

11/02/2020

★ Very costly to fix errors in requirements specification ★

It effect everything (design, implementation; testing)

### Design

Abstraction: Modular programming, ayirma.

### Implementation

Traceable → <sup>bilgi dökümeli</sup> <sub>değerlendirme</sub>

Traceable → değer dökümeli - uygun göstergelerin yapımı

### Testing

★ full testing is not possible.

Testing can be used to show presence of errors not their absence

### Maintenance

Plan Driven VS Agile Driven Processes

Kritik yılamlar → Plan Driven

Değerlendirme yok → Agile (rapid) Driven

### The Waterfall Model

- ★ Ondan sonra birinden sonraki fazı geçme. Bir önceki fazın geçerlik mevcut değil.
- ★ İkinci kere hatalı.

11/02/2020

- Iterative dev.
- Plan-Driven
- Problem: User br sizes, Sourcing, geo. aliyors,
- 1- Requirements (No change in requirements)
- 2- Testing is very late

**NOT:** Testing'in öncesi problem lehannisini test etmenin yolu; **Review**,  
Müstakile Tepkileri, SRS (complete, consistent, nonambiguous) test edildi.

## Document Driven

Achieves

1- Maintainability

2- Traceability

3- Visibility

## Prototype

- Gain experience

- Clarify requirements

**NOT:** Sayıdən önce Waterfalllik yapılırsın.

## Incremental Model

1- Problemleri parçalar

2- İsteklerin sıralılıklarını

11.02.2020

SDE 7.15

3-Core part (Increment 1) is tested the most.

En önemli bilimin en fazla test edilmesi iyi.

4-Plan-driven

5-SRS'n littigi yar projenin sonu.

6-Integration early

+ Iterative

Problem: Assign requirements to increments

(Zor)

Aynı ister birden çok increment ile farklınesini.

## Evolutionary Models

1-Prototyp, Waterfall gibi alternatif. Daha uygun, kullnır.

2-Iterative

3-Plan-driven

## Risk

Mitigation; Bütince cihazlar.

Avoidance; Olayların ötem etme.

## Hybrid Process Model

\* Combination of process Models

11/02/2020

## 1-Spiral Model

\* Iterative

\* Risk-driven

\* 1.faz, 4. olsun.

**NOT:** 2. dene, birinci işli: Performance optimization

**ÖNEMLİ NOT:** Cihazın once SRS yapısına göre 3. faz JRS'den  
Kod yazmak şartıyla 3. faz, evrensel ne de olsa supradanışır.

\* Risk analizi çok önemli.

Bar-yonı

\* **NOT:** Spiral modeli, 4. fazı olsun **\***

\* Sırtılıcı yazılımlar kullanılır.

**NOT:** Modelin amacı ve desenlerin (2'nci tane), kullanım yerleri  
SIRAVERİ ÇIKACAK. (modeli de serede).

## WEEK 3

Step 3 in spiral model: Realization

XP-subtitle

Scrum + high level - management

18/02/2020

Orta 4/10

## Agile

★ Changes in these days for business is inevitable and rapid

★ 4 values for Agile much embrable. (Agile Manifesto)

1- Working software

2- Individuals

3- Customer collaboration

4- Responding to change

★ Moving fast, respond fast. (Agility)

★ Conventional method → SRS

Agile → User story (daha az detaylı)

★ Short life plan

★ Coding and design together

Refactoring: Aynisizlik: data tablo yapısının değişmesi  $(x^3y^2) = (x-y)(x+y)$   
explicit; your terms

↓ Simplifying today's problems will be easier tomorrow. (Easy to manage)

Face-to-face communication → Agile

Documentation

→ Conventional Model

Conveying of Information

18/02/2020

Sustainable development → Sürdürülebilir (Güvenlik için sürdürülebilir)

Up-front work (huge documentation)

Self documented code → Agile

Kritik sisteminde Agile kullanılır.

## Extreme Programming

Agile method

Reinvented software - strict base methodology

Pair programming → 2 tırnak cyn, bilgi paylaş, birbirlerini öğrenir  
collective ownership → Projekti her kişi tercih eder

Test driven

Ölçü değişiklikleri gerekçe değil, sınırlı olası.

Once test (Clarity requirements)

Test driven

\* Criar mockup bir istenilen enin oluştur → Prototip yapınız.

XP → grey box (black and white box)

CRC → Class name (Module)

Responsibilities (Modülün neyi yapması istenilen)

Collaboration (Modülle ilişkili diğer modüller)

over loaded : remove

18/02/2020

18/02/2020

Stand up meeting : ciddi toplantı

## WEEK 4

25/02/2020

### XP Coding

Pair programming advantages : Check syntax, online review, learning from each other, motivation, rapid feedback  
Heras bir kişi her zaman sorumlu hesap verebilir (accountable)

Collective ownership

integration machine or, github

Integration often: kütüphane kütüphane bir kişi test ederse sit.

for plan-driven → documentation (collective ownership)

(<sup>XP</sup> → communication) (Test driven)

Refactoring during testing

More people control: pair programming (şartlı hizmet) [Need Help]  
Telli, Change Pair

Test automation; automation design. (Test automation tool)

\***NOT:** XP'den testing'in önemi bir 2-3 taneini bl. Sıktır. \*

### Advantages of Pair Programming

- 1- Alternatifleri doğrudan karşılaşıp, Spiral en kısa alternatifleri değerlendirebilir.
- 2- Automation (test toolbars) : her.

25/02/2020

## Agile Project Management

↳ On time and planned budget

### Scrum

Management ağırlıklı, XP tabanlı, tıbbatlı, doğrudır.

