



Organization Software Engineering

Minimal Practices

(Minimální nároky na softwarový proces organizace)

Tomáš Smolík
Profinit, s.r.o.
tomas.smolik@profinit.eu
<http://www.profinit.eu>

Historie:

| Verze | Datum | Autor | Popis verze | Jméno souboru |
|-------|------------|-----------------|-----------------------------------------|-------------------------------------|
| 01.00 | 27.05.2008 | Michal Podzimek | Vytvoření z dokumentů v themeChecklists | ProfinitSoftEngMinimalPractices.pdf |
| | | | | |

Obsah:

| | | |
|---|------------------------------------|----|
| 1 | Organizace projektu vývoje sw..... | 2 |
| 2 | Configuration Management..... | 4 |
| 3 | Development Environment..... | 5 |
| 4 | Requirements..... | 6 |
| 5 | Design..... | 8 |
| 6 | Programming..... | 9 |
| 7 | Testing..... | 10 |
| 8 | Documentation..... | 11 |
| 9 | Validation and Verification..... | 12 |



1 Organizace projektu vývoje sw

| checklist | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| Téma | Organizace projektu vývoje sw |
| Minimální nároky: | |
| Je třeba mít zvládnuté následující momenty a jejich vzájemný vztah: | |
| 1. vědět čím se tráví čas při vývoji sw systému | |
| 1.1. elementární ekonomika vývoje/ práce | |
| 1.2. co vlastně se na projektu děje: režie, ... komunikace, testování, ... programování je jen zlomek a vše žere čas a peníze | |
| 2. rozsah: iniciační stanovení rozsahu | |
| 2.1. rozsahem se rozumí závazek dodavatele k tomu, co udělá, jakým způsobem to udělá a za jakých okrajových podmínek to udělá | |
| 2.2. používat checklist na nabídku/ rozsah | |
| 2.3. komentář: dejte mi dobře stanovený rozsah a okrajové podmínky a pohnu ... prostě manévrovací schopnosti, zde jsou dány stanovením rozsahu a jeho podmínky | |
| 2.4. jasné stanovení okrajových podmínek, když není ze strany odběratele to a to tehdy a tehdy tak: | |
| 2.4.1. prodlužuje se termín | |
| 2.4.2. narůstá pracnost protože to váže zdroje | |
| 2.5. nesmí chybět přesné vymezení, co je, a co není v rozsahu projektu, tj. nesmí se mi stát, že se po mně něco chce a já to nemám v odhadu pracnosti a nejsem schopen dokázat, že se to po mně nemá chtít, toto může být i nepřímé, typicky příprava testovacích dat (to po mě chce logika věci a já si je musím dělat v některých situacích, ale mám-li na testování 5 dní celkem a jenom příprava dat mi zabere 5 dní, pak není něco v pořádku) | |
| 3. rozsah: průběžně řízení rozsahu | |
| 3.1. evidence ZR/ úprav | |
| 3.2. evidence chyb | |
| 3.3. jednoznačně komunikovat se zákazníkem, že neplnění požadované součinnosti bude mít tyto dopady (nesplníte toto, posouváme termín o tolik a chceme tolik peněz navíc) | |
| 4. měření | |
| 4.1. měření spotřebovaných zdrojů | |
| 4.2. měření základních věcí: time, size, effort, quality | |
| 5. odhady, stanovení pracnosti, kal. trvání | |
| 5.1.1. taxativní stanovení, co VŠE se musí udělat, slíbený způsob, nepohodlí etc. | |
| 5.1.2. společný odhad vůči této dekompozici | |
| 5.1.3. měření | |
| 5.1.4. příště s vědomím jak to minule dopadlo | |
| 5.2. (platí pro mikro i makro) | |
| 5.3. dostatečnost zdrojů vzhledem k závaznému rozsahu a odhadované pracnosti a termínu -- zapsat úvahy v čas t, toto provádět periodicky v čase t+1 ... a porovnávat odhady/ tvrzení/ smýšlení (nejen v odhadu pracnosti vs. skutečnosti, ale i výčtu toho co se děje) v časech t vs. t+1 ... přivádět lidi k realistickému pohledu na to co se skutečně děje a kolik to trvá | |
| 5.4. pracovat s určitou rezervou, včas si říkat o další lidi | |
| 6. plán na den týden, měsíc, rok -- sdílený s lidmi na projektu, šéfy u nás, šéfy u odběratele (parafráze mít neustále aktuální projektový plán sdílený se zákazníkem) | |
| 7. problémy, rizika | |
| 7.1. evidence problémů (typicky chyby, ale nejen ty) interních | |
| 7.2. evidence problémů (typicky chyby, ale nejen ty) externích | |
| 7.3. pravidelný status report externí/ interní (nemusí být, pokud se na tom shodne TS, PJ a šéf projektu) | |
| 8. jednání/ komunikace | |
| 8.1. jednání se zákazníkem (šéfy, středními šéfy, těmi co pracují, těmi co nepracují, ...) | |
| 8.1.1. komunikace se zákazníkem, jasná, přesná, slušná, nepodlézává, chránící rozsah, čas, zdroje | |
| 8.2. jednání s interními šéfy a obchodníky | |
| 8.3. jednání mezi sebou (povídat si o systému, procesu) | |
| 9. vedoucí projektu musí znát finanční stránku projektu a pravidelně ji komunikovat s obchodníkem (smlouvy, fakturace, atd.) | |
| 10. kontrakt mezi šéfem projektu a PJ: | |
| 10.1. rozsah | |
| 10.2. zdroje | |
| 10.3. kal čas | |
| 10.4. hlášení problémů PJ | |
| 11. znalosti vedoucího projektu odpovídají obsahu článku <i>Poznámky k vedení větších projektů vývoje software</i> | |



