**Časť I – Softvér a inžinierstvo**

**ČO TO JE SOFTVÉROVÉ INŽINIERSTVO? .......................................................8**

PREČO JE ŠTÚDIUM SOFTVÉROVÉHO INŽINIERSTVA DÔLEŽITÉ?..................................8 ZAVEDENIE POJMU SOFTVÉROVÉ INŽINIERSTVO ........................................................8

NIEKOĽKO DÔLEŽITÝCH POJMOV...............................................................................9

**HISTÓRIA SOFTVÉROVÉHO INŽINIERSTVA................................................10**

60-TE ROKY .............................................................................................................10

70-TE ROKY .............................................................................................................10

80-TE ROKY .............................................................................................................11

90-TE ROKY .............................................................................................................12 ZAČIATOK 21. STOROČIA.........................................................................................12

**SOFTVÉROVÝ PRODUKT....................................................................................14**

TYPY SOFTVÉROVÝCH VÝROBKOV...........................................................................14 TYPY SOFTVÉROVÝCH APLIKÁCIÍ.............................................................................14 VLASTNOSTI SOFTVÉRU...........................................................................................15

PROBLÉMY S TVORBOU SOFTVÉRU...........................................................................16

**PROCES VÝVOJA SOFTVÉRU ............................................................................20** ŽIVOTNÝ CYKLUS SOFTVÉRU...................................................................................21 MODEL ŽIVOTNÉHO CYKLU SOFTVÉRU ....................................................................23 MODEL VYTVOR A OPRAV .......................................................................................23 V-MODEL VÝVOJA SOFTVÉRU..................................................................................24 ZNÁME MODELY ŽIVOTNÉHO CYKLU SOFTVÉRU ......................................................26

ŠPECIALISTI V ŽIVOTNOM CYKLE SOFTVÉRU............................................................27

**SOFTVÉROVÉ PROCESY .....................................................................................29**

KLASIFIKÁCIA SOFTVÉROVÝCH PROCESOV..............................................................30

**ETICKÝ KÓDEX SOFTVÉROVÉHO INŽINIERA............................................31**

**NA ZÁVER... .............................................................................................................36** ČO JE PRAVDA?........................................................................................................36

**Čo to je softvérové inžinierstvo?**

− ??

−

−

−

**Softvérové inžinierstvo** ≠ **programovanie!!!!**

**Prečo je štúdium softvérového inžinierstva dôležité?** − ? −

−

hardvér

vývoj softvéru

náklady

údržba softvéru

1955 1970 1990

**Zavedenie pojmu softvérové inžinierstvo** − zavedenie pojmu “softvérové inžinierstvo” na konferenciách v rokoch

???-??? spolu s pojmom “softvérová kríza”, − softvérové inžinierstvo má **?? rokov**

