

Projekt 1 - Testování a dynamická analýza 2021/22

Cíl projektu

Cílem projektu je návrh testovací sady pro regresní testy informačního webu VALU3S:

1. Seznámit se s testovanou aplikací formou exploratorních testů (tj. seznámit se s aplikací “klikáním” v ní a používat selský rozum). V sekci SUT níže je popsána část aplikace, na kterou by testy měly cílet.
2. Zvolit a navrhnout, co je cílem testů (které části povrchově do šířky nebo hloubkově do detailu budou testované).
3. Nastudovat chování řízený vývoj (Behaviour-driven development, BDD).
4. 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 [3]. 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. Není potřeba navrhnout komplikované testy nebo velký počet testů, ale efektivní testy, které s ohledem na svoji velikost pokryjí zajímavě velkou část aplikace.

Testovaná aplikace

CMS (Content Management System) prostředí [Plone](#) přizpůsobený pro účely projektu [VALU3S](#) (dále jen *VALU3S web*). VALU3S web slouží pro mapování verifikačních a validačních metod (tzv. **Methods**) na nástroje implementující dané metody (tzv. **Tools**) a případové studie, které mají demonstrovat užitečnost metod a nástrojů (tzv. **Use Cases**). Každý Use Case popisuje vývoj nějakého produktu. VALU3S web se zaměřuje pouze na část vývoje, konkrétně na verifikaci a validaci (že se správně vyvíjí správný produkt). Správné parametry jsou tvořeny požadavky na produkt (tzv. **Requirements**). Ověření produktu je založeno na tzv. evaluačních scénářích (**Evaluation Scenarios**). Každý scénář se skládá z několika testovacích případů (**Test Cases**). Každý testovací případ by měl pokrývat nějaký požadavek. K provedení testovacího případu je potřeba nějaká verifikační metoda. K samotné verifikaci nám může pomoci nějaký verifikační nástroj (tj. utilita, která implementuje verifikační metodu).

Všechny tyto vazby by měly být implementovány ve VALU3S webu. Uživatelé webu mají několik rolí. Pro projekt ITS si vystačíme se dvěma:

1. producent (administrátor, reviewer, přihlášený uživatel),
2. konzument (veřejnost bez přihlášení).

Producent vytváří obsah webu:

- Use Cases
- Requirements
- Evaluation Scenarios
- Test Cases
- Methods
- Tools

a hlavně vazby mezi nimi. Konzumenti používají VALU3S web jako zdroj informací a inšpiraci pro verifikaci jejich vlastních produktů.

Datové závislosti obsahu

Následující diagramy tříd reprezentují datové závislosti daného obsahu (zdroj: neveřejný report projektu VALU3S), další diagramy jsou dostupné v adresáři [template](#).

Testovací cíle

Cílem projektu je **navrhnout** (nikoliv implementovat) regresní testy, které by měly zaručit, že po aktualizaci CMS systému Plone (např. kvůli opravené bezpečnostní chybě) nevznikne problém při následné aktualizaci obsahu webu. Tedy, že se nezmění funkcionality pro producenty a konzumenty nového obsahu webu.

Mezi aktivity webu, které je třeba testovat, patří převážně:

1. Tvorba nového obsahu (přidání, změna, smazání).

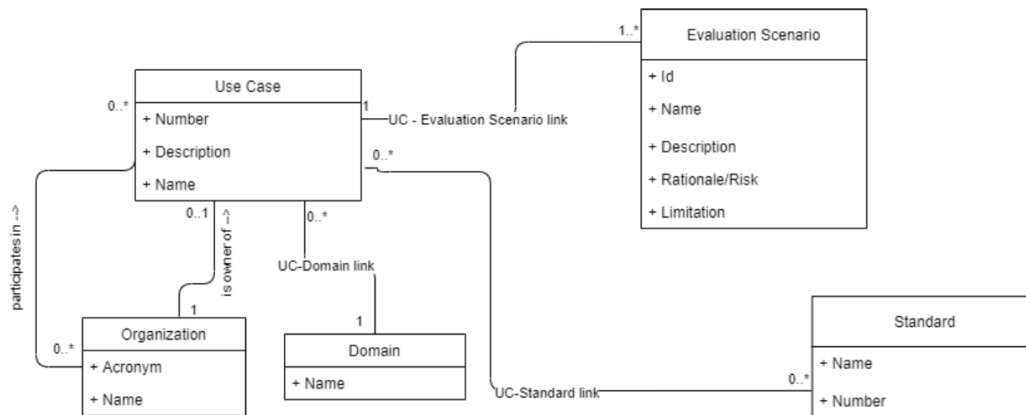


Figure 1: Přímé závislosti Use Case

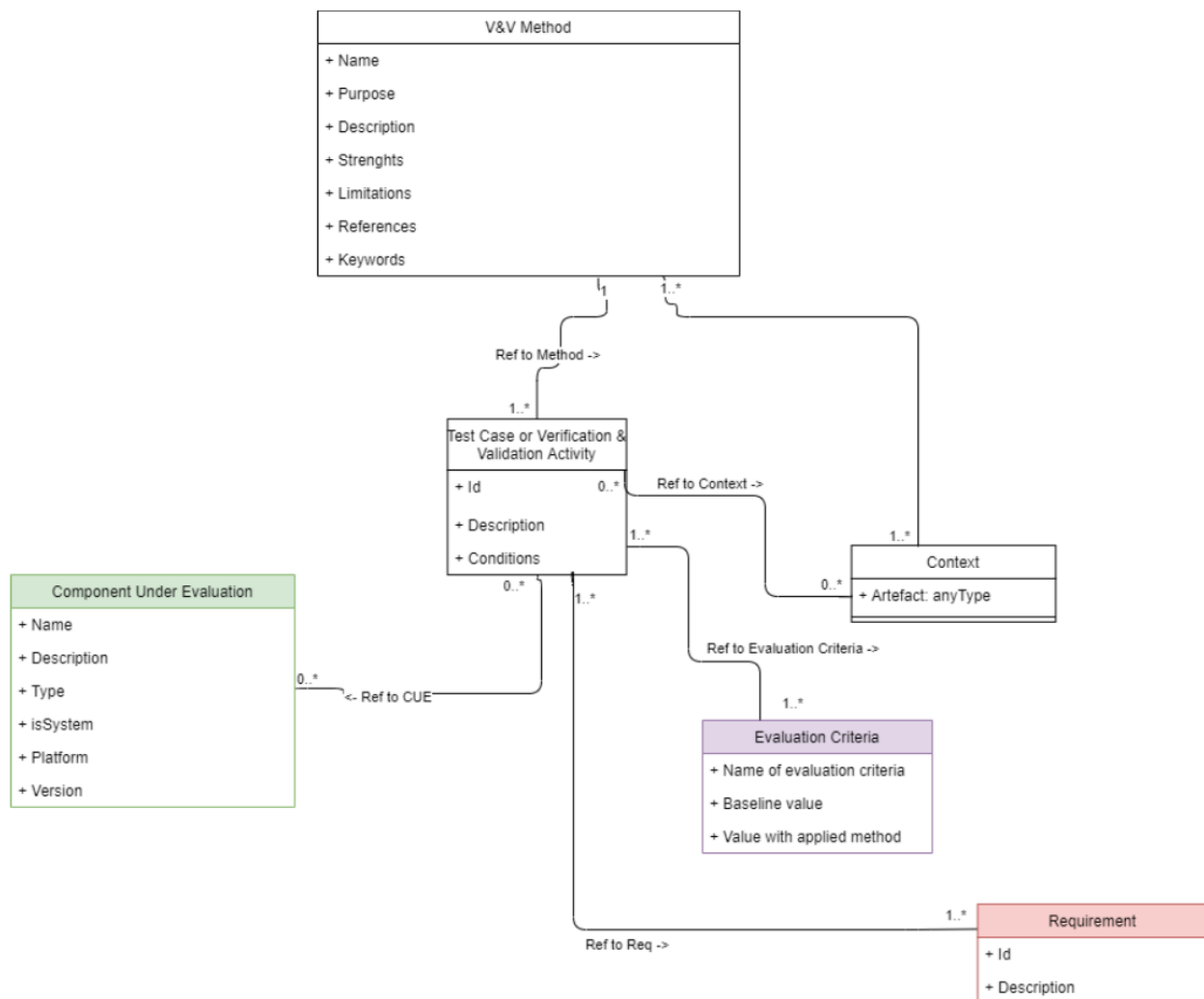


Figure 2: Test Case

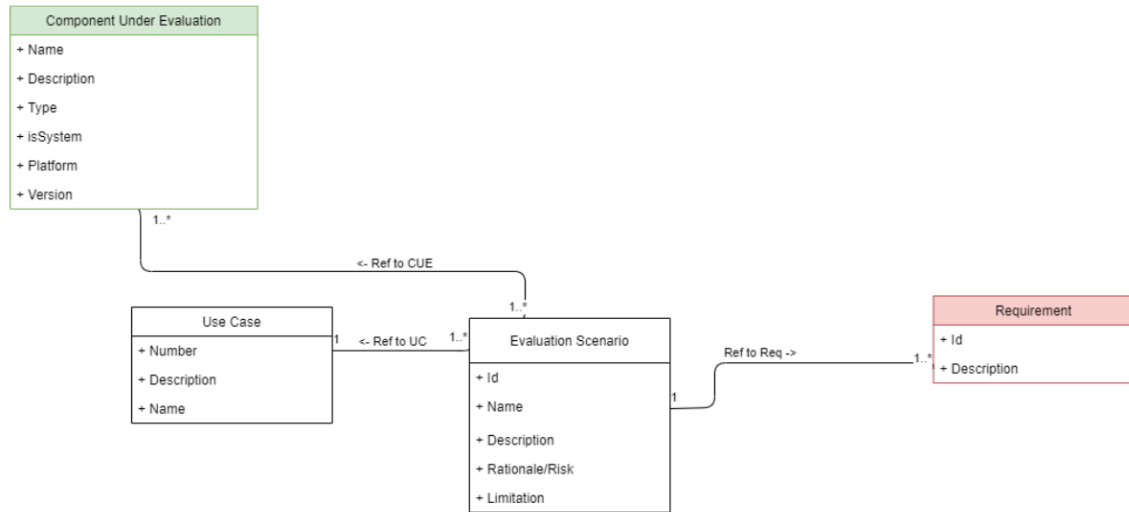


Figure 3: Evaluation Scenario

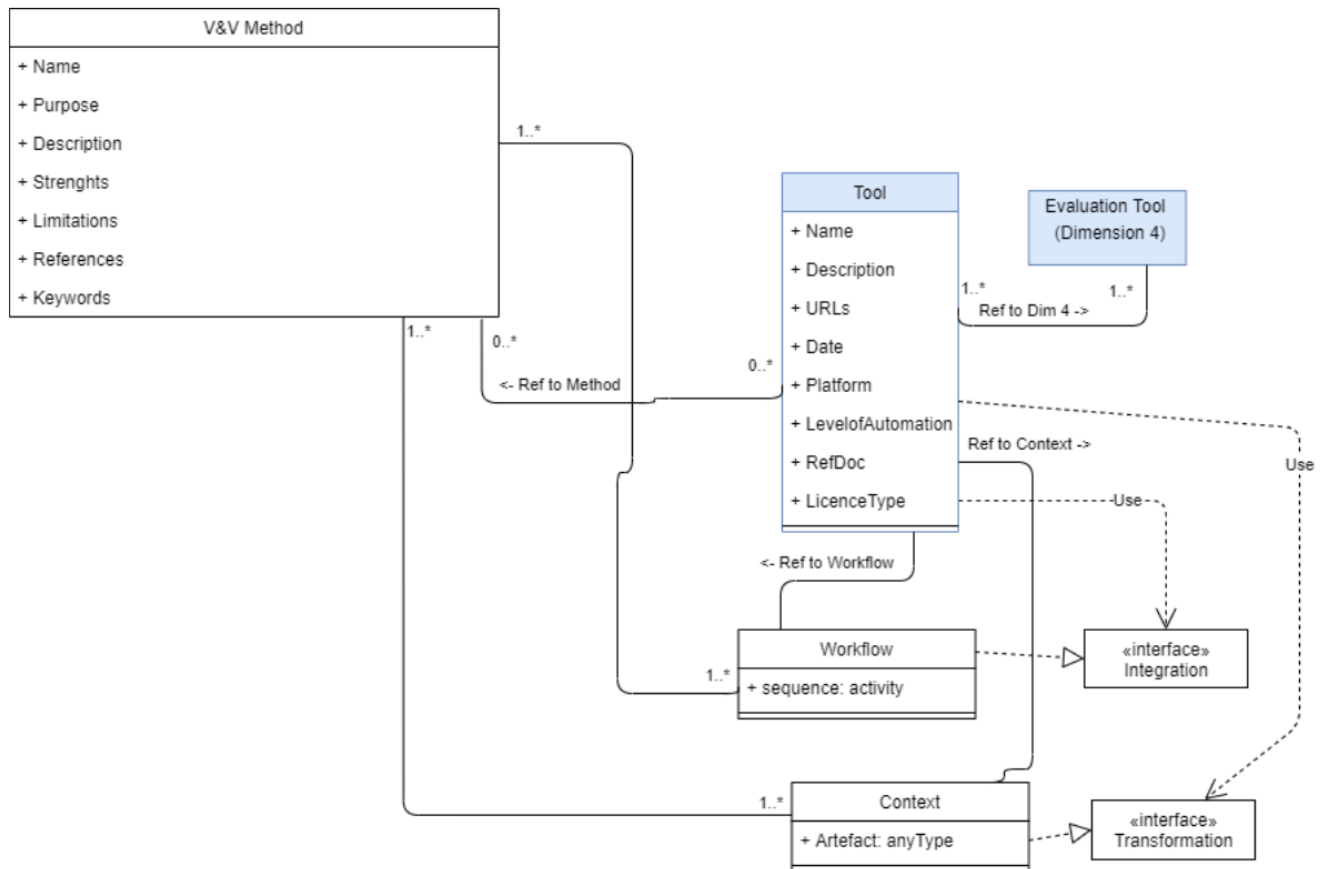


Figure 4: Závislosti Method a Tool

2. Úprava vazby obsahu (přidání, změna, smazání).
3. Změna úrovně zveřejnění obsahu (zveřejnění, schování).

Nezabývejte se uživatelskými účty, konfigurací webu, formátováním včetně obrázků a konkrétními texty obsahu. K tomuto účelu již dostatečně slouží existující testy projektu Plone.

Příkladem jednoduchého testu může být ověření, že:

1. Půjde vytvořit nový Use Case, který bude mít nějaké Requirements, jeden nebo více Evaluation Scenarios, kde každý bude mít alespoň jeden Test Cases pokrývající alespoň jeden Requirement.
2. Use Cases s ostatními daty bude možné zviditelnit (zveřejnit), a veškerý obsah bude dohledatelný nepřihlášeným uživatelem.
3. Půjde vytvořit novou metodu s vazbou na nástroj a konkrétní testovací případ. Všechny tyto části budou dohledatelné konzumenty postupným klikáním z hlavní stránky VALU3S webu.

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 [1], 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 report v souboru `README.md`, který bude obsahovat stručný popis testovaných vlastností **ve formě matice pokrytí**.

- Cílem souboru je seznámit čtenáře s rozsahem testovaných vlastností (dále jen *artefakty*) a vztahem těchto vlastností k jednotlivým testům a souborům popisující scénáře BDD.
- 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.
- Matice pokrytí, které testy pokrývají které testované vlastnosti, bude ve formě tabulky. Řádky tabulky jsou pokryté artefakty (webové stránky aplikace, ovládací prvky, aktivity, ověření vlastností), sloupce jsou číselně označené testy.

