

## Státnice – odborné č. 12

Projekt, správa projektů, správa požadavků. Odhad pracnosti, zdrojů, času a nákladů. SW metriky. Dekompoziční techniky, použití empirických vzorců. Plánování projektů, řízení projektů podle plánu a změn.

### Projekt

Projekt je jedinečný proces sestávající z řady koordinovaných a řízených činností s daty zahájení a ukončení, prováděný pro dosažení cíle, který vyhovuje specifickým požadavkům, včetně omezení daných časem, náklady a zdroji (dle ČSN – předn.).

Projekt je jednorázová transformace vstupů (informace, prostředí, materiál, peníze, schopnosti a dovednosti zúčastněných lidí) na výstupy – cílové produkty za pomoci vývojových činností (uspořádaných do etap, kroků a úkonů) a koordinovaných řídicími činnostmi.

Projekt vždy zaměstnává skupinu lidí a ovlivňuje jiné skupiny lidí. Projekt je vždy spojen s rizikem neúspěchu je jedinečný nikdy zcela přesně nevíme, co nás v průběhu jeho realizace čeká nebo zaskočí.

### Správa projektů

#### Zralost organizace

Organizace, které realizují SW projekty jsou na tuto činnost různě připraveny a vybaveny. Softwarové inženýrství je založeno na předchozích zkušenostech a informacích. Čím více jich organizace má, tím lépe je na realizaci projektu připravena. Z pohledu připravenosti organizace rozeznáváme několik úrovní a mluvíme v této souvislosti o úrovni zralosti organizace.

Pět základních úrovní zralosti organizace je:

1. Počáteční – SW proces je prováděn ad hoc, příležitostně až chaoticky. Několik procesů je definováno, případné úspěchy závisí na osobním úsilí.
2. Opakovatelná – jsou zavedeny základní procesy řízení projektu. Je snaha opakovat procesy úspěšných projektů podobných aplikací.
3. Definovaná – SW proces jak z hlediska řízení, tak i z hlediska SI je dokumentován, standardizován a integrován do ostatních procesů organizace.
4. Řízená – jsou prováděna podrobná měření kvality SW procesu a produktu.
5. Optimalizovaná – soustavné zdokonalování procesů využívá kvantitativní zpětnou vazbu z procesů a testů nových myšlenek a technologií.

## Správa požadavků

### Požadavek

1. Podmínka nebo funkce, kterou účastník projektu potřebuje pro řešení problému nebo dosažení nějakého cíle.
2. Podmínka nebo funkce, kterou musí systém nebo jeho část splňovat, aby vyhověl smlouvě, standardu, specifikaci nebo jinému dokumentu, jenž se na něj formálně vztahuje.
3. Dokumentovaná podoba některého z předchozích dvou bodů.

Požadavky jsou popisem toho, co všechno by se mělo implementovat. Popisují žádané chování systému a jeho vlastnosti a mohou představovat nějaká omezení procesu vývoje systému.

#### Typy požadavků

- Podnikatelské – cíle organizace nebo zákazníka, zadavatele, kterých by ráda prostřednictvím systému dosáhla.
- Uživatelské – cíle uživatelů a úkoly, které musí být uživatelé schopni se systémem provést.
- Funkční – softwarová funkcionalita realizovaná vývojáři systému pro splnění úkolů uživatelů a tím i podnikatelských požadavků.
- Systémové – výrobek složený z většího počtu podsystémů, popřípadě kombinace funkcí plněných SW, HW nebo svěřených lidem.
- parametrické – podnikatelská pravidla, předpisy, nařízení, standardy, kvalitativní parametry, omezení, vnější rozhraní.

Vlastnosti požadavků – úplnost, správnost, proveditelnost, nepostradatelnost, priorita, jednoznačnost, ověřitelnost.

Vlastnosti celé specifikace – úplnost, jednotnost, přizpůsobitelnost, dohledatelnost.

### Řízení požadavků

- Vývoj požadavků
  1. Sběr požadavků
  2. Analýza
  3. Specifikace
  4. Kontrola

Úkoly spojené se získáváním, vyhodnocováním, a dokumentací požadavků na realizovaný systém:

- identifikace tříd uživatelů, kteří budou systém používat,
- Získávání požadavků od zástupců jednotlivých tříd.

- Pochopení jednotlivých uživatelských úkolů a podnikatelských cílů, které se tyto úkoly snaží splnit.
- Analýza informací od uživatelů a odlišení jejich cílů od funkčních požadavků, parametrických požadavků, podnikatelských požadavků, navrhovaných řešení a nadbytečných informací.
- Rozdělení části požadavků mezi softwarové moduly dané architekturou systému.
- Seřazení kvalitativních parametrů podle důležitosti.
- Vyjednání implementačních priorit.
- Zpracování nasbíraných uživatelských potřeb do podoby modelů a psané specifikace požadavků
- Kontrola dokumentovaných požadavků, aby byla jistota, že uživatelským požadavkům rozumí všichni stejně, a aby se případné chyby našly dříve, než je převezme vývojářský tým.

- Správa požadavků

znamená shodnout se se zákazníkem na požadavcích projektu a tuto shodu udržovat; shoda je zakotvena ve specifikaci a modelech. Souhlas zákazníka je jen jedna část, druhou část tvoří schválení požadavků vývojáři – souhlas s realizací systému podle specifikace. Správou požadavků jsou následující činnosti:

- definice směrné podoby požadavků pro danou iteraci projektu (směrná = aktuálně směrodatná dokumentace schválená zákazníkem i vývojáři).
- posouzení navrhovaných změn v požadavcích a vyhodnocení pravděpodobných následků každé změny před jejím schválením.
- řízené zapracování schválených změn do projektu.
- aktualizace projektových plánů podle požadavků.
- vyjednávání nových závazků podle očekávaného dopadu změn požadavků.
- sledování jednotlivých požadavků až k odpovídajícím návrhům, zdrojovému kódu a testovacím scénářům.
- sledování stavu jednotlivých požadavků a změn v průběhu celého projektu

## Odhad pracnosti, zdrojů, času a nákladů

### Odhad ceny a pracnosti

Tento odhad je možno provést jako:

- Odhad se zpožděním
- Počáteční odhad podle minulého podobného projektu
- Použití dekompozičních technik
- Použití empirických modelů

Přesnost odhadu projektu je ovlivněna:

- Přesností odhadu velikosti produktu.
- Schopností převést odhad velikosti na odhad pracnosti, času a finančních nákladů (závisí na dostupnosti spolehlivých metrik z minulých projektů).
- Schopnostmi projektového týmu.
- Stabilitou požadavků projekt a vývojovým prostředím.

### Dekompoziční techniky

Dekompoziční techniky jsou odhady vycházející z problému (rozdělení hlavních funkcí a odhad velikosti nebo pracnosti implementace každé funkce). Používají veličiny LOC (počet řádků kódu) a FP (funkční bod). LOC a FP se používají jako proměnné pro odhad různých veličin v projektu a jako základní údaje o minulých projektech. K odhadu používáme tzv. tříbodový odhad vzorečkem:

$$EV = (s_{opt} + 4 s_m + s_{pes}) / 6$$
, kde  $s_{opt}$  je optimistický odhad,  $s_m$  je střední odhadovaná hodnota,  $s_{pes}$  je pesimistický odhad.

Pro provedení odhadu je třeba stanovit funkce odhadovaného systému a jim odpovídající moduly. Na modulech odhadujeme počty řádků kódu nebo funkčních bodů. Nutné jsou historické údaje o průměrné produktivitě pro realizaci obdobných systémů.

### Empirické modely odhadu

odvozené formule pro pracnost a čas. V odhadech pracnosti empirických modelů vystupují veličiny FP nebo LOC vycházející z řešeného projektu a empiricky odvozené konstanty z předchozích projektů A,B,C. Při jejich použití je vhodné provést porovnání odhadu podle několika modelů. Obecný tvar rovnice modelů je  $E = A + B (ev)^C$ , kde E je pracnost, ev je proměnná LOC nebo FP

$E = 5,2 (KLOC)^{0,91}$  Walson-Felixův model

$E = 5,5 + 0,73(KLOC)^{1,16}$  Bailey-Basilioho model

$E = 3,2 (KLOC)^{1,05}$  Boehmův jednoduchý model

$E = 5,288 (KLOC)^{1,047}$  Dotyův model pro KLOC >9

$E = -13,39 + 0,0545 \text{ FP}$  Albrecht a Gaffneyův model

$E = 60,62 \times 7,728 \times 10^{-8} \text{ FP}^3$  Kemererův model

$E = 585,7 + 15,12 \text{ FP}$  Matson, Baret a Mellichampův model

## Model COCOMO

nejpracovanější a nepoužívanější empirický model je definován ve třech úrovních:

Základní COCOMO model - pracnost a cena jako funkce velikosti programu v LOC.

Střední COCOMO model - pracnost a cena jako funkce velikosti programu v LOC a množiny faktorů (produkt, HW, lidé, projekt).

Pokročilý COCOMO model - navíc odhad faktorů každé etapy softwarového procesu.

Jsou definovány pro tři třídy projektů:

·1. Organický mód, 2. Přejídný mód, 3. Uzavřený mód

Rovnice základního modelu

$E = a (\text{KLOC})^b$ ,  $D = c (E)^d$ , kde E je pracnost v člověkoměsících, D je doba vývoje v měsících, koeficienty a, b, c, d, jsou dány v závislosti na třech třídách projektu

Rovnice středního modelu:

$E = a (\text{KLOC})^b \times \text{EAF}$ , kde EAF faktor pracnosti nabývá hodnot mezi 0,9 a 1,4 dle charakteristik projektu – atributů produktu, hardware, týmu a projektu; koeficienty a, b dány dle tříd projektu.

## Softwarová rovnice

$$E = (\text{LOC} \times B^{0,333} / P)^3 \times (1/t^4)$$

E je pracnost v mm, t je trvání projektu v měsících nebo letech, B je faktor zkušenosti pro  $\text{KLOC} = 5..15$ ,  $B = 0,16$ , pro  $\text{KLOC} > 70$ ,  $B = 0,39$ , P je parametr produktivity (zralost procesu řízení, použité metody, programovací jazyk, SW prostředí, zkušenost týmu, složitost aplikace, 2000 systémy reálného času, 10 000 telekomunikační systémy, 28 000 obchodní aplikace)

minimální doba vývoje  $t_{\min} = 8,14 (\text{LOC}/P)^{0,43}$ , v měsících pro  $t_{\min} > 6$

pracnost  $E = 180 B t^3$ , v mm pro  $E \geq 20$  mm, kde t je v rocích

## Softwarové metriky

metriky produktu, procesu a projektu jsou kvalitativní charakteristiky programů, procesu jejich tvorby a projektu. Mezi důvody pro měření metrik patří: plánování projektu (odhad nákladů, pracnosti, času), kontrola kvality produktu, odhad produktivity, zdokonalení práce (růst výkonnosti organizace). Používané metriky vychází převážně z historických zkušeností (jaká byla produktivita, kvalita, jak extrapolovat a využít pro plánování a odhad současného projektu).

## Základní klasifikace metrik

Ukazatel (indikátor) je metrika nebo kombinace metrik, které poskytují náhled na projekt, proces nebo produkt. Slouží k jejich ohodnocení pro možnost provedení případné nápravy.

Ukazatele procesu umožní náhled na efektivitu existujícího procesu. Metriky procesů jsou sbírány dlouhodobě v průběhu řešení různých projektů. Indikují zlepšení softwarového procesu (strategie).

Ukazatelé projektu umožní odhadnout stav projektu, potenciální rizika, odkrýt oblasti problému, dříve než budou kritické, přizpůsobit směr práce a úkolů, vyhodnotit schopnosti projektového týmu řídit kvalitu software (taktika).

Určujícími prvky kvality software a efektivitu organizace jsou: zkušenosti a motivace lidí, složitost produktu, technologie (metody softwarového inženýrství), obchodní podmínky (obchodní pravidla, termíny), charakteristiky zákazníka (snadnost komunikace), vývojové prostředí (CASE).

Metriky softwarového procesu mohou hrát významnou roli v růstu výkonnosti podniku a měly by být používány v souladu s doporučeními: při interpretaci používat zdravý rozum a cit, poskytovat zpětnou vazbu týmům a jednotlivcům sbírajícím data a provádějícím měření, nepoužívat metriky k hodnocení jednotlivců, vedení jednotlivců a týmů ke stanovení jasných cílů a metrik, s jejichž pomocí by cílů dosáhli, nepoužívat metriky k vyjednávání s jednotlivci ani s týmy, data indikující nějaký problém (nejsou negativní) mají posloužit ke zlepšení procesu, nezaměřovat se na jednu metriku a nezapomínat na ostatní důležité metriky.

Metoda SSPI (Statistical Software Process Improvement) definuje postup

- chyby a defekty jsou kategorizovány podle původu,
- je stanovena cena za opravu chyby,
- sečte se počet chyb podle kategorie, je stanovena celková cena chyb podle kategorie,
- analyzují se kategorie s nejdražšími chybami,
- je snahou modifikovat proces k eliminaci frekvence výskytu chyb v této kategorii.

