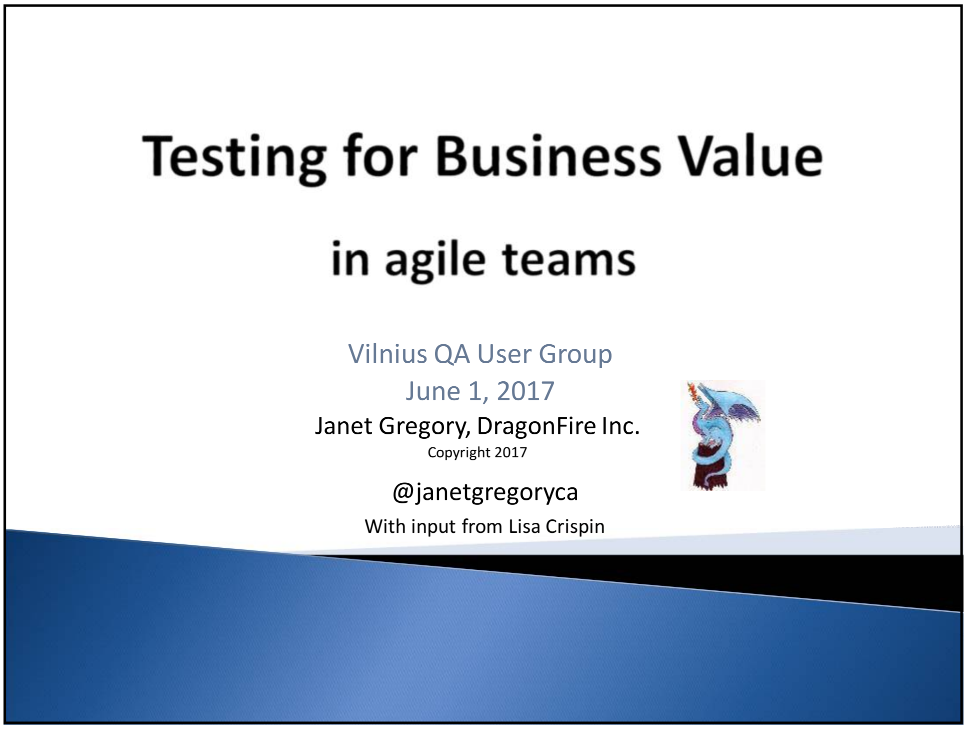


Testing for Business Value

in agile teams

Vilnius QA User Group
June 1, 2017

Janet Gregory, DragonFire Inc.

@janetgregoryca

With input from Lisa Crispin

A little about me

Agile Testing; 2009
More Agile Testing: Oct 2014
Website:
www.agiletester.com
www.agiletester.ca

DragonFire Inc.
www.janetgregory.ca
@janetgregoryca
janet@agiletester.ca

애자일 테스팅
Agile Testing
A Practical Guide for Testers and Agile Teams
Гибкое тестирование
実践アジャイルテスト
More Agile Testing
Learning Journeys for the Whole Team
Janet Gregory
Lisa Crispin
The Addison Wesley Signature Series

2

My Message Today

How testing helps create
shared understanding
of what you are building!
and
How to test for that business value.

We want to collaborate and develop a shared understanding of what we are building

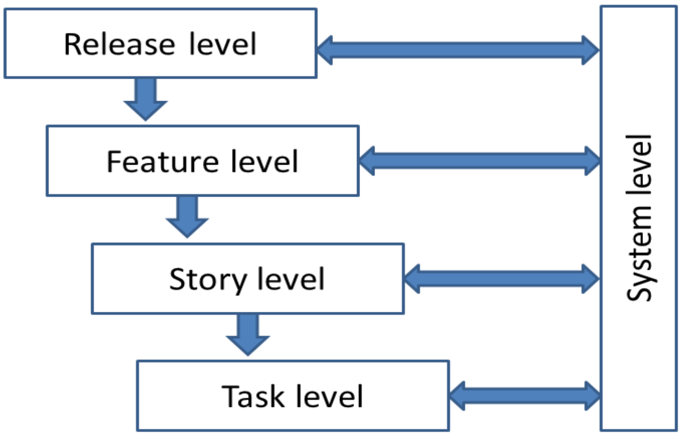
Small differences can cause big misperceptions

Can something be right?
But not meet a
customer’s needs?

For your value, which is better?

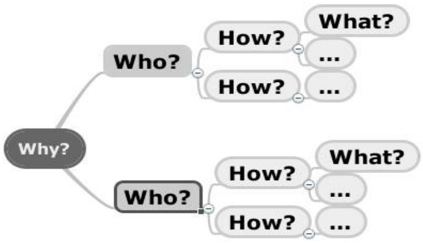
Which is right?

Levels of precision



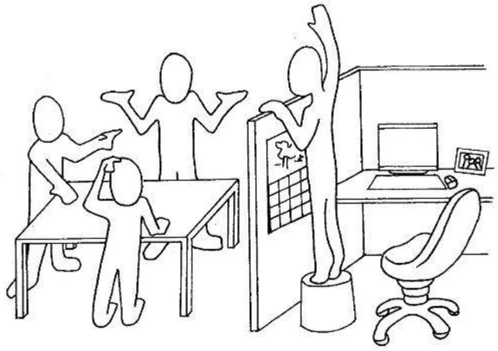
Impact Mapping – Gojko Adzic, www.impactmapping.org

- Why are we doing this? --- the goal
- Who can help or hinder? --- the stakeholders
- How can they help or hinder? --- the impacts!
- What can we do to help? --- the deliverables



