



Pozvánka

Už **13. dubna od 17:30** budeme opět po roce grilovat zkušeného projektového manažera. Grilování je postaveno na přímé interakci grilovaného s publikem, účastníci mohou klást i „nepříjemné“ otázky, především se ptát na to, co se nepovedlo a jaké poučení z toho vzniklo. Cílem je především odnést si z tohoto setkání jak se daný člověk staví k nepříjemným situacím. Grilovat budeme osobu velice známou, hlavního hybatele firmy PM consulting, **Jana Doležala**. Má zkušenosti především s projekty a produkty z IT, strojírenství, elektrotechniky, akademického i veřejnoprávního sektoru. Projektovému řízení se aktivně věnuje od roku 2001, agilním přístupům pak od roku 2014. Sejdeme se u něho ve firmě PM consulting, na adrese Sokolovská 178/247. Pozor, hledání na ulici Sokolovská není úplně snadné – je to blízko metra Vysočanská.

V květnu budeme diskutovat o práci s požadavky v našich projektech. Požadavky jsou základem projektového řízení. Ať už v klasickém projektovém řízení, v němž jsou známým způsobem ověřovány, dekomponovány a postupně transformovány do úkolů. Nebo v agilním pojetí, kde postupně směřují do back-logu či obdobných struktur. Jak vy pracujete s požadavky? A jak s nimi pracují ostatní? Co se osvědčuje a co ne? Porovnejte své zkušenosti s ostatními účastníky, nepochybně odejdete s novými podněty. Diskuze proběhne 18. května.

Také tady rádi čtete příspěvky doc. Bartošky pod názvem O rizicích vážně nevážně? Věnuje se v nich různým aspektům práce s riziky a objasňuje základy činnosti s nimi. Nejsou ale určeny jen juniorům (jakkoliv na ně cílí primárně), ale vyvolal již i zajímavé mailové diskuze. Poslechněte si tedy tyto informace a využijte možnost diskutovat o nich s autorem. Poslední předprázdninové setkání proběhne právě s ním a na toto téma dne 15.6.

Na všechny akce se lze přihlásit, a to prostřednictvím [webu](#), nebo emailem na adresy Igor.Luhan@mypmi.eu nebo Jaroslav.Barta@mypmi.eu. Vaše případné připomínky k chystanému programu očekáváme tamtéž. Budeme rádi i projevíte-li zájem o konání některé z chystaných akcí u Vás.

Co se stalo

Břežnová diskuze na téma Projektové vzdělávání proběhla opět po delší době v budově Raiffeisen. Krásný výhled a inspirativní diskuze se zajímavými účastníky. Krátký záznam lze jako obvykle najít na našem WEBu.

Renesance kvality ve vývojovém prostředí

Navazujeme na minulý příspěvek Ivana Valíčka o rozdílech v kvalitě u nás a ve Švýcarsku. Rádi otiskujeme jeho delší výzvu na toto téma.

Obsah

- * Pozvánka
- * Co se stalo
- * Renesance kvality ve vývojovém prostředí
- * Doplnky pamětníka



Mantra: "Software (SW) má vždycky chyby". Bohužel, už není zdůrazněno, že chybám se musí předcházet, vyhledat je a opravit.

Během posledních měsíců jsem zaregistroval různé SW chyby. Při vyplnění formuláře pro odběr Hospodářských novin jsem narazil na požadovaný formát pro telefonní číslo. Velice brzy jsem zjistil, že pro zemi Švýcarsko s předčíslem 0041 není akceptováno a tudíž jsem nemohl uzavřít objednávku. Po krátkém zaváhání jsem natipoval 00420, moje švýcarské telefonní číslo a objednávku jsem mohl uzavřít. Kontaktoval jsem redakci, upozornil je na problém, který byl následně komunikován s vývojovým oddělením. Ve Švýcarsku jsem si nechal posílat noviny na přechodnou adresu. Protože přicházely až odpoledne, rozhodl jsem se, že je poslední den zruším, tedy zakázku zkrátím o jeden den a najdu je doma ve schránce. Systém to vyhodnotil jako novou zakázku a nedal mně šanci to jednoduše editovat.

