Planiranje projekta

Stana Obrenić 1120/2012

Teme koje ćemo obraditi:

- Troškovi softvera
- Razvoj vodjen planom
- Raspored projekta
- Agilno planiranje
- Tehnike procenjivanja

Planiranje projekta

- Planiranje projekta podrazumeva podelu posla na delove i njihova dodela članovima tima, predviđanje problema koje mogu nastati, kao i rešenja na te probleme.
- Plan projekta, koji je nastao na početku, koristi se za komunikaciju izmedju projektnog tima i klijenta i da pomogne proceni napretka projekta.

Faze planiranja

- Faza predloga
  - kada se nadmećemo za ugovor o razvoju ili obezbeđivanje softverskog sistema
- Tokom početne faze projekta
  - kada treba da isplaniramo ko će da radi na projektu, kako će projekat da bude podeljen na korake, kako će sredstva biti raspoređena po kompaniji...
- Periodično tokom projekta
  - kada modifikujemo plan zbog novog stečenog iskustva i informacija dobijenih praćenjem napretka projekta.

Cena softvera

- Procene se prave da bi otkrile cenu softverskog sistema koji se proizvodi.
  - Uzimaju se u obzir hardver, softver, troškovi putovanja, treninga i troškovi usluge
- Nema jednostavne veze između cene razvoja i cene koju plaća klijent.
- Treba uzeti u obzir širu organizacionu, ekonomsku, političku i biznis situaciju, sve to utiče na cenu koja se naplaćuje.

Faktori koji utiču na cenu softvera

Faktor	Objašnjenje
Tržišna mogućnost	Razvojna organizacija može da utiče na nisku cenu jer želi da se preseli na novi segment softverskog tržišta. Prihvatanjem niskog profita na jednom projektu može dati organizaciji priliku da kasnije napravi veći profit.
Nesigurna procena troškova	Ukoliko organizacija nije sigurna u svoju procenu računa, to može da nepredviđeno poveća cenu iznad svog normalnog profita
Uslovi ugovora	Klijent može biti voljan da dozvoli razvijaocima da zadrže vlasništvo izvornog koda i da ga koriste u drugim projektima. Cena onda može biti manja nego ako je izvorni kod predat kupcu

Faktori koji utiču na cenu softvera

Faktor	Objašnjenje
Nestabilnost zahteva	Ako su zahtevi promenljivi, organizacija može da snizi njihovu cenu da bi dobili ugovor. Nakon što su dobili ugovor, veća cena može biti naplaćena zbog promena zahteva
Finansije	Razvijaoci u teškoj finansiskoj situaciji mogu da spuste cenu da bi dobili ugovor. Bolje je da prave manji profit nego normalan profit ili pauzu. Novčani tok je važniji od profita u teškim ekonomskim vremenima

Razvoj vođen planovima

- Razvoj vođen planovima je pristup softverskog inžinjeringa u kom su razvojni procesi isplanirani detaljno.
  - Razvoj vođen planovima se zasniva na inžinjerskim tehnikama upravljanja projektima i to je tradicionalni način upravljanja velikim softverskim projektima.
- Projektni plan sadrži poslove koje treba obaviti, ko će to uraditi, raspored razvoja i proizvod rada.
- Menadžeri koriste plan u donošenju odluka i kao način merenja napretka.

Razvoj vođen planovima – za i protiv

- Argumenti u korist razvoja vođenog planovima su to da rano planiranje omogućava organizaciona pitanja (dostupnost kadrova, ostali projekti, ...) da se pažljivo uzmu u obzir, i to da potencijalni problemi i zavisnosti se otkriju pre nego što projekat počne.
- Glavni argument protiv razvoja vođenog planovima je taj što mnoge rane odluke moraju da se revidiraju zbog promena okruženja u kom će softver da se razvija i koristi.

Planovi projekta

- U razvoju vodjenog planovima, plan projekta određuje raspoložive resurse, poslovne celine (*work breakdown*) i raspored za obavljanje rada.
- Sekcije plana:
  - Uvod
  - Organizacija projekta
  - Analiza rizika
  - Zahtevi za hardverske i softverske resurse
  - Poslovne celine
  - Nadgledanje i izveštavanje

