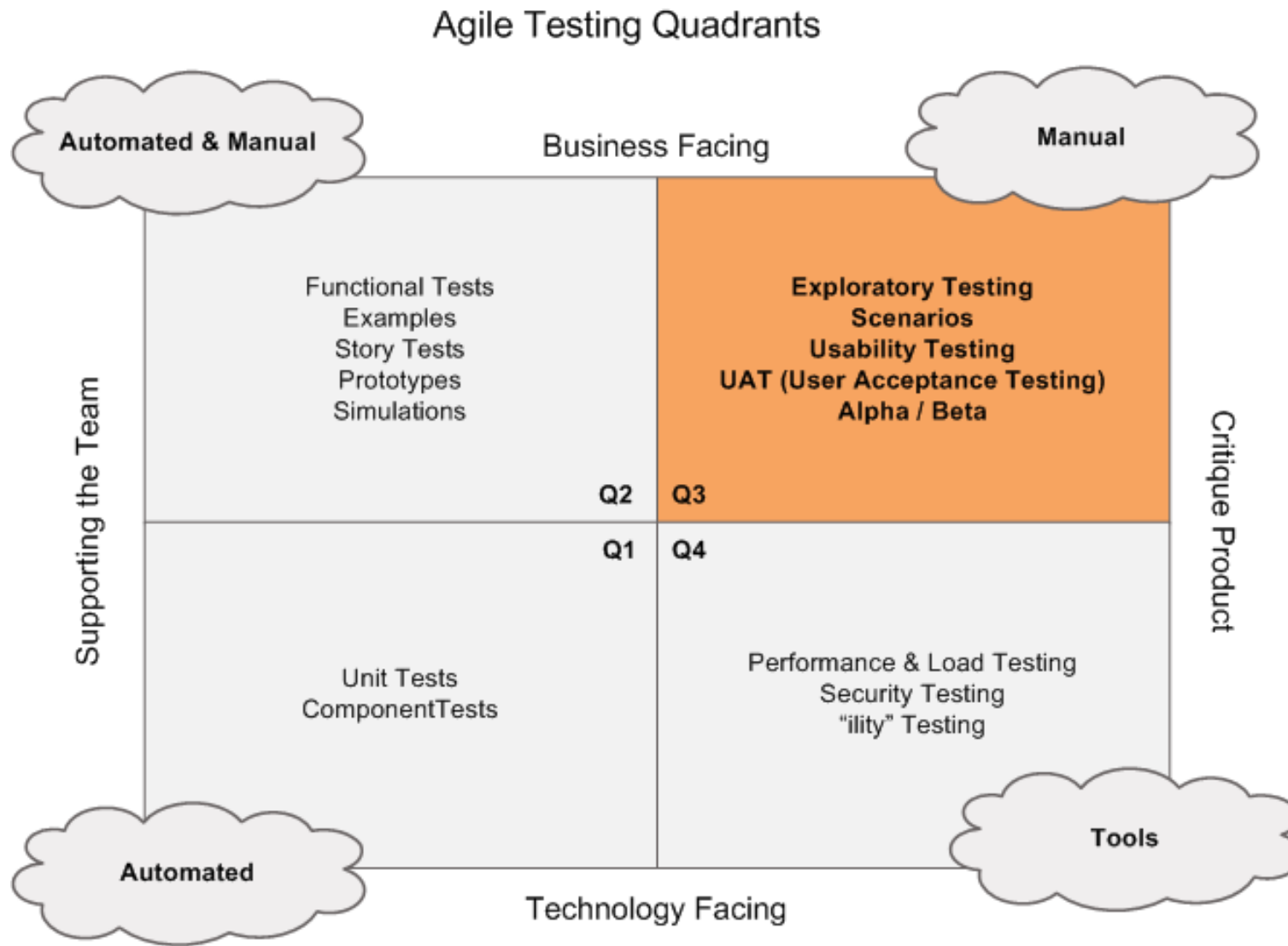
Example:

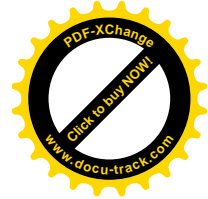
Option	Address	Postal Code	Weight	Cost
Ground	Seattle	123456	5 lbs	3.52

Business-facing Tests that Support the Team



Quadrant 3





Quadrant 3 Tests

Business-facing tests that critique the product

- Exploratory Testing
- Test for Usability
- Collaborative Testing
- Demos
- User Acceptance Testing

