

3/2/2020

UNIT - 3

Agile Software Development

Different approach for rapid software development, they have some characteristics.

1. The process of specification and implementation are interleaved / inter-related
 - There no detailed sdm specification and no detailed design document
 - user req's document only defines the most sdm characteristics
2. Sdm is developed in a series of versions
 - End users and stakeholders are involved
 - They may propose changes to sdm in later versions.

3. S/W user interfaces are often developed using an interactive development S/W that allows the interface design to be quickly created by drawing and placing icons.

Issues with Rapid S/W development

1. Since it is inter-related we cannot directly go the other step
2. No proper documentation.
3. Difficult with interactive process.

Solution for this

Agile S/W Development

- Agile method universally rely on an incremental approach
- Best suited to application development where req's change rapidly
- They intend to deliver working S/W quickly
- Aim is to cut down on process by avoiding working that has long-term value and eliminating documents

* Agile System has been successful for some types of sl/m development

1. Product development : where a sl/w company is developing small or medium-sized product for sale.
2. customer sl/m development within an organisation where there is clear commitment from the customer

The Principles of Agile Methods.

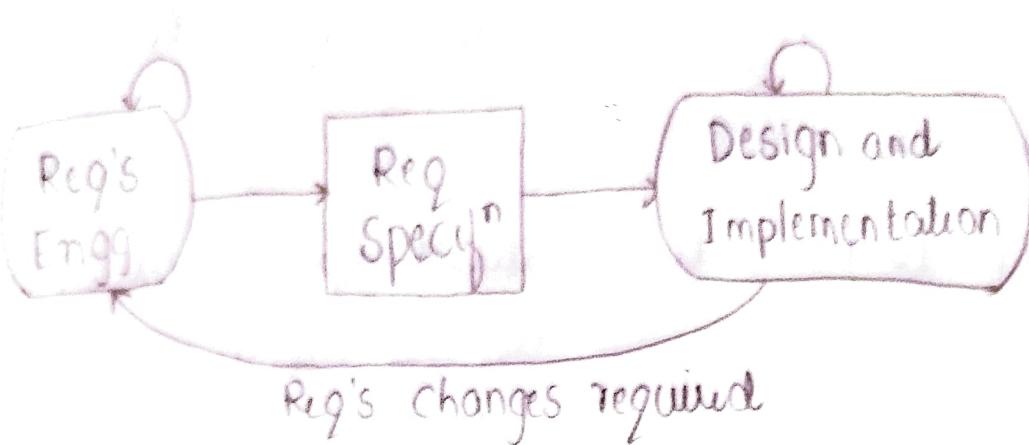
1. Customer involvement
2. Incremental delivery
3. People not process
4. Embrace changes
5. Maintain simplicity.

6/2/2020 :

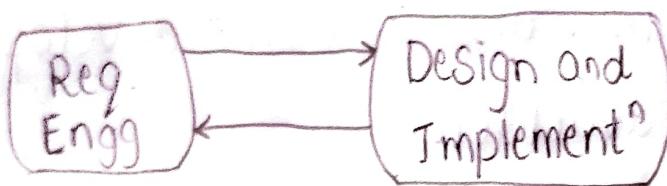
- * Another non-technical problem - general problem with incremental development and delivery occurs when the sl/m customers uses an outside organisation for sl/m development.

- * There are 2 Q's that should be considered when considering agile method and maintenance.
 1. Agile slm's that are developed using an agile approach maintainable?
 2. Can Agile methods be used effectively for evolving a slm in response to customer change requests.
- * Main difficulty after slw delivery is likely to be keeping customers involved in the process.
- * Other problems.
 - maintaining continuity of the development team
 - as team rely on team members, understanding aspect.
- * Plan - Driven & Agile development.
 - * Agile approaches to slw development consider design and implementation to be central activity
 - * Other activities are
 - req's elicitation and testing into design and implementation.

1 Plan Based Development



2 Agile Development



Plan Based Development

- 1 Iteration occurs with ~~activities~~ activities with formal documents used to communicate
- 2 Process cannot support incremental development and delivery.
- 3 Feasible to allocate req's and plan the design and development

Agile Development

- 1 Iteration occurs across activities, Req's and design are developed together.
- 2 Process is not inevitably code-focused and may produce some design
- 3 Mostly unplanned

To decide to balance between :

Plan based and Agile development , few Q's have to be answered.

1. Is it imp to have a detailed specification and design?
2. Is incremental delivery strategy where delivery of a sm and rapid feedback realistic
3. How large is the sm. that is being developed.
4. What technologies available?
5. Are there any cultural issues.
6. How good are designers?

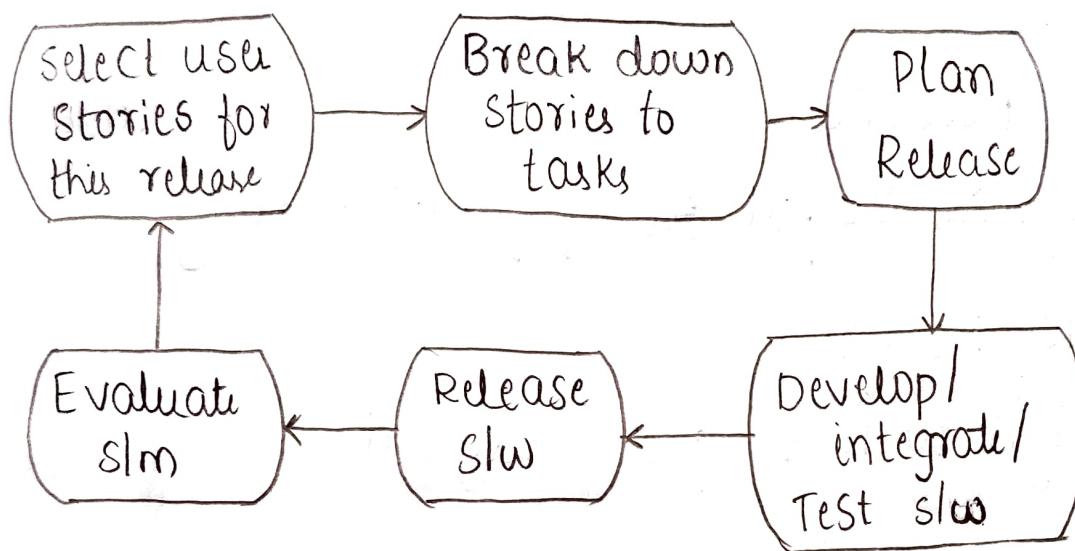
10/2/2020.

3 Extreme Programming (XP)

- Best known and most usually used ^{agile} method
- Name coined by Beck (2000)
- Approach was developed by pushing recognized good practice
- XP - requirements are expressed as scenarios (called user stories)

- Here programmers work in pair and develop test for each task before writing codes.
- All tests must be successfully executed when new code is integrated.
- XP - process to produce an increment of the S/w.

Extreme Programming Release cycle:



Principles of XP :

1. Incremental development is supported through small, frequent release of the s/w.
2. Customer support is supported throughout.
3. People → not process are supported.
4. Change is embraced through regular s/w release to customers

5. Maintaining simplicity is supported

Extreme Programming Practices:

1. Incremental Planning

- Req's are recorded on story cards.
- Stories to be included in release's
- Developer breaks the stories into development tasks.

2. Small Releases :

- minimal useful set of functionality

3. Simple Design :

- design to meet and carry out

4. Test first Development :

- Unit test framework is used.

5. Refactoring

- All developers are expected to refactor the code continuously.

6. Pair Programming :

- Developers work in pair.
- check each others work
- provide support.

1. Collective ownership:
 - Developers work on all areas of the system
 - Developers take responsibilities for all codes.
 2. Continuous integration
 3. Sustainable pace
 4. On-site customer.
-
1. Testing XP
 - Difference betⁿ incremental and plan based?
 - with incremental - issues.
 - To avoid some of problems of testing and system validation in XP.
 2. XP - emphasises the imp. of program testing
- Key features of testing in XP are
1. Test-first development
 2. Incremental test development from scenarios
 3. User involvement in the test development and validation.
 4. The use of automated testing framework

2. Pair programming:

- another practice introduced in XP.
- Programmers work in pair
- Same pair do not program together.

Advantages:

- 1 It supports collective ideas
- 2 Acts as informal review process
- 3 It supports refactoring.

17-2-2020.

SLM Modelling:

- The process of developing abstract models of a SLM
- Each model → diff represent a diff view or perspective
- Diff types or methods
 - i. Notation based
 - ii. Graphical based.

- * Models are used during reg's engg. process
- * Helps to derive reg's for slm.
- * You may develop models of both the existing slm and the slm to be developed.

1. Models of existing slm :

- used during reg's engg
- helps clarify about existing slm
- can be used as base.

2. Models of the new slm.

- used during reg's engg.
- helps to explain proposed reg's.
- used to discuss design proposal
- possible to generate complete or partial slm.

UML has 5 diagram types :

1 Activity Diagram :-
- shows interaction in a process or data.

2. Use Case Dgm:-
- shows interaction between slm and its environment.

3. Sequence Dgm : shows interaction between actors and the slm and betⁿ slm component.

4. Class Dgm :-

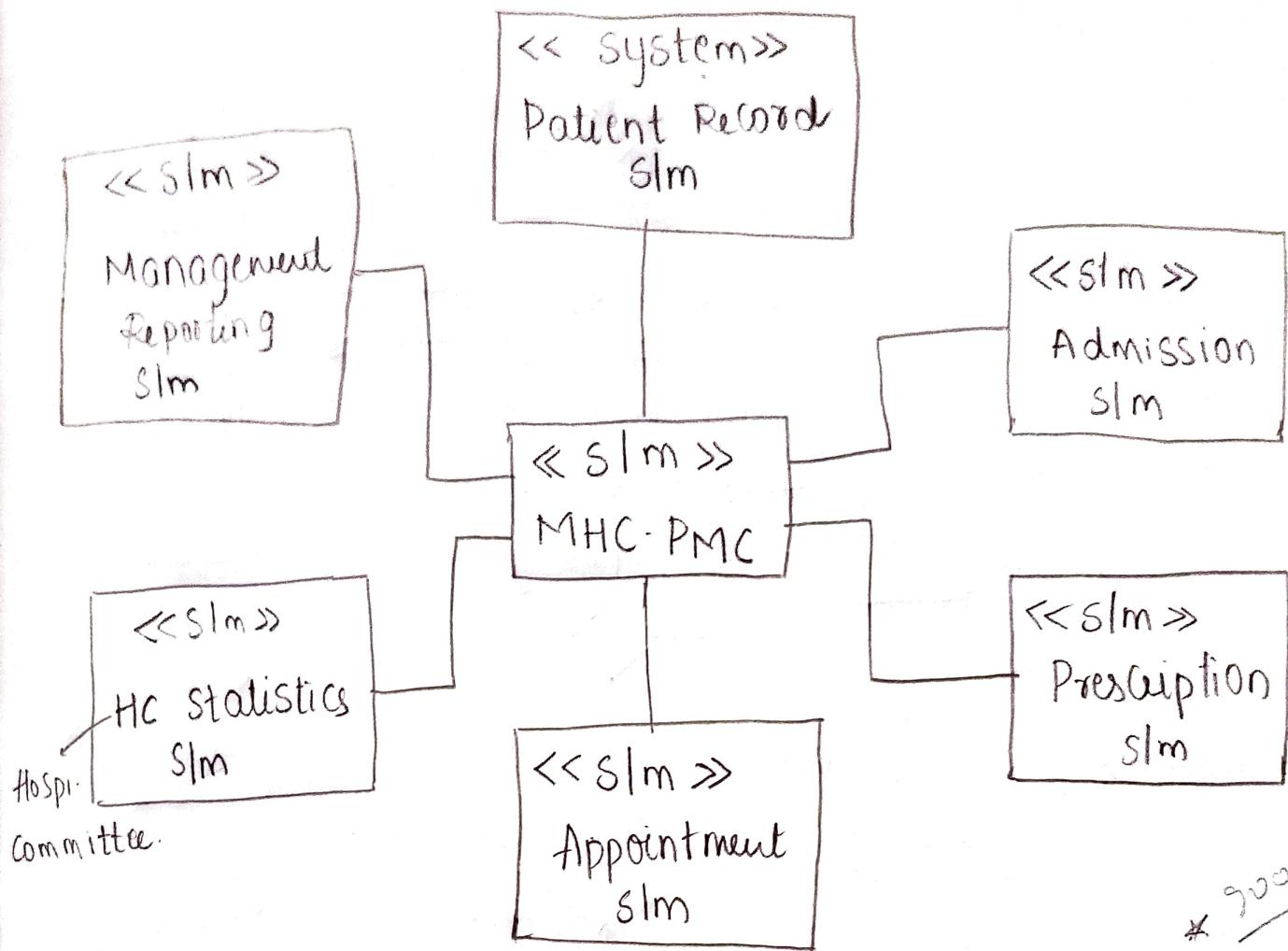
- Shows object class in the slm and the association betⁿ the classes

5 State Dgm :-

- Shows how the slm reacts to internal and external events

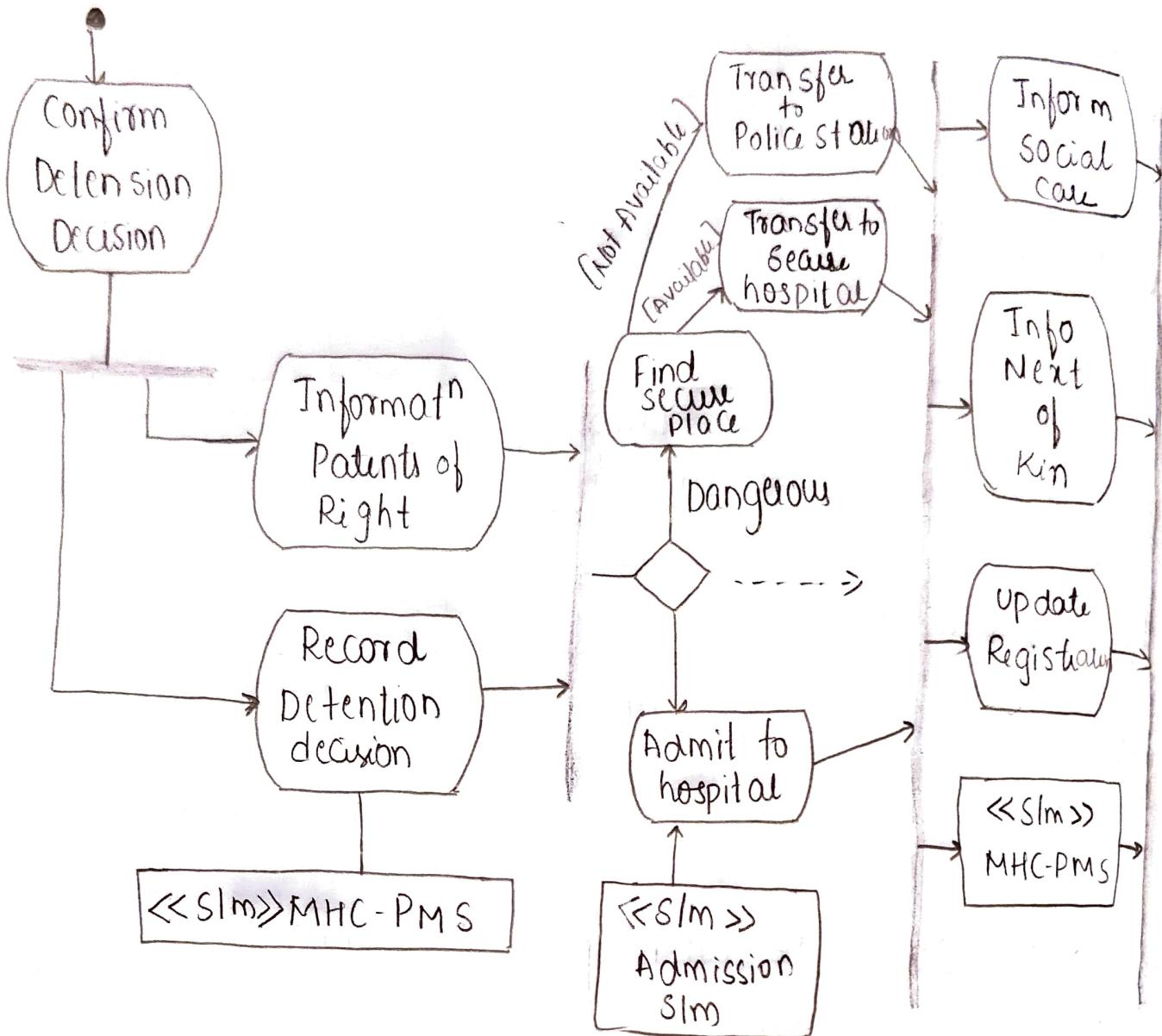
1. Context Model :

