P11 Evolucija programske opreme

1 Uvod

1.1 Sprememba programske opreme

Sprememba programske opreme **je neizogibna** zaradi številnih razlogov:

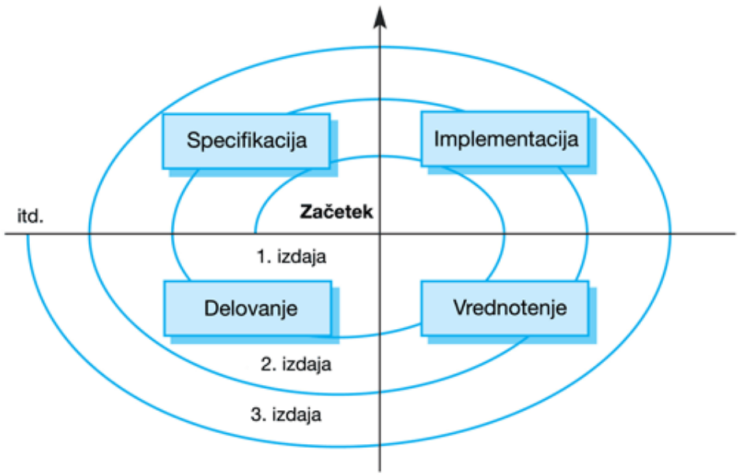
- pojavljanje novih zahtev
- spreminjanje poslovnega okolja
- potreba po popravilu napak
- sistemu se dodajajo novi računalniki in oprema
- potreba po izboljšanju kakovosti

Izvajanje in **upravljajnje sprememb obstoječih sistemov** programske opreme je **ključna težava vseh podjetij**.

1.2 Pomen evolucije

Večina proračuna za programsko opremo v večjih podjetjih je **namenjena spreminjanju** in **prilagajanju obstoječe programske opreme** in NE razvoju nove programske opreme.

1.3 Evolucija in spiralni model

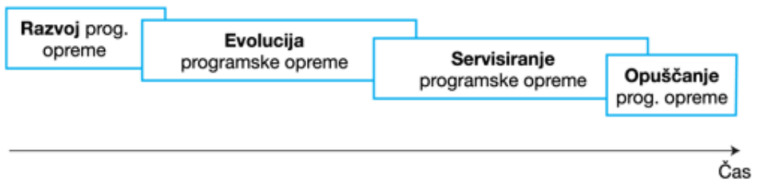


Slika 21.1: Spiralni model razvoja in evolucije

Figure 1: Spiralni model razvoja in evolucije

1.4 Evolucija in servisiranje programske opreme

- **Evolucija** programske opreme je stopnja življenskega cikla izdelave sistema programske opreme, kjer se sistem uporablja in razvija v skladu z novimi zahtevami.
- Na stopnji **servisiranja** je programska oprema še vedno uporabna, vendar so izvedene zgolj tiste spremembe, ki so potrebne, da sistem ostane operativen.
- Na stopnju **opuščanja** se programska oprema lahko še vedno uporablja, vendar ni prisotnih nobenih sprememb.



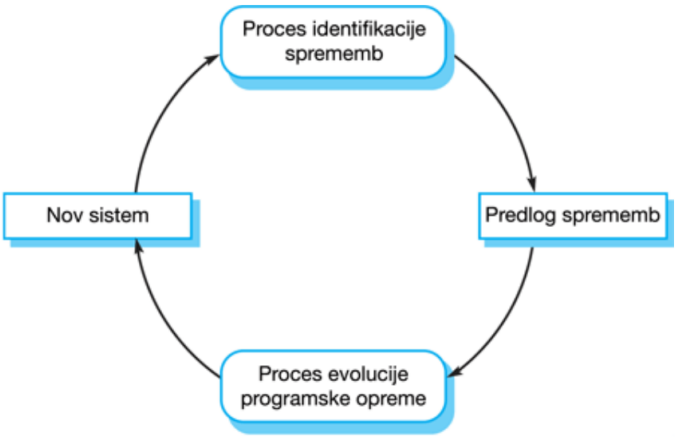
Slika 21.2: Evolucija in servisiranje programske opreme

Figure 2: Evolucija in servisiranje programske opreme

2 Evolucijski procesi

Evolucijski procesi programske opreme so odvisni od:

- vrste programske opreme
- uporabljenega razvojnega procesa
- spretnosti ter izkušenj vključenih akterjev



Slika 21.3: Proces identifikacije sprememb in evolucije programske opreme

Figure 3: Proces identifikacije sprememb in evolucije programske opreme

2.1 Implementacija sprememb

Implementacija sprememb predstavlja iteracijo razvojnega procesa, kjer se zasnuje, implementira in testira revizija sistema.