Business-facing Tests that Critique the Product



Exploratory Testing

- Learn & design
- Session-based (time-box)
- Provides feedback into new stories
- Test for usability
 - understand end users - personas
 - who will be using the system
- Follow “smells”, your instincts
- Touring (James Bach, James Whitaker)

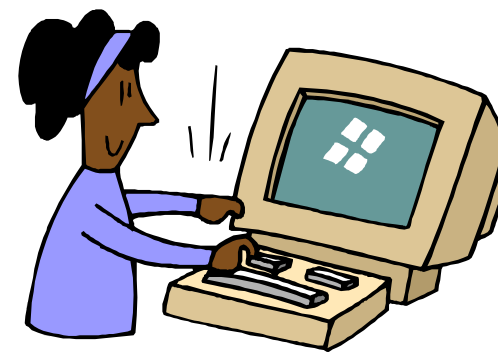


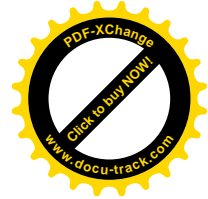
Business-facing Tests that Critique the Product



Other Types of Testing

- Scenario testing
 - Process flows
 - Realistic data
 - Soap opera testing (Hans Buwalda)
- Usability testing
 - Personas
 - Navigation
 - Observing users
- Don't forget documents, reports, help text





Collaborative Testing

- Provide feedback
 - Turn learnings into tests that drive new features
 - Change process as needed
- Iteration reviews
 - Builds confidence
 - Quick feedback loop
- Informal demos
 - Pair exploratory testing with customer
 - Even on unfinished code

Business-facing Tests that Critique the Product





Benefits for Q3 Testing

- Feedback to Quadrants One and Two
- Evaluation of the actual product
- Recreate actual user experiences
- Realistic use and data

Remember your context

- What works for your situation
- “It depends”; a tool, not a rule

Business-facing Tests that Critique the Product



Tools to Critique the GUI

- Record/playback
 - ▣ Traditional vendor tools
 - ▣ Some open source tools have recorders
- Open source libraries for web browsers
 - ▣ Watir is Ruby library for testing thru browser
- Specifying tests
 - ▣ Canoo WebTests specified in XML



Tools for Exploratory Testing

- Test scenario setup
 - eg. Watir/Watij scripts
- Generate test data
 - eg. PerlClip, Ruby script, Fit
- Simulators
 - Simulate data, feed to app over time
- Monitors
 - Watch log files
- Emulators
 - Duplicate system behaviour
 - eg. mobile devices