(viz níže)

12. *tbd ts subcontrats; review of outsourced; nabídka; historie; stavy projektu a co v nich mít etc. (bude jeden extra checklist)*

Platné nároky z *Nároky na proces vývoje sw pro nové projekty I* (file: Profinit_SSPO_Naroky_ProcesI.doc):

id 7; vedoucí projektu zná (de facto nárok na elementární plán): (i) své závazky: co dodá, kdy to dodá; (ii) své zdroje (komentář: vedoucí projektu sděluje a hájí své závazky a své zdroje všem a před všemi zainteresovanými šéfy na obou stranách (politické ohledy probírat s PJih))

id 8; všechny nalezené chyby v testování na straně dodavatele a u zákazníka jsou evidovány a spravovány (komentář: realizovat bugzilla)

id 9; všechny požadavky vzhledem k danému rozsahu jsou evidovány a spravovány (komentář: realizovat bugzilla)

id 10; evidence vydávaných zdrojů: (i) absolutně; (ii) typy činností (vývoj, opravy chyb ...) (komentář: realizovat Sybis, bugzilla)

id 11; sdělování rizik a věcí k řešení všem zainteresovaným šéfům na obou stranách (komentář: politické ohledy probírat s PJih; parafráze nenechat se dlouhodobě dusit, což vede k šizení důležitých věcí)

id 12; detailní plán: každá osoba vždy přesně ví, co přesně se po ní chce v horizontu týdne, co má jaké priority

Platné nároky z *Elementární nároky na RFP* (file: CSAS_RFPSummary.doc): sekce č. 2, nároky č.: 1, 2, 3, 6, 8, 9