Three phase:

1- Planning

2- Sprint cycle

3- Project close (Geleceği, geleceğin kontrolü)

\* Product Owner; Internal customer, Gereklilikleri söyleyen kişi.

\* Development Team; Kullanacak olan teknoloji hakkında bilgi

Spesifik teknoloji, tasarım yok.

\* Scrum Master; This process happens as smoothly as possible.  
(facilitator) Scrum'ı iyileştiren kişi.

Product backlog; List of things to do. (Product Owner, Scrum Master'in yanında bulduğu)

Sprint backlog; (Cycle içinde) items to do. (Development Team yapımı)  
(Sprint)

Sprint output; Potentially shippable software

Planning; Selecting tasks from the Product Backlog  
Dev. Team içinden "Pencere" şeklinde brainstroming.

25/02/2020

Sprint'in definition of Done önemlidir.

Definition of Done, Sprint'in kavram belirleme.

\* Regresyon testi: Yeni eklenen işin öncü yetisi yarayı borusu in die türk testlerini gecelijse yaratır.

Definition of Done vs. Acceptance Criteria

Süreç (Workflow) (sil)  
İkresim (Sprint) Hedefi  
Diagnostik (Hata tespiti)  
Item bitirme

Bölge sorumlusu

Retrospective: catman, implement, sevgili.

Booking

Acceptance Criteria

2 SINAV

Agile Mühendislik dellileri.

WEEK 5

03/03/2020

1. İlk bir önceki retrotaki hedeflere back, requirement, functional  
Ör: oda carce, (hod grebilgen)  
Sistem genelini iletimdeki se nelliklerde en çok requirement, nonfunctional

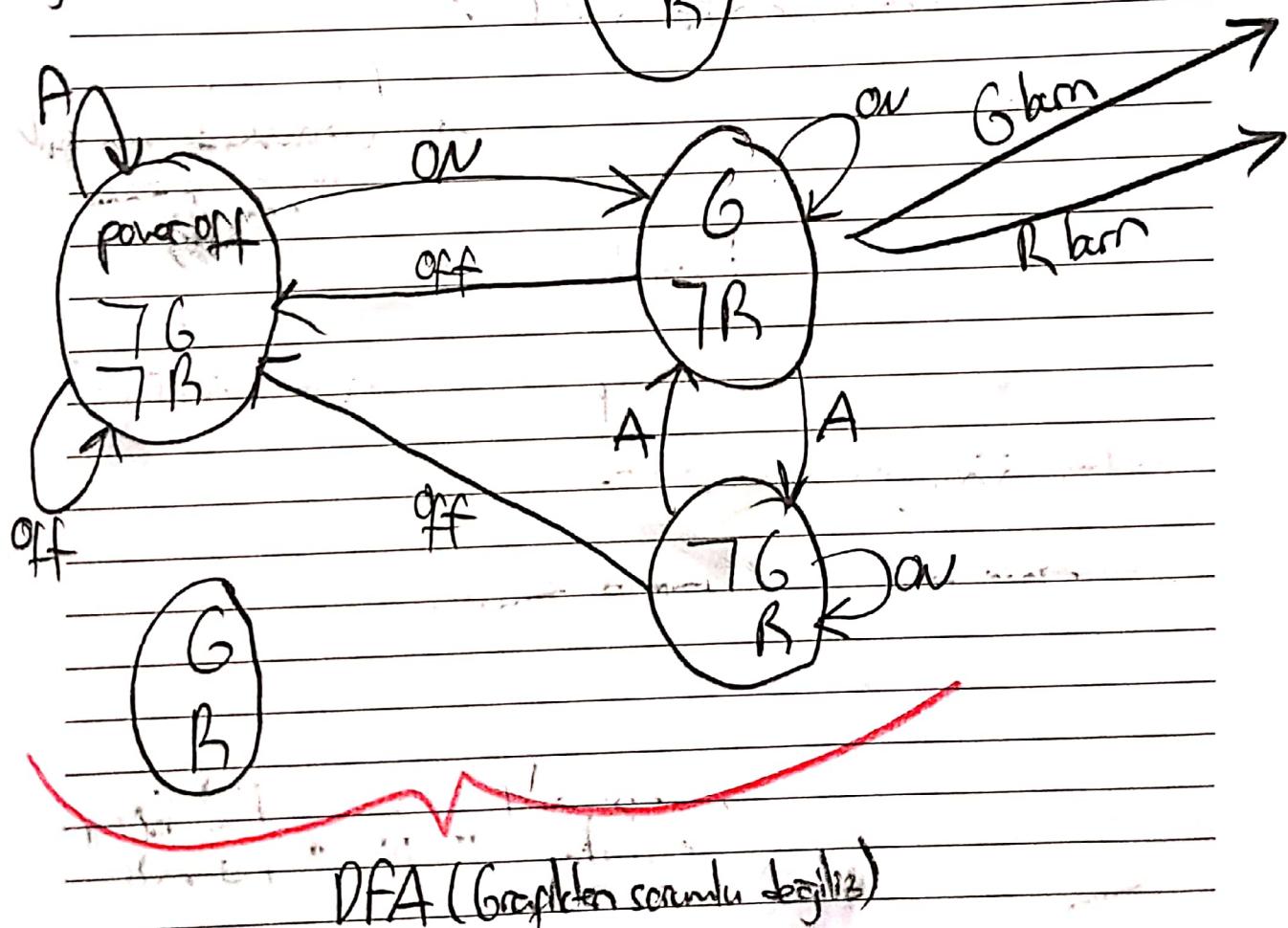
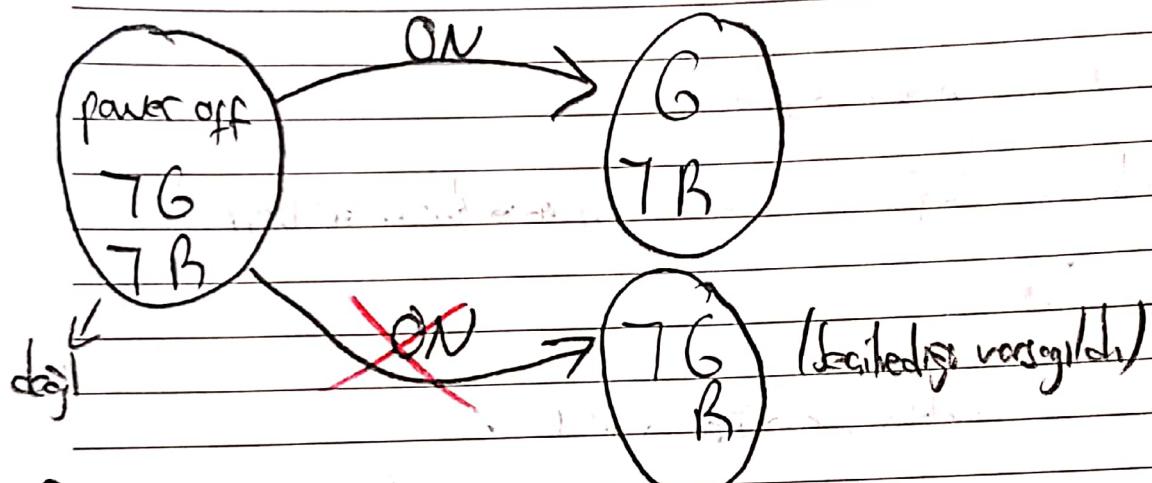
Ör: efficiency, easy to use, reliable (-ibilitiy, changeability, reliability)

