SHAJIB = IT-21042

1, User story 1: log in securely

Tasks: 1. Design login page UI
2. Implement login functionality.
3. Implement password encryption
4. write unit tests for login

User story 2: search for product by category

Tasks: 1. Design search UI with filters
2. Implement search API
3. add category filtering
4. write unit tests for search

Tracking on Scrum Board:

1. To Do: Tasks planned but not yet started
2. In progress: Tasks actively being worked on
3. Done: completed task

Prioritization During Sprint planning:

(*) User story 1 (login): Higher priority due to basic e-commerce
(*) User story 2 (search): High value but secondary to login

SHAHIB
IT-21042

Q4 Risk Management ad Adaptability:

* Spiral Methodology: Focuses heavily on risk assessment and management through iterative cycles.
* Agile Methodology: Emphasizes flexibility and iterative progress through short time-boxed sprints. It allows for constant feedback.
* Extreme Programming (XP): An Agile-based approach that focuses on technical excellence, continuous feedback and close collaboration with the client.

Most Suitable Methodology:

Agile would be the most suitable for a project with significant risk and evolving requirement. It allows for continuous feedback, iterative development and flexibility, addressing both the uncertainty in client needs and the high risks.

3. Comparison of Methodologies:

* Waterfall: Almeare, Sequential model best for projects with well-defined requirement and a clear, fixed timeline.
* Agile: Iterative and flexible, ideal for projects with evolving requirement and frequent customer feedback.
* Extreme Programming (XP): A more intense form of agile, focused on high-quality code, constant customer collaboration and rapid iterations.
* Spiral: Combines iterative development with risk management its ideal for high risk projects where frequent adjustm are needed.

Conclusion:

* Project A: Waterfall is best for predict -bility and meeting strict deadlines
* Project B: Agile or spiral is best for adaptability and managing evolving req

Principles of Software engineering Ethics:

1. public welfare: prioritize and safety
2. Honesty and Intergrity: Be truthful in all profession communication
3. Quality and competence: Strive for excellence and maintain skill.
4. Confidentiality: protect sensitive client information
5. Professional Responsibility: Be accountable for your work

Issues Related to professional Responsibility:

* Conflicts of Interest: Act in way benefit
* Accountability: Be truthful and
* Intellectual property:

ACM/IEEE code of ethics:

* public Interest: Accept responsibility for
* Honesty and Integrity:
* Accountability: Only take on tasks
* Competence:

SHADIB
II-21042

5. Functional Requirements:

1. Flight Reservation: User can search and book flight based on criteria like destination and date.
2. User Authentication: users must login to access and manage
3. payments Integration: user can securely pay for booking.
4. Booking Modification: user can modify or cancel.
5. Flight Availability Notifications: The system should notify users of flight availability and changes

Non Functional:

1. preformance (Response time)
2. Scalability: should handle large data
3. Security;
4. Usability;
5. Maintainability;

SHADIB
IT-21042

6.) V-Model of testing phases:

1. Requirements Analysis → Acceptance Testing
   * Development: Gather and define requir—
   * Testing: Acceptance tests
2. System Design → System testing
   * Development: Design architecture and components
   * Testing:
3. High-Level Design → Integration Testing
4. Low-level Design → Unit Testing
5. Implementation → Coding Reviews and unit testing

Explanation:

The V-Model highlights parallelism between development and testing. As each phase of development is completed.

7. Prototype Development process:

1. Requirement Gathering: Initial requirem are collected.
2. Developing the prototype: A quick, low fidelity version of the software
3. User Evaluation: The prototype is presented to users for feedback
4. Refining the prototype: Based on user feedback
5. Repeat: The cycles of feedback an refinement

Benefits of the prototyping Model:

* User feedback: Allows users to inte
* Risk Reduction: Helps identify ten
* Interative development: Enables gradual improvement and refinemen

