Software Engineering

* Software Engineering:

✓ The software is a collection of integrated programs.

✓ Engineering is the application of scientific and practical knowledge to invent, design, build, maintain, and improves frameworks or processes.

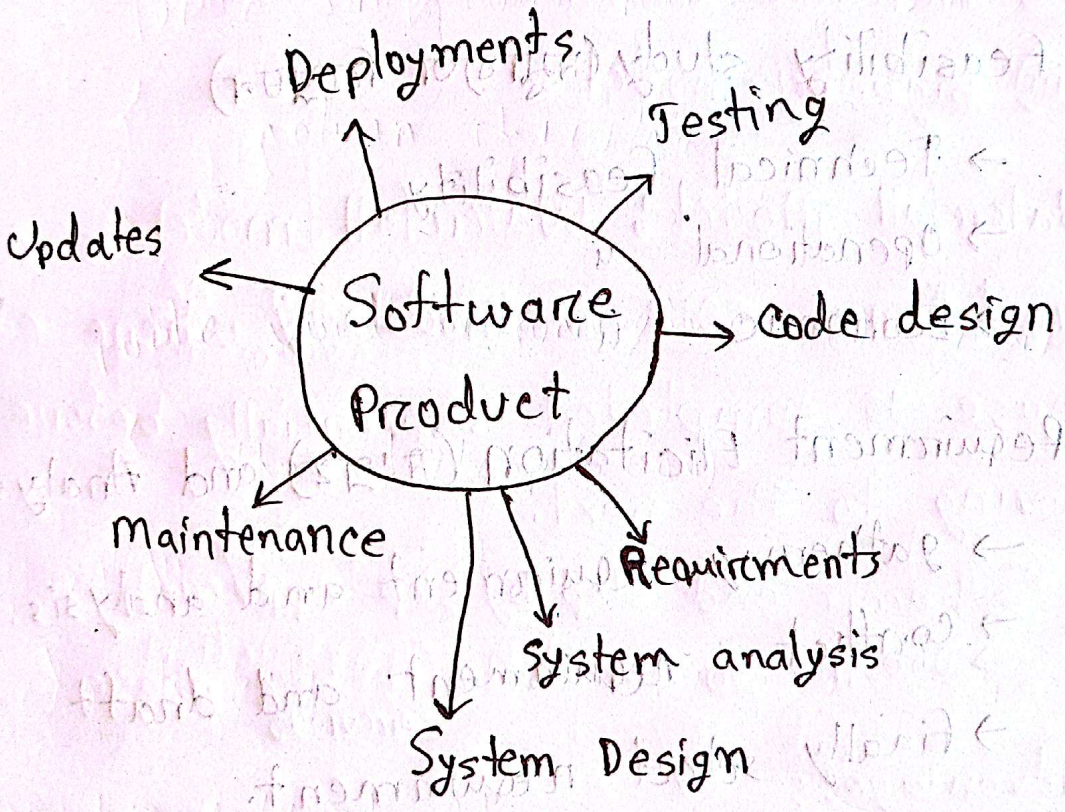


Fig: Primary needed requirments of software engineering.

* Importance of software Engineering:

→ Create relrable software
→ Decrease time
→ Effectiveness
→ Reduce complexity
→ Minimize software cost
→ Handling Big ~~cost~~ project.

* Step of Requirment Engineering:

1. Feasibility study (সম্ভাব্যতা যাচাই)
   → Technical Feasibility
   → Operational "
   → Economic "
2. Requirment Elicitation (তথ্য সংগ্রহ) and Analysis.
   → gathering requirment and analysis.
   → conflict requirment and draft
   → finally select requirment.
3. Software requirment specification:
   → Convert final selected requirment to technical term. using,
   - Data Flow Diagrams
   - ER Diagram

4. Software requirment validation:

→ Connect or incorrect

→ Possible or not possible

→ needed or don't needed

5. Software requirment management:

→ can be change the selected requirment while implementation, maintain on updates.

* Software Development Life cycle:

i) Plannig and requirment analysis

ii) Defining Requirments

iii) Designing software

iv) Developing the project (coding)

v) Testing

vi) Deployment (Pilot programe → massive program)

vii) maintenance

* SDLC Models:

i) Waterfall model ✓
ii) RAD model ✓
iii) Spiral model
iv) V-model
v) Incremental model ✓
vi) Agile model ✓
vii) Iterative model
viii) Bigbang model ✓

* Waterfall Model: Waterfall model is a traditional SDLC methodology where each phase is completed sequentially before moving to the next.

→ linear and step by step approach
→ Doesn't go back to the ~~before~~ previous phase.

Advantages:

→ simple and easy to understand
→ clear documentation
→ small code
→ suitable for small well-know requirments
→ suitable for fixed requirments

