About Final Exam.

Time: 10:30—12:30am, June 27 (**Thursday**)

Place: 玉泉教7-208

Form: Closed-book Exam.

---I. Multiple Choices : 20pts (10 questions)

---II.True/False Statements : 10pts (10 questions)

---III. Brief answer:20pts (3 questions)

---IV. Big question:50pts(5 questions)

Review the course

------Software Engineering

A Practitioner’s Approach

王章野

May 27, 2024

Ch.1 The Nature of Software

- **Software is a set of items or objects that form a configuration that includes: instructions (computer programs) ; data structures ; documents.**

--- Software doesn’t wear out, but it does deteriorate!

- **Software Application Types (7 categories, p.6-7)**

- **Legacy Software (Why need to evolve?)** and **Changing (p.7-11)**

- Software Myths and their misleading attitudes **(See PPT)**

---e.g. If we get behind schedule, we can add more programmers and catch up.

Ch.2 Software Engineering

- **SE– A Layered Tech**.---"Quality" Focus/ **Process Model** / Methods/ Tools
- **5 Process Framework--- Communication/Planning/Modeling/Construction/Deployment**
- **7 General Principles (KISS** — **K**eep **I**t **S**imple, **S**tupid!**)**

Ch.3 Software Process Structure

- **4 Process flow: Linear / Iterative/ Evolutionary / Parallel**
- **Process Patterns**--- Initial & Resulting context/Solution/ Related patterns/Known examples (p.35-36)
- **Process Assessment** --- ISO 9001:2000 for Software
- **CMMI**: Capability Maturity Model Integration ---**6 Levels**

Ch.4 Process Models

- Prescriptive Models, Waterfall Model, Incremental Process Models: RAD Model, Evolutionary Process Models: Prototyping; Concurrent(协同) Development Model.

- Specialized Process Models / The Unified Process

- Personal Process Models VS Team Process Models

1) Planning	1) Each project is "launched" using a "script" that defines the tasks to be accomplished
2）High-level design	2）Teams are self-directed
3）High-level design review	3）Measurement is encouraged
4）Development	4）Measures are analyzed with the intent of improving the team process
5）Postmortem (后验)	

Ch.5 Agile Development

- **Agility**:

-- Effective response to **change/communication;**

--Driven by customer’s requirement;

--Self-**organization/control**;

--Rapid, incremental delivery of software

- 12 **Agile Principles**

1) Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2) Welcome changing requirements, even late in development. Agile processes harness (利用) change for the customer's competitive advantage.

3) Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

……

- **XP**--Extreme Programming: **Pair programming; Unit Tests before Coding**

Ch.5 Agile Development(2)

- **Agile Process Models**:

-- **ASD** (Adaptive Software Development);

--DSDM (Dynamic Systems Development Method);

--Scrum: 15 min daily meeting (3 questions)

--Crystal;

--FDD (Feature Driven Development)

- **AM-**-Agile Modeling

Ch.6 Human Aspects of Software Engineering

- Traits of Successful Software Engineers; The Psychology of SE;
- Effective Software Team Attributes; **The Software Team / Agile Teams;**
- SE **using the Cloud; Collaboration Tools; Global Teams: Communication,Collaboration,Coordination; Principles that Guide Practice;**

Ch.7 Principles that Guide Practice

- Software engineering principles are likely to serve a professional programmer throughout his /her career (p.104-128).

---Ex. Agile Modeling Principles 2#: Travel light – don’t create more models than you need.

Ch.8 Understanding Requirements

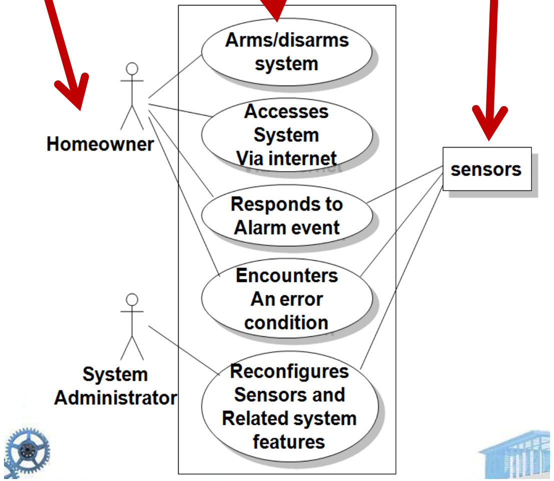
- **8 Requirements Engineering Tasks: Inception**（stakeholders） **/ Elicitation(引出，Normal /Expected /Exciting** requirements**,** Non-Functional Requirments, Use cases**)**/ **Elaboration**（ building analysis model ）**/ Negotiation / Monitoring**/ **Specification/ Validation(**Consistency / Omissions / Ambiguity**)/ Requirements management** （changes）

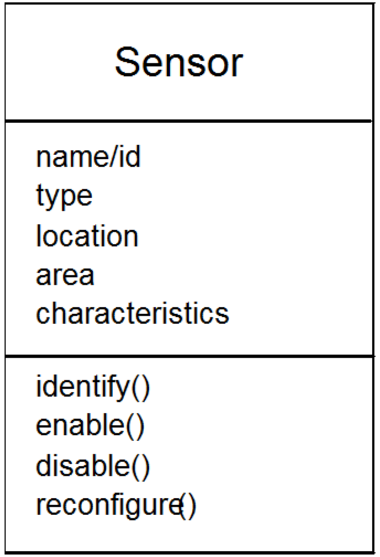
Use-Case Diagram/ Class Diagram/ Activity Diagram/ State Diagram !!

Three part of Use-Case Diagram:

Role, Functons, third partied things

Use-Case Diagram


Homeowner
System Administrator
Arms/disarms system
Accesses System Via internet
Responds to Alarm event
Encounters An error condition
Reconfigures Sensors and Related system features
sensors

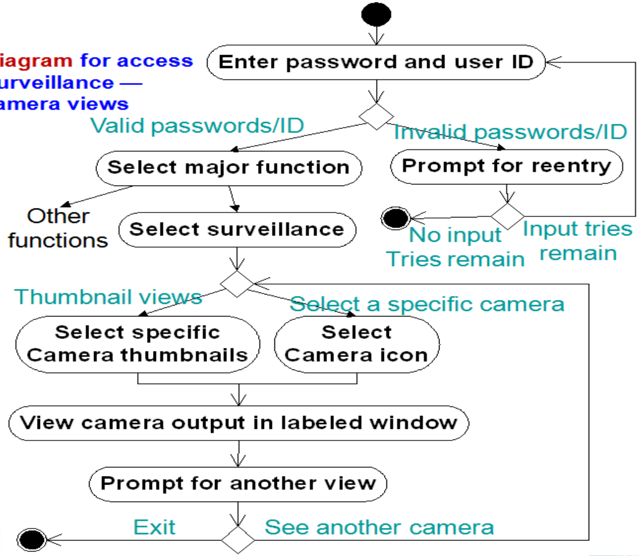


Sensor

name/id
type
location
area
characteristics

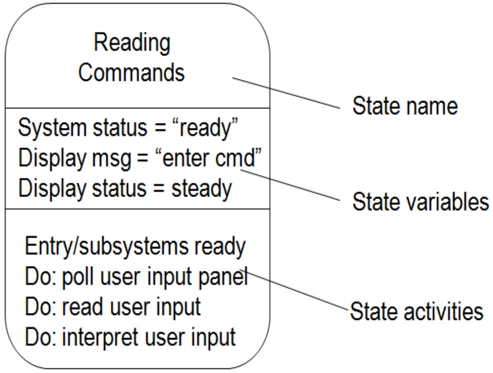
identify()
enable()
disable()
reconfigure()