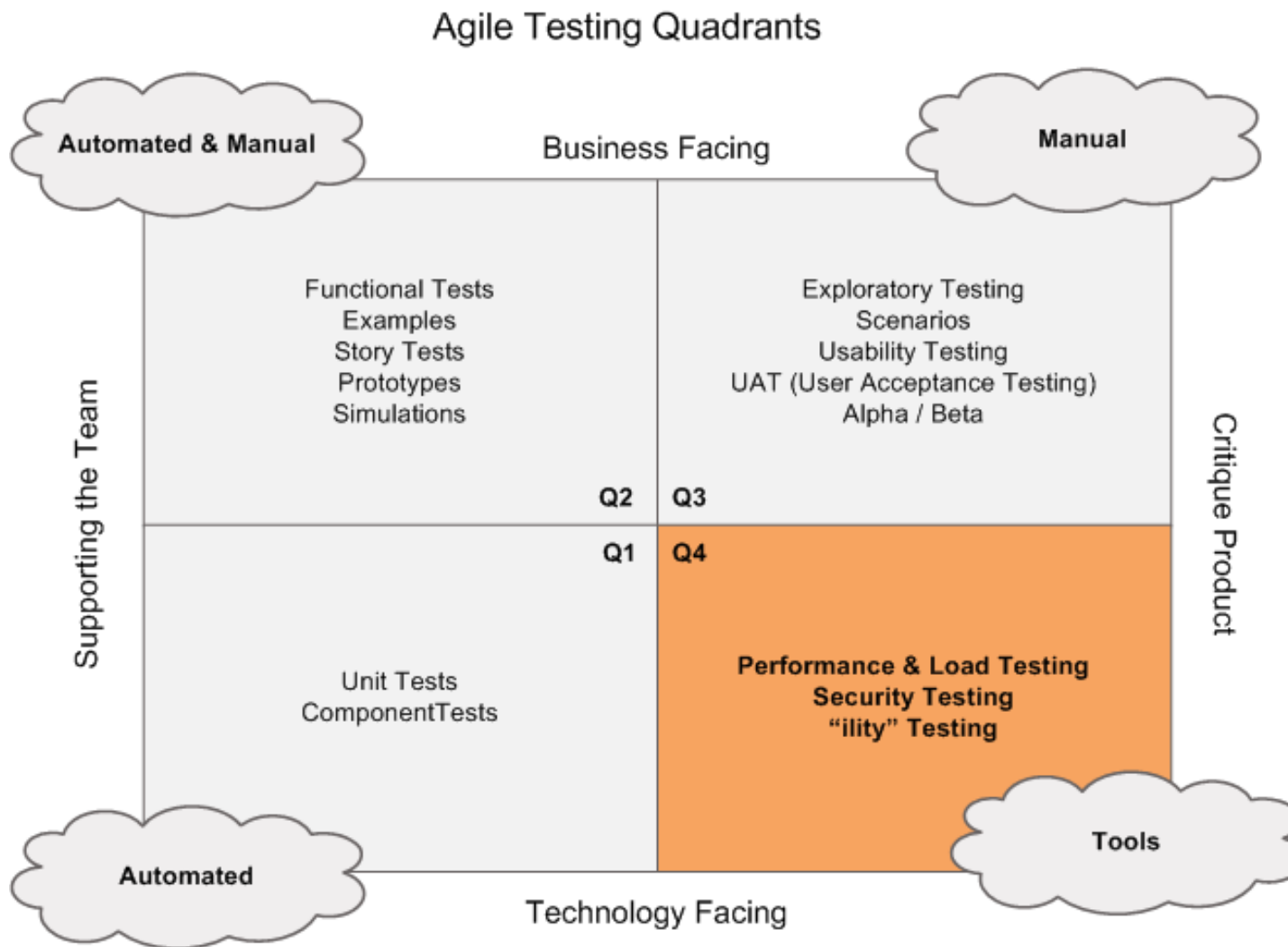


Exploratory Test Exercise

- *Story:* As an online shopper, I want to be able to delete items from my shopping cart.
- Think of some personas or characters and devise exploratory scenarios which that role might get into, for example:
 - Senior citizen who never shopped on the web
 - Hacker looking to cause trouble
 - Working mom in a rush
 - Internet-savvy teen



Quadrant 4



Quadrant 4 Tests

Business-facing tests that critique the product

- Non-functional tests
- “ility” testing
- Performance, scalability, stress, load
- Memory management
- Security testing
 - Roles & permissions , system ‘hacking’
- Data migration
- Infrastructure Testing
- Recovery