Hedefe göre nüfuslu bir bölgeye göre çok requirement, domain  
özellikleri şubeler.

NOT: Behavioral = Functional

03/03/2020

Örnek: İlk sırada statik şaltı mı kumru mu belli değil.  
 Yani hangi türden ne olacak?  
 Beşinci hafızan basitçe ne olacak?



03/03/2020

03/03/2020

**NOT:** Nedenle nonfunctional requirement gösterilmesi, functional gösterilmesi için.

Structural perspective: Sistemde ne var? Neler var? Hiyerarşide? Örnekleri sisteme gösterin.

Function Oriented Problem Analysis

FAP: Data Flow Diagram (Fonksiyon içində data akışını gösterir)

### İLK ÖDEV.

\* Transformation; functions, processes (trans.)

\* Okunabilecek sistemdeki verileri data → (istemdeki ismiyle)  
SINAV için de  
(NAME)

\* functionlarda Verb kullan. (NAME)

\* External entity: Sistemin dışındaki sistem inputu veya outputu

\* File: file, file, database

**NOT:** Vericinin kodu: transformationda grafiği.

Level 0: context diagram

**NOT:** Diagram'ın içlerindeki her bir entity (entity özniteliği) her ayrı yerde gösterilebilir. Entity adları unique olmalı. Örnekte, ilıca ve restoran aynı entity

**NOT:** Oluşturulan transformation olmaz.  
ve "gitmez"

03/03/2020

★ file'da file'a datagrams, file'da file'ın içinde transfer edilebilir, diğer diğer datagramlar gizlilikte.  
External entity file'a algılanır.

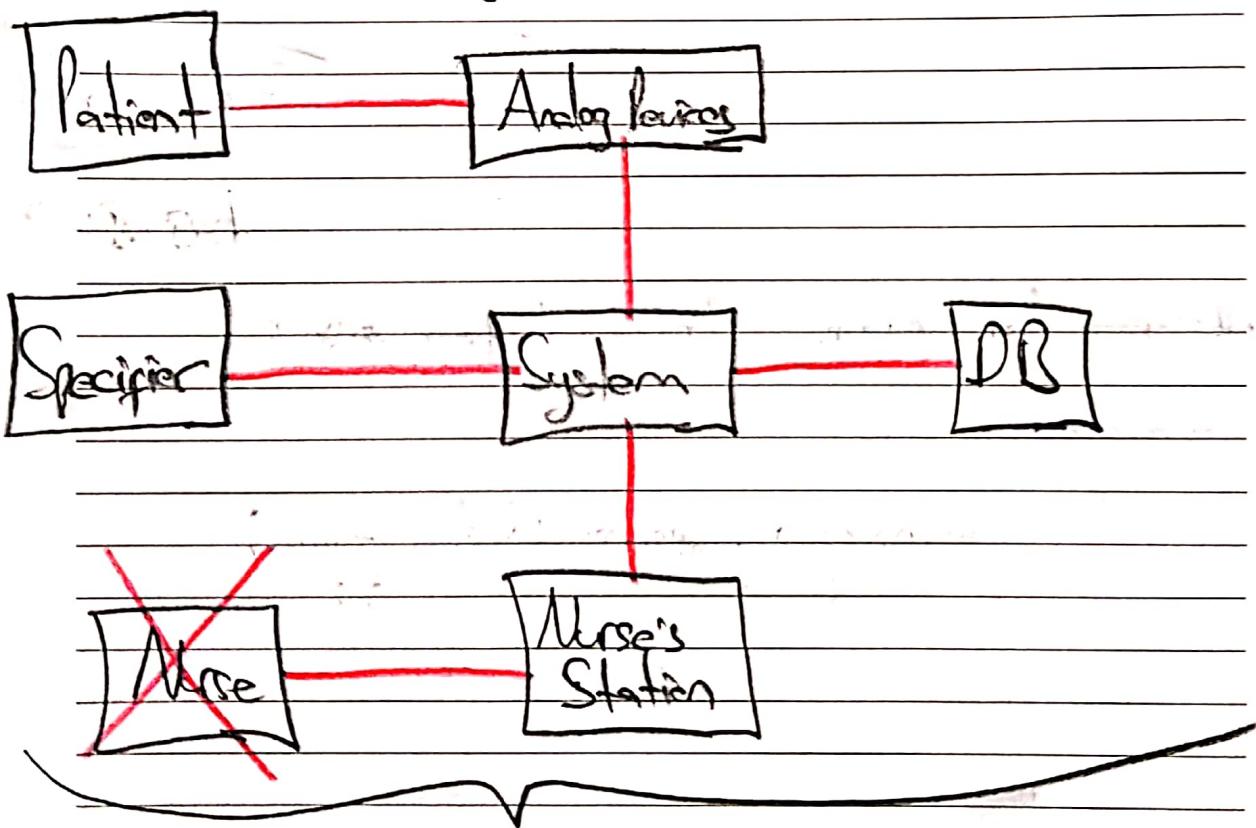
★ External entity'ler var: db, printer, workstation, client, server, ya da implement ettiğiniz.

