Why You <u>Want</u> to Practice Behavior Driven Development

Even Though You Might Not Know it Yet!

George Dinwiddie — @gdinwiddie

What?

BDD is…

- A practice that has many facets
- A practice that is widely misunderstood or misconceived with half-truths.

https://en.wikipedia.org/wiki/File:Blind_men_and_elephant3.jpg

BDD is…

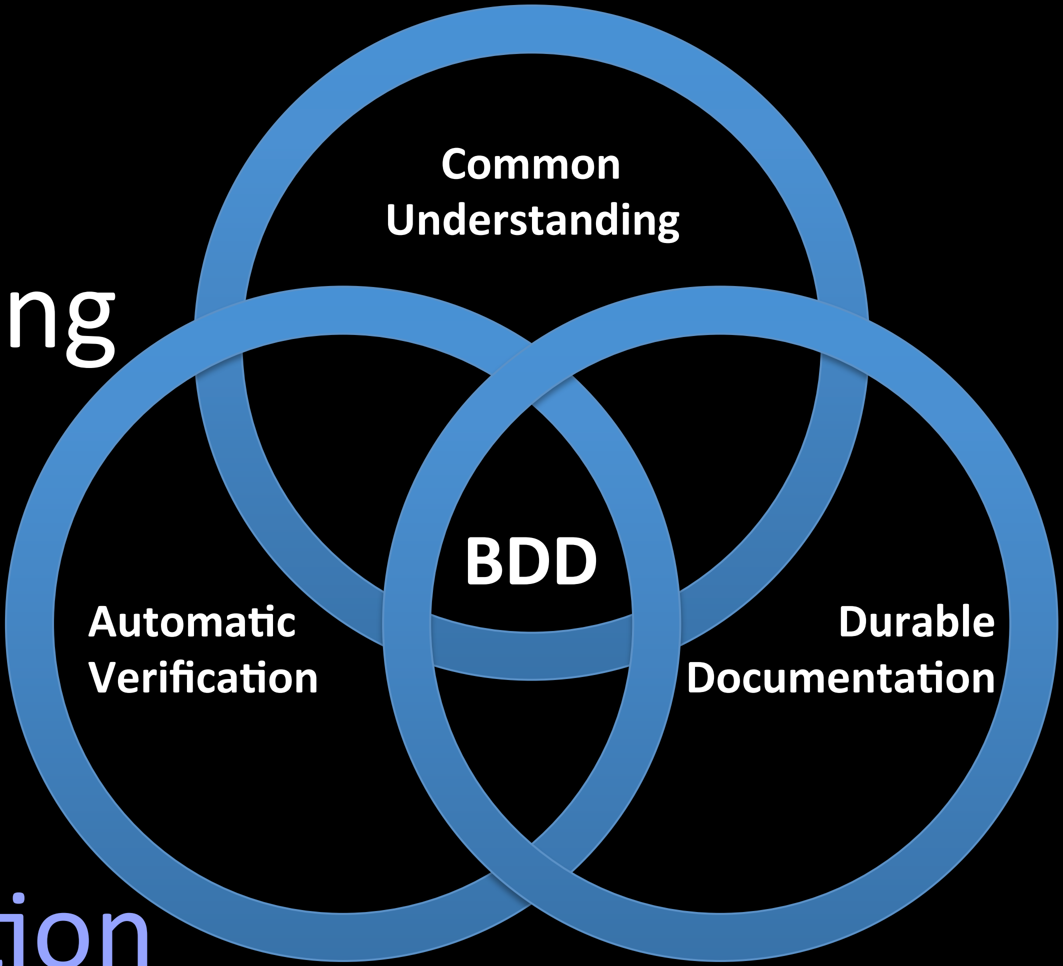
- A **conversation** about how you want the system to act, illustrated with examples
- A **meeting of the minds** among the stakeholders and implementers