2 Configuration Management

| checklist | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Téma | Configuration Management |
| Minimální nároky: Je potřeba mít zvládnuté následující momenty a jejich vzájemný vztah: | |
| <ol style="list-style-type: none">1. vše co konstituuje sw systém musí být v CVS a/ nebo ošetřeno ekvivalentně disciplinovaným způsobem (e.g. data),2. prostředí (typicky vývoj, akceptace, produkce), ve kterých je systém nainstalován v nějakém release/ verzi a aplikovanými nějakými operativními zásahy,3. release/ verze sw systému (existují, jsou pojmenované),4. operativní zásah do software systému v daném prostředí (existují, jsou identifikované, mají vztah k release/ verzi),5. každý ví vždy, jakým způsobem má co a kdy commitovat,6. výrobu dodávky (způsob jak dodat release, počítat s opravou release/ verze)<ol style="list-style-type: none">6.1. vlastní tvorba (co obsahuje, kde se to vezme vzhledem k CM),6.2. instrukce pro instalaci,6.3. uložení v knihovně projektu, zálohování7. výrobu malé dodávky (způsob, jak dodat operativní zásah do software, počítat s několika prostředími),<ol style="list-style-type: none">7.1. vlastní tvorba (co obsahuje, kde se to vezme vzhledem k CM)7.2. instrukce pro instalaci,7.3. uložení v knihovně projektu, zálohování8. evidenci příčin zásahů do systému (chyby, úpravy, změny) vs. zdrojový kód (toto nemá smysl u vývoje z "0") (pomocí bugzilla a CVS commit protokolů),9. evidenci zásahů do zdrojového kódu vůči release (pomocí bugzilla),10. co nabízí CVS kromě implicitního verzování: tag, branch, | |
| <p>Platné nároky z <i>Nároky na proces vývoje sw pro nové projekty I</i> (file: Profinit_SSPO_Naroky_ProcesI.doc): id 3; definovaný způsob tvorby dodávek systému odběrateli, id 4; všechny pracovní produkty k projektu jsou: (i) v konfiguračním systému (cvs), (ii) v knihovně na prague-nt (je-li vše v CVS nemusí být); struktura adresářů může vycházet ze vzoru id 5; definované konfigurační řízení ke zdrojovému kódu minimálně vzhledem k tomu co je: (i) na strojích jednotlivých lidí (ii) na vývojovém prostředí (iii) dodané u odběratele (iv) diference mezi jednotlivými dodávkami odběrateli.</p> | |



3 Development Environment

| checklist | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| Téma | Development Environment |
| Minimální nároky: | |
| <ol style="list-style-type: none">1. pro daný vyvíjený sw systém existuje definované vývojové prostředí (což nejsou počítače jednotlivých osob)2. vývojové prostředí je popsáno, popis je aktualizován, popis je v CVS3. vývojové prostředí je zálohované, způsob zálohování je popsán a aktualizován a v CVS, zálohování je kontrolované, zálohování je prováděno tak, že při zničení vývojového prostředí je snadné jej zrekonstruovat4. prostředí v možné míře řízené pragmatismem napodobuje cíl (počítače, OSs, instalovanou sw infrastrukturou (e.g. co běží v tomcat ještě nemusí běžet v jaguar), daty v dbs, konfigurací ve files, způsobem používání)5. je definováno jaká verze sw systému je právě ve vývojovém prostředí:<ol style="list-style-type: none">5.1. existuje definovaný postup pro každodenní život: počítače vývojářů --> CSV --> vývojové prostředí5.2. existuje definovaný postup pro nahrání libovolné verze sw do výv. prostředí (zde to nemusí být jednoduché nebo vůbec prakticky možné, pak je s tím potřeba vědomě pracovat)6. maximálně se doporučuje automatické noční CVS update, compile, link, build, deploy, test být v rudimentární podobě7. způsob používání vývojového prostředí není umělým způsobem degradován (např. vyřazení expirací uživatelů etc.)8. pro projekt/ systém existuje soupis veškeré (i.e. i pro vývoj udělatek) používané a předpokládané sw infrastruktury (typicky překladače, os, servery, knihovny) včetně verzí; je zajištěna archivace "instalaček"; dále u používaných knihoven je zajištěna archivace jak "binární" (v našem kontextu typicky jar) podoby nutné pro běh systému tak zdrojových kódů nutné pro příp. úpravy či ladění; toto není v žádném případě ponecháno náhodě | |
| <p>Platné nároky z <i>Nároky na proces vývoje sw pro nové projekty I</i> (file: Profinit_SSPO_Naroky_ProcesI.doc): id 1; definované a zdokumentované vývojové prostředí id 2; definovaný a zdokumentovaný proces záloh</p> | |
| <p>Platné nároky z <i>Elementární nároky na RFP</i> (file: CSAS_RFPSummary.doc): sekce č. 2, nároky: id 14; Vývojové prostředí dodavatele: dodavatel má definované, zálohované vývojové prostředí, ve kterém je v danou chvíli nahrán definovaný zdrojový kód.</p> | |



4 Requirements

checklist

Requirements

Minimální nároky:

Je potřeba mít zvládnuté následující momenty a jejich vzájemný vztah:

1. *Specifikace požadavků* (poskytnuta funkční definice termínu) musí být schopná sloužit minimálně k:
 - 1.1. ustanovení kontraktu pro vývoj (jak z "nuly", tak při rozvoji/ údržbě),
 - 1.2. kvalifikační testování,
 - 1.3. akceptační testování,
 - 1.4. rozhodnutí, co patří, a co nepatří do rozsahu projektu nebo daného Change Request,
 - 1.5. rozhodnutí zda hlášený problém je (i) chyba nebo (ii) požadavek na změnu nebo (iii) šedá zóna mezi tím (pozn. je-li se šedou zónou vědomě zacházeno, pak to je klidně velmi racionální).
2. Zjišťování, analýze, specifikaci a vhodné formě ověření je věnováno přiměřené množství kalendářního času, lidských zdrojů (podstatné je kvalita i kvantita) a příp. politického tlaku na zákazníka, aby se tomu taky jeho strana dostatečně věnovala (aktivní a iterativní přezkoumání všemi stranami zákazníka je samozřejmostí!).
 - 2.1. Pokud není možné požadavkům věnovat přiměřeně času a zdrojů pak je to potenciálně vážný problém, který se musí hlásit vedení na obou stranách. (Pozn. symptomy typu gold-plating a analysis paralysis v kontextu naší organizace zatím prakticky nehrozí.)
 - 2.2. Nesmí se pokračovat rutině dál (tj. čerpat regulérní zdroje pro vývoj, není-li jasné, co se má dělat); zdroje se v takovém případě musí primárně použít na zjištění toho, co se má dělat.
3. Obsah specifikace požadavků splňuje tyto nároky:
 - 3.1. strukturálně vychází ze vzoru srs_template.doc (viz sekce povinné níže),
 - 3.2. je vědomě poměřován následujícím checklist requirements_review_checklist.doc (viz sekce povinné níže).
4. Forma zaznamenání specifikace splňuje tyto minimální nároky:
 - 4.1. Základem specifikace požadavků je [maximálně] strukturovaný text (zde jako opak "viktoriánské novely") vhodně doprovázen: (i) datovým modelem (úroveň abstrakce tj. konceptuální, logický, fyzický záleží na okolnostech), (ii) modelem GUI, (iii) modely, které odpovídají dané domněně a jsou s nimi zkušenosti.
 - 4.2. Poznámky k závaznému výkladu nároků na formu:
 - 4.2.1. povinná četba: *Writing Effective Natural Language Requirements Specifications* (viz sekce povinné níže),
 - 4.2.2. je zakázáno věnovat regulérní úsilí do věcí typu "Pro txt nebudeme mít word, ale text dáme do txt záložek něčeho, co je v PowerDesigneru etc." (pozn. toto lze přeložit jako zákaz investice zdrojů do značného odbočení od čistého osvědčeného principu),
 - 4.2.3. je možné, někdy dokonce vysoce vhodné, použití:
 - 4.2.3.1. UML: ale pro každý zamýšlený diagram je třeba zvážit: je člověku blízký? umí ho? použije ho správně? bude to udržovat? umožní to sdílení mezi všemi zainteresovanými? není např. vhodný právě jen pro ilustraci daného typu momentu (tj. stačí jeden obrázek místo n)?
 - 4.2.3.2. Use-case: ale je třeba s vědomím jejich limitů (hlavně: uvědomit si implicitní [funkční] dekompozici problému, uvědomit si jejich implicitně zúžený záběr; dále udělat specifikaci požadavků based on use case je daleko složitější a nepřirozenější než text-based, která je vhodně doplněna DM, modelem GUI, scénáři příp. dalšími modely odpovídající dané situaci)
 - 4.3. výše uvedené nároky platí od specifikace mající 1 stranu po 1000 stran; u většího celku může postupovat s logickou a fyzickou dekompozicí v souladu se zde uvedenými nároky a inspirovat se lze příklady níže.
5. Vědomé zacházení se vztahem k okrajovým podmínkám projektu a/ nebo systému. Lze ilustrovat následujícími body:
 - 5.1. snaha brát v potaz veškeré okrajové podmínky,
 - 5.2. brát v potaz rozdíl vývoje z "nuly" vs. CR u velkého systému, kde nutně musím cítit "kde co", kde musím brát ohled na architekturu, implementaci, možnosti rozvoje etc.,
 - 5.3. mít maximum, že si neustále snažím "představovat celý systém v hlavě", aby byl pro mne pořád realizovatelný
 - 5.4. nesmí dojít k absurdnímu a doslovnému rozdělení lidí na analytiky, designéry, programátory, testery etc.,
 - 5.5. neustále vztahovat požadavky k architektuře sw systému.
6. Vědomě pracovat se vztahem k architektuře sw systému. Architekturu chápeme nezúženě:
 - 6.1. kontext nadřazených systémů nejen počítačových,
 - 6.2. klasický [makro] strukturální pohled (např. 4 + 1 pohled),
 - 6.3. a nakonec a možná především: všechna prg./ des. rozhodnutí ústící v techniky, které způsobují rutině opakující se prg./ des. momenty.