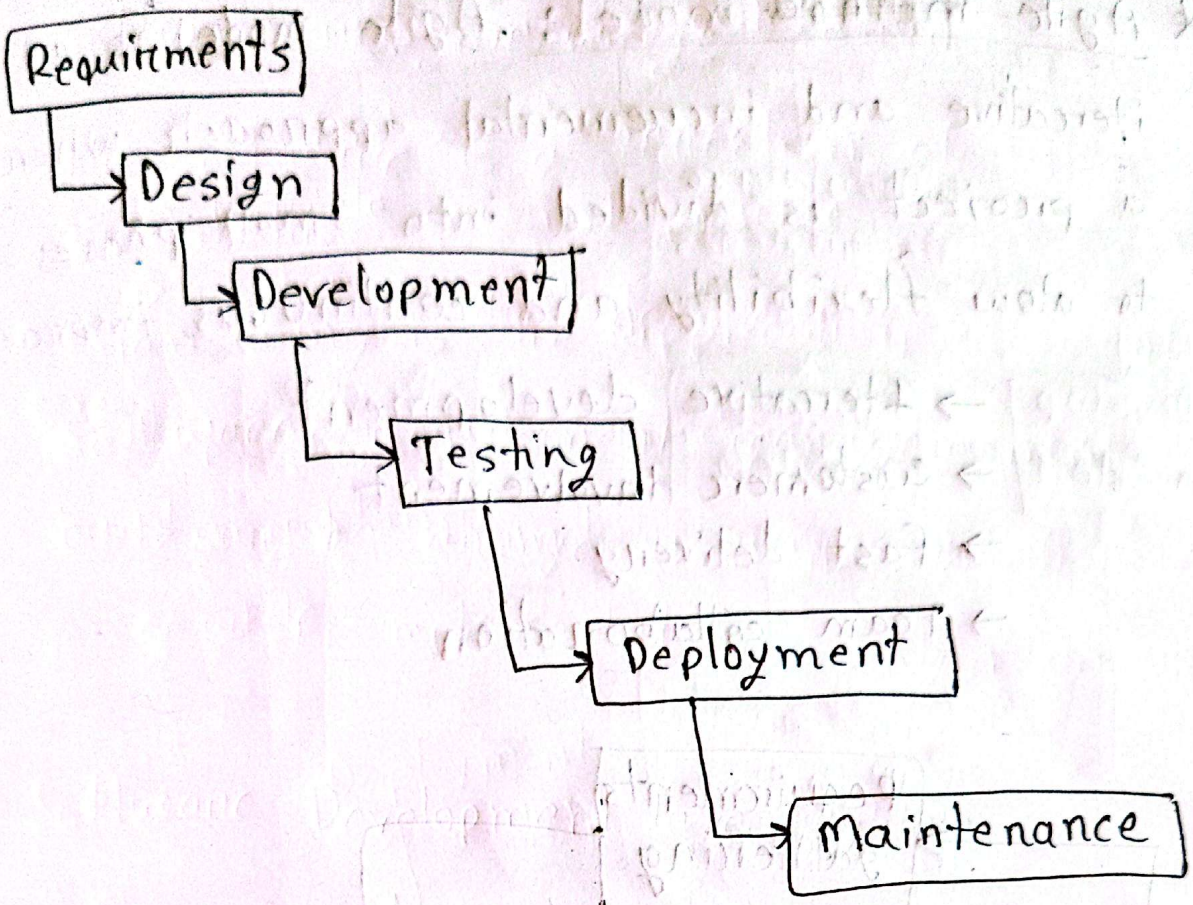


Fig: Waterfall model

Disadvantages:

→ Hard to make change after completing the phase
→ Suitable for small project
→ Hard to maintain for complex project
→ ~~Risky~~ No feedback path
→ No overlapping phase

Example:

→ Banking software
→ Air traffic control system
→ Government database management

* Agile ~~method~~ model: Agile model is an iterative and incremental approach where a project is divided into small phases to alow flexibility and continuous improveme

→ Iterative development
→ Customer Involvement
→ Fast delivery
→ Team collaboration

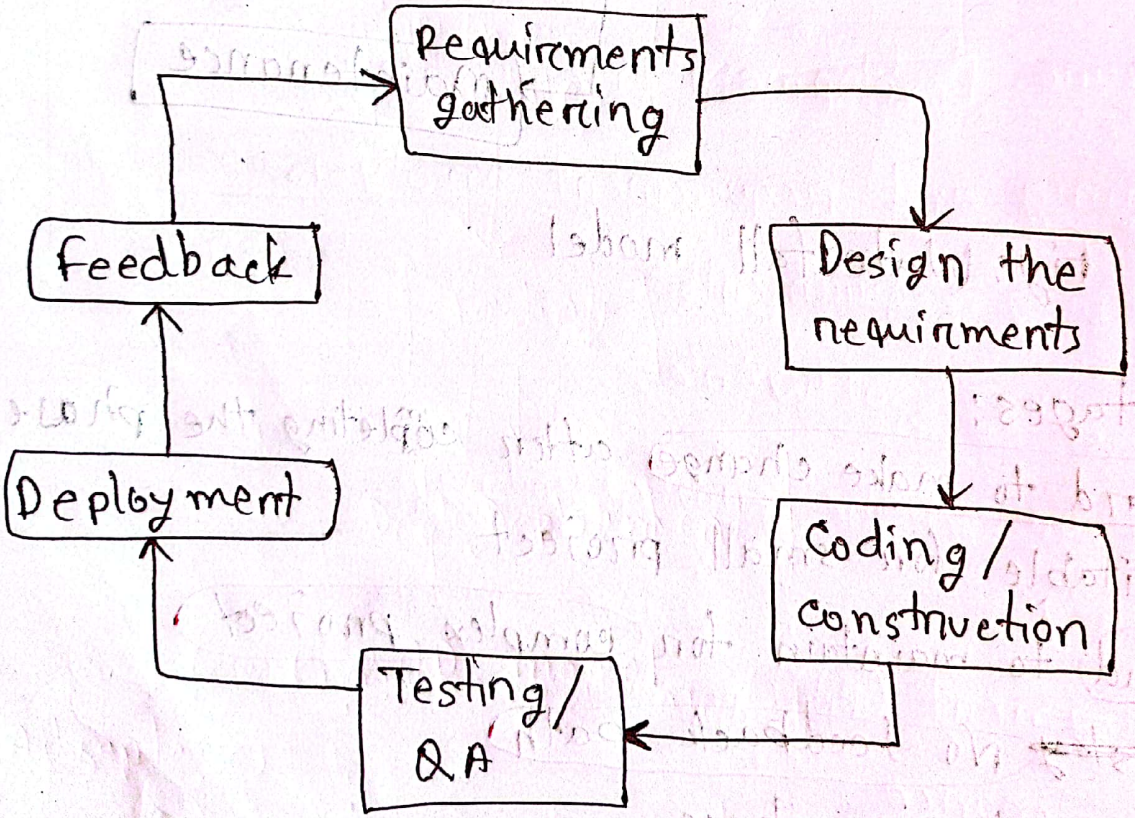


Fig: Agile model

Advantages:

i) Opportunity of requirment changing
ii) Increase quality
iii) CI (continous integration) / CD (continous development)
iv) multiple team involvement.
v) lower risk
vi) fast delivery
vii) Better customer satisfaction
viii) Risk management.

Disadvantages:

i) Requires skilled team member
ii) Endless development (no finishing point)
iii) Higher development time
iv) " " cost

* **RAD model:** RAD stands for Rapid Application Development which is a type of incremental process model that emphasizes (জোর দেয়) on quick development and iterative feedback.

