

# Posudek oponenta bakalářské práce

<b>Autor práce (jméno a příjmení)</b>	Hana KAMARÁDOVÁ
<b>Název práce</b>	Příklady dobré praxe při vývoji aplikací na platformě .NET
<b>Název a typ studijního programu</b>	Systémové inženýrství a informatika / Bakalářský
<b>Fakulta / katedra</b>	Ekonomická fakulta / KMI
<b>Vedoucí práce</b>	Remeš Radim, Mgr.
<b>Oponent</b>	Mgr. Petr Lukeš

## Hodnocení práce

### 1. Význam a náročnost tématu 2.0

**Poznámka:** Best practices je stěžejní téma pro jakýkoliv vývoj software. Právě proto k němu existuje bezpočet literatury. Mnoho postupů je možno odpozorovat z existujících open source projektů. Domnívám se, že existují i náročnější témata.

### 2. Logická struktura práce 1.0

### 3. Naplnění cíle práce 1.5

**Poznámka:** Stanovené cíle práce byly dosaženy. Mezi cíli ovšem mohlo být porovnání metrik po provedení navržených úprav, viz 4) .

### 4. Metodický postup 2.0

**Poznámka:** Chybí zpracování metrik po aplikaci změn.

### 5. Hodnocení teoretického zobecnění a přínosu 1.0

**Poznámka:** Proč jsou ty které postupy vhodné v textu najdeme. Mohla by být užitečná kapitola, která uvádí případy, které objasní kriteria podle kterých metodiku zvolit. Kdy SCRUM, kdy UP/RUP, kdy vodopád atp...

### 6. Praktický přínos práce 1.0

**Poznámka:** Práce může dobře sloužit jako metodická příručka i v soukromých firmách.

### 7. Práce s literaturou 1.0

### 8. Formální stránka 1.0

## Závěr

Hodnocení práce známkou: **výborně**

Doporučuji práci k obhajobě: **ANO**

## Otázky pro diskusi a poznámky

### Kritické poznámky a celkový přínos, celková hodnota práce

Autorka v práci shrnuje výčet možných metodik a doporučení při vývoji software obecně. Teoretická část obsahuje popis postupů včetně rizik, která pomáhají řešit. Kladně lze hodnotit, že jsou v práci metodiky a návrhy z nich vycházející zvažovány i v případech, kdy jejich použití nemusí vést ke zkvalitnění kódu.

Přestože již v titulu práce je uveden konkrétní framework .NET, práce zpracovává téma bez újmy na obecnosti a všechny příklady a doporučení jsou jazykově indiferentní.

Práci by velmi prospělo závěrečné porovnání zvolených metrik před a po aplikaci navrhovaných

změn, což by umožnilo kvantifikovat úpravy. Znalost takového porovnání je pak cenná na rozsáhlejších projektech s větším počtem tříd, kde predikce zvolené změny a její dopad na vývoj v projektu má významný podíl u rozhodování o budoucnosti projektu.

Zajímavé je užití citátů významných osobností. Některé principy obecně dobrého uvažování se prolínají různými obory. Hodnotím kladně tento postřeh.

### **Otázky nebo témata pro diskusi před komisí**

1. Některé metodiky a doporučení mohou být v přímém konfliktu, například princip interface decouplingu a YAGNI, KISS a zavádění eventů jako součást single responsibility atd. Rozvěďte prosím, ve kterých případech je vhodné použít, který princip.

2. Představme si další možné požadavky na změnu, jak by se provedly? Zvládne je stávající návrh?

- více druhů počítačových hráčů, např. jeden hraje vždy náhodně, další si pamatuje 2 tahy dozadu atd.

- je potřeba otočit 3 stejné kartičky

- veškerou logiku a vyhodnocování tahů budeme provádět na serveru a na klientech ponecháme jen otáčení kartiček.

3. V praxi se často stává, že na refactoring je jen omezený rozpočet, vývojáři tak nemohou "best practices" aplikovat na již existující kód. Mohou však toto dělat s novým kódem. K zamyšlení a diskusi, jaké principy a metody v takových případech hrají největší roli?

Datum: 10.05.2016

Podpis oponenta bakalářské práce