Projektové metriky slouží k taktickým účelům (odhady času, nákladů, monitorování projektu a vyhodnocování). Metodika je použitelná k měření procesu i projektu (postupně pro každou vývojovou fázi projektu). Model projektové metriky zahrnuje:

- vstupy: měří se zdroje (lidé, zařízení) potřebné pro práci,
- výstupy: měří se předané výstupy nebo produkty vytvořené během projektu,
- výsledky: měří se efektivita (užitečnost) výstupů.

Měření softwaru – používáme dva odlišné přístupy:

- přímá měření – počet řádek kódu, rychlost výpočtu, velikost paměti, počet chyb za určitou dobu,

- nepřímá měření – funkčnost, kvalita, složitost, pracnost, spolehlivost, schopnost údržby.

Metriky orientované na velikost jsou odvozeny z velikosti produktu a normalizovány faktorem kvality či produktivity. Vycházejí ze statistiky minulých projektů. Vstupní hodnoty odvozené z předcházejících projektů zahrnují: počet řádek kódu, pracnost (člověkoměsíce m/m), cenu, počet stran dokumentace, počet chyb, počet defektů, počet realizátorů. Z těchto údajů můžeme odvodit: počet chyb na KLOC, počet defektů na KLOC, cena LOC, počet stran dokumentace na KLOC, počet chyb za člověkoměsíc, počet LOC za člověkoměsíc, cena stránky dokumentace.

Funkčně orientované metriky pracují s funkčním bodem FP. Tato veličina je dána empirickým vztahem mezi spočitatelným měřením informační domény a odhadem složitosti softwaru.

Rozhodující pro její určení jsou údaje: A - počet logicky různých vstupů, B - počet výstupů, C - počet dotazů, D - počet vnitřních logických souborů, E - počet souborů na rozhraních, A-E tvoří TOTAL

$FP = TOTAL \times (0,65 + 0,01 \times \text{suma}(F_i))$ , kde  $F_i$  jsou faktory ( $i = 1..14$ ) složitosti zpracování 0-5.

Rozšířené funkční metriky – třírozměrná funkční metrika pracuje ve třech rozměrech (vedle A-E s dalšími dvěma rozměry F,G), tj. datový (jako FP), funkční (F počet operací transformujících vstup na výstup), řídicí (G počet přechodů mezi stavy)

Existují hrubé odhady počtu LOC na 1 FP pro různé programovací jazyky.

Metriky kvality softwaru – ověření správnosti a spolehlivosti je těžší než odhad složitosti algoritmů

- správnost – měřena počtem chyb na tisíce řádků kódu (KLOC),
- udržovatelnost – snadnost, s níž může být program opraven (při chybě), upraven (při změně prostředí), zdokonalován (při změně požadavků)

Používáme přímé metriky MMTC – střední doba potřebná k realizaci změn (analýza, návrh, implementace, testy, distribuce). Rozeznáváme: Spoilage = cena za opravu defektů objevených po distribuce, Integrita = odolnost vůči náhodným nebo záměrným útokům z vnějšího prostředí, Užitečnost = snaha změřit uživatelskou přívětivost.

Charakteristiky dobrého softwaru z pohledu zákazníka: fyzická a intelektuální dovednost nutná pro zvládnutí systému, čas uživatele potřebný k nabytí mírné zdatnosti v ovládnutí systému, vzrůst produktivity uživatele vzhledem k novému systému, subjektivní ocenění uživateli.

Měření účinnosti odstraňování chyb softwaru – DRE (Defect Removal Efficiency)

$DRE = E / (E + D)$ , E je počet chyb nalezených před předáním uživateli, D je počet defektů po předání. Hodnotu DRE lze definovat vzhledem k jednotlivým vývojovým fázím:

$DRE_i = E_i / (E_i + E_{i+1})$ , kde  $E_i$  je počet chyb nalezených v etapě i,  $E_{i+1}$  je počet chyb nalezených v etapě i+1, vzniklých a neobjevených v etapě i.

## Plánování projektu, řízení projektu podle plánu a změn

Projekt je možno z pohledu posloupnosti akcí a aktivit rozdělit do několika etap, z nichž kromě zahajovací a ukončovací etapy bývají zpravidla časově významnější etapy plánování, realizace, sledování a kontroly. Tyto etapy v sobě obsahují procesy plánování, výkonné procesy a procesy sledování a kontroly.