Activity diagram for access camera surveillance — display camera views function



State Diagram(状态图)

State Diagram



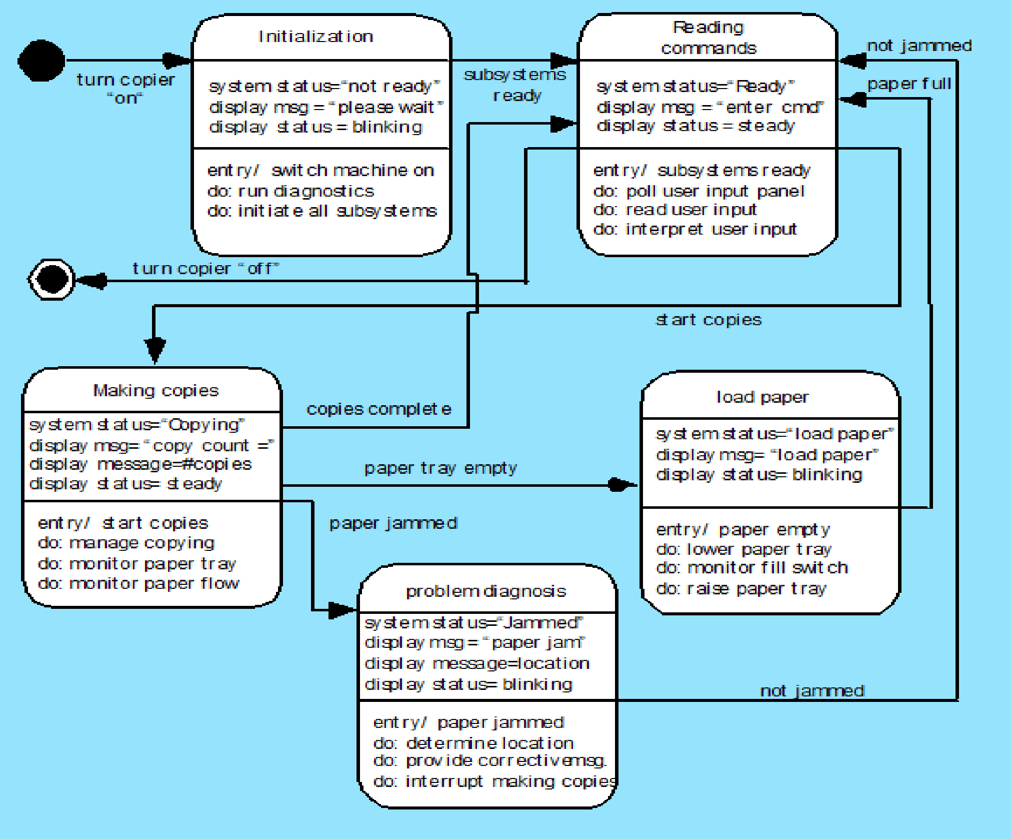


Figure 7.6 Preliminary UML state diagram for a photocopier

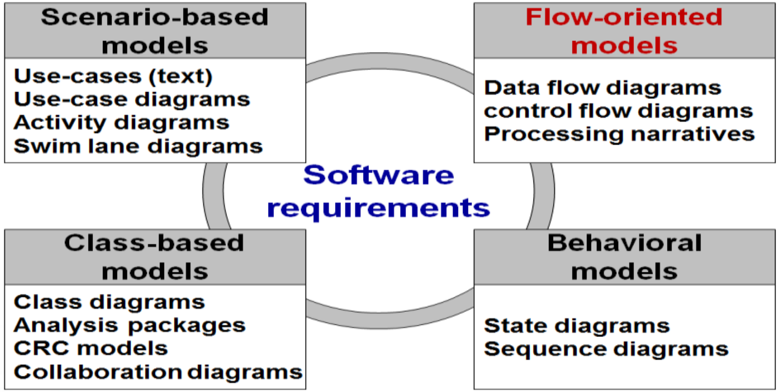
Ch.9 Requirements Modeling：Scenario-Based Methods

- **4 Requirements Models------**
- Scenario-Based Modeling

---Use-Cases: ***actors & users***

--- Developing & Reviewing a Use-Case

---Activity and Swim Lane Diagrams



Scenario-based models

Use-cases (text)
Use-case diagrams
Activity diagrams
Swim lane diagrams

Flow-oriented models

Data flow diagrams
control flow diagrams
Processing narratives

Software requirements

Class-based models

Class diagrams
Analysis packages
CRC models
Collaboration diagrams

Behavioral models

State diagrams
Sequence diagrams

Ch.10 Requirements Modeling: Class-Based Methods

- O-O(Object-Oriented) analysis:

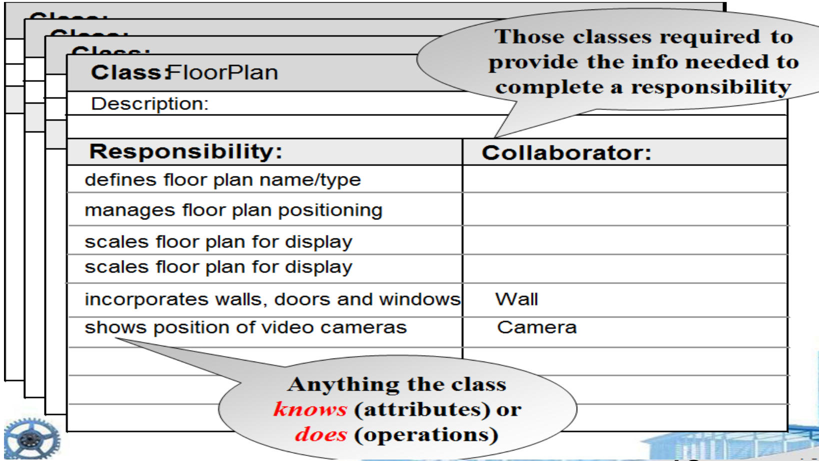
  ---Classes and objects/Attributes and operations

  ---Encapsulation

  ---Class Hierarchy

- **Class Diagram**;
- **CRC Modeling!!!**

  ----Class, Responsibilities, Collaborators



Class:

Class:

Class:

Class: FloorPlan

Description:

Responsibility:	Collaborator:
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

Those classes required to provide the info needed to complete a responsibility

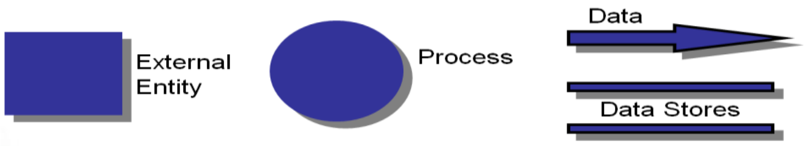
Anything the class *knows* (attributes) or *does* (operations)