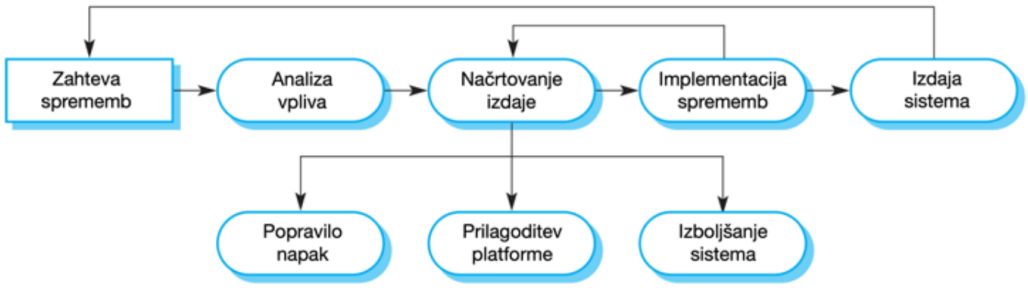
2.2 Nujni zahtevki sprememb

Spremembe se bodo absoltno morale izvesti če:

- je treba popraviti resno napako sistema, da se omogoči nadaljevanje normalnega razvoja
- imajo spremembe v okolju sistema nepričakovane učinke
- obstajajo poslovne spremembe, ki zahtevajo zelo hiter odziv (npr. izdaja konkurenčnega izdelka)

2.3 Evolucija in agilne metode

Agilne metode temeljijo na inkrementiranem razvoju, zato je prehod iz razvoja v evolucijo **enostaven**.



Slika 21.4: Splošen model evolucijskega procesa

Figure 4: Splošen model evolucijskega razvoja

Slika 21.5: Implementacija sprememb

Figure 5: Implementacija sprememb

Slika 21.6: Proces izvedbe nujnih popravkov

Figure 6: Proces izvedbe nujnih popravkov

Avtomatizirano regresijsko testiranje je pri spremembah v sistemu še posebej pomembno.

Spremembe se lahko izrazijo kot **dodatne uporabniške zgodbe**.

2.3.1 Težave pri predaji

Težave pri predaji lahko pridejo v naslednjih scenarijih:

- **razvojna ekipa** uporablja **agilni pristop**, medtem ko **ekipa za evolucijo** ne pozna agilnih metod in raje uporablja *pristop, ki temelji na načrtovanju*
- ali pa obratno

3 Podedovani sistemi

Podedovani sistemi so starejši sistemi, ki se zanašajo na jezike in tehnologije, ki se ne uporabjajo več za razvoj novih sistemov.

3.1 Komponente podedovanih sistemov

Slika 21.7: Elementi podedovanega sistema

Figure 7: Elementi podedovanega sistema

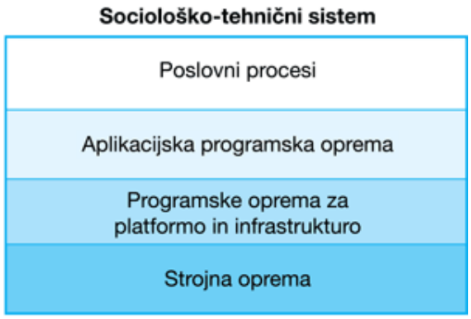
- **Sistemska strojna oprema**, kjer so podedovani sistemi morda razviti za strojno opremo, ki ni več na voljo.
- **Podporna programska oprema**, kjer so podedovani sistemi mogoče odvisni od vrste programske opreme, ki je zastarela oz. nepodprta.
- **Aplikacijska programska oprema** zagotavlja poslovne storitve, ki so običajno sestavljene iz številnih aplikacijskih programov.
- **Aplikacijski podatki**, ki jih obdeluje aplikacijski sistem in so lahko neskladni, podvojeni ali shranjeni v različnih podatkovnih bazah.
- **Poslovni procesi**, ki se uporabljajo v podjetju za doseganje določnih poslovnih ciljev.
- **Poslovne politike in pravila** so opredelitve, kako je treba opraviti poslovanje, in omejitve poslovanja.

3.2 Sloji podedovanega sistema

3.3 Zamenjava podedovanega sistema

Zamenjava podedovanega sistema je lahko **tvegana** iz več razlogov:

- pomankanje specifikacije celotnega sistema
- testna integracija sistemskih in poslovnih procesov
- poslovna pravila, vgrajena v obstoječi sistem, niso dokumentirana
- razvoj nove programske opreme zamuja ali presega prvotno določen proračun



Slika 21.8: Sloji podedovanih sistemov

Figure 8: Sloji podedovanega sistema

3.4 Sprememba podedovanega sistema

Spreminjanje podedovanih sistemov je zahtevno in drago:

- neskladnost sloga programiranja
- uporaba zastarelih programskih jezikov z malo razpoložljivimi razvijalci
- neustrezna dokumentacija sistema
- poslabšanje strukture sistema
- zaradi optimizacije programov jih je težje razumeti
- napake v podatkih, podvajanje in nedoslednost

3.5 Upravljanje podedovanega sistema

Podjetja, ki uporabljajo podedovane siteme, se morajo odločiti za **strategijo razvoja teh sistemov**:

- popolna opustitev
- nadaljevanje vzdrževanja
- prenova sistema
- zamenjava sistema

Izbrana strategija mora biti odvisna od kakovosti sistema in njegove poslovne vrednosti.

Slika 21.9: Primer ocene podedovanega sistema

Figure 9: Primer ocene podedovanega sistema

3.6 Ocenjevanje poslovne vrednosti

Pri **ocenjevanju poslovne vrednosti** je treba upošteva različna stališča:

- končnih uporabnikov sistema
- poslovne stranke
- vodstvenih delavcev
- vodij IT
- višjih vodstvenih delavcev

Pri ocenjevanju poslovne vrednosti se pojavijo **različna uprašanja**:

- **uporaba sistema**, kjer ima sistem nizko vrednost, če imajo *premalo* število uporabnikov
- **podprti poslovni procesi**, kjer ima sistem nizko vrednost, če uporabnike prisli k uporabi neučinkovitih poslovnih procesv
- **sistemska zanesljivost**, kjer ima sistem nizko vrednost, če ni zanesljiv in težave neposredno vplivajo na poslovne stranke
- **izhodi sistema**, kjer ima sistem visoko vrednost, če je poslovanje odvisno od njegovih rezultatov

3.7 Ocenjevanje kakovosti sistema

- **Ocena poslovnih procesov** - *kako dobro poslovni proces podpira trenutne cilje podjetja?*
- **Ocena okolja** - *kako učinkovito je okolje sistema in kako drago je njegovo vzdrževanje?*
- **Ocena uporabe aplikacije** - *kakšna je kakovost aplikacijskega sistema?*

3.7.1 Ocena poslovnih procesov

Uporabi se pristop, ki je osredotočen na **pogled akterjev** sistema:

- ali obstaja definiran procesni model in ali ga uporabljamo?
- ali različni deli organizacije uporabljajo različne procese za isto funkcijo?
- kako je bil proces prilagojen?
- kakšna so razmerja z drugimi poslovnimi procesi in ali so potrebna?
- ali proces uspešno podpira podedovano programsko opremo?

3.7.2 Ocena okolja

3.7.3 Ocena uporabe aplikacije

3.7.4 Merjenje sistema

Za ocenjevanje kakovosti aplikacije lahko zbiramo **kvantitativne podatke**:

- **število zahtevkov za spremembo sistema**
- **število različnih uporabniških vmesnikov**
- **obseg podatkov**

4 Vzdrževanje programske opreme

Vzdrževanje programske opreme je spreminjanje le-te po njeni izdaji.

Vzdževanje običajno **ne vključuje večjih sprememb v arhitekturi sistema**.

4.1 Vrste vzdrževanja

- **Popravki napak** predstavljajo spreminjanje sistema zaradi napak oz. raznljivosti in odpravljanje pomankljivosti.