→ Rapid development → parallel development
→ customer involvement
→ fast delivery
→ Divide into different modules

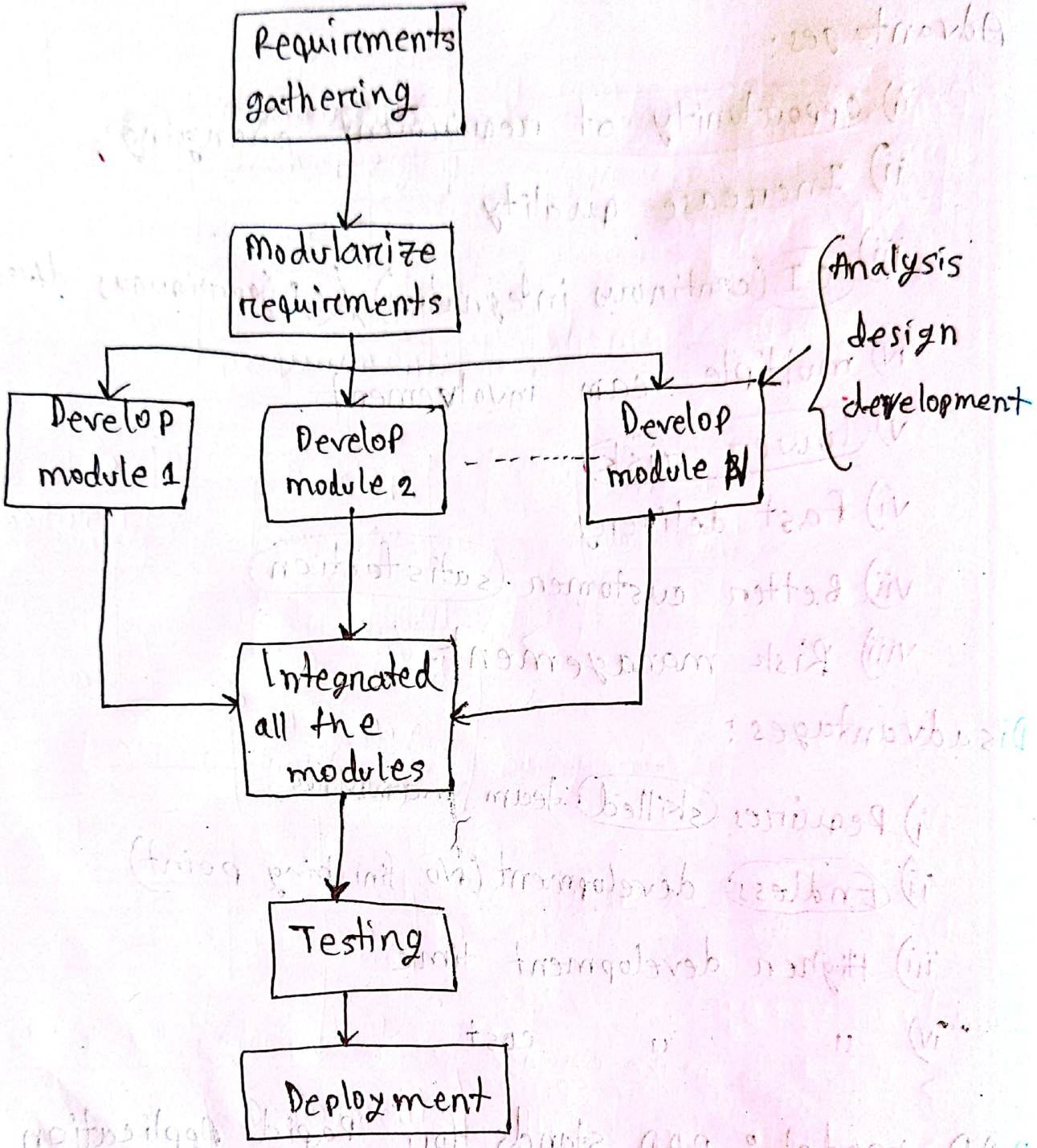


Fig: RAD model

Advantages;

i) Speedy development
ii) lower development time
iii) " " cost

iv) Fast delivery
v) Customer participation
vi) Better customer satisfaction
vii) Increase quality

Disadvantages:
i) Requires skilled team members
ii) " many people
iii) can be created communication gap.

* Incremental model: Incremental model is a software development approach where requirment is divided into different version/module and which is designed, implemented and tested incrementally step by step untill the product is finished.

→ Divided into module/version
→ ~~Develop~~ step by step development
→

~~Advantages:~~

~~i) Speedy development~~
~~ii) lower development time~~
~~iii) " cost~~
~~→ small team~~
~~→ easy to test and debug~~
~~→ more flexible~~

