COMPUTER ENGINEERING DEPARTMENT

ASSIGNMENT NO-01

SUB: Software Engineering

COURSE: T.E. **Year: 2020-2021** **Semester: VI**
DEPT: Computer Engineering
SUBJECT CODE: CSC601 **SUBMISSION DATE: 20/03/2021**

==========

Name: Amey Thakur **Roll No.: 50**

Class: TE Comps B-50 **ID: TU3F1819127**

Assignment No 1

Sr. No.	Question	CO mapping	PO mapping	PSO Mapping
1	What is Agile Methodology? Explain it with principles used. Explain any one agile methodology.	CO1	PO2,PO3, PO10, PO11, PO12	1
2	Differentiate the Waterfall model and Spiral model. Explain the advantages and disadvantages of each.	CO1	PO2,PO3, PO10, PO11, PO12	1
3	Define Requirement. What are the various types of requirements? Also, explain different requirement gathering techniques in detail.	CO2	PO1PO2, PO3,PO4, PO5,PO7, PO8,PO9, PO10, PO11, PO12	1

<table>
<tr><td>4</td><td>What is DFD?
What are the various elements of DFD?
Draw DFD for Online Course Enrolment system (Up to level 2) and also draw DFD for withdrawal of money using ATM.</td><td>CO2</td><td>PO1,PO2, PO3,PO4, PO5,PO7, PO8,PO9, PO10, PO11, PO12</td><td>1,2</td></tr>
<tr><td>5</td><td>What is an object Point?
Explain the COCOMO II model in detail</td><td>CO3</td><td>PO3,PO4, PO6,PO7, PO8,PO9, PO10, PO11, PO12</td><td>1,2</td></tr>
<tr><td>6</td><td>For CAD software, calculate the size of software using function point metric; using the following data.
<table><tr><th rowspan="2">Information Domain Value</th><th rowspan="2">Count</th><th colspan="3">Weighting factor</th></tr><tr><th>Simple</th><th>Average</th><th>Complex</th></tr><tr><td>No. of inputs</td><td>24 (20 simple, 2 Average,2 Complex)</td><td>3</td><td>4</td><td>6</td></tr><tr><td>No. of outputs</td><td>16 (14 simple, 1 Average, 1 Complex)</td><td>4</td><td>5</td><td>7</td></tr><tr><td>No. of enquiries</td><td>22 (18 simple, 3 Average, 1 Complex)</td><td>3</td><td>4</td><td>6</td></tr><tr><td>No. of files</td><td>4 (2 simple, 1 Average, 1Complex)</td><td>7</td><td>10</td><td>15</td></tr><tr><td>No. of External files</td><td>2 (1 simple, 1 Average)</td><td>5</td><td>7</td><td>10</td></tr></table>
The complexity weighting factors are as follows:
Backup and recovery: 4 Data communications: 2
Distributed processing: 0 Performance critical: 4
Existing operating environment: 3 On-line data entry: 4
Input transaction over multiple screens: 5 Master files updated on-line: 3
Information domain values complex: 5 Internal processing complex: 5
Code designed for reuse: 4 Conversion/installation in design: 3
Multiple installations: 5 Application designed for change: 5</td><td>CO3</td><td>PO3,PO4, PO6,PO7, PO8,PO9, PO10, PO11, PO12</td><td>1,2</td></tr>
</table>

AMEY TE B 50 Amey

Page No.: Date: YOUVA

Q1. What is Agile Methodology ?
Explain it with principles used.
Explain any one Agile Methodology.

Ans:

Agile Methodology

- The Agile process, light-weight methods are People-based rather than plan-based methods.
- The agile process forces the development team to focus on software itself rather than design and documentation
- The agile process believes in iterative method.
- The aim of agile process is to deliver the working model of software quickly to the customer.

For example : Extreme programming is the best known of agile process.

Principles of agile methodology.

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of monts, with a preference to the shorter timescale.
- Business people and developers must work together daily through the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

AMEY TE B 50 Amey

Page No.: Date: YOUVA

- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity — the are of maximizing the amount of work not done — is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Example

Extreme Programming

- Extreme Programming (XP) is one of the best known agile process. The extreme programming approach was suggested by Kent Beck in 2000.
- Extreme Programming is explained as follows:

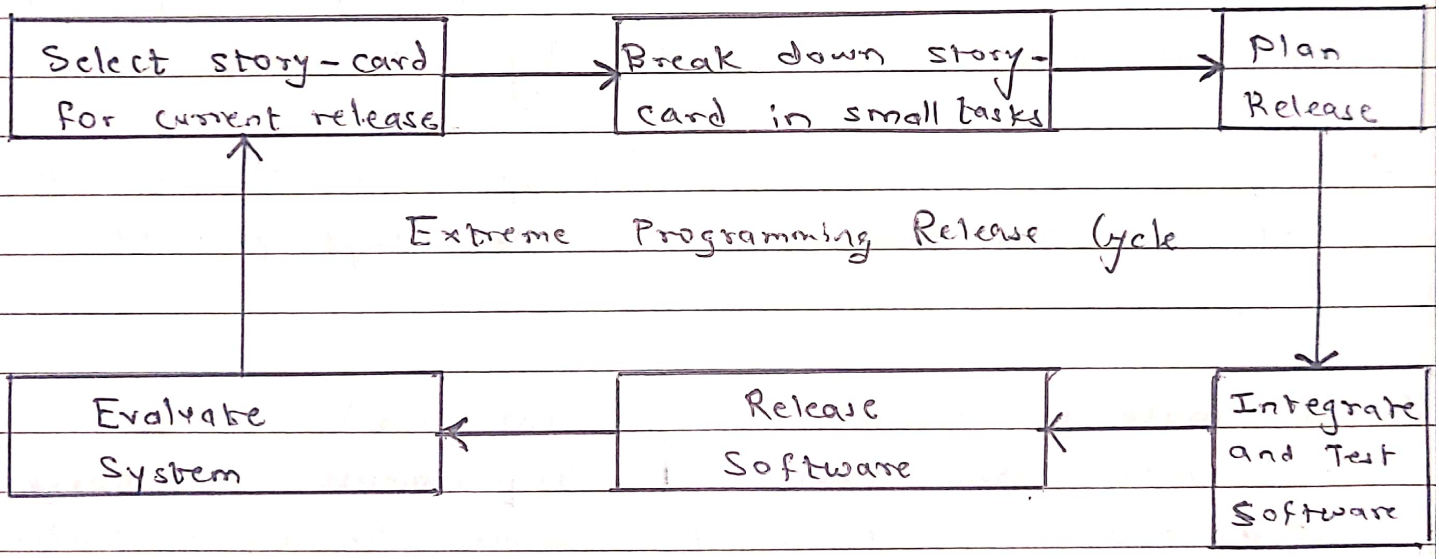
- Customer specifies and prioritize the system requirements. Customer becomes one of the important members of development team. The developer and customer together prepare a story-card in which customer needs are mentioned.
- The developers team then aims to implement the scenarios in story-card.

AMEY TE B 50 Amey

Page No.:
Date:
YOUVA

- After developing the story-card the development team break downs the total work in small tasks. The efforts and the estimated resources required for these tasks are estimated.
- The customer priorities the stories for implementation. If the requirement changes then sometimes unimplemented stories have to be discarded. Then release the complete software in small and frequent releases.
- For accommodating new changes, new story-card must be developed.
- Evaluate the system along with customer. Process is demonstrated by the following figure. —

Extreme programming release cycle.



AMEY TE B 50 Amey

Page No.:
Date:
YOUVA

Q2 Differentiate the Waterfall model and Spiral model. Explain the advantages and disadvantages of each.

Waterfall Model	Spiral Model
① Waterfall model works in sequential method.	① Spiral model works in evolutionary method.
② In waterfall model, errors or risks are identified and rectified after the completion of stages.	② In spiral model, errors or risks are identified and rectified earlier.
③ Waterfall model is adopted by customers	③ Spiral model is adopted by developers
④ Waterfall model is applicable for small project.	④ Spiral model is used for large project.
⑤ In waterfall model, requirements and early stage planning is necessary.	⑤ In spiral model, requirements and early stage planning is necessary if required.
⑥ Flexibility to change in waterfall model is difficult	⑥ Flexibility to change in spiral model is not difficult.
⑦ There is high amount risk in waterfall model	⑦ There is low amount risk in spiral model.
⑧ Waterfall model is comparatively inexpensive.	⑧ Cost of spiral model is very expensive.

AMEY TE B 50 Amey

Page No.: Date: YOUVA

Waterfall Model

Advantages:

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones
- Easy to arrange tasks.
- Process and results are well documented.

Disadvantages:

- No working software is produced until late during the life cycle.
- High amounts of risks and uncertainty.
- Not a good model for complex and object-oriented projects
- Poor model for long and ongoing projects
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a big-bang. At the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

AMEY TE B 50 Amey

Page No.:
Date:
YOUVA

Spiral Model

Advantages:

- Changing requirements can be accommodated.
- Allows extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

Disadvantages:

- Management is more complex.
- End of the project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.
- Process is complex
- Spiral may go on indefinitely.
- Large number of intermediate stages requires excessive documentation.

AMEY TE B 50 Amey

Page No.:
Date:
youva

Q3 Define Requirement. What are the various types of requirements? Also, explain different requirement gathering techniques in detail.

Ans:

Requirement:

- The software requirements are description of features and functionalities of the target system.
- Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.
- A condition or capability needed by a user to solve a problem or achieve an object.
- A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents.

Types of Requirements:

1. Functional Requirements
2. Non-functional requirements
3. Domain Requirements.

① Functional Requirements:

- These are the requirements that the end user specifically demands as basic facilities that the system should offer.
- All these functionalities need to be necessarily incorporated into the system as a part of the contract.
- These are represented or stated in the form of input to be given to the system, the operation performed and the output expected.

- They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements
- Example: Banking System
  Functional Requirements: User Bank Account Creation, Withdrawal of money, Cheque Clearance.

(2) Non-Functional Requirements
- These are basically the quality constraints that the system must satisfy according to the project contract.
- The priority or extent to which these factors are implemented varies from one project to the other.
- They basically deal with issues like:
  1. Portability
  2. Security
  3. Maintainability
  4. Reliability
  5. Scalability
  6. Performance
  7. Reusability
  8. Flexibility
- Example: ATM System
  Non-Functional Requirement: System must be available for 24*7.

(3) Domain Requirements
- These are the requirements which are characteristic of a particular category or domain of project.
- The basic functions that a system of a specific domain must necessarily exhibit come under this category.

- Example: Academic System
  Domain Requirements: In an academic software that maintains record of a school or college, the functionality of being able to access the list of faculty and list of students of each grade is a domain requirement.

Requirement Gathering Techniques

1. One-on-one interviews
2. Group interviews
3. Facilitated Sessions
4. Joint Application Development (JAD)
5. Questionnaires
6. Prototyping
7. Use cases
8. Following People Around
9. Request For Proposals (RPFs)
10. Brainstorming

① One-on-one interviews

- The most common technique for gathering requirements is to sit down with the clients and ask them what they need.
- The discussion should be planned out ahead of time based on the type of requirements you're looking for.
- There are many good ways to plan the interview, but generally you want to ask open-ended questions to get the interviewee to start talking and then ask probing questions to uncover requirements.

