

ZAJIŠTĚNÍ JAKOSTI PŘI BUSINESS MODELOVÁNÍ

Robert Pergl

Česká zemědělská univerzita, Provozně ekonomická fakulta, Katedra
informačního inženýrství, pergl@pef.czu.cz

Abstrakt

Zajištění jakosti softwarového produktu je problematika, které začíná být v současné praxi (právem) věnována značná pozornost. Těžištěm pozornosti jsou však především fáze tvorby a testování softwaru. Pro jeho kvalitu je však klíčová i fáze analýzy, na jejíž kvalitě podstatným způsobem závisí kvalita výsledného produktu. Tento příspěvek se zaměřuje právě na otázky související se zajištěním kvality v prvních fázích analýzy (business modelování).

Klíčová slova

tvorba softwaru, analýza, jakost softwarového produktu, business modelování, BORM

1 ÚVOD

V posledních letech jsme svědky prudkého nárůstu využívání výpočetní techniky ve všech oblastech průmyslu a služeb spolu se značným nárůstem složitosti informačních systémů. Tvorba softwarových produktů, a především informačních systémů, se i přes neustálý rozvoj nástrojů a metodik potýká se značnými problémy, které mají za následek nízké procento úspěšnosti softwarových projektů.

Probíhá mnoho výzkumů a projektů směřujících ke zlepšení tohoto stavu. Na jedné straně jsou to různá vylepšení nástrojů a metodik, či vznik metodik nových, a na straně druhé vzniká snaha vhodně definovat a normalizovat jakost softwaru, metriky k jejímu měření a postupy k jejímu dosažení. Tyto dvě snahy by měly být v maximální možné míře provázány a navzájem se podporovat a doplňovat.

2 VZTAH JAKOSTI A METODIKY

Jakost lze definovat jako „soubor podstatných vlastností produktu, které určují míru uspokojení daných (obecně očekávaných) a stanovených potřeb uživatele produktu, v případě užití produktu stanoveným způsobem“ [1].

Současně s trendem zvyšování jakosti softwarových produktů však dochází k nežádoucímu jevu: k odtrženému chápání pojmu jakosti. Pokud má jakost dostát své definici uvedené výše, nelze ji chápat zúženě pouze jako soubor obtěžujících norem a nařízení, které je třeba dodržovat, aby firma získala certifikát, kterým se může honosit. Tento přístup je bohužel dosti rozšířený a vede k určité skepsi o tom, že snahy vedoucí k zajištění jakosti budou znamenat „lepší produkt“. Pokud se totiž nad definicí jakosti hlouběji zamyslíme, můžeme prohlásit, že v podstatě všechny metodiky tvorby softwaru mají za cíl zvýšit jakost výsledného produktu, neboť jedním z hlavních cílů všech metodik je, aby zákazník obdržel program, který naplňuje jeho očekávání. **Jakost je tedy nedílitelnou součástí metodiky.**

Pro každou metodiku tvorby softwaru bychom si měli položit několik otázek:

1. Jakým způsobem zajišťuje metodika jakost výsledného produktu?

2. K dosažení jakosti musí být v metodice pravidla a postupy. Jak jsou tato pravidla a postupy kontrolovatelné a měřitelné?
3. Co jsou charakteristiky jakosti výsledného produktu, jak jsou měřitelné?

Normy týkající se jakosti softwarových produktů se bohužel věnují především pouze poslední otázce, zabývají se tedy již výsledným produktem. Součástí těchto norem (např. současná ISO/IEC 14598) jsou sice části týkající se plánování, řízení a vývoje, jedná se však o obecné postupy, které nejsou vytvořeny pro žádnou konkrétní metodiku.