Use structured conversations

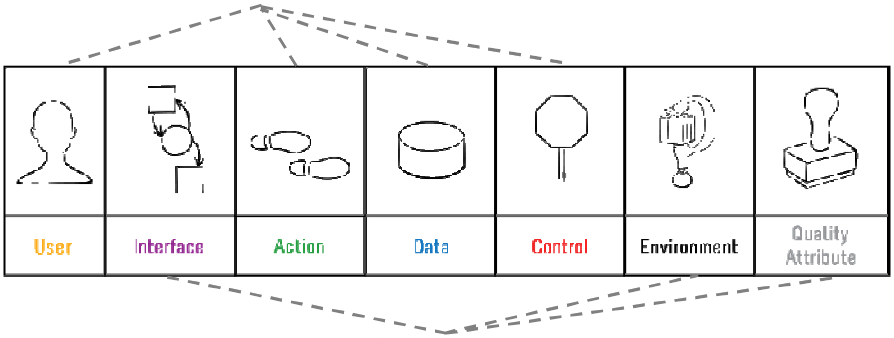
- Get the right people together
- Use tools to help focus the conversations



9

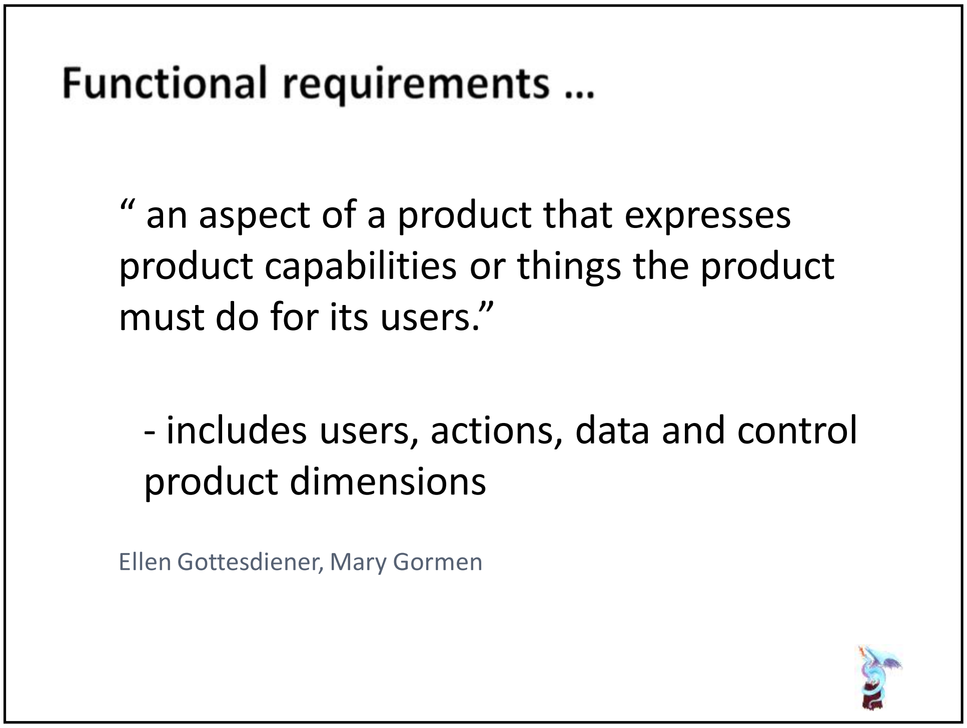
7 Product Dimensions

functional



nonfunctional

Source: *Discover To Deliver*, Gottesdiener & Gorman, 2012

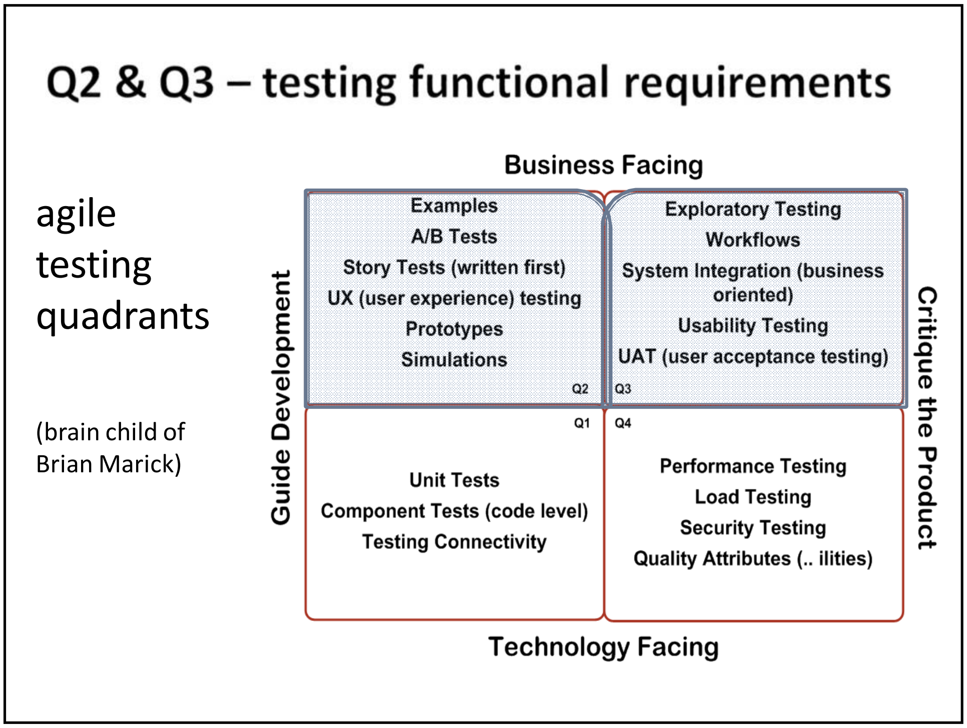


Functional requirements …

“ an aspect of a product that expresses product capabilities or things the product must do for its users.”

