

Product



Bas Dijkstra & Gáspár Nagy

**I know it's only ReqnRoll (but I like it) -
Making the most of the Automation
phase in BDD**

15.25 – 17.00 – P030



Reqnroll

... or making the most out of the Automation phase in BDD

*Open-source
community*

Independent

Sustainable

reqnroll

*Friend of
Cucumber*

*BDD & test
strategy
showcase*

*I needed
one for
NET 8*

*Ramp-up
time of 2
years*

Whom of you uses
Reqnroll 'on the
job'?

Any specific
questions or things
you would like to see
covered?

What we're going to
do today

How Reqnroll works

Reqnroll is a great
tool for creating
readable tests

Yay or nay? Why?

Reqnroll is a great
tool for creating
readable tests

Yay or **nay**? Why?

Reqnroll is a great
tool for creating
living documentation
from your
specifications

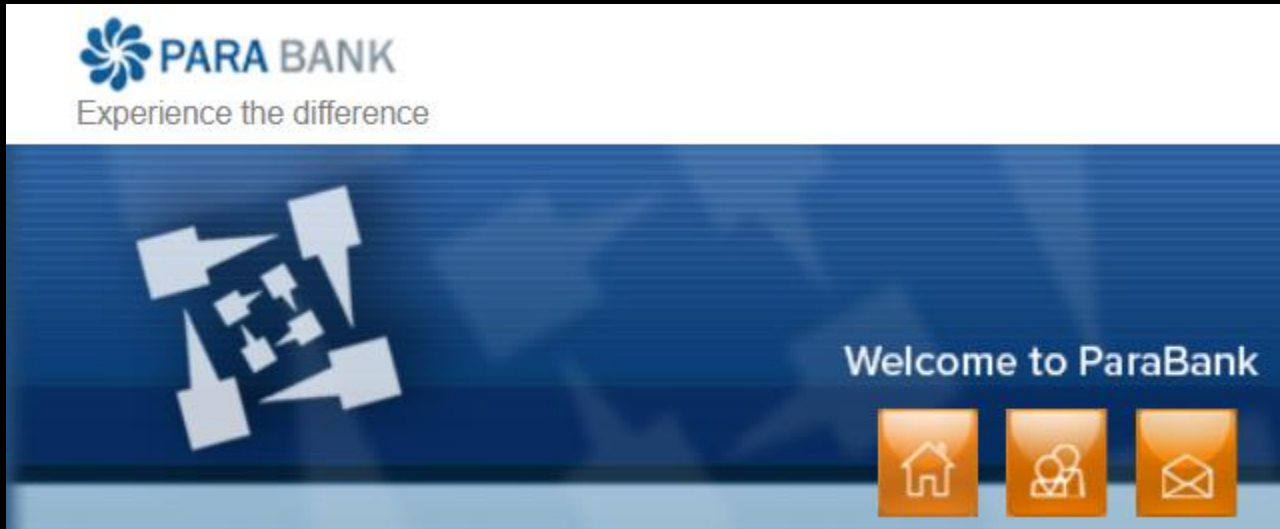
Our application under test

_ParaBank

_The world's least safe
online bank

_<https://parabank.parasoft.com>

_Examples use an instance running locally



Let's have a look at
a typical feature

In groups of 3-4

What do you think of this
feature and the way it is
written?

What would you improve, and why?

(~7 minutes)

Table arguments in Gherkin

Processing table
arguments in Reqnroll

A few more examples
of the power of using
tables



Another example

In groups of 3-4

What do you think of this
feature and step definition
implementation?

What would you improve, and why?

(~5 minutes)

Would this be better?

Why (not) ?

Hooks in Reqnroll

[BeforeTestRun]	Run once before / after each
[AfterTestRun]	test run
[BeforeFeature]	Run once before / after each
[AfterFeature]	feature
[BeforeScenario]	Run once before / after each
[AfterScenario]	scenario (also [Before], [After])
[BeforeScenarioBlock]	Run once before / after each
[AfterScenarioBlock]	scenario block (Given, When, Then)
[BeforeStep]	Run once before / after each
[AfterStep]	step

Let's add
[BeforeScenario] and
[AfterScenario] hooks
to our code

Where do you or where
would you use these
hooks in your
context?



Let's add another
scenario to this
feature file...

What's the problem
here?

How could we address
this?