② Group Interviews:

- Group interviews are similar to the one-on-one interview, except that more than one person is being interviewed -- usually two to four.
- These interviews work well when everyone at the same level or has the same role.
- Group interviews require more preparation and more formality to get the information you want from all the participants.
- You can uncover a richer set of requirements in a shorter period of time if you can keep the group focused.

③ Facilitated Sessions

- In a facilitated session, you bring a large group (five or more) together for a common purpose.
- In this case, you are trying to gather a set of common requirements from a group in a faster manner than if you were to interview each of them separately.

④ Joint Application Development (JAD):

- JAD sessions are similar to general facilitated sessions.
- However, the group typically stays in the session until the session objectives are completed.
- For a requirements JAD session, the participants stays in session until a complete set of requirements is documented and agreed to.

AMEY TE B 50 Amey

Page No.:
Date:
YOUVA

(5) Questionnaires
- Questionnaires are much more informal, and they are good tools to gather requirements from stakeholders in remote locations or those who will have only minor input into the overall requirements.
- Questionnaires can also be used when you have to gather input from dozens, hundreads, or thousands of people.

(6) Prototyping
- Prototyping is relatively modern technique for gathering requirements.
- In this approach, you gather preliminary requirements that you use to build an initial version of the solution a prototype.
- You show this to the client, who then gives you additional requirements.
- You change the application and cycle around with the client again.
- This repetitive process continues until the product meets the critical mass of business needs or for an agreed number of iterations.

(7) Use cases
- Use cases are basically stories that describe how discrete processes work.
- The stories include people (actors) and describe how the solution works from a user perspective.
- Use cases may be easier for the user to articulate, although the use cases may need to be distilled later into the more specific detailed requirements.

AMEY TE, B 50 Amey

Page No.:
Date:
YOUVA

(8) Following People Around:

- This technique is especially helpful when gathering information on current processes.
- You may find, for instance, that some people have their work routine down to such a habit that they have a hard time explaining what they do or why.
- You may need to watch them perform their job before you can understand the entire picture. In some cases, you might also want to participate in the actual work process to get a hands on feel for how the business function works today

(9) Request For Proposals (RPFs)

- If you are a vendor, you may receive requirements through an RPF.
- This list of requirements is there for you to compare against your own capabilities to determine how close a match you are to the clients needs.

(10) Brainstorming

- On some projects, the requirements are not "uncovered" as much as they are "discovered."
- Brainstorming sessions generate a lot of material that must be filtered and organized
- Brainstorming works by focusing on the problem, and then deliberately coming up with as many solutions as possible and by pushing the ideas as far as possible.
- There are four basic rules in brainstorming.
  1. No criticism
  2. Welcome unusual Ideas
  3. Quantity wanted
  4. Combine and improve ideas.

AMEY TE B 50 Amey

Page No.:
Date:
YOUVA

Q.4 What is DFD ? What are the various elements of DFD? Draw DFD for Online Course Enrolment System (up to level 2) and also draw DFD for withdrawal of money using ATM.

Ans:

Data Flow Diagram (DFD)

- Data Flow Diagram is a graphical model of system, which shows what are the various functions (activities) performed by the system and how data flows among various functions.
- Each function is considered as a separate process.
- A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured Design).
- On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.
- A DFD provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel.
- It is therefore, quite different from a flowchart, which shows the flow of control through an algorithm allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored. (all of which are shown on a DFD).

AMEY TE B 50 Amey

Page No.:
Date:

Elements of Data Flow Diagram

① ═══ → Database store

Data Store - A data store is a holding place for information within the system.

② ⬭ → Process

Process - A process performs transformation or manipulation on input data and produces output data.

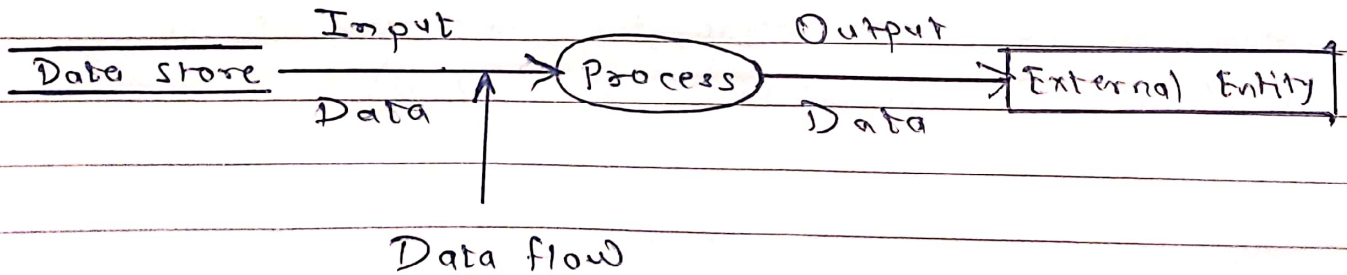
③ ▭ → External Entity

Entity - An entity is anything that interact with system and is source or destination of the data

④ ⟶ → Data Flow

Data Flow - A data flow shows shows the flow of information from its source to its destination.

