

Software Engineering

FEUP-MIEIC-ESOF-2019-20

Ademar Aguiar, Filipe Correia

In a previous episode...
(on user stories)

Three Cs

Card

- Stories are traditionally written on note cards.
- Cards may be annotated with estimates, notes, etc.

Conversation

- Details behind the story come out during conversations with product owner

Confirmation

- Acceptance tests confirm a story was coded correctly



Details as conditions of satisfaction

As a user, I can
cancel a reservation.

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

- Verify that a premium member can cancel the same day without a fee.
- Verify that a non-premium member is charged 10% for a same-day cancellation.
- Verify that an email confirmation is sent.
- Verify that the hotel is notified of any cancellation.



Most importantly...

Don't forget the purpose

The story text we write on cards is less important than the conversations we have.



Behaviour-Driven Development (BDD)

Behaviour-Driven Development

“BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters”

-- Dan North

Behaviour-Driven Development

“It's using examples to talk through how an application behaves... And having conversations about those examples.”

-- Liz Keogh

Behaviour-Driven Development

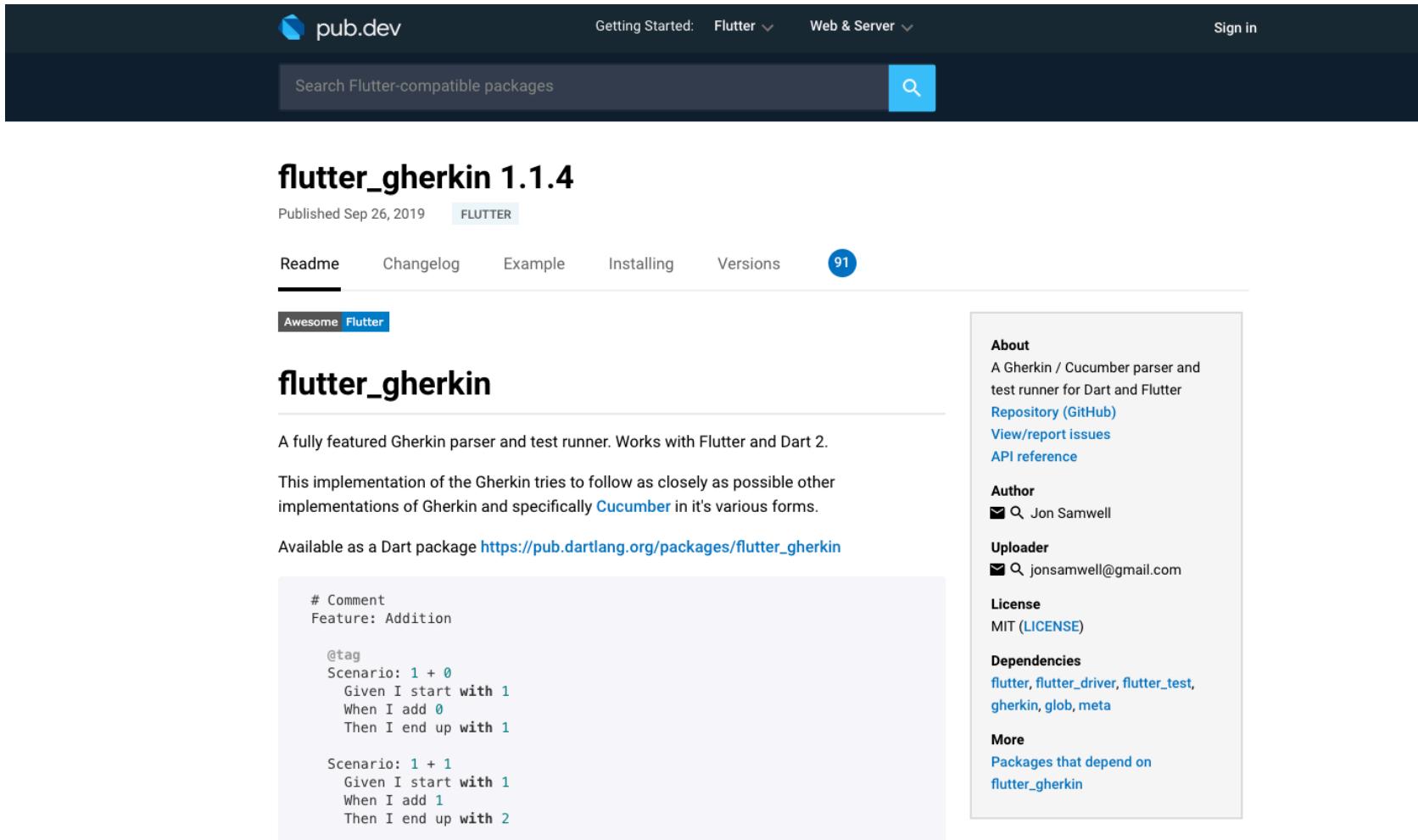
- Focused on finding the places where there's a lack or mismatch of understanding;
- Deliberate discovery: assume that we don't know what we're doing;
- Uses *examples* to promote conversations;
- Requires tools but the main focus are the conversations, not the tools.