Tags

Scoping hooks to tags

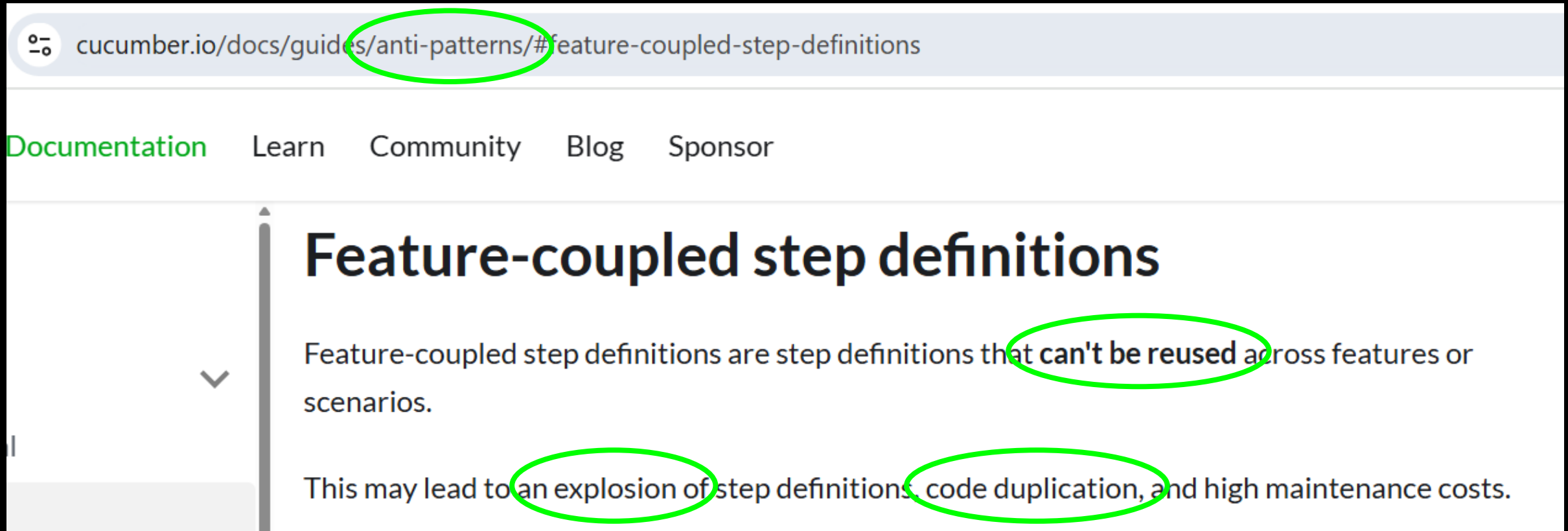
In groups of 3-4

You can also scope step definitions to tags in Reqnroll

Do you think this is a good idea? Why (not)?

(~5 minutes)

This is what the Cucumber folks have to say on this...



The screenshot shows a web browser window with the address bar displaying `cucumber.io/docs/guides/anti-patterns/#feature-coupled-step-definitions`. The page has a navigation bar with links for [Documentation](#), [Learn](#), [Community](#), [Blog](#), and [Sponsor](#). The main heading is

Feature-coupled step definitions

. Below the heading, the text states: "Feature-coupled step definitions are step definitions that can't be reused across features or scenarios." A subsequent paragraph reads: "This may lead to an explosion of step definitions, code duplication, and high maintenance costs." Several terms are circled in red: "anti-patterns" in the URL, "can't be reused" in the definition, and "an explosion of step definitions", "code duplication", and "high maintenance costs" in the consequences paragraph.

cucumber.io/docs/guides/anti-patterns/#feature-coupled-step-definitions

[Documentation](#) [Learn](#) [Community](#) [Blog](#) [Sponsor](#)

Feature-coupled step definitions

Feature-coupled step definitions are step definitions that can't be reused across features or scenarios.

This may lead to an explosion of step definitions, code duplication, and high maintenance costs.



We cannot keep on
adding step
definition methods to
the same class

(or, well, at least we shouldn't...)

Splitting our step
definitions into
multiple classes

What's the
problem here?

Sharing data between
step definitions in
different classes

Dependency injection and ScenarioContext

Let's see how we can
resolve this issue

In groups of 3-4

What are potential pitfalls here?

When would you (not) use a context
and dependency injection?

(~5 minutes)



A final thought...

[https://github.com/basdijkstra
/living-documentation-event](https://github.com/basdijkstra/living-documentation-event)

Contact

Email: bas@ontestautomation.com

Website: <https://www.ontestautomation.com>

LinkedIn: <https://www.linkedin.com/in/basdijkstra>



Thank you for your attention!

Share your insights using the hashtag **#LDE25** and tag **@ICT Improve!**





PROGRAMME

Living Documentation Event

10 April 2025

14.00	Walk in	
14.30	Opening	Auditorium
14.35 - 15.15	Keynote Gáspár Nagy - RAMP up your testing solution: test automation patterns	Auditorium
15.25 - 16.10	Choose between three tracks: Karl van Heijster Testing: A Philosophical Retrospective P083 Jennek Geels The journey is the reward Auditorium	
15.25 - 17.00	Workshop Bas Dijkstra & Gáspár Nagy I know it's only ReqnRoll (but I like it) - Making the most of the Automation phase in BDD (part 1) P030	

16.15 - 17.00

Choose between two tracks:

Rob Albers, Ronald Holthuisen & Martijn van Tienen - BDD, (A)TDD and DevOps practices as a recipe for continuous compliance P083

Rick Easton Tracy - Castles, not Silos Auditorium

Continuation

Workshop **Bas Dijkstra & Gáspár Nagy** - I know it's only ReqnRoll (but I like it) - Making the most of the Automation phase in BDD (part 2) P030

17.05 - 17.50

Choose between three tracks:

Jacob Duizer - From Team Topologies to Behavior-Driven Development: Building Teams That Deliver P083

Pieter Withaar - AI-First BDD, what if we redesign BDD to be AI-first? Auditorium

Machiel van der Bijl - Model Driven Design (MDD): A new approach to Living Documentation P030

17.55 - 18.50

Dinner: Beer and pizza's

18.55 -19.35

Keynote: **Angelo Hulshout** - GenAI and creativity - threat, or tool Auditorium

19.35 -20.15

LDE Community + Panel Discussion Auditorium

20.15 - 21.00

Drinks