Örnek, Paypal external entity.

★ 1. oturum sağlayıcıya sadece transfer etmek istenir. ★

NOT: DFD, tem requirement hem design için kullanılabilir.

Domain = external entity



Context Diagram (Jackson)

Büyük renk gösterilen var. → Context Diagram

03/03/2020

Symbols and on Example staff re concerning locs istenisi.

LOCS staff from demand side.

★ Sales X : Order filling detail (Marketing staff)

All Divisions X : Dizayn graph visitors (Administration staff)

NOT: PFD problem analizi için bulunulur.

**SRP:** Workload basis.

Herkesin aynı işi yapması.

BirlikteConflict resolution yapı.

Transpondanlı birşirket. 1 adet O döşke.

NOT: Administration bl.

## WEEK 6

10/03/2020

★ Model şirket kurardığımız notları şık etmenizi sağlayır.

### Software Requirements

- description ... → functional requirement (System services)

- operational constraints → nonfunctional " (Constraints)

### Requirements from customer (user)

★ Tarihi

★ Musteriler

★ Kedyzen } low level requirement

★ Test upon

★ Yöneticiler } high level requirement

10 / 03 / 2020

★ Who's the requirements and what purposes **SİNAUDA GİKAÇAK** ★

Skylt: Who Uses Requirements?

★ Architecture, incremental model, XP' de laste yordugu. Matikli.

★ Functional, Non-functional, Doriga ayrica şartlı cümlelerde sorulur. ★

Kod yapmak      Sistemin gerekini      Alıcı olduğu alımbılgılı  
Kullanılmak      ile birlikte      bilgiler verilipasa

**SİNAUDA GİKAÇAK,**

Ambiguous'lu terminoloji:

- 1- Prototype
- 2- Review by laste insabır (bad graph for li)
- 3- Test

★ Gözleme nonfunctional requirement ile functional requirementlerden daha farklı.

Nonfunctional Req.

- 1- Product Req.
- 2- Organizational Req.
- 3- External Req.

10/03/2020

★ Sınırda cümlelerde "easy to use" gibi genel, olumsal, çift anlam gelebilerek cümleler ve de lojine grifler var mı?  
Önden sırın sorusuna çok. ★  
**SINAUDA ÇIKACAK!**

★ Robustness: Normal durum basılırla sistem nasıl etkilenir.

★ Portability: Sistemin başka bir environmenta taşıması.

★ Portability sağlanan en kolay yolu: Isolation Layer

★ Monte Carlo tulumda sisteme bittiğinden beri hata var mı problemi.  
**SINAUDA ÇIKACAK!** ★ Slaytta önde var.

Reliability requirementları okulabilir hale getiriliyor.

20 inserted legs

Bütün 60 hatalı 12'si inserted buginden.

20 12

$\times$  60

$\underline{\underline{x = 120}}$

$100 - 60 = 40$

$40 - 8 = 32$  bittiğen leg sayısı.

Sonuçlu Elde Edilen (Monte Carlo sonuçları)

1- Sayıları sortigu

2- Sample size

3- Buglerin birbirini maskelemesi

24/03/2020

## Chapter 5 - System Modeling

: Order → data (Variable yetki)  
[place] → condition

Variable Name: Type | Type = Class = Module

## Activity Diagrams

## Sequence Diagrams

||; active

response: ← --- (dashed line)

## Alt, Altercation (Choices)

\* Interaction between participants (Show)

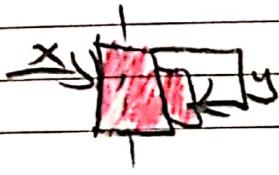
\* Selection of program yoldabilir. Sequence diagram'dan.

class A {

:A

x(); }  
y(); }

y(); }  
... }



}

26/03/2020

★ Flashcard indiraktion methodu net return.

Messages exercise. (verb, function have abstract.)

### Class Diagram

1..\* : at least 1

1..4 : 1, 2, 3, 4

Erkileşen.

NOT: Straße → gibi syntaktic cümləyəcək \*

### Best Video Exercise

Sınıf deseni → hətəhmə. Vütbəyinə deyim dəlini həssəm  
dələk təqib.

### Generalization (Prog. Lec. - Inheritance)

31/03/2020

#### Generalization

public } : Herkəs tətbiq olur  
private } method : Kimsə gəzənə tətbiq olur. (d. Helper method)  
protected } : Inner iş. Bobbəd hərəkət nəzərdən mali.  
or,

+ public

- private

± protected

+ registered

NOT: Attributes ya private ya protected olub, public olmur.