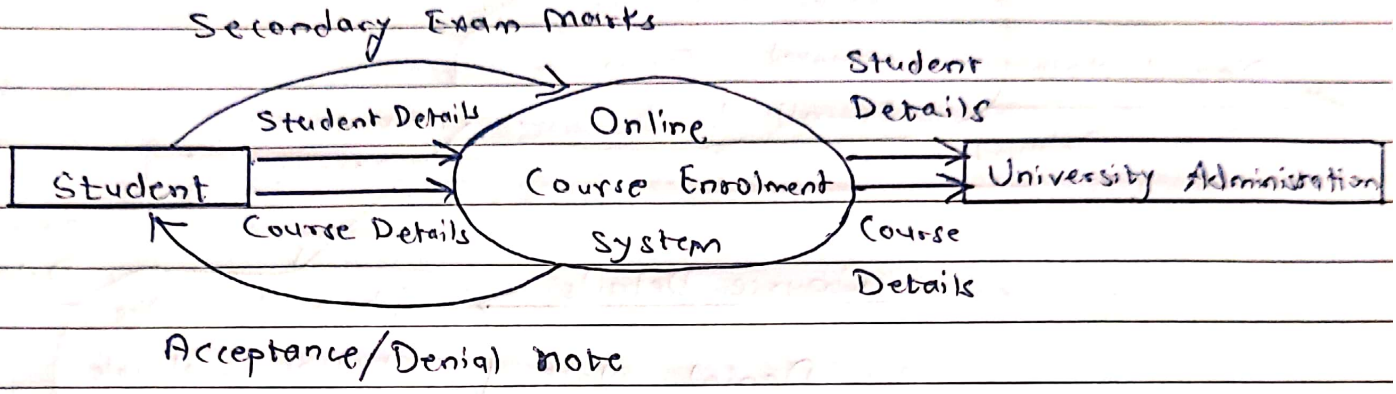
Data Flow Diagram Example



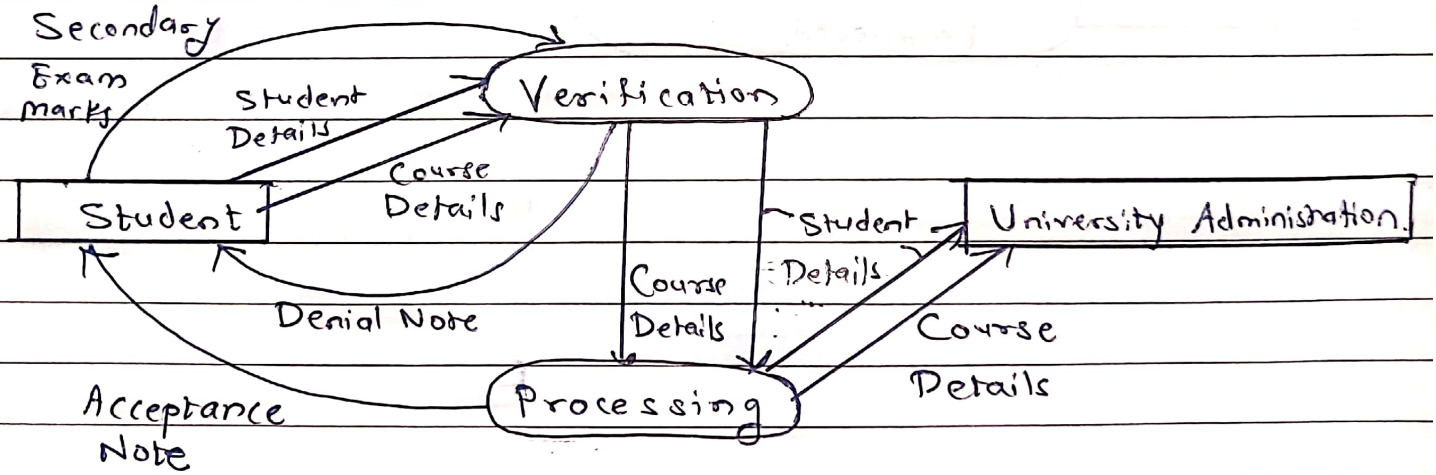
AMEY TE B 50 Amey

DFD for online course enrolment system.