Technology-facing Tests that Critique the Product

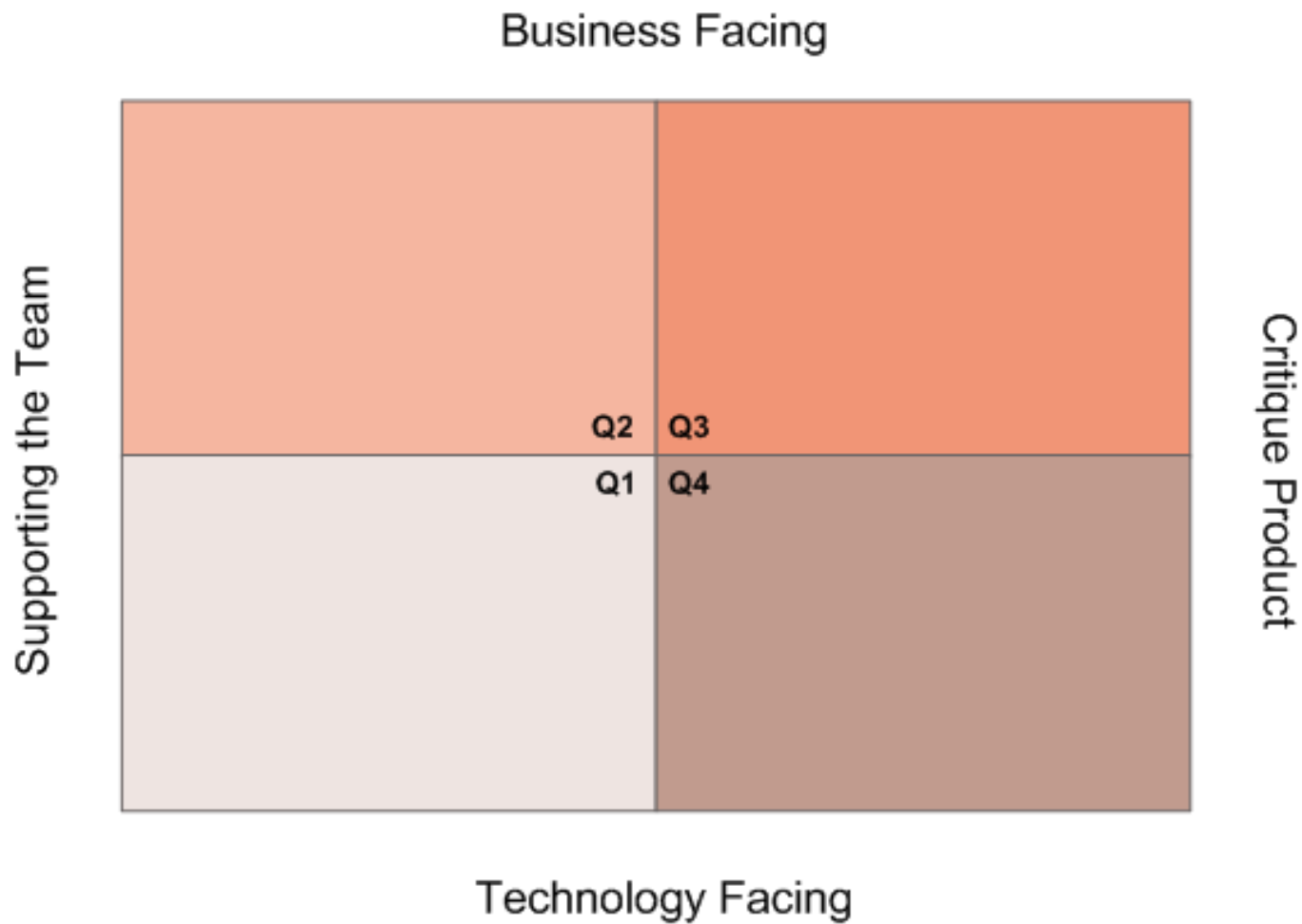


Benefits of Q4 Testing

- “Non-functional” requirements may be higher priority than “functional”
 - Performance
 - Stability
 - Security
 - ...other quality criteria
- Helps your application deliver the ‘right’ value
- Makes the ‘finished’ product



Any questions about the quadrants?



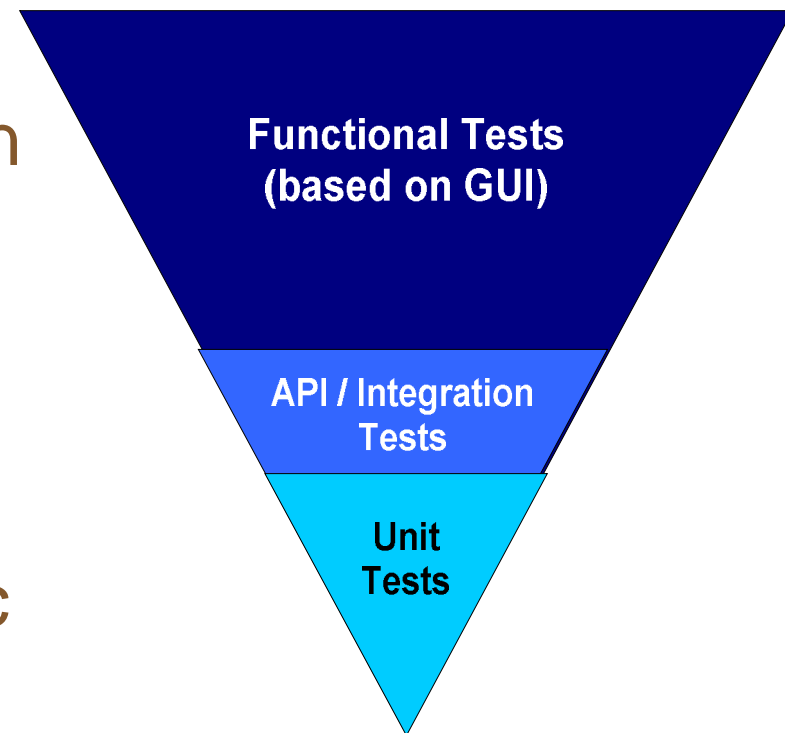
Why Automate?