31/03/2020

Class Diagram / State Diagram

Sequence " : Interaction "

Behavioral Models

\* DFD is not UML because UML is next objectified.

Activity Diagram.

State Diagram



Sequence, Branch, Hebben den sek.

\* State Diagram is developed then.

Nesil okuyucular bilgiler. \*

\* SRS says 1-2 questions  
in exam.

Graphical Models part 2

Requirement Engineering

\* Structured scenario = Use case

\* Multiplication

\* High process rate / bulk!

Affected requirements for robots to xyz.

\* Requirements analysis

— / — / —

24 saatlik son. → DFP (detyl PFD sadece) 10. prototip.  
Square degam, hrgs sertolu.  
Senyos yaribat.

Vic Case: Hıjazda hukuki bir sistemi buluyor.

0506 939 9990 → WhatsApp grup

Requirement Engineering'li 27.5kgs added.



a) Vic Case me

\* 50-55 cm stand yet. Requirement Frequency std  
IEC62312-100-1998 stand yet. \* SRS std

**NOT:** Performance = Non-functional requirement

\* SRS'in karakterleri örenli \*

Unambiguous

Consistent

Verifiable

3D circa son ve.

10 multiple docke

5 start moves (list)

Ver son requirement sistemi girl de (döndür, nonfunctional  
veriyelle, unambiguous)

14.04.2020

## Formal Specification

Proposition: öneme

Predicates: Yatılıt önerme, Variable'a degratbyne Predicte,  
Proposition olur.

Predicale, degratbyne, True, False olur.

Set = capital

Slide 22, R<sub>4</sub> → total function  
R<sub>5</sub> → partial .. Set → Customer  
partial function

NOT: Rekabetinde elementi hali gösterileli. Daha detaylı: Slide 33'te

P Set (powerset) hali mi?

- Schema

- Operations

Input (?)

Output (I)

State before (plain)

State After (nested)

\* 'scoring' : after purchase

\* Önekteli sold bir function.

14/04/2020

elemanların arsında neşet etmek istenir.  
Karakterler arsında neşet etmek istenir.

### Operation

Input variable

Output ..

İşlem öncesi function or operation  
İşlem sonrası .. .. ..

} Declaration

Variablen arsında liste .. } Predicate

Return (Bild inde) - Sild J9'da kisa hali VAR.

-- Return --

BoxOffice  
BoxOffice'

s? : Seat

c? : Customer

Seating' = Seating

(s? → c!) ∈ sold

sold = sold' ∨ { (s? → c?) }  
veya sold \ sold' = { (s? → c?) }

21.Oc/2020

E

query operation  $\rightarrow$  lookup, select, insert, updatemodifier ..  $\rightarrow$  purchase, select, delete**NOT:** Initialization'da, database'ın after kılavuzunda **create**.**NOT:** 2'de yaptığıme programınas doğrul, specification.**O:** Fihrist = [In Person  $\rightarrow$  Phone]

Insert

 $\Delta$  Fihrist

p?; Person

k?; Phone

$$m' = m \cup \{ p? \rightarrow k? \}$$

p?  $\notin$  dom(m)

Update

 $\Delta$  Fihrist

p?; Person

k?; Phone

$$m' = m \oplus (p? \rightarrow k?)$$

**NOT:** Darcı subtraction, setten set çıkartılsın,

$$Sld' = S_{s?} \triangleleft Sld \quad (\text{True}) \checkmark$$

$$Sld' = s? \triangleleft Sld \quad (\text{False}) \times$$

21/01/2020

001.01.21

Delete \_\_\_\_\_

A fibrist

Number? : P Phone (Power set phone number. Birbir çok phone var.)

m' = Number?  $\leftarrow m$

NOT: Number?, birden çok phone içerdiginden bu bir türne. Ondan dolayı S' y' göre yok.

Types:

Set

Pair

function

relation

Yeni type: Sequence

Sequence

21/04/2020

[Book]

Stack of Book = [data; seq Book]

Pop

Δ Stack of Book

att! : Book

data = <att!> ^ data'

att! = head(data)

Type Books

Admin

1- State formula

2- Initialization

3- Operation (Before - After)

NOT: Program yazılmaz, program çalıştırınca doğa kalıytırı  
book buluyaz.

21/01/2020

Phone - Person kisminin şifreleme 1 sefer var ~~var~~

29/01/2020

## PROJECT MANAGEMENT

6- Create WBS (Work Breakdown Structure)

- Coding

- Testing

Central document, to do list (What to do)

Some documents: DR; contract document, contains related documents

WBS'deki en büyük parçalar ve spor, iş tipi ve adımları  
hedef, Chart'taki yaratılışının adı, tarih, eyübünlükler,  
(leaf)

DR, Release Implementation

WBS'deki herhangi bir nöde, activity olarak adlandırılır. Hangi  
seriye olduğunun önem yok.Work package, Chart'in en alttakı nödelevels (leaf) yaratılışının adı.

★ vague = belirsiz

LOC: Lines of Code

A Length metric, direct metric. Geliştirilen branch per year.  
Tüm, sonuc yıl. Geliştirilen birileri aynı size belirterek★ Functionality metric

- Indirect

- Kullanıcı (User) Belirçayıncı size notlarıgi

- Bir kullanıcı sonucu projenin sonucu doğurmuyor

\* Fraction card, preparing by yourself  
 Frequency, project location search your capabilities  
 Size =  $\frac{1}{n}$  of fractions (factors)

- 1 Input factors analysis: Add Case { Choose strategy of internal system  
 Input Case }  
Output functionality analysis: Print and Repeat  
 Setting a Case Schedule  
 Adds for some results from user

ab initio: System yes some queries (Ex, ATM del quey batre)  
 Historian extract from history analysis  
 Inputs { User perspective  
 No query }

Internal logical file: Data may fluctuate with time.