Ch.11 Requirements Modeling: Behavior, Patterns, and Web/Mobile Apps

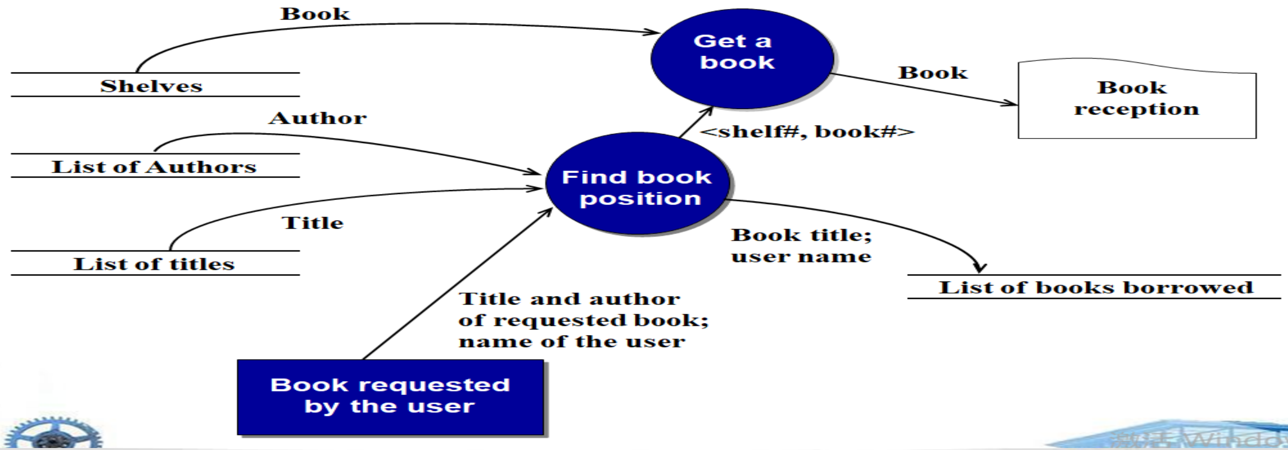
- Behavioral Modeling--- the states of each class / System
- **State Diagram/ Sequence Diagram**
- **DFD (Data Flow Diagram)**
- Specification Guidelines
- Requirements Modeling for WebApps

---5 **Analysis (Content/ Interaction/ Functional/ Configuration/ Navigation)**

- **Data Flow Diagram**

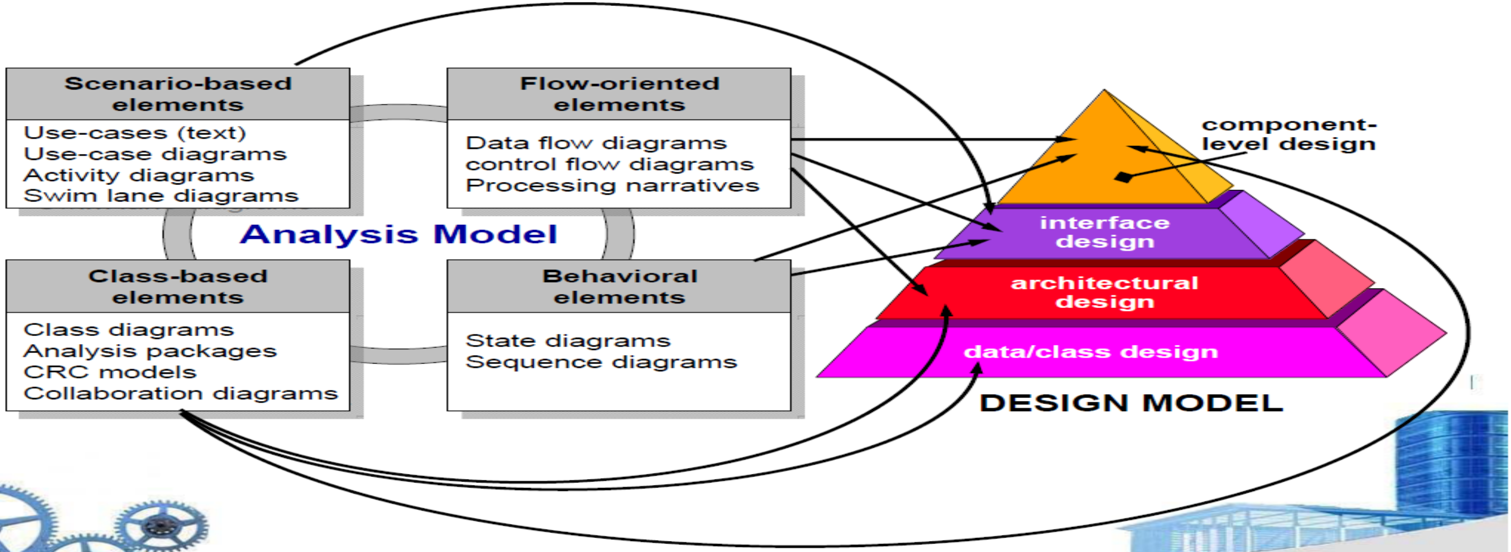


Book request = Find book position + Get a book



Ch.12 Design Concepts

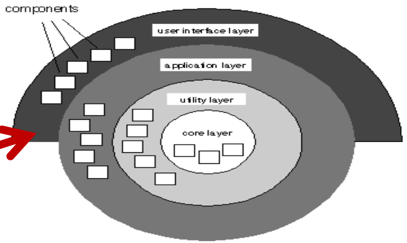
- Good software design: ***Firmness, Commodity, Delight***
- **4** Designs (Data/Class, Architectural, Interface, Component-level)
- **10 Design Principles**
- **Modularity: Trade-offs**
- **Information Hiding/ *High Cohesion, Low Coupling***
- **5 Design Model Elements---***Data*, Architectural, *Interface*, Component, *Deployment*

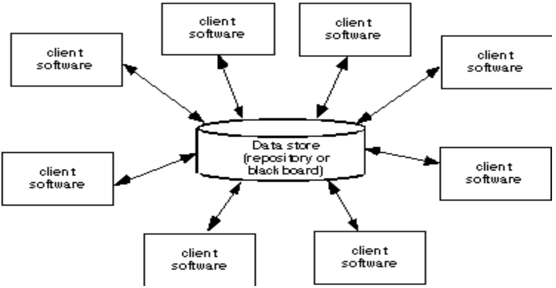


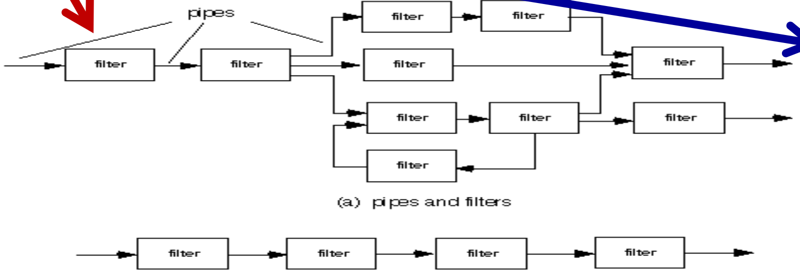
Ch.13 Architectural Design

- **4 Architectural Genres**

--- Data-Centered, Data Flow , Call and Return , Layered!!

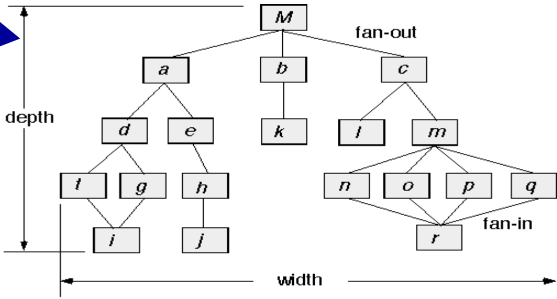






(a) pipes and filters

(b) batch sequential



- **Architectural Patterns:** *Concurrency, Persistence, Distribution*
- **Mapping Data Flow: 1)Transform Flow; 2) Transaction Flow**
- **Partitioning Program Architecture: Horizontal , Vertical**
- *Architectural Description Language (ADL):* UML

Ch.14 Component-Level Design

- 7 Basic design principles (Ex. Open-Closed Principle)
- Component Level Design (Cohesion & Coupling)
- Component Design for WebApps (content & functional design)
- Component-Based Development: reuse (OMG/CORBA, Microsoft COM, Sun JavaBeans))
- The CBSE Process（ Component Based Software Engineering ）

Ch.15 User Interface Design

- Three Golden Rules： 1）Place the user in control； 2）Reduce the user’s memory load； 3）Make the interface consistent
- 4 Interface Analysis and Design Models: User model , Design model, Mental (or system perception) model, Implementation model
- 14 Interface Design Principles: p.338-339
- Web / Mobile Apps Interface Design Workflow : p.341-342

Ch.16 Pattern-Based Design→Reuse

- Three-part rule: context, problem, solution
- Frameworks: An implementation-specific skeletal infrastructure, vs Architectural
- Pattern-Based Design in Context / Common Design Mistakes
- Architectural/ Component-Level/ UI/ WebApp/ Mobile Apps Patterns

Ch.17 WebApp Design

- **Two basic approaches：** artistic ideal & engineering ideal
- WebApp Design Quality：*Security/Availability*（24/7/36）*/Scalability/Time to Market*
- **6 Design Goals:** *Simplicity/Consistency/Robustness/Navigability/Visual appeal/Compatibility*
- **WebApp Design Pyramid: Interface / Aesthetic/ Content / Navigation(NSU) / Architecture (MVC)/ Component Design**

Ch.18 MobileApp Design

- **Development Process Model**: **Formulation /Planning/ Analysis/ Engineering/ Implementation and testing/ User evaluation**
- **MobileApp Design Mistakes: Kitchen sink, Overdesigning, Non-standard interaction , etc.**
- **MobileApp Design: Best Practices**

Ch.19 Quality Concepts

- **Software Quality: Durability/ Serviceability/ Aesthetics/ Perception**
- Software Quality Dilemma------ “**Good Enough**” Software
- 3 Cost of Quality： *Prevention costs* ***/*** *Internal failure / External failure* ***costs***
- **3 Impact of Management Decisions**： Estimation / Scheduling / Risk-oriented *decisions*

Ch.21 Software Quality Assurance （SQA）

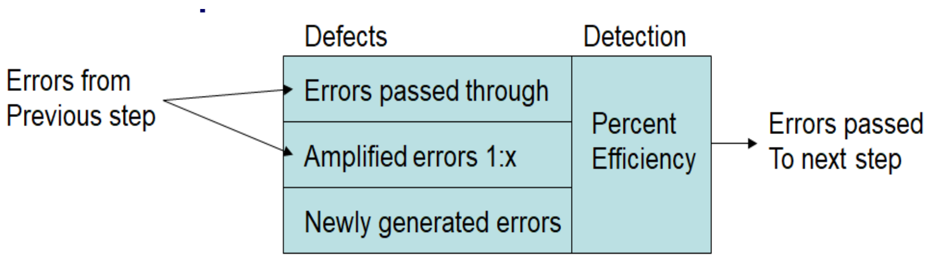
- **Elements of SQA**;
- SQA Goals: **Requirements quality / Design quality / Code quality/ Quality control effectiveness**
- **Six-Sigma** for Software Engineering;
- Software **Reliability** and **Availability**

Ch.20 Review Techniques

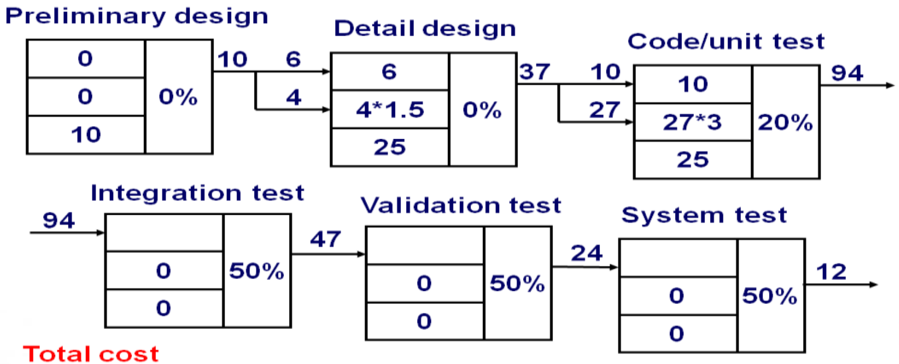
- **Errors vs defects**
- **Defect Amplification Model**
- **Informal Reviews:** ***pair programming***
- **Formal Technical Reviews（FTR）：**

------walkthroughs **and** ***inspections***

- **10 Review Guidelines**

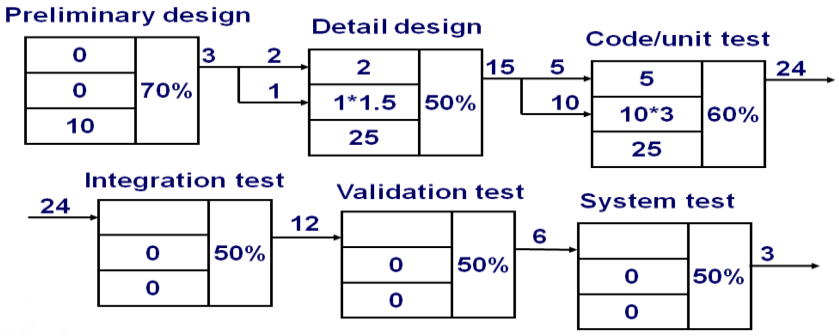


- Example: Defect Amplification No Reviews



Total cost
= (10+27*3+25)*20%*6.5 + (94+47+24)*50%*15 + 12*67 = 2177

- Example: Defect Amplification With Reviews



Total cost = (10*70%+28.5*50%)*1.0 + (5+10*3+25)*60%*6.5
+ (24+12+6)*50%*15 + 3*67 = 771

Ch.22 Software Testing Strategies

- **General testing strategy for software product:**

--- Conceptual Testing、Unit Testing、Integration Testing 、 Regression Testing 、 Validation testing、System Testing、 User Experience Testing、Stability Testing、 Connectivity Testing、Performance Testing、Compatibility Testing、 Navigation Testing、 Security Testing、Certification Testing

- **Verification VS Validation;**

- **Independent Test Group VS Developer Group**

- Unit Testing: **driver→**module**→stub**; Class Testing **for O-O software**

- Integration Testing: **Top-down, Bottom-up,** Regression testing, Smoke testing; thread-based testing, use-based testing, cluster testing **for O-O software**

- **WebApp & Mobile Testing:** User Experience /Stability /Connectivity /Performance /Compatibility /Navigation /Security /Certification **Testing**

Ch.22 Software Testing Strategies (2)

- **High Order Testing:** ***Validation/*** *System/ Alpha/Beta/* ***Recovery****/ Security/* ***Stress****/ Performance* ***testing***
- **4 Debugging Techniques:** ***brute force / backtracking /induction/deduction testing***

