Ebony and Ivory
Devs and Testers in Perfect Harmony

Practical implementation of
ATDD / BDD / TDD

SoftEd

Talk objective

How to apply:

- **Acceptance Test Driven Development** (**ATDD**),
- **Behaviour Driven Development** (**BDD**) and
- **Test Driven Development** (**TDD**).

Colin and **Toby** walkthrough the practical application of ATDD, BDD and TDD.

- Optimising communication and collaboration in an agile project team

Introduction to ATDD, BDD and TDD

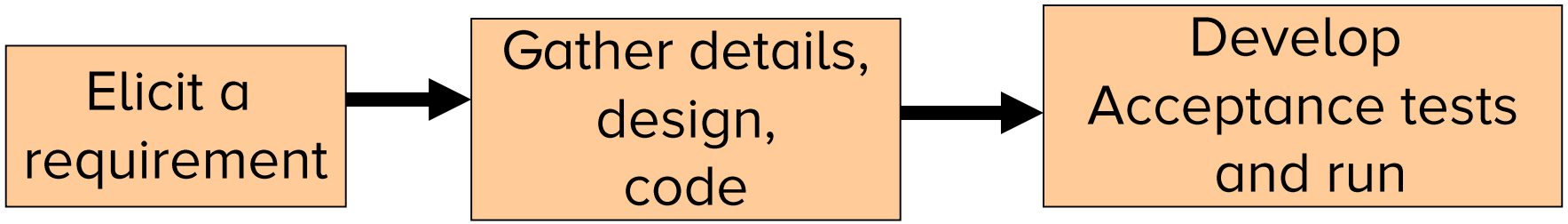
Meet Colin The Coder

Introduction to ATDD, BDD and TDD

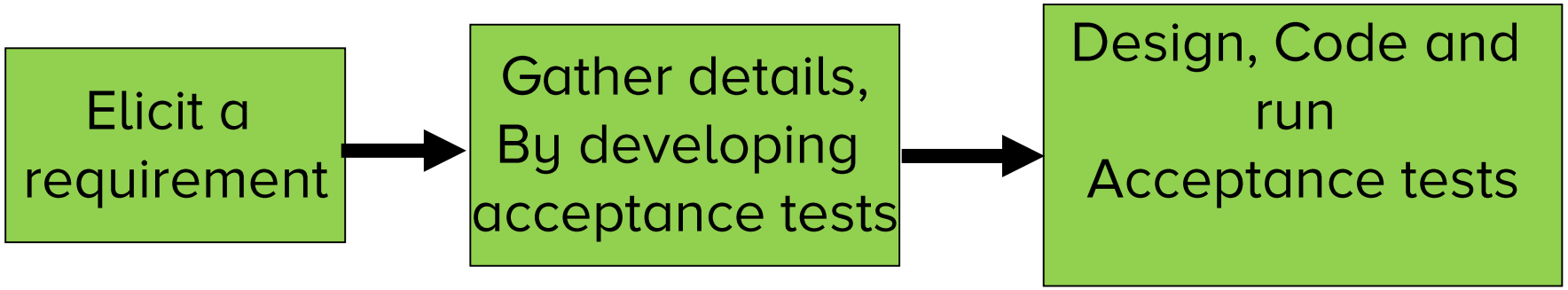
Meet Toby The Tester

Acceptance Test-Driven Development

1. The **TRADITIONAL APPROACH**

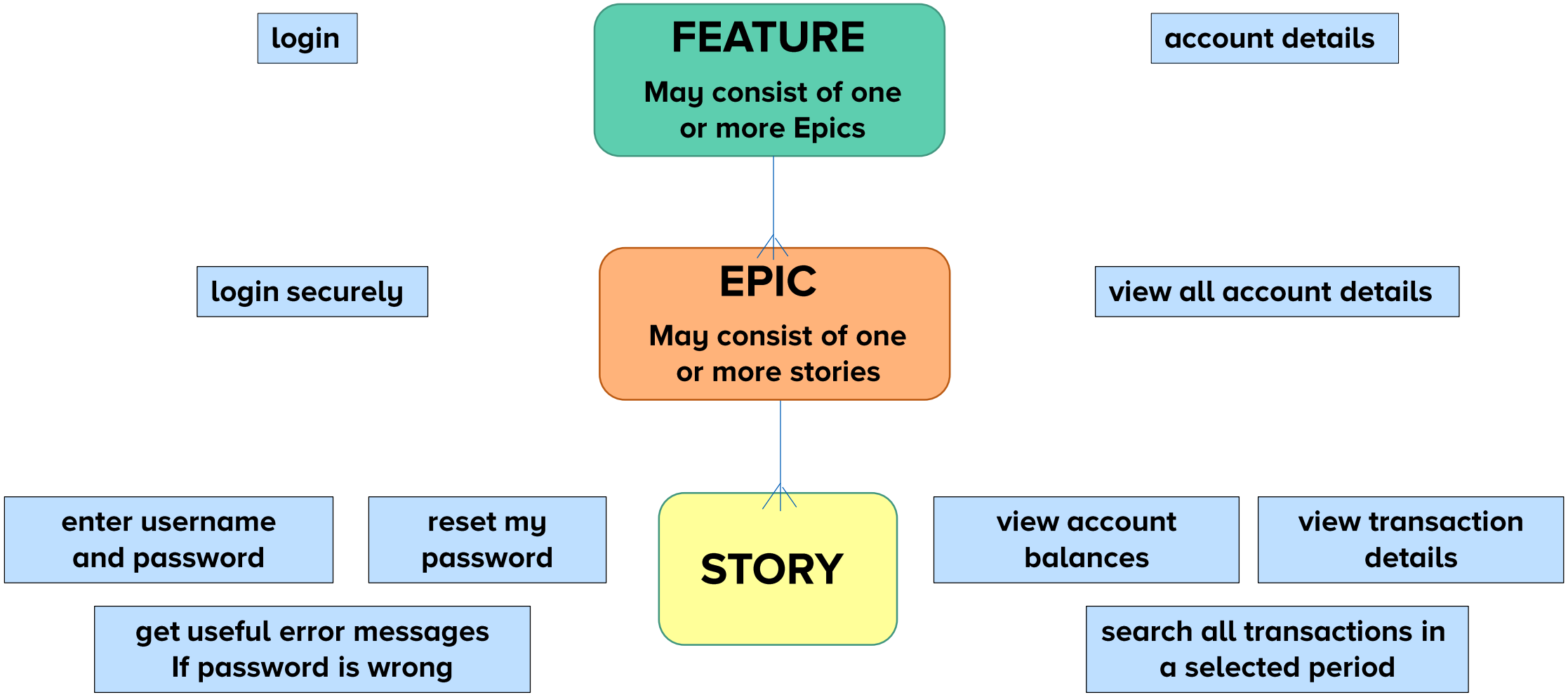


2. The **AGILE APPROACH**

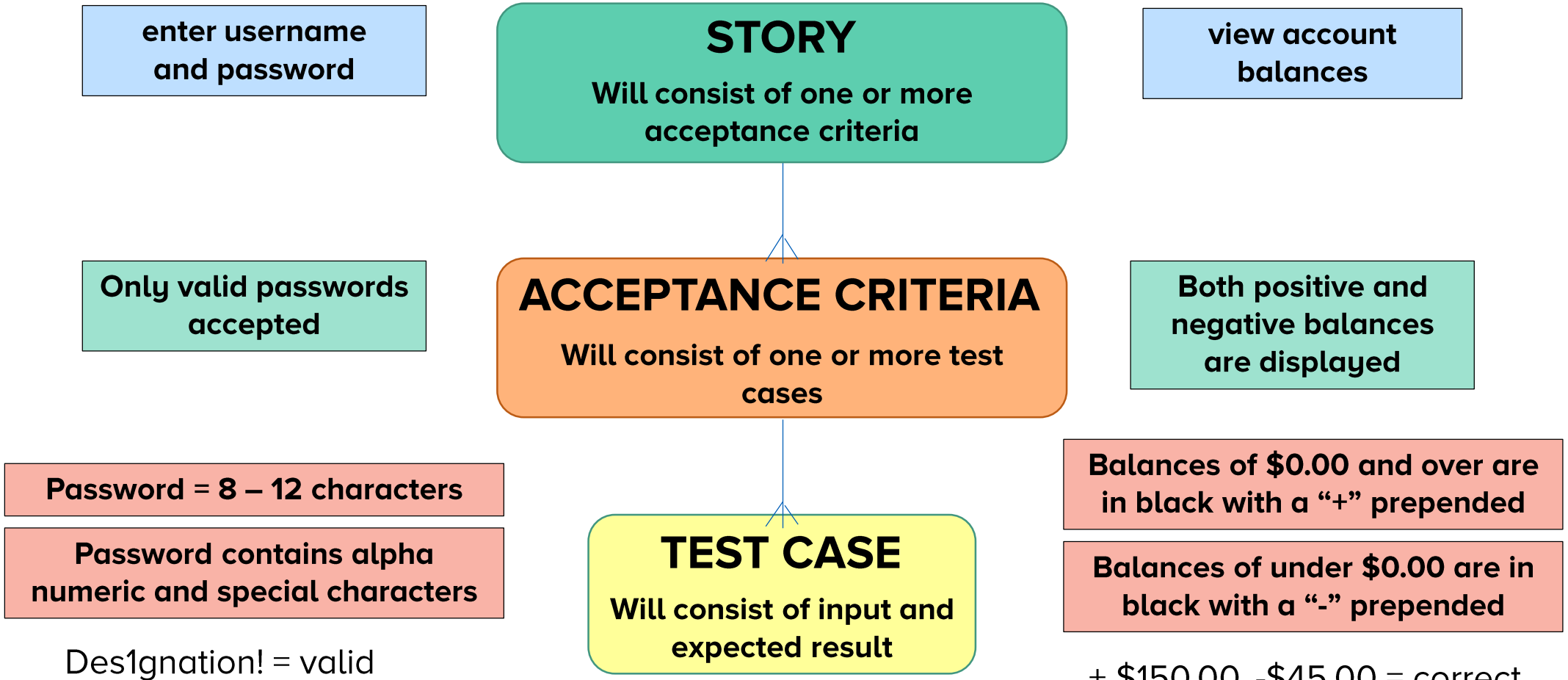


At the heart of ATDD, BDD and TDD is that tests are written before the code is implemented

