TESTING IN AGILE

Dan Ashby
@DanAshby04

www.DanAshby.co.uk
www.SoftwareTestingClinic.com
www.TestingInThePub.co.uk
Q: what is software testing?
The most common answers to this question are…

“Checking that the software meets the requirements”

or

“Make sure the software works as expected”
This kind of thinking stems a BIG PROBLEM

“Testing is DEAD”...?

Testing is DEAD!
Automate ALL THE THINGS!

Really??
Come on......!!!
To understand **TESTING** and where it fits within Agile, we first need to think about **INFORMATION**
Let me tell you a story about the RED BUTTON
Meet my wife Amanda
How do you **TEST A BUTTON** without being allowed to **PRESS IT...?**
I pressed the button...
New **info** after the 1\textsuperscript{st} attempt of pressing the button:

The button didn't appear to do anything

- No light went on or off...
- No noise...
- No heat or aircon changes...
- Nothing happened to the TV...
- Nothing happened to the mains sockets next to it...

New **info** after the 2\textsuperscript{nd} attempt of pressing the button:

**THE BUTTON IS STUCK INTO THE WALL** - I can't un-press it...

*Amanda is going to kill me... 😞*
uh-oh...
Is THIS good Enough?
Think about **Information** in this **Story**
Let's Think about
SOFTWARE
Information = DNA
All software starts with a small piece of info...
we put processes and activities in place to investigate and evaluate that info...
That investigation allows us to form more artefacts with more info...
Those artefacts containing more info can then be used to stem other activities...
Those activities create the project outputs & allow us to measure success and feed back into new ideas.
However... "Information" has many **QUIRKS**
INFORMATION
KNOWN
EXPLICIT
INFORMATION

TACIT &
IMPLICIT
INFORMATION

UNKNOWN
(LACK OF)
INFORMATION

LACK OF
AWARENESS
OF UNKNOWNS
The 5 ORDERS of IGNORANCE
KNOWLEDGE

Known information
Expectations

Unknown information that we are aware of

Unknown information that we are unaware of

No means of becoming aware of unknowns

Unawareness of the orders of ignorance…
KNOWLEDGE

Known information
Expectations

LACK OF KNOWLEDGE

Unknown information that we are aware of

Unawareness of the 5 orders of ignorance…
KNOWLEDGE

Known information
Expectations

LACK OF KNOWLEDGE

Unknown information that we are aware of

LACK OF AWARENESS

Unknown information that we are unaware of
LACK OF PROCESSES
No means of becoming aware of unknowns

LACK OF AWARENESS
Unknown information that we are unaware of

LACK OF KNOWLEDGE
Unknown information that we are aware of

KNOWLEDGE
Known information
Expectations

Unawareness of the 5 orders of ignorance…
META IGNORANCE
Unawareness of the 5 orders of ignorance...

KNOWLEDGE
Known information
Expectations

LACK OF KNOWLEDGE
Unknown information that we are unaware of

LACK OF AWARENESS
Unknown information that we are aware of

LACK OF PROCESSES
No means of becoming aware of unknowns

META IGNORANCE
Unawareness of the 5 orders of ignorance...
So... what's the **Relationship** between **TESTING** and **INFORMATION**?
Assertive Testing (Scripted) (Test Cases)
enables

Assertive Testing (Scripted) (Test Cases)
INFORMATION

KNOWN EXPLICIT

TACIT & IMPLICIT

UNKNOWN (THAT WE’RE AWARE OF)

UNAWARENESS (OF UNKNOWNs)

confirms

enables

Assertive Testing
(Scripted)
(Test Cases)
INFORMATION

KNOWN EXPLICIT

TACIT & IMPPLICIT

UNKNOWN (THAT WE’RE AWARE OF)

UNAWARENESS (OF UNKNOWN)

confirms

enables

Assertive Testing

(Scripted)

(Test Cases)

Person -> Automation
Automation enables confirms

Person ----------------> Automation

Assertive Testing
(Scripted)
(Test Cases)

Investigative Testing
(Exploratory)
(Test Charters)

INFORMATION

KNOWN EXPLICIT

TACIT & IMPLICIT

UNKNOWNNS
(THAT WE’RE AWARE OF)

UNAWARENESS
(OF UNKNOWNNS)
Automation enables person confirms

Investigative Testing (Exploratory) (Test Charters) uncovers

INFORMATION

KNOWN EXPLICIT

TACIT & IMPLICIT

UNKNOWN (THAT WE’RE AWARE OF)

