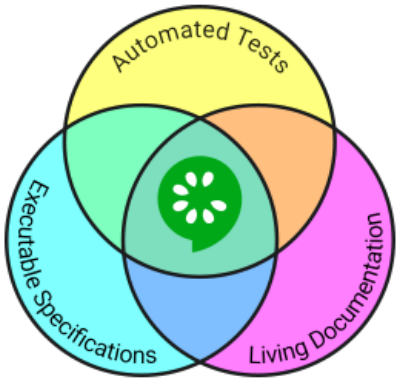
cucumber ltd

BDD & Cucumber Training Outline

About Behaviour-Driven Development



BDD - Behaviour-Driven Development - is an approach to software development that not only focuses on fast and maintainable automated tests, but also on more precise requirements specifications using collaboration techniques for the whole team.

These techniques help non-technical stakeholders communicate better with developers and testers so they can discover misunderstandings *before* the software is written. BDD encourages testers to shift their focus from finding bugs to *preventing* bugs.

With BDD and Cucumber, the whole team collaborates on *executable specifications* in plain text that serve as both *automated tests* and *living documentation* of the implemented system. In other words, a single source of truth for the software, shared and owned by the whole team.

About our training programme: BDD Kickstart

Our training is designed by the Cucumber core team, the world's leading experts in BDD. We've refined this training programme over several years, helping organisations like the BBC, Sky, Amazon, and Aviva to improve and accelerate the way they build software. Over a thousand people worldwide have already benefitted from this training.

Each instructor is a highly qualified expert, with years of practitioner experience to share with the class. We believe in learning by doing, so our classes are highly interactive and practical.

The BDD Kickstart training consists of several one-day modules, focussing on different aspects of BDD. We will tailor the training to your organisation's needs.

	Training *(Theory + Exercises)*	*Coaching* *(Work on your own domain)*
Collaborative analysis (non-technical)	**BDD Fundamentals**	**Discovery Workshops applied**
Test Automation (technical)	**BDD with Cucumber**	**Cucumber applied**
Software Architecture (technical)	**Testable architecture**	**Testing legacy code**

BDD Fundamentals

We involve the whole team (BAs, product owners, domain experts, UX designers, developers, testers, and any other important stakeholders).

On this day we give a thorough introduction to BDD, which is the *process* that Cucumber supports. There is no automation on this day; the focus is on analysis and collaboration techniques.

We'll teach you *Example Mapping* - a powerful technique we have developed to help business, IT and QA break requirements down into concrete examples. This exposes misunderstandings early and will help you write better executable specifications and automated tests later.

Learning outcomes:

- The fundamental principles and practices of BDD
- The importance of Rules and Examples
- Translating examples into Gherkin - Cucumber's format for executable specifications
- Conversation patterns for discovering edge cases
- The importance of a ubiquitous language for problems and solutions
- Using Example Mapping and Discovery Workshops to achieve shared understanding
- Roles and responsibilities on a BDD team

Discovery Workshops applied

Equipped with the skills learned during *BDD Fundamentals*, we invite you to bring 3-5 stories from your own backlog. We'll facilitate a series of Discovery Workshops on each story.

This is an eye-opener for most team members. You'll experience how easy it can be to bring everyone's understanding of requirements to the same level. You'll also learn simple techniques to detect when there are too many unknowns to proceed with development.

Testers will learn how they can get involved before development starts, using their insight to help developers prevent defects.

We'll show you how you can get started with the collaborative aspects of BDD without changing your existing process or organisational structure.

Learning outcomes:

- Facilitating your own discovery workshops
- Impact mapping - deriving scope and stories from business goals
- Hunting for rules and acceptance criteria
- Creating useful examples to illustrate rules
- Capturing questions
- Splitting big stories
- How to write good Gherkin
- Simplify estimation
- Making sure everyone understands what to build
- Striking a balance between specification, testing and documentation
- Measuring the effects of BDD

BDD with Cucumber

This day is a solid introduction to behaviour-driven development (BDD) with Cucumber. You will learn to write *executable specifications* with Gherkin - Cucumber's plain-text format for automated tests. You will also learn how to use Cucumber to drive the development of a system, and see the design of the system gradually emerge. You'll end up with a system that does the right thing, and is easy to test.