7. Vztah k rozsahu. Tzn. požadavky jsou stanoveny tak, aby byl jasný závazek k pracnosti (pozn. lze a je třeba využívat jak pozitivních tak negativních vymezení). Pokud není tohoto stavu dosaženo pak:
- 7.1. je to potenciálně vážný problém, který se musí hlásit vedení na obou stranách,
 - 7.2. je třeba investovat zdroje, aby tento stav byl dosažen.

Platné nároky z *Nároky na proces vývoje sw pro nové projekty I* (file: Profinit_SSPO_Naroky_ProcesI.doc):

id 6; požadavky jsou stanoveny tak aby byl jasný závazek k pracnosti, tj. rozsah. (pokud není tento stav, pak to znamená investovat zdroje aby byl tento stav dosažen).



5 Design

checklist

Téma Design

Minimální nároky:

Je potřeba mít zvládnuté následující momenty a jejich vzájemný vztah:

1. *Návrh software* (opět funkční definice) slouží minimálně k:
 - 1.1. realizaci specifikace požadavků (pozn. týká se nejen funkčních nároků, ale potenciálně všeho, typicky výkon),
 - 1.2. možnosti zadat programátorské práce více různým osobám obecně rozprostřeně v čase s tím, že:
 - 1.2.1. stejné momenty jsou řešeny vždy shodně
 - 1.2.2. a hlavně, že to, co se rozmyslelo, navrhlo, příp. už vyzkoušelo je systematicky ctěno,
 - 1.3. dosažení opravitelnosti, udržitelnosti a rozvíjitelnosti,
 - 1.4. dosažení obecně uznávaných technických kvalit návrhu sw systému, což jsou:
 - 1.4.1. [rekursivní/ hierarchická] dekompozice do modulů,
 - 1.4.2. dobře volené abstrakce a rozhraní,
 - 1.4.3. zapouzdření (information hiding),
 - 1.4.4. dobře vyvážený vztah mezi datovými strukturami a algoritmy (pozn. tento se u RMD vtělil v učení o normálních formách).
2. Návrhu je věnováno přiměřené množství kal. času a zdrojů (kvalita i kvantita):
 - 2.1. Pokud není možné návrhu věnovat přiměřené množství času a zdrojů, pak je to vážný problém, který se musí hlásit vedení na obou stranách.
 - 2.2. Věci, které nejsou rutině zvládnuty, a je důvod se domnívat, že by mohly působit problémy je nutno nejdříve vyzkoušet, než se na nich začne systematicky stavět.
3. DBS design je činěn při znalosti učení o tzv. normálních formách (tj. o redundanci v datech) a při nárocích na algoritmy (typicky jde o rychlost, množství zpracovávaných dat, granularita transakcí etc.), které budou daty manipulovat. Výsledkem je návrh, který je buď splňuje min. 3NF nebo vědomě zavádí redundance (např. kvůli výkonu).
4. Design v oo/ procedurálním prostředí (e.g. Java, tsql) je činěn při znalosti zásad (tak jak to pro dané prostředí lze, např. srov. java vs. tsql, c++ vs. fortran etc.): (i) dekompozice systému do modulů, (ii) volby abstrakcí, (iii) zapouzdření.
5. Návrh architektury v žádném případě neignoruje:
 - 5.1. specifikace požadavků (míněno skutečně všechny typy: agenda (fční požadavky), okolní systémy, výkon, uživatelská rozhraní, udržitelnost, rozvíjitelnost, sw infrastruktura, organizace práce etc),
 - 5.2. publikované zkušenosti na dané téma,
 - 5.3. zkušenosti v dané organizaci.
6. Návrh architektury (resp. kritická rozhodnutí na úrovni architektury) musí být přezkoumán.
7. Obsah zaznamenání návrhu a architektury je takový aby:
 - 7.1. bylo možno zadat práci programátorům,
 - 7.2. bylo možno systém udržovat a rozvíjet,
 - 7.3. bylo dosaženo, že stejné momenty jsou programovány/ navrhovány stejně,
 - 7.4. bylo sděleno to, co architekt/ návrhář nakonec rozhodl a předepsal.
8. Poznámky k formě. Pro kontext daného projektu je nutno zvážit:
 - 8.1. popis prg./ design technik pro opakující se momenty,
 - 8.2. při nutnosti dokumentovat architekturu systému: 4+1 view,
 - 8.3. při nutnosti dokumentovat návrh systému: construx template.