- Free up time for most important work
- Repeatable
- Safety net
- Quick feedback
- Help drive coding
- Manual tests are error prone
- Tests provide documentation
 - You know what your system does



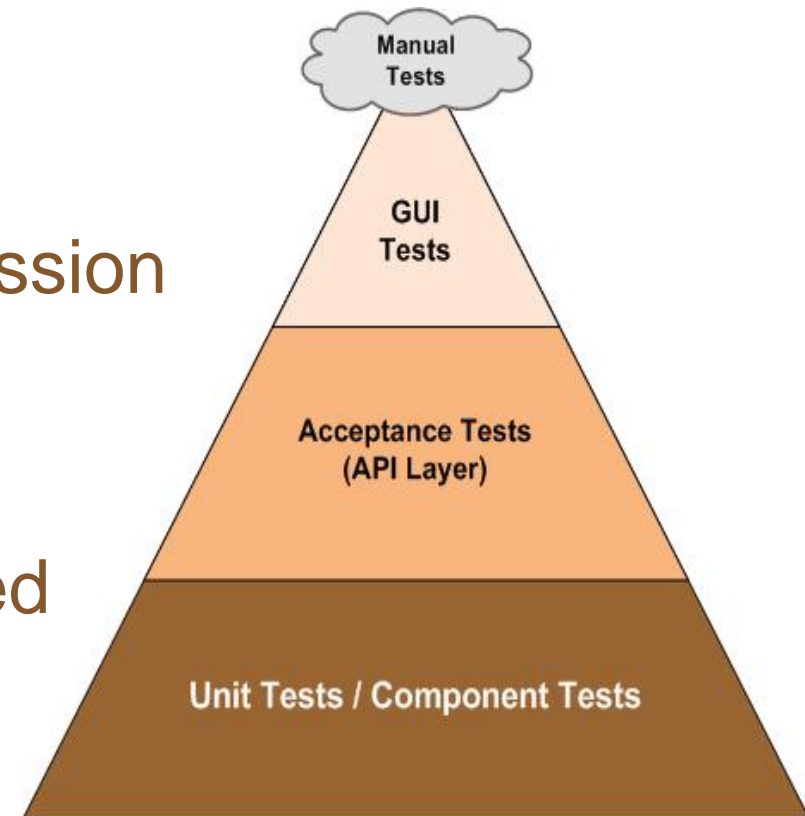
Test Pyramid – OLD model

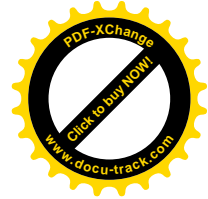
- Most testing was done by the QA team
 - manually or
 - Play / record automation
 - Through the GUI
- Based on the V-model
 - Unit tests were sporadic
- Developers knew little about testing



Automated Test Pyramid - Agile (Mike Cohn)

- Unit tests are the base layer
 - Provide the fastest feedback
 - Best ROI
- Middle layer
 - becomes functional regression tests
- GUI layer
 - May be partially automated
 - Mostly exploratory testing

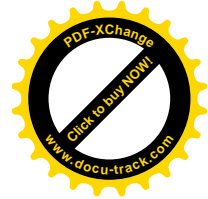




Test Coverage

- How are you going to know when you are done?
- What is enough?
- Collaborate to decide what coverage you want
- Think risk – include the customer
- Functional coverage
- Lines of code? ... be careful
- What tools can you use?





Test Plan

- Benefit is in the planning
- Project specific
- Focus on what is really needed
- Refer to a strategy or testing approach doc for static information
- Should it be static or dynamic?



Project Test Planning Tools

- Quality Management Strategy
 - Outlines the consistent test approach
- Agile Testing Quadrants
- Automated Test Pyramid
- Test Plan Alternatives
 - Test Plan (simplified)
 - Test Workbook
 - Test Matrix
- Test Coverage Checklist



Light-weight Test Plan

- Project specific
 - highlights project critical information
- Risks, high level scope, assumptions, constraints
- Benefit is in the planning
- Focus on what is really needed
- Refer to QM Strategy doc for static info
- Should it be static or dynamic?