Non-technical people are welcome to stick around for this day, but they may find things start to get beyond their comfort zone.

- Installing Cucumber and setting up your development environment
- The TDD / BDD cycle
- Writing your first scenario
- Writing step definitions
- Using parameters in step definitions
- Techniques to keep your test code clean
- Backgrounds, Tags, Tables, Scenario Outlines and other Gherkin tricks
- Domain modelling by Example
- Costs & benefits of working test-first

Cucumber applied

By the end of *BDD with Cucumber*, you should have a thorough understanding of the BDD process and the tools used to write and execute automated tests. Now you're wondering how it's going to work on your own project. Time to find out!

On this day we help you with test automation on your own project. You'll take a real manual test case and translate it to an automated test using Cucumber. If you already have automated tests, we'll examine how they can be improved to be less brittle, faster, and provide more documentation value.

We can show you how you can use Cucumber with Selenium to test through a UI, but we'll also show you more advanced techniques such as testing "underneath the UI" as well as working with databases, microservices, asynchronous and distributed systems.

Our aim is to get your team off and running so you can carry on using Cucumber on a daily basis once the training is complete.

Testable architecture

Automated tests and executable specifications can only work with software that has a testable architecture.

A testable architecture enables fast and reliable tests that are easy to write, execute and maintain. Many organisations have the opposite - slow and unreliable tests that are difficult to write and expensive to maintain.

On this day, we will teach you how to decouple your domain logic from your infrastructure so that you can test at different levels, with maximum confidence and minimum cost.

You will learn advanced techniques such as ports and adapters (hexagonal architecture), contract testing and test pyramid - all essential for a testable architecture.

You will learn how to use these techniques with different kinds of architectures such as microservices, SOA or monolithic systems.

Testing legacy code

This is a hands-on day where we help you retrofit automated tests in your own legacy codebase, using the techniques from the Testable Architecture module.

You choose a critical part of the system that doesn't have automated tests, and we help you test it in the most efficient way.

Coaching

We also offer an option in which, each month, the trainer returns to visit you and spends two days working with the team. The coaching might involve doing code reviews, attending or facilitating discovery workshops, pair-programming with the team, or running shorter training sessions on specific techniques.

During the coaching period, the instructor will support the team via email, phone, or video calls. This provides the team with the day-to-day mentoring and support they need to make sure the ideas from the training succeed in practice.

Lunchtime talks & evangelising

As industry experts, we're often asked to speak to a wider audience of staff members whilst visiting for a training. We're happy to provide lunchtime talks and Q&A discussions as part of our visits on a variety of subjects around agile software development. This can be a helpful part of a broader cultural change.

Feedback from clients

" We had been using Cucumber for about 3 years and had made some great progress, but we also struggled in some areas. We needed some help to ensure we had the right focus going forward. Our main focus was at automated testing, but I didn't feel we were getting the real gains of BDD by focussing on conversations and The Three Amigos approach. It's hard to estimate how much our savings have been, but it probably runs into £100,000's per year. Being trained by the team who developed Cucumber gave the team a real lift. Aslak was a great trainer."

— Steven Granados, Digital Test Manager

" I know we have saved money over the past year as a direct result of our consultation. We are much better at planning, which means we are more efficient developers, and spend a lot less time on missed requirements, bug fixes, and rework. We are churning out new features and products faster than ever, bringing value to our customers sooner and reaping the rewards therein. If I had to put a figure on it, I would say we have saved and/or earned $20k - $40k over the past year as a result of the knowledge we gained from our consultation."

— T.J. Owen, Test Lead

About our instructors

Our training is very hands-on, and an experienced instructor is an essential part of that experience. All of our coaches are world-renowned conference speakers, authors, and creators of the open-source tools your team uses every day.

Below are biographies of some of our trainers. (We can provide resumes on request.)

Aslak Hellesøy

Aslak is the creator of Cucumber. He co-wrote *The Cucumber Book, Behaviour-Driven Development for Developers and Testers* with Matt Wynne.

During his career Aslak has worked with both small and large organisations in several industries such as telecom, trading, banking, retail, insurance, car manufacturing, education and government.

