Think Like a User; Test Like a Machine!

Marianne Erickson, PhD, CSM
What is your role?

Scrum Master/Coach?
Product Owner?
Developer?
Tester?
Other?
Warm Up: Is this a Good User Experience?

What is a user?
Give it a try!

Now, on a sheet of paper, write a UA/UI/UX script that tests for a better user experience.

Don’t share it with anyone yet – we’ll come back to this later!

“There Grandma, I fixed it!”
What is UI testing and why do we care?

It’s testing the UI for a fluid user experience and intuitive process flow, from the perspective of the end user (not the development team).

Yeah, fine definition, but….  

 Doesn’t Unit Testing take care of all that?

• The unit test determines whether units of code are fit for use, but this is at best a tenuous relationship to usability.
• Programmers focus on code, not on the end user experience.
• Designers are not typical users (but they are continually growing more mindful of users as the UX knowledge base grows).
Then what about Regression Testing?

- A regression test is great for testing like a machine, but not so good at thinking like a user.
- Back button example
- Regression testing is not a part of UI/UA testing because it tests whether new code works with old code. If the end user performs a task that isn’t a part of the regression test, and that test fails in production, how could that failure have been avoided?
What is UI testing and why do we care?
It's testing the UI for a fluid user experience and intuitive process flow, from the perspective of the end user (not necessarily the product owner).

Well then, there’s always Functional Testing, right?

- Think about the question!
- Functional testing tests the function of the code (or slices of its function), not the interaction between the application and the user!
- Functional testing is important because it verifies that your system is fixed for release. However, it does not necessarily ensure that your application is user-friendly or user-centric.
What is UI testing and why do we care?

It’s testing the UI for a fluid user experience and an intuitive process flow, from the perspective of the end user (not necessarily the product owner).

But UAT is all about the user, so a good UA Testing phase should give us everything we need to know about the User Experience.

• UAT is really a Waterfall concept, and one that focuses more on a specific set of test cases than on the overall User Experience.

• If done correctly, UAT requires a set of end users as subjects to test the application and report back to the development team. UAT can be both expensive and time-consuming.

• UAT doesn’t account for response bias or participant bias, which we will discuss shortly!
How accurate are usability surveys?

Response bias and Participant bias

These are comprehensive terms that include a number of bias types, some or all of which can come into play when we ask users what they want in a UI:

- **Sponsorship bias** – Pro or con, based on who is doing the asking and how the user feels about them (influence bias). Can be conscious or unconscious.
- **Acquiescing** – Subject really wants to please the tester (especially if the subject is being paid).
- **Interviewer bias** – The interviewer influences the subject. (“Do you like the delicious red soda or the artificially-flavored, chemically-enhanced blue drink?”)
- **Questionnaire format issues** (1 to 5, for example)

### Website User Survey

1. The website has a user-friendly interface.
   - ![Likert scale example](image1)

2. The website is easy to navigate.
   - ![Likert scale example](image2)

3. The website’s pages generally have good images.
   - ![Likert scale example](image3)

4. The website allows users to upload pictures easily.
   - ![Likert scale example](image4)

5. The website has a pleasing color scheme.
   - ![Likert scale example](image5)

A survey using a Likert-style response set. This is one example of a type of questionnaire that can be highly vulnerable to the effects of response bias.
How accurate are usability surveys?

Response bias and Participant bias

Cognitive biases – A full list is available for anyone interested in learning more. A few examples include commonly known effects like the Framing effect, as well as lesser known but pervasive Planning Fallacy.

- The Framing Effect happens when different conclusions are drawn from the same information, depending on how that information is presented.
- The Planning Fallacy is the tendency to underestimate task-completion times.

Old Coke vs. “New” Coke” – A Marketing Disaster

A poll showed that only 13 percent of soda drinkers liked the new Coke. The pop was a bust of epic proportions. Pepsi took full advantage by launching a commercial featuring a girl who asked: "Somebody out there tell me why Coke did it? Why did Coke change?“ (CBS News, April 23, 1985)

Question: Why did Coca-Cola change their formula after nearly 100 years of success?
A strong argument for continuous testing: Who knows the product better than the team that builds it?

- The Agile team, especially an LLFT, knows the history of the product and what its past issues have been.
- The Agile team can test to the requirements, and this is sometimes the only way to discover flawed or incomplete requirements.

Testing all along, not restricted to a testing window, saves rework.

- Why not test all through the process? It’s fast, fun, and Agile!
- If everyone on the team is thinking like a user instead of assembly-line thinking, the end result is a user-friendly product that has been tested by real live humans.
- While there may be response biases, they are not the same biases as in UAT.

Isn’t this concept sort of like DevOps?

- Yes indeed! And continuous development requires continuous testing.
- Any member of the Dev Team can set aside a few minutes to an hour daily to run through a series of tests.

So let’s come back to “What is UI testing and why do we care?”

