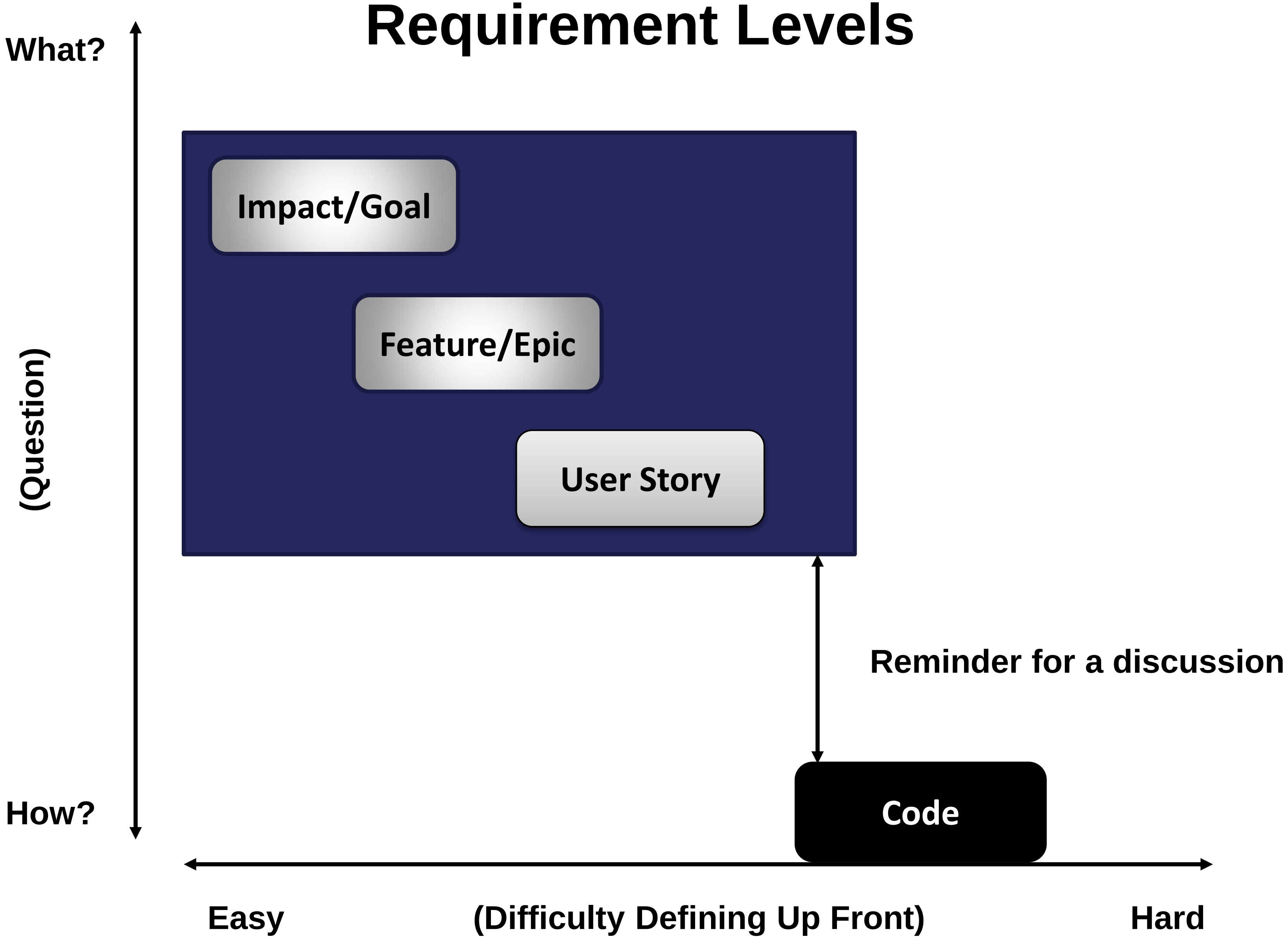
Partnered with thousands. Delivered millions.

Bridging the Communication Gap

Todd Meinershagen

Requirement Levels



What?

(Question)

How?

Impact/Goal

Feature/Epic

User Story

Reminder for a discussion

Code

Easy

(Difficulty Defining Up Front)

Hard

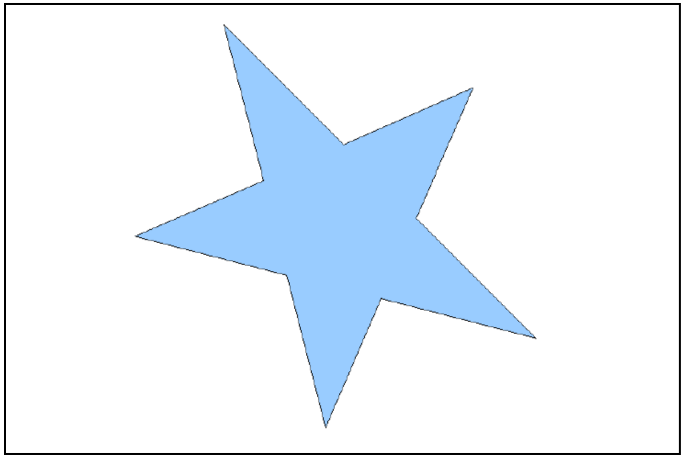
Items to Discuss

1. **Acceptance Criteria:** Specification by Example
2. **Automated Acceptance Testing:** SpecFlow
3. **Living Documentation:** Relish/Pickles