Poznámky: 8

|  |  |  |
| --- | --- | --- |
| **Niekoľko dôležitých pojmov**  **Kvalita** - súhrn vlastností a charakteristík výrobku, procesu alebo služby, ktoré preukazujú jeho schopnosť splniť určené alebo odvodené potreby.  (ISO 8402).  − nie ako absolútna miera, ale ako stupeň splnenia požiadaviek, resp. potrieb.  − ak cieľom bude vyvinúť nespoľahlivý softvér, potom čím menej bude spoľahlivý, tým je kvalitnejší.  S  p  lnenie  p  ožiadaviek na v  ý  robok  Č  **Kvalita**  as  C  e  na  **Kvalita je ...**  − miera stupňa dokonalosti (Oxfordský slovník)  − splnenie požiadaviek (Crosby)  − vhodnosť k danému účelu (ISO 9001)  − schopnosť produktu alebo služby plniť dané potreby (BS 4778)  **Metóda**  − procedúra, postup ako dosiahnuť nejaký výsledok  **Technika**  − spôsob ako vykonať určité činnosti  **Prostriedok**  − nástroj, ktorý sa použije pri riešení problému  **Metodológia**  − súbor metód a techník predstavujúcich overený postup Pozor na anglické pojmy!!!  Poznámky: 9 | | **História softvérového inžinierstva**  **60-te roky**  − ťažkosti spojené s vývojom väčších programov; − zavedenie pojmu “softvérové inžinierstvo” na konferenciách v rokoch  1968-1969 spolu s pojmom “**softvérová kríza**”;  − hľadanie jednoduchého a účinného riešenia na existujúce ťažkosti vyúsťuje do **štruktúrovaného programovania**, ktorého začiatok sa zvykne stotožňovať s Dijkstrovým článkom o používaní príkazu GOTO.  **70-te roky**  **Začiatok 70-tych rokov**  − snahy o prekonanie softvérovej krízy vedú cez výskum „dobrých“ programovacích praktík;  − zdôrazňovanie ľudských faktorov v programovaní;  − rozpoznanie výhod **návrhu zhora nadol**, **postupného zjemňovania** a **modulárneho programovania**;  − zavedenie nových jazykov (vrátane **Pascalu**) a nových techník programovania v tímoch (pravidlo vedúceho programátora);  − uvedomenie si **životného cyklu tvorby softvéru**;  − jednoduché techniky štruktúrovaného programovania sa nahrádzajú **metodológiami**: štruktúrovanou analýzou, návrhom;  − podpora manažmentu tvorby softvéru;  − snahy o **zabezpečenie kvality** spôsobujú vývoj procedúr zameraných na systematické testovanie, zavedenie pojmov formálnej správnosti programu.  Poznámky: 10 |
|
| **Koniec 70-tych rokov**  − snahy o automatizáciu úlohy **syntézy programov** vedú k návrhu niekoľkých metód schopných automatickej syntézy programov (veľkosti programov nepresahujú niekoľko riadkov);  − využitie deduktívneho, induktívneho a iných transformačných prístupov pri realizácii **formálnej transformácie špecifikácie do programu**;  − používanie **abstrakcie** a modulárnej dekompozície ako návrhových techník;  − rozpracovanie pojmu **abstraktných dátových typov**, ktoré intenzívne  pokračuje aj v 80-tych rokoch;  − ďalší rozvoj metodológií, pričom sa vyhraňujú dátovo a procesne orientované metódy;  − uvedomenie si významu a dôležitosti etáp špecifikácie a návrhu;  − prvé princípy zo začiatku 70-tych rokov sa začínajú široko používať v počítačovom priemysle.  **80-te roky**  − rozšírenie používania programovacích prostredí;  − snaha o počítačovú podporu jednotlivých metód s čím súvisí vývoj **CASE prostriedkov**;  − vývoj nových programových paradigiem ako **objektovo-orientované programovanie**;  − ďalší vývoj **funkcionálneho**, **logického** ako aj imperatívneho  programovania;  − vývoj **formálnych metód špecifikácie** a návrhu väčších programov;  − pokroky v oblasti **paralelného programovania**;  − zvýšenie pozornosti etape prevádzky a údržby softvéru, dôsledkom čoho je vývoj systémov na podporu údržby verzií softvérových objektov a riadenia konfigurácií softvérových systémov.  Poznámky: 11 | | **90-te roky**  − rozšírenie **prototypovania**;  − vývoj softvéru na základe **znovupoužiteľnosti** (angl. reusability) a **komponentov** (najmä v súvislosti s objektovo-orientovaným prístupom k tvorbe softvéru);  − ďalší vývoj objektovo-orientovaného programovania, rozšírenie jazyka **Java**;  − pozornosť sa venuje objektovo-orientovanej špecifikácii a návrhu softvérových systémov, definujú a používajú sa **schémy** a **vzory** (napr.  návrhové vzory)  − aplikácia techník znalostných systémov a umelej inteligencie do softvérového inžinierstva;  − sledovanie kvality/vyspelosti **softvérového procesu** a softvéru použitím **metrík**;  − vývoj **otvorených softvérových systémov**, často s otvoreným zdrojovým textom programu (angl. open source software);  − snahy o efektívne využitie **Internetu**; nové metódy, techniky a prostriedky spolupráce; pozornosť sa venuje tvorbe distribuovaného softvéru a metódam a technikám distribuovanej tvorby softvéru.  **Začiatok 21. storočia** − ???????  **Hľadajte o histórii informácie na webe!!!**  Poznámky: 12 |
|