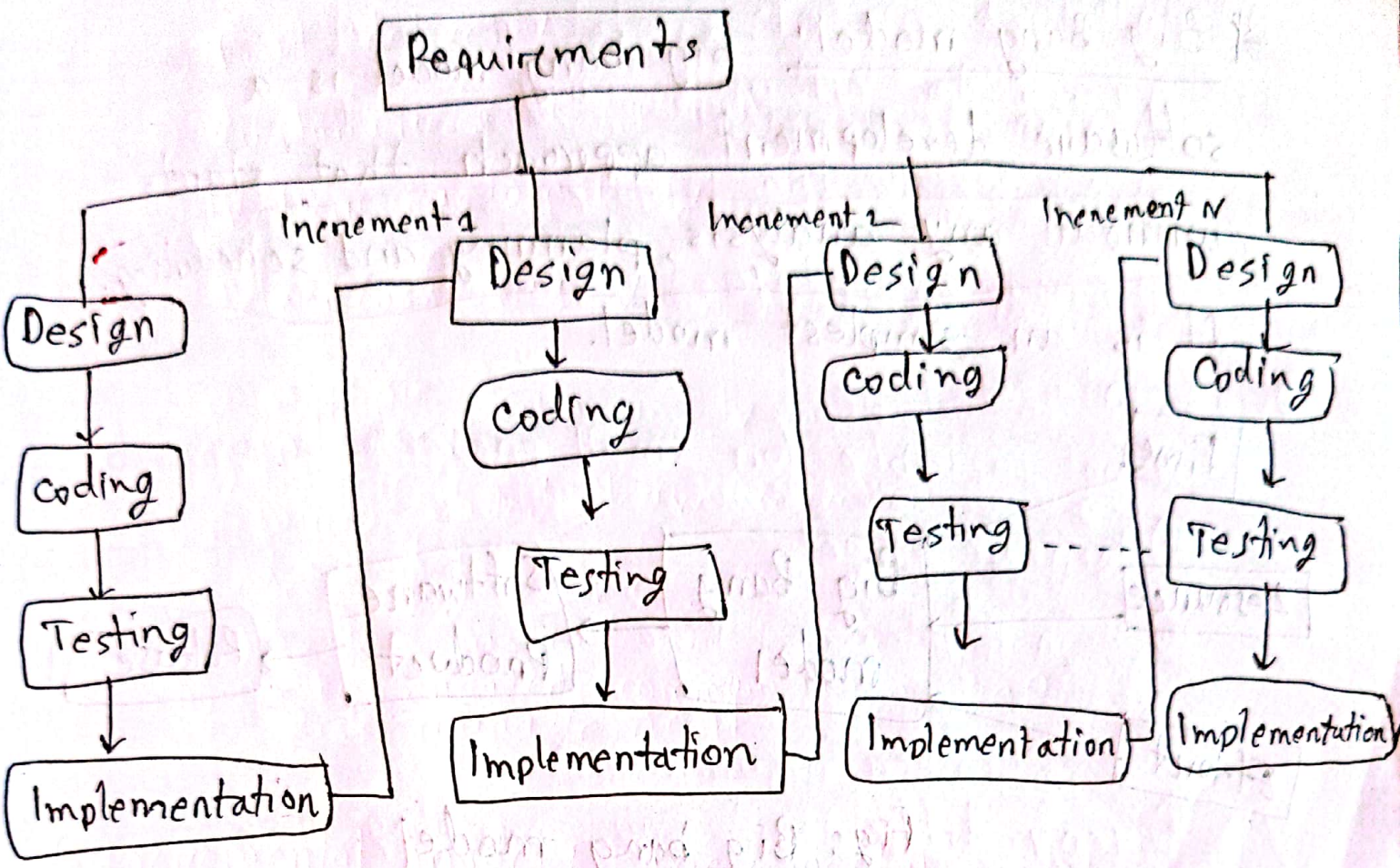


Fig: Incremental model

Advantages:

i) Small team works
ii) easy to test and debug
iii) easy to update
iv) More flexible
v) Give priority of customer

Disadvantages:

i) Higher ~~time~~ development time, cost
ii) "
iii) Need for good planning

*Big Bang model: Big bang model is a software development approach that starts without any analysis, planning and scheduling. It is an simplest model.

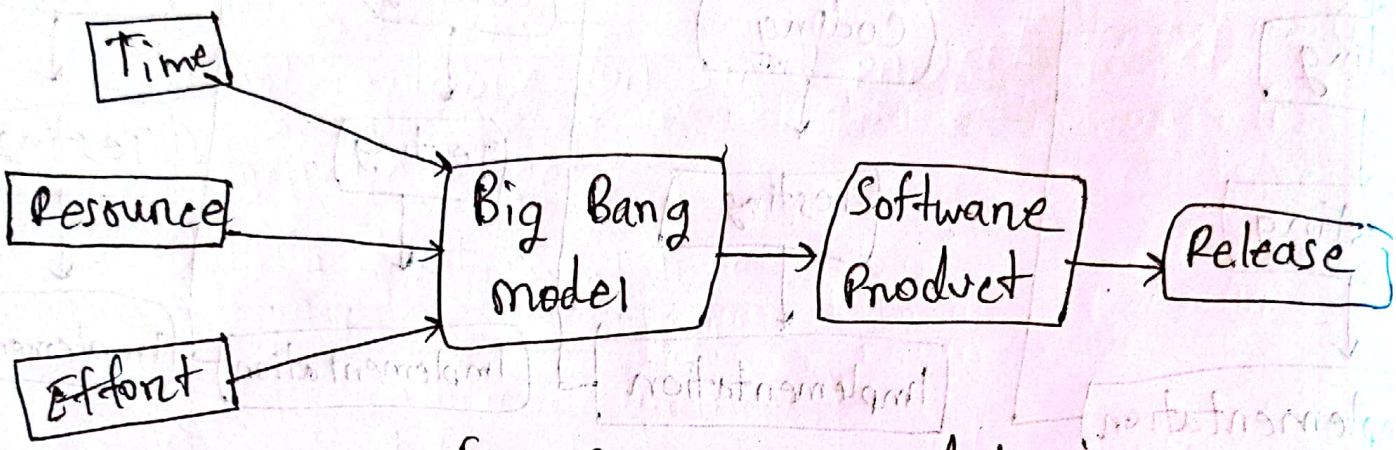


Fig: Big bang model

Advantages:

i) ~~no planning~~ Don't need planning
ii) ~~no an~~ Don't need analysis
iii) " " scheduling

Dis-advantages:

i) High Risk
ii) Difficult maintenance
iii) only used for short project
iv) " " low budget
v) Lack of flexibility

* Iterative model: In iterative model we start developing the software with some requirements and when it is developed then it is reviewed. If there are requirments for changes it, then we develop a new version of the software, based on the requirments. This process repeated itself untill we get our final project.