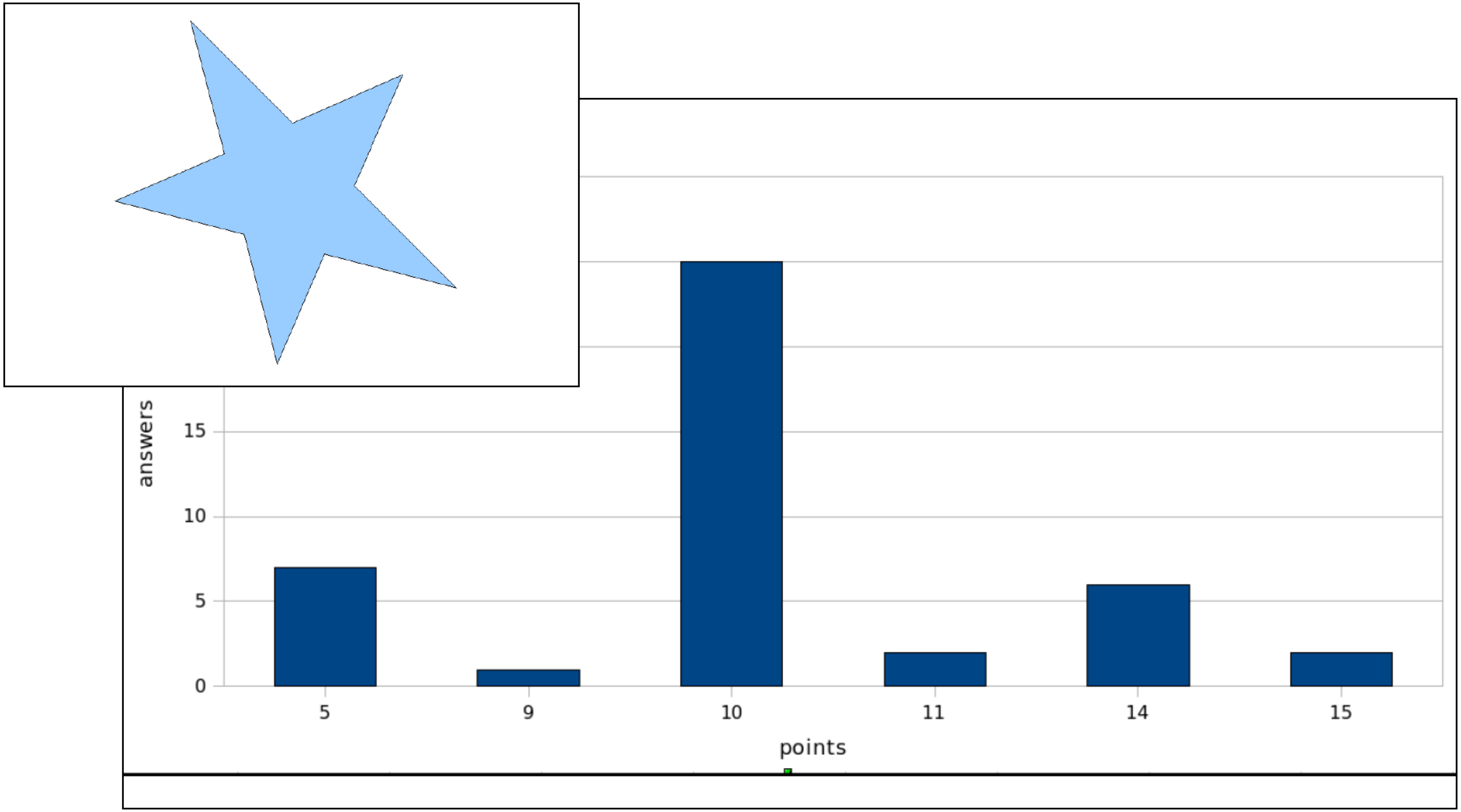
Items to Discuss

1. **Acceptance Criteria:** Specification by Example
2. **Automated Acceptance Testing:** SpecFlow
3. **Living Documentation:** Relish/Pickles

Traditional requirements are too abstract and take obvious things for granted.

