

Бъдещи модели за оценка на софтуера. Софтуерни средства за оценка на разходите.

Курс: NT272 Икономика на софтуерното инженерство

Автор: инж. Пламен Петков
София, 2013

Съдържание

- Метрики за размер на софтуер
- Метод за оценка SEER-SEM
- Софтуерни средства за оценка

Съдържание

- **Метрики за размер на софтуер**
- Метод за оценка SEER-SEM
- Софтуерни средства за оценка

Обектни точки (Object Points)

- Мярка за размер на функционалност на софтуер основаваща се на обектно-ориентиран поглед или дизайн на приложението
- Технологично ориентирани, но все още от високо ниво
- По-лесни правила за броене и оценка

Алгоритъм за оценка на ОР

1. Идентифициране на класовете на приложението
2. Класификация на класовете на вътрешни и външни
3. Изброяване на атрибутите на класовете
4. Изброяване на класовете наследници
5. Изчисляване на сложността на класовете
6. Идентифициране на методите на класовете
7. Изброяване на реферираните класове и атрибути от методите
8. Изчисляване на сложността на методите

Класове

- Вътрешен клас – дефиниран е в рамките на приложението (граница на приложението)
- Външен клас – дефиниран е във външни приложения и е извън рамките на приложението
- Броя се само базовите класове, не се броят наследниците
- Всеки клас наследник на базовите класове се брои за RET

Сложност на вътрешни класове

	1 – 19 DETs	20 - 50 DETs	51+ DETs
0 – 1 RETs	L	L	A
2 – 5 RETs	L	A	H
6+ RETs	A	H	H

Сложност на външни класове

	1 – 19 DETs	20 - 50 DETs	51+ DETs
0 – 1 RETs	L	L	A
2 – 5 RETs	L	A	H
6+ RETs	A	H	H

Методи/Услуги

- Inquiry service – External Inquiry
- Изходна услуга/метод (Output Service) – External output
- Входна услуга/метод (Input Service) – External Input

Сложност на методи/услуги

- Реферирани класове - броят се реферирани класове от метода – FTR (Наследените класове се броят като базов клас)
- Реферирани атрибути – броят се използваните атрибути на класове – DET
- Реферирани методи/съобщения – всяко извикване на метод на друг клас се брои, като FTR

Сложност на inquiry услуги

	1 – 5 DETs	6 - 19 DETs	20+ DETs
0 – 1 FTRs	L	L	A
2 – 3 FTRs	L	A	H
4+ FTRs	A	H	H

Сложност на изходна услуга

	1 – 5 DETs	6 - 19 DETs	20+ DETs
0 – 1 FTRs	L	L	A
2 – 3 FTRs	L	A	H
4+ FTRs	A	H	H

Сложност на входна услуга

	1 – 4 DETs	5 - 15 DETs	16+ DETs
0 – 1 FTRs	L	L	A
2 FTRs	L	A	H
3+ FTRs	A	H	H

Сравнение на FP с OP

OP	Съответствие	Сложност	Съответствие
Вътрешен клас	ILF	Брой наследници	RET
		Брой атрибути	DET
Външен клас	EIF	Брой наследници	RET
		Брой атрибути	DET
Извличане на инф.	EQ	Брой класове	FTR
		Брой атрибути	DET
Външен изход	EO	Брой класове	FTR
		Брой атрибути	DET
Външен вход	EI	Брой класове	FTR
		Брой атрибути	DET