- includes users, actions, data and control product dimensions

Ellen Gottesdiener, Mary Gormen



Q2 & Q3 – testing functional requirements

agile testing quadrants

(brain child of Brian Marick)

Business Facing

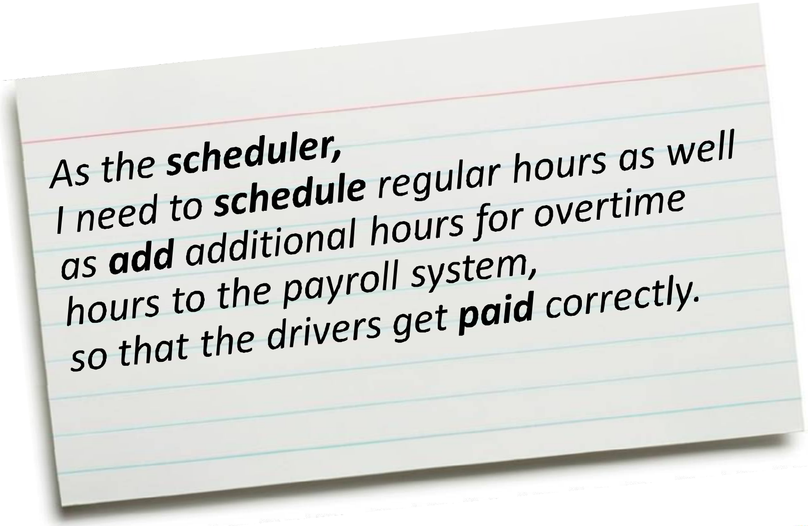
Guide Development

Critique the Product

Technology Facing

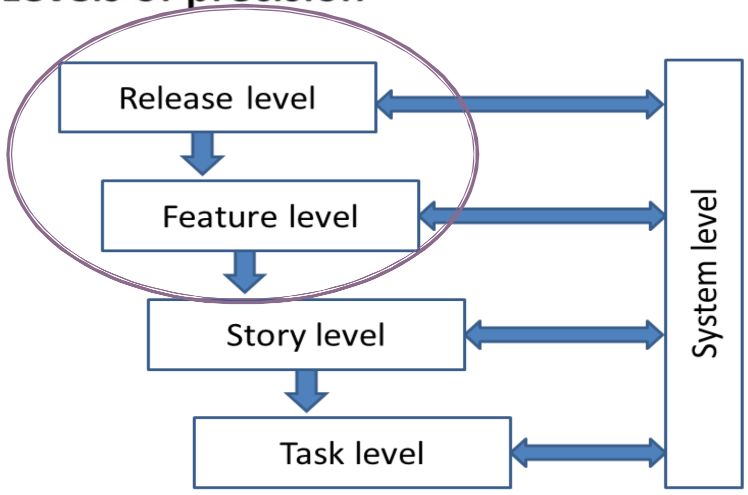
Q2	Q3
Examples A/B Tests Story Tests (written first) UX (user experience) testing Prototypes Simulations	Exploratory Testing Workflows System Integration (business oriented) Usability Testing UAT (user acceptance testing)
Q1	**Q4**
Unit Tests Component Tests (code level) Testing Connectivity	Performance Testing Load Testing Security Testing Quality Attributes (.. ilities)

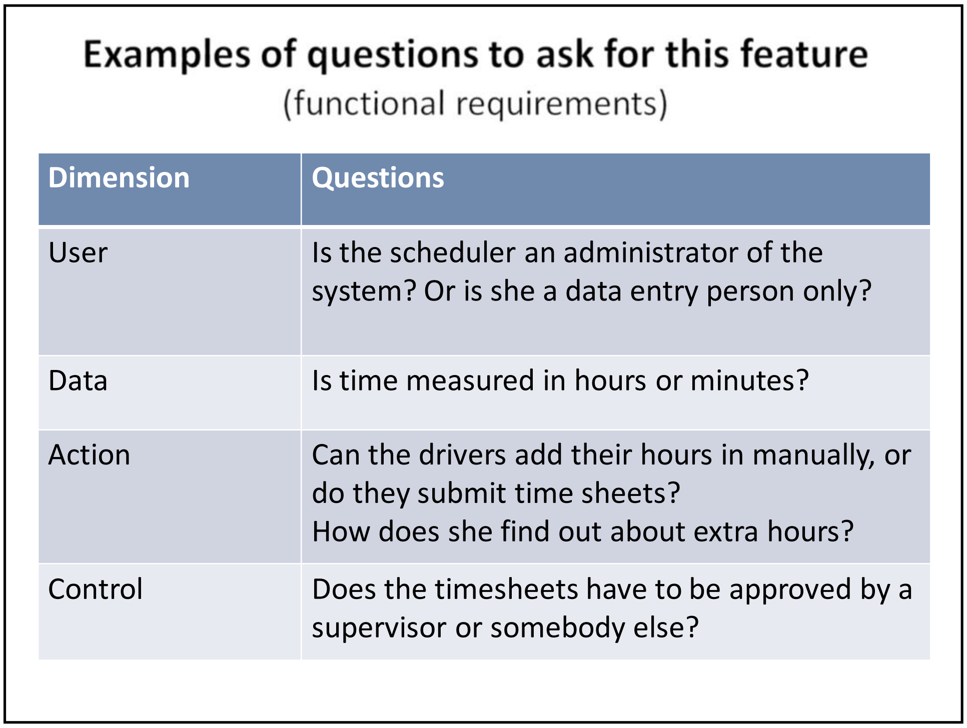
Let’s work through an example feature



*As the **scheduler,***
*I need to **schedule** regular hours as well*
*as **add** additional hours for overtime*
hours to the payroll system,
*so that the drivers get **paid** correctly.*