Artifact	1	2	3	...
Page XYZ1	x	x		
Page XYZ2	x		x	
Page ...			x	x
Setting mapping OPQR to ABCD	x		x	
Deleting XYZ	x		x	
Doing something ...				x
Checking: relationAToB	x			x
Checking: relationXtoY		x	x	
Checking: A covers B	x	x	x	
Checking ...				x

- Matice pokrytí bude doplňovat matice implementovaných testů, tj. který test je pospán ve kterém souboru s BDD scénářem.

Feature file	1	2	3	...
file1.feature	x	x		
file2.feature			x	x

- Šablonu pro report `README.md` najdete v repozitáři s webovou aplikací:
<https://pajda.fit.vutbr.cz/smrcka/valu3s-its-2022/-/tree/master/templates>

Způsob odevzdání

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

VALU3S web

Testovanou aplikaci jako webovou aplikaci můžete rozjet na svém počítači pomocí PaaS technologie Docker (kontejnerová služba mimo jiné zajišťující také izolaci běžících aplikací). Pro tento účel je pro vás připravený repozitář: <https://pajda.fit.vutbr.cz/smrcka/valu3s-its-2022> (další návod včetně přihlašovacích údajů najdete na této stránce).

1. Nainstalujte si [docker](#) a [docker-compose](#).
2. Stáhněte si repozitář s webovou aplikací (bude potřeba mít nainstalovanou podporu [git lfs](#)):

```
$ cd path/to/its/project
$ git clone https://pajda.fit.vutbr.cz/smrcka/valu3s-its-2022.git
```

3. Vytvořte kontejner s aplikací:

```
$ cd valu3s-its-2022
$ docker-compose build
```

4. Spustte aplikaci:

```
$ docker-compose up -d
```

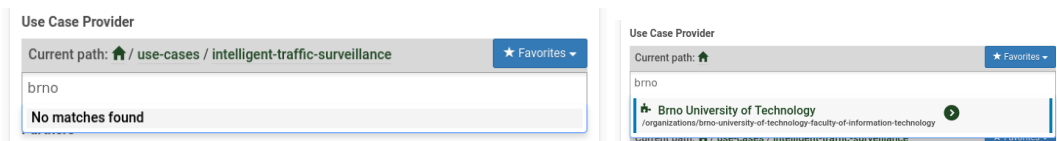
5. Ve webovém prohlížeči je aplikace dostupná na adrese: <http://localhost:8080/repo>

6. Pro ukončení aplikace stačí:

```
$ docker-compose down
```

Odkazy v systému Plone

Webová aplikace by měla být intuitivní. Je zvykem řadit obsah webu do složek (Folders) a případně vytvářet reference na jiné. Při vkládání referencí na jiné části obsahu se využívá vyhledávací ovládací prvek a vyhledávání pomocí několika písmen obsažených v názvu obsahu. Například u reference “Use Case Provider” při editaci Use Case 1 je potřeba vytvořit referenci na obsah typu “Organisation”: Ve vyhledávacím poli stačí zadat “brno” a nabídne se daný obsah, pokud je v dané cestě nalezen (v cestě, kde je umístěn obsah Use Case 1 nic takového není, ale v prohledání od kořene, tj. “domečku” již ano):



Literatura

- [1] Syntaxe Gherkin pro psaní BDD. <https://cucumber.io/docs/gherkin/reference/>
- [2] Behaviour-Driven Development. <https://cucumber.io/docs/bdd/>
- [3] Selenium a behave. <https://www.bddtesting.com/using-the-behave-framework-for-selenium-bdd-testing-a-tutorial/>
- [4] Nejlepší praktiky BDD. <https://automationpanda.com/2017/01/30/bdd-101-writing-good-gherkin/>
- [5] Antivzory BDD. <https://cucumber.io/blog/2016/07/01/cucumber-antipatterns-part-one>