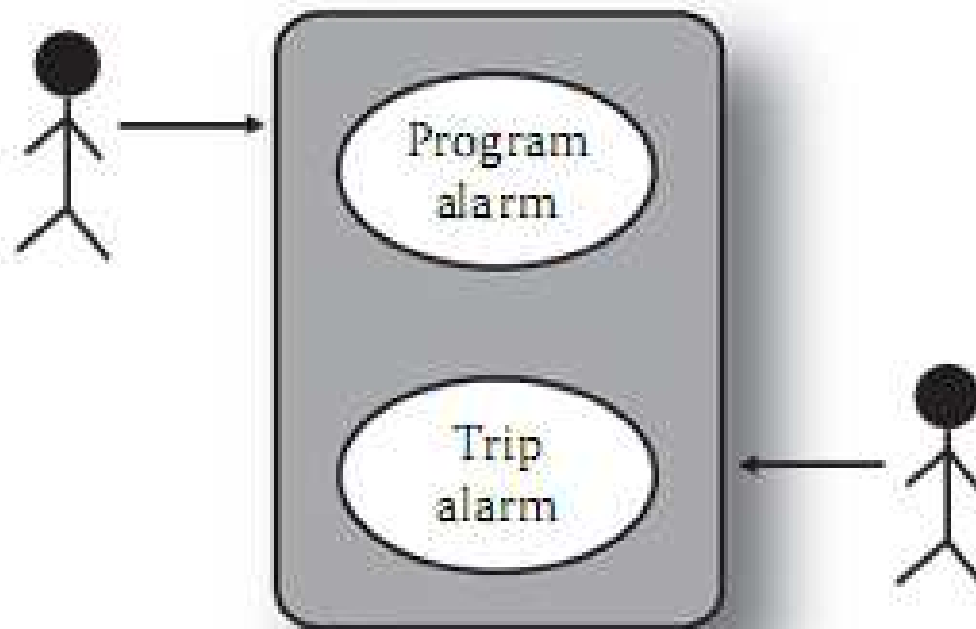
Object Points

Тип	L	A	H
Вътрешен клас	7	10	15
Външен клас	5	7	10
Inquiry услуга	3	5	6
Изходна услуга	4	5	7
Входна услуга	3	4	6

Изчисляване на некоригирани OP

$$UOP = \sum_i IC_i + \sum_i EC_i + \sum_i QS_i + \sum_i OS_i + \sum_i IS_i$$

Use-case points



Изчисляване на размер на базата на UCOP

1. Идентифициране на актьорите и UC
2. Определяне на сложност на актьорите и UC, чрез сумиране с тегловен фактор
3. Сума на всички UCOP – UUCP (Unadjusted Use-case Points)
4. Прилагане на коригиращи фактори

Тегловни коефициенти на Use-Case

Тип	Описание	Фактор	Брой на UC	Окончателен тегловен коефициент
Прост	1 – 3 транзакции	U1	N_{U1}	$N_{U1} \times U1$
Среден	4 – 7 транзакции	U2	N_{U2}	$N_{U2} \times U2$
Сложен	8+ транзакции	U3	N_{U3}	$N_{U3} \times U3$
				Тегловен Фактор за UC

Тегловни коефициенти на актьори

Тип	Описание	Фактор	Брой на УС	Окончателен тегловен коефициент
Прост	Интерфейс към външна система	A1	N_{A1}	$N_{A1} \times A1$
Среден	Интерактивен интерфейс или протокол	A2	N_{A2}	$N_{A2} \times A2$
Сложен	GUI	A3	N_{A3}	$N_{A3} \times A3$
				Тегловен Фактор за актьори

Коригиране на UUCP

$$AUCP = UUCP \times TF \times EF$$

$$Technical_Factor(TF) = 0.6 + (0.01 \times TWF)$$

$$Environmental_Factor(EF) = 0.4 + (-0.03 \times EWF)$$

$$V_i = \left\{ \begin{array}{l} 0 - \text{няма} - \text{влияние} \\ 3 - \text{умерено} - \text{влияние} \\ 5 - \text{силно} - \text{влияние} \end{array} \right\}$$

Технически фактори

Технически Фактор	Тегло	Коефициент	Изчислен фактор	Описание
1	2	V_1	$2 \times V_1$	Разпределена система
2	2	V_2	$2 \times V_2$	Изисквания за скорост на отговор
3	1	V_3	$1 \times V_3$	Ефективност (Online)
4	1	V_4	$1 \times V_4$	Сложни вътрешни обработки
5	1	V_5	$1 \times V_5$	Пре-използване
6	0.5	V_6	$0.5 \times V_6$	Лесно инсталиране
7	0.5	V_7	$0.5 \times V_7$	Лесно използване
8	2	V_8	$2 \times V_8$	Поддръжка на различни платформи
9	1	V_9	$1 \times V_9$	Поддръжка на бъдещи промени
10	1	V_{10}	$1 \times V_{10}$	Паралелна обработка
11	1	V_{11}	$1 \times V_{11}$	Сигурност
12	1	V_{12}	$1 \times V_{12}$	Достъп до модули на други производители
13	1	V_{13}	$1 \times V_{13}$	Изисквания за обучение