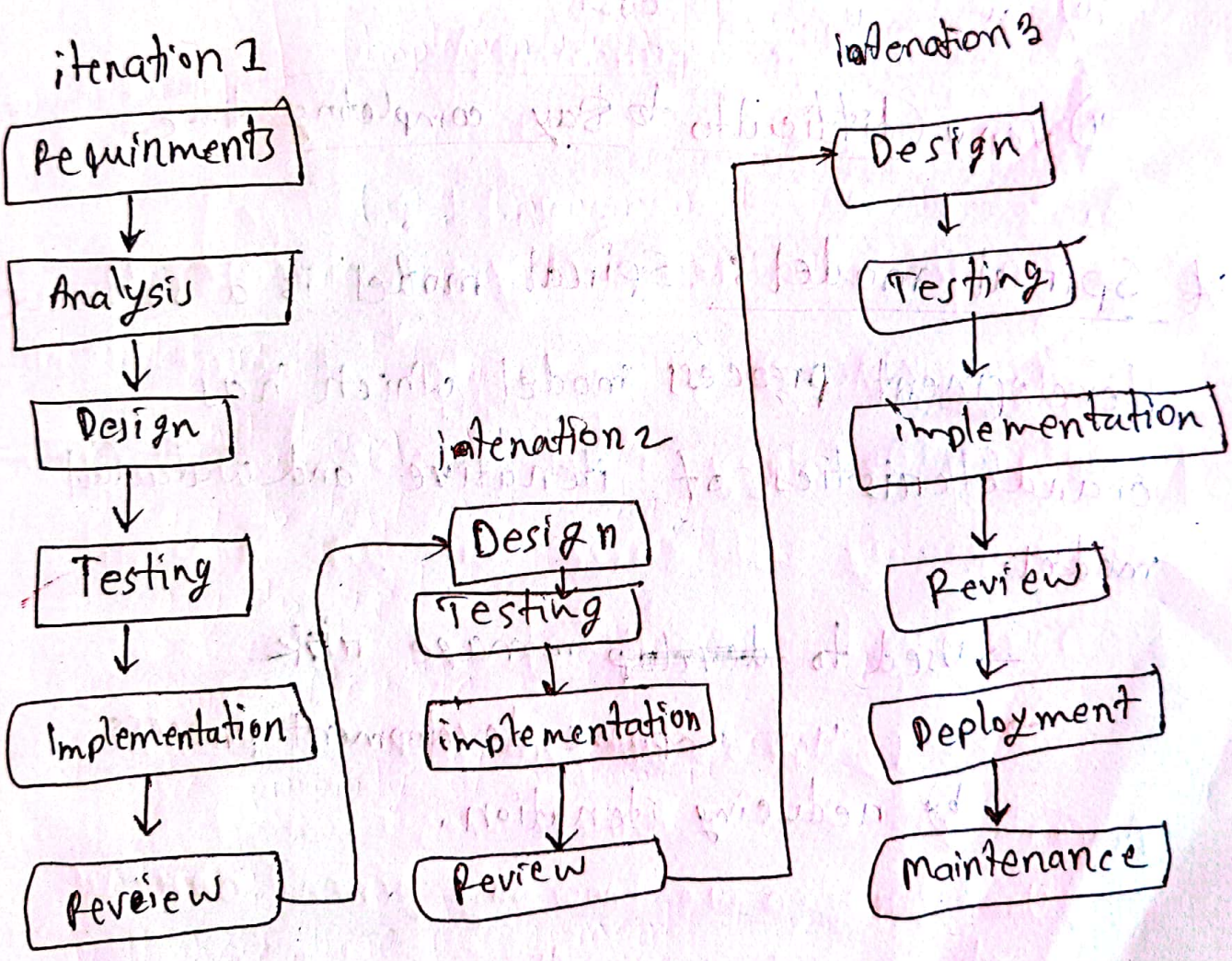


Fig: Iterative model

Advantages:

i) Easy to test and debug
ii) Reliable feedback
iii) Easy to update

Disadvantages:

i) not suitable for small project
ii) Increase budget
iii) High development time
iv) " " cost
v) very difficult to say completing time

* Spiral model: Spiral model is a software development process model which has characteristics of iterative and waterfall model.

→ Used to ~~develop~~ manage risk
→ " " improve development process by reducing iteration.

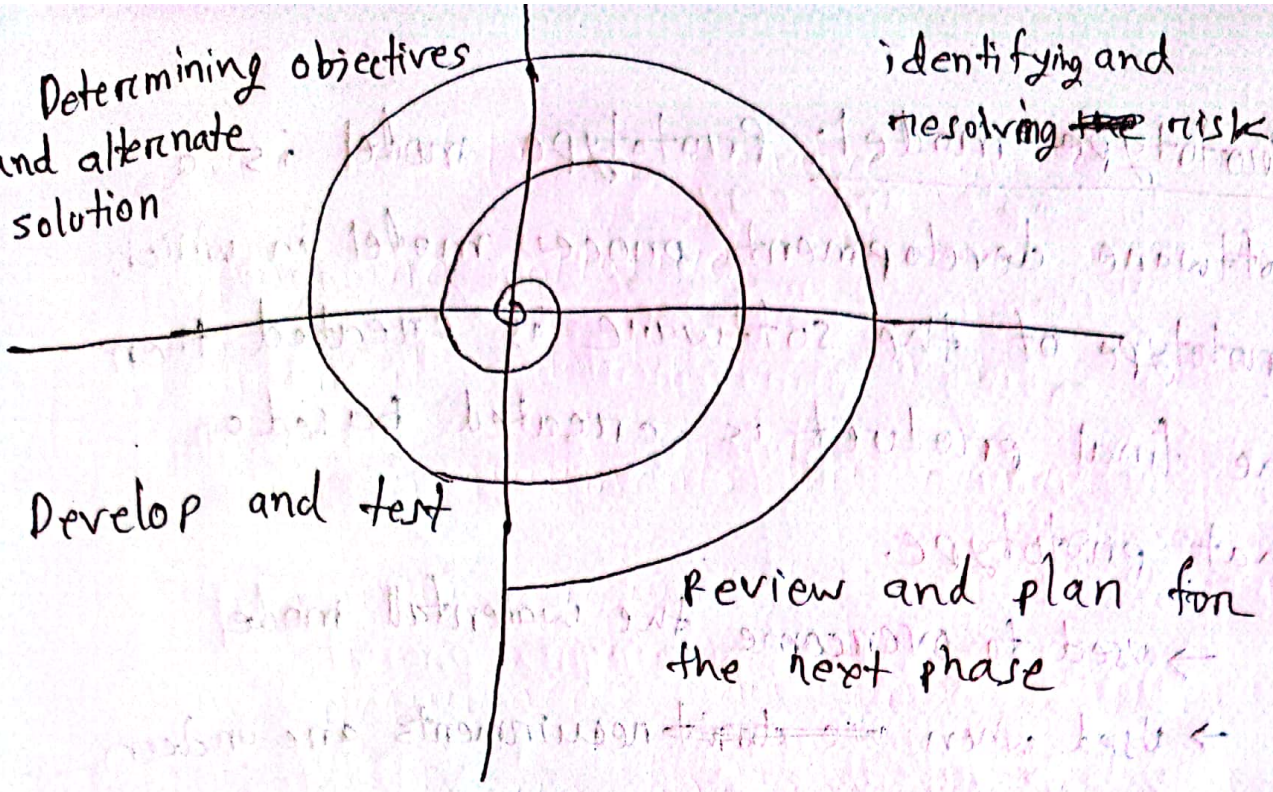


Fig: Spinal model

Advantages:

i) Suitable for large and complex project
ii) Higher customer satisfaction
iii) Easy to estimate the cost
iv) Risk management High

Disadvantages:

i) most complex model
ii) not suitable for small project
iii) High cost
iv) requires more documentation
v) A experience teammate

* Prototype model: Prototype model is a software development process model in which prototype of the software is created then the final product is created based on that prototype.

→ Used to overcome the waterfall model

→ Used when ~~we don't~~ requirments are unclear.