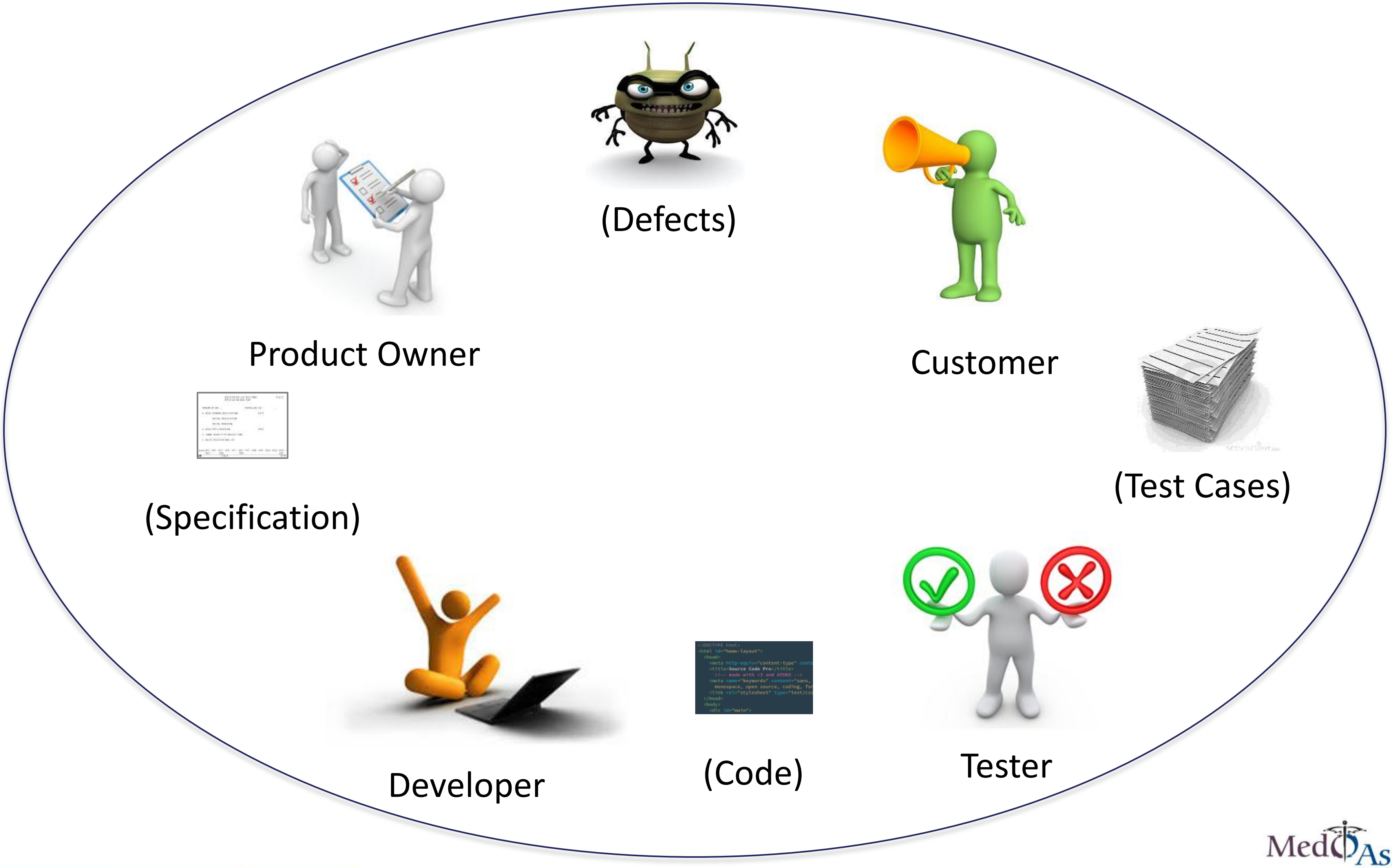


Traditional requirements are too abstract and take obvious things for granted.



An example is better than a list of instructions.

Examples live everywhere.



(Defects)

Product Owner

Customer

(Specification)

(Test Cases)