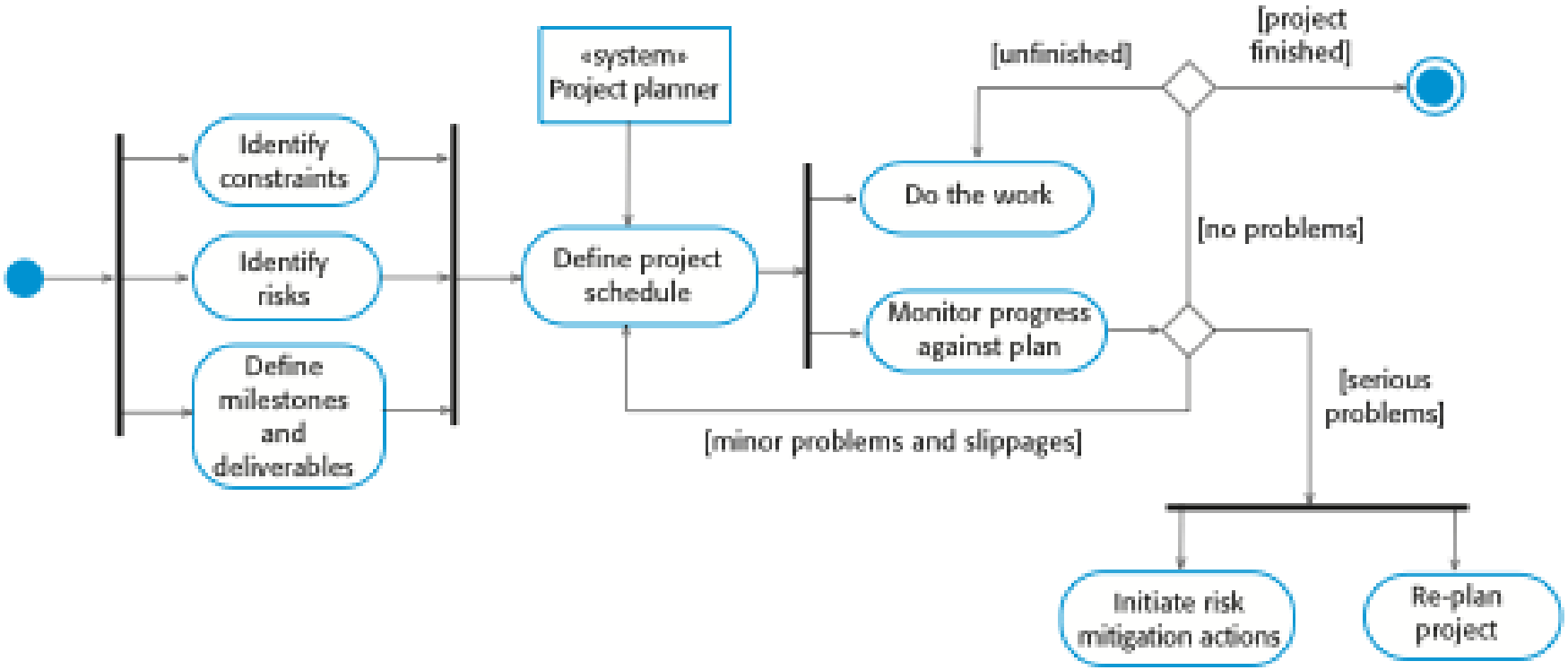
Dodaci planu projekta

Plan	objašnjenje
Kvalitet	Opisuje kvalitet procedure i standarde koji će biti korišćeni u projektu
Validacija	Opisuje pristup, resurse i raspored koji se koristi za validaciju sistema
Upravljanje konfiguracijama	Opisuje upravljanje konfiguracijama procedura i struktura koje će se koristiti
Održavanje	Predviđa zahteve za održavanje, troškove i napor
Razvoj osoblja	Opisuje kako će veštine i iskustvo članova projektnog tima da se razvija

Proces planiranja

- Planiranje projekta je iterativni proces koji počinje kada kreirate inicijalni početni plan tokom početne faze projektovanja.
- Promene plana su neizbežne.
  - Kako više informacija o sistemu i projektnom timu postaju dostupnije tokom projekta, tako treba da se redovno revidira plan da odražava zahteve, raspored i promene rizika.
  - Menjanje poslovnih ciljeva takođe vodi do promena u projektnom planu. Kako se poslovni planovi menjaju, tako to može da utiče na sve projekte, koji će možda nakon toga morati da se ponovo isplaniraju.

Proces planiranja projekta



Raspored projekta

- Raspored projekta je proces odlučivanja kako će rad u projektu biti organizovan u zasebne zadatke i kada će i kako će ovi zadaci biti izvršeni.
- Proceni se vreme potrebno da se izvrše svaki zadaci, usluge potrebne za to i ko će da radi na tim zadacima.
- Takođe, treba da se procene potrebni resursi za svaki zadatak (npr. prostor na disku), potrebno vreme za specijalizovane hardvere (kao što je simulator) i budžet za putovanja.