Platné nároky z *Elementární nároky na RFP* (file: CSAS_RFPSummary.doc): sekce č. 2 a 3:

id 2.13; Neignorovat elementární programátorské techniky jako jsou: používání podprogramů, modularita, zapouzdření.

id 3.3; Design: dokumentace design/ prg. technik, které používá dodavatel (prim. slouží dodavateli).



6 Programming

checklist

Téma Programming

Minimální nároky:

Je třeba mít zvládnuté následující momenty a jejich vzájemný vztah:

1. *Programy* (lze též zdrojový kód) mají minimálně tyto charakteristiky:
 - 1.1. ctí design (ve všech jeho formách a podobách (prg./ des. techniky, zadání dekompozice etc.)),
 - 1.2. jsou pochopitelné, čitelné, odladitelné, testovatelné, opravitelné, udržovatelné i někým jiným než autorem i někdy jindy než do 10 dnů,
 - 1.3. dosahují obecně uznávané technické kvality pro zdrojový kód, minimalisticky řečeno jsou:
 - 1.3.1. přehledné a jednotné
 - 1.3.2. strukturované
 - 1.3.3. komentované,
 - 1.4. nevymýšlí se kolo, neobjevují se známé slepé uličky.
2. Obecné nároky na programování v malém:
 - 2.1. jednoduchý a jednotný styl,
 - 2.2. komentáře,
 - 2.3. dodržovat zásady strukturovaného programování (sekvence, selekce, cyklus, plus správně používat moderní strukturovaná gotos, uměřená velikost modulů, uměřená složitost modulů),
 - 2.4. rutinní používání rutin (ať už v daném imperativním prg. prostředí je implementován jakkoli, e.g. procedura, funkce, program, subroutine, method ...),
 - 2.5. znát a ctít odborné principy realizované konstrukcemi prg. jazyka anebo základních knihoven pro:
 - 2.5.1. klasické ADT (seznamy, asociativní paměť etc.)
 - 2.5.2. paralelismus
 - 2.5.3. distribuovanost
 - 2.5.4. i/o (fs, síť),
 - 2.6. znát a ctít (tak, jak má smysl v daném programátorském prostředí):
 - 2.6.1. vlastní prg. jazyk (e.g. java),
 - 2.6.2. jeho kvalitní používání (viz techniky v e.g. Effective Java),
 - 2.6.3. rutinní obraty mikro designu (e.g. The Prototype Pattern),
 - 2.6.4. knihovny (e.g. java.sql),
 - 2.6.5. frameworks (e.g. spring)
 - 2.6.6. a run-time prostředí (e.g. jvm + web container).
 - 2.7. znát a ctít asymptotickou složitost programovaných problémů; pokud např. složitost problému roste jako $n \cdot \log(n)$, kde n je velikost vstupu, a můj program má n^2 v situaci, kdy spotřeba zdrojů (čas, paměť) hraje roli (tzn. n je velké), pak je to zmetek; pouze v situaci, kdy znám okrajové podmínky a vím, že n bude vždy malé, si mohu dovolit "strategii" quick&dirty,
 - 2.8. rámcově znát časové charakteristiky programátorského prostředí (paměť, fs, disk, síť, dbs, process-to-process etc.),
 - 2.9. rutinní udržování ponětí o možnostech knihoven a frameworks pro dané programátorské prostředí.
 - 2.10. má-li někdo problém ve výše uvedeném, pak to hlásí vedoucímu projektu (typicky aby se to buď doučil anebo aby se poradil).
3. Konkrétní nároky:
 - 3.1. viz povinné čtení níže,
 - 3.2. viz povinné checklists níže.
4. Ad java programming:
 - 4.1. Sun Java Code Conventions,
 - 4.2. javadoc,
 - 4.3. elementární prg. techniky:
http://cvs/viewcvs/viewcvs.cgi/SOFTENG/PAL/OSSP/naroky/goodies/Prg/Profinit_JavaPrgTechniques.doc.
5. Ad dbs programming:
 - 5.1. elementární prg. techniky :
http://cvs/viewcvs/viewcvs.cgi/SOFTENG/PAL/OSSP/naroky/goodies/Prg/Profinit_DbsPrgTechniques.doc.
6. ad ? programming -- *tbd* život

