

Uvod u programsko inženjerstvo

Programsko inženjerstvo (120/5 & 550/5 & 950/1)

Ciljevi poglavlja

- Objasniti pojam profesionalnog razvoja softvera.
 - Što označava pojam programsko inženjerstvo.
 - Odgovoriti na nekoliko najčešćih pitanja o programskom inženjerstvu.
- Etika u programskom inženjerstvu.

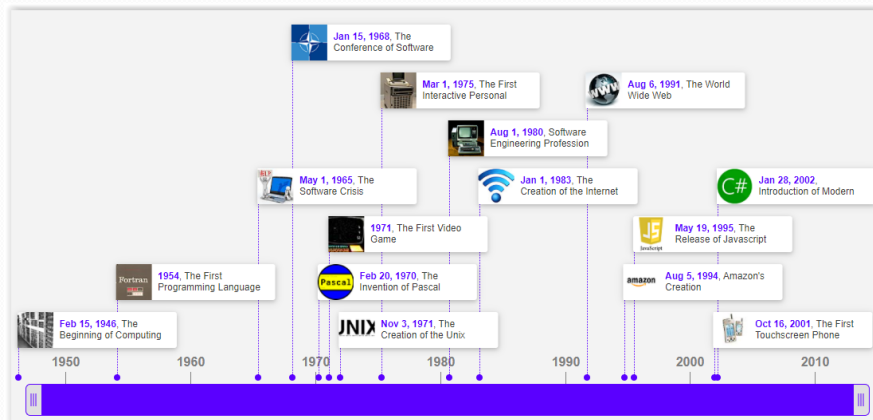
Programsko inženjerstvo

- Danas je život nemoguće zamisliti bez softvera i podrazumijeva se da on radi ispravno cijelo vrijeme.
 - Korištenje *mail-a*, pretraživača, ...
 - Softver je u semaforima, automobilima, televizijama, frižiderima, klima uređajima, ...
- Većina softvera koji se danas koristi je jako velik.
 - Npr. Windows Vista ima oko 50 milijuna linija koda a MacOSX oko 86 milijuna.
 - 50 milijuna linija koda
 - na A4 papirima to je 88 m visok toranj
 - ako se čita s razumijevanjem brzinom 120 linija/h trebalo bi nekome 417,000 sati ili 47 ½ godina.

Programsko inženjerstvo

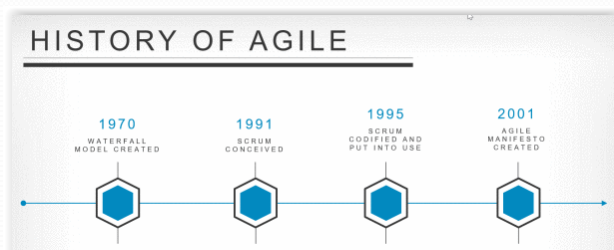
- Programsko inženjerstvo se bavi time kako napraviti softver bez obzira na njegovu veličinu i kompleksnost te da taj softver radi ispravno.
- Osnovna definicija:
 - Programsko inženjerstvo je inženjerska disciplina koja se bavi svim aspektima razvoja softvera, tj. bavi se teorijom, metodama i alatima za profesionalan razvoj softvera.
- Pojam programsko inženjerstvo se po prvi put pojavljuje 1968. na konferenciji koja je održana zbog tzv. "softverske krize".

Povijest programskog inženjerstva



Preuzeto s: <https://www.timetoast.com/timelines/the-history-of-software-engineering>

Povijest programskog inženjerstva



Preuzeto: <https://the-agilesphere.com/2016/09/18/the-history-of-agile/>

Programsko inženjerstvo

- Primjenjuje strukturu na proces izrade softvera što olakšava upravljanje veličinom i složenošću produkta.
- Glavni cilj je što jeftinija izrada kvalitetnog softvera.
- Postoje mnogi primjeri velikih i uspješnih softverskih produkata koji rade ispravno i imaju jako malo grešaka (engl. *bug-ova*).
 - Npr. Googleovi veliki projekti (Google search, Gmail, ...) su izradili timovi od nekoliko tisuća programera.

Softverski produkti

- Dva osnovna tipa softverskih produkata:
 - **Generički:**
 - Samostalni sustavi dostupni u javnoj prodaji (PC softver poput programa za obradu slika, softver za upravljanje projektom, programi za obradu teksta, tablica, ...)
 - Organizacija koja razvija program kontrolira i njegovu specifikaciju, prodaje se velikom broju različitih korisnika – (engl. *Comercial Of The Shelf* - COTS)
 - **Po narudžbi (engl. Custom):**
 - Kupac koji naručuje program definira njegovu specifikaciju i njen je vlasnik.
 - Program za kontrolu zračnog prometa, program za upravljanje financijama, ...
- Novi softver se može napraviti tako da se razvije iz početka, konfiguriranjem generičkih produkata ili ponovnim korištenjem postojećeg softvera.

Najčešća pitanja o PI-u

- Što je softver?
- Što je programsko inženjerstvo?
- Kako su raspoređeni troškovi kod razvoja softvera?
- Koje su karakteristike dobrog softvera?
- Koje su osnovne aktivnosti programskog inženjerstva?
- Koji su osnovni izazovi s kojima se susreće programsko inženjerstvo?
- Koju tehniku / metodu je najbolje koristiti kod programskog inženjerstva?
- Što je web promijenio kod programskog inženjerstva?