[Sample test plan](#)



Test Matrix

- Mechanism to think about functionality and possible test conditions
- High level view for release planning
- Value is in the planning
- Value can also be in the visible progress report



Test Plan Matrix – one example

Project A	Data Integrity	Look & Feel	Calculations	Currency	Import	Create	Update	Delete	Filter	Sorting	Pagination	Excel export	Access (security)	Boundary conditions	Load / Performance		FIT	Wwatir	Unit Tests
Fees																			
Summary Report																			
Adjustments																			
Company Maintenance																			
User Maintenance																			
Message Log																			
Log in / passwords																			
Access by roles																			
End to End tests																			
Legacy System																			
Release Manager																			
Data Migration																			



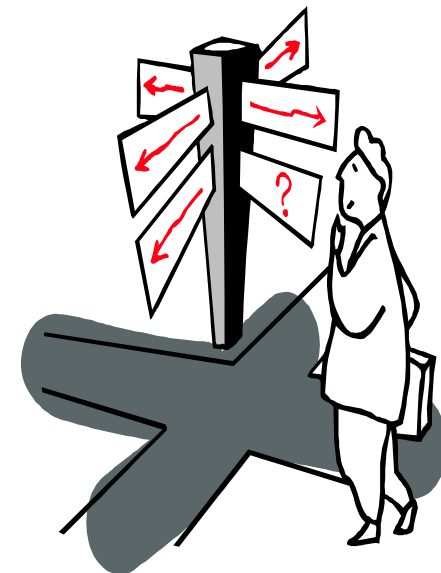
Testing Planning Tools

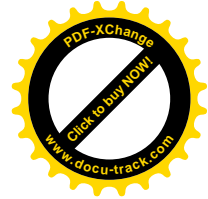
- Heuristics / Mnemonics
 - [Elisabeth Hendickson](#)
 - Jonathan Kohl / James Bach ([SFDPOT](#))
- Decision table
 - Logical relationships
- Mind mapping
- Use cases – flow mapping
 - Cause & effect
 - Focus on user
 - Happy path
 - Other paths
- State transition diagrams



Planning Your Test Strategy

- Consider scope, priorities, risks
- Tools that solve the problem
- Involve customers
- Collaborate with programmers
- Document only what is useful
- Consider all four quadrants
- Test matrix - big picture
 - allows whole team to understand

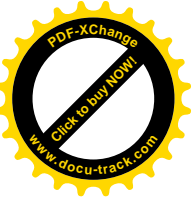




Group Discussion

- What quadrants does your team use 'back home'
- Which quadrant could your team "back home" improve?
- What will you do to address this, when you go back?
- How will you approach your test planning





Session Retrospective

- Use lessons learned to improve
- Review - are we doing all we could?
- What would you like to know more of?
- What was the most useful?



“The” Book

Agile Testing: A Practical Guide for Testers and Agile Teams

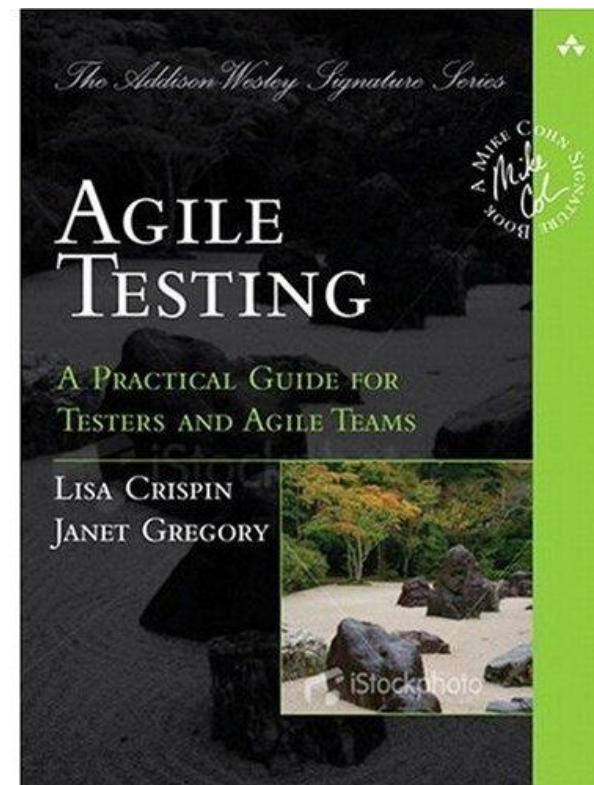
By Janet Gregory and Lisa Crispin

Available on

- Amazon.com
- Amazon.ca

www.agiletester.ca

www.janetgregory.ca





Resources

- *Collaboration Explained:* Jean Tabaka
- *Testing Extreme Programming,* By Lisa Crispin and Tip House
- *Fearless Change: Patterns for introducing new ideas,* Linda Rising and Mary Lynn Manns
- agile-testing@yahoogroups.com
- www.exampler.com (Brian Marick's site)
- www.testobsessed.com
- www.satisfice.com