| **Nové skutočnosti v tvorbe softvéru**   1. zníženie času na vývoj softvéru (komerčné výrobky) 2. zvyšovanie zložitosti softvérových systémov 3. posuny v ekonomike (zníženie ceny hardvéru a zvyšovanie cien vývoja softvéru a údržby) 4. rozširovanie používania softvéru v najrôznejších oblastiach ľudskej činnosti 5. nové požiadavky na softvér (vnorené systémy, mobilné systémy, adaptivita, neviditeľné počítanie) 6. dostupnosť osobných počítačov (posun v zodpovednosti za vývoj) 7. rozširovanie sietí (dostupnosť informácií, distribuované aplikácie, distribuovaná tvorba softvéru) 8. dostupnosť a používanie objektovo-orientovaného prístupu (znovupoužitie, komponentový prístup k vývoju softvéru, multiagentové systémy) 9. grafické rozhrania (okná, ikony, menu,…) 10. problémy s vodopádovým modelom vývoja softvéru − potreba paralelnej práce a teda iného modelu; prototypovanie     Poznámky: 13 | | **Softvérový produkt**  Členovia jednej kultúry vytvárajú veci pre členov inej kultúry.  **Softvér vs. program**   * **softvérový systém** * **softvérový produkt** * **softvérový výrobok**   − zbierka počítačových programov, procedúr, pravidiel a s nimi spojenou  dokumentáciou a údajmi (IEEE, 1994).  Softvér zahŕňa napr.: požiadavky, špecifikácie, opisy návrhu, zdrojový text, testovacie údaje, príručky, …  **Prečo vlastne vytvárame softvér?** − ??  − ?  − ? − ?  **Typy softvérových výrobkov**  − **generické:** softvér sa predáva ľubovoľnému záujemcovi (off the shelf),  COTS – commercial-of-the-shelf,  MOTS – modified-of-the-shelf  − **zákaznícke (na objednávku):** softvér sa vytvára na základe požiadaviek pre konkrétneho zákazníka. **Typy softvérových aplikácií** − ???  Poznámky: 14 |
|
| **Vlastnosti softvéru**  − vyžadujú sa akonáhle sa softvér začne používať. Nejde o služby (funkcie), ktoré softvér zabezpečuje (vykonáva).  **Správnosť:** miera, do akej výrobok spĺňa špecifikáciu.  Správnos  ť    Spo  ľ  ahlivos  ť    *zmena*  *výrobku*  *použitie*  *výrobku*  prenos  výrobku  Udržovate  ľ  nos  ť  Pružnos  ť  Testovate  ľ  nos  ť  Prenosnos  ť    os  ť    Znovupoužite  ľ  n  Interoperabilita  Efektívnosť  Použiteľnosť  Bezpečnosť  **Spoľahlivosť:** správanie sa výrobku pri výpadku – výrobok by nemal pri  výpadku systému spôsobiť ani fyzické ani ekonomické škody.  **Efektívnosť:** splnenie kritérií na využitie zdrojov počítačového systému, na čas potrebný na realizáciu a ďalších kritérií spojených so samotným vývojom (napr. náklady).  **Použiteľnosť:** úsilie, ktoré treba vynaložiť na to, aby sa dal výrobok používať.  **Bezpečnosť:** miera odolnosti voči neoprávneným zásahom do systému.  **Prenosnosť:** úsilie, ktoré treba na prenos výrobku z jednej platformy na inú (prostredie, v ktorom sa prevádzkuje).  **Znovupoužiteľnosť:** miera, do akej možno jednotlivé časti výrobku znovu použiť iných podobných aplikáciách. | | **Interoperabilita:** úsilie, ktoré treba na zabezpečenie spolupráce systému s inými systémami.  **Udržovateľnosť:** úsilie, ktoré treba vynaložiť na ďalší vývoj a údržbu výrobku podľa meniacich sa potrieb zákazníka a aj meniaceho sa okolia.  **Pružnos ť, modifikovateľnosť:** úsilie, ktoré treba na modifikáciu výrobku v  prevádzke (napr. zvýšenie jeho funkcionality).  **Testovateľnosť:** úsilie, ktoré treba vynaložiť na testovanie vlastností  výrobku, napr. či vykazuje *Vzťah medzi nákladmi na vývoj* požadované správanie. *softvéru a výkonnosťou softvéru*  **Dokumentovateľnosť:** miera, do akej  sú všetky rozhodnutia počas vývoja zdokumentované a kontinuita dokumentácie počas všetkých etáp.  *Vylučovanie sa niektorých vlastností* výkonnosť  náklady  *navzájom*  **Problémy s tvorbou softvéru Podstatné, vnútorné problémy**  − **zložitosť**  − **prispôsobivosť**  − **nestálosť**  − **neviditeľnosť** (*syndróm 90% hotovo*)      **Nie zákonité problémy** − špecifikácia požiadaviek  • komunikácia s používateľom  Poznámky: 16 |
| Poznámky: 15 | |
| −  −  −  −  −  − | * nejasná a neúplná formulácia požiadaviek, neucelená predstava používateľa o výslednom softvérovom systéme * nejednoznačnosť spojená s častou špecifikáciou požiadaviek v prirodzenom jazyku * nestálosť, protirečivosť požiadaviek * prirodzená neúplnosť a nepresnosť pri špecifikácii veľkých softvérových systémov * nedostatok znalostí z analyzovanej oblasti (problémy s plánovaním projektu) * problémy s testovaním a verifikáciou špecifikácií **programátorská produktivita** (extrémne individuálne odchýlky, až 1:20) **slabá opakovateľnosť v tvorbe softvéru** (málo štandardizácie, väčšinou sa softvér tvorí vždy od začiatku; málo produktov sa zostavuje z už existujúcich súčiastok) náchylnosť softvéru na chyby absencia “výroby” softvéru   **práca v tíme** (organizácia práce v tíme pri veľkých softvérových projektoch)   * komunikačné problémy * plánovanie procesu tvorby softvéru **tvorba dokumentácie** (vs. samotný proces tvorby programu) * enormné množstvo dokumentácie čo do kvantity aj rozmanitosti (napr. vo veľkých vojenských softvérových projektoch bolo vytvorených 400 anglických slov na každý príkaz v programovacom jazyku Ada) * problémy s udržiavaním dokumentácie (meniaca sa programová zložka softvéru) * problémy s konzistentnosťou a úplnosťou dokumentácie   Poznámky: 17 | Problémy s dokumentáciou sa najviac prejavujú pri prevádzke softvéru, keď pri údržbe softvéru sú nevyhnutné zmeny softvérového výrobku.  − **mnohé chyby, nedostatky sa objavujú až v prevádzke** (návrat k predchádzajúcim etapám vývoja softvéru)  − **problém mierky**  − **spôsob „starnutia softvéru“**  • stála akumulácia prídavnej funkcionality, časté opravy chýb → degradácia štruktúry a zníženie spoľahlivosti softvérových systémov s časom  *chyb*  *y*  skuto  č  ná krivka  zmena  ideálna  krivka  •  softvér sa fyzicky neopotrebuje  *chyb*  *y*  **hardvér** *čas* **softvér** *čas*  Príčiny zastavenia softvérových projektov  Dôležité faktory úspechu softvérových projektov  Poznámky: 18 |
|