SHAJIB
IT-21042

81 Process Improvement cycle in Software Engineering:

1. Assessment: Evaluate the current Software process to identify are
2. Goal setting: Define specific measurable objects for the
3. planing: Develop a plan for goals
4. Implementation: Apply the planned changes to the process.
5. Monitoring and Measurement: continuously track the effects of
6. Evaluation: Review the results of the changes
7. Refinement: Make further adjust

Commonly used process Metrics:

1. Defect Density: The number of defe per unit
2. cycle time: The amount of time
3. code coverage: The percentage code taken
4. customer satisfication

21 SEI capability Maturity Model (CMM):

1. Level 1: Initial
   * Description: process are unpredictable and ad hoc, with little to no
   * Contribution: At this level, organization often face
2. Level 2: Managed
   * Description: Basic project management
   * Contribution: Enables better controls
3. Level 3: Defined
   * Description: process are standardized
   * Contribution: creates a common tune
4. Level 4: Quantitatively Managed
   * Description: processes are measured
   * Contribution: Allow for data driven
5. Level 5: optimizing
   * Description: Focus on continuous
   * Contribution: Drives ongoing improver

Each level of the SEI CMM contributes t improved organization perfor mane

SHAJIB = II-24

10. Core principles of Agile software Development:

1. customer collaboration over contract Negotiation: Focus on continuous
2. Responding to change over following a plan: Adapt to changes in requirem
3. working software over comprehensive Documentation: prioritize delivering
4. Individuals and Interactions over process and tools: Emphasize.
5. Simplicity: delivering only what is
6. self-organization Teams:
7. Continuous Improvement:

Benefits of Agile:

(*) Flexibility:
(*) customer satisfaction
(*) Past delivery
(*) Improved collaboration

Application in Different Environment:

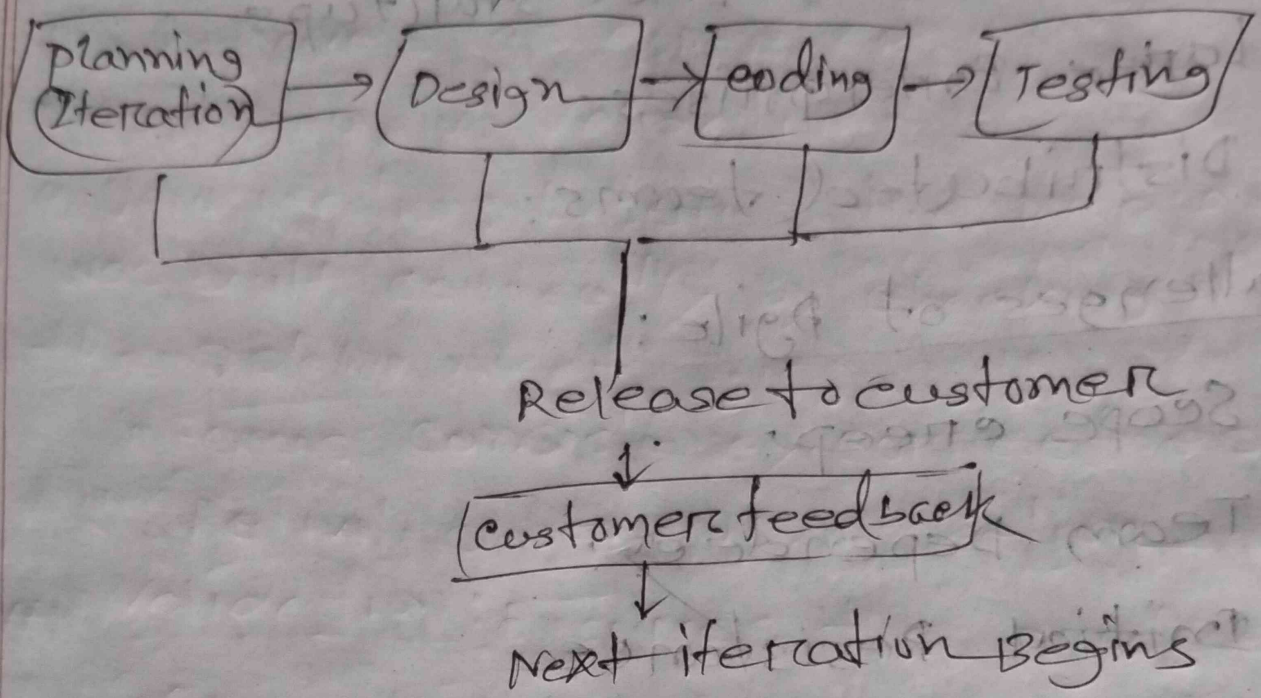
(*) Small Teams and startups
(*) Large Enterprises:
(*) Distributed teams:

challenges of Agile:

(*) Scope creep:
(*) Team Dependency:
(*) Initial Resistance
(*) Complex projects

The effectiveness of Agile varies depending on project size, team experience, and organization.

11. Here simple visual representation of the release cycle of extreme progra



key programming practices in XP:

1. Continuous Integration: Developers frequently integrate their code
2. Test-Driven Development (TDD): Writing tests before writing the code
3. pair programming: Two developer work Together
4. Refactoring: Continously improving the code structure
5. Simple Design: The simplest solution that works is preferred
6. Collective code ownership

12/ In the digital library management system the entities and their attributes can be structured as follows.

(1). Book

i Attributes:

* BookID * Title * Author * ISBM * Genre

ii Relationship

* A book can be borrowed by many member

(2) Member

(i) Attributes

(*) memberID (*) Name (*) membershipID

(ii) Relationship

(*) A member can borrow many books

(3) Borrowing

(i) Attributes:

(*) BorrowingID (*) Date (*) ReturnDate

(4) Overduebooks(optional, to track overdue book specifically)

(5) ERD overview:

(*) Book and member are connected

(*) Borrowing tracks individual

3. Testing is the process of evaluating a system or its components to ensure it behaves as expected

Verification vs validation:

(*) Verification: Ensures the product was built correctly by checking if the system meet.

(i) Focus: consistency, correctness

(ii) Activites: Reviews, inspection

(*) Validation: Ensures the product fulfiles its intended use and meet user need

(i) Focus: user requirement and real world usage

(ii) Activities: Testing user acceptance testing.

Verification check if the product was built correctly. while validation checks if it fulfill the intended purpose.

14. In a layered architecture for an online judge system

1. Presentation Layer: This layer handles user interactions, such as submitting problems, viewing results and accessing

2. Application Layer: This Layer process user requests from the presentation layer, orchestrates

3. Business Logic Layer: This Layer is resp for implementing core system Logic, inclu problem evaluation, test

4. Data Layer: This Layer manages the persistence of data, such as user pro problem sets, submission histories.

Scalability: Each layer can be scaled independently, ensuring that the syst can handle increasing traffic.

Maintainability: with clear spration of concerns, updates or changes in one lay

15] DFD (Data Flow Diagram)

Level 0: context Diagram

This level provides an overview of the hospital reception system

* Entities:
  (i) patient: provides personal inform
  (ii) Doctor: May provide appointment