- Early stage - decide on slm boundaries
- Involves stakeholders to determine boundaries
- Look for possible overlaps in functionalities with existing system.
- shows the environment
- includes several other automated slm
- does not show the type of relationship between the slm and environment
- External slm might produce data for or consume data from the slm.



- data might be with SLM or connect directly through a network.
- SLM may be physically co-located or located in separate building.
- Used along other model.

Activity Diagram :



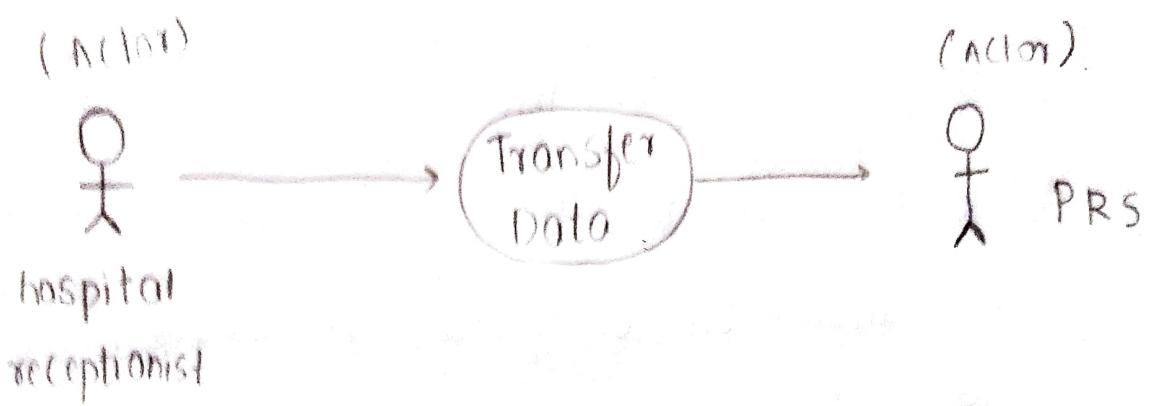
24 - 2 - 2020 :

- Interaction Models
- All slm involves interactions
- Involves user, slm, other interaction
- Involves i/p and o/p's
- Interaction betⁿ slm and the component being developed.

- Interaction is important.
 1. It helps to identify user requirement
 2. Helps to understand if a proposed slm structure working.
- It covers 2 related approaches.
 1. Use case modeling - used to model interaction between a slm and external actor.
 2. Sequence Diagram : Used to model interaction between slm, components, although external agents may be involved.
 - Involves high level use case.

1 Use-Case Model :

- i. Widely used to support req.s
- ii. Can be taken as scenario to describe what a user expects from a system.
- iii. Each use case represents a discrete task that is involved.