External interfaces: Sistemik with trapping outgoing files  
 (history) (Ex)

## 2- Establish Complexity factors for each and apply

\* Attribute function (add customer, customer file...) Simple, average  
 very complex objects twice.

Or; add customer function  
Complexity Levels (Abrahm) Table

- External input
- Procedure 1 step vs. (Complexity score = 1)
- Add, Search, count, TC, others, Td No descriptive objects  
 yes number elements (6 data elements  $\rightarrow$  Complexity score = 2)

/  /  

$$1+2=3 \quad (2-3: \text{simple})$$

Thus, add customer, simple external input functionality

★ Dosyaformatı sütun sayısı + datalı elemler  
 " " satır tipi sayısı + record types

Internal File ve External Interface

1. Cr, Ad, Sipol, TC, address closed by sütün olun.

1. satırda Ad, Sipol

2. .. Ad, Sipol, TC

3. .. Ad, Sipol, TC, Address

4. .. Ad, Sipol

} # record types = 3

Ad, Sipol sadece diğer 2 (TC, address) optional de

∴ # record types = 4

TC address	
Ø	Ø
Ø	✓
✓	Ø
✓	✓

★ External Inquiry, input/output funktionları en yüksek kompleksiteye sahip olur.

NOT: Complexity (simple, average, complex) bulutları sınırlı

sayı (sayı) ile sağlanmamıştır.

— / —



★ Single, Average, Complex take one project  
size behaviour

Or, odd new customers = 3

★ Turn function in size behaviour  
Or,  $f_1 = 30$  (Unadjusted Function Point)

★ Size-Weight

Size of the Project = <sup>Unadjusted</sup> 30 Function Point

3- Compute and "Influence Multiplier" and apply

4- Results in "function point total"

Size of the Project = 31.35 Function Point

**NOT:** Function Point dörtlü adet genel dimensionde LOC'a  
çarpanı gerekir. Bu da tıpkı projenin katsayı Function Point  
hesap edilip projenin ildən eftə branch LOC's təyin olunur.

1.27 (Video)

Size Estimation Methods (SfB)

— / — / —

## PERT shows:

- ★ Minimum Completion Time (Longest path - dashed line)
  - ★ Longest Path in the job
  - ★ For each task, what should be done before
  - ★ Critical path
  - ★ Dependencies between tasks
- NOT: Critical path'de hizmet biri gecitirse onunla birlikte tüm işler gecitir.

NOT: Sürecede Tasklar arc'ın üzerinde olmalı.

NOT: PERT, sadece 1 basılıgın懦ksin ve sadece 1 bitiş noktası var.

## Gantt Chart

- ★ İl başta critical path'de hizmet taskları yerleştirilir. Daha sonra diğerleri.
  - ★ PERT'inin gecitini içermeyen Gantt Chart'sı var.
  - ★ PERT'te O(yarım) işe özyapılı etki genelde sadece yarısılır. Fakat, nüfuslu yeri.
- NOT: Basıç effect O(düzen). (Benefit of Base Calculation).

10/05/2020

## PROJECT MANAGEMENT 2

### Estimating Schedule

Generic Schedule

Effort

Labour

Cost

Yukarıda verilen sayılar her biri:

### COCOMO Levels

1 Basic

2 Intermediate }

3 Detailed

### Basic COCOMO

Grafikteki her nöde bir projeyi temsil eder.

X ekran size

Y .. effort

$$E = a \cdot size^b$$

where b constant

### Line fitting Algorithm

\* Her formülü hatırlamamız istenir \*

Size, KLOC üzerinden elde edilen 2000 Line of Code

varsa projek Size = 2 olur.

Size funkcı Point dose direk KLOC crıckt olmaz form KLOC'a  
çevrilirken Son hikma degr Size'a esit olur.

— / —

## Problems with COCOMO

Basis Model Modeli uygulamamıza yarar. Incremental Model de kullanabilir.

COCOMO'nun büyük problemi deneysel program geliştirme fikrine dayalı  
bulunduruculuk. COCOMO bir basis teknikine sırf teknik konular  
diumurda gerekli basitce adıjustment yaparak COCOMO'yu kullanır.

## Schedule Estimating

Schedule = Time to complete

Schedule =  $2.5 * \text{Effort}$

Time Divisionde Requirements ve Design'inin yarısı : Code'dan  
genellikle bu da en Code'da çalışan inşaat işlerinin paralel olmasına  
bağlıdır.

## Effort Schedule

ADD-ONS: COCOMO'nun göz ardı edilen taskları. (Quality Assurance,  
Configuration Management, Management issue etc.) dahil edilir.

Head count: Çalışması gereken kişi sayısı. Or; 2.5 ile 2 full  
time, 1 part time çalışan personel o zaman çalışır.

## Labor Schedule

Ay boyunca herhangi birini çalıştırırmamın təqdim etməsi hərəkət  
uygunluğunu sağlayır.

— / —

## Cost Categories

SS: Software Support, diğer core integrable (SD, ST)  
parallel yarışmalarından selected'a etkisi yok.

12/05/2020

Size'a belli懦ue scire hedefleniyse  
1 Function Point, 1 COCOMO (ne kadar effort, how much like),  
1 PERT Chart, Gantt Chart (Task'ın ve ömrünün) Critical Path  
Verilenin Shifting along, Minimum Time (cycle)  
Birkaç intermediate cocaine point,  
Belli bir güzergah eklenip cocaine'ın etkisi (Requirement). Firma nasıl çok  
Activity're package ne?  
1 son 2 aylardır.