$$TWF = \sum_{i=1}^{13} V_i W_i$$

Бъдещи модели за оценка
на софтуера

Фактори на средата

Технически Фактор	Тегло	Коефициент	Изчислен фактор	Описание
1	1.5	V_1	$1.5 \times V_1$	Познаване на жизнения цикъл на приложението
2	0.5	V_2	$0.5 \times V_2$	Опит с приложението
3	1.5	V_3	$1.5 \times V_3$	Опит с OOA/OOD
4	0.5	V_4	$0.5 \times V_4$	Възможности на главен анализатор
5	1	V_5	$1 \times V_5$	Мотивация на екипа
6	2	V_6	$2 \times V_6$	Стабилност на изискванията
7	-1	V_7	$-1 \times V_7$	Членове на екипа на не-пълнен работен ден
8	2	V_8	$2 \times V_8$	Трудност на програмния език

$$EWF = \sum_{i=1}^8 V_i W_i$$

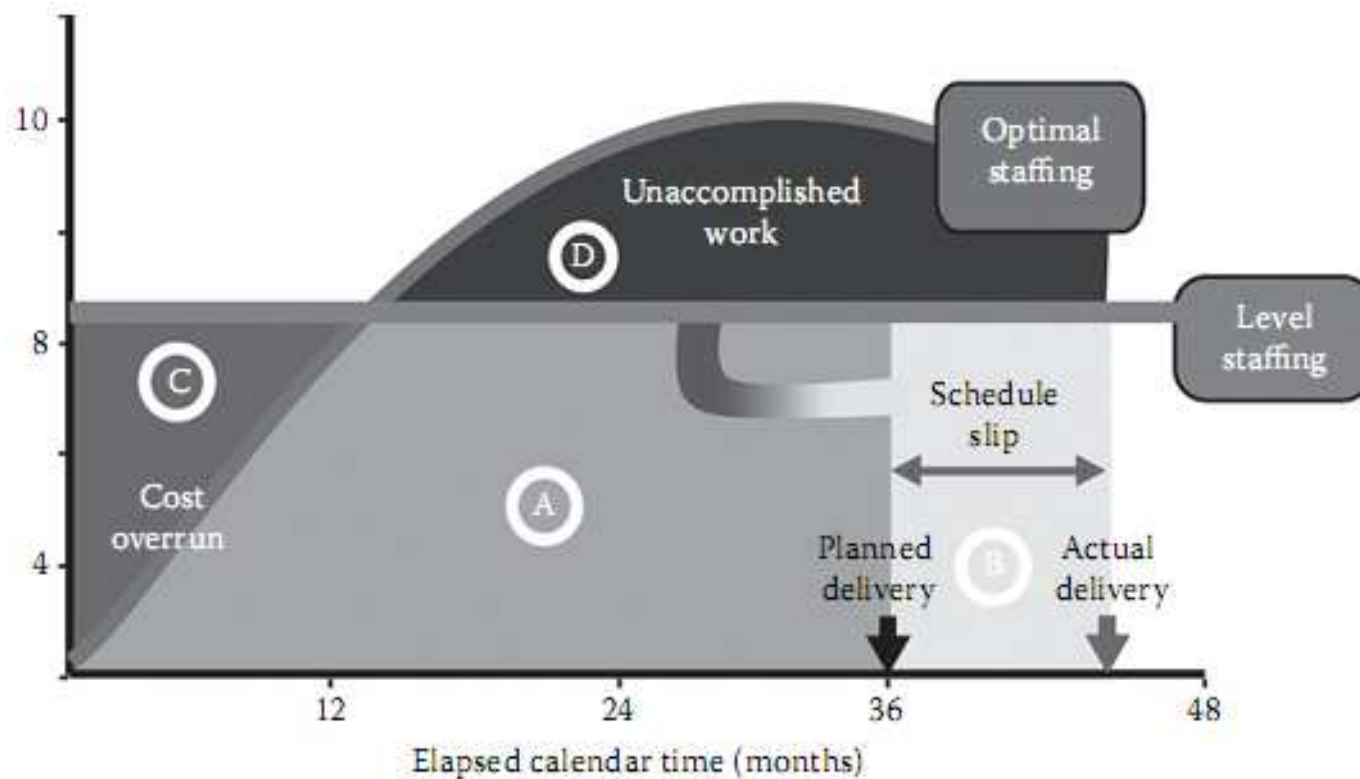
Съдържание

- Метрики за размер на софтуер
- **Метод за оценка SEER-SEM**
- Софтуерни средства за оценка

