

Procena veličine, cene i planiranje projekta

Predrag Dimitrijević 1126/2015

Uvod

- Određivanje vremena i novca potrebnog za izradu projekta je proces koji se sastoji od više koraka.
- Taj proces uključuje procenu količine truda, osnivanje plana i izračunavanje cene projekta.
- Proces se ponavlja tokom čitavog životnog ciklusa projekta, kako dolazi do promena zahteva i ostalih faktora.

Procena cene kao proces

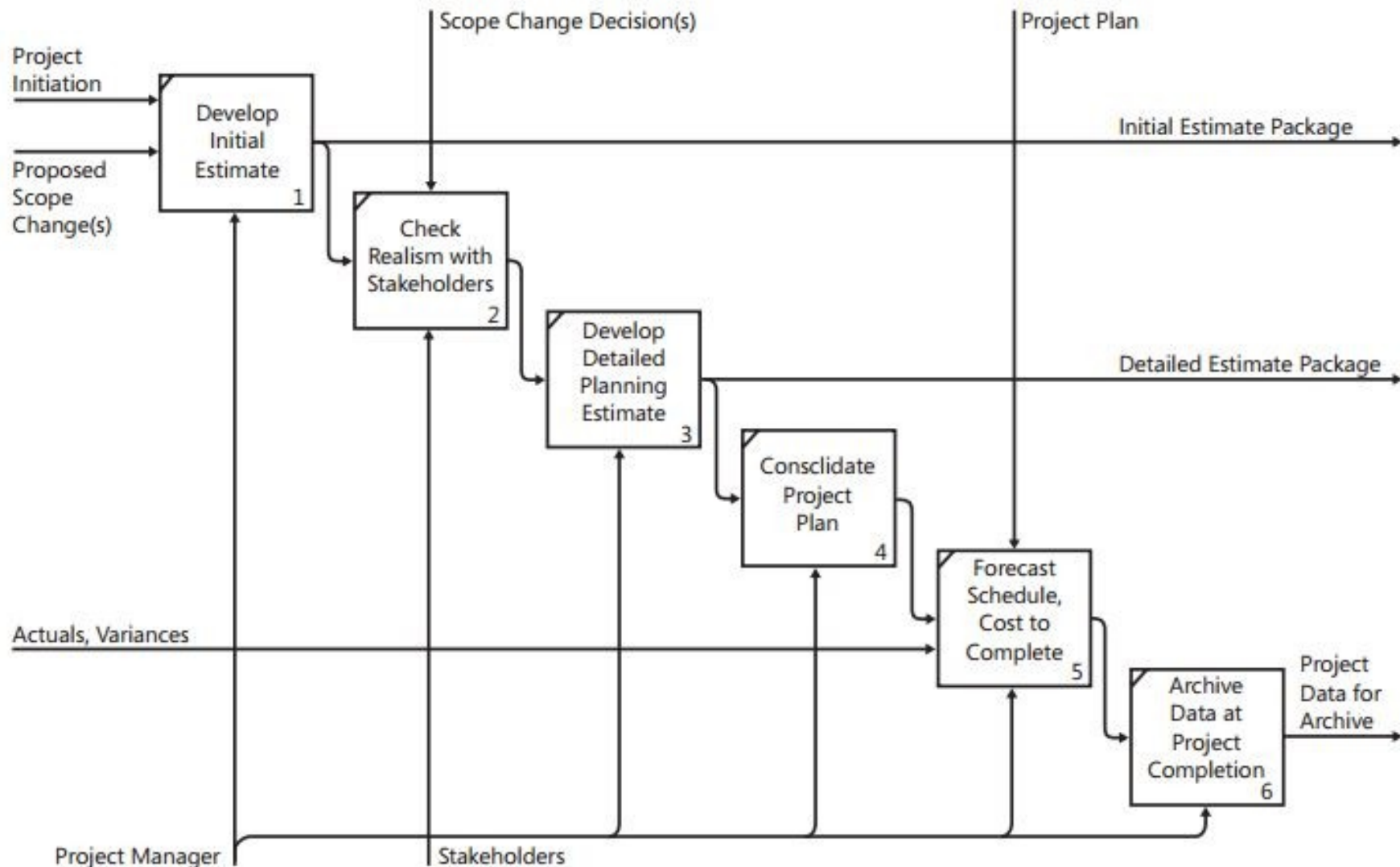


FIGURE 7-1 Software estimation as a lifecycle process.