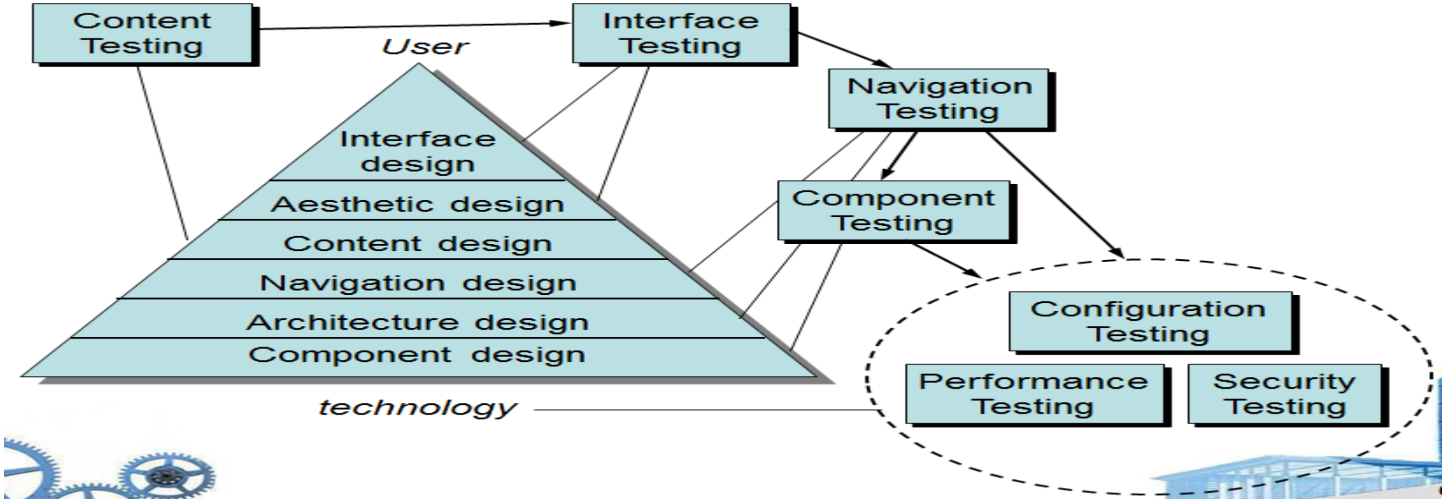
Ch.23 Testing Conventional Applications

- **White-Box Testing VS Black-Box Testing**
- **White-Box: Cyclomatic Complexity V(G)=P+1 ;Drawing the independent paths! Loop Testing;**
- **Black-Box:** Equivalence Partitioning; Boundary Value Analysis

Ch.24 Testing Object-Oriented Applications

- **Classes**→attributes, operations, messages
- **Class Model Consistency→CRC Model**
- **Testing Methods:** *Fault-based/Class &Class Hierarchy/ Scenario-Based /* Random / Partition ***Test ing***

Ch.25 Testing Web Applications



Ch.26 Testing Mobile Applications

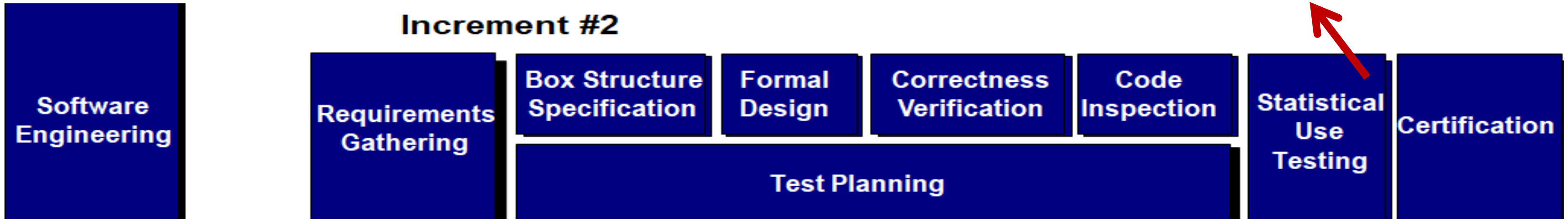
- **9 Mobile App Testing**: Conceptual /Unit and System /User Experience/Stability /Connectivity /Performance /Compatibility /Security /Certification **Testing**
- Testing tools; Cloud Testing

Ch.27 Security Engineering

- Analyzing Security Requirements: *Exposure, Threat analysis, Controls*
- Online Security Threats: *Social Media / Mobile Applications / Cloud Computing / Internet of Things*
- Security Engineering Analysis; Security Assurance; Security Risk Analysis

Ch.28 Formal Modeling and Verification

- 2 methods: 1) Cleanroom software engineering; 2) Formal methods.
- Cleanroom Strategy: Increment Planning “Usage probability distribution”



Software Engineering

Increment #2

Requirements Gathering

Box Structure Specification

Formal Design

Correctness Verification

Code Inspection

Test Planning

Statistical Use Testing

Certification

- Formal Specification: 1) Desired properties— *consistency, completeness, and lack of ambiguity;* 2) Formal syntax —interpreted *in only one way;*
  3) *Consistency* is ensured by *mathematically proving*

Ch.29 Software Configuration Management

- ***Software Configuration Items (SCI):*** **programs, data, documents**…
- **Baselines: System Specification/ Software Requirements/Design Specification/ Source Code/Test Plans / Procedures / Data/ OS**

- **SCM Repository:** the set of mechanisms and data structures that allow a software team to manage change in an effective manner.

- **SCM for Web & Mobile Engineering:** ***Content, People, Scalability, Politics***

- **4 Major capabilities** of **Version control System: p.634**

Ch.30 Product Metrics

- **Measures, Metrics and Indicators**
- **Size-Oriented** Metrics(LOC) vs **Function-Based** Metrics (FP)
- **Architectural Design Metrics; Metrics for Source Code: Halstead's Theory**
- **Maintenance Metrics**→SMI = $[M_T - (Fa + Fc + Fd)]/M_T$

Ch.31 Project Management Concepts

- **The 4P's:** *People, Product, Process, Project*
- **5 main kinds of Stakeholders and their roles:**

1)Senior managers **who define the business issues that often have significant influence on the project .** ***2)Project (technical) managers*** **who must plan, motivate, organize, and control the practitioners who do software work.** ***3)Practitioners*** **who deliver the technical skills that are necessary to engineer a product or application.** ***4)Customers*** **who specify the requirements for the software to be engineered and other stakeholders who have a peripheral interest in the outcome.** ***5)End-users*** **who interact with the software once it is released for production use.**

– **4 Organizational Paradigms:** ***closed / random / open / synchronous paradigm***

Ch.32 Process and Project Metrics

- **Process metrics** – *effectiveness of a process* → ***A Strategic View***
- **Project metrics** – *workflow, real-time approach* → ***A Tactical View***
- Statistical SQA (Software Quality Assurance) -error categorization & analysis
- **Defect removal efficiency→DRE = E /(E + D)**

Ch.33 Estimation for Software Projects

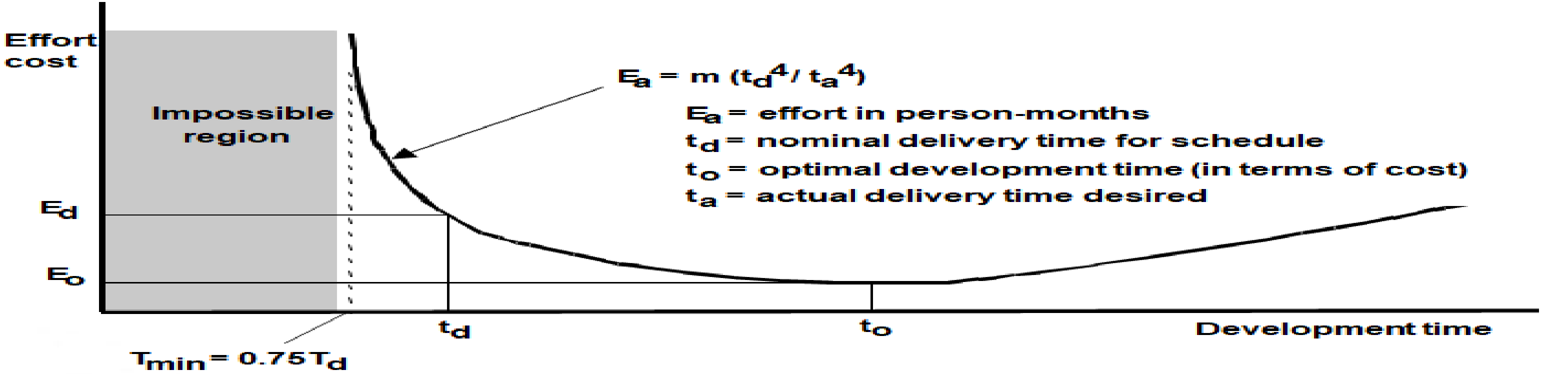
- **What to estimate: How long / How much effort / How many people /Resources (hardware + software) + Risks**
- **Empirical Estimation models:**