Što je softver?

- Računalni program i **pripadajuća dokumentacija** (zahtjevi, modeli dizajna, korisnička dokumentacija).

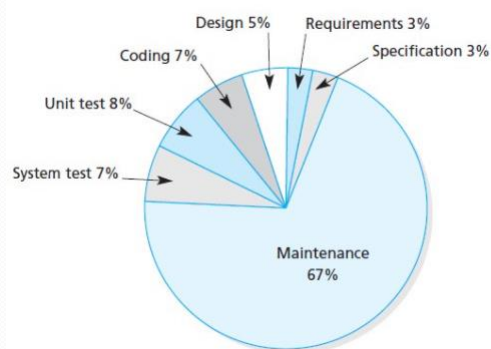
Kako su raspoređeni troškovi razvoja softvera?

- Troškovi razvoja softvera ovise o vrsti sustava koji se razvija, te o karakteristikama koje bi taj sustav trebao zadovoljiti (performanse, pouzdanost, ...).



Preuzeto: <https://freshcodeit.com/freshcode-post/how-to-calculate-the-cost-of-software-development>

Kako su raspoređeni troškovi razvoja softvera?



Preuzeto: https://www.researchgate.net/publication/232590980_On_the_Improvement_of_Cost-Effectiveness_A_Case_of_Regression_Testing/figures?lo=1

Koje su karakteristike dobrog softvera?

- Softver bi trebao **omogućiti** korisniku **traženu funkcionalnost i performanse**, a pri tome bi ga trebalo biti **lako održavati**, biti **pouzdan i prihvatljiv**.
- **Održavanje** - Program bi trebao biti pisan tako da ga se lako može mijenjati u skladu s korisničkim zahtjevima.
- **Pouzdanost** - Ne smije uzrokovati fizičke i/ili ekonomsku štetu u slučaju pada sustava.
- **Efikasnost** - Ne smije prekomjerno koristiti resurse (procesor, memorija) i odziv mora biti što brži.
- **Prihvatljivost i iskoristivost** (engl. *acceptability & usability*) - Proizvod mora biti prihvaćen od strane korisnika.

Koje su osnovne aktivnosti programskog inženjerstva?

- Postoje četiri osnovne aktivnosti koje su zajedničke za sve softverske projekte:
 - **Specifikacija** – što sustav treba raditi i o kojim ograničenjima treba voditi računa pri razvoju.
 - **Razvoj** (engl. *development*) – produkcija specificiranog softvera.
 - **Validacija** – provjera da softver radi ono što je kupac naručio.
 - **Održavanje** (engl. *evolution*) – promjena softvera u skladu s promjenom zahtjeva tržišta.

Koji su osnovni izazovi programskog inženjerstva?

- Težnja da se udovolji promjenjivim zahtjevima na tržištu, kratko vrijeme razvoja, razvoj softvera kojem se može vjerovati.
- **Heterogenost** - Ovo je izazov kako razviti tehniku za izradu softvera koji je dovoljno fleksibilan da se može nositi s različitim platformama i izvršnim okolinama.
- **Isporuka** - Ovo je izazov čiji je cilj što više skratiti vrijeme razvoja softvera bez da se kompromitira njegova kvaliteta.
- **Povjerenje** - Razvoj tehnika koje demonstriraju da korisnici mogu vjerovati softveru.



Koju tehniku/metodu je najbolje koristiti?

- Razvojem svih softverskih projekata treba upravljati na profesionalan način, ali različite metode su prikladne za različite projekte.
- Postoji velik broj različitih softverskih sustava i ne postoji univerzalan niz tehnika koji se može primijeniti na sve njih.
- Koje će se metode i alati programskog inženjerstva koristiti ovisi o vrsti aplikacije koja se razvija, zahtjevima korisnika i početnom znanju razvojnog tima.



Što je web promijenio kod prog. inženjerstva?

- Razvoj weba je doveo do korištenja servisa i mogućnosti razvoja vrlo distribuiranih sustava baziranih na servisima.
- Doveo je do velikog razvoja programskih jezika i ponovnog korištenja softvera (engl. *Software reuse*).
 - Ponovno korištenje softvera je osnovni pristup koji se koristi za izradu sustava baziranih na web-u.
- Web sustavi bi se trebali razvijati i isporučivati inkrementalno.

Profesionalna i etička odgovornost

- Programsko inženjerstvo uključuje mnogo širi spektar odgovornosti nego jednostavnu primjenu tehničkih znanja.
- Etično ponašanje je mnogo više od ponašanja u skladu sa zakonom a između ostalog se odnosi na:
 - Povjerenje
 - Kompetencija
 - Intelektualno vlasništvo
 - Zlouporaba računala
- Organizacije poput ACM-a, IEEE i "British Computer Society" su objavili kod profesionalnog ponašanja tzv. "kod etike" koji definira osnove etičnog ponašanja

Principi ACM/IEEE "Koda etike"

1. PUBLIC - Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Primjeri etičkih dilema

- Ne slaganje s odlukama višeg menadžmenta.
- Zaposlenik se ponaša na neetičan način:
 - bez završenog testiranja otpušta sustav s kritičnom sigurnošću.
- Sudjelovanje u razvoju vojnog naoružanja ili nuklearnih sustava.

Uvod u prog. inženjerstvo

Kraj