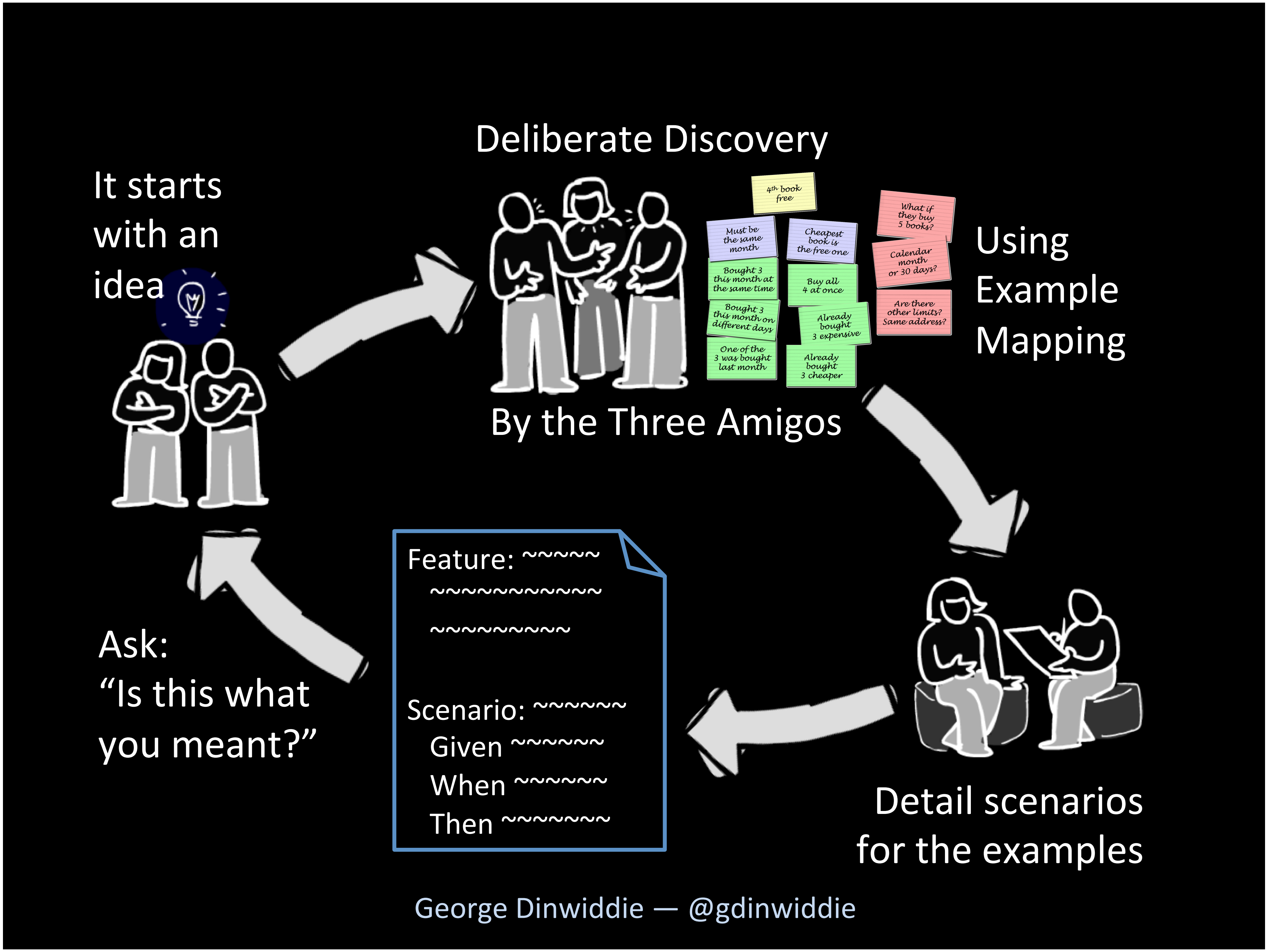
These stakeholders and implementers are often called the **Three Amigos** because they include