Sınav Tarihi: ~~9 Haziran~~ 16 Haziran

## DESIGN

Cohesion & Coupling sınırları giderilir.  
Separating concern ve architecture very good funktionallaşırma  
evalut etre.

Final sınav 2'leren itibaren alınamaz.

2, cocaine, function Point

Design principles

Information Hiding

het 1 belki diğer hetki entem

Architecture ne işe yarar?

Modularity ne işe yarar? Ne işe yarar?

Major areas of Design: Risk

Architecture

Interface  
Components

— / — / —

Design Representation akış şeması.

COCOMO'ın işleyiş bütçesi yolu.

19/05/2020

2 (ödev sorunu benzer), PERT-Gantt Chart, Function Point Hesaplama (ödev sorunu), Function Point up (meg Code up) COCOMO hesaplama (COCOMO epki ve school ile benzer), functional design (DESIGN) principle, coupling, separation of concern nedir? Information Hiding nedir? yearlər? Architecture'da bitişiklik? Architecture stilleri nedir? Bir programın hangi dərinlikdən yuxarıya doğru yüksək olmaye çalışır? Vertical architecture, layering principle? Architecture'da separation of concern nedir? Voice correlation nedir? Architecture'da cohesion, coupling nedir?

## ARCHITECTURE PATTERNS

19/05/2020

\* Architecture refector 2nd, Design review 2nd,  
Anteberkək Architecture'in Fazları

1- Erstelleni John Eddy

2- frontere (Ör, Sprint) uygulanması uygun

Tiplər

1- Model/View/Controller (MVC)

\* Web based tətbiqetmə kılavuzu.

\* Sistemini 3 parçaya bölmə.

1- Model

2- View

3- Controller

— / —



★ PC üzerinde Graphical User Interface (GUI) kullanılarak bilgiler.

★ Data ve User Interface'i birbirinden ayıracak.

User Interface → View component (Aracı, satırı view istenilen yerde)

View component içinde hedef database query (sql table) olacak.

View component içinde hedef hizmet logic olacak.

View'in istek button'unu click ettiğinde gerekli yankılarla gerekli sayfalar (werkstrom)

View oluşturur, ama detaylı manipüle etmez. Sadece göstergeleri.

View, Web App'eki HTML'lerini.

View, konar componenti.

★ Model (hizmet; detayı tutmak, detayı manipüle etmek).

★ Model'de Student Class', Add Course functionality var.

Model Classları arası View metodlarını çağırırıf.

View    "    Model    "

★ Design for separation of concerns var.

Istevi: Aynı detay farklı view'larla gösterilebilir.

★ Controller'ın uses konusundan.

★ Reusability yüksek.

★ Maintainability var.

— / — / —



★ Cohesion var.

★ Information Hiding var.

★ Devezatlı; Sistem interactīn dēindekī koden proje bittīle kod kompleksit̄i yükselīdir. Uygulamalar dār.

★ Highly user interaction oldugu dramātik hale getirilecektir.

## 2- Layered Architecture

★ User interaction varsa kullanabilir.

★ Security īzin kullanabilir.

★ Functionalityleri alp kategorilere bölüp layerler oluştururuz.

Her layer'in bir işi olur.

★ Separation of concerns var.

★ Layerler high level

★ Layerler arası iletişim ar. Sadece üst layer alt layerden iş istenir. Alt layer ile temas bitirin.

★ Capting ar. Sadece üst layer Alt layer ile cap.

★ Üst layerde yeri ne sağıyor?

Incremental development yapabilmek

Fazla gerekten fazla layeri vermemek.

— / —



Portability sağlayır.

Changeable sağlayır.

**NOT:** MVC, model and view.

Multiple views

Layred, portability, changeability

Herkes bir konuda bağlıdır.

} Keywords

### 3- repository Architecture

★ 1 componentin içinde veya componentlerin ortak hali olduğu  
bir tek bir ucu varsa o da repository'lludur.

★ Repository = Depo

★ Example; IDE (Interactive Development Environments)

★ Data, ortaya koyma, işlevsellikler aramaları etrafında.

★ Separation of concerns var.

★ Kategorize var.

★ Farkların etrafındaki gizlilik, birbirlerine konumlaşmış, farklı yerlerden indirekt  
ileşimeinde.

★ Coupling az. Sadece kategoriye göre.

★ Efficient sharing var.

— / — / —

★ Negativity: Single point of failure (db) → Project Repository

Databasi hata hentesi ettiligini.

★ Avantaj: Huge data with efficient sharing

## 4- Client-Server Architecture

★ Distributed

★ Her bir istekten farklı serverlar servisleri.

Birimleri birbirinden separan olarak desktop uygulamalar  
logik oluşturuyor.

★ Server ≈ Model in MVC

★ Separation of concern var.

★ Redundant server (1 serveren yoksugılık) tercih edilebilir. [Replication]

★ Single computera kurulabilir.

★ Transparent: Sistemin dışarıdan görünümü değişildiğinde kullanıldığında

## 5 - Pipe and filter Architecture

Pipe: Bir ıslemin sonucu kışla bir ıslen uygulandı.

- ★ Bir türdeki outputu başka bir türdeki inputu değiştir. Etbe eklenen yeri düşünen (Sabit konası)
  - ★ Uyg. Interface arayız uyg. lendere kılavuzları.  
Cf; Batch Processing ve Transaction işleme teknikleri  
Billing sistemleri  
Yıl sonu istemciyi (Banka)
  - ★ İstek, geri dönüşler içeri taşıyanız.
  - ★ Separation of concern var
  - ★ Coupling var.
  - ★ Workflow var.
  - ★ Information hiding var.
  - ★ Input ve Output var.

