Posudek oponenta diplomové práce

Jméno a příjmení autora posudku:	Lubomír Bulej
Jméno a příjmení autora práce:	Tadeáš Palusga
Název práce	Extensibility Framework for a Generic Benchmarking Environment

Text posudku

EverBEEN je distribuované prostředí pro spouštění úloh, jehož hlavním cílem je usnadnit systematické vyhodnocování výkonnosti během vývoje softwarových projektů. Cílem práce bylo upravit stávající implementaci prostředí EverBEEN tak, aby byl výsledný systém více modulární a výrazně méně svázán s konkrétními knihovnami a technologiemi. Nutnost takové úpravy vychází ze skutečnosti, že EverBEEN je stále poměrně těžké nasadit (a především udržovat v kontextu nových a rychle se rozvíjejících technologií) a to i přes to, že EverBEEN představuje již třetí generaci prostředí pro spouštění úloh a jeho zhruba 22 tisíc řádků kódu je výsledkem práce 4-členného týmu.

Při snaze o větší modularizaci prostředí EverBEEN autor nakonec došel k závěru, že vazby mezi různými částmi (zhruba 40-50 modulů) stávající implementace jsou natolik těsné, že není v silách jednoho člověka v rámci diplomové práce zavést do projektu větší míru abstrakce, která by rozvolnění vazeb umožnila. Jako řešení autor nakonec zvolil cestu reimplementace klíčových komponent a vytvořil nové prostředí, BootBEEN, které sice není funkčně ekvivalentní prostředí EverBEEN, ale mělo by poskytovat modulární a především udržitelnou architekturu, která umožní jednotlivé části prostředí rozvíjet a rozšiřovat podle potřeby a nezávisle na ostatních komponentách.

Práce je implementačního charakteru a její rozsah činí zhruba 10 tisíc řádků kódu. V implementaci prostředí se autor snaží důsledně aplikovat princip *dependency inversion*, kdy jednotlivé části závisí na abstrakcích, nikoliv na konkrétních implementacích. Pro všechny klíčové komponenty prostředí jsou definována abstraktní rozhraní a k nim pak konkrétní implementace. Za správně propojení komponent za běhu je pak zodpovědný framework Spring Boot, který do jednotlivých komponent injektuje příslušné závislosti. Nad rámec zadání pak práce obsahuje také základní webové rozhraní.

Zatímco k implementační části práce v podstatě nemám výhrady, o textové části práce to říci nemohu. Přestože oceňuji, že práce je napsaná anglicky a je jazykově na velmi dobré úrovni, k obsahu mám určité výhrady. V první řadě práce viditelně trpí tím, že příliš spoléhá na to, že čtenář zná kontext, ve kterém prostředí EverBEEN a jeho předchozí verze vznikaly. Přestože v úvodu autor věnuje trochu místa historickému přehledu, z práce v podstatě není možné vyčíst, jaké jsou vlastně požadavky na prostředí, jehož implementaci se práce věnuje. Následuje přehled nedostatků stávající implementace prostředí EverBEEN, které jsou vesměs objektivní a reálné, ale bez konfrontace s požadavky, které projekt EverBEEN utvářely to vypadá, jako když je na projektu EverBEEN téměř všechno špatně. Následuje krátké představení architektury prostředí BootBEEN a po něm popis implementace. Druhou polovinu textové části pak tvoří uživatelská dokumentace.

Obecně bych v této práci očekával rozsáhlejší analýzu věnovanou architektuře prostředí EverBEEN a demonstraci toho, proč nelze prostředí jednoduše modularizovat, resp. zavést abstraktní rozhraní a dependency injection. Zároveň bych očekával opětovnou a podrobnější

analýzu požadavků, které utvářely výslednou podobu předchozích verzí projektu a jejich kritické zhodnocení s ohledem na to, jak se předchozí projekty vyvíjely a kam nakonec dospěly. Vedle rozhodnutí prostředí reimplementovat by výstupem takové analýzy měl být seznam požadavků a abstrakcí redukovaný na absolutní minumum, které je nutné mít v jádře a seznam požadavků, které je vhodnější uspokojit formou rozšíření tohoto minimálního jádra. Z architektury prostředí BootBEEN je vidět, jaké moduly tvoří jádro, ale zdůvodnění chybí a není jasné, jestli je toto minimální jádro pouze výsledkem toho, že autor reimplementaci dělal sám, nebo jestli by byl výsledek stejný i v případě, že by měl opět k dispozici čtyřčlenný tým. Práce v tomto ohledu bohužel nepřispívá k lepšímu pochopení problému a toho, co bylo na předchozích projektech (a nutně také požadavcích) špatně.

Rovněž bych očekával určitou diskuzi ohledně možností modularizace. Autor v závěru práce zmiňuje, že použití frameworku Spring Boot bylo důsledkem neuspokojivých výsledků při použití OSGi a pokusu o implementaci vlastního frameworku pro podporu modulů, ale to je poměrně pozdě a navíc to opět nepřispívá k analytické části práce. Jako finální výtku bych uvedl ještě fakt, že inzerovaná modularita a rozšiřitelnost frameworku BootBEEN vlastně není nikde demonstrována. Není tedy vidět, zda jsou abstrakce a rozhraní, na nichž různé moduly závisí (případně je implementují) vhodně zvolené a koherentní. Zde bych očekával, že autor naimplementuje několik modulů a demonstruje rozšiřitelnost a nezávislost jádra na konkrétních implementacích.

Uvedené výhrady vnímám vesměs jako nenaplněný potenciál především textové části práce—autor je zároveň jedním z původním tvůrců prostředí EverBEEN, takže je pochopitelné, že některé aspekty práce již nepovažuje za tak důležité a soustředí se spíše na technickou stránku věci. Rozhodně však musím konstatovat, že autor bezpochyby prokázal svou schopnost řešit netriviální technický problém, stejně jako schopnost toto řešení zpracovat na solidní technické úrovni.

Doporučení k obhajobě

Z výše uvedených důvodů práci *doporučuji* k obhajobě.

Soutěž studentských prací

Vynikající práce vhodná soutěže studentských prací: **NE**.

V Praze dne 1.6.2017

Podpis: