Software Engineering

IAT-I Solutions

1. What are the essential attributes of good software, and also explain software engineering ethics? 5M

A:- **Attributes of Good software:-**

Maintainability :- Software should be written in such a way that it may evolve to meet the changing needs of customer.

Dependability :- Software dependability has range of characteristics including reliability, security and safety.

→ Dependable software should not cause physical (or) economical damage in the event of system failure.

Efficiency :- Software should not make wasteful use of system resources such as memory & processor cycles.

→ Efficiency therefore includes responsiveness, processing time, memory utilization.

Usability :- Software must be usable, without undue effort, by the type of user for whom it is designed.

→ It should have appropriate user interface and adequate documentation.

Professional and ethical responsibility :- 5M

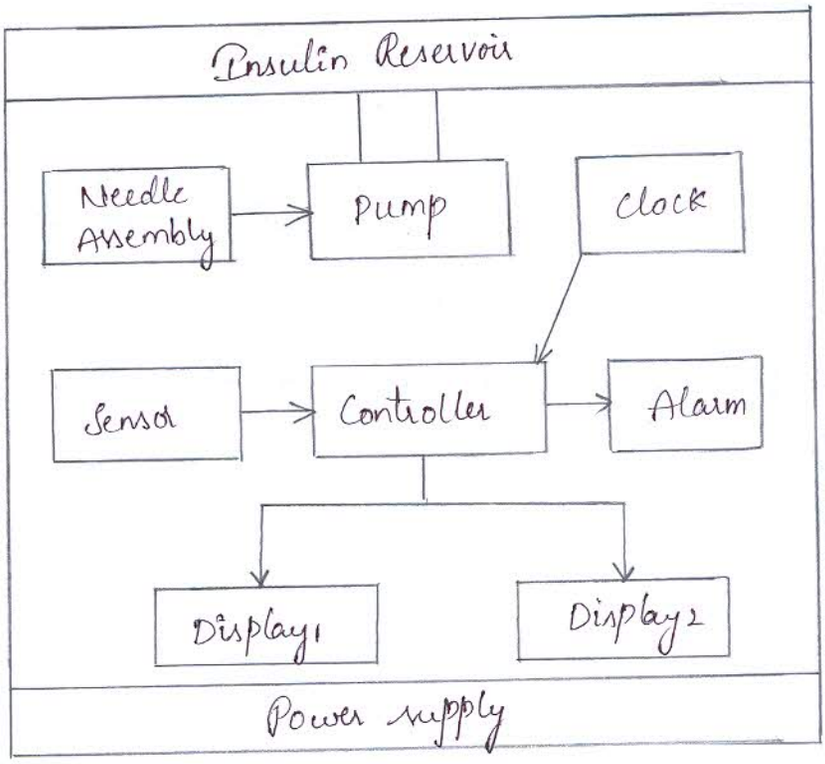
a) Confidentiality :- you should normally respect the confidentiality of your employees or clients irrespective of whether a formal confidentiality agument has been signed.

b) Competence :- you should not mispresent your level of competence. you should not knowingly accept work that is outside your competence

c) Intellectual property rights :- you should be aware of local laws governing the use of intellectual property such as patents & Copyright.
→ you should be careful to ensure that the intellectual property of employees and clients is protected.

d) Computer Misuse :- you should not use your technical skills to misuse other peoples computer.

2. With a neat diagram explain an insulin pump control system. 3M



→ There are many types of critical computer-based systems, ranging from control systems for devices and machinery to information and e-commerce systems.

Eg:- taken here is a medical system that simulates the operation of the pancreas (Internal Organ), this example is chosen because we all have some understanding of medical problems and it is clear why safety and reliability are so important for this type of system. System chosen is Intended to help people who suffer from diabetes

Problem:- Diabetes is a relatively common condition where the human pancreas is unable to produce sufficient quantities of hormone called insulin.

→ Insulin metabolises glucose in blood. The conventional treatment of diabetes involves regular injections of genetically engineered insulin. Diabetics measure their blood sugar levels using an external meter and then calculate the dose of insulin that they should inject.

→ The problem with this treatment is that the level of insulin in the blood does not just depend on the blood glucose level but is a function of the time when the insulin injection was taken. This can lead to very low levels of blood glucose or very high levels of blood sugar.