Levels of precision

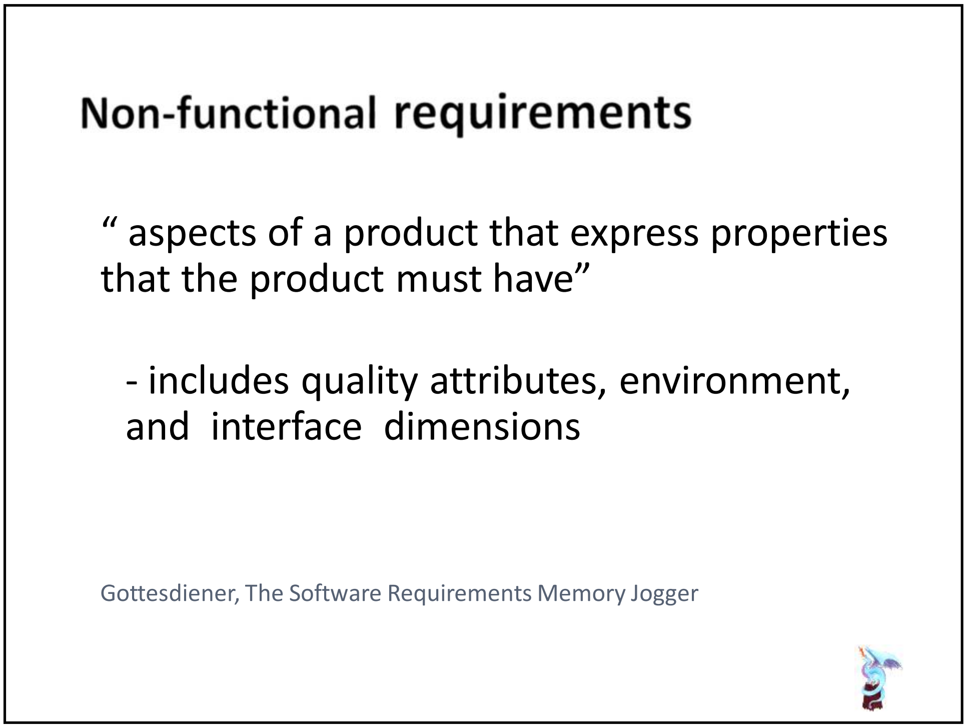




Examples of questions to ask for this feature

(functional requirements)

Dimension	Questions
User	Is the scheduler an administrator of the system? Or is she a data entry person only?
Data	Is time measured in hours or minutes?
Action	Can the drivers add their hours in manually, or do they submit time sheets? How does she find out about extra hours?
Control	Does the timesheets have to be approved by a supervisor or somebody else?



Non-functional requirements

“ aspects of a product that express properties that the product must have”

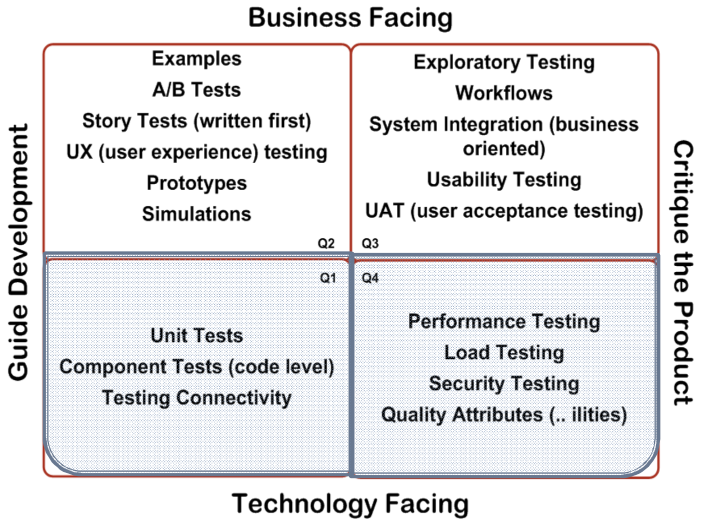
- includes quality attributes, environment, and interface dimensions

Gottesdiener, The Software Requirements Memory Jogger

Q1 & Q4 – testing quality attributes

agile testing quadrants

(brain child of Brian Marick)



Examples of quality attributes

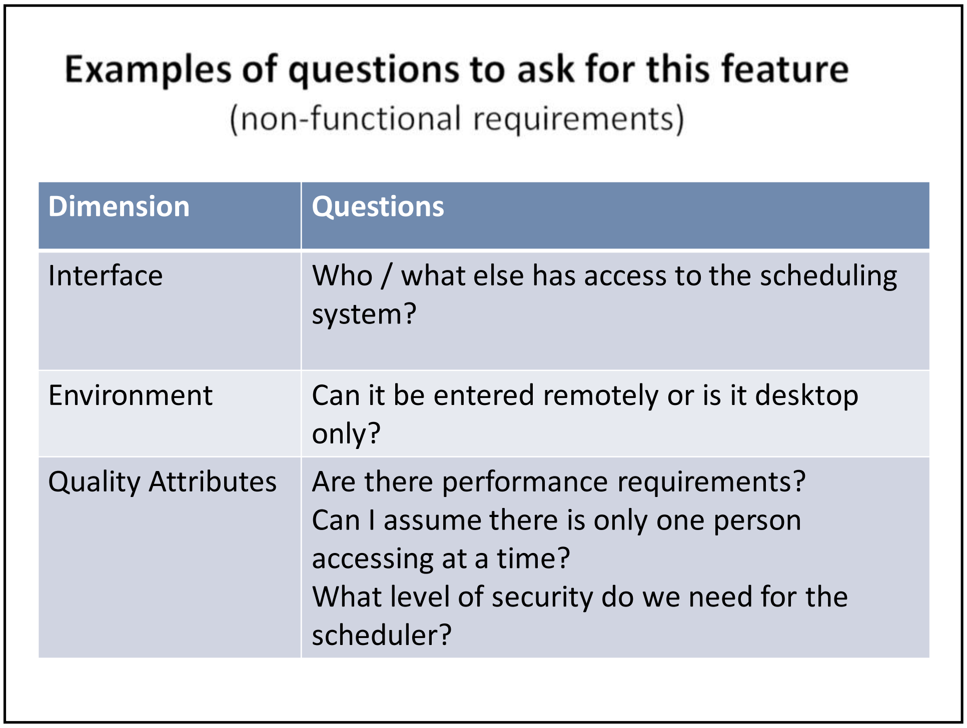
Operational

- Availability
- Recoverability
- Installability
- Robustness
- Interoperability
- Safety
- Performance
- Scalability
- Reliability
- Security
- Usability