Poznámky:

19

**Problémy s tvorbou softvéru – symptómy, prí**

**č**

**iny, riešenia**

**Na okraj…**

Ako vždy, motiváciou sú peniaze:

−

Ako by mohli softvéroví inžinieri vyvinú

ť

viac lepšieho a za menej

pe

ň

azí?

*Vízia:*

−

Budeme schopní vyvíja

ť

softvér spájaním hotových komponentov: rýchlo,

spo

ľ

ahlivo a lacno (product lines).

Symptómy

potreby používate

ľ

a

meniace sa

požiadavky

nepasujúce moduly

ť

ažká údržba

neskoré objavenie

slabá kvalita

slabá výkonnos

ť

tímová práca

Prí

č

iny

nedostato

č

ná analýza

požiadaviek

nejednozna

č

ná

komunikácia

krehké architektúry

ve

ľ

ká zložitos

ť

neodhalená

nekonzistentos

ť

slabé testovanie

subjektívne hodnotenie

vodopádový vývoj

neriadená zmena

nedostato

č

ná

automatizácia

Riešenie

iteratívny vývoj

riadenie

požiadaviek

komponentové

architektúry

vizuálne

modelovanie

zabezpe

č

enie

kvality

riadenie zmien

Poznámky:

20

**Proces vývoja softvéru**

Vývoj softvéru – proces, v kto

rom sa potreby používate

ľ

a

transformujú na požiadavky na

softvér, tieto sa transformujú na

návrh, návrh sa implementuje

v príslušnom programovacom

jazyku, tento sa testuje a odov-

zdá používate

ľ

ovi.

**Je vývoj softvéru umenie**

**alebo inžinierstvo?**

**Proces definuje**

−

kto robí

−

č

o,

−

kedy a

−

ako dosiahneme ur

č

itý cie

ľ

.

Nové alebo

zmenené požiadavky

Nový alebo

zmenený systém

**Softvérový**

**proces**

Definícia

problému

Technický

vývoj

Integrácia

riešenia

Status

quo

RIEŠENIE PROBLÉMU

|  |  |
| --- | --- |
| **Riešenie problémov**    **proces analýzy**  **proces syntézy**  **Životný cyklus softvéru**  − rozdelenie (dekompozícia) zložitejšieho problému na jednoduchšie, ľahšie zvládnuteľné problémy  − rozdiely v existujúcich prístupoch  Poznámky: 21 | − podstatnou charakteristikou každého modelu životného cyklu je, že   * **definuje jednotlivé etapy**a * pre každú z nich **činnosti**, ktoré sa majú vykonať, * rovnako ako vstupy a výstupy etapy   **Činnosti priamo spojené s vývojom softvéru** − ?? −  −  −  **Pravidlo 40 – 20 – 40, 60 – 15 – 25**  pred implementáciou  implementácia  po implementácii    **Systémové inžinierstvo a softvérové inžinierstvo**  Analýza  Implementácia  Návrh  Systémové  inžinierstvo  Biznis  analýza  Poznámky: 22 |
|

