

1. Uvod v programsko inženirstvo

1.1 Uvod

1.1.1 Programsko inženirstvo in izdelava programske opreme

Ekonomije so odvisne od programske opreme, vse več sistemov za krmiljenje uporablja programsko opremo, odhodki za programsko opremo predstavljajo znaten delež BDP vseh razvitih držav.

1.1.2 Stroški programske opreme

Ponavadi velja:

- stroški programske opreme > stroški računalniške opreme
- stroški vzdrževanja p.o > stroški razvoja p.o

1.1.3 Neuspešnost projektov izdelave programske opreme

Projekti ponavadi ne uspejo zaradi **povečane kompleksnosti sistemov** ali pa **neuporabe metod programskega inženirstva**.

1.2 Profesionalno programsko inženirstvo

1.2.1 Pogosta vprašanja o programskem inženirstvu

Programska oprema = računalniški programi + ustrezna dokumentacija

Lastnosti kakovostne prog. opreme:

- omogoča vse funkcionalne in nefunkcionalne zahteve
- vzdržljiva
- zanesljiva
- uporabna

Programsko inženirstvo = disciplina inženirstva, ki se ukvarja z vsemi vidiki izdelave programske opreme.

Razlika med programskim in sistemskim inženirstvom: sistemsko inženirstvo se ukvarja z vsemi vidiki izdelave **računalniško podprtih sistemov**, vključno s strojno in programsko opremo ter procesnim inženiringom. Programsko inženirstvo je del tega splošnega procesa.

Ključni izzivi programskega inženirstva:

- spopadanje z naraščajočo raznolikostjo
- zmanjševanje časa dostave
- razvoj zanesljive prog. opreme

Stroški programskega inženirstva = 60% stroški razvoja + 40% stroški testiranja. Evolucijski stroški pogosto presegajo stroške razvoja.

Najboljše tehnike in metode programskega inženirstva: za različne zahteve so najboljše različne tehnike in metode (*razvoj iger / razvoj varnostnega sistema / ...*).

1.2.2 Izdelki programske opreme

1. generični izdelki: samostojne programske rešitve, ki so na voljo vsem na trgu in rešujejo splošne vsakdanje probleme (*urejevalniki besedila, urejevalniki slik, programska oprema za specifičen trg - naročanje bolnikov na pregled*) - **za specifikaijo je odgovoren razvijalec**

2. prilagojeni izdelki: programska oprema, ki je izdelana po naročilu za specifičen namen (*vgrajeni krmilni sistemi, sistem za spremljanje prometa, sistem za nadzor letalskega prometa ...*) - **za specifikacijo je odgovorna stranka**

1.2.4 Ključne lastnosti kakovostne programske opreme

1. **vzdrževalnost**: izbran način izdelave mora omogočati prilagajanje spremembam uporabnikovih zahtev. Ker so spremembe neizogibna zahteva, je ta lastnost najbolj pomembna.
2. **zagotovljivost in varnost**: v primeru izpada programska oprema ne sme povzročati fizične/gospodarske škode. Zlonamerni uporabniki ne smejo imeti možnosti poškodovati sistema.
3. **učinkovitost**: prog. oprema ne sme povzročati neučinkovite porabe fizičnih virov. Učinkovitost zajema stvari kot so čas obdelave datoteke, čas procesiranja ...
4. **sprejemljivost**: uporaba prog. opreme mora biti razumljiva, uporabna in združljiva z ostalimi sistemi, ki jih uporabniki uporabljajo (*ko delamo app, ga razvijemo tako, da bo delal na Windows/Linux*).