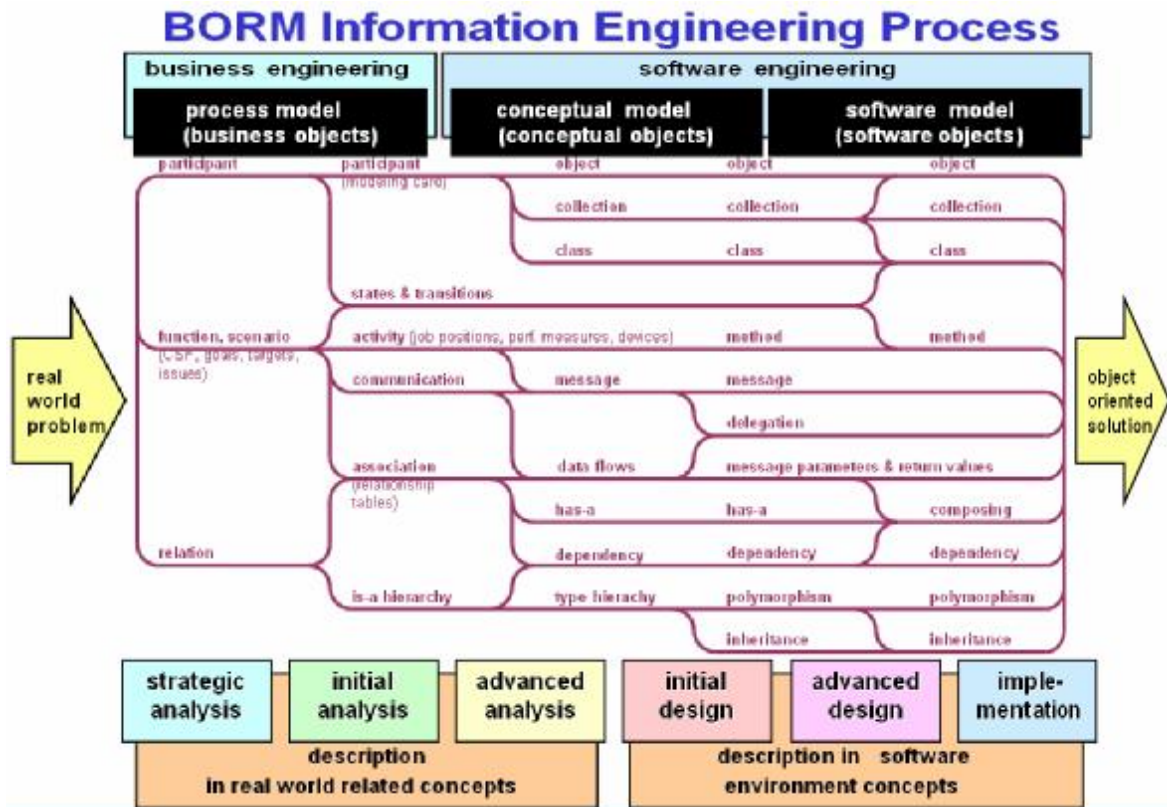
Důsledkem toho může při používání konkrétní metodiky dojít ke dvěma negativním jevům:

1. Omezení možností metodiky v důsledku nutnosti dodržení formálního postupu.
2. Nevyužití speciálních možností, které má sama metodika pro zajištění jakosti.

Z tohoto důvodu je třeba zabývat se zajištěním jakosti nejen na obecné úrovni, ale též studovat možnosti zajištění jakosti konkrétně pro určitou metodiku.