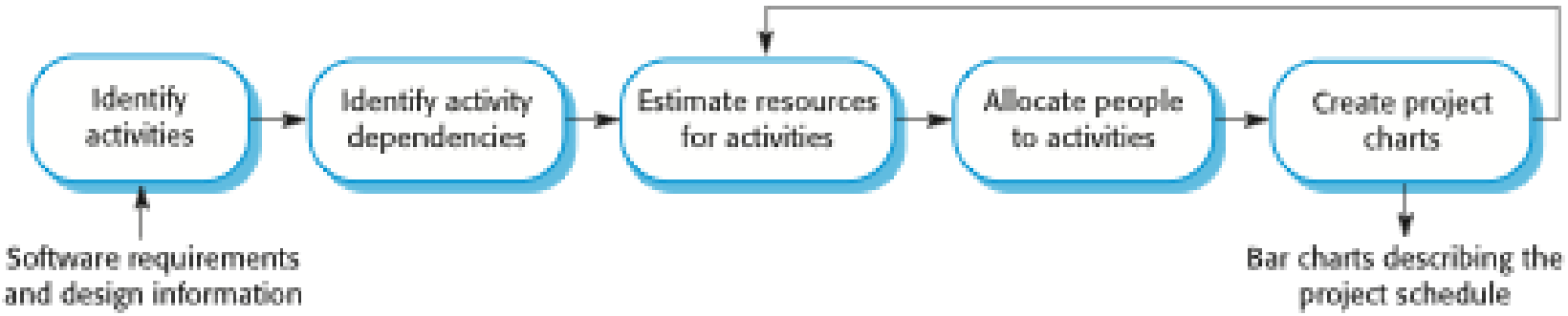
Raspored projekta - aktivnosti

- Podeliti projekat na zadatke i proceniti potrebno vreme i resurse za izvršavanje svakog zadatka.
- Konkurentno organizovati zadatke da bi se optimalno iskoristio radni prostor.
- Smanjiti međusobnu zavisnost zadataka da bi se izbegla kašnjenja uzrokovana time što jedan zadatak čeka drugi da bi se izvršio.
- Zavise od intuicije i iskustva projekt menadžera.

Prekretnice i isporuke

- Prekretnice (milestones) su tačke u rasporedu u kojima se može proceniti napredak
  - npr. kada se predaje sistem na testiranje
- Isporuke (deliverables) su rezultati rada koji se isporučuju klijentu
  - npr. dokument sa sistemskim zahtevima.

PROCESI RASPOREDA PROJEKTA



Problemi sa rasporedom

- Procenjivanje teškoće problema a samim tim i procena izrade rešenja je teško.
- Produktivnost nije propocionalna broju ljudi koji rade na zadatku.
- Dodavanje ljudi kasno u projekat uzrokuje kašnjenje zbog problema u komunikaciji.
- Neočekivane situacije se uvek dešavaju. Dozvoljavati pretpostavke u planiranju.

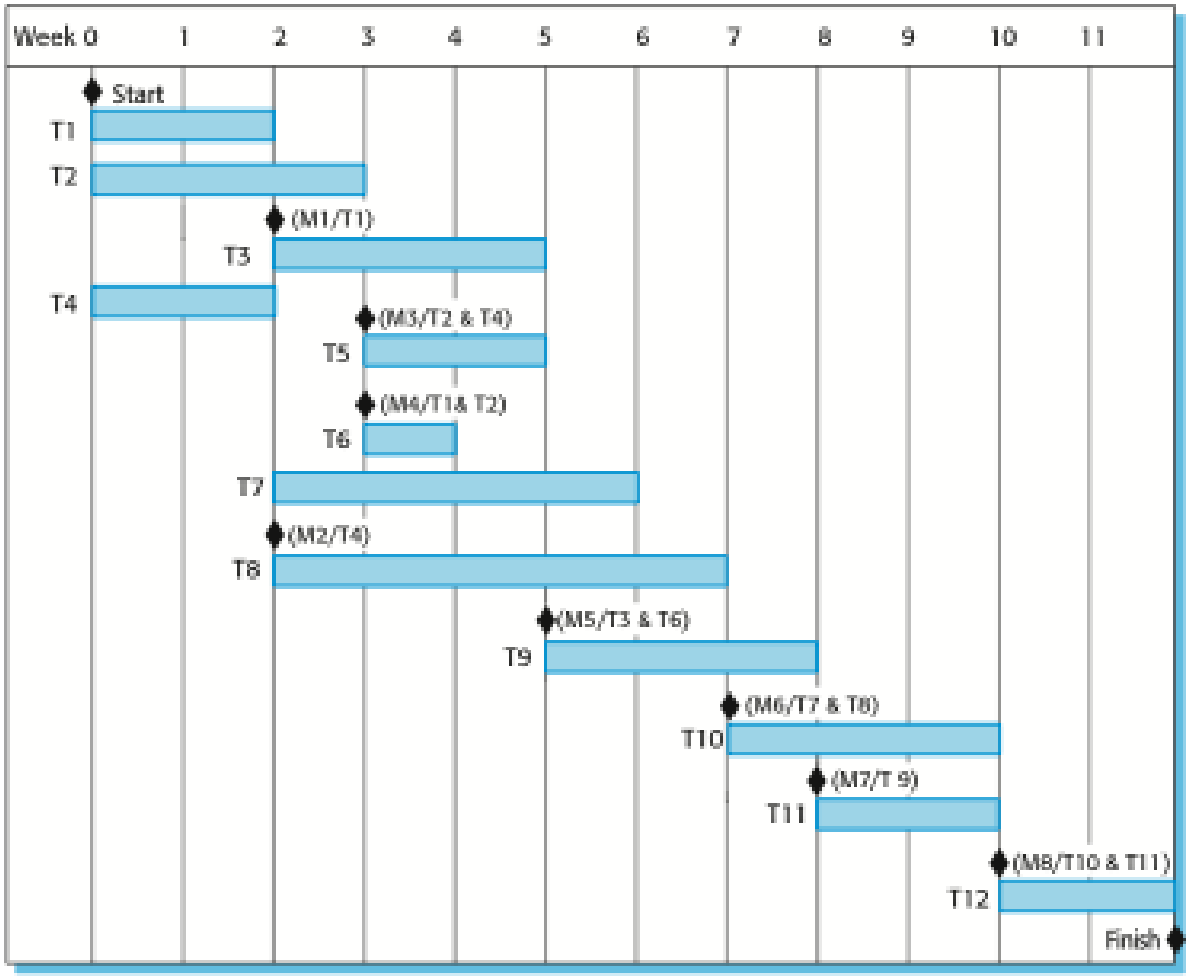
Prezentacija rasporeda

- Grafička notacija se uglavnom koristi za ilustrovanje rasporeda projekta.
- Pokazuju projekat podeljen na zadatke. Zadaci ne bi trebalo da budu previše mali, trebalo bi da traju oko jedne ili dve nedelje.
- Bar grafikoni su najčešće korišćena prezentacija za raspored projekta. Pokazuju zadatke kao aktivnosti ili resurse u zavisnosti od vremena.

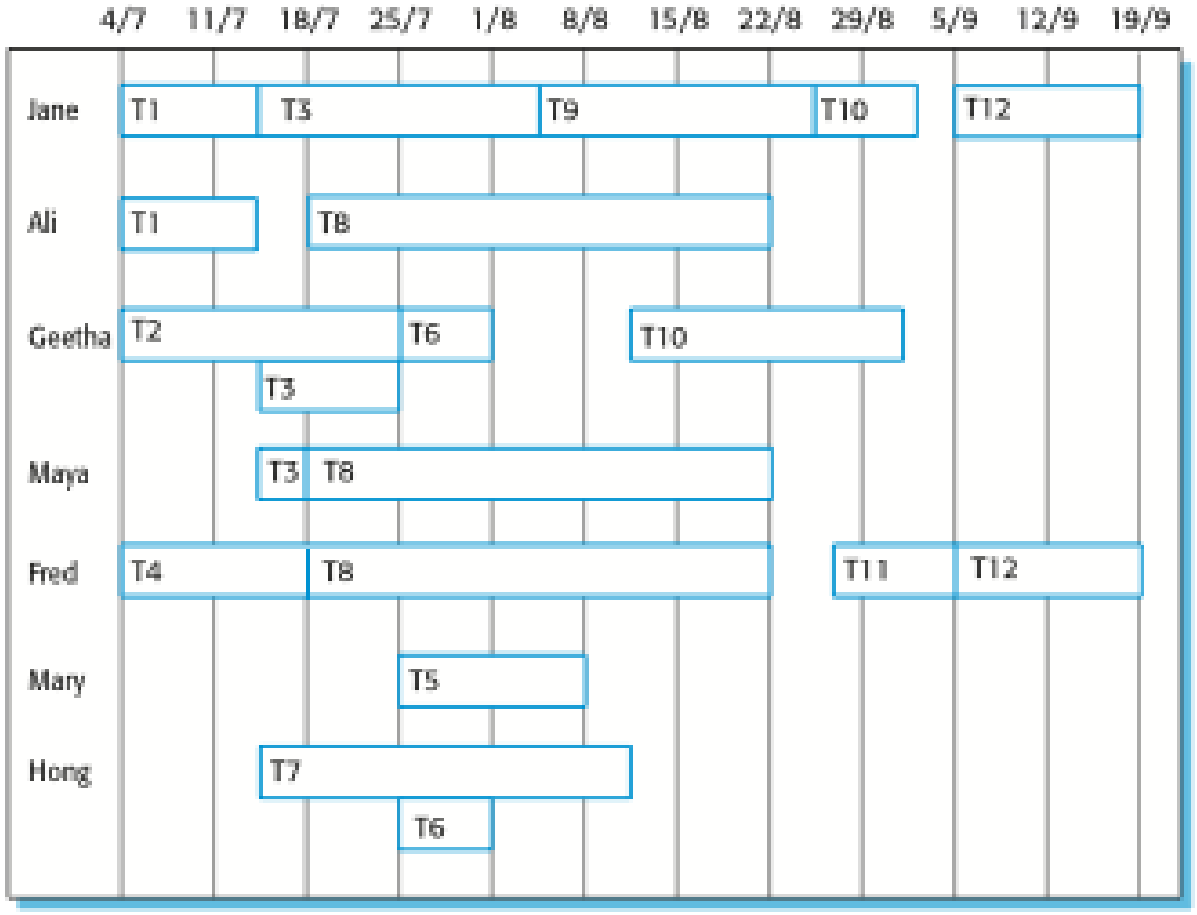
ZADACI, TRAJANJE I ZAVISNOSTI

Zadatak	Usluga (osoba/dan)	Trajanje (broj dana)	Zavisnosti
T1	15	10	
T2	8	15	
T3	20	15	T1 (M1)
T4	5	10	
T5	5	10	T2, T4 (M3)
T6	10	5	T1, T2 (M4)
T7	25	20	T1 (M1)
T8	75	25	T4 (M2)
T9	10	15	T3, T6 (M5)
T10	20	15	T7, T8 (M6)
T11	10	10	T9 (M7)
T12	20	10	T10, T11 (M8)

BAR GRAFIKON AKTIVNOSTI



GRAFIKON RASPOREDA OSOBLJA



Agilno planiranje

- Agilne metode su iterativni pristupi razvoju softvera gde se softver razvija i isporučuje klijentima u koracima.
- Za razliku od razvoja vođenog planovima, funkcionalnost ovih koraka nije planirana unapred, već se odlučuje tokom razvoja.
  - Odluke šta će biti uključeno u jedan korak zavisi od napretka i od prioriteta klijenta.
- Prioriteti klijenta i zahtevi se menjaju tako da ima smisla za fleksibilan plan koji može da se prilagodi tim promenama.

Agilno planiranje - faze

- Planiranje izdanja
  - za nekoliko meseci unapred
  - odluke o funkcijama koje trebaju da budu uključene u izdanjima sistema
- Planiranje iteracije
  - ima kraći rok i fokusira se na planiranje sledeće iteracije u sistemu
  - obično oko 2-4 nedelje

Planiranje u ekstremnom programiranju

Planiranje zasnovano na priči

- Specifikacija sistema u ekstremnom programiranju je zasnovana na korisničkim pričama koje oslikavaju funkcije koje treba da budu uključene u sistem.
- Članovi tima čitaju i diskutuju priče i rankiraju ih po vremenu potrebnom za njihovo izvršavanje.
- Planiranje izdanja uključuje odabir i preradu priča koje utiču na funkcije koje će se implementirati u izdanju sistema i red u kom te priče treba da budu implementirane.

Tehnike procenjivanja

- Organizacija treba da napravi procenu usluga i troškova. Postoje dve tehnike koje mogu da se koriste za procenu:
  - Tehnika zasnovana na iskustvu
    - Procena budućih napora se zasniva na iskustvu menadžera. U suštini, menadžer neformalno zaključuje koliki će troškovi i usluge biti.
  - Algoritamsko modeliranje troškova
    - U ovom pristumu koriste se formule, izračunavaju se na osnovu procene atributa proizvoda (kao što je veličina) i karakteristike procesa (kao što je iskustvo članova tima)

Tehnike zasnovane na iskustvu - pristupi

- Tehnike zasnovane na iskustvu se oslanjaju na presude na osnovu iskustva iz prethodnih projekata i na uloženom trudu na tim projektima u aktivnostima razvoja softvera.
- Obično se identifikuju isporuke koje trebaju da se proizvedu u projektu i različite komponente softvera ili sistemi koji trebaju da se razviju.
- To se dokumentuje u tabeli, procenjuju se individualno i računa se komplentna zahtevana usluga.
- Obično pomaže da se napravi grupa ljudi koja procenjuje troškove.