1.2.7 Aktivnosti izdelave programske opreme

1. **specifikacija** - inženirji in stranka opredelijo programsko opremo, ki jo je treba izdelati
2. **razvoj** - prog. opremo je potrebno načrtovati in izdelati
3. **vrednotenje** - preverba, kjer določimo ali je prog. oprema primerna za končne uporabnike (testiranje)
4. **evolucija** - prog. opremo nadgradimo/spremnimo glede na spreminjajoče se zahteve stranke/končnih uporabnikov

1.2.8 Splošne težave, ki vplivajo na prog. opremo

1. **heterogenost** - prog. oprema mora delovati na porazdeljenem sistemu, kjer so prisotne različne vrste računalniških/mobilnih naprav
2. **poslovne in družbene spremembe** - podjetja in družbe se zelo hitro spreminjajo s pojavom in dostopom do novih tehnologij
3. **varnost in zaupanje** - glede na to da prog. oprema vpliva na velik del našega življenja je dobro, da ji lahko zaupamo
4. **obseg** - prog. opremo je potrebno razvijati v minimalnih obsegih, kjer gre lahko le za en vgrajen sistem, lahko pa jo razvijamo na globalni ravni kjer jo bo uporabljalo več uporabnikov na svetovnem nivoju

1.2.10 Vrste programske opreme

- **samostojne aplikacije** - aplikacije, ki se izvajajo na lokalnem računalniku
- **interaktivne transakcijske aplikacije** - se izvajajo na oddaljenem računalniku in uporabniki do njih dostopajo s svojimi osebnimi računalniki oz. terminali (spletne aplikacije/spletne trgovine...)
- **vgrajeni krmilni sistemi** - prog. oprema za nadzor in upravljanje strojnih naprav
- **sistemi za paketno obdelavo** - poslovni sistemi, namenjeni za obdelavo paketov v velikih serijah
- **sistemi za zabavo** - namenjeni uporabni in zabavi končnega uporabnika (igre)
- **sistemi za modeliranje in simulacijo** - razvijajo jih znanstveniki za potrebe znanosti
- **sistemi za zbiranje podatkov** - zbirajo podatke iz svojega okolja z uporabo senzorjev ter jih pošiljajo drugim sistemom za obdelavo
- **sistemi sistemov** - sestavljajo jih številni drugi programski sistemi

1.2.11 Osnove programskega inženiringa:

- uporaba upravljanega in razumljivega razvojnega procesa
- zagotovljivost in zmogljivost
- razumevanje in upravljanje specifikacij
- ponovna uporaba obstoječih delov programske opreme

1.2.12 Programsko inženirstvo na spletu

Z razvojem spletnih storitev se aplikacije vedno bolj selijo iz lokalnih na oddaljene. Spletne storitve omogočajo dostop do funkcionalnosti aplikacij preko spleta.

1.2.13 Spletno programsko inženirstvo

- spletne sisteme je potrebno razvijati z inkrementalnim oz. agilnim pristopom, kjer je nepraktično dolčati vse zahteve že vnaprej
- pojav tehnologij AJAX in HTML5

1.3 Etika programskega inženirstva

1.3.1 Poklicna odgovornost

- **zaupnost** - spoštovati moramo zaupnost delodajalcev, če mamo to napisano na papirju ali pa ne
- **osnovna zmožnost oz. kompetenca** - ne smemo napačno predstavljati svojih zmožnosti in ne sprejemamo dela, ki je zunaj naše pristojnosti
- **pravice intelektualne lastnine** - zavedati se je treba zakonov, ki se nanašajo na pravice uporabnikov, lastnine itd.
- **napačna raba računalniških virov** - programski inženirji ne smejo uporabljati svojih tehničnih znajn v zlonamerne namene

1.3.2 Etični kodeks ACM/IEEE

Načela kodeksa:

1. Javnost - prog. inženirji morajo delovati v načelu z javnim interesom
2. Naročnik in delodajalec - prog. inženir deluje v interesu stranke in delodajalca ter v skladu z javnim interesom
3. Izdelek - prog. inženir mora zagotoviti, da njihov izdelek ustreza najvičjim strokovnim standardom
4. Presoja - prog. inženir mora v svoji strokovni presoji ohraniti integriteto in neodvisnost
5. Upravljanje - Vodja razvojne ekipe mora spodbujati etični pristop k upravljanju projekta
6. Poklic - prog. inženir mora skrbeti za razvoj integritete in ugleda poklica v skladu z javnim interesom
7. Sodelavci - prog. inženir mora svoje sodelavce spodbujati in jim biti v oporo
8. Lastna osebnost - prog. inženir sodeluje v vseživljenskem učenju in spodbuja etični pristop k opravljanju poklica