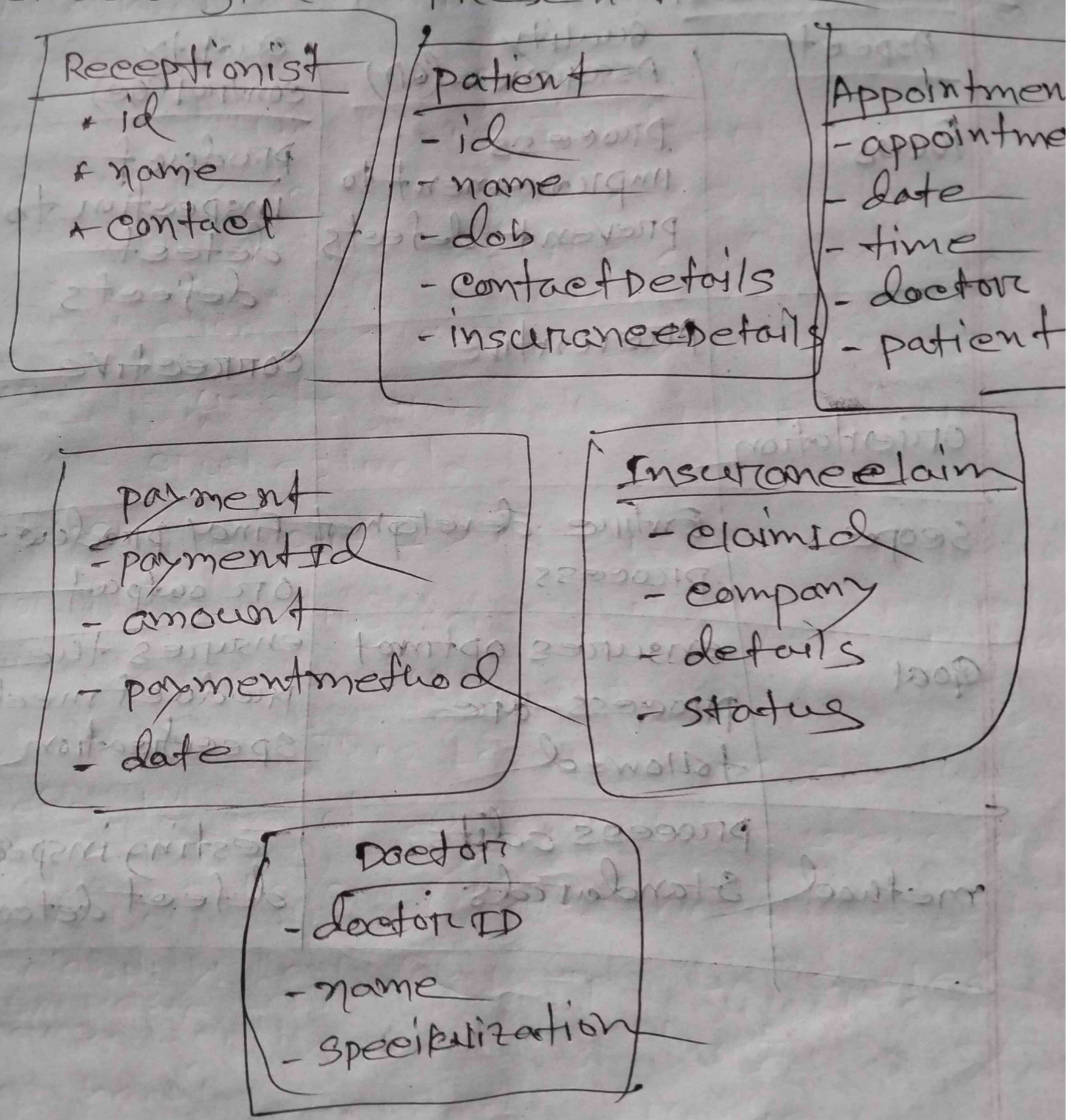
* process:
  (i) Hospital Reception system:

* Data stories:
  (i) patient database:
  (ii) Appointment database
  (iii) payment database

Level 1: Decomposition Diagram

This level breaks down the reception
1. Manage appointments
2. patient admission
3. payment processing
4. Insurance claims

16| class diagram Representation:



SHADIB
IT-21042

17) Differences:

Aspect	Quality Assurance (QA)	Quality control (QC)
Focus	process improvement to prevent defects	product inspection to detect defects
orientation	preventive	corrective
scope	Entire development process	Final product or output
Goal	ensures optimal process are followed	ensures the product meets specification
method	process audits, standards	Testing inspect defect detect

18) No, the goal of Quality Assurance is not just to find bugs early

Role of QA in Each SDLC phase:

1. Requirement Gathering: QA ensures that requirements are clear,
2. Design phase: QA reviews design docume for feasibility
3. Development phase: QA ensures developmen process are followed
4. Testing phase: QA conducts various test to detect defects
5. Development phase: QA ensures updates P
6. Maintence phase: QA tests updates, patel and new features, ensuring no new

Throughout all phases. QA's role is to ens Quality at every step