Feature, epic and story

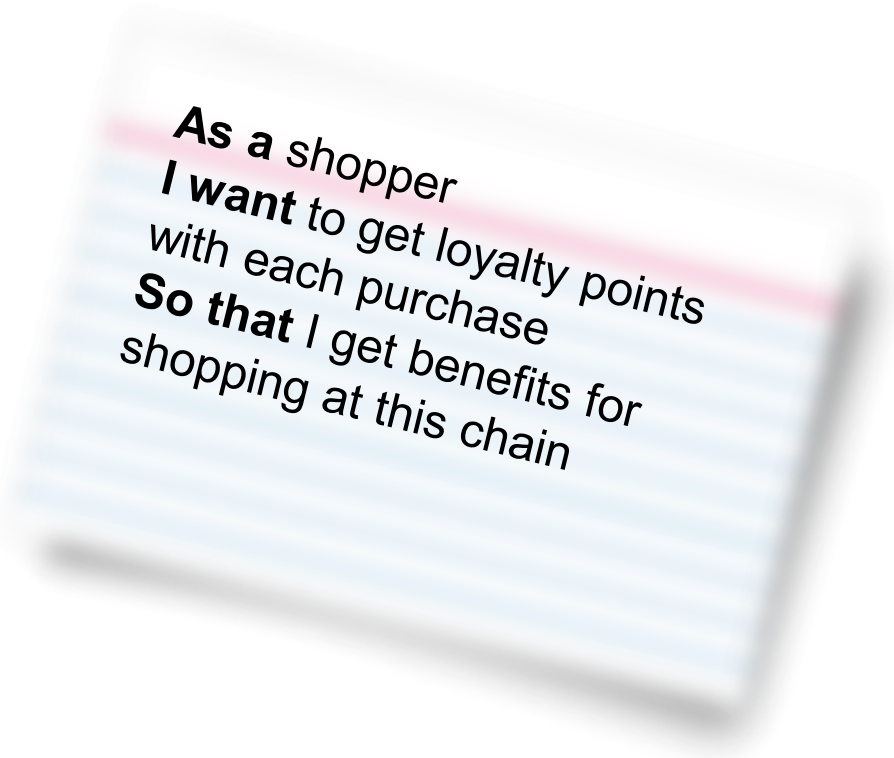


Feature, epic and story cont.



User story testing

- Basic unit of work
- Building block of all work to be done
- Reflected in definition of ‘done’
- Can be very difficult to interpret correctly!



Acceptance Test Driven Development

Since acceptance test-driven development is a test first approach, test cases are created prior to implementing the user story.

The test cases are created by the Agile team including the **developer**, the **tester**, and the **business representatives (aka Power of Three, Three Amigos)**

The tests are specified using terms that everyone in a specific domain understands

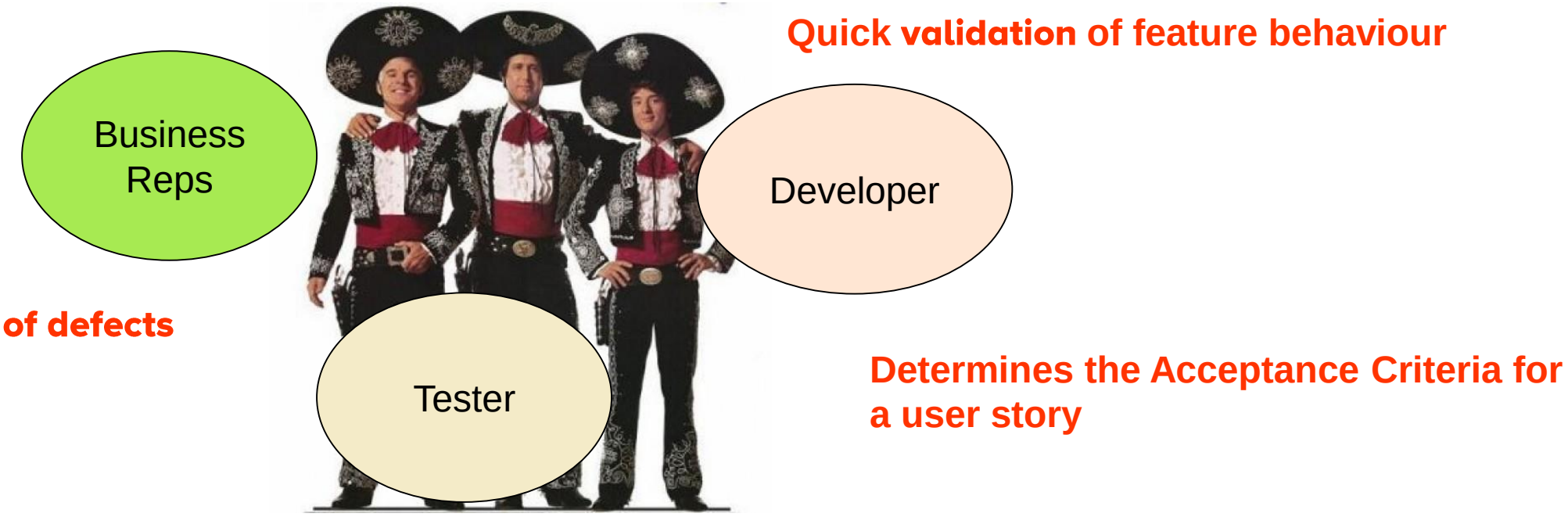
The goal is shared understanding and shared vocabulary about features / stories.