- A business representative
- A programmer
- A tester

BDD Outcomes

- Common Understanding
- Automatic Verification
- Durable Documentation

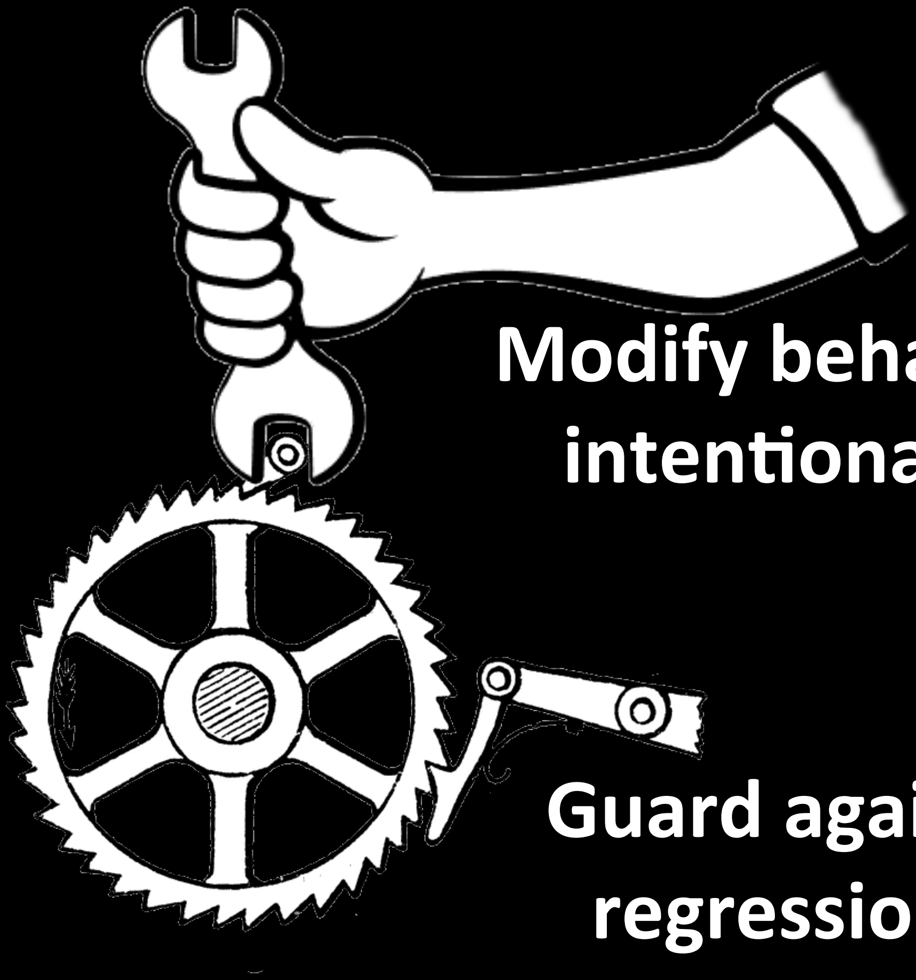



It starts with an idea
Deliberate Discovery
4th book free
Must be the same month
Cheapest book is the free one
Bought 3 this month at the same time
Buy all 4 at once
Bought 3 this month on different days
Already bought 3 expensive
One of the 3 was bought last month
Already bought 3 cheaper
What if they buy 5 books?
Calendar month or 30 days?
Are there other limits? Same address?
Using Example Mapping
By the Three Amigos
Detail scenarios for the examples
Feature: ~~~~~
~~~~~~~~~~~~~
~~~~~~~~~~
Scenario: ~~~~~~
Given ~~~~~~
When ~~~~~~
Then ~~~~~~~
Ask: "Is this what you meant?"

George Dinwiddie — @gdinwiddie

Executable Description

Document the intent



Modify behavior intentionally

Guard against regressions

Remember, as Liz Keogh says…

Having conversations

is more important than
capturing conversations

is more important than
automating conversations

Why?

You’re programming

and you discover that

you don’t know how

the system should behave

for some situation

Planning meeting run long

as people discuss

what's in and what's not in

a story being considered

You’re programming and

something in the requirements doesn’t make sense

You're programming and

you can interpret

something in the requirements

two ways,

but you can't do both

You code a feature

and the tester says

“That’s not how this should work!”

The programmer and tester go talk
to the business analyst
to find out which one is right

And they're <u>BOTH</u> wrong.