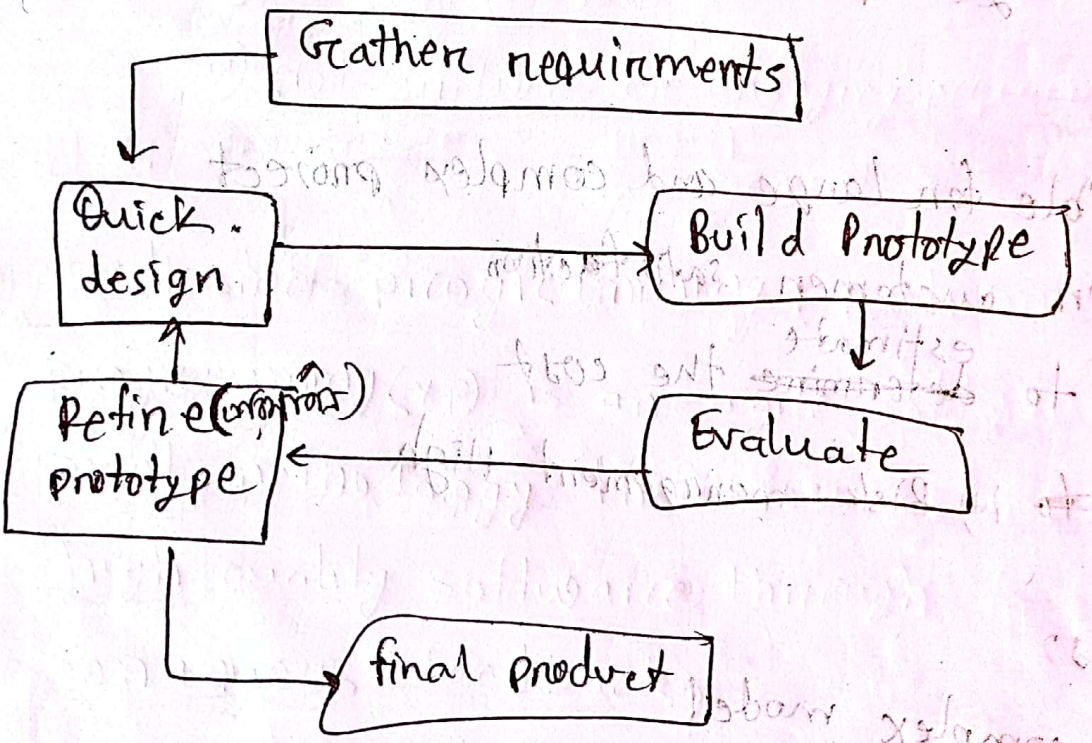


fig: prototype model

Advantages:

i) Used ~~for~~ when we know only general objective and don't know anything in details

ii) Used when developer is not very sure the capability

of an algorithms.

Disadvantages:

i) Not suitable for large project.

ii) Time consuming

iii) Increased cost

*V-model: V-model is a software development process model which is also called verification and validation model. In V-model, the execution of each process is sequential that is, a new phase starts after ending the previous phase.

→ Verification will be on one side

→ Validation " " " other side

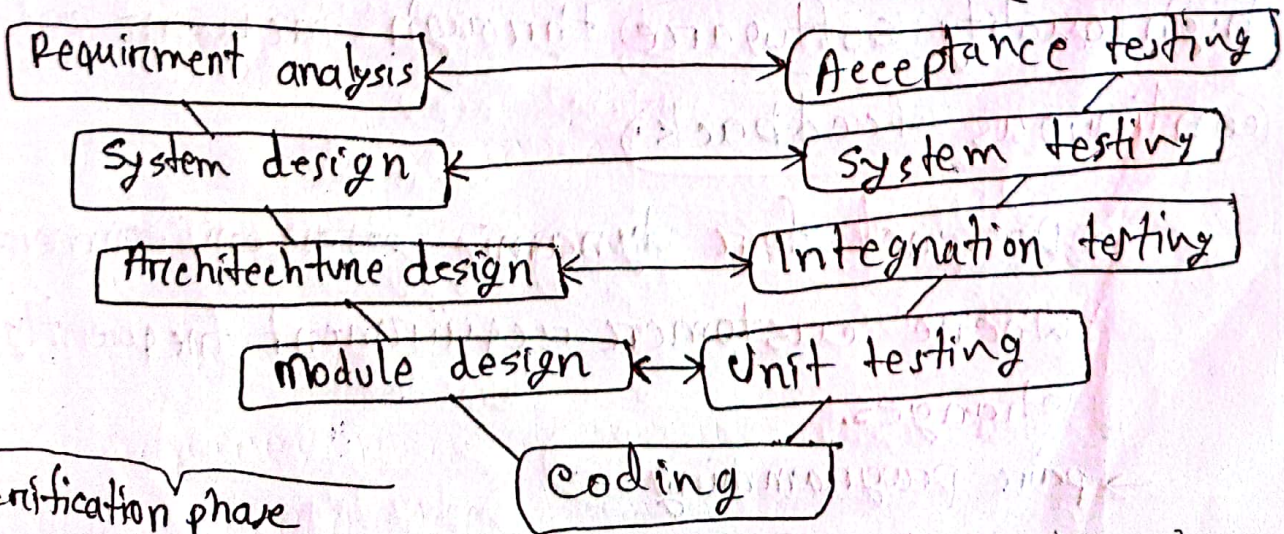


Fig: V-model

Advantages:

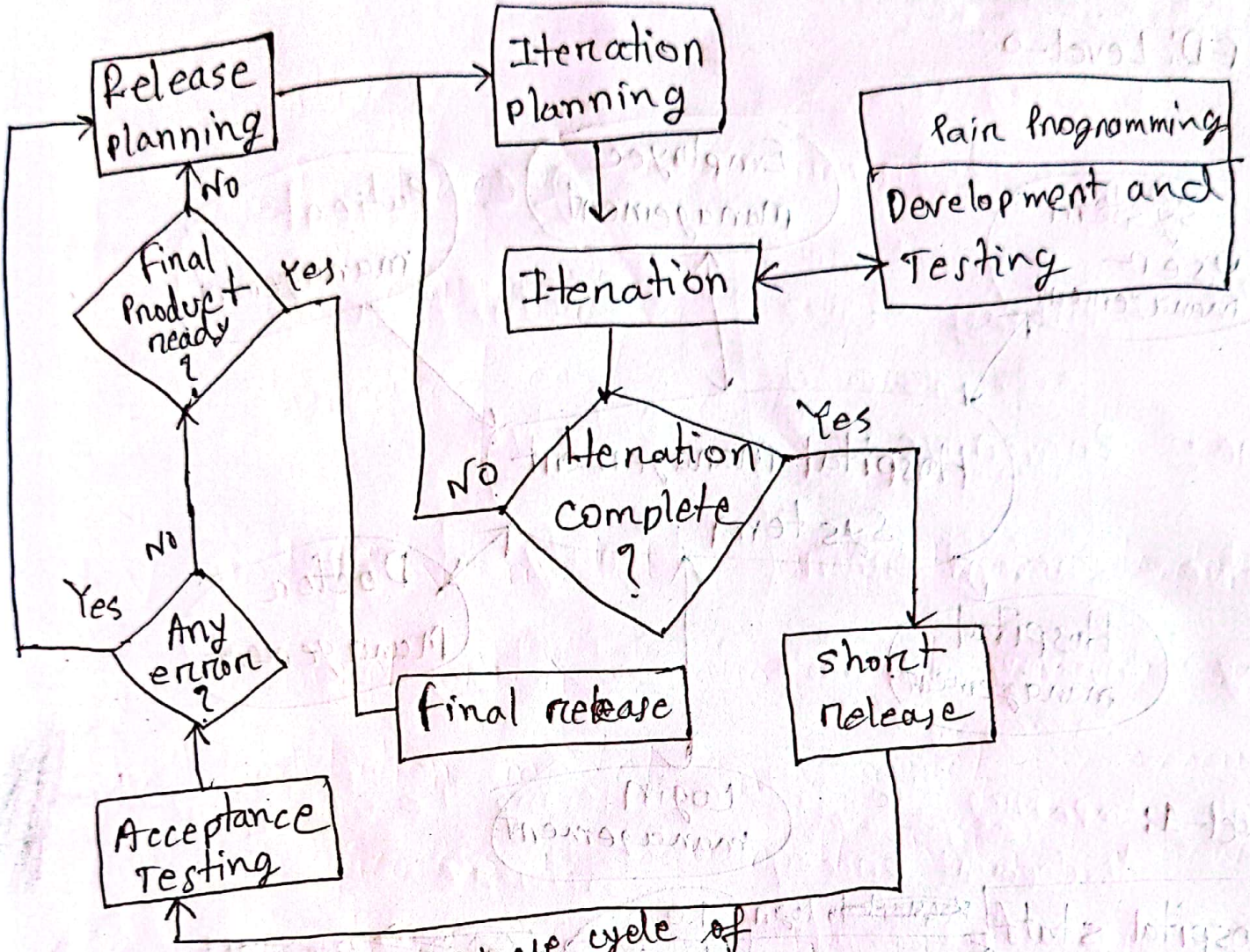
i) Simple and easy
ii) Analysis, designing, testing done before coding
iii) Very discipline model