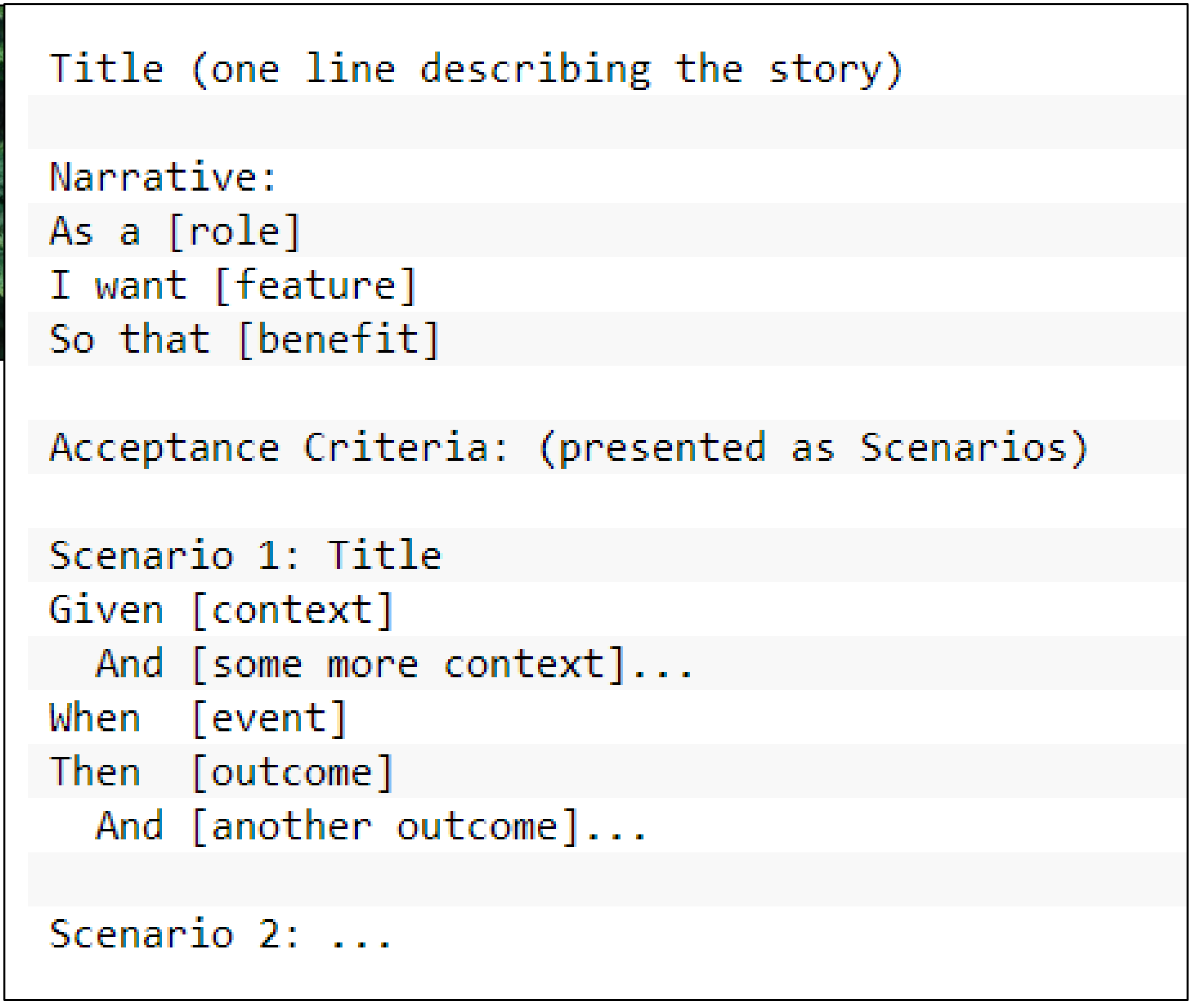
Developer

```
<!DOCTYPE html>
<html id="home-layout">
  <head>
    <meta http-equiv="content-type" conte
    <title>Source Code Pro</title>
      <!-- made with <3 and AFDKO -->
    <meta name="keywords" content="sans,
      monospace, open source, coding, for
    <link rel="stylesheet" type="text/css
  </head>
  <body>
    <div id="main">
```

(Code)