Avoid multiple interpretations of the user stories we are building.

Test cases can be Manual or Automated

Acceptance Test-Driven Development

- The whole team collaborates in order to understand how the feature has to behave and What the **developers, testers, and business representatives** need to do to ensure this behaviour.



Driving development with acceptance tests

ATDD is about “expressing functional story requirements as **concrete examples** or **expectations** prior to story development.” **Jennitta Andrea- Iterations**

Not everyone uses the same terminology, so ATDD can also be referred to as or is similar to:

- **Story Test Driven Development,**
- **Example Driven Development,**
- **Specification By Example**

and…

- **Behaviour Driven Development (BDD)**

Behaviour-Driven Development

- Tests are expressed in a way that every stakeholder is able to understand, containing sentences in natural language involving the necessary precondition (Given), if any, the input, (When) and the related output (Then)

  - **GIVEN some initial context,**
  - **WHEN an event occurs,**
  - **THEN ensure some outcomes.**

- From these requirements, the behaviour-driven development framework generates **code that can be used by developers** to create **test cases**.

BDD
ATDD
TDD

BDD in action

Customer withdraws cash

As a customer,
I want to withdraw cash from an ATM,
so that I don't have to wait in line at the bank.

Scenario: Account is in credit
Given the account is in credit
And the card is valid
And the dispenser contains cash
When the customer requests cash
Then the account is debited
And cash is dispensed
And the card is returned

Discounts

As a frequent shopper

I want to apply a discount code

So that I don't have to pay the full amount for the product

Scenario 1: 10% discount

Given the product "BBQ" costs $1000
And the code "SAVE" gives a 10% discount

When the cost of "BBQ" with the code "SAVE" is calculated

Then the cost should be $900

Scenario 2: No discount

Given the product "BBQ" costs $1000

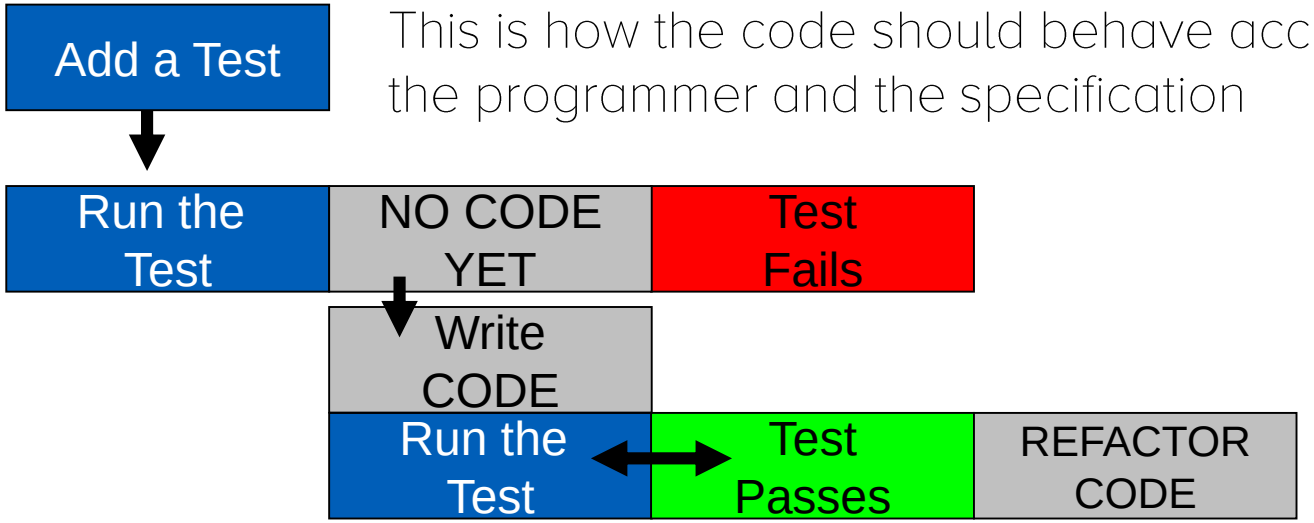
When the cost of "BBQ" with the code "ZERO" is calculated