---COCOMO II--E = $[LOC \times B^{0.333}/P]^3 \times (1/t^4)$

- **The Make-Buy Decision**

Ch.34 Project Scheduling

- The Putnam-Norden-Rayleigh (PNR) Curve; Timeline Charts



- Effort Allocation/Distribution→ 40-20-40 rule;
- Earned Value Analysis (EVA): i.e. Schedule performance index: SPI = BCWP / BCWS ~ 1.0

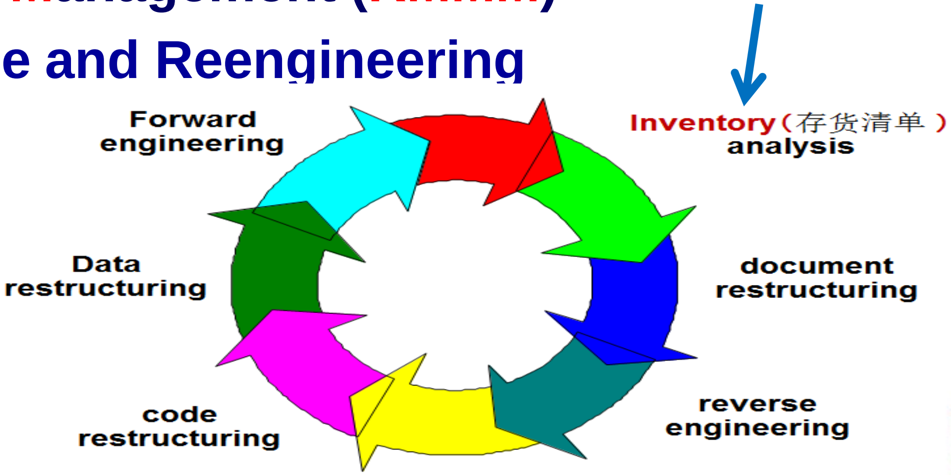
Ch.35 Risk Analysis

- **Reactive vs. Proactive Risk Strategies**
- **Risk Identification: Negligible, Marginal, Critical, Catastrophic**
- **Risk projection(*risk estimation):*** RE = P x C
- **Risk Mitigation, Monitoring, and Management (RMMM)**

Ch.36 Maintenance and Reengineering

- **Software Reengineering**
- Restructuring (**document, code, data**)
- Reverse Engineering
- Forward Engineering
- Economics of Reengineering

Beginning



A Software reengineering process model

Example

I. Please select the correct answers and fill in the answer sheet: (20 pts., 2pt. for each)

1. Which of the items listed below is one of the software engineering layers?

A. Process B. Manufacturing C. Methods D. Tools

Answer: ACD

***II. Please specify "T" (true) or "F" (false) for the following statements and fill in the answer sheet*: (10 pts., 1pt. for each)**

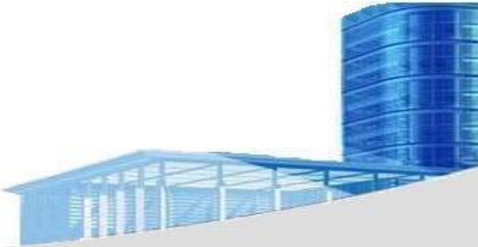
1. Software deteriorates rather than wears out because multiple change requests introduce errors in component interactions.

Answer: T

Example(2)

III. Please give brief answers to the following questions: (20 pts.)

1. What is the RMMM for the risk of software engineer? Take EMSS(疫情检测与防疫系统) project as an example to make the RMMM plan for the risk of software engineer change (8pts.)



Example (2)

Answer:

1) RMMM means the Risk Mitigation, Monitoring, and Management.

2) RMMM plan:

1. **Project: EMSS system**
2. **Risk type: Human resource risk OR Infrared hardware risk**
   Priority (1 low ... 5 critical): 3
3. **Risk factor:** In the process of software development, there are staff changes such as software engineer leaving.
4. **Probability: 40 %**
5. **Impact:** Software development process delay
6. **Monitoring approach:**
   1、Monitor the mood of engineers; 2、Check the productivity of engineers; 3、Investigate the salary levels of the competitor
7. **Mitigation (Contingency plan, 应急方案):**
 1、Group building parties; 2、Rich documents; 3、Frequent technology conference or training;4、Human resource pool.
8. **Management (Estimated resources):**
   1、Find a new person; 2、Work handover；3、Summarize the cause of the problem

Example (3)

2. What are the attributes of a good software test?**(6pts.)**

Answer:

- 1) Has a high probability of finding an error;
- 2) Not redundant;
- 3) Should be capable of uncovering a whole class of errors;
- 4) Should not be too simple or too complete.

Example (4)

IV. Garbage Collecting Service Platform for Family (GCSP) (50 pts.)

Software scope: A company wants to develop a Garbage Collecting Service Platform for family (GCSP) to facilitate the garbage collection and help us to build a beautiful world.

After inputting the name, sex, age, address and ID, customer can register a new account online. Logging in the system, he/she can book the garbage collection service by announcing the pickup time and the volume. Furthermore, customer can buy some goods using his/her scores, which were achieved by his garbage contributions. Customer can select his favorites and make a order using his/her scores to finish the payment.

The platform will evaluated the feasibility of service requests and orders according to the availability of resource, and suspend the unavailable requests and orders. Then the platform will analyze all of the available collection service requests and goods orders, and make an optimized execution plan, assign the related attendants to pick up the garbage and deliver the goods on time. The attendants also need to catalog the garbage to mine the value, calculate the scores for customer to close the requests, and select the destination of the garbage, such as selling some paper or metals directly to recycle stations or transporting the non-recyclable garbage to power plants. The administrator of platform will maintain the system, such as goods repository, customers list, attendants list and security policies.