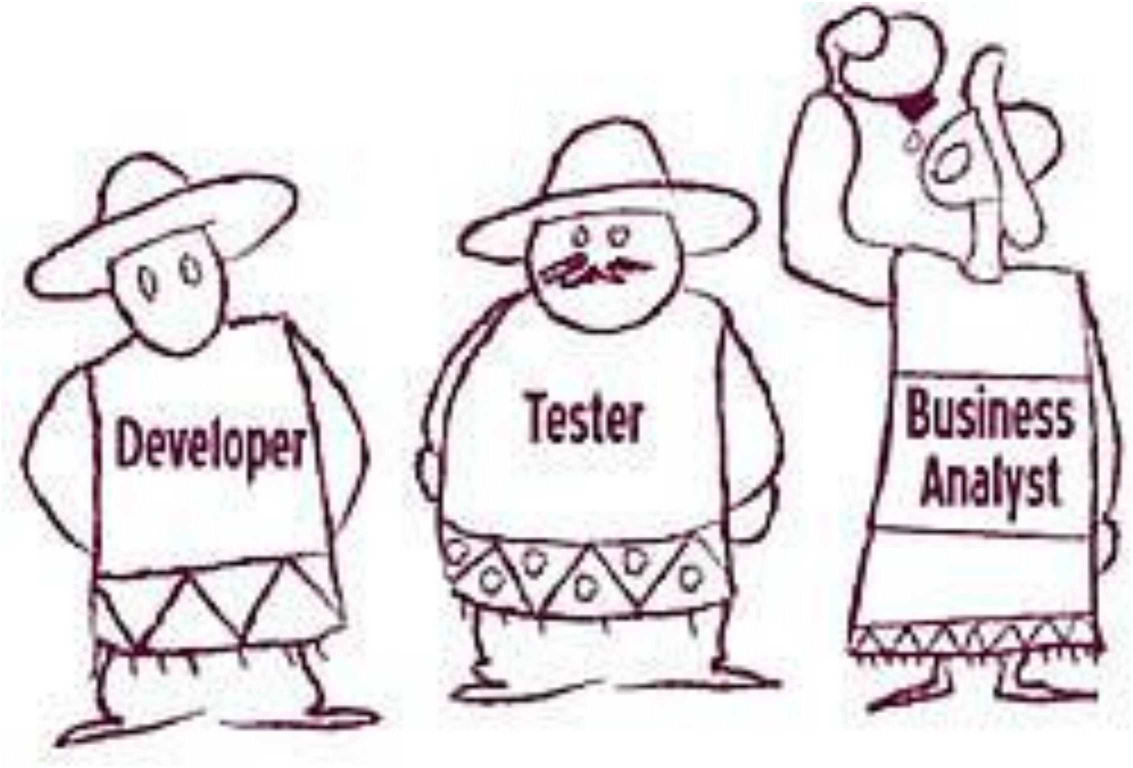
Tester

Use a consistent, gherkin syntax.

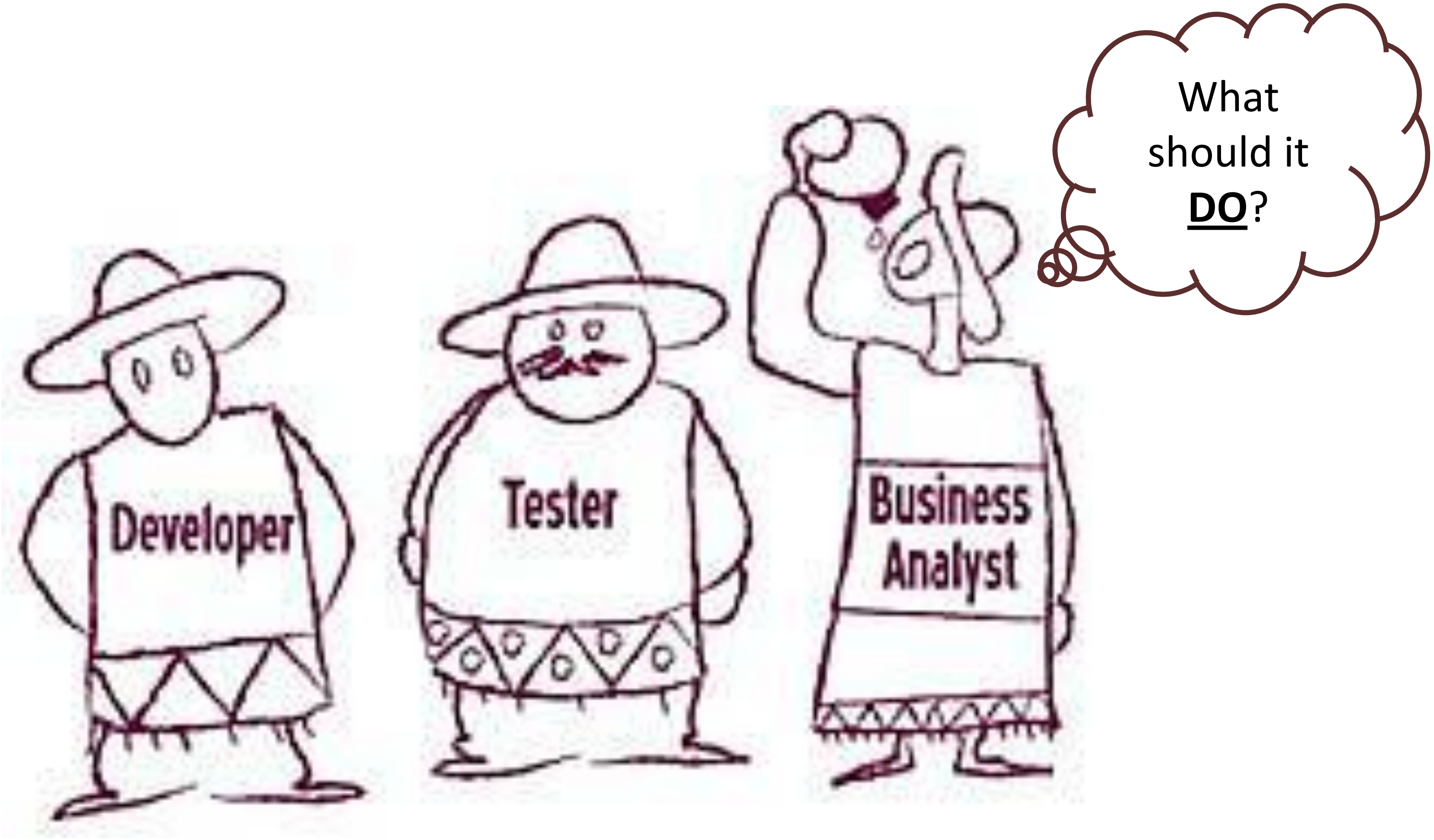


```
Title (one line describing the story)

Narrative:
As a [role]
I want [feature]
So that [benefit]

Acceptance Criteria: (presented as Scenarios)

Scenario 1: Title
Given [context]
  And [some more context]...
When  [event]
Then  [outcome]
  And [another outcome]...

Scenario 2: ...
```

Specify collaboratively as much as possible.