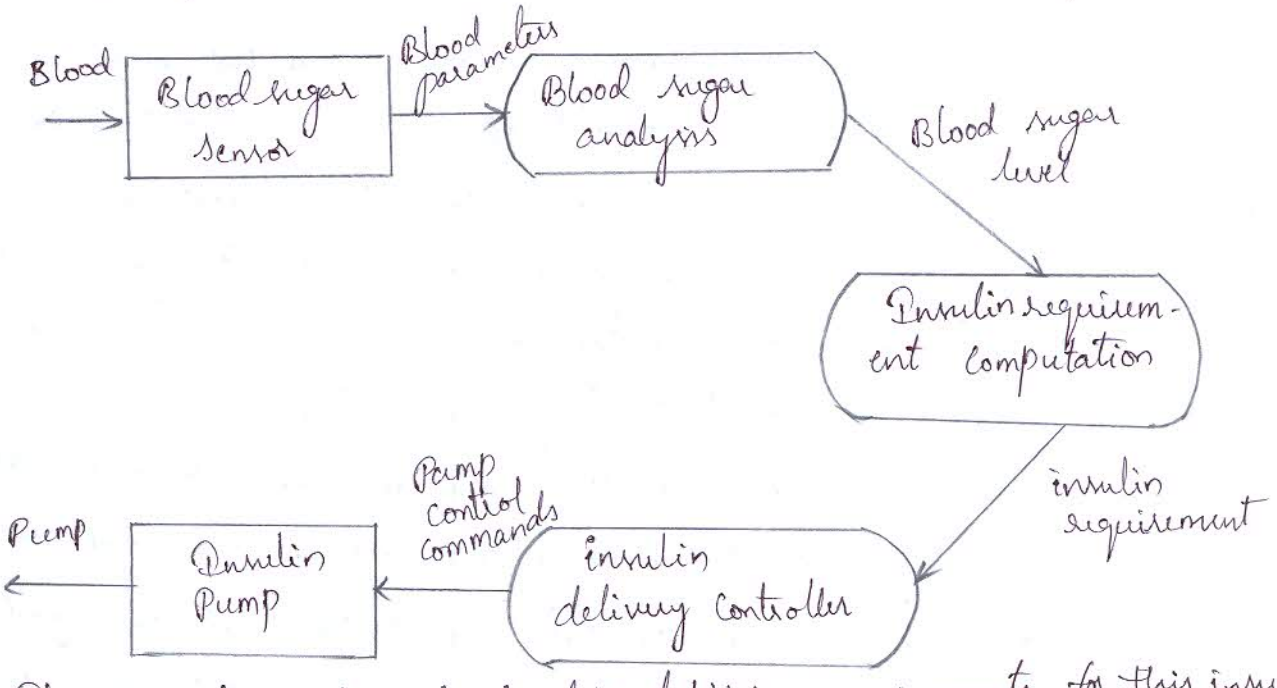
→ Low blood sugar is in short term, a more serious condition, as it can result in temporary brain malfunctioning and ultimately, unconsciousness and death in long term, Continual high levels of blood sugar can lead to eye-damage, Kidney damage & heart problems.

System:-
Current advances in developing miniaturised sensors have meant that it is now possible to develop automated insulin delivery systems. These systems monitor blood sugar levels and deliver an appropriate dose of insulin when required.

→ A software-controlled insulin delivery system might work by using a micro-sensor embedded in the patient to measure some blood parameter that is proportional to the sugar level. This is then sent to the pump controller.

→ This controller computes the sugar level and the amount of insulin that is needed. It then sends the signals to a miniaturised pump to deliver the insulin via a permanently attached needle.

Data flow model of the insulin pump. 1M



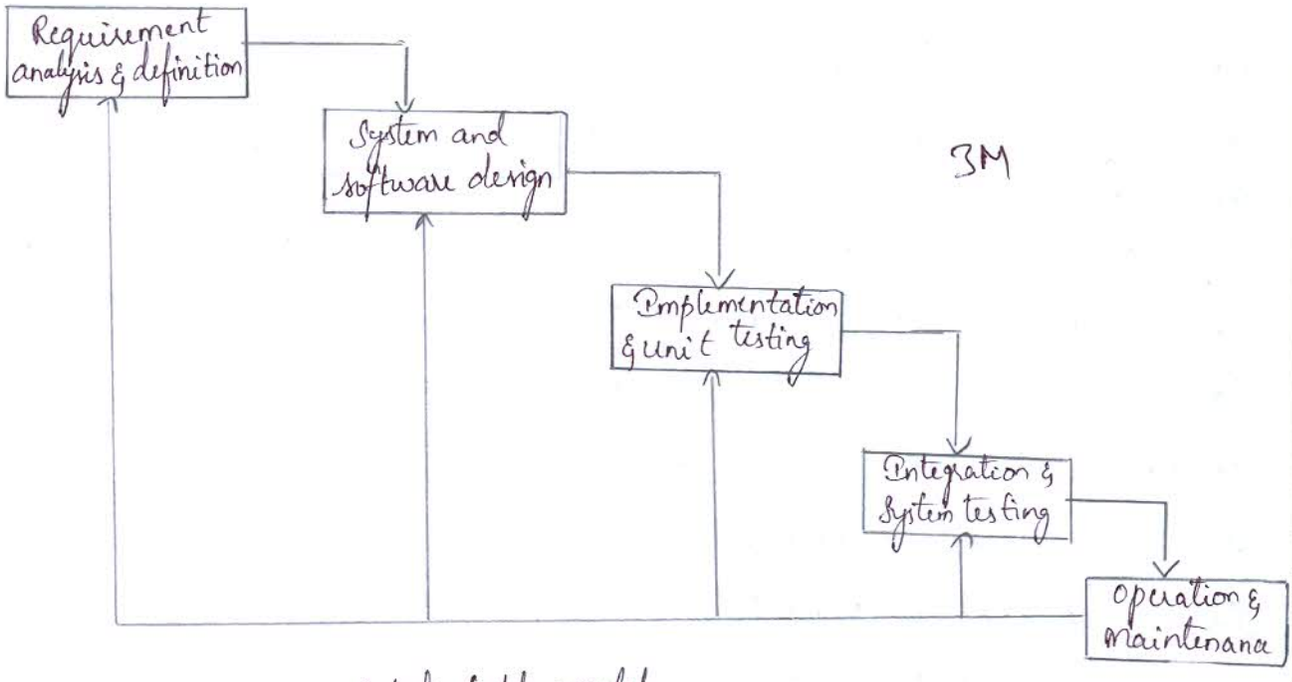
There are two high-level dependability requirements for this insulin Pump systems.

1) The system shall be available to deliver insulin when required.

2) The system shall perform reliably and deliver the correct amount of insulin to counteract the current level of blood sugar.

3. With a neat diagram explain the waterfall model. Explain its advantages and disadvantages.

A:→ This takes fundamental process activities of specification, development, validation and evolution & represents them as seperate process phases.



Waterfall model

→ Because of the cascade from one phase to another, this model is known as 'waterfall model (or) software life cycle.

→ It is an example of a plan-driven process in principle you must plan & schedule all of the process activities before starting work on them.

→ Principal stages of the waterfall model directly reflect the fundamental development activities.

1. Requirement analysis and definition:- The system's services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

2. System and software design:- The system process allocates the requirements to either hardware (or) software systems by establishing an overall system architechture. Software design involves identifying & describing

the fundamental software system abstractions & their relationships.

3. Implementation and unit testing :- During this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

4. Integration and system testing :- The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

5. Operation and maintenance :- Normally, this is the longest life cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were discovered in earlier stages of their life cycle. improving the implementation of system units and enhancing the system's services as new requirements are discovered.

Strengths of waterfall Model :-

1) Easy to understand, easy to use
2) Clear project objectives.
3) Provides structures to inexperienced staff.
4) Staff stable project requirements.
5) Program of system is measurable, helps to plan and schedule the project.