-

IV. Question lists

1. Please draw the data flow diagram for processing the garbage. (12 pts.)

2. Please give the two CRC cards for classes “customer” and “attendant”. (10 pts.)

3. Please give the state diagram for the “order” class. (8 pts.)

4. Please draw the layered software architecture of GCSP. (10 pts.)

5. Please describe the testing strategy for GCSP platform. (10 pts.)

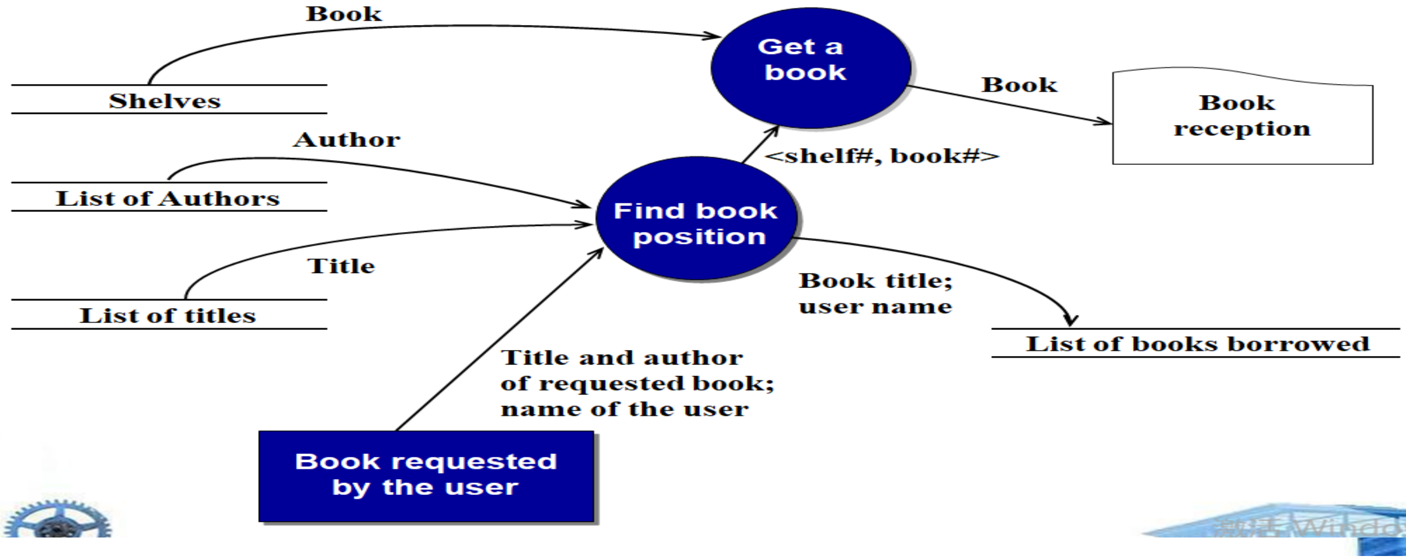
Answer(1)

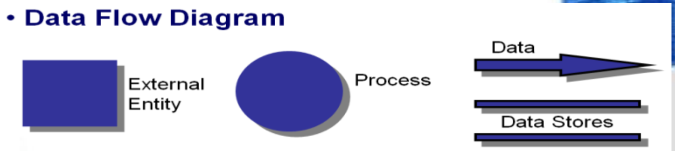
1. Please draw the data flow diagram for processing the garbage. (12 pts.)

Answer:

---Format like：

Book request = Find book position + Get a book



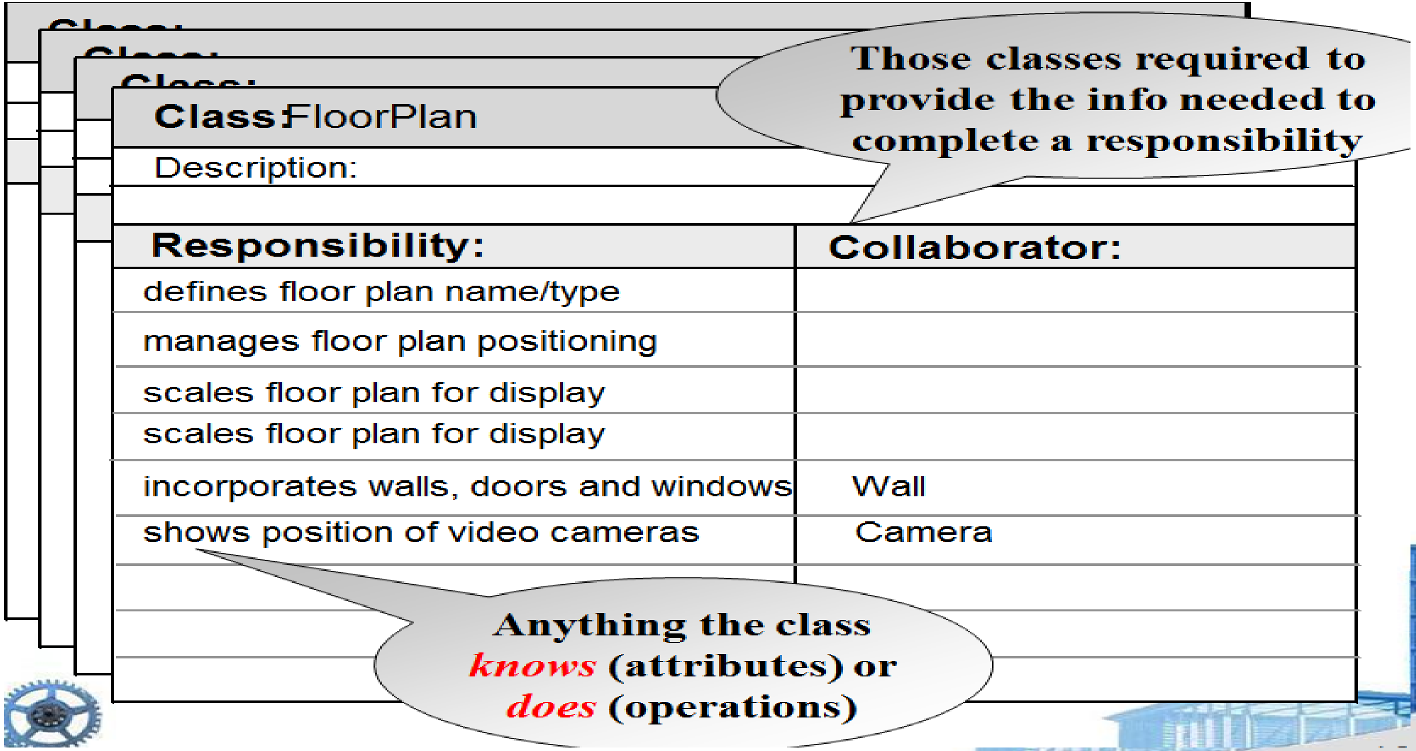


Answer(2)

2. Please give the two CRC cards for classes “customer” and “attendant” (10 pts.)

Answer:

---Format like:



Class: FloorPlan

Description:

Responsibility:	Collaborator:
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

Those classes required to provide the info needed to complete a responsibility

Anything the class *knows* (attributes) or *does* (operations)

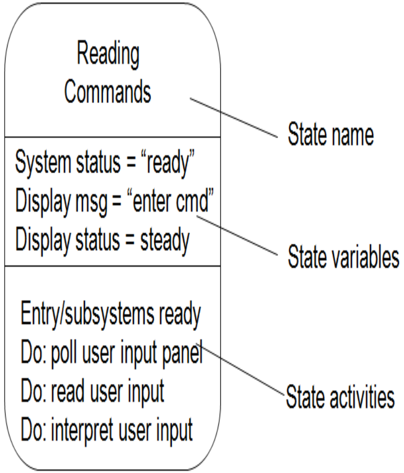
Answer(3)

3. Please give the state diagram for the “order” class. (8 pts.)

Answer:

---Format like:

State Diagram



Reading Commands

System status = “ready”

Display msg = “enter cmd”