Procena cene kao proces (2)

- Inicijalna procena
- Provera sa zainteresovanim stranama oko izvodljivosti inicijalnog plana
- Detaljna procena projekta
- Konsolidacija plana
- Određivanje rasporeda plana i cene potrebne da se projekat završi
- Arhivirati podatke nakon završetka projekta

Promenljivost procene kao funkcija perioda trajanja projekta

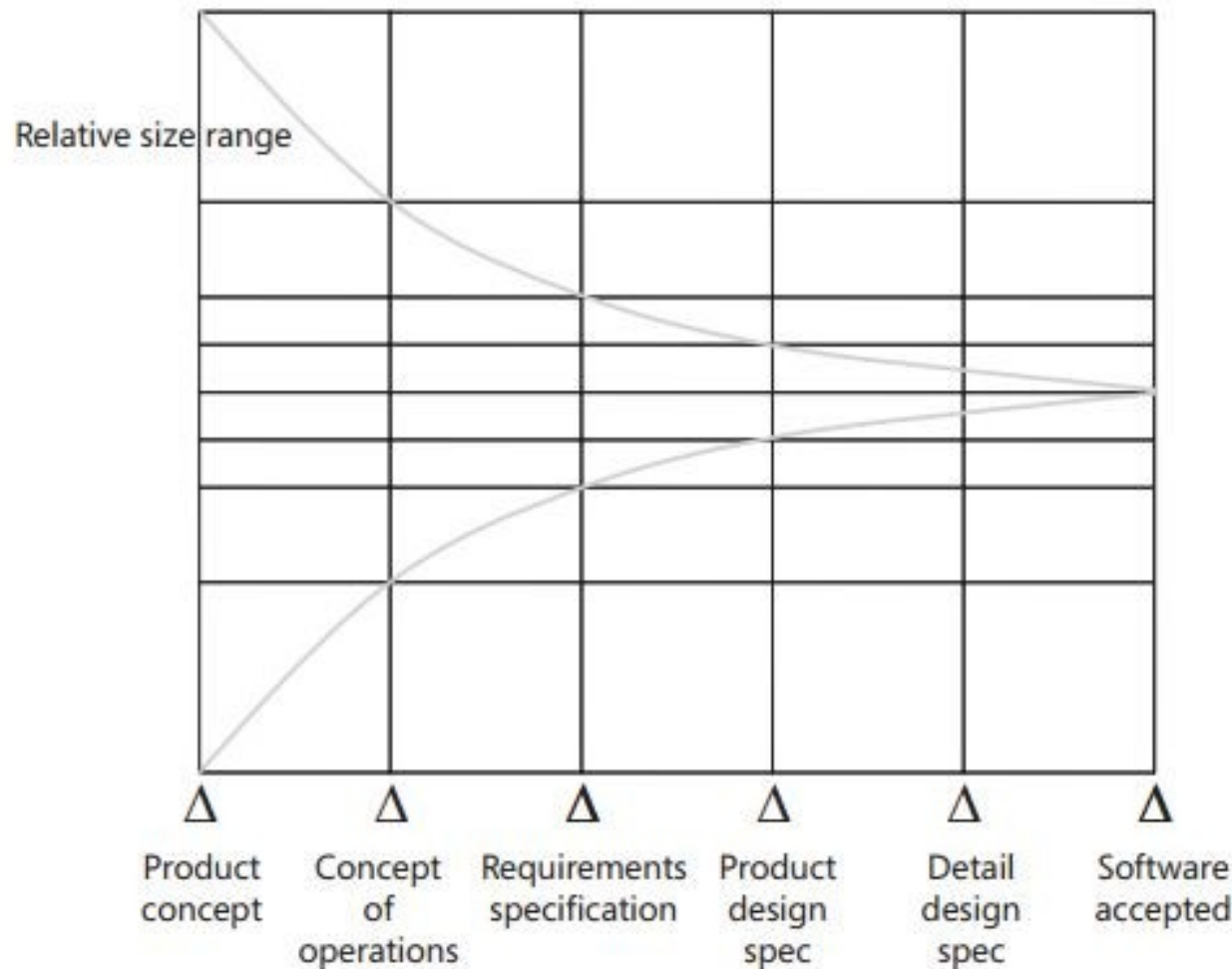


FIGURE 7-2 Variability in key estimates through the lifecycle. (Adapted from Deutsch, 2003. Used with permission of M. Deutsch.)

Promenljivost procene kao funkcija perioda trajanja projekta (2)

TABLE 7-1 Variability in Estimate Data Through the Lifecycle

Phase	Effort and Size		Schedule	
	Optimistic	Pessimistic	Optimistic	Pessimistic
Initial product definition	0.25	4.0	0.60	1.60
Approved product definition	0.50	2.0	0.80	1.25
Requirements specification	0.67	1.50	0.85	1.15
Product design specification	0.80	1.25	0.90	1.10
Detailed design specification	0.90	1.10	0.95	1.05

Source: Adapted from Deutsch, 2003, and used with the permission of M. Deutsch.

Promenljivost procene kao funkcija perioda trajanja projekta (3)

- Što je više vremena prošlo u radu na projektu, to tačnije možemo proceniti kad će se projekat završiti, koliko truda će nam biti potrebno i koliko će to sve koštati.
- Na grafu vidimo optimističku i pesimističku procenu. Kako vreme odmiče, razlika se smanjuje.

Određivanje cene i veličine softverskih projekata

- Više od 40 godina je u softverskoj industriji bilo pokušaja da se osnuju metode za precizno određivanje cene i veličine projekata.
- Nedostatak sličnosti između projekata određivanje čini težim, međutim određivanje sličnih elemenata doprinosi olakšanju.
- Veliki broj softverskih naučnika je odabralo da proučava projekte u smislu težine, rizika i kompleksnosti i da proučava veštine ljudi koji će raditi na projektu.

Određivanje cene i veličine softverskih projekata (2)

- Menadžerski aspekt je uglavnom bio ignorisan.
- Istraživanja su pokazala da tehnike procene pokazuju da je stopa greške 85% do 610% na samom početku projekta, odnosno 50% do 100% nakon kalibracije.
- Jedan od načina za određivanje jeste procena broja linija izvornog koda (source lines of code - SLOC)

Bitne tačke procene

- Procena (cene i vremena) projekta nije njegov plan, već samo njegova ulazna informacija.
- Funkcija promenljivosti bi trebalo da prati svaku procenu.
- Promena faktora kako bi se zadovoljili odgovori prizivaju nevolju.

Bitne tačke procene (2)

- Udružiti se sa klijentom kako bi zajedno delili rizike i probleme kao odrasli.
- Biti iskren - ako raspored plana ne deluje izvodljivo, verovatno nije.
- U izmeni plana projekta uzeti u obzir šta je klijentu prioritet da se završi.

Osnovni oblik formule za procenu projekta

- $E = p * S$
- E - Količina truda koja se meri u Osoba-Mesecima (Person-Month)
- p - Konstanta produktivnosti čija vrednost zavisi od autora i koja predstavlja broj SLOC-a po jednoj Osobi-Mesecu
- S - Veličina projekta koja se meri u hiljadama linija izvornog koda (KLOC)

Nalazi IBM-a

- Istraživanje IBM-a nad 60 projekata je proizvelo model sledećeg oblika.
- $E = 5.2 * L^{0.91}$
- L je veličina projekata koja se meri u KLOC-ovima
- 5.2 je skalirajući faktor

Nalazi IBM-a (2)

TABLE 7-2 IBM Early Scaling Factor Approach Values

Effort (Person-Months)	Size (Thousands of Lines of Source Code)	Productivity (Effort/Size)
36.44	10	3.64
296.18	100	2.96
2407.48	1000	2.41

Menadžment Softverskog životnog ciklusa

- Software Lifecycle Management (SLIM)
- Uglavnom korisna za projekte sa visokim stepenom truda
- $S = C * K^{(1/3)} * t^{(4/3)}$
- S je veličina projekta merena u KLOC-ovima
- C konstanta tehnologije
- K je ukupni trud meren u programer-godinama
- t je broj godina potreban za razvoj softvera

Menadžment Softverskog životnog ciklusa (2)

TABLE 7-3 Factors Used in the SLIM Method

Project Type Descriptor	Value of C
Real-time embedded	1,500
Batch development	4,894
Supported and organized	10,040

Metod Procene Funkcijskih Poena

- Function Point Estimation Method
- Ovom metodom se relativno precizno dobija veličina projekta merena u SLOC-ovima

Metod Procene Funkcijskih Poena (2)

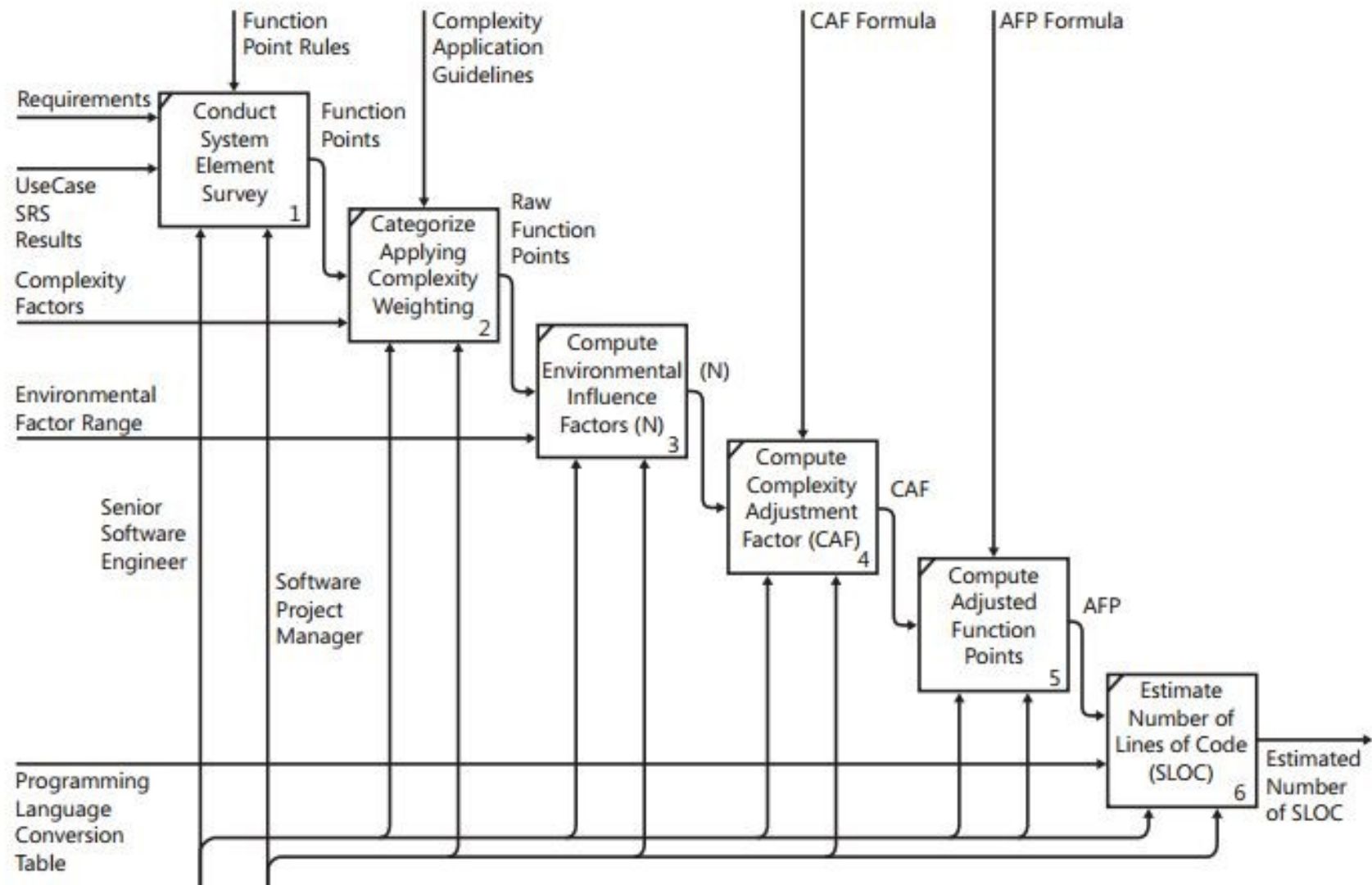


FIGURE 7-3 The function point method as a process.

Metod Procene Funkcijskih Poena (3)

- Svaka funkcija se može okarakterisati kao jedan od sledećih tipova

TABLE 7-4 Descriptions of Function Types

Type	Example
External inputs	Filenames
External outputs	Messages
Queries	Inputs from a user of an interactive system requiring a response
External files or interfaces	Files shared with other software systems
Internal files	Files not visible outside the system

Metod Procene Funkcijskih Poena (4)

- Potrebno je svakoj funkciji dodeliti kompleksnost (S - prosto, A - srednje, C - kompleksno) i na osnovu toga dobiti nepromenjeni funkcijski broj (UFC - unadjusted function count)

TABLE 7-5 Values of the Coefficients Used in Function Point Estimation

Function Point Type	Characterized As:		
	S-Simple	A-Average	C-Complex
External inputs	3	4	6
External outputs	4	6	7
Queries	3	4	6
External files	5	7	10
Internal files	7	10	15

Metod Procene Funkcijskih Poena (5)

- Iz tabele faktora kompleksnosti svakoj stavci dodeliti odgovarajuću vrednost od 0 do 5 gde 0 predstavlja stavku koja najmanje komplikuje projekat, a 5 najviše. Sabrati sve te vrednosti.

TABLE 7-6 Technical Complexity Factors for Function Point Analysis

Complexity Factors	
Data communication	Online update
Distributed data processing	Complex processing
Performance	Reusability
Heavily used configuration	Ease of installation
Transaction rate	Ease of use
Online data entry	Multiple sites
End user efficiency	Facilitate change

Metod Procene Funkcijskih Poena (6)

- Izračunati faktor kompleksnosti tehnologije koristeći formulu
- $T_{cf} = 0.65 + (N / 100)$
- Izračunati korigovane funkcijske poene koristeći formulu
- $FP = UFC * T_{cf}$

Metod Procene Funkcijskih Poena (7)

- Iz tabele konverzije FP u SLOC na osnovu programskog jezika izračunati ukupnu veličinu projekta u SLOC-u

TABLE 7-7 Conversion Factors from Function Points to Language Statements

Programming Language	Source Statements per Function Point
Assembler	320
C	150
COBOL	106
FORTRAN	106
Pascal	91
Smalltalk	21
Query languages	16
Spreadsheet languages	6

3D Metod Funkcijskih Poena

- Pošto je prethodni metod bio razvijen na osnovu izrada poslovnih aplikacija, mnogi programeri koji su radili na stvarno-vremenskim sistemima nisu smatrali da je ovaj metod njima upotrebljiv.
- Zbog toga je razvijena nova metoda bazirana na prethodnoj, a koja uzima u obzir i ostale vrste softvera.
- Zove se 3D zato što razmatra 3 dimenzije aplikacije.

3D Metod

Funkcijskih Poena (2)

- Dimenzija podataka - dimenzija preuzeta iz prethodne metode. Obraduje interne datoteke, ulaz, izlaz, upite...

TABLE 7-5 Values of the Coefficients Used in Function Point Estimation

Function Point Type	Characterized As:		
	S–Simple	A–Average	C–Complex
External inputs	3	4	6
External outputs	4	6	7
Queries	3	4	6
External files	5	7	10
Internal files	7	10	15

3D Metod

Funkcijskih Poena (3)

- Funkcijska dimenzija - dimenzija usredsređena na obrade koje se dešavaju unutar sistema.

TABLE 7-13 Semantic Evaluation Schemes for 3D Function Points Method

Processing Steps	Number of Semantic Statements		
	1–5	6–10	> 11
1–10	Low	Low	Average
11–20	Low	Average	High
≥ 21	Average	High	High

3D Metod

Funkcijskih Poena (4)

- Dimenzija kontrole - usredsređuje se na aspekt stanja stvarno-vremenskog sistema. Bitni faktori su konkurentnost, dešavanja (event), vreme itd. Vrednost se meri brojem tranzicija stanja potrebnim za određenu aktivnost.

3D Metod

Funkcijskih Poena (5)

TABLE 7-14 Complexity Weighting Factor Set Used in 3D Function Point Method

	Complexity Level			Total
	Simple	Average	High	
External inputs	3	4	6	
External outputs	4	6	7	
Queries	3	4	6	
External files	5	7	10	
Internal files	7	10	15	
Transformations	7	10	15	
Transitions	—	10	—	
Total 3D UFC				

Metod procene zasnovan na pseudokodu

- Jedan od načina za dobijanje broja linije koda zasnovan na što ranijem pisanju pseudokoda.
- Razviti tabelu koja za svaku vrstu iskaza određuje koliko je potrebno linije koda.
- Na osnovu toga dobiti ukupan broj linija koda.

Model Konstruktivne Cene

- Constructive Cost Model (COCOMO)
- Jako popularan ali nije onaj koji je najčešće korišćen.
- Od njega je nastao COCOMO II

Model Konstruktivne Cene (2)

- COCOMO je zasnovan na osnovu sledeće dve jednačine
 - $MM = a * KDSI * b$
 - $TDEV = c * MM * d$
- MM - procenjeni trud meren u osoba-mesecima
- KDSI - hiljade isporučivih linija izvornog koda
- TDEV - vreme razvijanja softvera mereno u osoba-mesecima

Model Konstruktivne Cene (3)

- Koeficijenti a , b , c i d se uzimaju iz tabele u zavisnosti od načina razvijanja.

TABLE 7-16 Coefficients Used in COCOMO

Development Mode	Value of a	Value of b	Value of c	Value of d
Organic	3.2	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	2.8	1.20	2.5	0.32

Table 7-17 Descriptions of Classification Terms Under COCOMO

Development Mode	Description
Organic	The project involves known problems (for example, inventory control); uses small teams in a stable environment.
Semidetached	Project involves broader, mixed range of knowledge and experience on project team(s) with tighter constraints than organic projects but not as tight or demanding as embedded projects.
Embedded	Development occurs under tight constraints; project is very complex with high likelihood of requirements changes (for example, device drivers, some defense systems, development for new hardware, fixed release date).

Model Konstruktivne Cene (4)

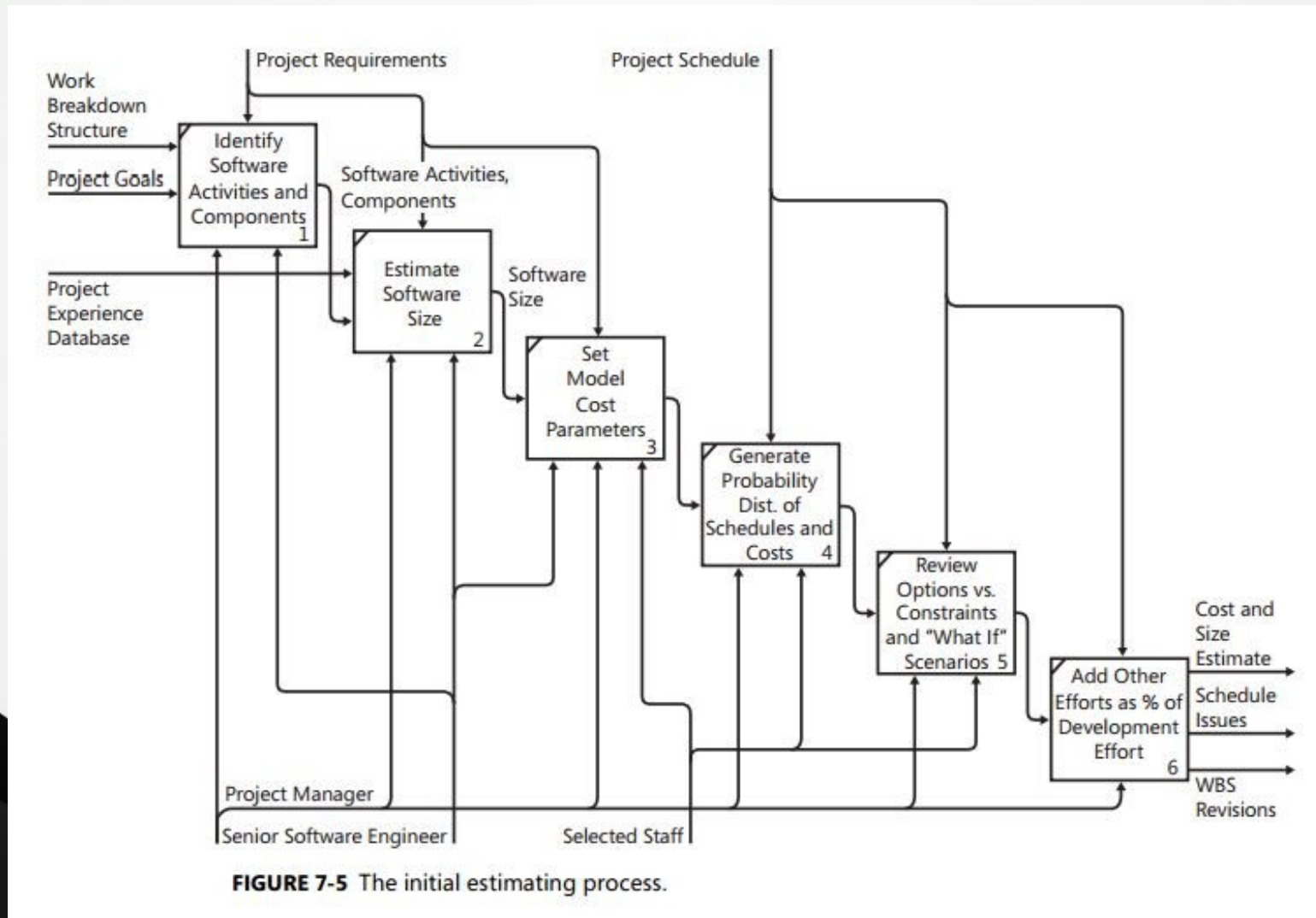
- COCOMO se sastoji iz sledeća 4 koraka
 1. Odrediti način razvijanja.
 2. Proceniti veličinu projekta u KDSI
 3. Izračunati MM
 4. Izračunati TDEV

COCOMO II

- COCOMO II tretira problem kroz 3 faze
 1. Model delova aplikacije
 2. Model ranog dizajna
 3. Post-Arhitektonski model

Izračunavanje truda kao dvofazni proces (COCOMO II)