- Level 0 DFD or Context Level DFD



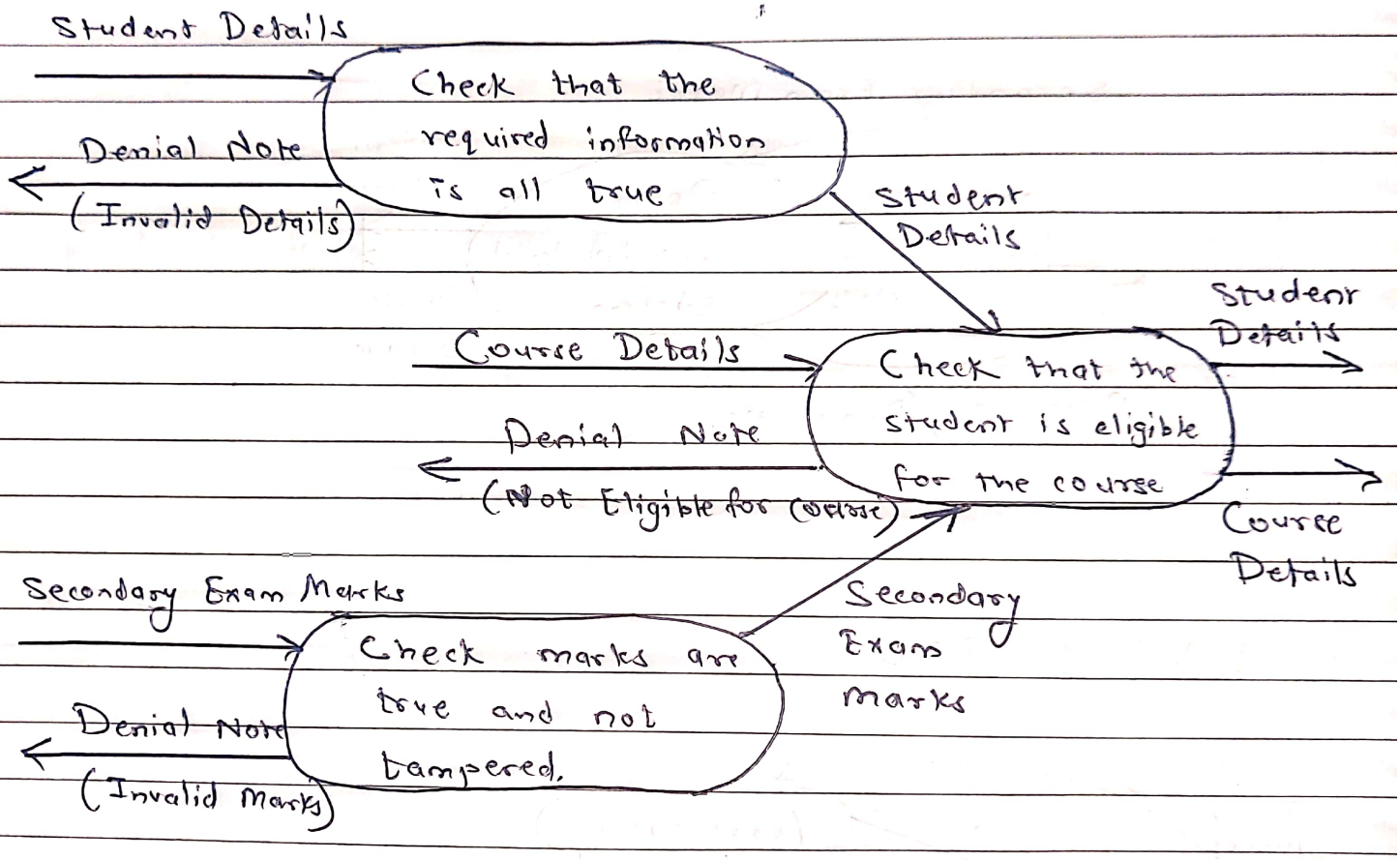
- Level 1 DFD.



AMEY TE B 50 Amey

Page No.:
Date:
YOUVA

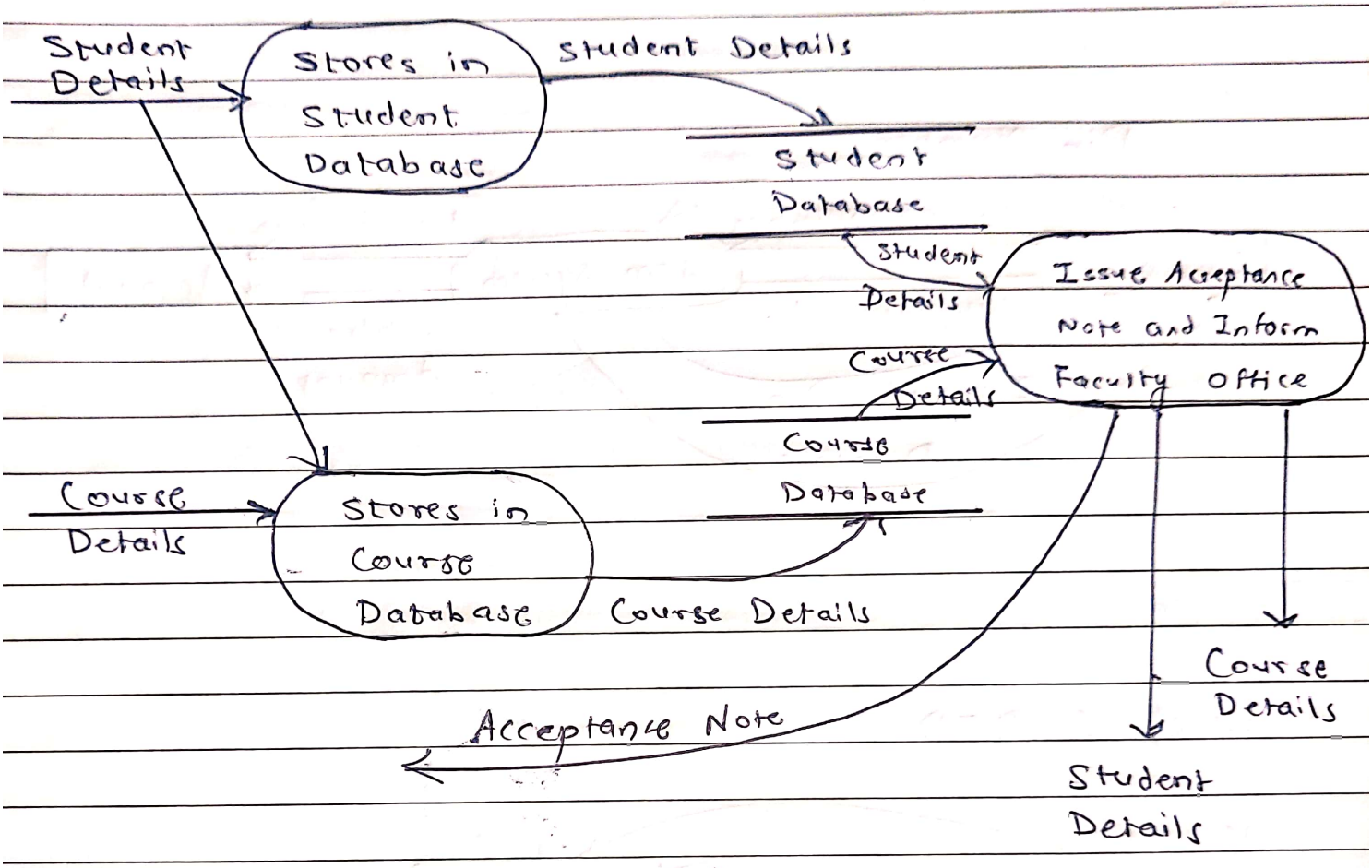
- Level 2 DFD for Verification Process of Level 1 DFD