Цели на SEER-SEM

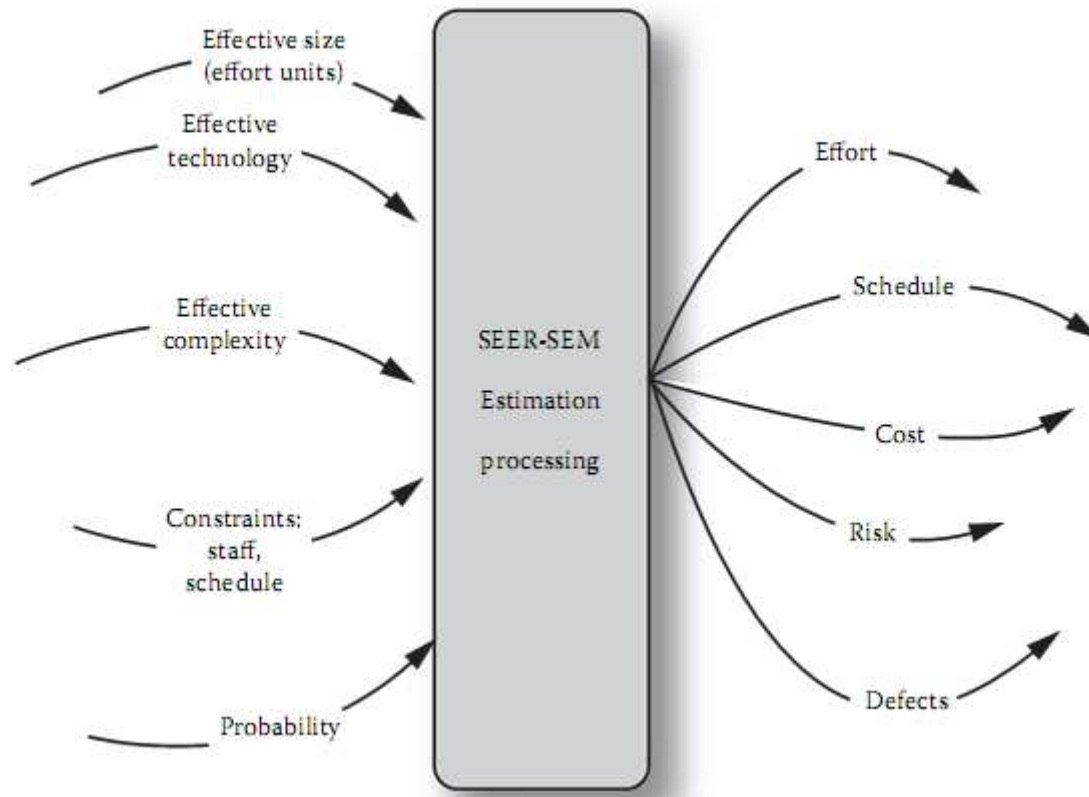
- Определяне на размер на софтуер
- Определяне на ниво на продуктивност
- Определяне на усилията за проект
- Определяне на продължителност за проект
- Определяне на усилията и продължителността на проект при наличие на ограничения
- Определяне на цена на проект
- Определяне на количество дефекти
- Определяне на усилия за поддръжка на софтуер

Използване на ограничения в SEER-SEM



(A) = Effective staffing (B) = Staffing beyond plan (C) = Overstaffed (D) = Understaffed

Входни и изходни данни за модела SEER-SEM



Форма за входни данни

Create/Modify WBS Element

Description: GUI
Analyst: J. Smith

Element Types
Rollup Program Component COTS Unit

This Item Is:
Level 3

Platform: Business and Non-Critical MIS
Application: Database
Acquisition Method: General - new and pre-existing
Development Method: Spiral
Development Standard: Commercial Low
Class: No Knowledge

Sizing
☒ Lines
☒ Functions
☒ Others:
Mark II Function Point /
Metric - Fast Function F
Metric - Function Base
Metric - Source Lines of
Metric - Unadj Function
none

Modified
Date: 9/15/05
Time: 10:51:50

OK Cancel
Insert Next WBS

Форма за входни данни

- Платформа – MIS, ERP, mission...
- Приложение – транзакционно, база от данни...
- Метод на доставка – нова разработка, модификация на съществуващ...
- Метод на разработка – RUP, Web...
- Стандарт за разработка – ISO9000, J-016...
- Размер – SLOC, FP, OP, UUCP, екрани...