Then the cost should be $1000

We can embed examples into the scenarios

Test-Driven Development

TEST DRIVEN DEVELOPMENT – not a test technique



Repeat this process for the next small piece of code, running the previous tests as well as the added tests

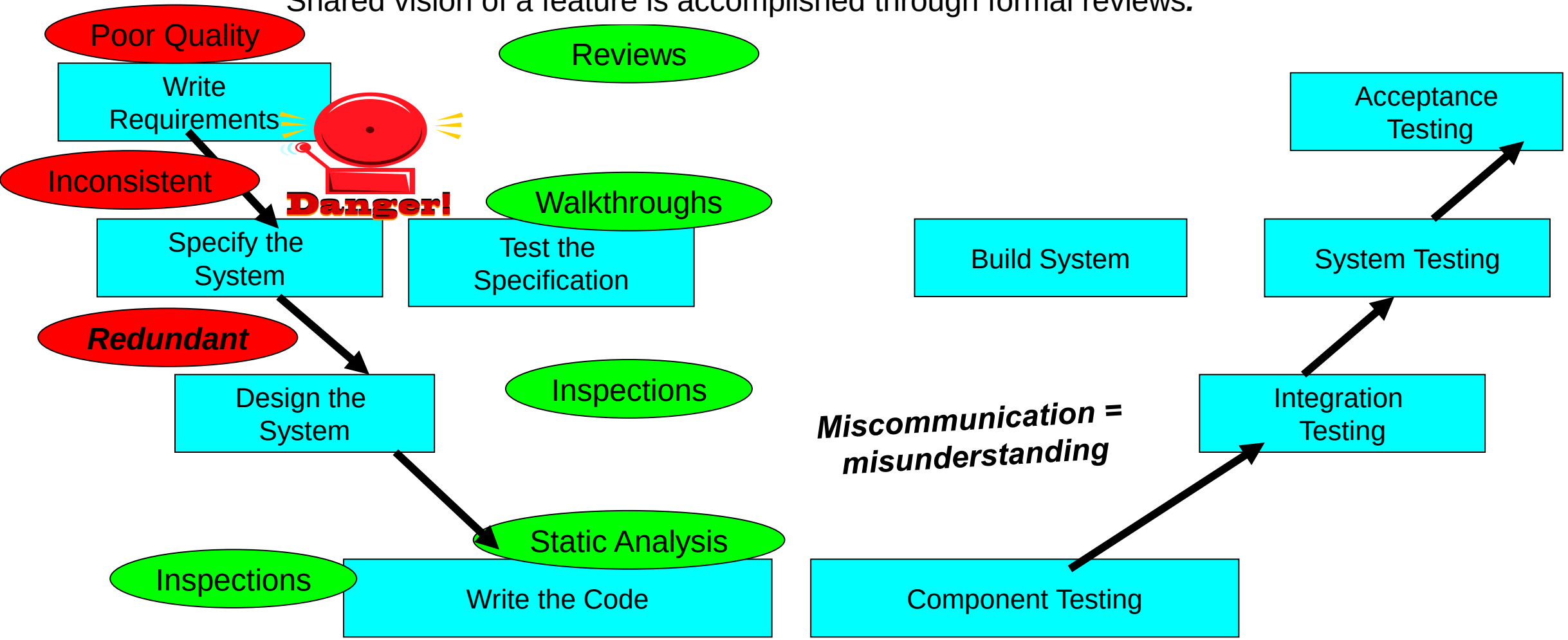
Colin wants a fast car?

It takes 2 minutes to reach 160 km/h

Excellent – so we all agree then….I'm glad we're on the same page….

In a more traditional setting...

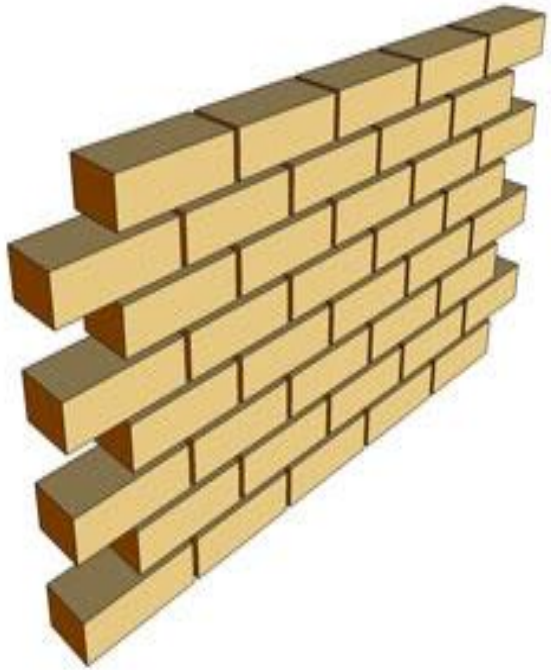
Shared vision of a feature is accomplished through formal reviews.