Dis advantages:

i) Not suitable for complex project
ii) High risk
iii) Not suitable for unclear requirments
iv) Not suitable for changing requirments

* Extreme programming: Extreme programming (XP) is an agile software development methodology that focuses on delivering high quality software through frequent and continuous feedback.

→ Designed for dynamic environment where customer requirment frequently changes.
→ pair programming
→ fast iterations



release cycle of

fig:- Extreme programming

Advantages:

i) Continuous feedback

ii) Pair programming

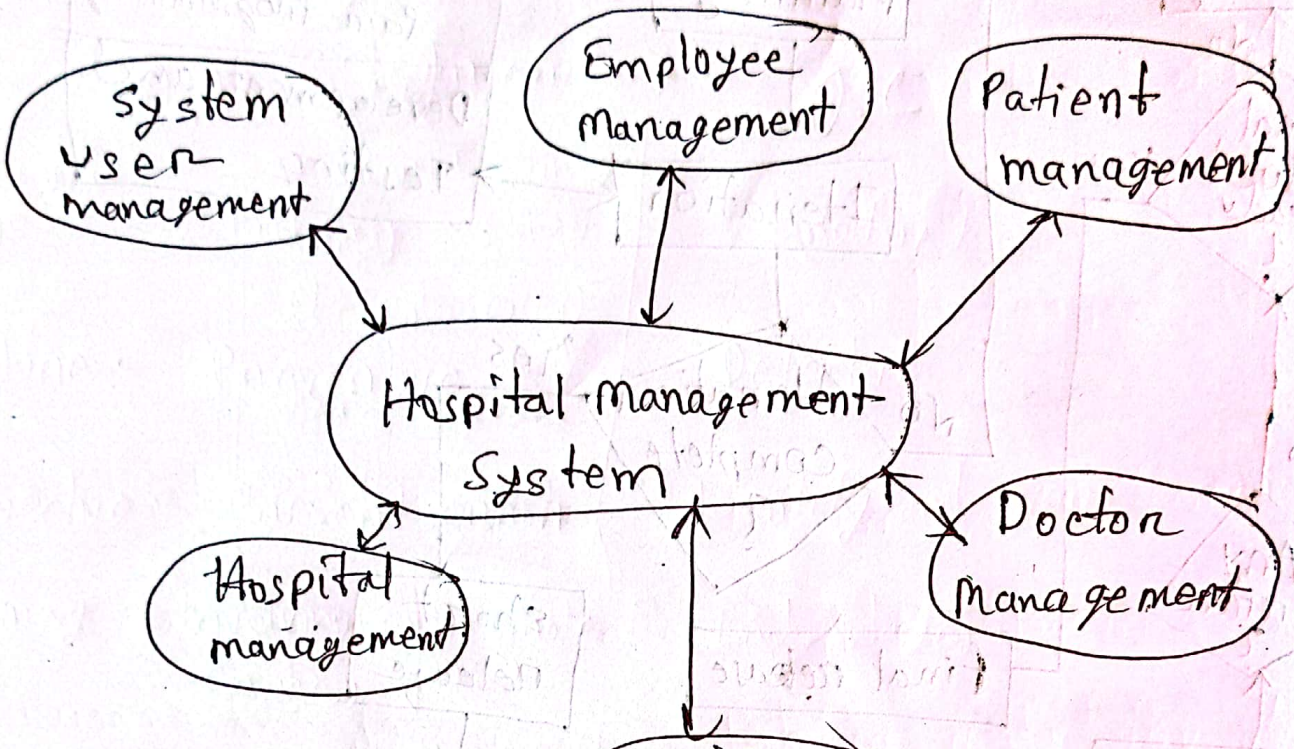
iii) Fast iteration

iv) Customer collaboration very high

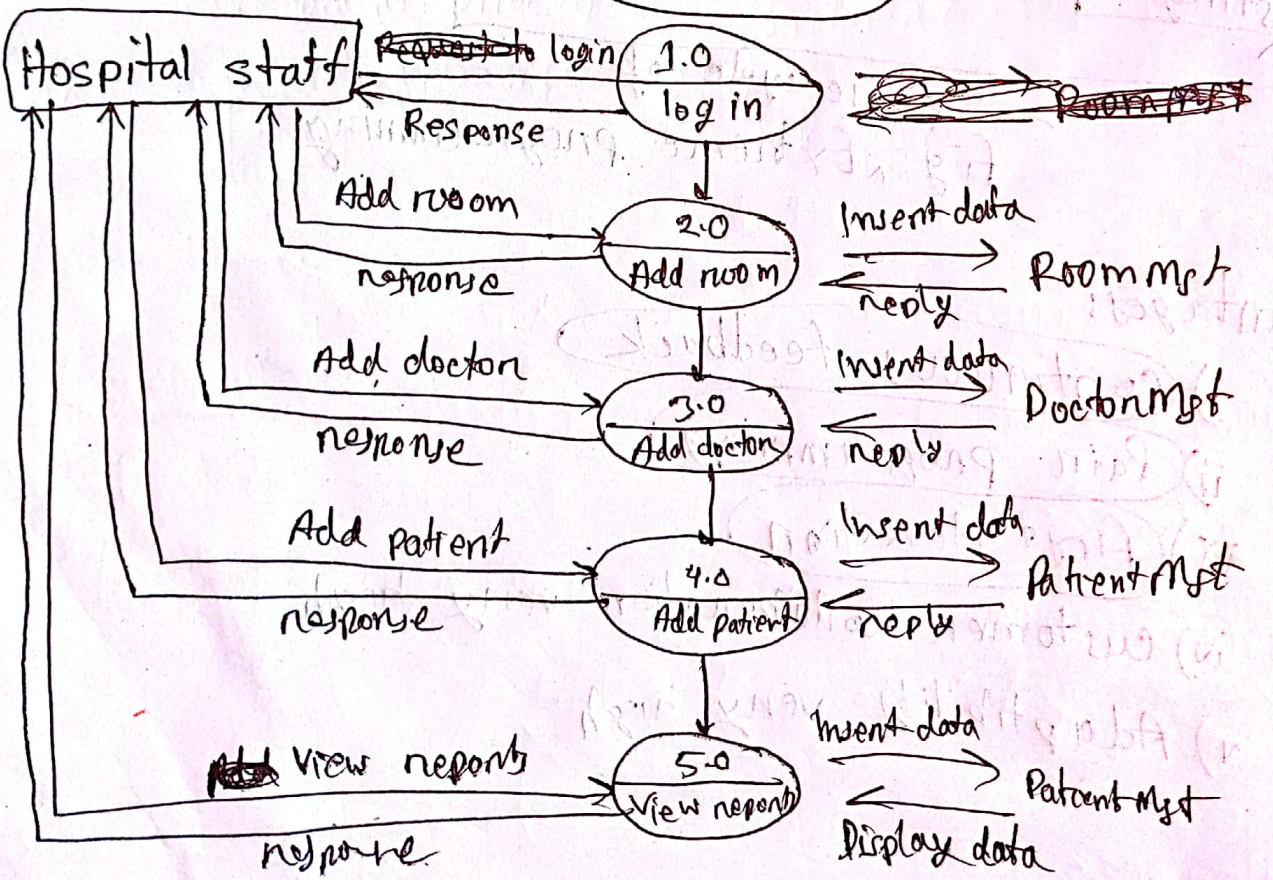
v) Adaptibility very high

Question-15:

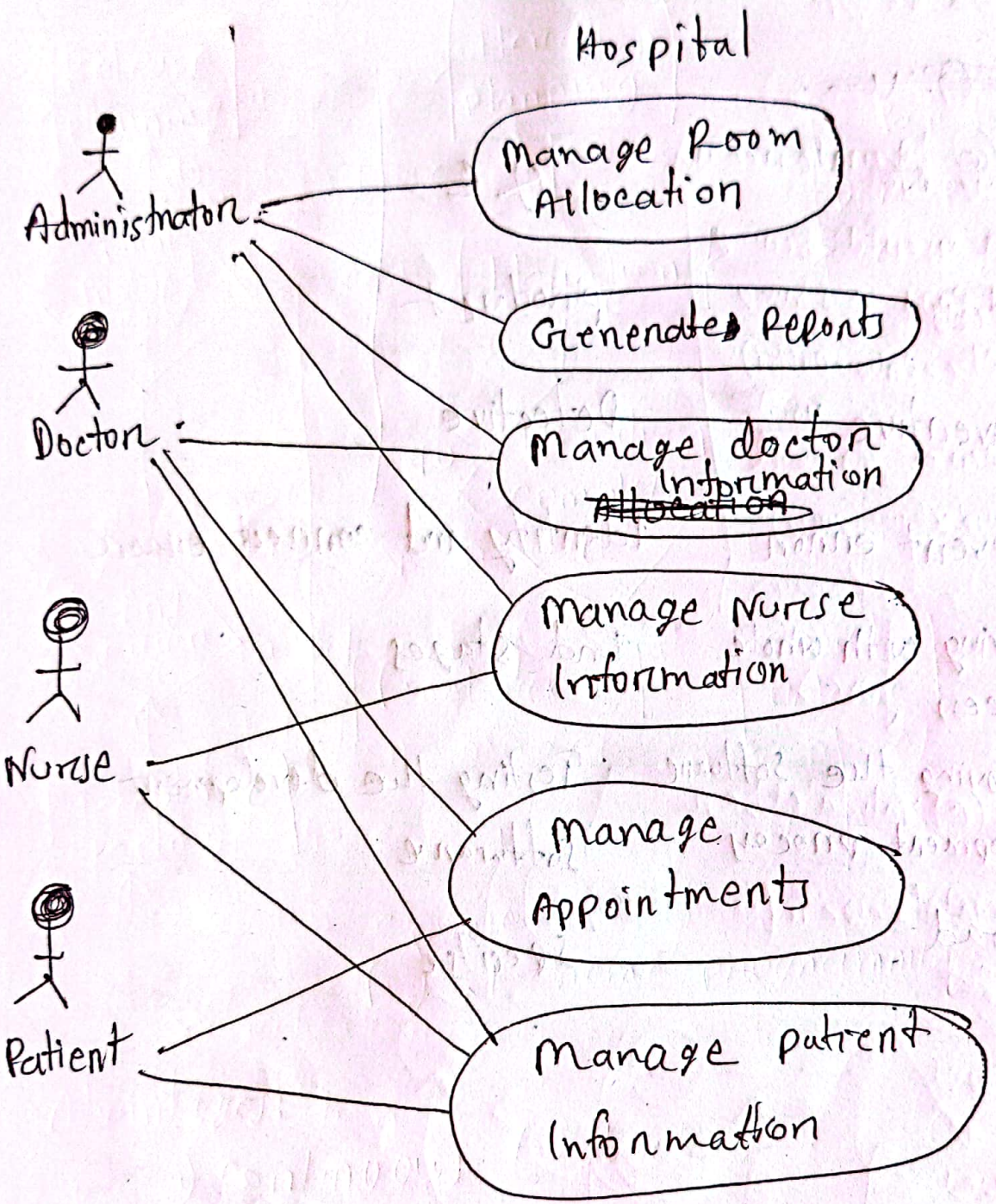
DFD: Level-0:



Level-1:



Use case diagram:



	QA	QC
Focus:	Process	product
Nature:	Preventive	Detective
objective:	Prevent error	Identify and correct error
timing:	Involving with whole Process	End stages
Example:	Improving the software development process	Testing the development software
Scope:	broad	specific