And if there are all of these response biases, can we even trust our end users?

This is precisely why we care!!
User Personas

“Personas aid designers to create different designs for different kinds of people and to design for a specific somebody, rather than a generic everybody.”  --Maria Kuhn

User Personas ...

• Can be updated as the user base changes and grows
• Can account for a number of user styles, demographics, and devices
• Are based on actual users and hybrids of actual users
• Provide perspectives similar or identical to those of real users
• Should incorporate user feedback
• Help the development team to avoid self-referential design and focus on the end user
How to create a persona? Multiple user personas?

- Know your customer/user base!
- Reflect your users as they are, not as you wish they were.
- Don’t be afraid to create multiple personas to portray many realistic users as you can without getting too granular.
- Make them real and relatable.
- Don’t be afraid to think about your competitors and what customers like about their product or application.
Examples

Music App

Who is your primary user base?
Secondary user base?
What is their primary device?

Persona 1: Jacques
• 34 year-old hipster
• Likes a variety of music genres
• Very tech-savvy
• Uses multiple devices (PC, Tablet, Phone)
• Likes to segment music types into different playlists

Persona 2: Fiona
• 90 years old
• Uses her iPod daily
• Likes Classical and Big Band music
• Needs a simple application that she can remember after her grandson teaches her to use it
• Likes to see icons that tell her what the system is doing
Utility Company Website

User base is both tech-savvy and Luddite. Everyone in a large metropolitan area is a potential user.

Persona 1: Simone
- 25 year-old environmentalist
- Statistician (likes to view usage trends)
- Very tech-savvy
- Uses multiple devices (PC, Tablet, Phone)
- Despises pop-ups

Persona 2: Richard
- 85 years old
- Very proud that he knows how to pay his bill online
- Online Bill Pay is the only feature he uses
- Thinking about enrolling in paperless billing
- A little nervous about clicking the wrong thing on the UI
- Likes to see a phone number “just in case”
The Kent smoker ...
An elegant, sophisticated, well-educated male, or the man who strives to reach the status of a typical Kent smoker.
(©Lorillard Research, 1960.)

The Kent Persona: Dr. David Brown

- 36 years old
- Brand-loyal
- PhD in Chemistry
- Affluent and worldly, with impeccable taste
- Charming, sophisticated, well-dressed
- Confident, analytical, intelligent, and sensible.
Did the creators of these products think about end users?

Disney Garden: Hanna Montana “Red Cherries” (2009)

Colgate Kitchen Entrees (1982)

Sapporo Diet Water actually is a real product, still available for purchase. And yes, it sells well.
Exercise
Some motivation required.
Stand up and stretch!
Then find someone you don’t know and create a realistic user persona, using your new partner as the primary user of your new application or product!

Keep in mind....

• Your new user persona is not your only customer, but s/he is an important customer archetype.
• Keep it real. Ask your user questions about what s/he likes and dislikes about your application (or product).
Stating the Obvious?
Application usability is enhanced when users know how to operate the UI and it guides them through the workflow.

(Don Norman, author of... The Design of Everyday Things)

Behavior Driven Development?
You suspected I would get here eventually, didn't you?
• You must test your UI, preferably iteratively, as you refine features.
• If you don’t like testing your application, it’s guaranteed your users won’t like testing it either.

User Feedback, if you are fortunate...
Don’t take it personally!
• You asked. If they give you honest feedback, it’s a win-win.
• The user or customer base is your product or application’s reason for being, so consider them (and their input) seriously!

How do we get there and stay Agile?
Think like a user! Employ your own variation of the Golden Rule: If I were an end user, what would I want the dev team to test?
• Keep it simple: One test at a time!
• Never forget that the UI is for the user, not for developers, business lines, systems, or nameless entities. Your personas must reflect your real users.
Luckily, you already have a tool to help your team get there without risking your Agility!

**Gherkin**
- [gur-kin]
- noun --
  1. the small, immature fruit of a variety of cucumber, used in pickling.
  2. the small, spiny fruit of a tropical vine, Cucumis anguria, of the gourd family, used in pickling. Also called **bur gherkin, gooseberry gourd, West Indian gherkin**.
  3. the plant yielding this fruit.
  4. a small pickle, especially one made from this fruit.
  5. in software development, a business-readable, domain-specific language that serves as documentation, automated test, and development aid.

Gherkin the language is:
- Logical
- Reusable
- Plain English
- Ready to use for UAT
- Easy to convert to Cucumber
- Easy to convert to Selenium
Fun Facts about Gherkin

- Aslak Hellesøy began the Cucumber Project in 2008. At the time, he had three major goals:
  - Help stakeholders express what they want via executable specifications
  - Help programmers write better software by making TDD/BDD easy
  - Reduce misinterpretations through a ubiquitous language
- Gherkin as a plain-text DSL came along soon after, primarily out of the Agile community and users on the Cucumber mailing list, so it is a true collaborative effort.
- Gherkin is written in a business-readable, domain-specific language that serves as documentation, automated test, and development aid, all rolled into one format.
- A domain specific language can exploit the syntax of the domain and allow validation.
- A simple “Given  →  When  →  Then” format
  - Given I am [Who]
  - When I use [What]
  - Then this happens. [Expected Behavior]
- Example:
  - Given I am an xyz.com user
  - When I discover an error on the site
  - Then I can click a Contact icon to launch a feedback page and report the error.
# How do we use it?