Excellent – so we all agree then….I’m glad we’re on the same page…

In the past....

Colin and Toby may have been separated by this



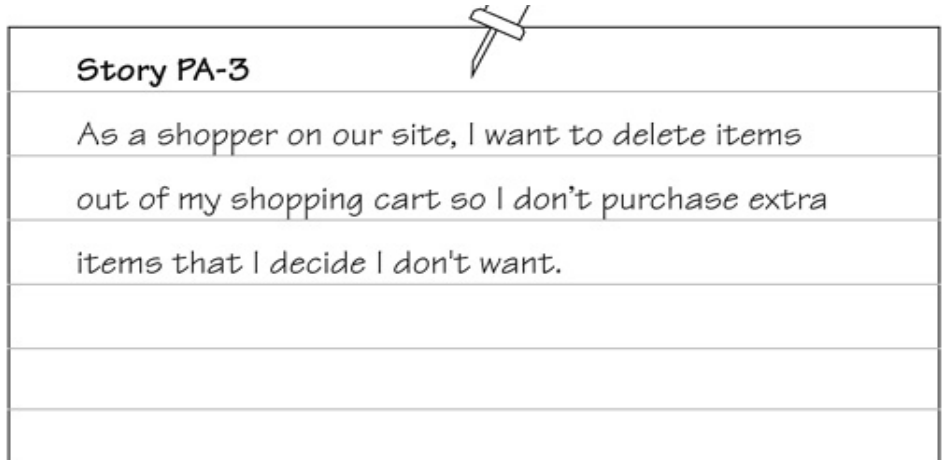
Now they work together

...in perfect harmony

Applying Acceptance Test-Driven Development

STEP 1

- The first step is a **specification workshop, backlog refinement, elaboration session** where the user story is analysed, discussed and written by developers, testers, and business representatives.



How will we delete? Is it consistent with other ways of deleting? What happens to the deleted item?

Ambiguity, omissions, defects in the user story are fixed during this process.

Applying Acceptance Test-Driven Development

STEP 2

- **Create the tests** – either by:
  - **The team together** or by the
  - **Tester individually.**
- An independent person such as the business representative validates the tests.
- The tests are **examples** that describe the specific characteristics of the user story. These **examples** will help the team to implement the user story correctly.

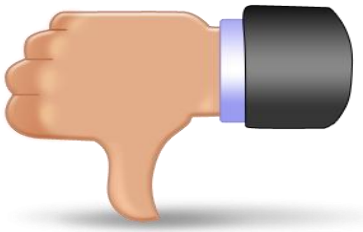
Applying Acceptance Test-Driven Development

Start with positive path tests.

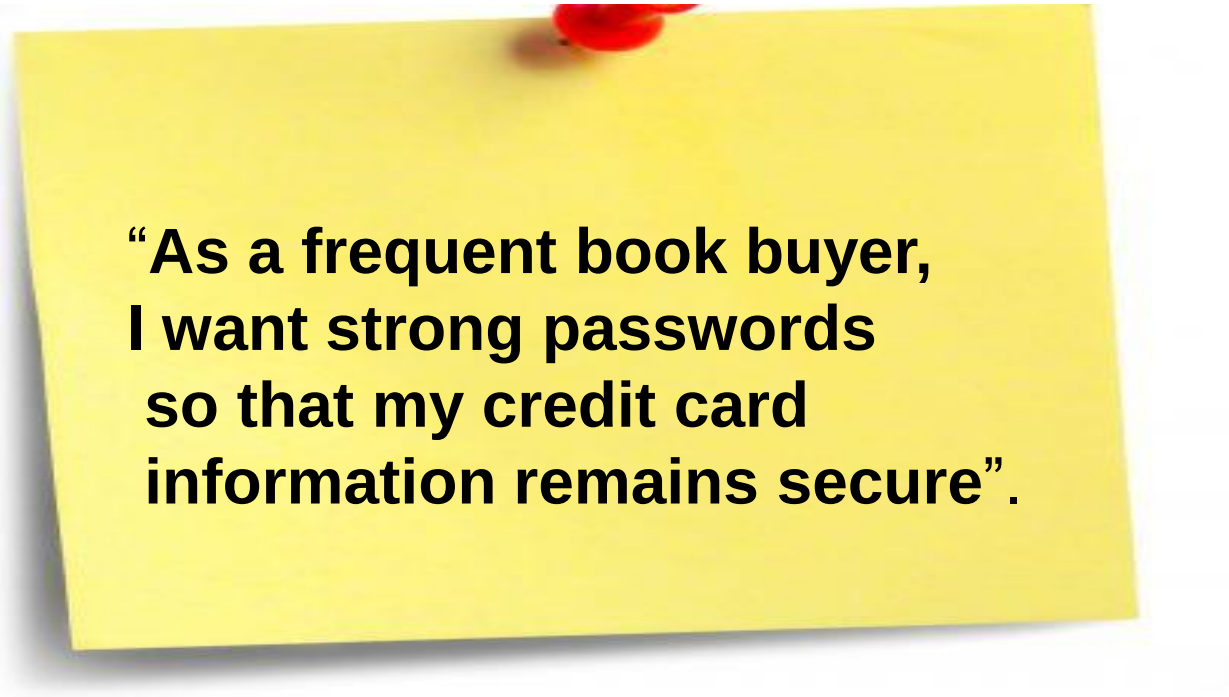
- Confirm correct behaviour without exception or error conditions, comprising the sequence of activities executed if everything goes as expected.

