Creating software products is challenging.

First you have to decide what to build; then you have to get your team to build those features without making too many mistakes.

Yet we're human, and mistakes do happen:

· **Misunderstood requirements waste time:** features that don't behave the way you wanted have to be reworked.

· **Defects cause delays:** poor build quality delays the launch of new features and irritates customers.

· **Schedule slips damage your reputation:** when you need to hold back a release for rework, your customers and stakeholders are disappointed.

Mistakes like these waste time and money, damage morale and reputation, and cause your team to lose focus on what's really important.

Many teams using Scrum and other agile methods still suffer these problems. It doesn't have to be this way.

There is a better way to develop software: one which helps your product and technology specialists to clearly understand one another; that breaks work down into small, manageable chunks; that ensures you have automated tests, keeping the product defect-free.

Behaviour-Driven Development rise

This approach is called Behaviour-Driven De the performance of agile teams by taking th

Some of the benefits that BDD provides to s

Focused communication: good conversatio software team. BDD gives you a framework at the right time, avoiding rambling specifica

Shared understanding: by making a delibera collaboratively, everyone on the team becon

Smooth, predictable pace: with the skills to requirement into small pieces of work, your their delivery schedules.

Enhanced build quality: automated testing o BDD, stopping defects before they leave dev

Living documentation: when documentation team produce system documentation that's

All in all, BDD teams are more predictable ar

Start enjoying these benefits by booking a BDD Kickstart course today. Visit cucumber.io/training or email hello