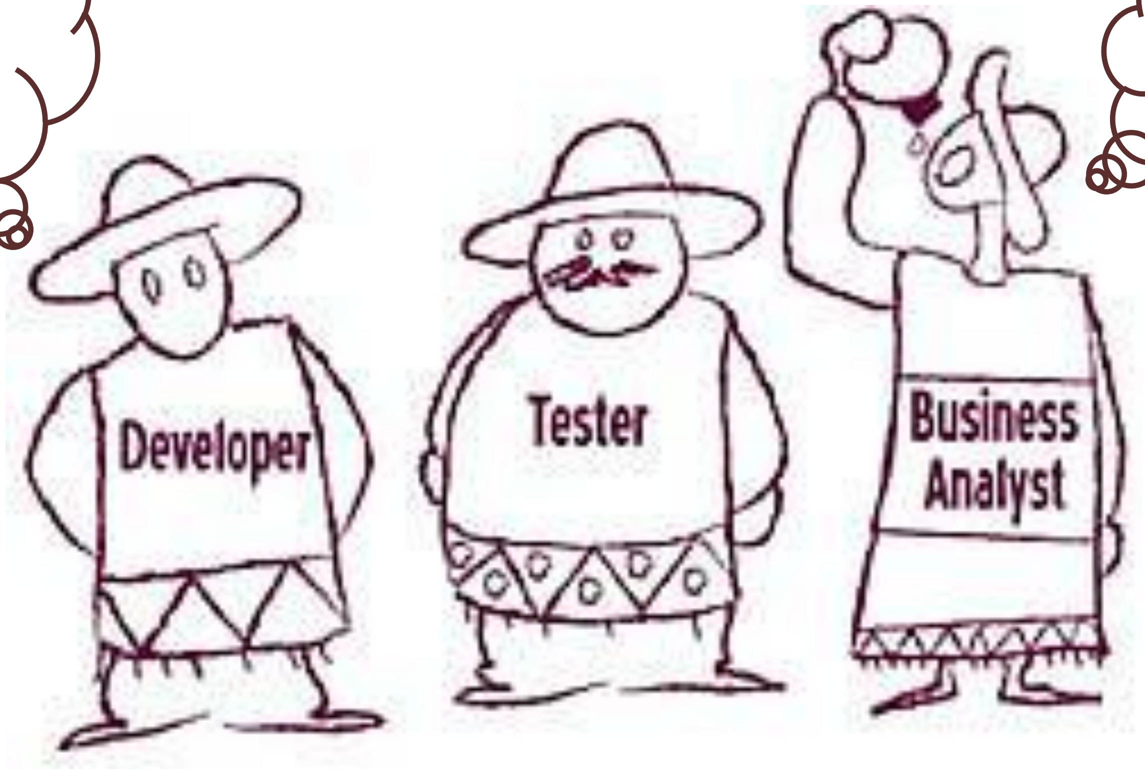
Specify collaboratively as much as possible.



What should it **DO**?