Followed by negative path tests

and cover **non-functional attributes** as well (e.g., performance, usability).



Let's give it a go...



In order to confirm we are on the right track we need to talk to a business stakeholder – typically a Product Owner, Business Representative or a Business Analyst...

Meet Brian The BA

The aim is to gain a better understanding of what Brian wants and that we all understand what is meant by ‘strong passwords’. (recall ‘fast car’)

“As a frequent book buyer,
I want strong passwords
so that my credit card information remains secure”.

Does this apply to our existing customers?
Should we force them to change their passwords?
Should we allow spaces in the password?
How long should the password be?
Can you give me examples of weak passwords?
How can we enforce people to not use the names of their pets etc?
Can you give me examples of strong passwords?
What should the password contain? Letters and Numbers?
Do we care about case sensitivity?
What are examples of ‘special characters’?
What is the outcome when we enter valid passwords ?
What is the outcome when we enter invalid passwords ?

Brian The BA

Password Can't be too short ...6 is too short , what's too much – yes 8 sounds right

Don't worry about our existing customers for now

Too weak: 12345678, abcdefg, password, P@ssword, Dictionary word

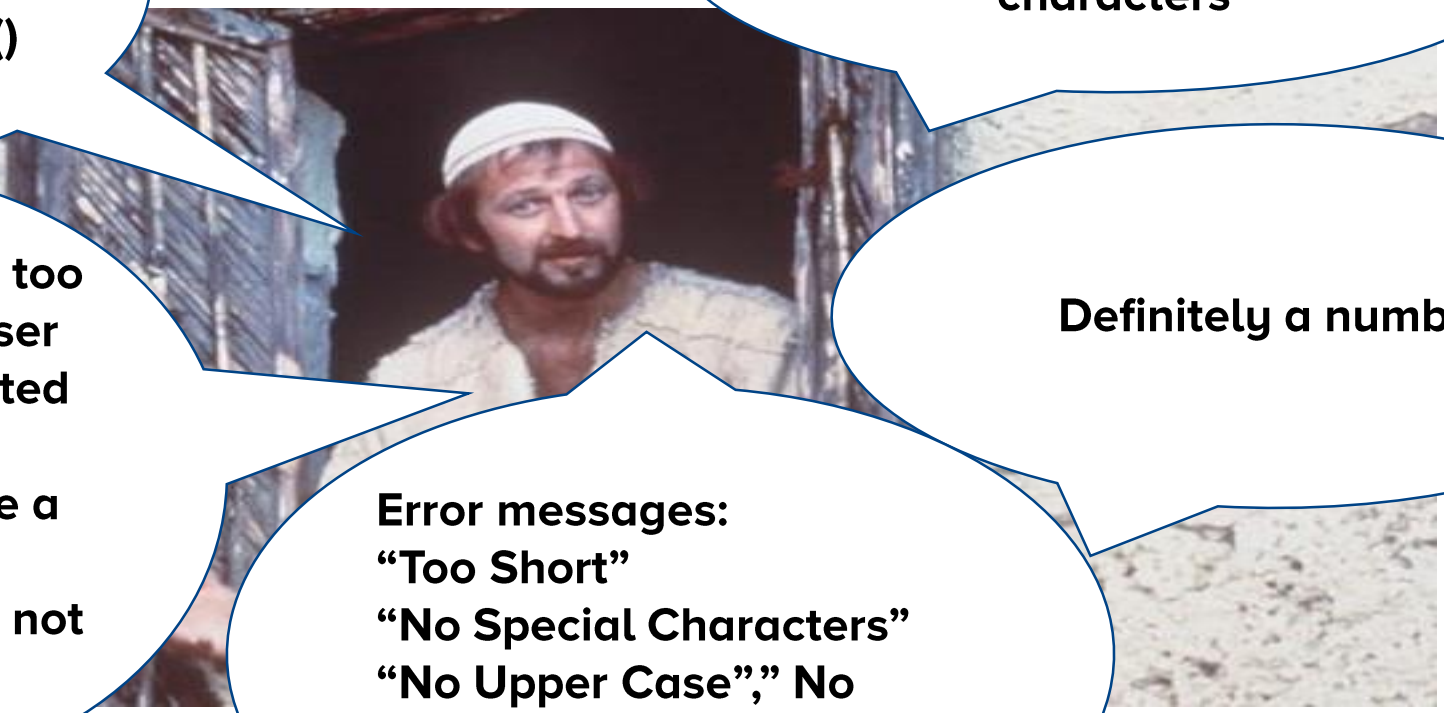
I don't think we should include spaces – too complicated

Too weak: No family pets, names like Brian12, or iamthemessiah, My phone number, address

Brian The BA

Special characters
At the top of the keyboard !@#$%^&*()

I think it needs Upper and Lower case alpha characters



If the password is too weak then the user should be prompted with an error message: “you’re a very naughty boy”…no perhaps not that

Definitely a number

Error messages:
“Too Short”
“No Special Characters”
“No Upper Case”,” No Lower Case”, “No numbers”

Agree on the acceptance criteria

"**As a frequent book buyer, I want strong passwords so that my credit card information remains secure**".