### Plánování rozsahu projektu

V rámci této etapy je nutné vytvořit provést plánování rozsahu projektu. Odhady pracnosti, času, zdrojů a nákladů jsou hlavními aktivitami etapy označované jako plánování rozsahu projektu. Odhady vychází z hodnot daných: složitostí projektu, velikostí projektu, metrikami minulých projektů, variabilitou v softwarových požadavcích

### Rozsah projektu

je dán řešeným problémem, jmenovitě: funkcí systému, chováním, rozhraním, omezujícími podmínkami a spolehlivostí. Rozsah problému se stanoví na základě interview se zákazníkem

### Schůzky se zákazníkem

Předběžná řízená schůzka se zákazníkem

1. Context free questions (bezkontextové otázky) Kdo tu práci požaduje, kdo ji bude užívat, jaký bude ekonomický užitek při úspěšném ukončení, je ještě jiná možnost, jak to vyřešit?
2. Hlubší pochopení problému a názoru zákazníka Jak byste charakterizoval "dobrý" výstup? Na jaké problémy je toto řešení zaměřeno? Ukažte mi (popište) prostředí, kde to bude systém pracovat? Jsou nějaké speciální požadavky na chování systému a na jeho výstupy.
3. „Meta otázky“ Jste ta správná osoba, která mi může na tyto otázky odpovědět? Jsou vaše odpovědi oficiální? Jsou mé otázky relevantní k danému problému? Nedávám vám moc otázek? Je tu ještě někdo další, kdo by mohl poskytnout doplňující informace? Je ještě něco, na co bych se měl zeptat?

Další setkání jsou formálnější (řešení problému, vyjednávání a specifikace) FAST (Facilitated Application Specification Techniques)

·schůzka realizátorů a zákazníků, řízena neutrální stranou (facilitátorem), pravidla pro přípravu a průběh, agenda, používá se tabule, flip chart apod.

·cílem je identifikovat problém, navrhnout řešení, vyjednat odlišné přístupy a specifikovat předběžnou množinu požadavků

### Plánování času

Při plánování času je nutné rozdělení na zvládnutelné aktivity a úkoly, jejich seřazení a stanovení vzájemných závislostí, aktivitám je potřeba odhadnout pracnost a alokovat zdroje. Je potřebné stanovit odpovědnosti a požadované výstupy v definovaných milnících. Kromě odhadu pracnosti a dekompozice funkce produktu je vybírán vhodný model procesu a také typ projektu a množiny úkolů. Typy projektu rozumíme vývoj nového konceptu, nové aplikace, zdokonalení aplikace, údržbu aplikace nebo reinženýrství. Dále zohledňujeme stupeň rigoróznosti (neformální, strukturovaný,



přísný, rychlé reakce) podle velikosti, počtu uživatelů, životnosti, stability požadavků komunikativnosti uživatele, omezující podmínky a realizátoři.

## **Tvorba plánu projektu**

- komunikace o rozsahu a zdrojích s managementem, realizátorem a zákazníkem,
- definice rizik a návrh technik jejich řízení,
- definice ceny a rozvrh kontrol managementu,
- poskytnutí celkových informací o přístupu k řešení projektu všem, kdo na projektu budou spolupracovat (rozsah, rozpis prací),
- určení způsobu zajištění kvality a řízení změn.

Plán projektu obsahuje: 1. úvod, 2. projektové odhady, 3. strategie řízení rizik, 4. časový harmonogram, 5. zdroje projektu, 6. organizační členění, 7. způsob kontroly, 8. přílohy.

## **Sledování plánu**

- periodické schůze o stavu projektu (zprávy o pokroku a problémech),
- vyhodnocování výsledků všech kontrol během procesu,
- sledování plnění formálních milníků v plánovaných termínech,
- porovnání skutečných a plánovaných časů pro zahájení všech úkolů,
- neformální setkání s realizátory k získání jejich subjektivního odhadu postupu a předpokládaných problémů.

nebo formálně:

- vedení a řízení výkonu projektu, sledování a kontrola prací na projektu, integrované řízení změn,
- ověření rozsahu projektu, kontrola rozsahu,
- kontrola časového plánu,
- kontrola nákladů,
- operace pro zajištění kvality, provádění kontrol kvality,
- vybudování projektového týmu, řízení projektového týmu,
- distribuce informací, zprávy výkonu, řízení účastníků,
- sledování a kontrola rizik.