UNAWARENESS (OF UNKNOWN)

assertive testing (scripted) (test cases)

person ------------------ automation

enables

confirms
Automation enables confirm uncovers

KNOWLEDGE

EXPLICIT

TACIT & IMPLICIT

UNKNOWN S

(THAT WE'RE AWARE OF)

UNAWARENESS

(OF UNKNOWN S)

INFORMATION

INFORMS

Assertive Testing (Scripted) (Test Cases)

Person Automation

Investigative Testing (Exploratory) (Test Charters)
Automation enables confirms uncovers

I N F O R M A T I O N
KNOWN EXPLICIT
TACIT & IMPLICIT
UNKNOWNNS (THAT WE’RE AWARE OF)
UNAWARENESS (OF UNKNOWNS)

Investigative Testing
(Exploratory)
(Test Charters)

informs
uncover

Assertive Testing
(Scripted)
(Test Cases)

confirms enables assists

Person Automation

INFORMATION
**INFORMATION**

- **KNOWN**
  - **EXPLICIT**
- **TACIT & IMPLICIT**
- **UNKNOWN**
  - *(THAT WE’RE AWARE OF)*
- **UNAWARENESS**
  - *(OF UNKNOWN)*

**CONTEXT**

- **Assertive Testing** *(Scripted)* *(Test Cases)*
- **Investigative Testing** *(Exploratory)* *(Test Charters)*
- **Person**
- **Automation**

**Enables**

- **Informs**
Each of the 2 approaches: (Investigative Approach & Assertive Approach) are EQUALLY IMPORTANT

If you are only doing scripted assertive testing, then you're missing out A LOT of testing that you should be doing!
Where does this Testing fit within AGILE?
AGILE
Testing in Agile

**AGILITY:**
Adapting to the new information that we discover through investigation

**RISKS:**
Investigating to identify risks // Investigating the risks we've identified

**Continuous Testing:**
Testing the...
...ideas of new features, ...artefacts, ...ux/ui designs, ...code design, ...code, ...execution of the code, ...ci pipeline, ...release processes, ...agile processes

**Lateral & Critical Thinking**
Connect all 9 dots by drawing only 4 straight connecting lines...
Can you connect all 9 dots by drawing only 3 straight connecting lines this time?
Can you connect all 9 dots by drawing only 3 straight connecting lines this time?
TESTING isn't just about testing Developed Software
Continuous Testing

Testing the ideas, artefacts, designs, code, software processes and more...

Uncover (of unknowns)
TESTING in Dev-Ops
Plan → Branch → Code → Merge → Build → Deploy → Release → Monitor

We test here → And we test here → And we test here → And here of course...

Dev

Ops

Can we test here? Of course we can!
UNKNOWN INFO exists everywhere and always will...
(we can NEVER think of EVERYTHING)
Darkness represents our ignorance (unknowns AND unawareness)

The torch light represents our knowledge
The more we investigate, the more the light grows and the more we learn.

We are still ignorant to what else is out there...
But we are still ignorant to what else is out there...
We can never fully discover everything...
There will always be unknowns.

The more we learn, the more we realise the complexities...
And the Dunning-Kruger effect kicks in.
Multiple torches represent our investigation of different RISKS at different points in SDLC

But obviously, there is still continued ignorance...
But obviously, there is still continued ignorance... Even more than we know...
TESTING IN AGILE

Uncovering information continuously
to inform decisions and enable agility
to adapt and stay on track
with building the best product we can

ASK QUESTIONS
CHALLENGE ASSUMPTIONS
INVESTIGATE TO UNCOVER RISKS
INVESTIGATE THE RISKS
My challenges to you:

1) Try incorporating an Exploratory Approach AND an Automation Approach to your testing (if you aren't already doing so).

2) Try to sneak into the planning sessions to test the ideas, the artefacts and the designs

3) Try pairing with your developers to test the code and feed into the code design
THE CURVE OF ACCEPTANCE

Emotion

Complete denial
Physical push-back
Growing in understanding
Point of realisation
ACCEPTANCE!
Point of promotion
Active promotion and practice
Growing enthusiasm
Mastery and Fluent promotion

Time

Time taken to actually begin listening
Thank you!

- Dan Ashby
  @DanAshby04
  www.DanAshby.co.uk
  www.SoftwareTestingClinic.com
  www.TestingInThePub.co.uk