## Architecture Patterns tipkileri bittir.

Transaction Based Architecture → Pipe and filter Architecture

## Information Systems

1

$\rightarrow$  Layered

11

## Language Processing

1

→ Pipe and filter

“Repository  
(Symbol Table)

02.06.2020

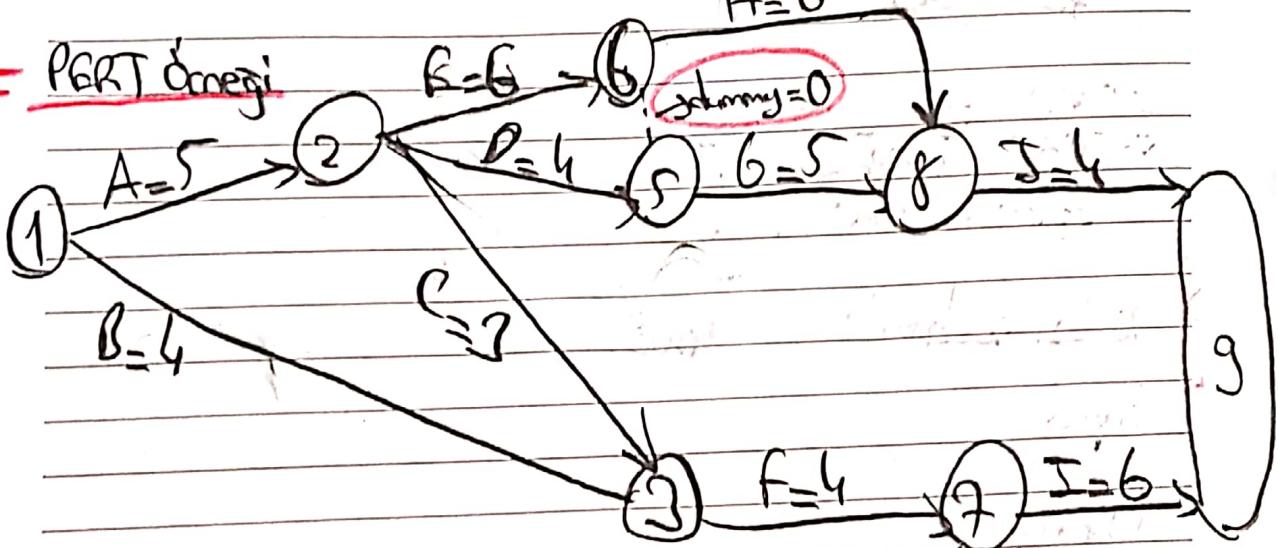
SORU & CEVAP

★ 2 ayrı quiz, 90 dk ve 90 dk

★ PERT Chart göre var Tasklar aralarında obak. Karşın

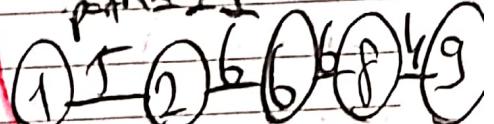
★ Cocuk ve friend Point serbest. Hoca quiz formatta real sebeğin devamı

$$H = 6$$

1- PERT ÖrneğiActivity Predecessor Relation

A	P	F
B	-	4
C	A	3
D	A	4
E	A	6
F	B, C	4
G	D	5
H	P, E	6
I	F	6
J	H, G	4

Tot Duration of critical path = 21



(dummy işlendiği için)

★ Hoca dummy transition, emreydi!

**NOT:** Activity (Task) on Arc YAPILACAK.

★ İşte! Ondan sonra tıpkı ve dekompleksi gidernez  
Birim planlaması giderir.

Sayı Şimdi VDC haly step nötrlik effect vermek.

Sayı Kıyasla? İhtiyaç? evin halleridigi formül kullanılsın.

Sayı İşte! ne işe yarar

Sayı dekompleksi bir hali

Sayı İhtiyaçla kesişenin bir hali.

Şimdi de onunla hala aynı gibi

1- Modelleme

Personas için: (auding and design)  
(Ab) (CDK)

## 2- Information Hierarchy

Modelleme: İhtiyaç (istediğimiz...) else dekompleksi hali,  
Amas seviye dəqiqi.

Təcəmmu həyəti

1. Föde

2. Arxitektura

3. Sənaye (Nəsim-Bələd Nüfuzları)

4. Corpant

Sayı Nüfuzları bilməməni giderir.

2-2 Orij (3. ödev screen)

GraphPurchase operation:

koltuk? : P Seat (set of seat): Birbirin arası sıradır. P Seat var.)  
müşteri? : Customer  
A BoxOffice

seating' = seating  
sold' = sold  $\forall x \rightarrow y | y = \text{misted}, x \times \in \text{koltuk?}$

success

success [rep!, message, error : rep! = ok]

Message = {ok, error1, error2, error3}

error

koltuk? : P Seat

BoxOffice

rep! : Message

rep! = error1

$\exists x \in \text{koltuk?} | x \in \text{seating}$

error2

BoxOffice

bütün? : PSeat

rep! : Message

rep! = error2

$\text{bütün?} \subseteq \text{dom(sold)}$  [Bütün biletler bir bireh randevu attıktır]

error3

BoxOffice

rep! : Message

bütün? : PSeat

rep! = error3

$\exists x \in \text{bütün?} \mid x \in \text{dom(sold)}$  veya

$\text{bütün?} \cap \text{dom(sold)} \neq \emptyset$

$\boxed{\text{GraphPurchaseT} = (\text{GraphPurchase} \setminus \text{success}) \vee \text{error1} \vee \text{error2} \vee \text{error3}}$