Tabela 21.1: Dejavniki, prisotni pri oceni okolja

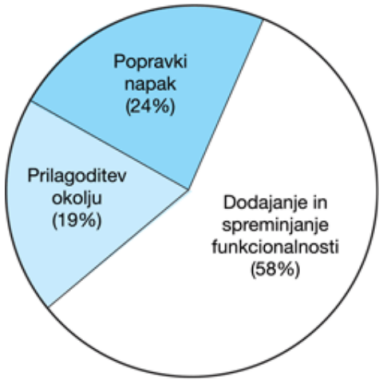
Dejavnik	Vprašanja
Stabilnost dobavitelja	Je dobavitelj še vedno na voljo? Ali je dobavitelj finančno stabilen in bo z večjo verjetnostjo nadaljeval s poslovanjem? Ali kdo drug vzdržuje sistem, če dobavitelj ne posluje več?
Stopnja napak	Ali ima strojna oprema visoko stopnjo prijavljenih napak? Ali se podporna programska oprema pogosto zruši in povzroči ponovni zagon sistema?
Starost	Koliko je stara strojna in programska oprema? Starejša kot je strojna in podporna programska oprema, bolj bo zastarela. Morda še vedno pravilno deluje, vendar lahko prehod na sodobnejši sistem prinese velike gospodarske in poslovne koristi.
Zmogljivost	Ali je zmogljivost sistema ustrezna? Ali imajo težave z zmogljivostjo pomemben vpliv na uporabnike sistema?
Zahteve za podporo	Kakšna lokalna podpora je potrebna za strojno in programsko opremo? Če so s to podporo povezani visoki stroški, morda velja razmisliti o zamenjavi sistema.
Stroški vzdrževanja	Kakšni so stroški vzdrževanja strojne opreme in licenc za vzdrževanje programske opreme? Starejša strojna oprema ima lahko višje stroške vzdrževanja kot sodobni sistemi. Podporna programska oprema ima lahko visoke letne stroške licenciranja.
Povezljivost	Ali obstajajo težave pri povezovanju sistema z drugimi sistemi? Ali se lahko prevajalniki npr. uporabljajo s trenutnimi različicami operacijskega sistema? Ali je potrebna emulacija strojne opreme?

Figure 10: Dejavniki, prisotni pri oceni okolja

Tabela 21.2: Dejavniki, prisotni pri oceni uporabe aplikacije

Dejavnik	Vprašanja
Razumljivost	Kako težko je razumeti izvorno kodo trenutnega sistema? Kako kompleksne so uporabljene krmilne strukture? Ali imajo spremenljivke smiselna imena, ki odražajo njihovo funkcijo?
Dokumentacija	Katera sistemska dokumentacija je na voljo? Ali je dokumentacija popolna, skladna in aktualna?
Podatki	Ali obstaja podatkovni model sistema? V kolikšni meri so podatki podvojeni? Ali so podatki, ki jih sistem uporablja, ažurni in skladni?
Zmogljivost	Ali je zmogljivost aplikacije ustrezna? Ali imajo težave z zmogljivostjo pomemben vpliv na uporabnike sistema?
Programski jezik	So za programski jezik, ki se uporablja za razvoj sistema, na voljo sodobni prevajalniki? Ali se programski jezik še vedno uporablja za razvoj novega sistema?
Upravljanje konfiguracij	Ali vse različice vseh delov sistema upravljajo sistemi za upravljanje konfiguracije? Ali obstaja natančen opis različic komponent, ki se uporabljajo v trenutnem sistemu?
Testni podatki	Ali obstajajo testni podatki za sistem? Ali so prisotni regresijski testi, ki so izvedeni ob dodajanju novih elementov sistema?
Usposobljenost osebja	Ali je na voljo osebje, ki je usposobljeno za vzdrževanje aplikacije? Ali je na voljo osebje, ki ima izkušnje s sistemom?

Figure 11: Dejavniki, prisotni pri oceni uporabe aplikacije



Slika 21.10: Porazdelitev napora pri vzdrževanju

Figure 12: Porazdelitev napora pri vzdrževanju

- **Prilagoditev okolju** je vzdrževanje za prilagoditev programske opreme različnim izvajalnim okoljem.
- **Dodajanje** in **spreminjanje funkcionalnosti** predstavlja spreminjanje sistema pri izpolnjevanju novih zahtev.

4.2 Stroški vzdrževanja

Stroški vzdrževanja so običajno **višji** od stroškov razvoja.

Običajno je **dražje dodati nove funkcije v sistem med vzdrževanjem** kot dodajanje enakih funckij med razvojem, kjer:

- nova ekipa mora razumeti, kateri programi se vzdržujejo
- ločevanje vzdrževanja in razvoja pomeni, da razvojna ekipa nima motivacije za pisanje programske opreme, ki jo je lažje vzdrževati
- vzdrževanje programov ni priljubljeno
- s starostjo programov se njihova stuktura poslabša in jih je težje spremeniti

4.3 Predvidevanje vzdrževanja

Predvidevanje oz. **napovedovanje vzdrževanja** se nanaša na izdelavo ocene, kateri deli sistema lahko povzročijo težave in imajo visoke stroške vzdrževanja.