- Inicijalna procena plana

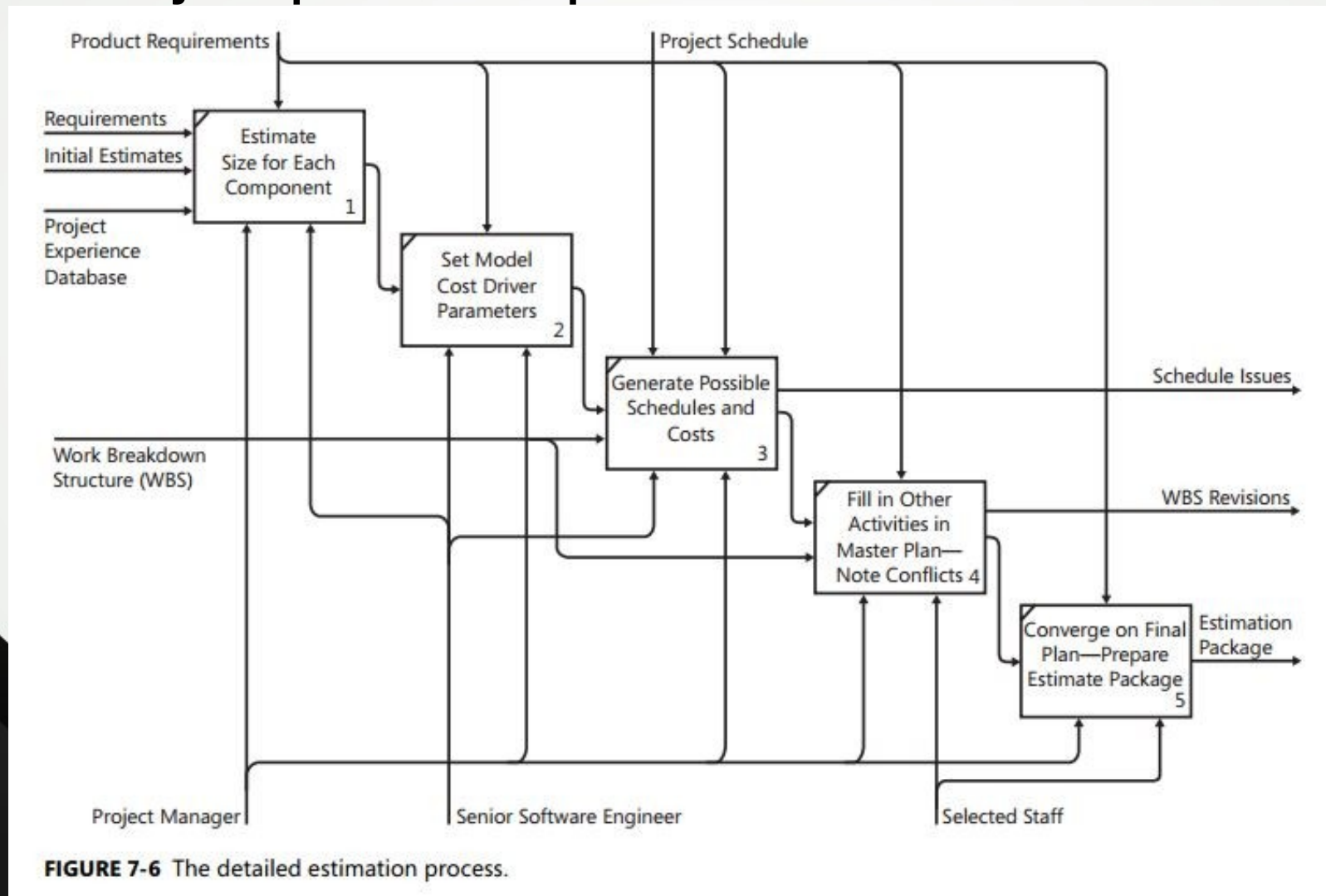


Izračunavanje truda kao dvofazni proces (COCOMO II) (2)

- Inicijalna procena plana
- Indetifikovati aktovnosti i komponente softvera
- Izračunati veličinu softvera (SLOC)
- Postaviti parametre cene modela
- Generisati raspodelu verovatnoće rasporeda i cena
- Proučiti mogućnosti u odnosu na ograničenja i "šta ako" scenarija
- Dodati ostale vrednosti truda kao procenat razvojnog truda

Izračunavanje truda kao dvofazni proces (COCOMO II) (3)

- Detaljna procena plana



Izračunavanje truda kao dvofazni proces (COCOMO II) (4)

- Detaljna procena plana
- Proceniti veličinu svake komponente
- Postaviti parametre cene modela
- Generisati raspodelu verovatnoće mogućih rasporeda i cena
- Uneti ostale aktivnosti u master plan, beležeći konflikte
- Konvergirati ka krajnjem planu i spremiti paket procene

Procena razvojnog truda (COCOMO II)

- $PM_{nominal} = A \times (Size)^B$
- PM – trud meren osoba-mesecima
- A – konstanta
- B – skalirajući faktor
- S – veličina projekta u KLOC

Metoda cene promenljivosti

- Koraci ove metode su:
 1. Odrediti cenu projekat bez ikakvih nepredvidivih cena
 2. Odrediti dužinu trajanja projekta bez nepredvidivih događaja
 3. Proceniti broj izmena u jedinici vremena.
 4. Odrediti prosečnu cenu promene kao deo originalne procene.

Metoda cene promenljivosti (2)

- Formula za računanje cene promenljivosti je
- $CV = n^2 * (f_c)^2 * (\mu_c)^2 / (C)^2$
- CV - cena promenljivosti
- n - broj izmena koje se dese u projektu
- f_c - koeficijent promenljivosti cene od izmene
- μ - prosečna cena izmene kao decimala
- C - originalna cena projekta