Jindy po odhlášení švýcarské pojistky na konec roku překvapivě dostanete novou průkazku pro další rok. Větší překvapení nastalo, když švýcarská banka zaslala po šesti měsících od úmrtí klienta formuláře, aby je vyplnil. V obou případech bylo na mutaci času více než dost. V posledních dvou případech je dobré se zaměřit na rozhraní a proces. Společný jmenovatel výše uvedeného je problém s kvalitou produktu. Ten pocit mám i u agilní metody, kdy kvalita je v pozadí, sice máme testery, ale nakonec zákazník nám to nahlásí a my jako dodavatel to opravíme.

Pozor, dnes jsou SW produkty připravené a čekají jen na registraci a nastartování. To je jiná situace a dovoluji si tvrdit, že je třeba provést renesanci všech aspektů, které zajistí kvalitu produktu. Zákazník vám už nehlásí problémy, ale podívá se na produkt dalšího konkurenta!

Z hlediska PM, v případě trojimperativu, požadavek na kvalitu SW najdeme jako jeden z projektových cílů, který je třeba definovat s odpovídajícím časem a náklady. Jinými slovy musíte na to mít zajištěné resource a vše integrovat do plánů jak fázového, tak finančního.

Inspiraci najdete v mikroprocesové technice a neškodí se podívat do historie.

Na začátku 80. let jsou na trhu produkty s mikroprocesory, a pak velké počítače. Mikroprocesory se používají pro technické aplikace, jako je řídicí technika různých přístrojů a zařízení, i pro tehdejší novinku PC. V bankovníctví nebo v pojišťovnách se pro zpracování dat používají velké počítače.

Zatímco mikroprocesory používají k programování Assembler nebo PLM (Program Language for Microprocessors), tak v bankovníctví se usídlil programovací jazyk COBOL. V obou případech se programovalo, ale ještě dnes jsou tu velké rozdíly. Aplikace a programování mikroprocesorů (μP) a potažmo mikrokontrolerů (μC) v průmyslu byla náročná disciplína ve srovnání s bankovním



softwarem. Posuďte sami:

$\mu P/\mu C$, jejich použití a charakteristika:

- Produkty řízené $\mu P/\mu C$ se vyráběly ve velkých seriích a nedaly se dělat změny HW ani SW. Teprve u větších zařízení a tím malých serií se použila kombinace s EPROM (Erasable Programmable Read-Only Memory).
- Limitovaná kapacita paměti, integrovaná jako ROM (Read-Only Memory) přímo v μC byla výzvou. To vyžadovalo někdy programovat pomocí Assembleru a optimalizovat program napsaný v PLM.
- Paralelně se vyvíjel HW (hardware). Zde vzhledem k ceně se při vývoji optimalizoval, protože náklady byly rozhodující. V konečné fázi vývoje nebyl HW stabilní a často se hledalo, zda problém je v HW nebo SW.
- S progresivním vývojem technologie se používaly nové, levnější a výkonnější μC s velkou integrací. To bylo provázeno novým vývojovým prostředím včetně nového programovacího jazyka a příslušného kompilátoru. Vývojáři se museli učit stále něco nového.
- SW vyvíjeli většinou čerství absolventi technických škol. Po prvních zkušenostech ve vývoji přecházeli na nová místa. I zde jsme museli udělat různá opatření, jako prodloužení výpovědní lhůty, samozřejmě se zvýšením platu. Cílem bylo bezstresové dokončení úloh a hlavně předání know-how.
- Vývoj SW měl jasnou deadline, a tou byl začátek seriové produkce. Jinak by to vedlo k výpadku výroby a i k nesplnění dodacích termínů u zákazníka, potažmo penále.
- V některých aplikacích, např. při řízení strojů, bylo třeba integrovat v projektu i řízení mechanických dílů. Rozhraní bylo definováno HW, SW i mechanikou. Např. při jednom troubleshootingu u zákazníka jsem musel dva přítomné mechaniky přesvědčit, že problém očekávám u prokluzující spojky, protože pomocí parametrizace jsem byl na konci rozsahu. Takže výměna spojky byla řešením, což se naštěstí potvrdilo.