3 BUSINESS MODELOVÁNÍ A METODIKA BORM

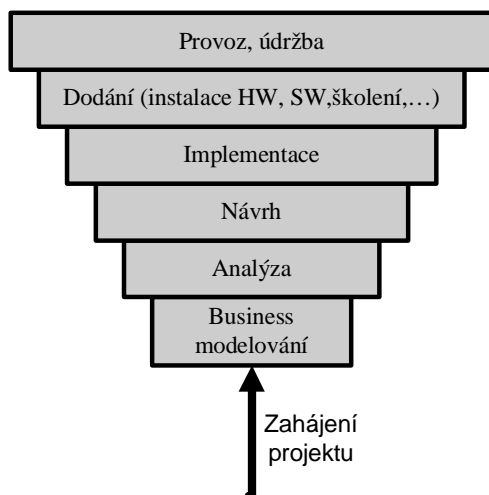
Metodika BORM (Business Objects Relation Modelling) byla vyvíjena postupně od roku 1993 v rámci mezinárodního výzkumného projektu. Od roku 1996 je další vývoj podporován firmou Deloitte&Touche, kde je BORM také prakticky používán. Jeho přínosem je komplexní přístup k úvahám nad informačními systémy. BORM je možné využít nejen ve tvorbě softwaru, ale i k analýze požadavků na projektovaný systém a na modelování business procesů. Právě z procesů a dalších pojmů business modelování se v BORMu postupně vytváří model požadovaného ICT řešení (viz Obrázek 1 a [4]). Detailněji se o metodě BORM lze dočíst v různých publikacích a příspěvcích ([2],[3]), my se zde zaměříme pouze na fázi business modelování (BM).



Obrázek 1 – Fáze tvorby IS metodikou BORM

3.1 Pozice business modelování v zajištění jakosti

V tomto textu budeme používat ještě jeden důležitý pojem a tím je *kvalita*. Kvalita bývá často ztotožňována s jakostí, zde však budeme kvalitu chápat jako dílčí prvek vedoucí k zajištění jakosti produktu. Zjednodušeně můžeme prohlásit, že jakost výsledného produktu je založena na kvalitní práci, kvalitních surovinách, atd. Dá se říci, že kvalita v tomto slova smyslu je vlastně jakost určitého prvku pro dodavatele, kterým se snaží zajistit jakost výsledného produktu.



Obrázek 2 – Vývoj IS z hlediska zajištění jakosti

Proces vzniku softwarového produktu je z hlediska zajištění jakosti obrácenou pyramidou (Obrázek 2). Kvalita kroku následujícího může stavět pouze na kvalitě kroku předchozího a jakost dále rozšiřovat. Pokud je např. v programu funkce špatně naimplementována, kvalitní dodání produktu na tomto nic nezmění. Máme sice mechanismy, které zajišťují zpětné zvýšení kvality, např. testování odhalí chyby implementace, ovšem jakmile dojde k závadě na kvalitě, vždy to znamená náklady navíc, čas navíc, zvýšení stresu a další negativní jevy.

Fáze, která hraje klíčovou roli v zajištění jakosti výsledného produktu je fáze analýzy a především business modelování, a to ze dvou důvodů:

1. Jedná se o první krok (základní kámen pyramidy na obr. Obrázek 1).
2. Při business modelování získáme od uživatele jeho představu o budoucím systému. Při opětovném pohledu na definici jakosti je jasné, že toto je naprosto klíčová záležitost, jelikož jakost je založena na subjektivní potřebě uživatele. Skvělý software, který nepřináší uživateli očekávaný užitek může být sice z různých hledisek výjimečně dobrý, není však jakostní!

4 CHARAKTERISTIKY JAKOSTI

Jakým způsobem tedy zajistit jakost ve fázi business modelování? Různí klienti budou mít různé požadavky na software a informační systémy v závislosti na svých prioritách, strategii a cílech, ke kterému má informační systém sloužit. V případě softwaru obsluhujícího jadernou elektrárnu bude prioritou bezpečnost a bezporuchovost, zatímco uživatelský komfort a minimální doba k zaškolení nebudou tak klíčové. V případě veřejného informačního terminálu tomu bude přesně naopak. Aby bylo možné stanovit úroveň jakosti odděleně podle skutečných potřeb, bylo dohodnuto rozdělit jakost softwarových produktů na šest kategorií nazvaných *charakteristiky jakosti*, které mají další *podcharakteristiky* [1]:

1. Funkčnost (Functionality)
 - i. Funkční přiměřenost (Suitability)
 - ii. Přesnost (Accuracy)
 - iii. Schopnost spolupráce (Interoperability)
 - iv. Bezpečnost (Security)
 - v. Shoda ve funkčnosti (Functionality Compliance)
2. Bezporuchovost (Reliability)
 - i. Zralost (Maturity)
 - ii. Odolnost vůči vadám (Fault Tolerance)
 - iii. Schopnost zotavení (Recoverability)
 - iv. Shoda v bezporuchovosti (Reliability Compliance)
3. Použitelnost (Usability)
 - i. Srozumitelnost (Understandability)
 - ii. Naučitelnost (Learnability)
 - iii. Provozovatelnost (Operability)
 - iv. Atraktivnost (Attractiveness)
 - v. Shoda v použitelnosti (Usability Compliance)
4. Účinnost (Efficiency)
 - i. Časové chování (Time Behaviour)
 - ii. Využití zdrojů (Resource Utilisation)
 - iii. Shoda v účinnosti (Efficiency Compliance)
5. Udržovatelnost (Maintainability)

- i. Analyzovatelnost (Analysability)
 - ii. Měnitelnost (Changeability)
 - iii. Stabilitnost (Stability)
 - iv. Testovatelnost (Testability)
 - v. Shoda v udržovatelnosti (Maintainability Compliance)
6. Přenositelnost (Portability)
- i. Přizpůsobitelnost (Adaptability)
 - ii. Instalovatelnost (Installability)
 - iii. Slučitelnost (Co-existence)
 - iv. Nahradiitelnost (Replaceability)
 - v. Shoda v přenositelnosti (Portability Compliance)

Rozeberme si nyní, které aspekty můžeme pozitivně ovlivnit již při business modelování a jakým způsobem. Při výkladu se budeme odkazovat na metodiku BORM. Z důvodu rozsahu článku zde nebude prostor na vysvětlování základů této metodiky, lze tedy doporučit konzultaci s literaturou ([2]).

4.1 Charakteristiky funkčnosti

Funkčnost je vymezena jako schopnost informačního systému či softwarového produktu obsahovat funkce, které zabezpečují předpokládané nebo stanovené potřeby uživatele při používání systému za stanovených podmínek.

Tato charakteristika tedy zjišťuje, **zda** jsou funkce vůbec zabezpečeny, **nikoliv jak**.

4.1.1 Funkční přiměřenost (Suitability)

Tato charakteristika požaduje, aby pro danou úlohu byla vždy k dispozici funkce, která naplňuje potřeby zákazníka¹. Při business modelování v metodice BORM shromažďujeme požadavky na funkčnost především v *seznamu funkcí a tabulce scénářů*. Seznam funkcí obsahuje přehled funkcí z hlediska podniku z vnějšího pohledu, tabulka scénářů navazuje na seznam funkcí a zaměřuje se na analýzu funkcí vnitřních, tedy do určité míry skrytých pohledu zákazníka firmy. Z hlediska zajištění jakosti je podstatné se soustředit na oba tyto kroky BM.

4.1.2 Přesnost (Accuracy)

Přesnost požaduje přesné výsledky. Ačkoliv se do značné míry jedná o implementační záležitost týkající se např. zaokrouhlovacích chyb a algoritmů, ve fázi BM je třeba si se zákazníkem ujasnit, jaká míra přesnosti je pro konkrétní výstupy požadována.

4.1.3 Schopnost spolupráce (Interoperability)

Jedná se o schopnost spolupráce IS s ostatními IT systémy ve firmě. Součástí BM tedy musí být analýza IT prostředí firmy, na jejímž základě je poté možné se rozhodnout pro konkrétní technologie a přístupy.

¹ *Zákazníkem* neboli *klientem* zde budeme rozumět podnik, instituci, pro kterou vyvíjíme IS. Zákazníka tohoto podniku či instituce budeme potom označovat výrazem *zákazník firmy*.

4.1.4 Bezpečnost (Security)

Na bezpečnost dat je v současnosti kladen vysoký důraz. S klientem je třeba si vyjasnit, jakou míru bezpečnosti potřebuje pro jednotlivé části systému. Systém musí být v maximální možné míře bezpečný, na druhou stranu snaha o implementaci přehnané bezpečnosti může zvýšit náklady, snížit uživatelský komfort, a tím snížit jakost systému.