Development

- Efficiency
- Flexibility
- Modifiability
- Portability
- Reusability
- Testibilty

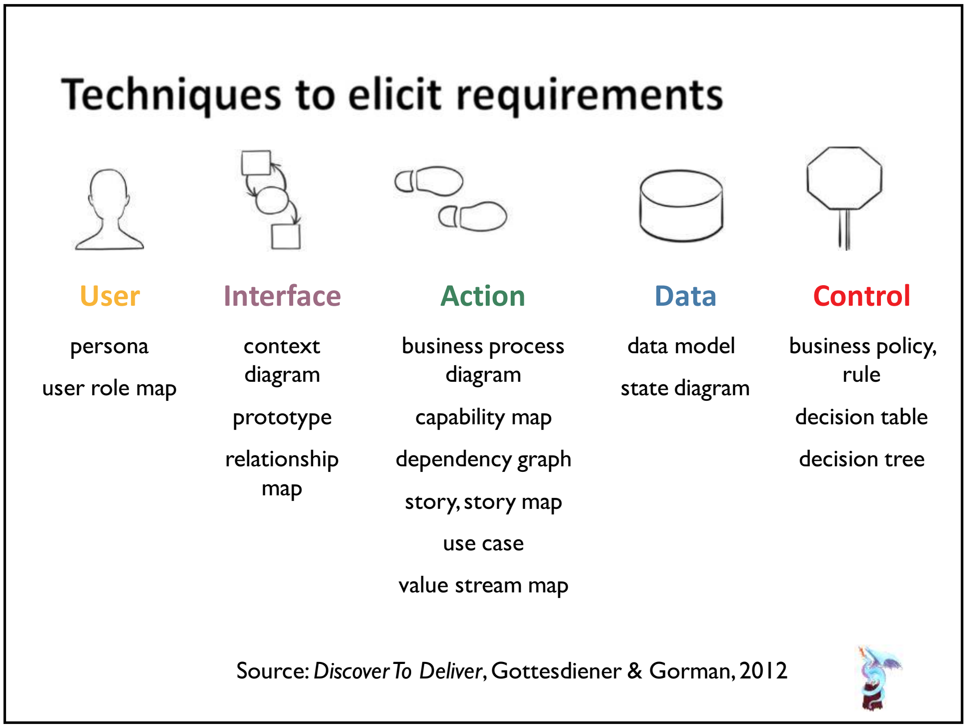
Adapted from: Gottesdiener, 2005



Examples of questions to ask for this feature

(non-functional requirements)

Dimension	Questions
Interface	Who / what else has access to the scheduling system?
Environment	Can it be entered remotely or is it desktop only?
Quality Attributes	Are there performance requirements? Can I assume there is only one person accessing at a time? What level of security do we need for the scheduler?



Techniques to elicit requirements

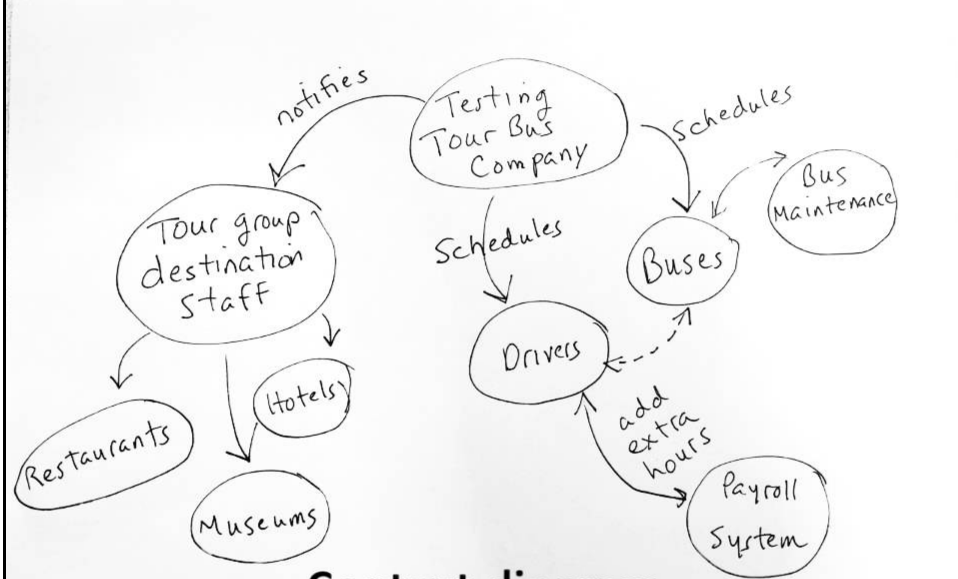
User	Interface	Action	Data	Control
persona user role map	context diagram prototype relationship map	business process diagram capability map dependency graph story, story map use case value stream map	data model state diagram	business policy, rule decision table decision tree

Source: *Discover To Deliver*, Gottesdiener & Gorman, 2012

Our example feature

*As the **scheduler,***
*I need to **schedule** regular hours as well*
*as **add** additional hours for overtime*
hours to the payroll system
*So that the drivers get **paid** correctly.*

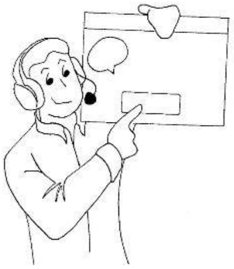
21



Context diagram (interfaces)

Persona is a typical end-user

Name: *Sam the Scheduler*



Description

Schedules buses
Schedules bus drivers
Matches bus drivers to buses
Arranges emergency buses and drivers
Ensures hours get logged
Liaises with the bus maintenance