Zpracování dat na velkých počítačích v bankách nebylo tolik limitováno. Bylo-li třeba rozšířit paměť, tak byl prostor. Pracovalo se se stabilní infrastrukturou. Moje známá z banky si vystačila s COBOLEM celých 40 let a ještě v důchodu byla přizvána do projektu jako expertka na COBOL. Vývojové prostředí bylo stabilní, standardní a prakticky se programátor mohl plně soustředit na svoji roli.

V 80-tých letech bylo novinkou na pracovišti PC s různými aplikacemi. Uživatel byl zásobován nejen základním SW, ale i novými verzemi. Při nefunkčnosti SW



stačilo vypnout a zapnout napájení a dalo se pokračovat. Zkrátka inicializace byla všelékem. V dalších létech šikovným marketingem se uživatelé stali testery a tak je tomu dodnes.

Mikroprocesorové aplikace v průmyslu musely být bez středních a kritických chyb, navíc vše muselo fungovat v reálném čase. Nefunkčnost produktu ze seriové výroby čítající desítky tisíc kusů by byla pro firmu likvidační. Všechno by šlo do šrotu. Ve firmě na telefony, ústředny, komunikační zařízení pro nouzové stavy, atd. to byla výzva.

Aby nenastal worst case, tak se kvalita brala vážně a to nejen u softwaru. V praxi se nasazovaly v té době všechna možná opatření, která vedla k zamezení, vyhledávání a odstranění chyb.

Postupně jsme zavedli tato opatření:

- Výběr vývojových pracovníků
- Interní školení pro mikroprocesovou techniku
- Testovací programy pro nově vyvinutý HW
- Didaktické pomůcky pro programování
- Zavedení FTR (Formální Technické Review) a Peer-Review
- Zavedení grafických metod pro znázornění struktury programu a průběhu dat.
- Fázový plán pro náš vývoj, t.j. SW, HW a mechaniku
- Týmové odsouhlasení základních projektových dokumentů za pomoci FTR
- Spolupráce s externími firmami a specialisty (nové metody, školení, vývoj)
- Aktivní spolupráce při certifikaci na ISO 9001
- Zavedení metody MbO (Management by Objectives) na přidělování a kontrolu úloh během zpracování
- Zavedení testovacích protokolů
- Spolupráce s auditory projektů
- Spolupráce s ostatními vývojovými odděleními ve firmě
- Zavedení know-how redundance u každého vyvinutého SW modulu
- Systematický test závěrečné SW verze; participace velkého interního



testovacího týmu (bez zákazníka)

- Vývoj a aplikace „Watch Dog“; osvěčil se při zablokování vlivem EMC (Electromagnetic Compatibility).
- Testování v klimatických podmínkách, EMC testy, mechanické zátěžové testy, atd.

I dnes se trh s požadavky mění. Jedná-li se o velký projekt se SAP pro velkou firmu, tak vývoj programy nevyvíjí, ale parametrizuje a testuje. To trvá i několik měsíců a i odladění se děje za pomoci zaměstnanců firmy. Registrují se nálezy, změny i nové funkce. V projektu se uplatní plánování ve fázích. U ERP (Enterprise-Resource-Planning) aplikací pro malé firmy se prosazuje koncept, kdy zákazník se doslova obslouží jako v samoobsluze (take it or leave it). Dostane šanci na testování a buď se cítí osloven, nebo to zkusí u jiné nabídky. V každém případě tu už není prostor na nějakou tel. podporu, max. per email.

Když tento business case kalkulujete, tak již dvě školené osoby na telefonu v pracovním týdnu je finančně náročná záležitost, takže se to vynechá. V tomto případě musí být SW již bez chyb, zkrátka připraven pro použití. Tady už není prostor pro diskuze a hledání chyb se zákazníkem, zkrátka vše musí fungovat a navíc oslovit co nejvyšší počet zákazníků. Kvalita se musí zajistit před „Go To Market“ s potenciálním zákazníkem a interně ve firmě.

Je to jako s rohlíky v samoobsluze (vyber -> vezmi -> zaplat). ERP pro velké firmy odpovídá krámu na pečivo přes pult, kdy můžete diskutovat o druhu mouky a potažmo lepku. Zde je kontakt se zákazníky intenzivní, čas se najde a zákazník to zaplatí. Je docela možné, že firma má ERP nabídku pro malé i velké firmy. V obou případech je různý postup a též marketingová strategie.