4.1.5 Shoda ve funkčnosti (Functionality Compliance)

Shoda hodnotí míru přizpůsobení IS lokálním požadavkům, zvyklostem a normám. Zajištění shody je *jedním z klíčových úkolů BM*, jelikož se jedná o jedinou fázi při tvorbě IS, kde je možné toto zajistit (pokud nepočítáme dodatečné korekční mechanismy).

Tato podcharakteristika je doplňována ke všem charakteristikám jakosti a má všude stejný význam, nebudeme ji tedy u dalších charakteristik již rozvádět.

4.2 Charakteristiky bezporuchovosti

Tyto charakteristiky se zaměřují na schopnost systému vyvarovat se poruchám, v případě výskytu poruch na míru zachování výkonu a na obnovení systému po odstranění poruchy. Podobně jako v případě přesnosti 4.1.2 je ve fázi BM třeba si ujasnit potřeby zákazníka. Jedná se především o tzv. *Business critical* části systému, tedy části, které jsou podstatné pro běh firmy. Postupy a odpovědnosti v případě selhání systému též musí být popsány v *Disaster Recovery Plan*, který je součástí firemního *Business Continuity Plan* [5].

4.3 Charakteristiky použitelnosti

Uvedené podcharakteristiky se do značné míry prolínají. Pro nás jsou podstatné dvě skupiny:

4.3.1 Srozumitelnost (Understandability)

charakterizuje snadnost porozumění možnostem systému a práce s ním. Tato podcharakteristika bude především zajímat zákazníka, který kupuje hotový software. Při vývoji informačních systémů na zakázku, na který se zde zaměřujeme, není tato podcharakteristika až tolik relevantní.

Nicméně v této souvislosti je třeba podotknout, že zákazník (ačkoliv si to nemusí připouštět) mnohdy nemá přesnou představu, co by měl od systému požadovat a jaký užitek (tzv. *Business Value*) mu přinese. Součástí práce analytika tedy často bývá spolu se zákazníkem vyjasnit i tyto otázky.

4.3.2 Naučitelnost, Provozovatelnost a Atraktivnost

jsou podcharakteristiky, které se do značné míry prolínají a zabývají se snadností a příjemností obsluhy systému. Zahrnují též úsilí, které je třeba vynaložit na školení personálu.

Zde můžeme konstatovat, že hlavní odpovědnost za tyto aspekty spadá do fáze návrhu IS, implementace a dokumentace. Analytik se však musí zaměřit na shodu v použitelnosti (viz 4.1.5).

4.4 Charakteristiky účinnosti

4.4.1 Časové chování (Time Behaviour)

Zajímá nás především *asymptotické* časové chování [6], tedy jak se budou měnit účinnostní charakteristiky systému (odezva, propustnost, ...) při nárůstu objemu zpracovávaných dat.

Časové chování je ovlivňováno při implementaci použitými technologiemi, výběrem vhodných algoritmů a vnitřních optimalizací systému. Zlepšení jedné charakteristiky bývá za cenu zhoršení charakteristiky jiné. Implementátoři proto potřebují znát co nejpřesnější odhady, kolik dat a jakým způsobem se bude zpracovávat a jaké budou nároky na jednotlivé účinnostní charakteristiky. Je tedy úkolem analytika se zabývat i těmito otázkami. Přijdou na řadu sice až ve druhé fázi, kdy již máme hotové business procesy a datová schémata, jsou však stále do značné míry součástí business modelování.

4.4.2 Využití zdrojů (Resource Utilisation)

Jedná se o analýzu efektivity využití všech ostatních zdrojů, a to jak hardwarových, softwarových, tak i nároků na obsluhu. Platí to samé, co pro předchozí podcharakteristiku.