Цели SEER-FBS

- Да позволи оценка на софтуер без специално обучение за FP анализ
- Да направи разбивка на общата цена на проекта по отделни важни за крайния потребител елементи
- Да улесни оценяването на проекти за научната област и вградени системи

Особености на SEER-FBS

- Формира обхват на оценка, а не една конкретна оценка
- SEER-FBS е апроксимация на FP анализа
- Прилага се изключително в началните етапи на проекта

SEER-FBS External Inputs

- Входни екрани
- Интерактивен вход
- Хардуерен вход
- Пакетни входни потоци

Оценката на сложността на SEER-FBS е същата, като на FP анализа

SEER-FBS External Outputs

- Екранни справки
- Печатни справки
- Външен медиен изход
- Външен софтуерен изход – формати на данни
- Външен хардуерен изход – сигнали и съобщения

Оценката на сложността на SEER-FBS е същата, като на FP анализа

SEER-FBS External Inquiries

- Заявка/Отговор
- Менюта
- Контекстна помощ
- Вградена комуникация с устройства

Оценката на сложността на SEER-FBS е същата, като на FP анализа

SEER-FBS External Interface Files

- Реферирани данни
- Фиксирани съобщения
- Споделени файлове с данни

Оценката на сложността на SEER-FBS е същата, като на FP анализа

SEER-FBS Internal Logical Files

- Групирани данни на приложението
- Таблици с данни
- Файлове на БД

Оценката на сложността на SEER-FBS е същата, като на FP анализа

SEER-FBS Вътрешни функции

- Internal Functions – отразява обработката на данните в рамките на приложението без информацията да пресича границите на приложението

Сложност на вътрешни функции

<i>Complexity</i>	<i>Function</i>
Low	Sorting routines
Average	Reasonably complex functions such as commercial data compression algorithms
High	Signal processing or data reduction algorithms; other functions of high logical complexity

Дърво на решения за SEER-FBS и FP анализ



Основно уравнение за размер на софтуер в SEER-SEM

$$S_{\text{eff}} = S_{\text{new}} + S_{\text{exist}} * (0.4 * S_{\text{redesign}} + 0.25 * S_{\text{reimpl}} + 0.35 * S_{\text{retest}})$$

S_{eff} – ефективен размер, EU

S_{new} – нова разработка размер, UFP

S_{exist} – съществуващ софтуер размер, UFP

S_{redesign} – размер пре – проектиране, UFP

S_{reimpl} – размер промяна, UFP

S_{retest} – размер пре – тестване, UFP

Изчисляване на ефективен размер на софтуер от SEER-FBS

$$S_e = L_x * (AF * UFP)^{\frac{Entropy}{1.2}}$$

S_e – ефективен размер

L_x – фактор за езика

AF – фактор за корекция

UFP – некоригирани ФП

$Entropy$ – 1.04 до 1.2

Изчисляване на усилия за проект в SEER-SEM

$$E = D^{0.4} * \left(\frac{S_e}{C_{te}}\right)^{1.2}$$

E - усилия в човекомесеци

D - скорост *на добавяне на нови ресурси*

S_e – *ефективен размер*

C_{te} – *технологична ефективност*

Изчисляване на продължителност на проект в SEER-SEM

$$T_d = D^{-0.2} * \left(\frac{S_e}{C_{te}}\right)^{0.4}$$

T_d - продължителност в месеци

D - скорост *на добавяне на нови ресурси*

S_e – *ефективен размер*

C_{te} – *технологична ефективност*

ОСНОВНИ СЪПКИ

1. Определяне на обхват и цел на оценката
2. Определяне на предпоставки, техническа база
3. Дефиниране на мерни единици, ресурсни категории и данни
4. Определяне на размер на проекта

ОСНОВНИ СЪПКИ - оразмеряване

Parameters - Component: Analysis and Query Tools			
- LINES (Classic)			
New Lines of Code	0	0	0
- Pre-exists, not designed for reuse	319	1,283	4,161
Pre-existing lines of code	5,555	6,666	7,777
Lines to be deleted in pre-exstg	0	0	0
Redesign required	5.00%	10.00%	40.00%
Reimplementation required	1.00%	5.00%	10.00%
Retest required	10.00%	40.00%	100.00%
+ Pre-exists, designed for reuse	0	0	0
Function Implementation Mechanism			
3rd Generation Languages			
- FUNCTIONS (Classic)			
- NEW			
New Functions	50	60	90
Software phase at estimate		Requirements	
- Pre-exists, not designed for reuse	0	0	0
Pre-existing functions	0	0	0
Funcs to be deleted in pre-exstg	0	0	0
Software phase at estimate		Done	