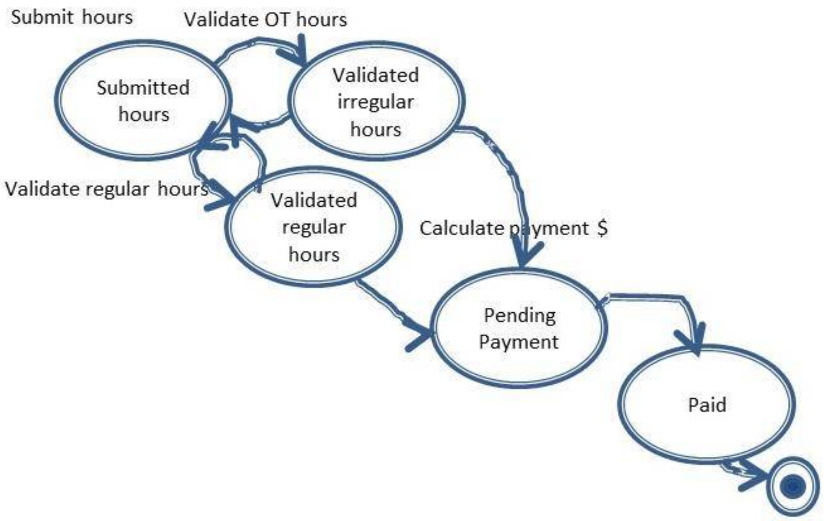
Values

Detailed oriented
Likes working with numbers
Likes trying new ideas

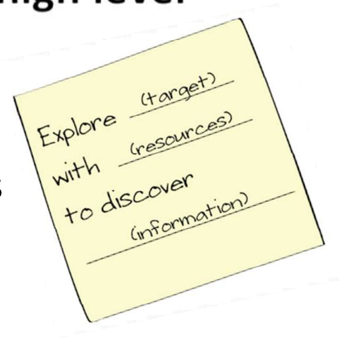
Likes

Likes the outdoors, camping
Introvert
Doesn't like conflict

State diagrams (lifecycle of data)



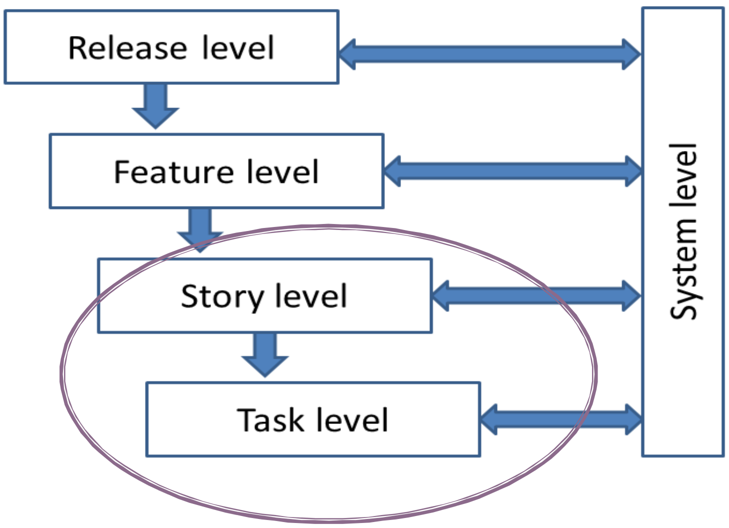
Scenarios – examples at a high level (action)



Scenarios for payment of hours

1. Regular hours only
2. Regular, plus overtime
3. Regular, plus addition call-out due to bus breakdown
4. Call-out only due to bus breakdown

Levels of precision


Release level

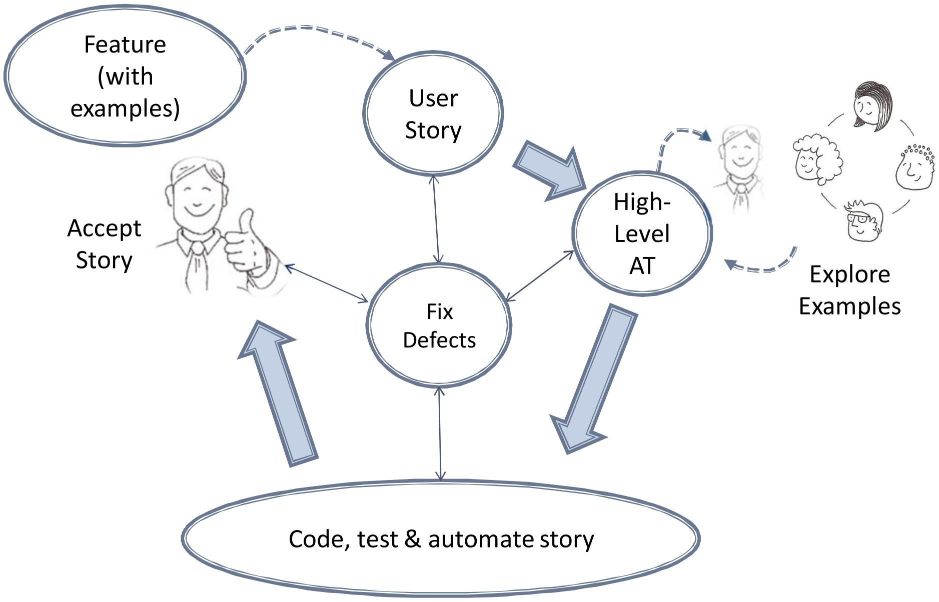
Feature level

Story level

Task level

System level

ATDD



Feature (with examples)

User Story

High-Level AT

Explore Examples

Fix Defects

Accept Story

Code, test & automate story

Examples .....

- Can become the actual tests
- Are a form of specification

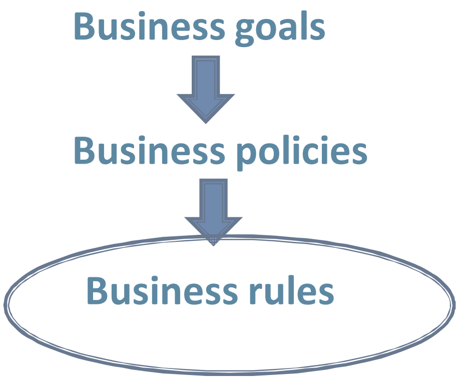
Credit and thanks to Brian Marick

Use examples .....