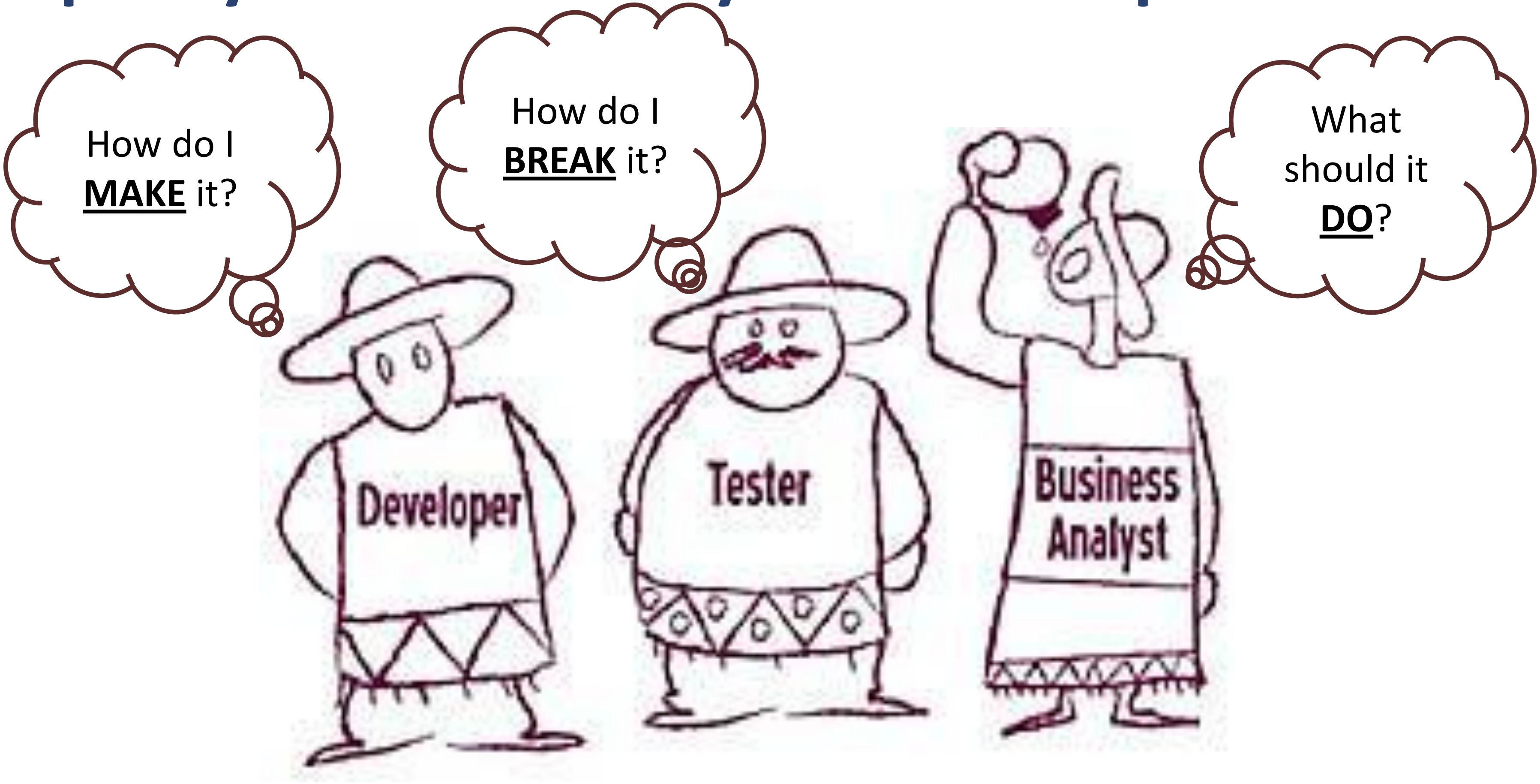
Specify collaboratively as much as possible.

How do I **MAKE** it?



What should it **DO**?

Specify collaboratively as much as possible.



How do I **MAKE** it?

How do I **BREAK** it?

What should it **DO**?

Items to Discuss

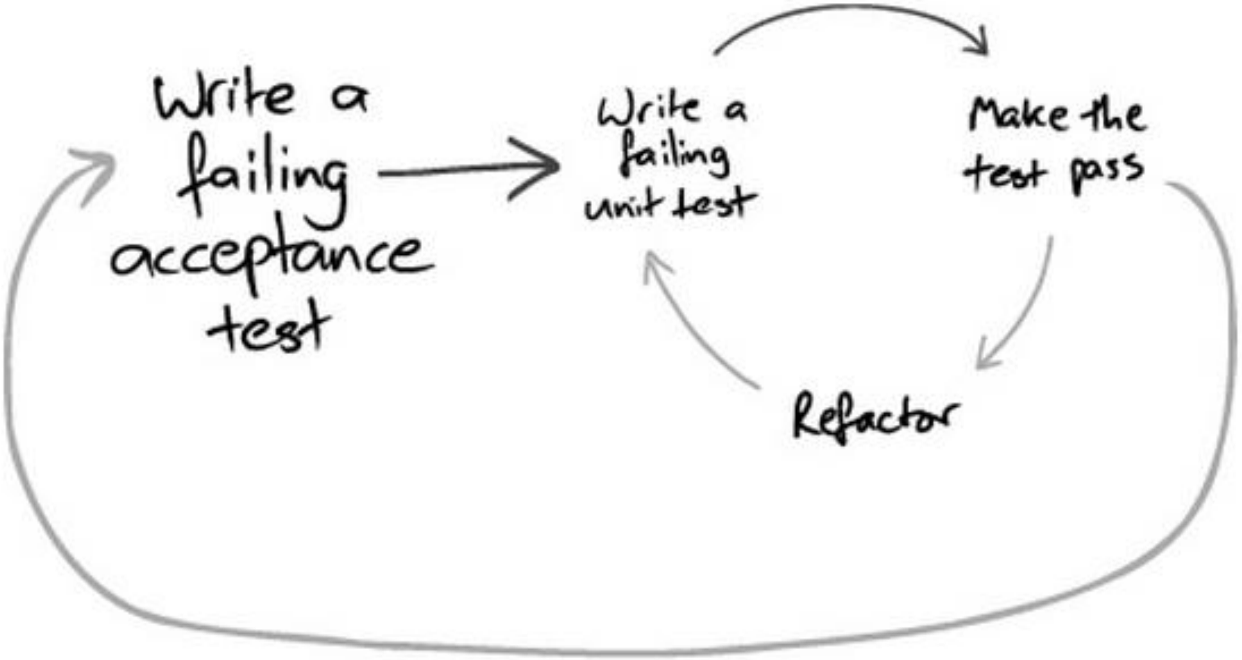
1. **Acceptance Criteria:** Specification by Example
2. **Automated Acceptance Testing:** SpecFlow
3. **Living Documentation:** Relish/Pickles

specflow :: About

PRAGMATIC BDD FOR .NET

- Created in 2009 and sponsored by IT WORKS - WE CARE TECHTALK
- Open source, available via nuget
- More info at http://www.specflow.org

specflow PRAGMATIC BDD FOR .NET :: Feature-level, not unit-level.