Gherkin

- Language used to capture examples of scenarios;
- Designed to be non-technical and human readable;
- Designed to promote *Behavior-Driven Development* practices for the whole team;
- Initially created for Cucumber (BDD tool for Ruby) but now supported for many different languages.

Using BDD tools with Flutter

https://pub.dev/packages/flutter_gherkin



The screenshot shows the flutter_gherkin package page on pub.dev. At the top, there's a navigation bar with the pub.dev logo, 'Getting Started' dropdowns for 'Flutter' and 'Web & Server', and a 'Sign in' button. Below the navigation is a search bar with a magnifying glass icon and a blue 'Search' button.

The main title is 'flutter_gherkin 1.1.4'. It was published on Sep 26, 2019, and is categorized under 'FLUTTER'. Below the title, there are links for 'Readme', 'Changelog', 'Example', 'Installing', 'Versions' (which has a '91' badge), and two buttons: 'Awesome' and 'Flutter'.

The 'Readme' section contains a Gherkin code snippet:

```
# Comment
Feature: Addition

@tag
Scenario: 1 + 0
  Given I start with 1
  When I add 0
  Then I end up with 1

Scenario: 1 + 1
  Given I start with 1
  When I add 1
  Then I end up with 2
```

The right sidebar contains sections for 'About', 'Author', 'Uploader', 'License', 'Dependencies', and 'More'.

About
A Gherkin / Cucumber parser and test runner for Dart and Flutter
[Repository \(GitHub\)](#)
[View/report issues](#)
[API reference](#)

Author
✉️ [Jon Samwell](#)

Uploader
✉️ jonsamwell@gmail.com

License
[MIT \(LICENSE\)](#)

Dependencies
[flutter](#), [flutter_driver](#), [flutter_test](#), [gherkin](#), [glob](#), [meta](#)

More
[Packages that depend on flutter_gherkin](#)

Table of Contents

- [Getting Started](#)
 - [Configuration](#)

Create feature file and test scenario

test_driver/features/counter.feature

Feature: Counter

The counter should be incremented when the button is pressed.

Scenario: Counter increases when the button is pressed

Given I expect the "counter" to be "0"

When I tap the "increment" button 10 times

Then I expect the "counter" to be "10"

Create custom steps for the scenario

- *Given, Then, When, And, But*

test_driver/steps/tap_button_n_times_step.dart

```
import 'package:flutter_driver/flutter_driver.dart';
import 'package:flutter_gherkin/flutter_gherkin.dart';
import 'package:gherkin/gherkin.dart';

class TapButtonNTimesStep extends When2WithWorld<String, int, FlutterWorld> {
  TapButtonNTimesStep()
    : super(StepDefinitionConfiguration()..timeout = Duration(seconds: 10));

  @override
  Future<void> executeStep(String input1, int input2) async {
    final locator = find.byValueKey(input1);
    for (var i = 0; i < input2; i += 1) {
      await FlutterDriverUtils.tap(world.driver, locator, timeout: timeout);
    }
  }

  @override
  RegExp get pattern => RegExp(r"I tap the {string} button {int} times");
```