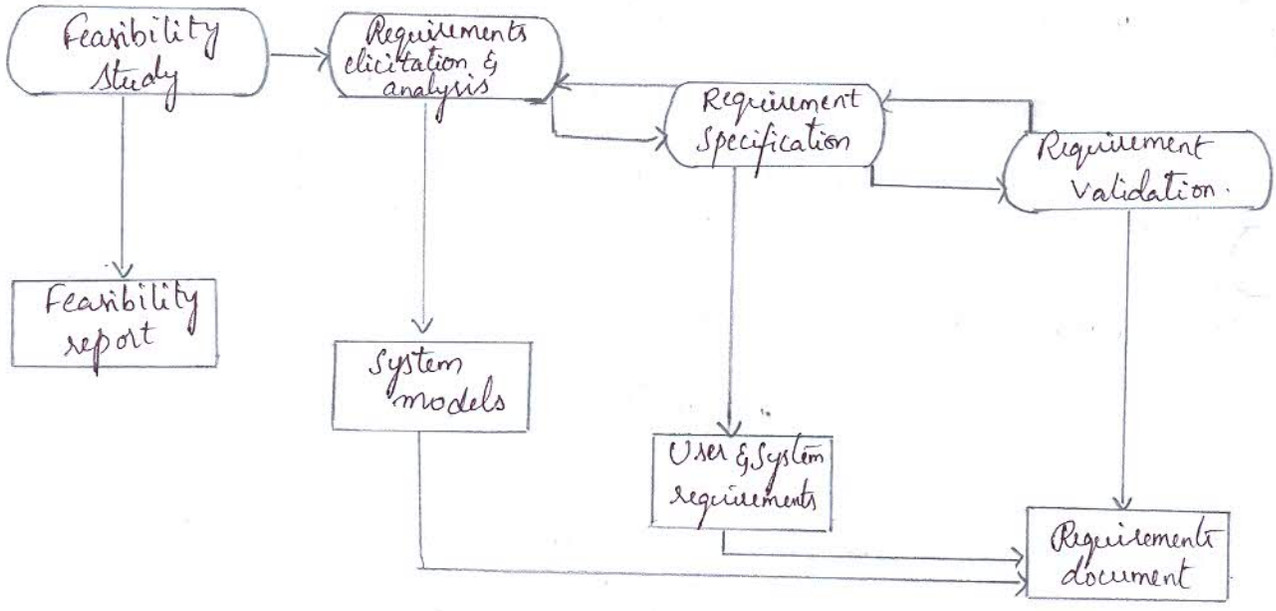
Disadvantages :-

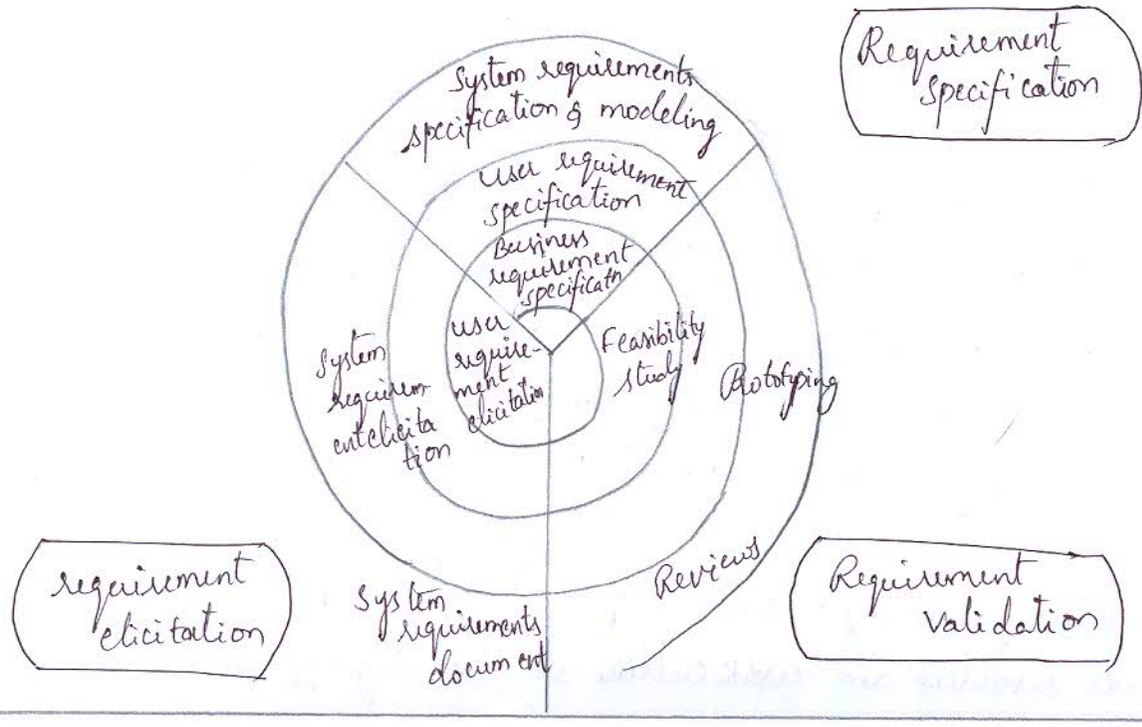
1) waterfall model requires all requirements explicitly, but it is often difficult for the customer to state all requirements explicitly.
2) Time Consuming.
3) Can't go backward.
4) Difficult in iteration.