Aslak is a co-founder of Cucumber Ltd, the company behind Cucumber.

Matt Wynne

Matt is one of the world's leading BDD practitioners. A programmer, coach, trainer and popular international speaker, he was invited to join the Cucumber core team in 2009. Together with Aslak Hellesøy, he's co-author of *The Cucumber Book, Behaviour-Driven Development for Developers and Testers* and a co-founder of Cucumber Ltd.

Julien Biezemans

Julien is the creator of Cucumber.js - Cucumber for JavaScript, Node.js, and browsers.

Julien has been coaching teams in BDD since 2009 and been building applications written in Ruby, PHP, Perl, and JavaScript since 2005.

Julien is a co-founder of Cucumber Ltd.

Steve Tooke

Steve is a software developer, trainer and coach. Passionate about improving his craft and helping others improve theirs. He specialises in helping teams produce business value continuously through improving communication and keeping their code maintainable. BDD is at the core of this process.

Steve has been getting paid to build software since 1997. Throughout his career he has worked with renowned organisations like Coach in a Box, the NHS and for a major UK bank.

Steve is a partner of Cucumber Ltd.

Seb Rose

Seb is a software developer, trainer and coach. He specialises in working with teams adopting and refining their agile practices, with a particular focus on delivering software through the use of examples using BDD and TDD.

Seb is the co-author of *The Cucumber for Java Book* together with Matt and Aslak. He has worked with many well-known companies, such as Amazon, IBM, NCR, HBOS, Standard Life and Aegon.

Seb is a partner of Cucumber Ltd.

George Dinwiddie

George helps organizations develop software more effectively. With thirty years of development experience from electronic hardware and embedded firmware to business information technology, he has broad technical experience.

Capable across the broad range from process improvements to engineering skills, George currently has special interests in team-building, skill-building, and effective collaboration techniques.

Paul Rayner

With over 20 years of hands-on software development experience, Paul is a seasoned agile design coach and leadership mentor, helping teams ignite their design skills via DDD and BDD.

His book *Behavior-Driven Development with Cucumber* (co-authored with Richard Lawrence) is forthcoming from Addison-Wesley.

Richard Lawrence

Richard is co-owner of Agile For All. He trains and coaches teams and organizations to become happier and more productive. He draws on a diverse background in software development, engineering, anthropology, and political science.

Richard is one of a handful of Certified Scrum Coaches and is a certified trainer of the accelerated learning method Training from the Back of the Room. His book *Behavior-Driven Development with Cucumber* (co-authored with Paul Rayner) is forthcoming from Addison-Wesley.

Dhaval Dalal

Dhaval is a Software Artisan with over 18 years of experience. He has worked on variety of real-time and non-real-time Web based and client-server architectures on JVM and .NET platforms.

Dhaval has extensive experience with TDD and test automation using over a dozen different tools.

He is interested in architecting applications, establishing environments, transitioning and orienting teams towards an Agile way of working.

Ryan Marsh

Ryan Marsh is an agile coach, programmer and trainer. He loves helping companies get better at making software. At 16 Ryan quit school to program full-time and pursued it with a passion. At 23 he took a break from software development to jump out of airplanes for the US Army. There he learned to train and lead diverse teams in complex environments. These days he coaches and trains software teams on BDD and Agile. Ryan's team building experience and love of programming make him a great match for teams who want to grow.

Thomas Sundberg

Thomas Sundberg is an independent consultant in Stockholm, Sweden. He has a Masters degree in Computer Science from the Royal Institute of Technology, KTH, in Stockholm. He has been working as a Java developer the last ten years. His first experience with test driven development was with JUnit the autumn of 2000. He has also worked as a lecturer at KTH teaching programming courses. There he realized that students who solve programming assignments in pairs normally produce better solutions compared to students working alone. Thomas has a passion for automation and has set up and maintained Continuous Integration systems since 2004 at different companies, including Sony Ericsson and H&M.

Thomas is a speaker at different conferences and developer venues. He was awarded as one of the five top speakers at GeeCON 2012.

What next?

We'd love to talk to you. Please drop us a line at sales@cucumber.io and tell us how you think we can help your team improve. We look forward to hearing from you!