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model životného cyklu**  − definuje jednotlivé činno  − definuje č asovú následnos  − **ne**definuje dĺžku trvania  − návrat k predchádzajúcim krokom  − odporúčania  − každá etapa dobre defin  − každá etapa vytvára „h m  − správnosť  **Znáz**  Riadenie | | | | | **softvéru**  sti (kroky), ktoré t r ť krokov  krokov a ich rozs  ovaná  atateľné“ výstupy  každej etapy možno vyhodnotiť **ornenie etáp**  Infor | | | | eba vykonať ah    mácie pre  riadenie projektov  Výstupná podmienka | | | **V-model vývoja softvéru** − aspekt následnosti  − aspekt abstrakcie  − návratu k niektorej z predchádzajúcich etáp  otestovaný  systém  dokument  špecifikácie  požiadaviek  Analýza a  špecifikácia  p  ožiadaviek  A  k  c  e  p  t  a  č  n  é      testovanie a  inštalácia | | | | | | | | | |
| Poznámky: | |  | | --- | | Architektonický návrh | | |  | | | Integrácia a testovanie systému | | |  |
| dokument návrhu architektúry | | odladené súčiastky | | |
| Vstupná podmienka  **Model vytvor a oprav** | | | | Kr ok V&V | | |  | |
|  | | |
|  | Podrobn ý návrh | |  | Testovanie súčiastok | | |  |
|  | | | | | | | | | | | |  |
|  | Vytvor prvú verziu | | Modifikuj d ovtedy, kým | | | | | | | | | dokument návrhu súčiastok | | implementované súčiastky | | |
| Poznámky: |  |
|  | Implementácia | | | |  |
| zákazník nie je spokojný | | | | | | |  | |
|  | |  |  | |  | |  | | | |
|  | |
| Prevádzka | | | | |  |
|  | | | | |
| 23 | | | | | | | | | | 24 | | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Analýza a špecifikácia požiadaviek**  − transformácia neformálnych požiadaviek používateľa do štruktúrovaného opisu týchto požiadaviek  − zdôraznenie **čo**chce používateľ a nie *ako* to možno zabezpečiť (realizovať)  − vykonanie štúdie vhodnosti  − získavanie, analýza, definovanie a špecifikácia požiadaviek  − plánovanie akceptačného testovania **Architektonický návrh**  − ujasňuje sa celková koncepcia systému  − návrh dekompozície systému  − určenie vzťahov medzi časťami systému  − špecifikácia funkcionality a ohraničení pre každý podsystém  − plánovanie testovania systému  − plánovanie nasadzovania systému do prevádzky, dohoda o postupe nasadzovania podsystémov, dohoda o pláne zaškoľovania používateľov **Podrobný návrh**  − podrobná špecifikácia softvérových súčiastok  − špecifikácia algoritmov realizujúcich požadované funkcie  − špecifikácia rozhraní pre jednotlivé súčiastky  − špecifikácia logickej a fyzickej štruktúry údajov, ktoré spracúva príslušná súčiastka  − špecifikácia spôsobu ošetrovania chybových a neočakávaných stavov − plán prác pri implementácii súčiastky  − plán testovania súčiastky, návrh testovacích údajov  − špecifikácia potrebných ľudských zdrojov (odhad dĺžky trvania a nákladov)  Poznámky: 25 | | | **Implementácia a testovanie súčiastok**  − programová realizácia softvérových súčiastok  − vypracovanie dokumentácie k súčiastkam  − testovanie implementovaných súčiastok  − začiatok školení budúcich používateľov **Integrácia a testovanie systému**  − spájanie súčiastok do podsystémov a systému  − testovanie podsystémov a celého systému  − integrácia podsystémov a systému  − testovanie podsystémov a systému (oprava nájdených chýb, časté návraty k etape implementácie) **Akceptačné testovanie a inštalácia** − testovanie systému používateľom  − preberacie konanie  − školenia použitia systému a nasadenie systému **Prevádzka a údržba**  − zabezpečenie prevádzky softvéru  − riešenie problémov s používaním softvéru  − oprava, rozširovanie, prispôsobovanie softvéru podľa požiadaviek okolia  **Známe modely životného cyklu softvéru**  **Vodopádový model**  − nasledujúca etapa začne až po skončení predchádzajúcej **Inkrementálny (prírastkový) model**  − systém sa vytvára a odovzdáva používateľovi po častiach (časti sa stanovujú na základe špecifikácie celého systému)  Poznámky: 26 | | |
|