- To elicit requirements
- To reduce uncertainty
- To test people’s understanding of the requirement

Business rules

Define, constrain or enable behaviour of the software, business processes, data structure



Business goals

Business policies

Business rules

Prioritize business rules

- It may help prioritize stories

Bus tour example: Schedule a distributed team meeting in our scheduling app.

1. Meetings with > 2 attendees need a meeting room
2. Meeting with remote attendees must have video link
3. Meetings are scheduled during normal work hours for all attendees (employees located across Europe).

Acceptance tests

- Means to evaluate capability from a user's perspective
- Provide the scope of the story (or feature)
- Results are accepted or not accepted
  - If not – revisit!

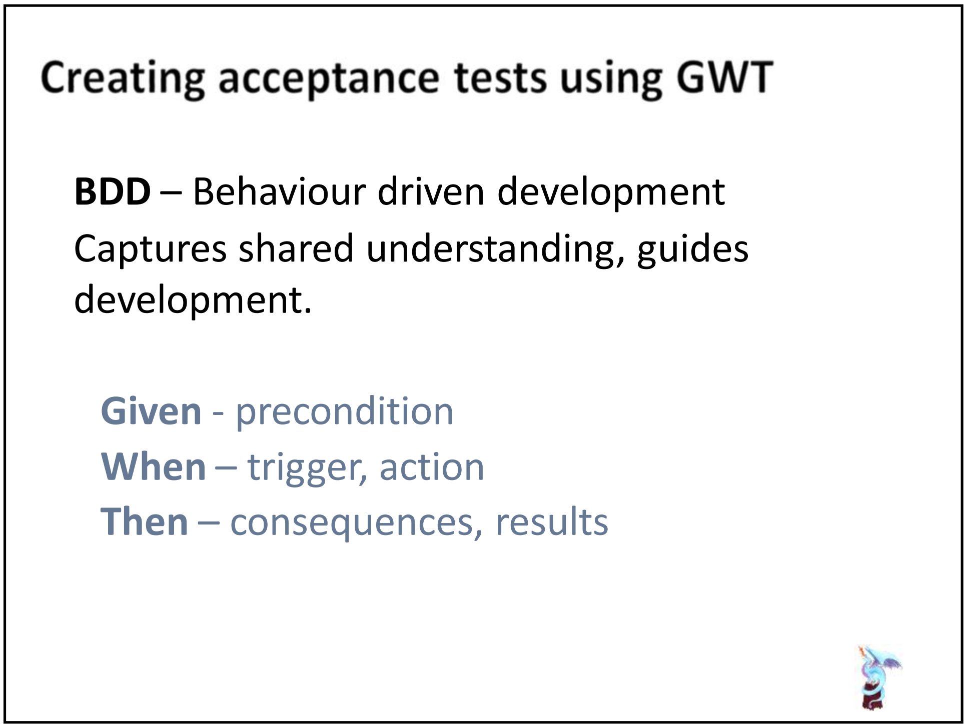
Example vs. Test

Scenario example:

A valid user "Jane" signs on with her valid email and password.

Test:

User	Email	Password	Expected Result	Comments
Jane	jane@agiletester.ca	Passw0rd22	Logged in	Valid login scenario



Creating acceptance tests using GWT

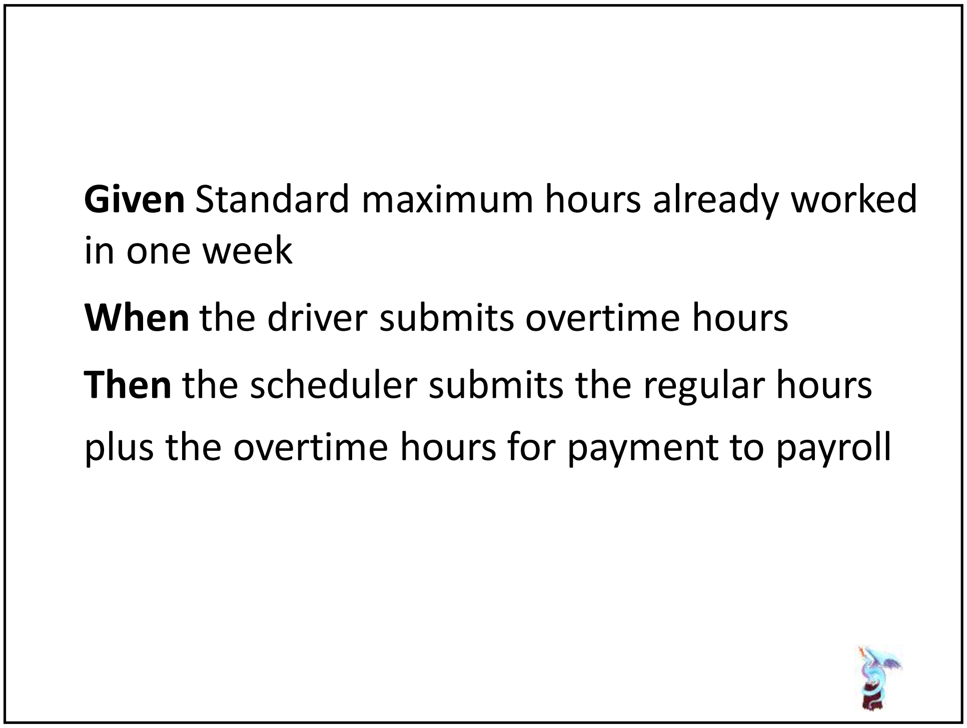
BDD – Behaviour driven development

Captures shared understanding, guides development.

Given - precondition

When – trigger, action

Then – consequences, results



Given Standard maximum hours already worked in one week

When the driver submits overtime hours

Then the scheduler submits the regular hours plus the overtime hours for payment to payroll

Concrete examples in tabular format

Driver	Regular Wage	Regular hours	Overtime hours	Callout time	Total paid
José	20.00	36.0	4.0	0.0	840.00
José	20.00	36.0	0.5	0.0	730.00
José	20.00	36.0	0.0	3.0	830.00
José	20.00	30.0	0.0	3.0	660.00

Quality attributes at the story level

Interfaces

- testing design – how does it interface with other apps?
- Different platforms, consistency, how does it react?