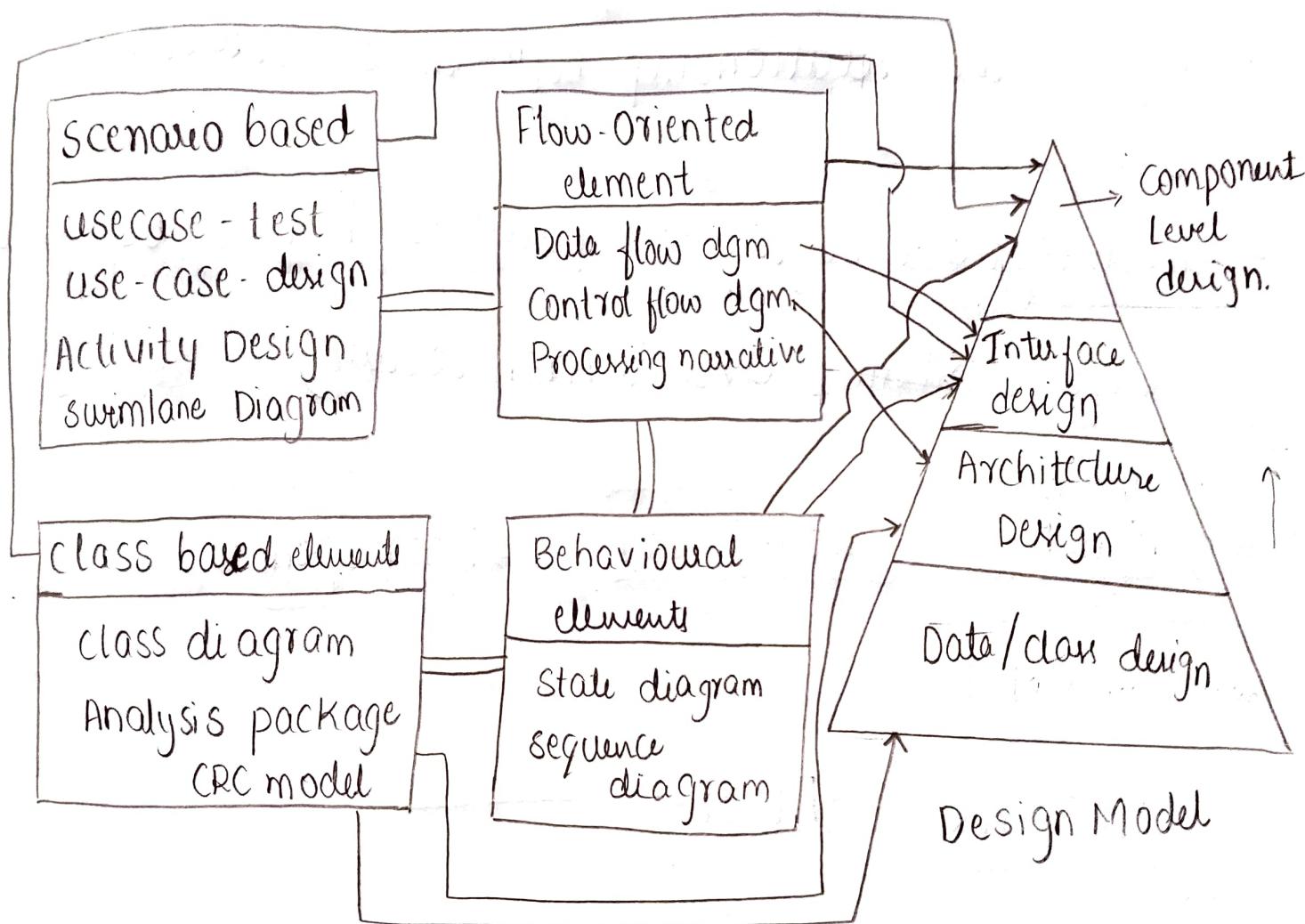
MHC - Reception Transfer Data

1. Actor : Hospital receptionist , patient record s/m.
2. Description : A receptionist - may transfer data to server.
 - i. Add patient record
 - ii. Update patient record
 - iii. delete
 - iv. Transfer
3. Data - patients info
4. Stimulus - Use command issued by receptionist
5. Response - Confirmation from PRS Patient Record s/m
6. Comments - Receptionist must have appropriate security permissions.