Display status = steady

Entry/subsystems ready

Do: poll user input panel

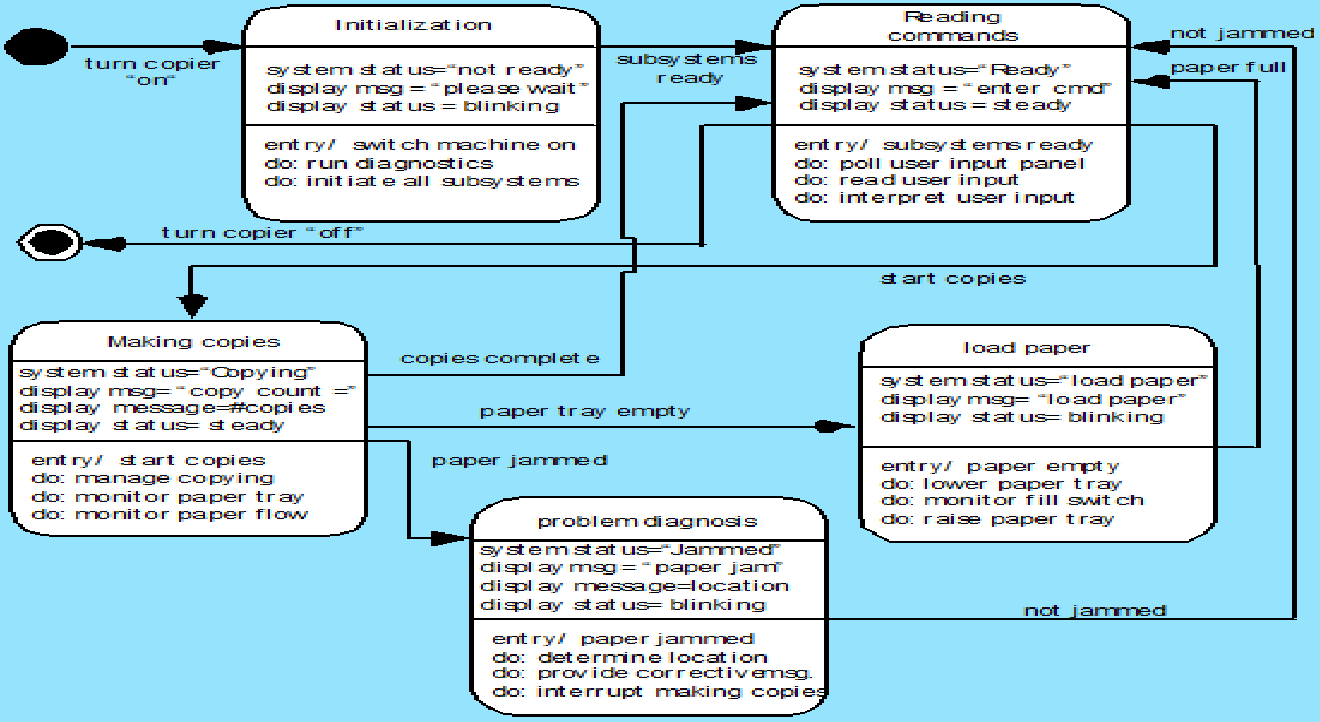
Do: read user input

Do: interpret user input

State name

State variables

State activities



Initialization

system status=“not ready”
display msg = “please wait”
display status = blinking

entry/ switch machine on
do: run diagnostics
do: initiate all subsystems

Reading commands

system status=“Ready”
display msg = “enter cmd”
display status = steady

entry/ subsystems ready
do: poll user input panel
do: read user input
do: interpret user input

Making copies

system status=“Copying”
display msg= “copy count =”
display message=#copies
display status= steady

entry/ start copies
do: manage copying
do: monitor paper tray
do: monitor paper flow

load paper

system status=“load paper”
display msg= “load paper”
display status= blinking

entry/ paper empty
do: lower paper tray
do: monitor fill switch
do: raise paper tray

problem diagnosis

system status=“Jammed”
display msg = “paper jam”
display message=location
display status= blinking

entry/ paper jammed
do: determine location
do: provide correctivemsg.
do: interrupt making copies

turn copier “on”

subsystems ready

not jammed

paper full

turn copier “off”

start copies

copies complete

paper tray empty

paper jammed

not jammed

Figure 7.6 Preliminary UML state diagram for a photocopier

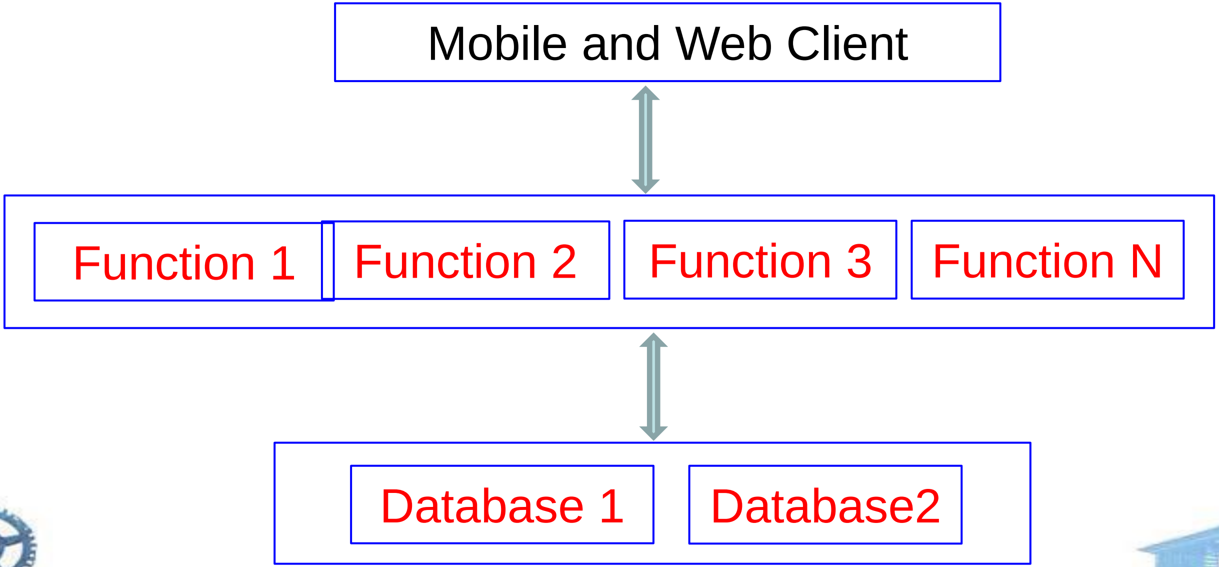
Answer(4)

4. Please draw the layered software architecture of GCSP. (10 pts.)

Answer:

---Format like:

GCSP



Mobile and Web Client

Function 1

Function 2

Function 3

Function N

Database 1

Database2

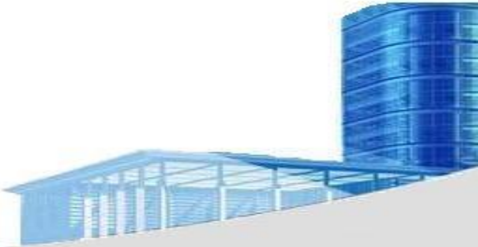
Answer(5)

5. Please describe the testing strategy for GCSP platform. (10 pts.)

Answer:

---Format like:

Component Testing (Unit Test),Content Testing, Interface Testing, Navigation Testing; Integration Testing, Regression Testing, Configuration Testing; Performance Testing, Security Testing, Certification Testing...



Tasks

- **Review** Ch. 35, 36
- **Finish** “Problems and points to ponder” in Ch.35, 36

- **Review** the whole Course**,** Do exercise on Course website

  ---(http://121.42.201.251/se/) “课后习题”

- **Show V 2 on June 3**

  ---时间：1）上午9:00始；2）课后11:35始；

  ---地点：曹西104室

- **Show the Function of Merge Version and PK on June 17 ?!**