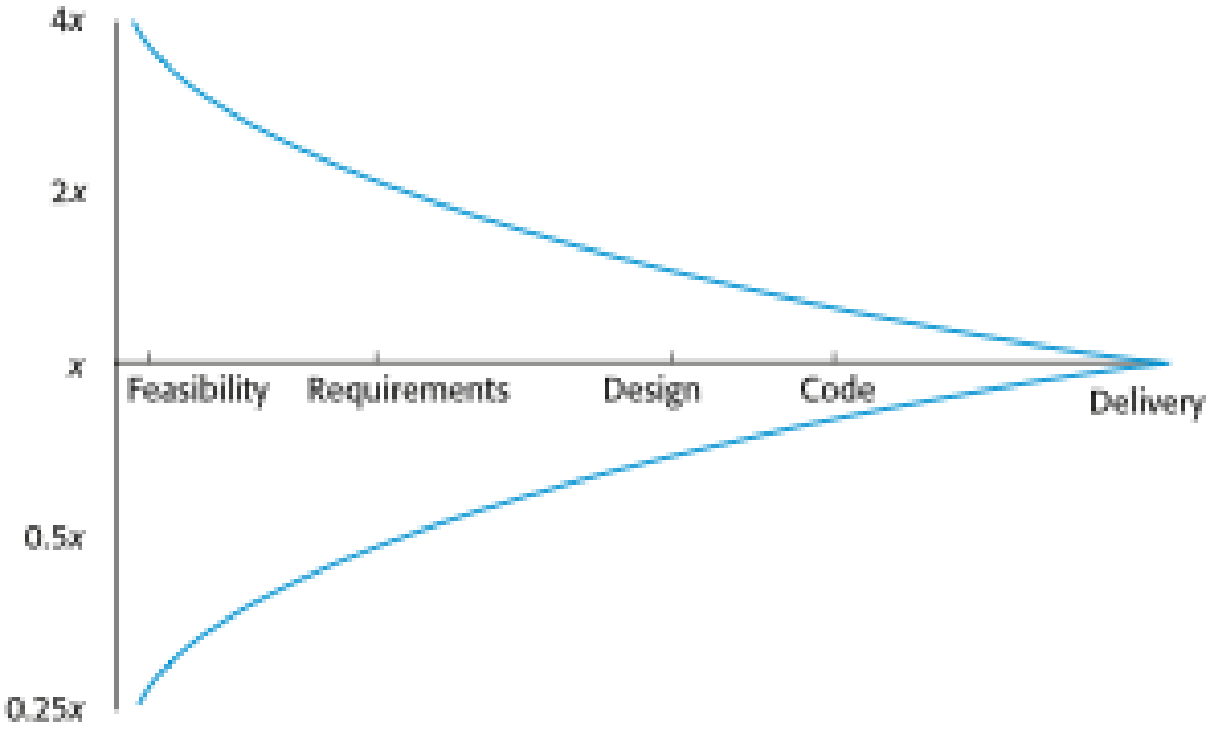
Algoritamsko procenjivanje troškova

- Algoritamska procena troškova koristi matematičke formule da predvidi cenu projekta na osnovu procene veličine projekta, tipa projekta i drugih faktora.
- Uglavnom se zasniva na formuli:
  - Usluga = $A \times Size^B \times M$
    - A je konstanta i zavisi od organizacije i tipa softvera, B je eksponent uglavnom između 1 i 1.5, M je multiplikator proizvoda, procesa i ljudskih atributa.
    - Vrednosti A, B i M procenjuje menadžer projekta
- Najčešće korišćen atribut proizvoda za procenu cene je veličina koda (SLOC).

Tačnost procene

- Veličina softverskog sistema može biti tačno izmerena samo kada se završi.
- Neki faktori koji utiču na konačnu veličinu:
  - Korišćenje COTS (*Commercial Off-The-Shelf*) i komponenti
  - Programski jezik
  - Distribucija sistema
- Kako razvojni proces napreduje tako se i tačnost procena povećava.
- Procena faktora koje doprinose B i M su subjektivne i variraju od presude procenjivača.