AMEY TE B 50 Amey -

Page No.:
Date:

- Level 2 DFD for Processing process of Level 1 DFD

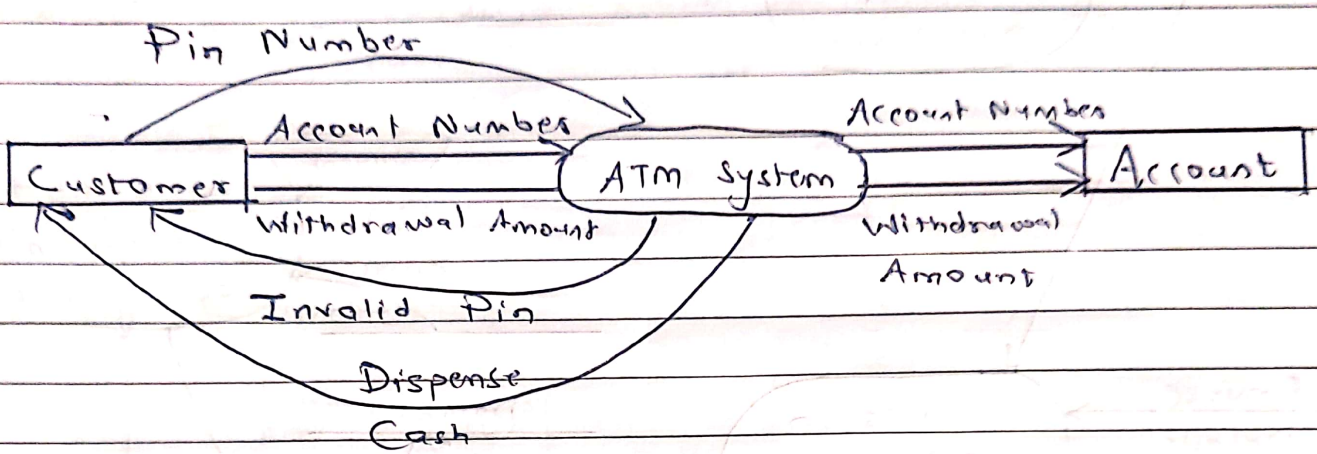


AMEY TE B 50 Amey

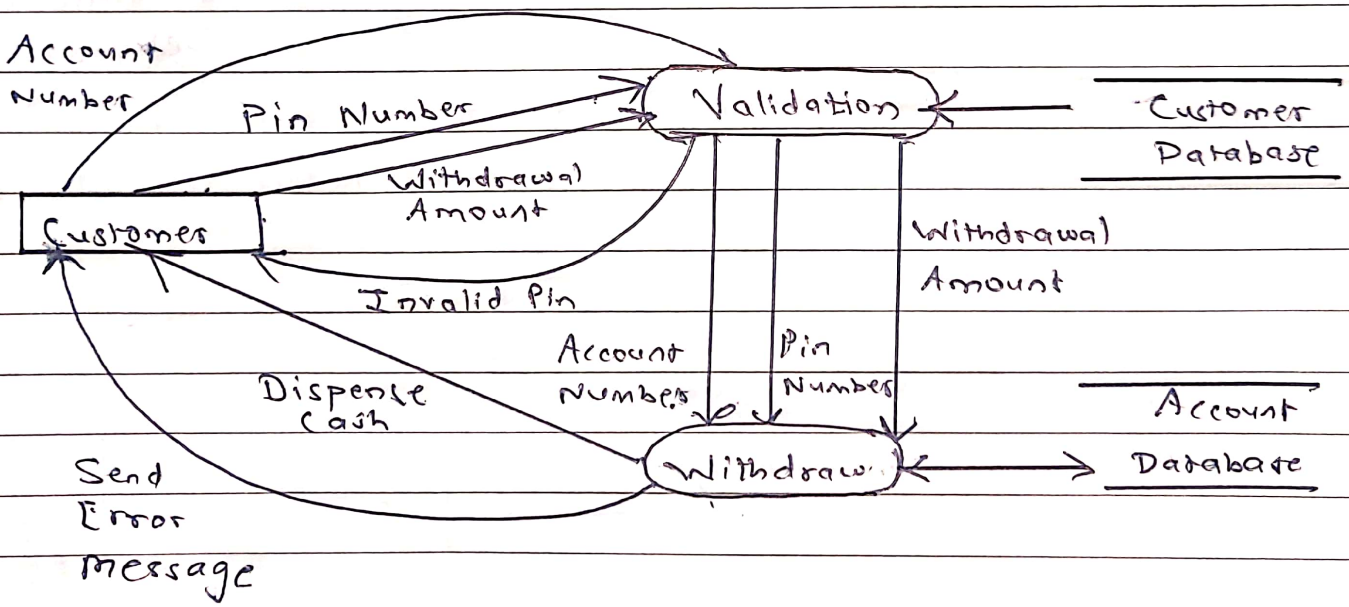
Page No.:
Date:
YOUVA

DFD for (withdrawal) of money using ATM

- Level 0 DFD



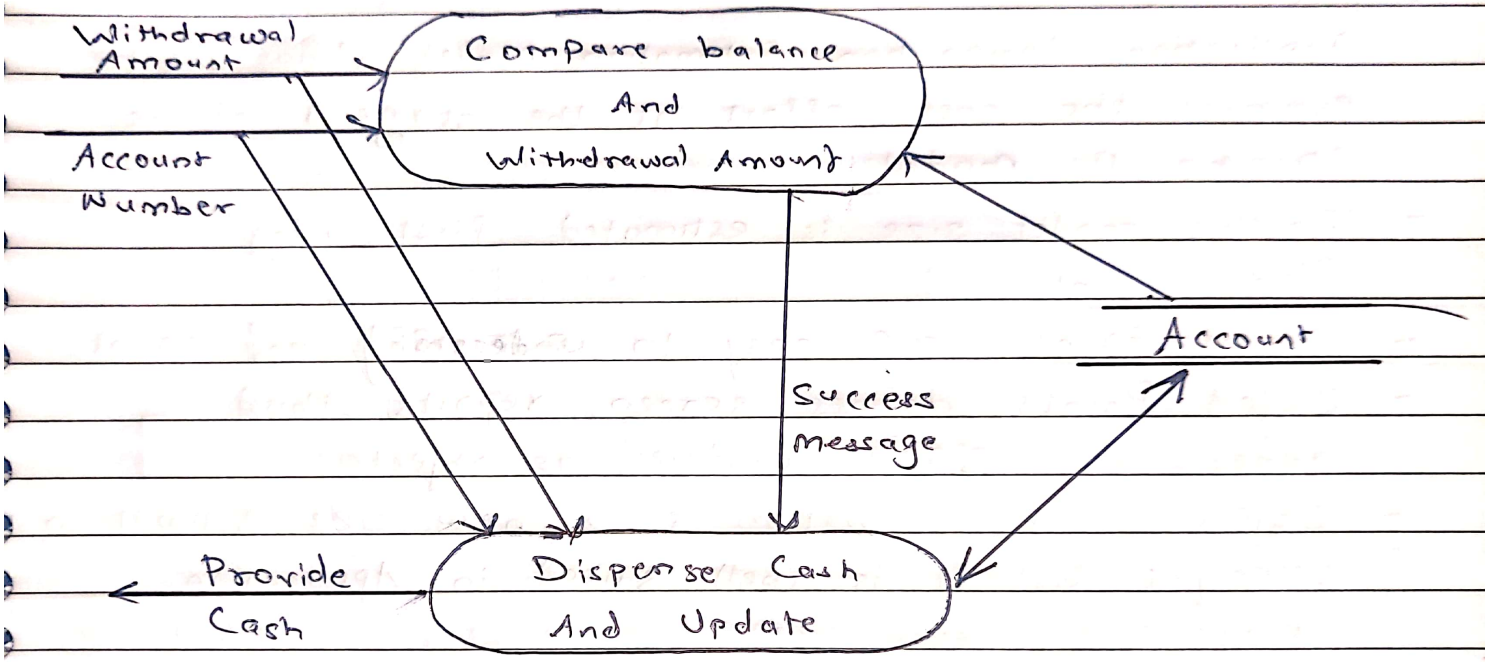
- Level 1 DFD



AMEY TE B 50 Amey

Page No.:
Date:
YOUVA

- Level 2 DFD for withdrawal process of level 1 DFD



AMEY TE. B 50 Amey

Page No.:
Date:
YOUVA

Q5. What is an object point? Explain the COCOMO II model in detail.

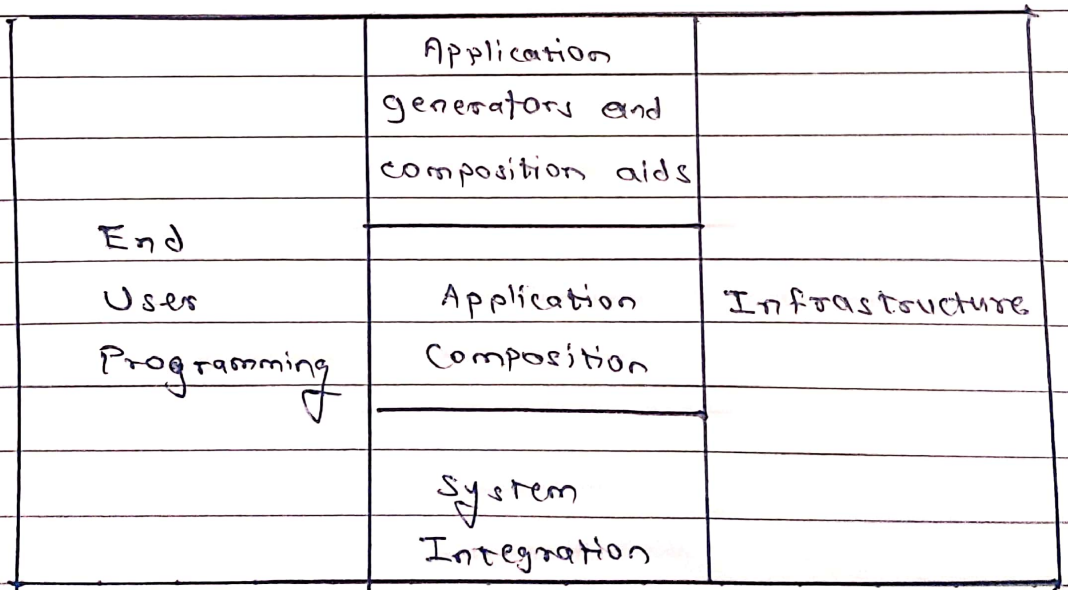
Ans:

Object Point:

- Application Composition Estimation Model allows one to estimate the cost, effort at the stage 1 of the COCOMO II model.
- In this model size is estimated first using Object Points.
- Object Points are easy to identify and count
- Object Points defines screen, reports, third generation (3GL) modules as object.
- Object Point estimation is a new size estimation technique but it is well suited in Application Composition sector

COCOMO II Model

- COCOMO - II is the revised version of the original COCOMO (Constructive Cost Model) and is developed at University of Southern California. It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity.
- It consists of three sub - models:



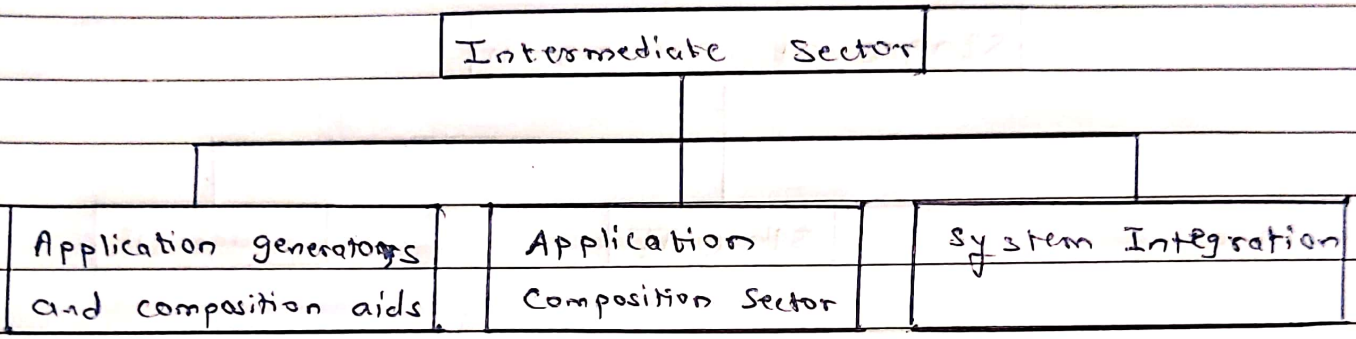
AMEY TE B 50 Amey

Page No.:
Date:
youva

① End User Programming:

- Application generators are used in this sub-model. End user write the code by using these application generators.
- Example: Spreadsheets, Report generator, etc.

② Intermediate sector:



A. Application Generators and Composition Aids

- This category will create largely prepackaged capabilities for user programming. Their product will have many reusable components. Typical firms operating in this sector are Microsoft, Lotus, Oracle, IBM, Borland, Novell.

B. Application Composition Sector

- This category is too diversified and to be handled by prepackaged solutions. It includes GUI, Databases, Domain specific components such as financial, medical or industrial process control packages.

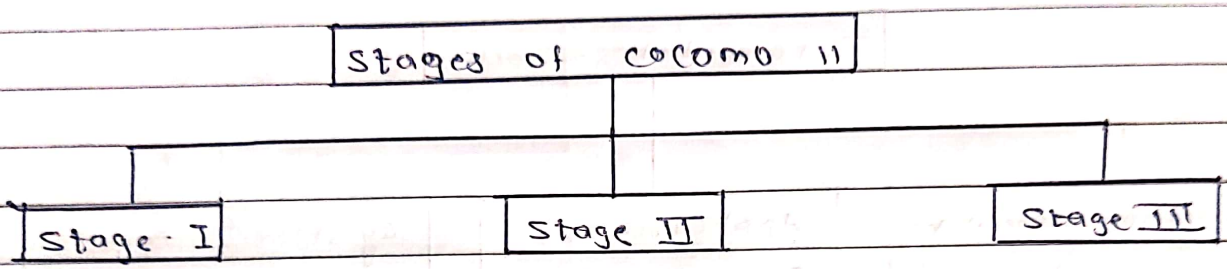
C. System Integration

- This category deals with large scale and highly embedded systems.

③ Infrastructure Sector:

- This category provides infrastructure for the software development like Operating System, Database Management System, User Interface Management System, Networking System, etc.

Stages of cocomo II:



A] Stage - I

- It supports estimation of prototyping. For this it uses Application Composition Estimation Model. This model is used for the prototyping stage of application generator and system integration.

B] Stage - II

- It supports estimation in the early design stage of the project, when we less know about it. For this it uses Early Design Estimation model. This model is used in early design stage of application generators, infrastructure, system integration.

C] Stage - III

- It supports estimation in the post architecture stage of a project. For this it uses Post Architecture Estimation Model. This model is used after the completion of the detailed architecture of application generator, infrastructure, system integration.

AMEY TE B 50 Amey

Page No.: Date: YOUVA

Q6. For CAD Software, calculate size of software using function point metric; using following data.

Ans:

Information Domain Value	Count		Weighting Factor			
			Simple	Average	Complex	
EI	24	×	(3)	4	6	= 72
EO	16	×	(4)	5	7	= 64
EQ	22	×	(3)	4	6	= 66
ILF	4	×	(7)	10	15	= 28
EIF	2	×	(5)	7	10	= 10
Count Total →						240

$$FP = \text{Count Total} * [0.65 + 0.01 * \Sigma(F_i)]$$

$$\Sigma(F_i) = 4+0+3+5+5+4+5+2+4+4+3+5+3+5$$
$$= 52$$

$$FP = 240 * [0.65 + 0.01 * 52]$$
$$= 280.8$$