4.5 Charakteristiky udržitelnosti

Vyjmenované podcharakteristiky specifikují schopnost hotového IS být modifikován. Modifikace zahrnují opravy nedostatků, vylepšování, adaptaci vzhledem ke změnám prostředí, změnám požadavků a změnám funkční specifikace. Hodnotí fázi identifikace (analyzovatelnost), fázi provedení změn (měnitelnost) a možnost otestování systému po změně, tj. udržení jakosti. Stabilita adresuje riziko negativního vlivu na funkčnost systému z důvodů zásahů do něj.

Změny hotových IS a obecně změny zadání při vývoji i po dokončení jsou velkým problémem a způsobují často značný nárůst nákladů a problematickou funkčnost systému. Jsou vyvíjeny metodiky, které se snaží tento stav zlepšit, jmenujme zde především rodinu tzv. *agilních metodik* [7],[8]. Vhodné použití těchto metodik umožňuje značně zvýšit flexibilitu systému.

Společným rysem těchto přístupů je však značné omezení fáze modelování. Je tedy třeba, aby projekt byl svým charakterem vhodný pro tyto metodiky, v případě nutnosti použití klasických metodik [9] je však udržitelnost omezena. V takovém případě je třeba při business analýzách v co největší míře identifikovat směr budoucích změn a systém na ně připravit.

4.6 Charakteristiky přenositelnosti

do značné míry souvisí s udržitelností 4.5. Jedná se nám zde o schopnost IS být přenesen do jiného prostředí, a to z hlediska softwarového i hardwarového. Platí zde – podobně jako v předchozích charakteristikách a podcharakteristikách – nutnost ujasnit si se zákazníkem současné a budoucí potřeby z hlediska předpokládané dynamiky jeho podnikání.

5 ZÁVĚR

V současné době používané i připravované normy se soustřeďují spíše na obecné otázky související s jakostí IS. Výrazné rozšíření možností zajištění jakosti přináší analýza konkrétních metodik z tohoto hlediska.

Charakteristiky jakosti, kterými hodnotíme výsledný softwarový produkt jsou značně ovlivňovány již v průběhu počátečních fází projektu při business modelování. Některé charakteristiky dokonce jsou podstatným způsobem ovlivnitelné pouze v této fázi. To se týká především všech podcharakteristik shody, tj. míru přizpůsobení IS lokálním požadavkům, zvyklostem a normám.

Důraz na zajištění jakosti již ve fázi business modelování znamená více času a nákladů pro tuto fázi, jelikož zvyšuje množství informací, které je třeba od zákazníka získat. Ve svém důsledku však šetří celkové náklady na projekt, protože podstatně snižuje náklady v dalších fázích budování IS a především náklady spojené s úpravami hotového produktu, které jsou nákladné a problematické.

LITERATURA

- [1] Vaníček J.: Měření a hodnocení jakosti informačních systémů. Praha: ČZU PEF, 2004, ISBN 80-213-1206-8
- [2] Merunka V., Polák J., Carda A.: Umění systémového návrhu. Praha: Grada Publishing 2003, ISBN 80-247-0424-2
- [3] Merunka V.: Procesní modelování jako důležitá část fáze získávání požadavků pro analýzu a návrh informačních systémů In: SYSTÉMOVÁ INTEGRACE ČSSI Praha, 2003 ISBN 1210-9479 ISSN 1210-9479
- [4] Pícka M.: Záznam procesu tvorby informačního systému. In: Agrární perspektivy XVIII, sborník prací z mezinárodní vědecké konference, díl II., Praha: PEF ČZU 2004 ISBN 80-213-1190-8
- [5] Hubálek J.: Jak se čelí katastrofám. *Business World*, 2004, č.11, s.16-18.
- [6] Kolář J: Teoretická informatika. Praha: Česká infromatická společnost 2000, ISBN 80-900853-8-5
- [7] Fowler M., Highsmith J.: The Agile Manifesto. <http://www.sdmagazine.com>
- [8] Beck K.: Extrémní programování. Grada 2002, ISBN 80-247-0300-9.
- [9] Pergl R.: Pozice a možnosti Extrémního programování při řízení softwarových projektů. Příspěvek doktorandského semináře PEF ČZU 2004