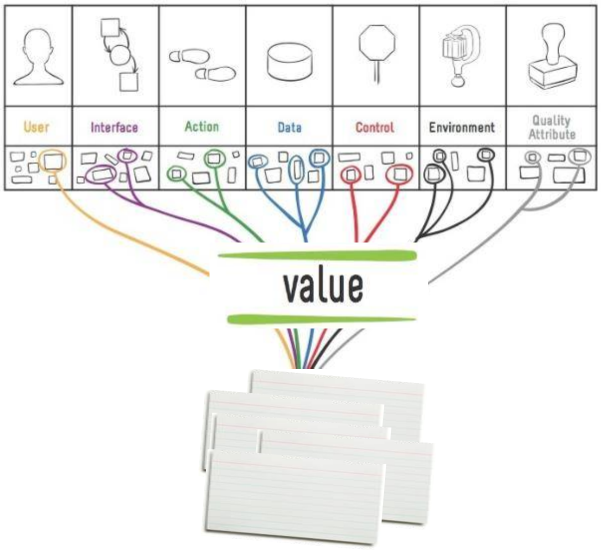
Quality attributes

- Constraints to be considered with every story

Implementation environments

- Requirements that limit build and deploy options

Assemble options



Source: *Discover To Deliver*, Gottesdiener & Gorman, 2012

We also need to explore what we built

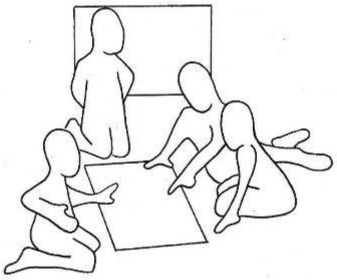
Personas

Tours

Risks

Workflows, Journeys

Think business value
Think testing
Test business value



- Who are the right people?
  - “Power of 3” (possibly 4 or 5)
- Look for the business rules
- Ask for concrete examples
- Focus on value to the business, customers, and end users

Agile Testing: A Practical Guide for Testers and Agile Teams

More Agile Testing: Learning Journeys for the Whole Team

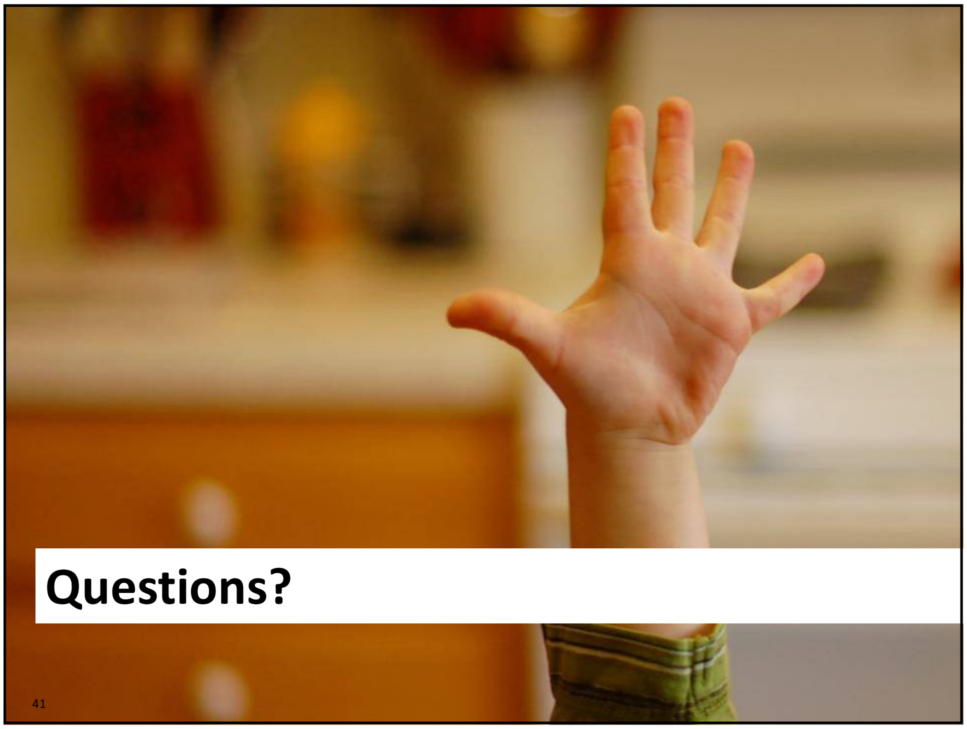
By Janet Gregory and Lisa Crispin

www.agiletester.ca
www.agiletester.com

Contact info

www.janetgregory.ca
Email: janet@agiletester.ca
Twitter: janetgregoryca

2 eBooks
The Addison-Wesley Signature Series
Agile Testing
A Practical Guide for Testers and Agile Teams
Lisa Crispin
Janet Gregory
A Mike Cohn Signature Book
More Agile Testing
Learning Journeys for the Whole Team
Janet Gregory
Lisa Crispin



Questions?

More Learning

- Adzic, Gojko, *Impact Mapping: Making a Big Impact with Software Products and Projects*, http://impactmapping.org
- Ellen Gottesdiener and Mary Gorman, Discover to Deliver: Agile Product Planning and Analysis, 2012
- Matt Wynne, Example Mapping
- Gärtner, Markus, *ATDD By Example: A Practical Guide to Acceptance Test-Driven Development*, Addison-Wesley, 2012a
- Wynne, Matt and Aslak Hellesoy, *The Cucumber Book: Behavior-Driven Development for Testers and Developers*, Pragmatic Programmers, 2012
- Karten, Naomi, "Are You Listening?", http://www.agileconnection.com/article/are-you-listening, Agile Connection, 2009