2-3-2020:

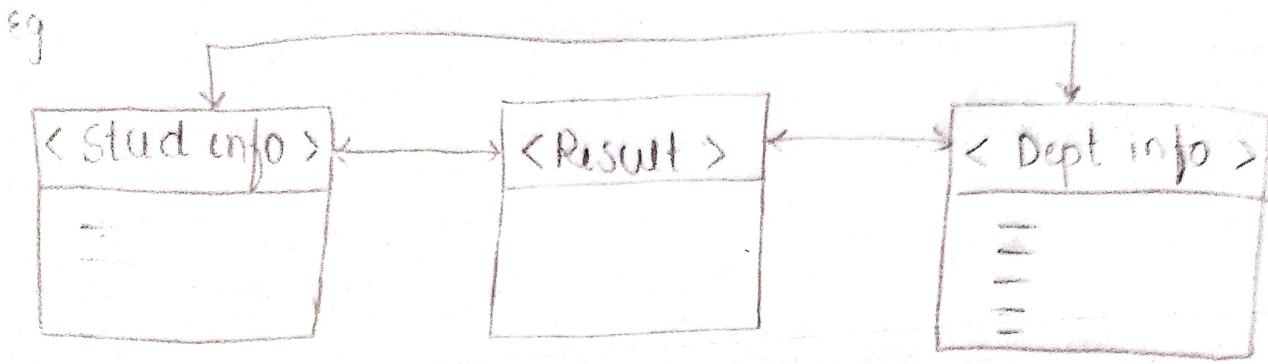
Design Concepts :

1. Design with the context of software Engg.
- * SW design is the last step engg action within the modeling activity.



1. Data / Class Design :

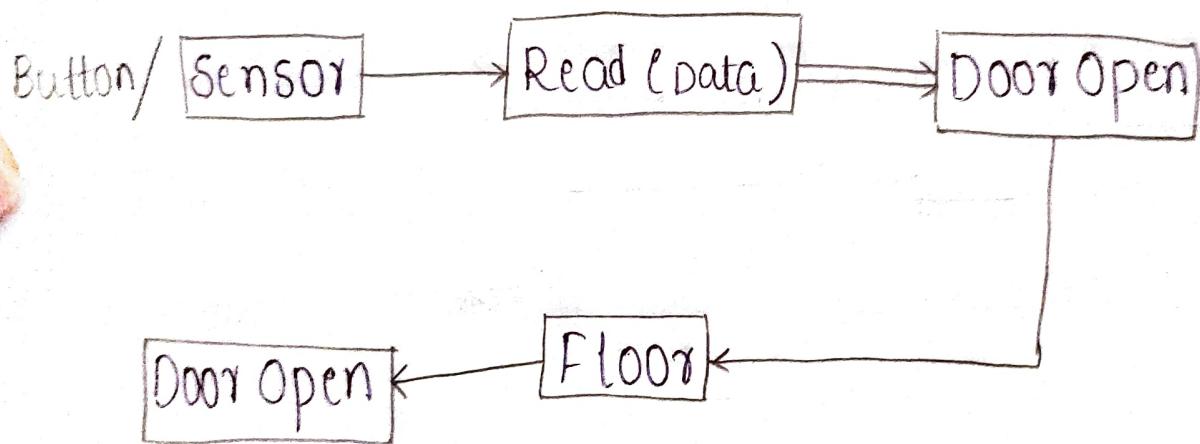
- * transforms class models to design class.
realisation and the request structure.



