Motivační příklad

Dejme tomu, že pracujeme ve vývojovém týmu, jehož cílem je vytvořit internetový obchod s mimozemskými artefakty. Softwaroví architekti vybrali jedno z obecných řešení s otevřeným zdrojovým kódem, ale nejsou si jistí jeho kvalitou. Bylo rozhodnuto, že kvůli dalšímu procesu vývoje je potřeba mít automatizované testy ověřující základní funkcionalitu daného řešení. Byla vybrána technologie BDD a Selenium, i když s danými technologiemi nemá ve vývojovém týmu nikdo žádnou zkušenost. Vaším cílem je tento nedostatek eliminovat, navrhnout a implementovat testy pro daný internetový obchod.

Cíl projektu

Cílem projektu je:

1. Podrobně se seznámit se zadanou oblastí testované aplikace (zdrojem nechť jsou dokumentace na internetu, selský rozum v případě neexistující dokumentace nebo informace od zadavatele projektu). **Zvolit a navrhnout**, co je cílem testů (které části povrchově do šířky nebo hloubkově do detailu budou testované).
2. Nastudovat chováním řízený vývoj (Behaviour-driven development, BDD).
3. Vypracovat **scénáře BDD** pro testování zadané oblasti testované aplikace. Podrobné pokyny jsou níže.

Na 1. projekt navazuje 2. projekt, ve kterém budete implementovat vámi navržené testy pomocí nástroje Selenium a Behave (Dokumentace). Toto je třeba zohlednit při tvorbě testovacích scénářů BDD. Na hodnocení prvního projektu bude mít největší vliv váš výběr testovaných částí a kvalita vypracování testovacích scénářů.

Testovaná aplikace

Testovanou aplikací je instance eCommerce platforma OpenCart.

Instance a jejich správa

- Každý student má přiřazenou jednu instanci. Seznam instancí se liší TCP portem, na které daná instance poslouchá.
- Každý eCommerce systém má své administrativní rozhraní (login a heslo je "admin").
- Reset instancí (návrat do původního stavu) je možný pomocí příslušného odkazu/tlačítka v seznamu instancí.
- Jednotlivé instance jsou obecně přístupné - buďte prosím slušní a nezasahujte do cizích instancí.

Testovací cíl

S danou aplikací se sezname a vyberte si podčást, pro kterou navrhnete testovací plán. Mějte na mysli, že vámi navržené testovací scénáře budete později implementovat jako automatizované testy. Inspirace pro vybrané podčásti:

- Nákupní proces produktu.
- Správa produktů v administraci.
- Správa uživatelů.
- Správa objednávek.

- ...

Pokyny k vypracování testovacích scénářů

Scénáře BDD vypracujte v souborech s příponou .feature. Syntaxi jazyka, význam příkazů a nejlepší praktiky pro psaní scénářů nastudujte dle dokumentace, doporučené četby a vlastních zdrojů. Minimální počet scénářů není definován, avšak bude mít vliv na hodnocení projektu. Zároveň ve volbě rozsahu scénářů zohledněte, že ve druhém projektu budete implementovat odpovídající testovací případy pomocí automatizovaných testů GUI. Testování GUI bude tématem dalších přednášek.

Soubory se scénáři musí doprovázet soubor README.md, který bude obsahovat stručný popis testovaných vlastností.

- Cílem souboru je seznámit čtenáře s rozsahem testovaných vlastností a vztahem těchto vlastností k jednotlivým souborům popisující scénáře BDD.
- Obsah souboru není definován, kladně hodnocené budou přehledné, stručné a zároveň úplné popisy.
- Soubor README.md bude ve formátu Markdown. Používejte co nejméně formátování, ideálně tedy variantu **strict**. Ověřit validitu textu ve formátu Markdown můžete na konvertoru pandoc. Jazyk textu může být CZ/SK/EN, vše UTF-8.

Způsob odevzdání

Soubory .feature a README.md zabalte do archivu .zip (bez vnitřní adresářové struktury). Archiv odevzdejte prostřednictvím informačního systému.

Četba, informační zdroje

- Repozitář Behave. https://github.com/behave/behave
- Online dokumentace. https://behave.readthedocs.io
- Nejlepší praktiky BDD. https://automationpanda.com/2017/01/30/bdd-101-writing-good-gherkin/
- Antivzory BDD. https://cucumber.io/blog/2016/07/01/cucumber-antipatterns-part-one
- Jednoduchý příklad Python+BDD+Selenium. https://testingbot.com/support/getting-started/behave.html