4.3.1 Predvidevanje sprememb

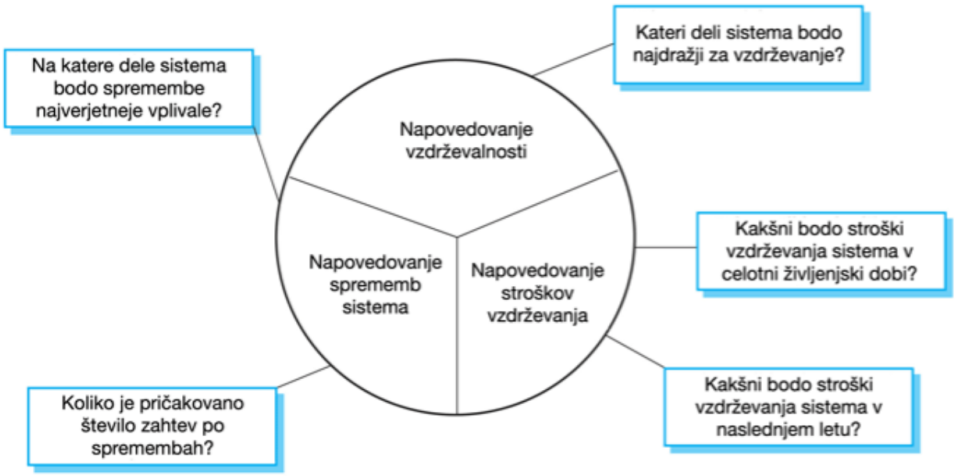
Napovedovanje števila sprememb zahteva razumevanje razmerij med sistemom in njegovim okoljem. Testno sklopljeni sitemi zahtevajo spremembe vedno, ko se okolje spremeni.

Dejavniki, ki vplivajo na to razmerje so:

- število in kompleksnost sistemskih vmesnikov
- število inherentno spremenljivih sistemskih zahtev
- poslovni procesi, v katerih se uporablja sistem

4.3.2 Metrike kompleksnosti

Napovedi vzdrževalnosti lahko izvedemo z **oceno kompleksnosti komponent sistema**. Študije so pokazale, da se večina napora pri vzdrževanju porabi za relativno majhno število komponent sistema.



Slika 21.11: Napovedovanje vzdrževanja

Figure 13: Napovedovanje vzdrževanja

Kompleksnost je odvisna od kompleksnosti:

- nadzornih struktur
- podatkovnih struktur
- objekta, metod in velikosti modula

4.3.3 Procesne metrike

Za oceno vzdrževalnosti lahko uprabimo **procesne metrike**:

- število zahtevkov za vzdrževanje odpravljanja napak
- povprečni čas, potreben za analizo vpliva
- povprečni čas, potreben za izpolnitev zahteve po spremebi
- število nerešenih zahtevkov za spremembe

4.4 Prenova programske opreme

Prenova programske opreme je prestrukturiranje ali ponoven razvoj dela ali celotnega obstoječega sistema brez spreminjanja njegove funkcionalnost.

Prenova programske opreme ima v primerjavi z zamenjavo naslednje **prednosti**:

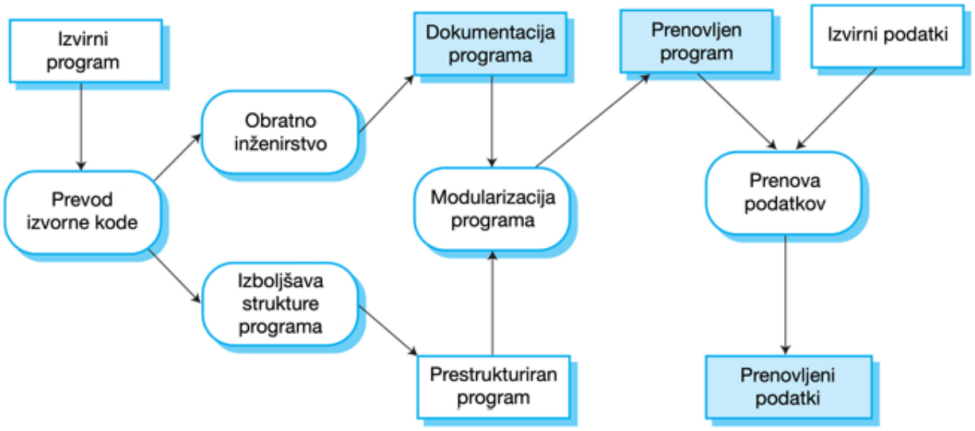
- **zmanjšano tveganje**, saj pri razvoju nove programske opreme obstaja veliko tveganje
- **znišani stroški**, ki so pri prenovi pogosto bistveno manjši od stroškov razvoja nove programske opreme

