

Software Process Improvement

(Obecné uvedení do tématu)

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

Obsah

1	ÚVOD	1
1.1	VYMEZENÍ TEXTU	1
1.2	STRUKTURA TEXTU	1
2	TERMINOLOGIE	1
2.1	DEFINICE TERMÍNŮ	1
2.2	DEFINICE ZKRATEK	1
3	NEFORMÁLNÍ „TAXONOMIE“ PROBLÉMU ZKVALITŇOVÁNÍ SOFTWAREHO PROCESU	1
3.1	ZÁBĚR ÚSILÍ O ZKVALITŇOVÁNÍ SOFTWAREHO PROCESU VZHLEDKEM K CELÉMU PODNIKU	1
3.2	SCHEMATICKÉ ROZČLENĚNÍ (ZÁKLADNÍCH) MOŽNÝCH PŘÍSTUPŮ K ÚSILÍ O SPI	2
3.2.1	<i>Standardní systematické přístupy: CMM, SEL/NASA</i>	3
3.2.1.1	CMM	3
3.2.1.2	SEL/NASA	4
3.2.2	<i>Doporučení Software Program Managers Network: Nejlepší praktiky pro akvizici softwaru</i>	4
3.2.3	<i>Možné uskutečňování jednotlivých „jednoznačně užitečných“ konkrétních kroků</i>	5
3.2.4	<i>Poznámky</i>	6
3.3	POTŘEBNÁ ORGANIZAČNÍ INFRASTRUKTURA (SOFTWARE ENGINEERING PROCESS GROUP)	6
3.4	POTŘEBNÉ ZDROJE A POTŘEBNÝ ČAS	6
3.5	DALŠÍ NEODDĚLITELNĚ SOUVISEJÍCÍ TÉMATA S REALIZACÍ SPI	6
3.6	ASOCIOVANÁ RIZIKA	7
4	MOŽNÉ KOMBINOVÁNÍ VÝŠE PŘEDSTAVENÝCH KONCEPTŮ	7
5	ZÁVĚR	8
6	REFERENCE	8

1 Úvod

1.1 Vymezení textu

Tento text si klade za cíl, aby umožnil první „kolo“ úvah o možnostech zlepšení softwarového procesu podniku. Text tak obsahuje pokus schematicky a nutně zjednodušeně popsat prostor, resp. podat jakousi neformální „taxonomii“ problému, kterým je zkvalitňování softwarového procesu.

1.2 Struktura textu

Kapitola č. 2 obsahuje nutnou terminologii a vysvětlení důležitých používaných zkratk. Kapitola č. 3 obsahuje neformální popis prostoru problematiky zkvalitňování softwarového procesu. Kapitola č. 4 ukazuje možnosti kombinování konceptů představených v předchozí kapitole a tím i naznačuje možné způsoby při uvažování o zkvalitňování softwarového procesu.

2 Terminologie

2.1 Definice termínů

Softwarový proces zahrnuje veškeré činnosti, metody, praktiky, atd., které jsou používány k vývoji a údržbě programového vybavení a přidružených produktů (např. plány, dokumentace návrhu, zdrojový kód, testovací případy, uživatelská dokumentace, atd.). Činnostmi jsou míněny jak obecně technické (např. specifikace, návrh, programování, konfigurační řízení, testování, dokumentace, atd.), obecně řídicí (tj. vedení, organizace, plánování, monitorování, operativní řízení, atd.) a veškeré další. Jednoduše řečeno, o softwarovém procesu má na jedné straně smysl mluvit v souvislosti s celou organizací (podnikem) – tj. to co se chce, aby splňovaly veškeré softwarové projekty v daném podniku, tj. tzv. standardní softwarový proces organizace (toto lze pak dále přesněji určovat (přizpůsobovat) pro různé domény – tj. části organizace, či typy vyvíjeného softwaru, atd.). Na druhé straně má smysl mluvit o softwarovém procesu daného konkrétního projektu (softwarový proces projektu).

2.2 Definice zkratk

CMM – Capability Maturity Model

DoD – Department of Defense

SE – Software engineering (softwarové inženýrství)

SEI – Software Engineering Institute (Institut softwarového inženýrství při Carnegie Mellon University, který byl založen kongresem pro potřeby ministerstva obrany)

SEL – Software Engineering Laboratory (Laboratoř softwarového inženýrství při NASA/GSFC s participací University of Maryland a Computer Science Corporation)

SEPG – Software Engineering Process Group (skupina pro proces softwarového inženýrství, resp. softwarový proces)

SPI – Software process improvement (zkvalitňování softwarového procesu)

SPMN – Software Program Managers Network, organizace při americkém ministerstvu obrany s následujícím posláním: „to enable managers of large-scale, software-intensive development or maintenance projects to more effectively manage and succeed by identifying and conveying to them management Best Practices, lessons-learned, and direct support“.

3 Neformální „taxonomie“ problému zkvalitňování softwarového procesu

3.1 Záběr úsilí o zkvalitňování softwarového procesu vzhledem k celému podniku

Konkrétní úsilí¹ o SPI může zasáhnout celý podnik, tj. veškeré softwarové projekty. Oblast úsilí o SPI může být, ale také různým způsobem omezena. Zvažme dvě kategorie omezení s pracovními názvy:

¹ Úsilím je míněn obecně jakýkoli přístup k zkvalitňování softwarového procesu, realizovaný obecně jakýmkoli způsobem.

- fyzické omezení
- a logické omezení.

Fyzické omezení může být jakékoli smysluplné omezení typu: divize, skupina divizí, tým, skupina týmů, produkt atd. Logické omezení může být jakékoli smysluplné omezení typu: vývoj informačních systémů, vývoj distribuovaných systémů, atd. Pro fyzické omezení je používán termín organizace, logické omezení částečně koresponduje s intencí termínu doména používaného v [Jeletic96]². V CMM [Paulk93] nejsou explicitě omezení zmiňována nicméně možnost fyzického omezení je zřejmá, na což upozorňuje např. [Kulpa98] (CMM se vztahuje k organizaci a [Kulpa98] říká: organizace není [samozřejmě] projekt a celý podnik nemusí být nutně organizace, v podniku může být více organizací; pochopitelně záleží na velikosti). Kategorizaci omezení a jejich vzájemnou kombinaci lze stanovit jakkoli vhodně.

Nejmenší jednotkou, kterou má v našich úvahách smysl uvažovat je projekt³, a to jen v případě zavádění něčeho nového, experimentování s něčím, atd. To je zřejmé, jelikož smyslem jakéhokoli úsilí o SPI je zkvalitnění výkonu (výkonu ve smyslu provádění softwarového procesu) nějaké organizace (dané fyzickým omezením) při všech projektech, které bude zabezpečovat (tj. instancích zkvalitněného softwarového procesu, tj. softwarového procesu projektu). V souvislosti s projektem pak má ještě smysl uvažovat nové či už existující projekty. Zde poznamenejme, že určité úsilí o SPI, např. systematické a důkladné zavedení inspekci, lze vhodně zavádět již u existujících projektů, obzvláště projektů, které jsou spjaty s nějakým konkrétním produktem. Dále v souvislosti s projektem uveďme, že má smysl rozlišovat projekty, které jsou spjaté s nějakým existujícím produktem, kde v určitých známých okrajových podmínkách probíhá *de facto* trvalý rozvoj, a projekty obecně z pohledu podniku nové.

3.2 Schematické rozčlenění (základních) možných přístupů k úsilí o SPI

V této sekci popsané rozčlenění přístupů k úsilí o SPI vzniklo pro potřeby tohoto textu a nepředstavuje tedy žádné běžně přijaté třídění. Používáme-li termín přístup k úsilí o SPI, tak je třeba říci co tím termínem bude myšleno. Zhruba řečeno tím budeme myslet vymezení následujících tří věcí (okruhů problémů):

- i. určení (přímo nebo strategie) prvků softwarového procesu, tedy čím, jakým tématem se zabýváme;
- ii. stanovení pořadí (přímo nebo strategie), tedy pořadí v jakém se tím zabýváme
- iii. a způsob jakým se tím zabýváme.

Kde, prvkem softwarového procesu obecně rozumíme jakoukoli smysluplně vymezenou oblast softwarového procesu typu: řízení požadavků, validace&verifikace, inspekce, návrh, primární aktivity SE, měření, odhadování a časové plánování, plánování softwarového projektu, softwarová architektura, atd. Slovem zabývat je obecně rozuměno vyvíjet snahu o zkvalitnění příslušného prvku. Přičemž zkvalitnit může obecně znamenat od zavádění věcí, které se nedělají, přes konsolidaci těch, které se dělají až po dosahování určitých kvalitativních a kvantitativních ukazatelů.

Před dalším řekněme:

1. Standardní systematické přístupy uvedené v sekci 3.2.1 (tj. CMM a SEL/NASA) jsou jak název sekce naznačuje celistvé, systematické, důkladné a plně pokrývají a až na málo plně vymezují tři výše zmíněné okruhy problémů: tj. určení čím se zabýváme, v jakém pořadí se tím zabýváme a jak se tím zabýváme. Aplikace těchto přístupů, tak jak jsou koncipovány a zamýšleny vyžaduje jisté neredukovatelné zázemí a zdroje, viz dále sekce 3.2.1, 3.3, 3.4. Tedy většina stupňů volnosti ukázaná dále v sekci 3.5 je pro tyto přístupy neaplikovatelná. Nicméně přístupy CMM a SEL/NASA se mohou smysluplně přizpůsobit, zredukovat, atd. Avšak pak již vlastně uvažujeme o jiných přístupech, viz níže. Dále, přístupy CMM a SEL/NASA se mohou velmi dobře kombinovat navzájem anebo s jinými přístupy.
2. Přístupy uvedené v sekcích 3.2.2 a 3.2.3 se vymezují pouze vzhledem k okruhu problému č. i, tedy čím se zabývat při SPI úsilí. Zůstává zde tedy mnoho volnosti co se týká stanovení pořadí a způsobu jak se vybranými prvky softwarového procesu zabývat. Tedy stupně volnosti ukázané v sekci 3.5 jsou pro tyto přístupy plně relevantní a co více, tyto přístupy musí být v daných mezích právě doformovány, resp. nakonfigurovány. (Pozn. přesně pro tento typ úvah, mimo jiné, má tento text být podkladem.) Například úvahy, zda se vybraným tématem, např. inspekci, zabýváme od pouhého poskytnutí příručky, přes nějaké motivování až po důkladné vyškolení a dohlídnutí na realizaci při projektech je zde tedy plně legitimní; u standardních přístupů CMM a SEL/NASA takovéto úvahy nepřípadají v úvahu. Dále pro přístupy uvedené v sekcích 3.2.2 a 3.2.3 je plně legitimní diskuse o všech aspektech potřebné infrastruktury a zdrojů popsaných v sekcích 3.3, 3.4, což v případě přístupů CMM a SEL/NASA zdaleka říci nelze.
3. Po představení jednotlivých typů přístupů bude k jejich vzájemnému vztahu ještě něco dodáno (konkrétně v sekci 3.2.4).
4. Přístupy lze obecně mnoha různými způsoby kombinovat, viz níže.

² Zde referovaný SPI přístup SEL/NASA a vzápětí referovaný SPI přístup SEI (tj. CMM) bude popsán v následující sekci 3.2.

³ Nebude-li řečeno jinak, projektem je míněn softwarový projekt, tj. projekt při kterém je vyvíjeno programové vybavení.

5. Obecně platí, že neexistuje žádný jednotlivý nástroj, metoda, technika, atd., který by izolovaně způsobil celkové markantní zlepšení produktivity, resp. kvality.

Schematické rozčlenění přístupů pro účely tohoto textu je následující:

1. Standardní systematické přístupy: tímto budeme rozumět CMM (Capability Maturity Model) [Paulk93] vypracovaný v SEI a přístup [Jeletic96] vypracovaný v SEL/NASA nazývaný „experience-based“, či též induktivní [Briand95]. Viz sekce 3.2.1.
2. Přijetí a použití tzv. Nejlepších praktik pro akvizici softwaru (Software Acquisition Best Practices), které poskytuje Software Program Managers Network (SPMN). Viz sekce 3.2.2.
3. Možné uskutečňování jednotlivých „jednoznačně užitečných kroků“, např. zavedení inspekcí návrhu a kódu, atd. Viz sekce 3.2.3.

3.2.1 Standardní systematické přístupy: CMM, SEL/NASA

Zde uvedeme přístup CMM [Paulk93] a přístup [Jeletic96] vypracovaný v SEL/NASA. Jak již bylo řečeno jedná se o dlouhodobé, důkladné, systematické přístupy, které postupně důkladně konsolidují vše co konsolidováno není. Tyto přístupy jsou zavedené. Od roku 1998 platí, že kontraktoři pro ministerstvo obrany spojených států musí být na úrovni 3, kterou definuje CMM. Kromě těchto existují další důkladné přístupy. Nicméně pro účely tohoto textu stačí uvést tyto dva. CMM představuje nejznámější a nejvíce zavedený z kategorie přístupů, tzv. „předepisujících“, které vycházejí z premisy, že je známo jaké věci představují nejlepší praktiky pro softwarový proces a ty předepisuje. SEL/NASA přístup představuje asi nejznámější a v samotné NASA přes 20 let praktikovaný přístup z kategorie tzv. „induktivních“ neboli „bottom-up“ přístupů, které vycházejí z premisy, že zlepšovat se má to, co odpovídá stavu, cílům, problémům, atd. konkrétní dané organizace.

3.2.1.1 CMM

CMM [Paulk93] předepisuje 5 úrovní vyspělosti / vyzrálosti. Pro každou úroveň předepisuje určitou sadu klíčových oblastí, které je nutno zvládnout. Pro každou oblast předepisuje cíle, nutné závazky, nutné schopnosti, klíčové praktiky, způsoby měření a analýzy, verifikaci realizace. Pět úrovní vyspělosti spolu s klíčovými oblastmi jsou:

1. Initial Level: *ad hoc* proces.
2. Repeatable Level (opakovatelná úroveň): řízení požadavků, plánování sw projektu, sledování a řízení sw projektu, řízení subdodávek softwaru, zajištění jakosti softwaru, konfigurační řízení.
3. Defined Level (definovaná úroveň): zaměření organizace na proces, definice procesu organizace, program školení a vzdělávání, integrované řízení vývoje softwaru, aplikace softwarového inženýrství, koordinace mezi skupinami, přezkoumání (Peer reviews).
4. Managed Level (řízená úroveň): řízení jakosti softwaru, kvantitativní řízení procesu.
5. Optimizing Level (optimalizující úroveň): prevence defektů, řízení změny technologií, řízení změny procesu.

Postup při postupování dle tohoto modelu je schematicky následující:

1. cílené zhodnocení stávajícího stavu (CMM-based assessment);
2. vypracování tzv. akčního plánu pro další postup (typicky o úroveň výše);
3. realizace plánu; Dále se pokračuje znovu od bodu č. 1.

Infrastruktura pro realizaci je nějaká podoba tzv. SEPG (Software Engineering Process Group). Doba pro posun o jednu úroveň bývá 2-3⁴ roky (ovšem s velkými variacemi 11 měsíců až 10ky měsíců). SEPG by měla mít dle doporučení SEI kolem 2% personálu⁵, kterému se slouží. Návržnost se uvádí 3 – 5 let. V SEI je k dispozici (v elektronické podobě a zadarmo; což platí obecně téměř vždy) mnoho materiálů k pomoci zavádění CMM a k mnoha podstatným tématům softwarového inženýrství.

Stručně ke kontextu, u zrodu CMM stál Humphrey nejdříve v IBM, od založení SEI pak v SEI. Humphrey byl inspirován prací P. Crosby v oblasti jakosti. Práce na CMM probíhaly od poloviny 80-tých let. První verze byla uvolněna v roce 1991 a nyní platná verze 1.1 v 1993. V současné době existuje celá rodina tzv. CMM modelů (pro systémové inženýrství, pro práci s lidmi, pro systémové a softwarové inženýrství), přičemž původní model a ten, o kterém je zde řeč se nazývá SW-CMM (tedy CMM pro software). Řadu let probíhá

⁴ Pokud nebude řečeno jinak, uváděné údaje pochází z následujících materiálů [Fowler90, Haley95, Hayes95, Herbsleb94, Jeletic96]. Pro stručnost pokud to nebude z nějakých důvodů důležité nebudou u jednotlivých údajů uváděny zdroje; na případnou žádost mohu upřesnit, doložit, atd.

⁵ Toto ovšem nepředstavuje úplnou personální náročnost. SEPG si typicky na jednotlivé úkoly *ad hoc* vytváří tzv. pracovní skupiny pro úkol, které daný úkol pod vedením SEPG realizují. Např. příručky k používání jednotlivých konkrétních nástrojů, tak aby byla nějaká část standardního softwarového procesu organizace (např. konfigurační řízení) vhodně zpřístupněna a realizována pro všechny zasažené lokální kontexty. Tak např. v [Haley95] je uváděno, že v různých pracovních skupinách zřízených SEPG někdy pracovalo i 100 lidí v kontextu vývojové organizace mající 1200 osob. Toto koresponduje s nároky uváděnými v SEL/NASA, kde na 200 vývojářů požadují 10-20 tzv. proces analytiků, viz dále.

odborná diskuse na téma verze 2 SW-CMM a jeho architektury (též inspirovaná existencí modelu SPICE). Pozn. dále bude v textu používáno zase již jen CMM.

3.2.1.2 SEL/NASA

Přístup SEL/NASA [Jeletic96] vychází ze základní premisy, že určitá vývojová organizace musí úsilí o zkvalitňování zaměřovat na zamezení minulých problémů a na opakování minulých úspěchů. Vychází z premisy, že zkušenost organizace získávána při projektech nesmí zapadat, ale musí být zdrojem pro úsilí o zkvalitňování. Ilustrujme, přístup CMM předpokládá, že když proces organizace dosáhne úrovně 2, tak výsledné projekty, resp. produkty, budou příslušně na výši a to je obecně dobré. SEL/NASA říká, vždyť pro tuto organizaci může být problém hlavní problém spolehlivost, pro jinou čas dodávky a od toho se příslušně odvozují napravné akce.

Kontext ve kterém se SEL/NASA realizuje je tzv. Software Process Improvement Organization. Tato organizace má tři části: organizaci vývojářů (typicky 200 osob), organizaci analytiků procesu (10-20 osob), organizaci podpory (4 lidi). Přičemž úkolem vývojářů (tj. analytici, návrháři, programátoři, vedení projektu, prostě všichni zúčastnění) je vývoj. Úkolem analytiků procesu je znát a průběžně aktualizovat znalost softwarového procesu organizace, sledovat konkrétní projekty, znát cíle organizace a tyto veškeré znalosti syntetizovat do modifikace procesu organizace (skrze manuály, příručky, školení, atd.), aby se neustále potřebným a chtěným směrem zkvalitňoval. Úkolem podpory je shromažďovat, udržovat a zpřístupňovat údaje a informace z projektů a výsledky práce analytiků procesu.

Proces zkvalitňování je trojfázový (neustále se opakující):

1. Understanding (Porozumění): porozumění softwarovému procesu organizace, analýza údajů z projektů, analýza problémů, analýza cílů organizace a konečně identifikace potenciálních oblastí na zlepšení (např. inspekce architektury a regresní testování);
2. Assessment⁶ (Zhodnocení): zhodnocení potenciálních zlepšení na pokusných projektech;
3. Packaging (Zavedení): zavedení těch věcí, které se osvědčily. Tímto je míněno důkladné zavedení, od poskytnutí nové dokumentace, přes modifikaci stávající dokumentace, přes školení, po příručky ke konkrétním nástrojům, atd. Dále se pokračuje znovu od fáze č. 1.

SEL/NASA má koncept domén, to znamená, že výše popsany proces zkvalitňování se děje vzhledem k nějaké doméně, např. vývoj administrativních systémů, vývoj v divizi XY, atd. Software Process Improvement Organization je spjata s nějakou takovou doménou (úsilí o zkvalitnění se týká pochopitelně všech projektů dané organizace).

Nároky jsou závislé pochopitelně na velikosti SPI organizace (počet podpůrného personálu je relativně stabilní, pro co více vývojářů slouží, tím menší režie, atd.). Vyjádřeno v % vzhledem k vývoji bez tohoto SEL/NASA přístupu, pak nároky jsou:

- dodatečné nároky na vývojáře: 2%
- analytici procesu: 5-15%
- podpůrný personál: 2-7% .

Vyjádřeno v osobách pro počet vývojářů 200, pak analytiků procesu je třeba 10-20 a podpůrného personálu stabilně 4 osoby. K uvedeným nárokům, nároky na analytiku procesu a podpůrný personál odpovídají skutečně plnohodnotnému, důkladnému, a systematickému provádění SPI úsilí. Dále k nárokům viz sekce 3.3 a 3.4 a poznámka pod čarou č. 5. Autoři [Jeletic96], kde je přístup SEL/NASA popsán poskytují podrobné koncepční srovnání obou přístupů, tj. CMM a SEL/NASA a dále upozorňují, že tyto přístupy nejdou principiálně proti sobě, naopak je možné a i vhodné je používat kombinovaně.

3.2.2 Doporučení Software Program Managers Network: Nejlepší praktiky pro akvizici softwaru

Software Acquisition Best Practices (doslova: nejlepší praktiky pro akvizici softwaru, dále jen „nejlepší praktiky“) je odpověď SPMN (zkratka plus vysvětlení viz sekce 2.2 na začátku textu) na velké problémy se zvládáním velkých softwarových projektů. Je to odpověď na volání vedoucích velkých projektů po něčem co je přímo použitelné, co má bezprostřední dopad, atd. Cituji doslova „Discussions with many managers of large-scale software projects in both DoD and industry indicate they need *directly useful strategies, practices, and techniques* to substantially accelerate the pace of process improvement.“⁷ [Norm96a]. Výsledkem těchto palčivých nároků bylo zformování SPMN nejprve v rámci U.S. Navy a od roku 1994 v rámci celé armády. SPMN vypracovalo sadu „nejlepších praktik“ a vydalo The Program Manager's guide to Software Acquisition Best Practices [Norm96b]. Dále je k dispozici několik doprovodných příruček.

Na identifikaci, vypracování a oponentuře „nejlepších praktik“ se podílely 100ky lidí z armády, průmyslu a universit. Identifikované praktiky prošly extensivní oponenturou. Doslova tyto praktiky představují téměř absolutní autoritativnost. Identifikované praktiky jsou empiricky prokázané, že adresují identifikaci, řešení a předcházení problémů při řízení velkých projektů.

⁶ Upozornění, termín „Assessment“ zde má jiný význam než v kontextu CMM.

⁷ Zdůraznění kursivou autorem tohoto textu.

K dispozici je:

1. devět principiálních nejlepších praktik;
2. tzv. řídicí panel projektu (zásadní ukazatele, identifikace zásadních metrik, varovné hodnoty, etc.);
3. sada otázek, která pomáhá zjistit jak vedení projektu rozumí a řídí daný projekt;
4. sada nejhorších praktik, návyků, atd., kterým se musí předcházet;
5. kvantitativní cíle, kterých je třeba dosahovat;
6. test stavu projektu.

Příčemž devět principiálních nejlepších praktik je:

1. Formal Risk Management;
2. Agreement on Interfaces;
3. Formal Inspections;
4. Metrics-Based Scheduling and Management;
5. Binary Quality Gates at the Inch-Pebble Level;
6. Program-wide Visibility of Progress vs. Plan;
7. Defect Tracking Against Quality Gates / Targets;
8. Configuration Management;
9. People-Aware Management Accountability.

Materiál [Norm96b] identifikuje ještě další „best practices“ sružené do následujících sedmi oblastí:

1. Risk Management;
2. Planning;
3. Program Visibility;
4. Program Control;
5. Engineering Practices and Culture;
6. Process Improvement;
7. Solicitation and Contracting.

Výše uvedená iniciativa představuje velmi cenný nástroj. Prostě existuje identifikovaná sada oblastí, velmi přesně vymezená, na kterou se může organizace bezprostředně koncentrovaně zaměřit. Přijmutí a realizace těchto praktik může probíhat různě, od pouhého dání k dispozici až po velmi důkladné vyložení, vyškolení a prosazení při každém projektu. Některé věci vyžadují přípravu, např. požadovaná měření, atd.

Zavádění těchto praktik lze též velmi dobře kombinovat s ostatními přístupy. [Norm96a] přímo píše: “Those familiar with process improvement models such as the CMM will quickly realize that these practices supply tactical solutions to the model’s strategic orientation. ...”

Tématem k diskusi je jistě jak tento nepochybně jedinečný nástroj, jakým „nejlepší praktiky“ jsou, rozumně opravdu uvést v život, tedy prosadit při projektech. Dále je třeba si uvědomit, že praktiky představují rovnou a najednou identifikovanou sadu věcí, na které se okamžitě zaměřit, což je nesmírně cenné, nicméně složitost celé věci dále zůstává (např. dělat dobře měření, které je bezpodmínečně nutné je problém sám pro sebe, atd.; na toto nepřímou též upozorňuje [Maibor97] v odstavečku příznačně nazvaném „The Myth of Best Commercial Practices“.

3.2.3 Možné uskutečňování jednotlivých „jednoznačně užitečných“ konkrétních kroků

Další možný přístup, který nemá, žádný protějšek v nějakém standardním modelu, atp. je nejprostší možnost, která je nasnadě. Tou možností je uskutečňování jednotlivých „jednoznačně užitečných“ konkrétních kroků. Takovými kroky, resp. činy mohou být:

1. skutečné prosazení věcí, které se empiricky osvědčily (někdy se též říká „leverage points“): např. důkladné zavedení inspekce návrhu a kódu, tam kde se předtím důkladně nedělaly, mívá velké přínosy; dalšími takovými věcmi jsou důkladné zvládnutí testování, důkladné zvládnutí problematiky architektury, etc.;
2. poskytnutí výkladu k důležitým věcem: např. rozbor a výklad koncepční organizace primárních aktivit SE (též lze říci problematika modelů SDLC);
3. poskytnutí užitečných věcí: např. mistry dokumentů, atd.

Zde je vidět, že se rýsuje potenciálně spojitý prostor na kolik takových věcí se zaměřit, jakým způsobem je dávat k dispozici, prosazovat, zajišťovat, atd. Dále jakým způsobem je identifikovat, atd.

Tento přístup, jakkoli „nakonfigurovaný“ lze velmi dobře kombinovat s přístupy uvedenými výše. Tento přístup lze odstupňovat různě pro různé kontexty v podniku, atd.

Další poznámky:

a) identifikace, assesment (ano / ne / jaký), využití DIDs (Data Item Definition), v [Haley95] jsou jejich konkrétní „leverage points“ (např. zohlednění systémového inženýrství), Peer review obecně a inspekce nejen návrhu a kódu!, článek No New Models! v IEEE Software, zpřístupnění prací typu Anchoring the Software Process od Boehm, atd.

3.2.4 Poznámky

Nyní můžeme říci, CMM spolu s SEL/NASA představují skutečně poctivý, dlouhodobý, náročný program. „Nejlepší praktiky“ představují sadu zásadních věcí, které lze hned začít prosazovat. „Jednoznačně užitečné“ kroky, resp. činy představuje dále již neredukovatelnou úroveň, jak se o něco snažit. Nicméně je třeba říci, že přístupy lze vhodně kombinovat, např. úspěšné uskutečnění jednoho izolovaného zlepšení např. s inspekcemi, může velmi stimulovat růst celého úsilí.

CMM a SEL/NASA postupují problém od problému a nejdou dál, dokud není dané téma skutečně důkladně vyřešeno. "Nejlepší praktiky" nastolí témata všechna najednou a je už jen otázkou dalšího úsilí jak jsou zvládnána.

Další poznámky:

a) Intenzivně a důkladně prováděnou syntézu přístupů CMM a SEL/NASA lze považovat za maximum. Přístupy představené v sekcích 3.2.2 a 3.2.3 jakkoli „nakonfigurované“ možnostmi představenými v sekcích 3.3, 3.4 a 3.5 a jakkoli vzájemně zkombinované jsou vždy z principu různými podmožinami výše uvedeného maxima.

b) Je evidentní, že obecně platí čím „skromnější“ úsilí o zkvalitnění, tím skromnější výsledky. Nicméně neplatí zde nutně přímá úměra. Vhodný dílčí „skromný“ zásah (eg. inspekce návrhu a kódu) může mít v daných konkrétních podmínkách relativně velký přínos; dlouhodobě a systematicky na to spoléhat však nelze.

3.3 Potřebná organizační infrastruktura (Software Engineering Process Group)

K zajištění jakéhokoli systematického úsilí o SPI je třeba nějaká organizační infrastruktura. Běžně se nazývá Software Engineering Process Group (SEPG). (výjimkou je kontext SEL/NASA).

V příručce pro založení a činnost SEPG [Fowler90] definují následující základní strukturu: steering committee, samotná skupina a technické pracovní skupiny. Samotná SEPG má mít 1-3% (doporučeno 2%) velikosti organizace, které slouží. Protože samotná SEPG je malá, tak dále využívá služeb technických pracovních skupin na jednotlivé úkoly. Obsazování SEPG a technických pracovních skupin osobami je třeba dělat vhodně tak, aby se zajistila stabilita a zároveň zajistila podpora v rámci organizace (lidé pracující pro technické skupiny mohou celé úsilí pomáhat dále prosazovat na svých pracovištích, atd.)

Konkrétní příklad SEPG z [Haley95]:

- Steering Committee;
- Vedoucí;
- Stálé pracovní skupiny pro: Policy & Procedures, Školení, Nástroje a metody, Databáze softwarového procesu;
- *Ad hoc* skupiny na konkrétní úkol (bývá 10 – 15); tyto jsou zřizovány stálými pracovními skupinami.

Nároky zde uváděné a nároky uváděné u SEL/NASA odpovídají té úrovni úsilí, tak jak je v CMM a SEL/NASA předpokládáno. Lze je tedy brát jako nijak restriktivní a pro realizaci menších úkolů lze z nich slevovat.

3.4 Potřebné zdroje a potřebný čas

Schematicky:

- případ SEPG: 1-3% personálu (doporučeno 2%), kterému se slouží plus obsazení pracovních skupin na jednotlivá témata (vyřešení jednoho tématu může představovat nároky např. od 1/4 člověka na měsíc po n člověko-let),
- případ SEL/NASA: 2% režie u vývojářů, 5-15% analytici procesu, 2-7% podpurný personál (% jsou z nákladů na samostatný vývoj).

Doba na posun o jednu úroveň v CMM je 2-3 roky při příslušném úsilí (pozn. data jsou k dispozici jen do úrovně 3 včetně).

Témata k další diskusi: opět, možné slevování při menších cílech.

3.5 Další neoddelitelně související témata s realizací SPI

Vybraná související témata:

1. Přístupy k zavádění (a prosazování) identifikovaných prvků softwarového procesu (Poskytnutí materiálů k dispozici; poskytnutí školení; důkladné prosazení; vyzkoušení na jednotlivých projektech; poskytnutí v čí k dispozici s nějakou motivací za aplikaci (s pomocí, či bez pomoci); etc.).

2. Měření (Je zde třeba říci, že měření se obejít nedá. V nějaké formě je třeba ho realizovat. CMM, SEL/NASA, „Nejlepší praktiky“ a některé „jednoznačně užitečné“ kroky (např. zavádění inspekce) na něm stojí.).
3. Assessment.
4. Popis a definice SP (dokumentace SP na úrovni organizace).
5. Školení a vzdělávání.
6. Centrum vs. lokality (Pro efektivní prosazování změn, je třeba zajistit, že standardní celopodnikové, resp. celo-organizační věci se příslušně promítnou do lokálních kontextů.).

3.6 Asociovaná rizika

Nejzákladnější bývá uváděno: dostatek podpory od vedení; dostatek prostředků, aby celé úsilí neustrnulo v paralýze; dostatečná motivovanost celé organizace, speciálně vedení všech stupňů.

4 Možné kombinování výše představených konceptů

Výše uvedený schematicky naznačený prostor problému zkvalitňování softwarového procesu, naznačil téměř nepřehledné množství konkrétních možností postupů, nikoli však principů – těch je, jak je vidět, pouze pár. Velké množství možností skýtá např. nakonfigurování „nejlepších praktik“ (viz sekce 3.2.2) anebo „jednoznačně užitečných“ kroků (viz sekce 3.2.3) koncepty ze sekcí 3.3, 3.4 a 3.5, atd.

Schematicky lze možný prostor kombinování výše představených konceptů, tj. vlastně možností jakou cestou se při úsilí o SPI v podniku konkrétně vydat, popsat následovně:

1. je možno kombinovat základní přístupy⁸ k úsilí o SPI (popsané v sekci 3.2);
2. tyto je možno, resp. nutno příslušně nakonfigurovat (naformovat).

Okamžitě dodejme, že konkrétní úsilí o SPI (z prostoru popsaneho výše uvedenými dvěma body) může mít v podniku různý záběr (viz sekce 3.1), tedy může probíhat jen v určitém kontextu. Takovýchto různých kontextů může být, ale v podniku určeno více a v každém z nich může probíhat jiné konkrétní úsilí o SPI (pozn. je pravděpodobné, že pokud by toto nastalo, pak by se různá konkrétní úsilí o SPI lišila spíše v intenzitě, důkladnosti nebo na pozadí jednoho centrálního úsilí by se paralelně děla různá dílčí úsilí v hodná pro dané lokální kontexty, atd.). Dále dodejme, že celý představený prostor možností může, resp. musí být doplněn vhodným časováním.

Podrobněji ad bod č. 1: Ke smysluplnosti a často i vhodnosti kombinování základních přístupů bylo učiněno dost poznámek již v průběhu textu. Shrnutí, obecně jsou možné všechny kombinace základních přístupů a některé jsou i velmi vhodné.

Podrobněji ad bod č. 2: Jak bylo již uvedeno výše v textu (začátek sekce 3.2), tak přístupy CMM a SEL/NASA nemají moc stupňů volnosti (míněno měřit se bude či nebude, atd.) zatímco zbývající dva přístupy je třeba dokonfigurovat (doformovat). Následuje tabulka č. 1, která schematicky ukazuje tyto možnosti.

stupeň volnosti \ přístup	CMM & SEL/NASA (sekce 3.2.1)	„nejlepší praktiky“ (sekce 3.2.2)	„jednoznačně užitečné“ kroky ⁹ (sekce 3.2.3)
přístupy k zavádění (sekce 1)	do značné míry až plně předurčeno ¹⁰ přístupem	nutno stanovit	nutno stanovit
měření (sekce 2)	stejně jako výše	nutno stanovit	nutno stanovit
assessment (sekce 3)	stejně jako výše	nutno stanovit	nutno stanovit
školení a vzdělávání (sekce 5)	stejně jako výše	nutno stanovit	nutno stanovit
centrum vs. lokality (sekce 6)	stejně jako výše	nutno stanovit	nutno stanovit

Tabulka č. 1

Poznámky k tabulce: i) Nároky na potřebnou organizační infrastrukturu (viz sekce 3.3) a zdroje (viz sekce 3.4) jsou u přístupů CMM a SEL/NASA více méně dány. U zbývajících přístupů jsou nároky buď odvozeny od jejich

⁸ Ve praxi to, ale znamená, že ty přístupy jsou už nějak vhodně „nakonfigurovány“ (resp. doformovány); viz bod č. 2 níže.

⁹ Tento přístup se „konfiguruje“ jednak tím, že se tyto kroky stanoví (může být jeden či více) a dále obecně pro každý takový krok platí uvedené stupně volnosti.

¹⁰ Tímto je míněno: důkladnost, rozsah, atd. nikoli však konkrétní způsob realizace.

konkrétního nakonfigurování a případné kombinace nebo *vice versa*. ii) Konkrétní určení jednotlivých stupňů volnosti není na sobě zcela nezávislé.

Další poznámky:

a) Možnost začít s něčím malým a jasným a poměrně jednoduchým (např. na jednom projektu zkusit inspekce); to může mít několik pozitivních výsledků: něco je vidět, ověří se „leverage points“, vyškolí se lidi, zkusí se měření, získá se „buy-in“, atd.

b) Zavádění „jednoznačně užitečných“ věcí s předchozím assessmentem (jakéhokoli druhu) či bez něj (*deus ex machina*).

c) Příklad TTM15 / Ericsson [Allen97] je poučný z několika důvodů: (i) ukázka prosazení „pár věcí“ skrz naskrz, (ii) ilustrace několika souběžných úsilí, projekt TTM15 běžel na popředí celkového úsilí o CMM a dalších!

d) Příběhy jednotlivých firem.

e) Může jet na různých místech v podniku paralelně více úsilí (běžné pro SEL, běžné pro velké podniky).
atd.

f) *Initial process improvement effort should be centered at leverage points, SEPG Guide [Fowler90], p. 102.*

g) Nepochybně smysluplné je např. „nejlepší praktiky“ dát *en bloc* - jak jinak - (tj. určité vybrané „vše“ a postupné realizování) spojit s jedním důkladným realizováním „leverage point“ (např. inspekce), který samozřejmě je součástí onoho vše u těch praktik! (A tím potenciálně budovat předpolí pro skutečně systematický důkladný dlouhodobý přístup typu SEI/SEL.)

h) Ad možnosti kombinace: lze více přístupů najednou a v různých variantách; např. (systematická) + „nejlepší praktiky“ + leverage point; zároveň praktiky mohou být někde zkoušené s podporou, jinde dobrovolné, může se užívat motivace; mohou se připravit balíky pro inspekce, měření, architekturu a motivace na jejich aplikaci, atd.; paralelně poskytnutí rámce, atd.

i) Ad příklady literatury, kde zmiňují / doporučují kombinace různých přístupů: i) přístupy CMM spolu se SEL/NASA [Jeletic96]; ii) CMM spolu s „nejlepšími praktikami“ [Norm96a]; iii) příklad projektu TTM15 [Allen97] (na pozadí úsilí o CMM a dalších projekt TTM15 usiluje o zlepšení v následujících oblastech: týmová práce, přezkoumání a inspekce, inkrementální vývoj, metody & nástroje & školení, zaměření na zákazníka a informace o projektu; a to k dosažení následujících konkrétních cílů: čas uvedení do provozu od „zmrazení“ požadavků zákazníka – 15 měsíců, 0.1 chyby na 1000 řádků efektivního zdrojového kódu během prvních 6ti měsíců provozu, velikost projektů pod 500.000 člověko hodin, nové releases po 6-9 měsících); iv) CMM nedělat izolovaně, ale kombinovat s dalšími věcmi [Wieggers98].

5 Závěr

Byl poskytnut text pro první kolo úvah o možnostech SPI v podniku.

6 Reference

- Allen97 Allen, G., et al. „TTM15 – A Large Multi-site Improvement Project,“ *In Proceedings of 6th European Software Engineering Conference & 5th ACM SIGSOFT Symposium on the Foundations of Software Engineering*. Springer & SIGSOFT, Zurich, Switzerland, September 1997.
- Briand95 Briand, L., et al. *AINSI An Inductive Method for Software Process Improvement: Concrete Steps and Guidelines*. Technical Report CS-TR-3498, University of Maryland, Computer Science Dep., MD, August 1995.
- Fowler90 Fowler, P., et al. *Software Engineering Process Group Guide*. SEI-90-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, September 1990.
- Haley95 Haley, T., et al. *Raytheon Electronic Systems Experience in Software Process Improvement*. SEI-95-TR-017, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, November 1995.

- Hayes95 Hayes, W., *et al.* *Moving On Up: Data and Experience Doing CMM-Based Process Improvement*. SEI-95-TR-008, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, August 1995.
- Herbsleb94 Herbsleb, J., *et al.* *Benefits of CMM-Based Software Process Improvement: Initial Results*. SEI-94-TR-013, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, August 1994.
- Jeletic96 Jeletic, K., Pajerski, R., Brown, C., *et al.* *Software Process Improvement Guidebook*. Revision 1, Software Engineering Laboratory Series SEL-95-102, Goddard Space Flight Center, NASA, Maryland, March 1996.
- Kulpa98 Kulpa, M. „Ten Things Your Mother Never Told You About the Capability Maturity Model,“ *In CrossTalk, The Journal of Defense Software Engineering*. Software Technology Support Center, DAPS, UT, September 1998.
- Maibor97 Maibor, D. “Software Acquisition for the ‘90s: One Big Dilemma,“ *In CrossTalk, The Journal of Defense Software Engineering*. Software Technology Support Center, DAPS, UT, July 97.
- Norm96a Norm, B. “Industrial-Strength Management Strategies,“ *In CrossTalk, The Journal of Defense Software Engineering*. Software Technology Support Center, DAPS, UT, August 1996.
- Norm96b Norm, B., (coordinator). *The Program Manager’s Guide to Software Acquisition Best Practices*. Version 2.1, Department of Defense, Software Acquisition Best Practices Initiative, 1996.
- Paulk93 Paulk, M., *et al.* *Key Practices of the Capability Maturity Model*, Version 1.1. SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, February 1993.
- Wiegiers98 Wiegiers, K. „Software Process Improvement: Eight Traps to Avoid,“ *In CrossTalk, The Journal of Defense Software Engineering*. Software Technology Support Center, DAPS, UT, September 1998.