TAČNOST PROCENE



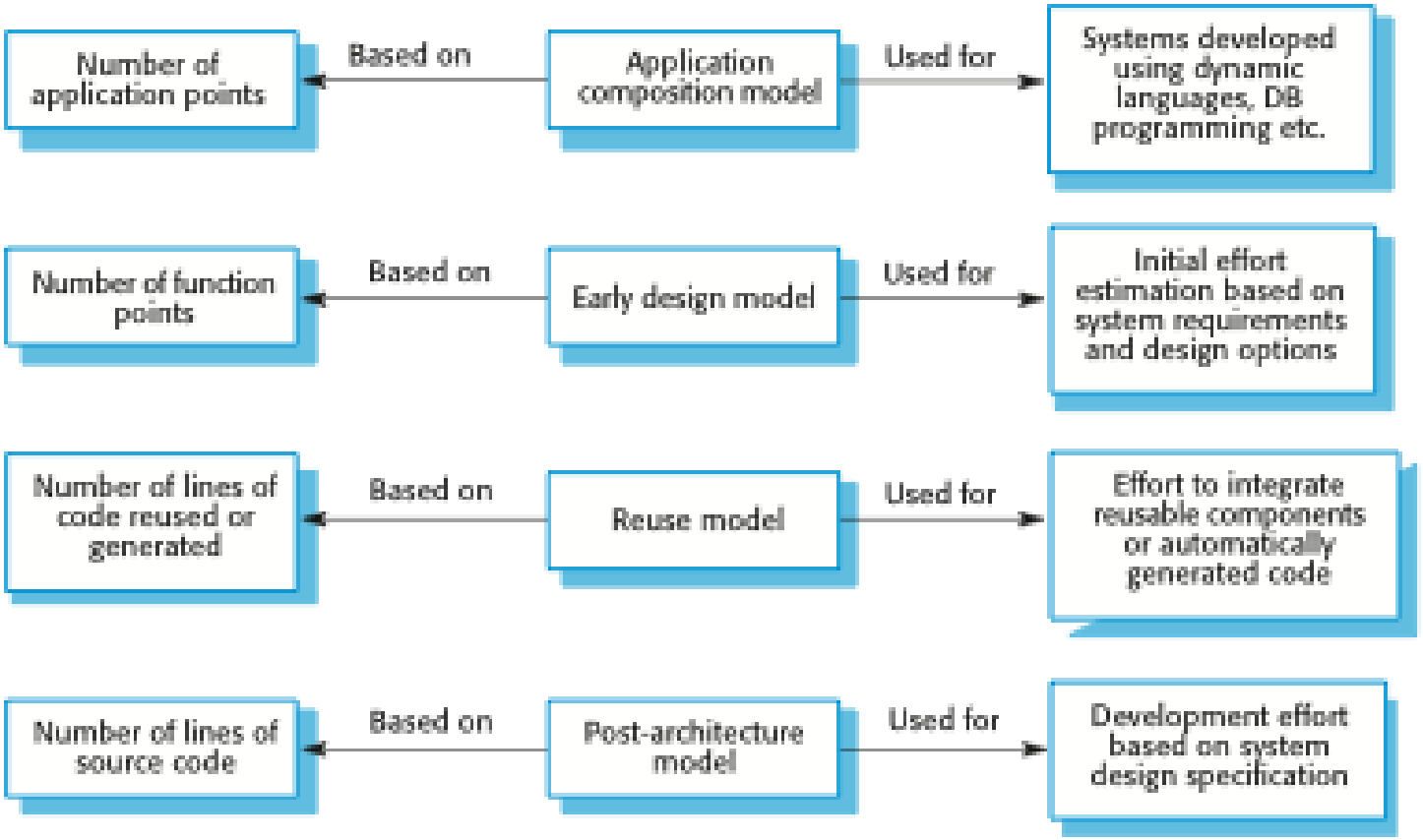
Model COCOMO 2

- Empirijski model zasnovan na iskustvima sa prethodnih projekata.
- Dobro dokumentovan, “nezavisan” model koji nije vezan za određenog proizvođača softvera.
- Dugačka istorija od inicijalne verzije objavljene 1981. (COCOMO-81) do raznih varijacija COCOMO 2.
- COCOMO 2 uzima u obzir različite pristupe razvoja softvera, ponovnu upotrebljivost, itd.

Model COCOMO 2

- COCOMO 2 obuhvata niz podmodela koje proizvode detaljnije softverske procene.
- Podmodeli u COCOMO 2 su:
  - Kompozicioni model
    - Koristi se kada je softver sastavljen od već postojećih komponenti
  - Rano dizajniran model
    - Koristi se kada su zahtevi poznati ali dizajn nije još uvek
  - Model ponovne upotrebe
    - Koristi se da se izračuna napor integrisanja ponovno upotrebljenih modela
  - Post-arhitekturni model
    - ko risti se kada se dizajnirana arhitektura sistema i dostupno je više informacija o sistemu

COCOMO MODELI PROCENJIVANJA



KOMPOZICIONI MODEL

- Podržava prototipove projekta i projekte gde postoji spajanje postojećih komponenti.
- Zasniva se na standardnim procenama razvoja softvera u primeni (objekat) poeni/mesec.
- Uzima CASE alat u obzir.
- Formula je
  - PM = (NAP×(1 - %reuse/100)) / PROD
    - PM je usluga osoba/mesec, NAP je broj aplikacionih poena, %reuse je procena ponovo upotrebljenog koda i PROD je produktivnost.

APLIKACIONI SASTAV - PRIMER

Iskustvo i sposobnost razvijaoca	Veoma slabo	Slabo	Normalno	Jako	Veoma jako
Zrelost i sposobnost ICASE	Veoma slabo	Slabo	Normalno	Jako	Veoma jako
PROD (NAP/month)	4	7	13	25	50

Rano dizajniran model

- Procene mogu da se naprave nakon što se zahtevi prihvate.
- Zasniva se na standardnoj formuli za algoritamske modele
  - $PM = A \times Size^{B} \times M$
    - M = PERS × RCPX × RUSE × PDIF × PREX × FCIL × SCED;
    - A = 2.94 je inicijalna saradnja, Size u KLOC, B je između 1.1 i 1.24 zavisi od novina u projektu, fleksibilnosti razvoja, pristupa rizicima i zrelosti procesa.

Multiplikatori

- Multiplikatori odražavaju sposobnost programera, nefunkcionalnih zahteva, bliskost sa razvojnim platformama, itd...
  - RCPX – pouzdanost i složenost projekta
  - RUSE – potrebno ponovno korišćenje
  - PDIF – složenost platforme
  - PREX – iskustvo osoblja
  - PERS – osposobljenost osoblja
  - SCED – zahtevan raspored
  - FCIL – podržavanje sadržaja

Model ponovne upotrebe

- Uzima u obzir kodove “crna kutija” koji se ponovo koristi bez menjanja i kod koji treba da se prilagodi da bi mogao da se integriše sa novim kodom.
- Imaju dve verzije:
  - ponovno korišćenje “crne kutije” gde se kod ne modifikuje. Procena veličine napora (PM) je izračunata.
  - ponovno korišćenje “bele kutije” gde se kod modifikuje. Procena veličine napora je ekvivalentna broju linija novog koda koji se menja.

Model ponovne upotrebe - procene

- Za generisan kod:
  - PM = (ASLOC * AT/100)/ATPROD
    - ASLOC je broj linija generisanog koda
    - AT je procena koda koji je automatski generisan
    - ATPROD je produktivnost inžinjera u integrisanju ovog koda.

Model ponovne upotrebe – procene

- Kada kod mora da se razume i da se integriše:
  - ESLOC = ASLOC * (1-AT/100) * AAM.
    - ASLOC i AT kao u prethodnom.
    - AAM je adaptacija multiplikatora na promene izračunata iz cene promene koda koji se ponovo koristi, cene razumevanja integrisanog koda i cene odluke ponovnog korišćenja.