| **Iteratívny model**  − systém sa vyvíja v iteráciách  − v každej iterácii sa vytvorí vykonateľný výsledok  Iterácia 1 Iterácia 2 Iterácia 3  **T**  **I**  **N**  **Š**    **T**  **I**  **N**  **Š**    **T**  **I**  **N**  **Š**  Č AS **Evolučný model**  − požiadavky sa všetky nedefinujú na začiatku, systém sa vyvíja po častiach vo viacerých verziách **Špirálový model, WinWin model**  − uvažuje aj aspekty manažmentu  − definuje množinu aktivít, ktoré smerujú k „výhre používateľa“ získaním produktu, ktorý zodpovedá jeho požiadavkám a „výhre vývojára“ vytvorením výsledku s reálnym rozpočtom a rozvrhom  **Komponentový model**  − využitie znovupoužiteľných súčiastok  **Formálna transformácia**  − vyžaduje formálnu (matematickú) špecifikáciu systému  **Agilné metódy tvorby softvéru**  − eXtreme Programming, Feature Driven Development, Lean Programming, Crystal  Poznámky: 27 | | | **Špecialisti v životnom cykle softvéru**  **Rôzne skupiny zákazníkov**  Zákazník  Dodávate  ľ  Používate  ľ    potreby  *p*  *oužíva*  *systém*  *sponzoruje vývoj*  *systému*  *vyvíja*  *systém*  záväzok  softvérový systém  potreby,  peniaze  − podľa činnosti (lokálny a globálny pohľad):   * lokálny pohľad, zaujíma sa o funkčnú stránku, zaujíma ho fyzický pohľad na systém * nemusí mať lokálny pohľad, pozná výkonnú stránku, zaujíma sa o rozpočet, sprostredkovateľ medzi manažérmi * globálny pohľad, inicializácia projektu, nemá priame skúsenosti s výkonnou stránkou, strategické ciele   − podľa úrovne skúseností s výpočtovou technikou **Úlohy v softvérovom tíme** − ???  − ?  **Ako trávia čas programátori?**  Písanie programov 13 %  Čítanie programov a príručiek 6 % Komunikácia týkajúca sa práce 42 % Ostatné (vrátane osobných) 39 % | | |
| Poznámky: 28 | | |
| **Softvérové procesy**  **Softvérový proces**  − určuje abstraktnú množinu činností, ktoré sa majú vykonať pri vývoji softvérového výrobku z pôvodných požiadaviek používateľa. **Softvérový projekt**  − vykonanie týchto činností pre špecifické požiadavky používateľa  − konkretizuje činnosti a poradie definované procesom (projektový plán) − časovo ohraničené úsilie, ktoré sa vyvíja s cieľom vytvorenia jedinečného výsledku; množina činností, technických aj riadiacich, ktoré sa požadujú na zabezpečenie podmienok *projektovej dohody.*  Projekt  1    Výsledok  n    Výsledok  2    Projekt  n    Výsledok  1    Softvérový proces  Projekt  j      **Vyspelosť softvérového procesu – CMM**    Poznámky: 29 | | | **Klasifikácia softvérových procesov**  Procesy vývoja  softvéru  Procesy tvorby softvéru  Procesy manažmentu  softvéru    **Softvérový proces**  Procesy manažmentu  procesu  Procesy manažmentu  projektu  **Proces vývoja softvéru**  − činnosti priamo spojené s vývojom softvéru − špecifikácia softvéru, realizácia softvéru, validácia softvéru a evolúcia softvéru. **Procesy manažmentu softvéru**  (označuje sa aj manažment softvérových konfigurácií)  − riadenie zmien softvérového systému, identifikácia jednotlivých verzií a konfigurácií počas jeho celého životného cyklu. **Procesy manažmentu projektu**  − použitie znalostí, zručností, prostriedkov a techník na projektové činnosti s cieľom dosiahnutia (alebo prekročenia) potrieb a očakávaní projektu. **Procesy manažmentu procesu**  − zlepšovanie procesu = porozumenie existujúcim procesom a ich zmena tak, že sa zlepšia vlastnosti softvéru a/alebo sa redukujú náklady a čas na vývoj (kvalita).      Poznámky: 30 | | |
|