ОСНОВНИ СЪПКИ

5. Определяне на технологични фактори
6. Изчисляване на усилията за проекта
7. Изчисляване на продължителността на проекта
8. Налагане на ограничения и преизчисляване на продължителност и усилия

Съдържание

- Метрики за размер на софтуер
- Метод за оценка SEER-SEM
- **Софтуерни средства за оценка**

Обзор на софтуер

Софтуер	Коментар
ACEIT	Parametric Life Cycle Hardware
COCOMO Interactive	COCOMO II
Construx	Parametric Stochastic COCOMO II FREE
COSMOS	COCOMO Parametric Function Point Putnam Method
COSTAR	Parametric COCOMO II Incremental COCOMO Database
CostXpert	Parametric Stochastic System Dynamic Knowledge-based Database

Обзор на софтуер (продълж.)

Софтуер	Коментар
PRICE-S	Parametric CER Life Cycle Link to MS Office Database
Estimate PРо	Parametric COCOMO II Putnam Method Monte Carlo Simulation
SEER-SEM	Parametric Link to MS Project Database
SLIM-Estimate	Parametric Life Cycle Database Link to MS Office

Основни методологии

- Putnam – всички успешни проекти следва един и същ модел, изразим математически с експоненциални уравнения
- COSOMO II
- Monte Carlo – модели за симулация на поведения при неопределеност и несигурност

SLIM (Putnam)

- Основава се на SLOC и модел на разпределение на персонала в проекта (модел на Raleigh)
- Основното уравнение на Raleigh свързва усиλιето в проекта (Total Person Months TPM) и размера в SLOC с необходимото време (Required Development Time RDT) до константата Technical Constant (TC):
- $TC = SLOC * TPM^{1/3} * RDT^{4/3}$
- TC от 200 за “зле” до 12000 за “отлично”.

ESTIMACS и SPOR

- SPOR/20 групира FP в класове за сложност основани на Алгоритми, Код, Данни, като всеки клас си има тегловен коефициент и оценка.
- ESTIMACS, фокусира се върху бизнес фактори. Въвежда понятието “Измерение на оценка”. Измерения са човеко-часове, брой персонал цена, хардуер, рискове

Базирани на динамика

- Акцентира върху това, че условията на проекта постоянно се менят и влияят върху усилията за разработка, която се счита основен компонент в цената на проекта.
- Оценката не следва да бъде статична основана на една снимка от проектни данни, а по-скоро основана на много снимки на проекта в различни ситуации

Регресивни

- Полезни са когато има значителни исторически данни
- Техниките включват стандартна регресия, устойчива регресия
- Използват променливи проектни данни за калибриране на модели, като COSOMO, COMOMO 2, SLIM

Калибриране

- Параметричните методи могат да доведат до неточни оценки
- Основен механизъм за подобряване на точността на оценката е калибрацията
- Необходима е база данни с информация за проекти