Post-arhitekturni nivo

- Koristi istu formulu kao i prethodni dizajn model ali sa 17 umesto 7 povezanih multiplikatora.
- Veličina koda se procenjuje korišćenjem parametara:
  - Broj linija novog koda koji treba da bude razvijen (SLOC)
  - Procena ekvivalentnog broja linija novog koda izračunat korišćenjem modela ponovnog korišćenja (ESLOC)
  - Procena broja linija koda koje treba da se modifikuju zbog promena zahteva

Faktori koji se koriste u posta-arhitekturnom modelu

Faktor	Objašnjenje
Prethodno iskustvo (*Precedentedness*)	Prethodno iskustvo organizacije sa takvim tipom projekta. *Veoma slabo* znači da nema iskustva, v*eoma jako* da je potpuno upoznata sa takvim tipom.
Fleksibilnost razvoja	Stepen fleksibilnosti u razvojnom procesu. *Veoma slabo* znači da se koriste propisani procesi, a *veoma jako* znači da klijent postavlja samo krajnje ciljeve.
Architecture/risk resolution	Obim sprovedene analize. *Veoma slabo* znači malo analize, *veoma jako* označava kompletnu i detaljnu analizu.
Povezanost tima	Koliko dobro članovi tima se međusobno poznaju. *Veoma slabo* znači veoma teška komunikacija, *veoma jako* znači da tim nema komunikacionih problema.
Zrelost procesa	Zrelost organizacije. Proračun ove vrednosti zavisi od CMM upitnika zrelosti, mada se procena može izračunati i oduzimanjem CMM zrelosti procesa nivo 5.

Multiplikatori

- Atributi proizvoda
  - Zahtevane karakteristike softvera koji se razvija
- Kompjuterski atributi
  - Ograničenja nametnuta softveru zbog hardverske platforme
- Lični atributi
  - Uzima se u obzir iskustvo i sposobnosti zaposlenih na projektu
- Projektni atributi
  - Određene karakteristike projekta

Uticaj troškova drajvera na procene napora

Vrednost eksponenta	1.17
Veličina sistema (uključujući faktore za ponovnu upotrebu i promenljive zahteve)	128,000 DSI
Početna COCOMO procena bez cene drajvera	**730 osoba-mesec**
Pouzdanost	Veoma jako, multiplikator = 1.39
Složenost	Veoma jako, multiplikator = 1.3
Memorijsko ograničenje	Jako, multiplikator = 1.21
Alat	Slabo, multiplikator = 1.12
Raspored	Ubrzan, multiplikator = 1.29
Prilagođena COCOMO procena	**2,306 osoba-mesec**

Uticaj troškova drajvera na procene napora

Vrednost eksponenta	1.17
Pouzdanost	Veoma slabo, multiplikator = 0.75
Složenost	Veoma slabo, multiplikator = 0.75
Memorijsko ograničenje	Nema, multiplikator = 1
Alat	Veoma jako, multiplikator= 0.72
Raspored	Normalan, multiplikator = 1
Prilagodjena COCOMO procena	**295 osoba-mesec**

Trajanje projekta i zaposleni

- Menadžeri moraju da procenjuju i vreme potrebno da se završi projekat i koliko zaposlenih će biti potrebno.
- Kalendarsko vreme se može proceniti korišćenjem COCOMO 2 formule
  - $TDEV = 3 \times (PM)^{(0.33+0.2*(B-1.01))}$
- Potrebno vreme zavisi od broja zaposlenih koji rade na projektu.

Potrebni zaposleni

- Broj ljudi koji rade na projektu varira u zavisnosti od faze projekta.
- Više ljudi koji rade na projektu, veća totalna usluga je obično zahtevana.
- Veoma brzo nagomilavanje ljudi često uzrokuje zaostajanje u odnosu na raspored.

Sažetak

- Cena troškova ne zavisi samo od procene cene razvoja, ona može zavisiti i od tržišta i organizacionih prioriteta.
- Razvoj vodjen planovima je organizovan tako da obuhvata ceo projektni plan koji definiše aktivnosti, usluge, raspored aktivnosti i ko je organizovan za koju aktivnost.
- Raspored projekta podrazumeva grafičku reprezentaciju projektnog plana. Bar čarovi trajanja aktivnosti i rasporeda osoblja su najčešće korišćene grafičke reprezentacije.
- U ekstremnom programiranju igra planiranja uključuje ceo tim u planiranju projekta. Plan se razvija inkrementalno i ako nastanu problemi prilagođava se njima. Funkcionalnost softvera se redukuje umesto odlaganja isporuke.

Sažetak (2)

- Tehnika procene softvera može biti zasnovano na iskustvu, gde menadžeri ocenjuju potrebne usluge, ili algoritamski, gde potrebna usluga se računa iz drugih procena parametara projekta.
- COCOMO 2 model je algoritamske procene troškova koji koristi projekte, proizvode, hardver i personalne atribute kao i veličinu proizvoda i složenost atributa koje procenjuju troškove.

Hvala na pažnji! ☺