Ve Švýcarsku se dává šance těmto produktům jen jednou. Jestliže hned v první verzi jsou neakceptovatelné problémy, tak zákazník se rozhodne rychle a věnuje pozornost další firmě. U jedné firmy jsem upozornil na možnosti a potenciál různých reviews. Vzali mě vážně (což tak často nebývá) a počet chyb se skokově zmenšil na polovinu.

Bylo by dobré se podívat na současné interní procesy související s kvalitou produktu, potažmo SW a upravit je tak, abyste byli úspěšní i na švýcarském trhu.

Doplňky pamětníka

Popsanou dobu i popsané technologie jsem na vlastní kůži zažil také a tak mohu doplnit pár vysvětlivek a zajímavý příběh. Nejprve ke zkratkám, se kterými se mladí čtenáři možná už ani nepotkávají (omlouvám sa za zjednodušení výkladu):

- mikroprocesor (μ P) je elektronická součástka - řídicí jednotka každého



počítače, která je obklopena další elektronikou, například pamětí

- mikrocontroller (μC) je kombinace mikroprocesoru, paměti a dalších elektronických komponent tak, že v jedné součástce je skoro celý počítač

COBOL je víceméně první programovací jazyk specializovaný na ekonomické výpočty. V té době fakticky existovaly tři jazyky pro sálové počítače, kromě Cobolu to byl ještě Algol (pro vědeckotechnické výpočty) a Fortran (z něhož se později vyvinul Basic). Žádný z nich nebyl objektově orientovaný (jen procesní), natož aby obsahoval další vymoženosti moderních programovacích jazyků. Ladění chyb v takových programech bylo umění, protože chyba v programu byla schopna potichu přepsat paměť celého počítače. Mimochodem, znalost jazyka Cobol má dnes zlaté dno, protože v některých bankách běží ještě staré corové systémy programované v tomto jazyce a znalci už jsou dávno v penzi.

Velké sálové počítače byly určeny pro dávkové zpracování, tedy program se spustí v dávce a zpracuje vše naráz. Jaký rozdíl proti dnešnímu on-line a interaktivnímu zpracování!

Za předchůzdce dnešních systémů lze tedy pokládat systémy o nichž kolega Valíček píše primárně, tedy systémy pracující v reálném čase. Takové systémy vznikaly v osmdesátých a devadesátých letech i tady. Ještě existovaly tzv. minipočítače, tedy obdoba dnešního PC ve velikosti malé skříně. Kromě dávkových úloh se používaly i pro rozsáhlé real-time systémy podobné aplikacím mikropočítačů (ale samozřejmě s primitivním operačním systémem nebo dokonce bez něho). Právě v této oblasti jsem se v té době pohyboval já. V těchto aplikacích převažovalo programování jak v assembleru, tak v jazyce Fortran. Jaký pokrok, když se na konci komunistické éry tady objevila na vysoké škole první implementace jazyka Pascal, který rozlišoval adresy od proměnných a dokonce umožňoval hlídat rozsahy proměnných!

Pokrok, který přinesl přechod od dávkového zpracování, které už je dnes pro málokoho představitelné, je možno ilustrovat na tomto příkladu z praxe. Konkurent mé tehdejší firmy, dnes už neexistující, v 90. letech vytvářel jako my systémy, které v reálném čase prováděly měření fyzikálních veličin (rozměrů, tlaků, sil, teplot, atd.), jejich vyhodnocení a řízení měřeného objektu (v té době už na základě počítačů PC). V jednu chvíli se rozhodli, že zkusí změnit použití svých systémů a v reálném čase zpracovávat bankovní transakce. Stačilo ten program trochu zjednodušit, protože bankovníctví mělo menší požadavky než průmysl. Tak vznikl systém známé e-banky, která způsobila malý převrat na bankovním trhu.

Redakční rada

Igor Luhan
Jan Bartoška
Stanislav Bílek
Tomáš Szaszi
Jiří Svoboda

Projektový
Underground ©

2022
www.mypmi.eu

Toto číslo připravili Ivan Valíček a Igor Luhan.