- Test that the system allows passwords with all of the following characteristics:
  - At least 8 characters
  - Contains at least one character from each of the following groups (equivalence partitioning):
    - Lower case alphabet
    - Upper case alphabet
    - Numbers
    - Special Characters (!,@,#,$,%,^,&,*)
- Test that the system disallows passwords that are lacking any one of the above characteristics.
  - **Test for invalid inputs i.e. leaving out any of the characters above**

Understandable Examples

After many conversations, the whole team have come up with a shared understanding of the acceptance criteria.

- No code
- No automation
- Just a Shared Understanding of what we mean by the acceptance criteria

Understandable Examples

In this case the team have come up with real examples of inputs and outputs to gain this shared understanding of the acceptance criteria

Data	Expected Result	Expected Message
IsThis$AGood11	Pass	
AAbbCC11	Fail	No Special Characters
$$$bbb111	Fail	No Upper Case
AAA%%	Fail	No Lower Case
AAAA%%%%bbbbb	Fail	No numbers
Aa9ab$	Fail	Too Short

We all got on pretty well

Now let's get Colin the Coder to talk through how BDD/TDD works

Coder (Developer)

Business Analyst

Tester

What about the tools Colin used?

All of these tools are free!

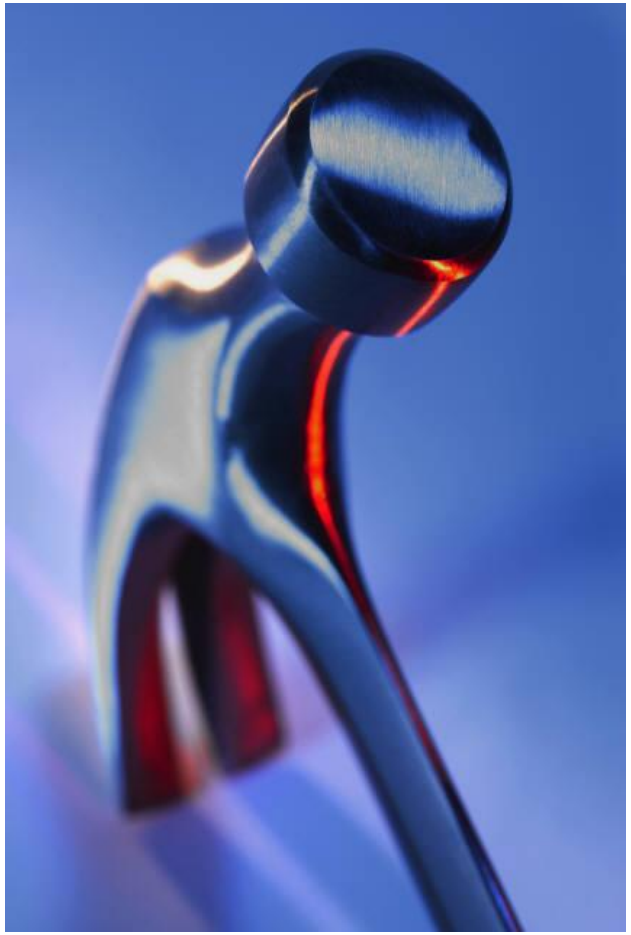
NUnit – TDD

Cucumber and Gherkin – ATDD / BDD

Git – Version Control

Jenkins – Continuous Integration Server

TestStack.White – Automate the User Interface



Summary

We have walked through the approaches of:

- **Acceptance Test Driven Development** (ATDD)
- **Behaviour Driven Development** (BDD) and
- **Test Driven Development** (TDD).

The tests we created were automated and the test results reported.

- Most importantly we were able to Collaborate and Communicate and effectively build and test the right product

References

Agile Atlas User Stories

Driving development with tests: ATDD and TDD Elisabeth Hendrickson

Lean-Agile Acceptance Test-Driven Development, Ken Pugh

Agile Testing , Crispin and Gregory

More Agile Testing , Crispin and Gregory

Bridging the Communication Gap, Gojko Adzic

Specification by Example, Gojko Adzic

Test-Driven Development By Example, Kent Beck

Any questions?

SoftEd

SoftEd

Educate • Enrich • Evolve

Visit www.softed.com

Email info@softed.com

Phone **0800 268 773**

Follow Us

http://blog.softed.com

@SoftEdMan

Software Education