Platné nároky z *Elementární nároky na RFP* (file: CSAS_RFPSummary.doc): sekce č. 2 a 3:
id 2.13; Neignorovat elementární programátorské techniky jako jsou: používání podprogramů, modularita, zapouzdření.
id 3.4; Zdrojový kód: dobře strukturovaný, komentovaný.



7 Testing

checklist

Téma Testing

Minimální nároky:

Je potřeba mít zvládnuté následující momenty a jejich vzájemný vztah:

1. Testování sw systému (opět funkční definice termínu) slouží minimálně k:
 - 1.1. strana odběratele nenachází při akceptačním testování neadekvátně mnoho chyb,
 - 1.2. strana odběratele nenachází v produkci neadekvátně mnoho chyb,
 - 1.3. strana dodavatele minimalizuje nároky na jednorázové kvalifikační testování před dodávkou a problémy z toho plynoucí (vede k nároku na průběžnost testování),
 - 1.4. strana dodavatele přiměřeným způsobem ověří, že systém funguje podle specifikace,
 - 1.5. strana dodavatele přiměřeným způsobem ověří, že systém je použitelný v reálných podmínkách (e.g. výkon),
 - 1.6. strana dodavatele nedodá zdrojový program, který nebyl alespoň nějak spuštěn.
2. Testování je věnováno přiměřené množství kal. času a zdrojů (kvalita i kvantita).
 - 2.1. Pokud není možné testování věnovat přiměřeně času a zdrojů, pak je toto vážný problém, který se musí hlásit vedení na obou stranách.
3. Testování je vhodně rozloženo v čase.
4. Do kvalifikačního testování vstupuje systém v odpovídajícím stavu, tzn. je nepřijatelná situace, kdy nekvalifikovaný tester během 1 minuty na něco "klikne" a kvalifikovaný vývojář má na hodiny co opravovat.
5. Nalezené problémy jsou zaznamenány do systému bugzilla (odběratelem i dodavatelem, co není standardně zaevidováno neexistuje):
 - 5.1. lze je reprodukovat,
 - 5.2. je zaznamenávána veškerá geneze řešení problému (nejen programováním),
 - 5.3. věci (míněno problémy), které by měla vědět strana odběratele jsou zapsány do externí bugzilla,
 - 5.4. korektně hlášené problémy nalezené v akceptaci a produkci po opravě konstituují obsah příští dodávky,
 - 5.5. jsou vhodně kdykoli vyhledatelné.
6. Testuje se na základě rozmyšlení, co a jak se při daných časových možnostech testuje (lze libovolně minimalisticky).
7. Je prováděn záznam o testování (lze libovolně minimalisticky).
8. Oprava chyby je retestována.
9. Pro projekt/ systém existuje osoba, která koordinuje veškeré aktivity spojené s testováním.
10. Výše uvedené platí jak u vývoje z "0", u vývoje po velkých dodávkách, tak u údržby.
11. Strana dodavatele usiluje o standardní akceptační testování na straně odběratele (existence, kal. trvání, zdroje). Není-li to možné, pak se tato situace se stranou odběratele výslovně probere a důsledky zohlední (např. navýšení zdrojů a kal. času pro testování na straně dodavatele nebo redukce záruky etc.)

