EAIE

Esports as Accelerator for Social Inclusive Education

LESSON

BEHAT STORY QUEST

AUTHOR

SERGIO ELVIRA PEREZ

CODEARTS SOLUTIONS

BEHAT STORY QUEST

Behat Story Quest teaches behaviour-driven development by asking learners to convert a fictional client's natural-language requirements into executable scenarios. The challenge focuses on clarity, shared understanding and traceability between user expectations and system behaviour.

Subject and level

VET backend and QA learners with introductory PHP experience who are learning BDD, acceptance testing and the relationship between user stories and automated checks.

Connection to curriculum

The activity connects requirements analysis, functional testing, technical documentation and client communication. It reinforces Given-When-Then structure, acceptance criteria and the implementation of reusable test steps.

Didactic and pedagogic considerations

The narrative begins with intentionally ambiguous client language so learners must negotiate meaning before automating anything. Peer exchange then reveals how wording affects testability, helping students understand BDD as a communication practice rather than only a tool.

Time estimate

2 hours

BEHAT STORY QUEST

Group size

10-15 students

Overall introduction to exercise

Story teams receive a client briefing for an online ordering service. They translate expectations into Behat feature files, implement step definitions and run the scenarios against a prepared training application.

The central learning outcome is precision: students experience how a vague requirement can produce different implementations, and how concrete examples create a shared contract between clients, testers and developers.

BEHAT STORY QUEST

Exercise as presented to the students

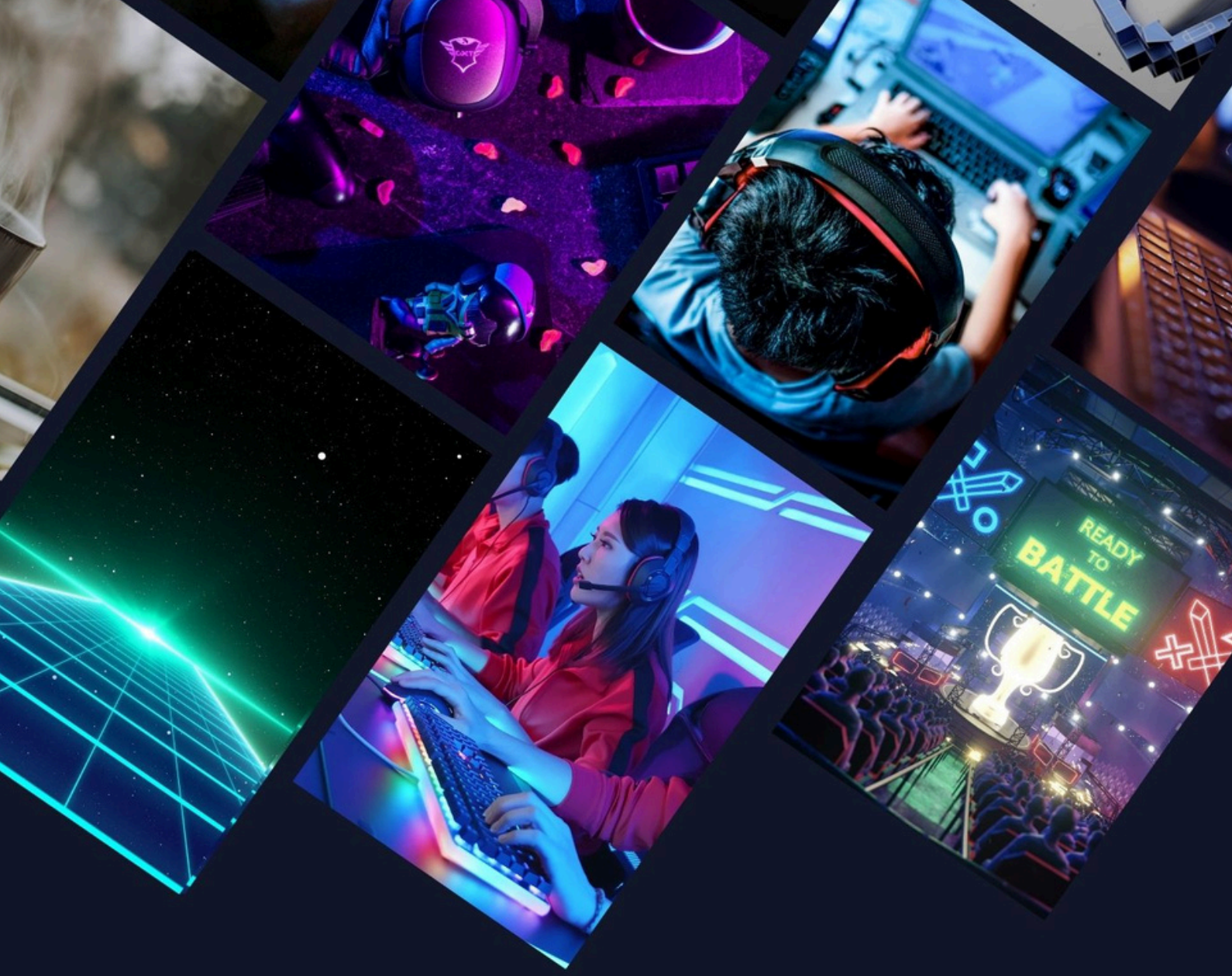
The teacher began in character as the owner of a fictional bakery whose online ordering system was producing confusing customer experiences. Instead of providing technical specifications, the 'client' described needs in everyday language, including deliberately vague statements such as 'customers should be able to change an order easily'.

Students formed story teams and interviewed the teacher for clarification. They transformed the answers into Behat feature files using Feature, Scenario, Given, When and Then. Before writing code, each team exchanged one scenario with another group to check whether the expected behaviour was unambiguous and testable.

Teams then implemented or completed PHP step definitions in a prepared project and ran vendor/bin/behat against the training application. Failed scenarios had to be diagnosed: sometimes the application was wrong, while in other cases the scenario itself was too vague or the step definition did not represent the intended behaviour.

Halfway through, the teacher introduced a new client rule affecting an existing scenario. Learners updated the feature first and then changed the implementation or steps, demonstrating how executable requirements can guide change. Reusable steps earned bonus recognition if they remained clear and not overly generic.

The final discussion compared the original client statements with the completed feature files. Students identified how examples, shared vocabulary and acceptance criteria reduced misunderstanding. The exercise developed automated testing skills while also strengthening listening, writing, negotiation and the ability to bridge business language with technical implementation.



EASIE
Esports as Accelerator for Social Inclusive Education

E-SPORT AS ACCELERATOR FOR SOCIAL INCLUSIVE EDUCATION

WWW.EASIE.GG

Co-funded by
the European Union

FUNDED BY THE EUROPEAN UNION. VIEWS AND OPINIONS EXPRESSED ARE HOWEVER THOSE OF THE AUTHOR(S) ONLY AND DO NOT NECESSARILY REFLECT THOSE OF THE EUROPEAN UNION OR THE EUROPEAN EDUCATION AND CULTURE EXECUTIVE AGENCY (EACEA). NEITHER THE EUROPEAN UNION NOR EACEA CAN BE HELD RESPONSIBLE FOR THEM.