You build some functionality

and then find out the

requirements were wrong

Your requirements document

asks for functionality

that's impossible to create

Some of the functionality
the business wanted
wasn’t specified in
the requirements document

You’re testing
some new functionality
with users
and discover
it’s not doing what
you intended to test

Finished code was
handed to testers and
came back for bug fixes

Functionality you wrote

and checked that it worked

is now broken

Functionality that was delivered

in a previous release

now doesn't work

You upgraded a library
or framework and
don’t know what effects that has
on your application’s functions

Testing takes too long

when you need

to ship a release

You can’t tell

how far along you are

in developing the functionality

the business wants

You’ve fixed

the same bug

more than once

You’ve forgotten

how a function

written last year

works

The business asked you

what are the business rules

for a specific situation

You think

it’s going to be hard

to get started

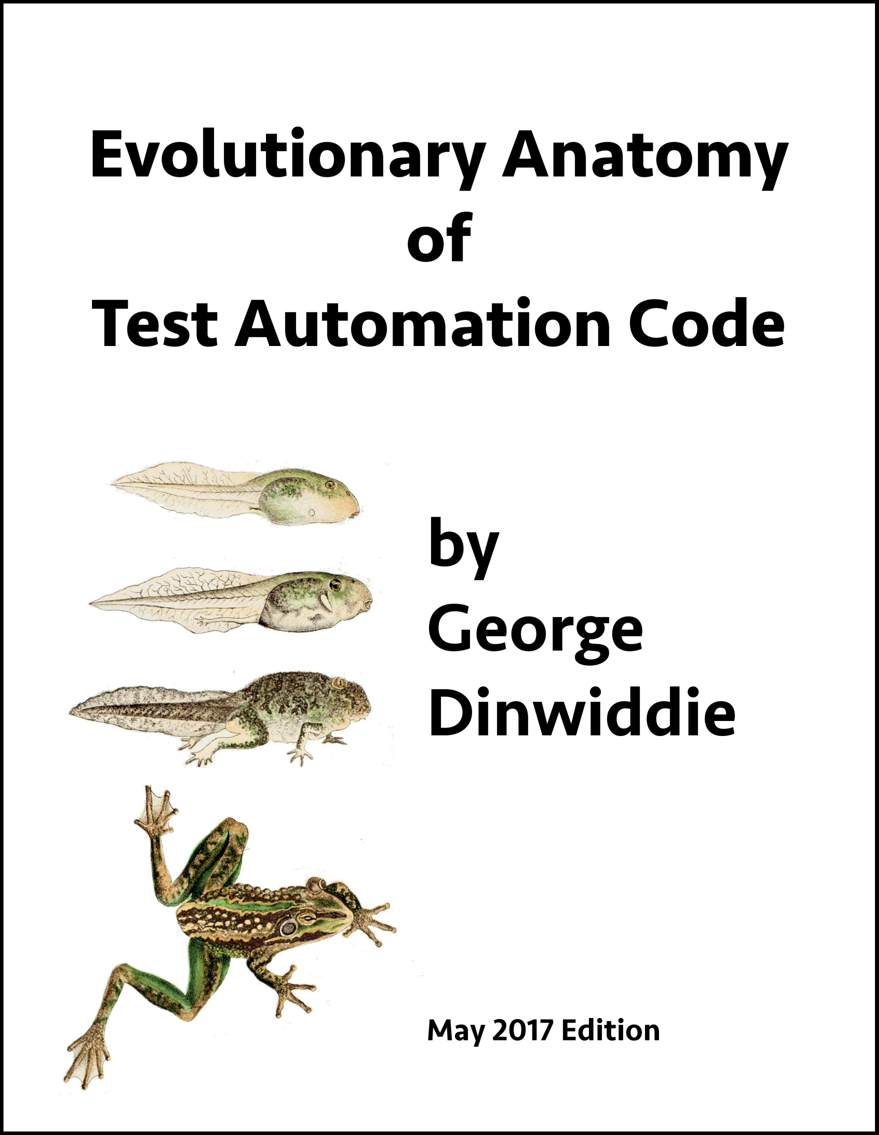
doing BDD

Basic BDD Resources

- **Introducing BDD**, by Dan North
  http://dannorth.net/introducing-bdd/
  - *A description from 2006 of the beginnings and essentials of BDD*
- **ATDD vs. BDD, and a potted history of some related stuff**, by Liz Keogh
  http://lizkeogh.com/2011/06/27/atdd-vs-bdd-and-a-potted-history-of-some-related-stuff/
  - *BDD described in the context of some related ideas*

Available on LeanPub

Code on GitHub



Evolutionary Anatomy of Test Automation Code

by George Dinwiddie

May 2017 Edition