specflow PRAGMATIC BDD FOR .NET :: Demo

Items to Discuss

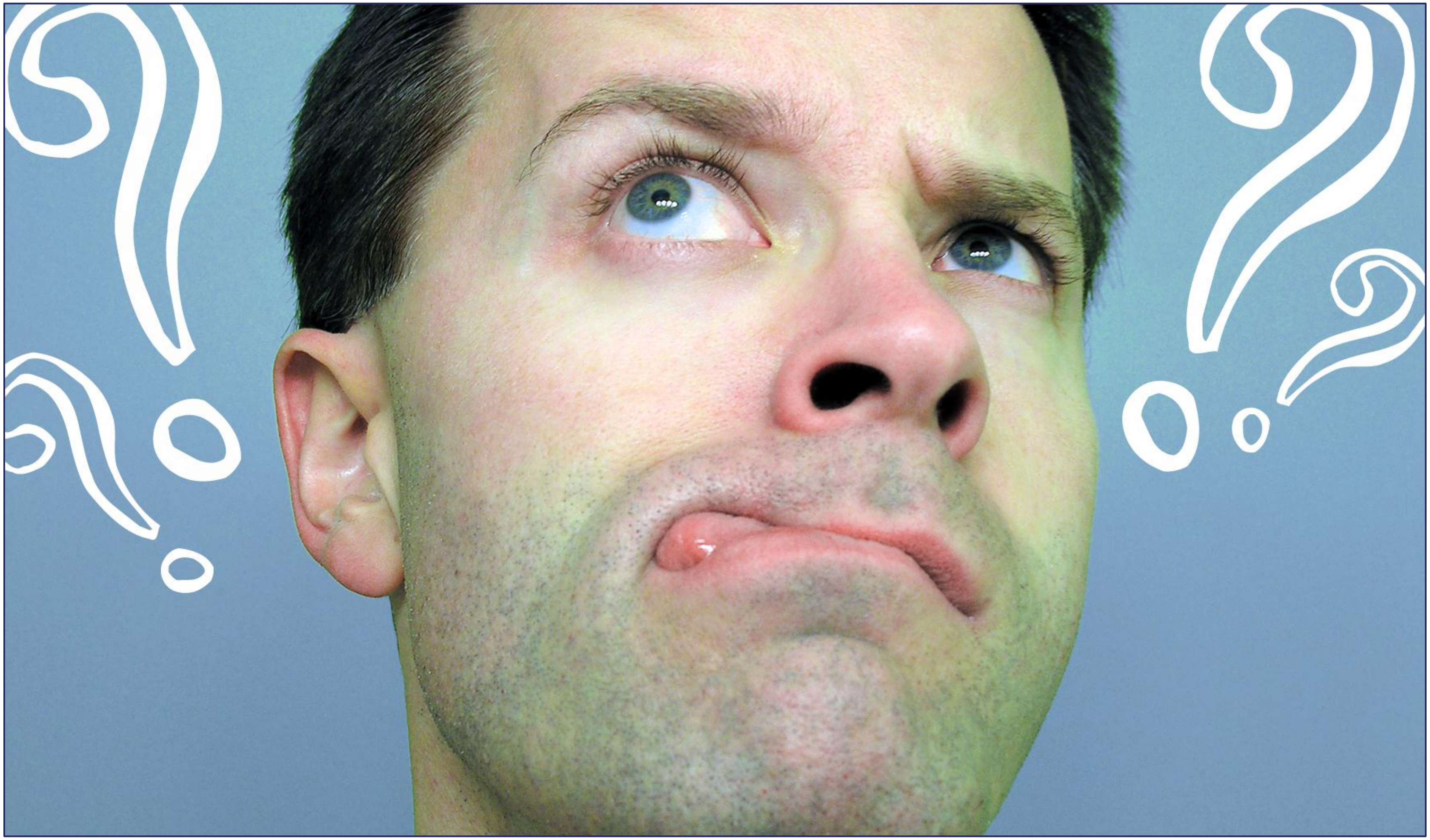
1. **Acceptance Criteria:** Specification by Example
2. **Automated Acceptance Testing:** SpecFlow
3. **Living Documentation:** Relish/Pickles

Relish :: About

- A service where developers can publish living documentation about the software they’re writing.
- Created in Scotland by Matt Wynne who works on Cucumber, the Ruby equivalent of SpecFlow.

Relish :: Demo

Questions?



Find Code For Demos Here:

https://github.com/toddmeinershagen/Demo.SpecByExample.Begin

MedAssets®