| **Etický kódex softvérového inžiniera**  Výňatok z etického kódexu), plné znenie nájdete na http://computer.org/tab/SWECP9912.htm  **Princíp 1: Verejnosť**  Softvéroví inžinieri by mali konať v zhode so všeobecnými verejnými záujmami. Obzvlášť by mali:  − Prijať plnú zodpovednosť za vlastnú prácu.  − Schváliť softvérový produkt iba v prípade, ak sú odôvodnene presvedčení o tom, že produkt je bezpečný, spĺňa špecifikačné požiadavky, neznižuje kvalitu života, neohrozuje súkromie a životné prostredie a úspešne vyhovel príslušným testom.  − Prezentovať vhodnej osobe alebo autorite každé existujúce alebo potenciálne ohrozenie používateľa, verejnosti alebo životného prostredia, o ktorom sú odôvodnene presvedčení, že je spôsobené alebo spojené so softvérom alebo príslušnými dokumentmi.  − Byť povzbudení dobrovoľne poskytovať svoje profesionálne znalosti na dobré ciele a prispievať k zvyšovaniu úrovne znalostí verejnosti v oblasti softvérového inžinierstva.  − ....  **Princíp 2: Zákazník a zamestnávateľ**  Softvéroví inžinieri by mali jednať v najlepších záujmoch svojho zákazníka a zamestnávateľa, v zhode s verejným záujmom. Obzvlášť by mali:  − Vedome nepoužívať softvér, ktorý je získaný alebo prechovávaný nelegálne alebo neeticky.  − Identifikovať, dokumentovať, zhromažďovať údaje a oznámiť ich zákazníkom alebo zamestnávateľom okamžite, keď by podľa ich názoru  Poznámky: 31 | | | mohol byť projekt neúspešný, preukázateľne drahý, mohol by porušovať zákon duševného vlastníctva alebo byť ináč problematický.  − Neakceptovať žiadnu vedľajšiu prácu, ktorá by mala škodlivý vplyv na prácu, ktorú uskutočňujú pre svojho hlavného zamestnávateľa.  − Nepodporovať žiadne záujmy zamerané proti svojmu zamestnávateľovi alebo zákazníkovi, iba ak je to v zhode s vyšším etickým záujmom. V takom prípade by o ňom mali informovať zamestnávateľa alebo inú vhodnú autoritu.  − ....  **Princíp 3: Výrobok**  Softvéroví inžinieri by mali zaistiť, aby ich produkty a súvisiace modifikácie spĺňali najvyšší možný profesionálny štandard. Obzvlášť by mali:  − Usilovať o vysokú kvalitu, akceptovateľné náklady a zmysluplný plán, zabezpečiac, aby významné kompromisy boli jasné a akceptované zamestnávateľom aj klientom a prístupné stanovisku používateľa a verejnosti.  − Zabezpečiť svoju kvalifikáciu na každom projekte, na ktorom pracujú alebo sa spolupodieľajú, primeranou kombináciou vzdelania, školení a skúseností.  − Dodržiavať pri aktuálnej práci primerané profesionálne štandardy vždy, keď je to možné, odchyľujúc sa od nich iba v eticky a technicky oprávnených prípadoch.  − Zabezpečiť adekvátne testovanie a kontrolu softvéru a prislúchajúcich dokumentov, na ktorých pracujú.  − Zabezpečiť adekvátnu dokumentáciu, obsahujúcu objavené dôležité problémy a použité riešenia pre každý projekt, na ktorom pracujú.  − Prácou vytvárať softvér a prislúchajúcu dokumentáciu, ktorá rešpektuje súkromie tých, ktorí budú ovplyvnení daným softvérom.  Poznámky: 32 | | |
|
| − Pristupovať ku všetkým formám softvérovej údržby s rovnakou profesionalitou ako k novému vývoju.  − ...  **Princíp 4: Posudzovanie**  Softvéroví inžinieri by mali zachovávať nezávislosť vo svojom profesionálnom posudzovaní. Obzvlášť by mali:  − Dodržovať profesionálnu objektivitu so zreteľom na akékoľvek softvérové alebo súvisiace dokumenty pri požiadavke o ich zhodnotenie.  − Informovať všetky zúčastnené strany, ktorých konfliktom záujmov nemožno primerane zabrániť alebo sa im vyhnúť.  − ...  **Princíp 5: Manažment**  Manažéri a lídri softvérového inžinierstva by mali podporovať a propagovať etický prístup k manažmentu softvérového vývoja a údržby. Manažéri a vedúci softvéroví inžinieri by obzvlášť mali:  − Zabezpečiť dobrý manažment pre všetky projekty, na ktorých pracujú, vrátane efektívnych procedúr na podporu kvality a znižovania rizika.  − Pritiahnuť potenciálnych softvérových inžinierov iba úplným a presným opisom podmienok zamestnania.  − Umožniť riadny proces pri konaní o porušení koncepcie zamestnávateľa alebo tohto Kódexu.  − Nikoho netrestať za vyjadrenie etických pripomienok k určitému projektu. − ...  **Princíp 6: Profesia**  Softvéroví inžinieri by mali zlepšovať bezúhonnosť a povesť profesie v súlade s verejným záujmom. Obzvlášť by mali:  − Zvyšovať povedomie verejnosti o profesii softvérového inžiniera.  Poznámky: 33 | | | − Rozširovať znalosť softvérového inžinierstva vhodnou účasťou v profesijných organizáciách, stretnutiach a publikáciách.  − Preberať zodpovednosť za hľadanie a opravu chýb, za vytváranie správ o chybách v softvéri a v príslušných dokumentoch, na ktorých pracujú.  − Oznámiť podstatné porušenia tohoto kódexu príslušným autoritám, keď je jasné, že konzultácie s dotknutými osobami sú nemožné, kontraproduktívne alebo nebezpečné.  − ...  **Princíp 7: Kolegovia**  Softvéroví inžinieri by mali byť spravodliví ku svojím kolegom a mali by ich podporovať. Obzvlášť by mali:  − Napomáhať kolegom v profesionálnom raste.  − Plne dôverovať práci ostatných a zdržať sa pripisovania nadmerných zásluh.  − Posudzovať prácu iných objektívne, poctivo a náležite zdokumentovaným spôsobom.  − Napomáhať kolegom k uvedomeniu si súčasných štandardných praktík vrátane bezpečnostnej politiky a procedúr na ochranu hesiel, súborov a ostatných citlivých informácií a bezpečnostné opatrenia všeobecne.  − V situáciách mimo svojho poľa pôsobenia požiadať o pomoc odborníka, ktorý má kompetencie v týchto oblastiach.  − …  **Princíp 8: Osobnosť**  Softvéroví inžinieri by sa mali zúčastňovať celoživotného vzdelávania týkajúceho sa ich profesnej praxe a mali by presadzovať etický prístup v profesnej praxi. Obzvlášť by sa mali softvéroví inžinieri usilovať:  − Zlepšovať svoju schopnosť vytvárať bezpečný, spoľahlivý a prospešný vysoko kvalitný softvér za rozumnú cenu a v rozumnom čase.  Poznámky: 34 | | |
|
|  | | |  | | |
| − | Rozširovať  a testovania softvéru a s procesu vývoja softvéru. | svoje znalosti analýzy, špecifikácie, návrhu, vývoja, údržby úvisiacich dokumentov spolu s manažmentom |  | **Na záver...** |  |
| **Fred Brooks: „How does a project really get into trouble?“**  **„One slip at a time.“**  − nestačí zamestnať najlepších programátorov  − ani použiť najnovšie metódy, či technológie − ani zapojiť používateľa v max. miere − ani najať najlepšieho manažéra...  Treba urobiť toto všetko. A dokonca viac...  **Softvérové inžinierstvo a ostatné inžinierske disciplíny** − „nehmotná“ povaha softvéru  − nemožnosť uvažovania všetkých podmienok, alternatív  − cieľom je minimalizácia škody pri neočakávaných podmienkach **Aspekty tvorby softvéru** − technický aspekt  − psychologický aspekt  − aspekt riadenia (manažmentu) projektu **Čo je pravda?**   1. Na všetky činnosti pri tvorbe softvéru dnes existujú štandardy. 2. Najdôležitejšie je mať najnovšie CASE prostriedky a iné nástroje spolu s najnovšou výpočtovou technikou. 3. Keď sa oneskorujeme v softvérovom projekte v pláne, treba pridať viac programátorov do projektu, aby sme stratu vyrovnali. 4. Požiadavky na softvér sa neustále menia, ale keďže softvér je pružný (a nehmotný) možno ich jednoducho zapracovať.   Poznámky: 36 |
| − Zlepšovať svoju schopn osť vytvoriť presnú, informatívnu a dobre napísanú dokumentáciu. | | |
| − Zlepšovať svoje porozumenie softvéru a príslušnej dokumentácie, na ktorej pracujú a prostredia, v ktorom sa použijú.  − Zlepšovať svoju znalosť tohto Kódexu, jeho interpretácie a jeho aplikácie v praxi.  − Nezaobchádzať nespravodlivo s nikým z dôvodu akýchkoľvek irelevantných predsudkov.  − Nenabádať iných na vykonanie činov zahŕňajúcich porušenie tohto Kódexu.  − Poznať, že osobné porušenie tohto Kódexu je nezlúčiteľné s postavením softvérového inžiniera. | | |
| Poznámky: 35 | | |

7

.

Výstup úspešného softvérového projektu je program.

Poznámky:

6

.

P

o

k

i

a

ľ

n

e

m

á

m

v

y

k

o

n

a

t

e

ľ

ný program, nemôžem o jeho kvalite ni

č

p

o

v

e

d

a

ť

.

5

.

K

e

ď

v

y

t

v

o

r

í

m

e

p

r

o

g

r

a

m

,

k

t

o

r

ý

f

u

n

g

u

j

e

–

p

r

á

c

a

s

o

f

t

v

é

r

o

v

é

h

o

i

n

ž

i

n

i

e

r

a

skon

č

ila.

37

**,**

**1**

**5-**

**3**

**X**

**1**

**X**

Špecifikácia

N

á

v

r

h

I

m

p

l

e

m

e

nt

áci

a

P

r

ev

ádz

ka

Et

apy

**60-**

**X**

**100**

**2-**

**6**

**X**

Náklady na zmenu