Platné nároky z *Nároky na proces vývoje sw pro nové projekty I* (file: Profinit_SSPO_Naroky_ProcesI.doc):
id 13; vyhrazený čas na testování před dodávkou.

Platné nároky z *Elementární nároky na RFP* (file: CSAS_RFPSummary.doc): sekce č. 2, nároky:
id 11; Průběžné testování, nejlépe nějaká forma automatických nebo poloautomatických testů.
id 12; Kvalifikační testování stranou dodavatele před dodávkou zákazníkovi (minimálně pak vyžadovat plány testů, testovací případy a výsledky testů na straně dodavatele).



8 Documentation

| checklist | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Téma | Documentation |
| Minimální nároky: <ol style="list-style-type: none">1. Dokumentace má minimálně sloužit pro:<ol style="list-style-type: none">1.1. zákazník je schopen systém nainstalovat, a bezchybně používat,1.2. dodavatel je schopen systém opravovat udržovat a rozvíjet,1.3. starost o systém je efektivně předatelná mimo okruh duchovních otců.2. Dokumentace je udržovaná nebo je o ní explicitě řečeno, že není udržovaná.3. Dokumentace je tvořena v souladu se zásadou "čistý princip", tzn. nejprve esence, pak obsah, pak forma.4. Dokumentace je tvořena jako integrální součást vlastní práce.5. Obsah a forma dokumentace jsou voleny při povinných znalostech, viz dále. <p>Platné nároky z <i>Elementární nároky na RFP</i> (file: CSAS_RFPSummary.doc): sekce č. 2 a 3: id 2.10; Průběžná tvorba dokumentace, která je pro odběratele klíčová. id 3.5; Dokumentace: existuje taková, aby odběrateli umožnila systém provozovat a dodavateli rozvíjet a udržovat.</p> | |



9 Validation and Verification

| checklist | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| Téma | Validation and Verification |
| Minimální nároky: | |
| <ol style="list-style-type: none">1. Validace a verifikace má sloužit minimálně pro:<ol style="list-style-type: none">1.1. rutinně nejsou vydávány zdroje na něco, co se musí předělat (chce se to, ale je to špatně udělané)1.2. rutinně nejsou vydávány zdroje na něco, co se vlastně nechce (je to dobře udělané, ale chce se něco trochu/ úplně jinak)1.3. rutinně nejsou vydávány zdroje na něco, co se špatně udržuje1.4. rutinně nejsou vydávány zdroje na něco, co se špatně rozvíjí1.5. rutinně nejsou vydávány zdroje na vymyšlení kola2. Na úrovni organizace jsou předepsána tato přezkoumání<ol style="list-style-type: none">2.1. nabídka (nelze bez vědomí PJ, PJ rozhoduje, zda je předmětem přezkoumání)2.2. specifikace (událost hlášená z projektu)2.3. architektura (událost hlášená z projektu)2.4. dbs design (událost hlášená z projektu)2.5. java etc design (událost hlášená z projektu)2.6. prg (průběžné z úrovně organizace)3. Na úrovni projektu vedoucí projektu prokazatelně usiluje o:<ol style="list-style-type: none">3.1. vědět několik nejhorších systematických problémů, které konzumují zdroje3.2. použít přezkoumání, aby k těmto problémům pokud možno nedocházelo tak často3.3. začínat vždy u minimálních nároků4. Na úrovni vztahu odběratel - dodavatel je zajištěno prokazatelné úsilí o donucení odběratele přezkoumat takové meziprodukty, u kterých setrvávající chyba může způsobit vážné ztráty, jak na zdrojích, tak na vztazích.5. Principy, forma, obsah jsou voleny při povinných znalostech, viz dále. | |
| <p>Platné nároky z <i>Elementární nároky na RFP</i> (file: CSAS_RFPSummary.doc): sekce č. 2, nároky:</p> <ul style="list-style-type: none">- id 4; Neformální odborná přezkoumání zásadních technických artefaktů.- id 15; Domluva na společných přezkoumání dodavatele a odběratele nároků a proces a výsledky vývoje. | |