4.4.1 Aktivnosti procesa prenove programske opreme

Stroški prenove programske opreme so odvisni od obsega opravljenega dela, kjer obstaja širok **spekter možnih pristopov k prenovi**, kot je prikazano na sliki.

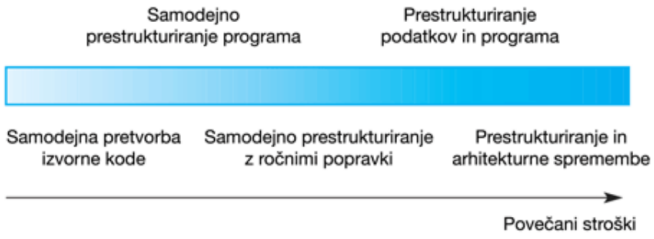
Na **stroške prenove programske opreme** vplivajo številni dejavniki:

- kakovost programske opreme, ki se prenavlja
- podpora orodij za izvedbo prenove
- zahtevan obseg pretvorbe podatkov
- razpoložljivost strokovnega osebja za pretvorbo



Slika 21.12: Proces prenove programske opreme

Figure 14: Proces prenove programske opreme



Slika 21.13: Pristopi k prenovi programske opreme

Figure 15: Pristopi k prenovi programske opreme

4.5 Preurejanje

Preurejanje oz. **preoblikovanje** je proces izboljšave programa z namenom upočasnitve postopnega poslabšanja programa, ki se zgodi z večkratnimi spremembami.

4.5.1 Preurejanje in prenova

Prenova programske opreme poteka po tem, ko je sistem nekaj časa vzdrževan in začnejo stroški vzdrževanja naraščati.

Preurejanje oz **preoblikovanje** pa je stalen proces izboljševanja pri razvoju in vrednotenju.

4.5.2 Problematični deli kode

Pri razvoju programske opreme obstajajo **stereotipne situacije**, kjer je mogoče kodo programa **izboljšati s preoblikovanjem**:

- **podvojena koda**
- **dolge metode**
- **switch stavki**
- **nakopičeni podatki**
- **špekulativna splošnost**

5 Zaključne ugotovitve

- Razvoj programske opreme in **evolucijo** lahko razumemo kot **integriran, iterativen proces**, ki ga lahko predstavimo z uporabo spiralnega modela.
- Pri sistemih po meri **stroški vzdrževanja** programske opreme običajno **presegajo stroške razvoja** programske opreme.
- Proces evolucije programske opreme temelji na zahtevah po spremembah in vključuje **analizo vpliva sprememb, načrtovanje izdaje** in **implementacijo sprememb.**
- **Podedovani sistemi** so starejši sistemi programske opreme, razviti s pomočjo zastarelih programskih in strojnih tehnologij, ki so še vedno uporabni za podjetja.
- **Vzdrževanje** obstoječega sistema je pogosto **cenejše in manj tvegano**, kot razvoj nadomestnega sistema, ki uporablja sodobno tehnologijo.
- Poslovno vrednost podedovanega sistema in kakovost aplikacije je treba oceniti, da se lahko odločimo, ali je treba sistem zamenjati, preoblikovati ali vzdrževati.
- Obstajajo **3 vrste vzdrževanja** programske opreme, in sicer: popravljanje napak, spreminjanje programske opreme za delo v novem okolju in izvajanje novih ali spremenjenih zahtev.
- **Prenova** programske opreme se ukvarja s prestrukturiranjem in ponovnim dokumentiranjem programske opreme, da jo lažje razumemo in kasneje tudi spreminjamo.
- **Preurejanje** je spreminjanje programa, kjer se ohrani prvotna funkcionalnost in je oblika preventivnega vzdrževanja.