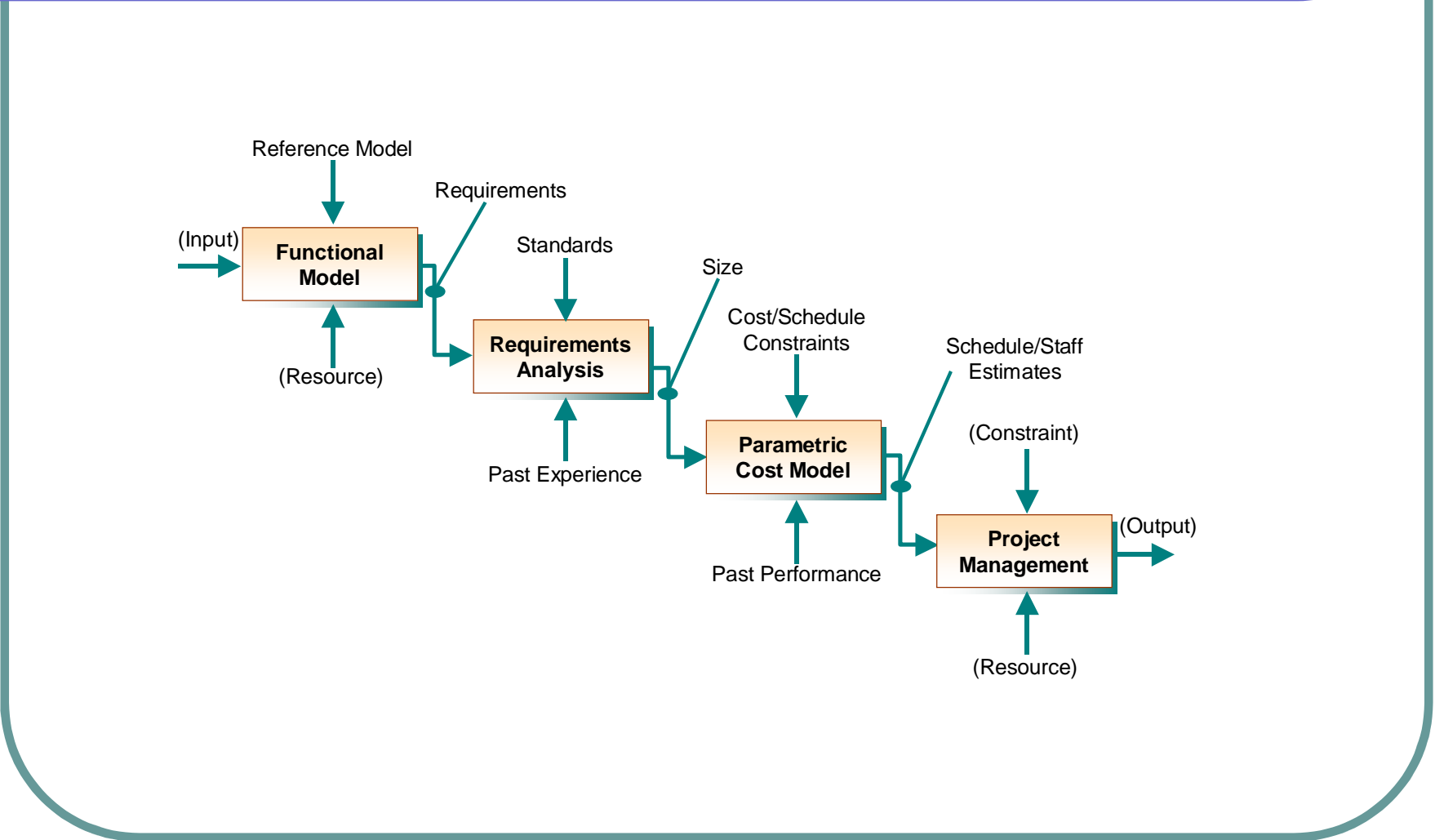
Използвани методологии за разработка при оценка

- Еволюционна
- Инкрементална
- ООП
- Протипиране
- Спираловидна – повтарящи се етапи и дейности извършвани с нарастваща детайлност
- Водопад – етап след етап, дейност след дейност

Обхванати стандарти от софтуер за оценка

- ISO9000
- SEI-CMMI
- Правителствени регулации в САЩ

Процес на оценка - модел



SEI CMMI и оценката на софтуер

- Software Engineering Institute (SEI) Capability Maturity Model (CMM) обхваща дейностите по управление на софтуерни проекти- *Key Process Area (KPA)* за maturity Level 2.
- Оценката на софтуер е фундаментален елемент от управлението на проектите и се използва за планиране и мониторинг на проекти.
- Ability to Perform (associated with KPA PP, Goal 1) – “*The software managers, software engineers, and other individuals involved in the software project planning are trained in the software estimating and planning procedures applicable to their areas of responsibility*”

Рискове от грешна оценка

- Разочарован клиент, загуба на бизнес.
- Във fixed-price проекти, загуба за компанията или загуба на престиж.
- Промяна на срокове, персонал и поява на нови допълнителни дефекти.

Управление на риска от грешна оценка

- Добри практики:
 - Използване на няколко метода за оценка.
 - Обучение на персонала в методи за оценка.
 - Грубите оценки са наистина груби— $\pm 35\%$
 - Оценките най-често са подценяващи отколкото надценяващи
 - Използвайте максимално детайлна WBS.
 - Преглеждайте предположенията на различните участници.
 - Използвайте предишен опит на хора и организации
 - Тясна връзка с програмистите
 - Опресняване на оценката с времето

Въпроси и отговори



Контакти

инж. Пламен Петков

E-mail: su.project.management@gmail.com