2. Architectural Design

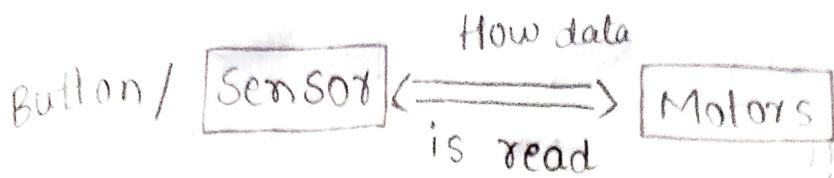
- * Defines the relationship bet'n major structural elements
- * Architectural style and patterns that can be used

Eg: Lift system (Overall Architecture)



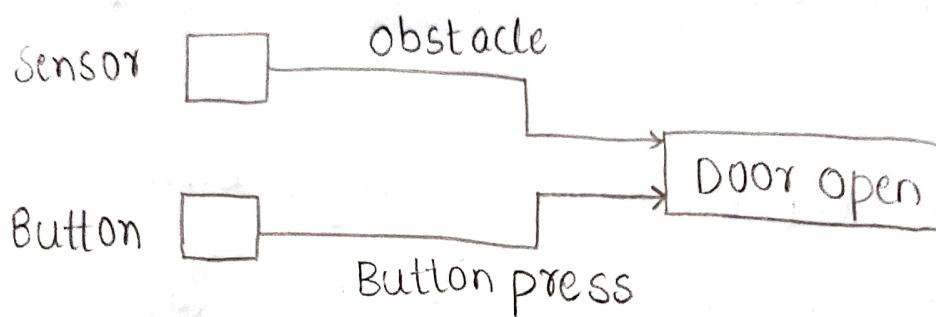
3. Interface Design :

- * How h/w communicates with s/w
- * Interface implies flow of information.



4. Component - Level Design.

- * Transforms structural elements of slm architecture



5 - 3 - 2020

The Design Process.

- * Slw design - an iterative process

- * where req's keep on adding:

1. Slw Quality Guidelines and attributes

There are 3 characteristics.

- i. Design must implement all of the explicit req's contained in the model.
2. Design must be readable, understandable to the coders and support the slw.

3. Design should provide complete provide picture of the s/w.

Quality Guidelines

1. A design should exhibit an architecture that
 - i. has been created using known architecture or patterns.
 - ii. Components that exhibit good design characters
 - iii. implementation of an evolution fashion.
2. Design should be modular logically partitioned
3. Design contains distinct representation of data, architecture, interface and components.
4. Design should lead to proper data structure
5. Design should lead to components that exhibit independent functionalities
6. Design should lead to reduce the complexity of connection betⁿ components
7. Design should drive using a repeated method that is driven by info.

8. Design should represent notation that clearly gives info. of working.

Quality Attributes :

- * Developed by HP - FURPS
- 1. Functionality - feature and capabilities of a pgm.
- 2. Usability - human factor → consistency and documentation.
- 3. Reliability - measuring
 - mean time to failure
 - recovery time
 - production
- 4. Performance - processing speed, response time, resource consumption, throughput and efficiency.
- 5. Supportability - adaptability, service ability, and ability to extend pgm.

2. Evolution of slw Design :

- * Evolution is a continuous process.
- * Earlier design → concentrated on criteria
 - Top down manner
 - procedural aspects.

- * Newer design → object-oriented approach
 - emphasis on slw. architecture and
 - ~~procedural~~ design patterns

Methods :

Analysis method : unique heuristics of notation

* Common characteristics :

1. mechanism for the translation of the reg's models into design representation
2. notations for representing functional components
3. method for refinement and partitioning.
4. guidelines for quality assessment.

Design Concepts :

- * Concepts have evolved over history.
- * few Q's to answer to find a design.
 1. What criteria can be used to partition slw.?
 2. How function or data structure is separated from conceptual representation?

3. What uniform criteria define technical quality of SW design?

Overview of imp SW design concepts.

1. Abstraction - brief explanation of your design
2. Architecture - procedural or data abstraction.
3. Patterns - proven solution for a problem
4. Separation of concerns - problem sub divided into pieces.
5. Modularity - handling the subdivided pieces and solve independently.
6. Info. Hiding - hide design decisions.
7. Functional independent - direct growth of modules
8. Refinement : top \rightarrow down design
 - refining levels of procedures.
9. Aspects : concerns includes
 - requirements - features
 - use cases - data structures
 - quality of services - patterns etc.
10. Refactoring - design activity suggested for many cycle methods.