## The Right Way

You can test a user interface with simple scripts that remain user-focused.

## The Wrong Way

You can try to test everything all at once just to fulfill the testing requirement, or you can make the users do your testing in production.

### When you write a script, remember:

- We are testing as users, not as members of the development team.
- Keep test scripts in the present tense to simulate the User Experience in real time.
- Test one function at a time. A script that tests multiple functions is difficult to automate and complicates the isolation of failures.

### Don’t succumb to Test Creep—control it!

- Keep tests user-centered
- One single outcome per script
- Don’t go crazy thinking about all of the things that could happen. *UI testing is for confirming a fluid, intuitive user experience.*
- Writing Gherkin is not the same as writing good Gherkin!
Real-life (Bad) Examples

(Names have been changed to protect the culprits.)

Customer Payment UI

**Given** I am an authenticated user
**When** I want to pay my bill online
And I click the Pay Now button on the Home page
And a Pay Now page displays
And I must choose whether to pay by electronic check or by credit or debit card
And I click Pay Now in the Credit or Debit column
And a payment information page displays
And the page font color is black
And the font size is 12 point
And the font type is Times New Roman
**Then** this test passes.
Real-life (Bad) Examples

(Names have been changed to protect the culprits.)

Login Failure & Lockout

**Given** I am a user with a UserID and password

**When** I enter my UserID in the UserID field
And I enter an incorrect password in the Password field
And I click the *Log In* button
And an error message displays
And I enter an incorrect password in the Password field a second time
And I click the *Log In* button
And an error message displays
And I enter an incorrect password in the Password field a third time
And I click the *Log In* button
And an error message displays
And I enter an incorrect password in the Password field a fourth time
And I click the *Log In* button
And an error message displays
And I enter an incorrect password in the Password field a fifth time
And I click the *Log In* button
And an error message displays
And I enter an incorrect password in the Password field a sixth time
And I click the *Log In* button
And an error message displays
And I enter an incorrect password in the Password field a seventh time
And I click the *Log In* button
And an error message displays
And I enter an incorrect password in the Password field an eighth time
And I click the *Log In* button
And an error message displays ...
And how long do you think this script goes on?
How to Write a Negative Test

Scenario: 1
Given I am a Transport Tracking application user
When I visit the application in the test environment
Then the URL is https://programs-stage/tracking

Scenario: 2
Given I am a Transport Tracking application user
When I visit the application in the production environment
Then the URL is https://programs/tracking

Scenario: 3 (not valid for production testing)
Given I am a Grain Truck Tracking application user
When I visit the application in the test environment
And the URL is https://programs-stage/oldname
Then a failure is expected

How to keep scripts simple and limited

Build one test on the previous one and test in series.

Scenario: 1 (Happy Path)
Given I am a Transport Tracking application user
When I select a facility from the filter option
Then all vehicles associated to that facility display

Scenario: 2 (Alternate)
Given I am a Transport Tracking application user
When I deselect a facility from the filter option
Then all vehicles associated to that facility are hidden

Scenario: 3 (Alternate)
Given I am a Transport Tracking application user
When I need to see a list of vehicles that are not associated to any facility
Then a filter is available to display these vehicles
EXERCISE
Some motivation required.
Now give it another try!

Remember that test script you wrote at the beginning of this session? Now try writing it again!

Which is better – the first or second one?
And another try!

On the sheet of “tips” on your tables, there are multiple examples of bad Gherkin scripts. Now that you’ve had some practice, rewrite those terrible scripts so that they would add value to the UX/UI testing process!

This is Where Agile People Connect
to learn, engage, exchange, deepen,
refine, experience, and share

279 sessions (from beginner to advanced)
+ social events, lightning talks, Lean Coffee sessions,
impromptu hallway discussions, and more!

Given I attend Agile2018
And I participate in Marianne’s
workshop
When I return to my day job
I am confident writing tests that
benefit my end user experience.
Thank you for coming!

Questions?

Contact Information:

Marianne Erickson, PhD, CSM, SAFe® Agilist
marianne.erickson@daugherty.com
www.daugherty.com
Sources


And one really fun YouTube video about Norman Doors: https://www.youtube.com/watch?v=yY96hTb8Wgl&feature=youtu.be

Want more information about Gherkin? Visit https://github.com/cucumber/cucumber/wiki/Gherkin
Thank you!
Questions?
Comments?