4. With appropriate block diagram explain briefly the requirement engineering process.

A:- → The goal of the requirement engineering process is to create and maintain a system requirement document.

→ Those are concerned with accessing whether the system is useful to the business discovering requirements, converting these requirements into some standard form and checking that the requirements actually define the system that the customer wants





Feasibility study

→ For all new systems, the requirement engineering process should start with Feasibility study.

→ A feasibility study is a short, focussed study that aims to answer a number of questions

1. Does the system contribute to the overall objective of Organization.
2. Can the system be implemented using current technology and within given cost & schedule.
3. Is the product concept viable?
4. What are the major risk involved in this project?

Carrying out feasibility study involves correctly information & writing report.

1) How would the Organization cope if this system were not implemented
2) what direct contribution will system make to the business objective & requirements.
3) What must be supported by system & what should not be
4) what are the problems with current system and how will the new system overcome?

Feasibility study:-

1) Economic Feasibility:- This part of feasibility study gives the top management the economic justification for new system.
2) Technological Feasibility:- Evaluation of the technical feasibility is the trickiest part of a feasibility study.
3) Legal feasibility - The development of cutain sensitive systems may have some legal ramifications.
4) Organization feasibility - Here developers get the information about the manpower resources and work culture of the Organization.

5. Explain the structure of a requirements document.

A:-

Chapter	Description.
Preface	→ This should define the expected readership of the document and describe its version history, including a rationale for the creation of new version and summary of the change made in each version.
Introduction	→ This should describe the need for the system. it should briefly describe its function and explain how it will work with other systems
Glossary	→ This should define the technical terms used in the document.
User Requirements definition.	→ The services provided for the user and the non-functional system requirements should be described in this section.
System Architechture	→ This chapter should present a highlevel overview of the anticipated system Architechture showing the distribution of functions across system modules.
System requirements specification	→ This should describe the functional & non-functional requirements in more detail.
System models.	→ This should set out one or more system models showing the relationships between the system components and system and its environment.
System evolution	→ This should describe the fundamental assumptions on which the system is based & anticipated changes ~~boosted~~ due to hardware evolution, changing user needs etc.
Appendices.	→ These should provide detailed, specific info related to the application.
Index	→ Several indexes to the document may to included.

6. Explain various requirement validation checks and requirements validation techniques.

A: Requirements validation checks :-

1. Validity check :- A user may think that a system is needed to perform certain functions however further thought and analysis may identify additional or different functions that are required.

2. Consistency check :- Requirements in the document should not conflit i.e., different description of the same system functions need to be checked.

3. Completeness check :- The requirement document should include requirements that define all functions and the constraints intended by the system user.

4. Realism check :- Using knowledge of existing technology the requirement should be checked to ensure that they can actually be implemented.

5. Verifiability :- System requirement should always be written so that they are verifiable this means that you should be able to write a set of tests that can demonstrate that the delivered system meets each specified requirements.

Requirement validation techniques :-

1. Requirement reviews :- The requirements are analysed systematically by a team of reviewers to check for errors and inconsistencies.

2. Prototyping :- In this approach to validation and executable model of the system in question is demonstrated to end users and customers they can experiment with this model to see if it meets their real needs.

3. Test-Case generations :- Requirements should be testable if the tests for the requirements are devised as part of validation process. This often reveals requirement problems.

7. Write a short notes on:
- Requirements management
- Requirements Specification.

A:- **Requirements management:-**

→ The requirements for large software system are always changing. One of the reasons is wicked problems - problems that cannot be completely defined which makes requirements bound to be incomplete.

→ Once a system has been installed and is regularly used, new requirements inevitably emerge as they will discover new needs and priorities.

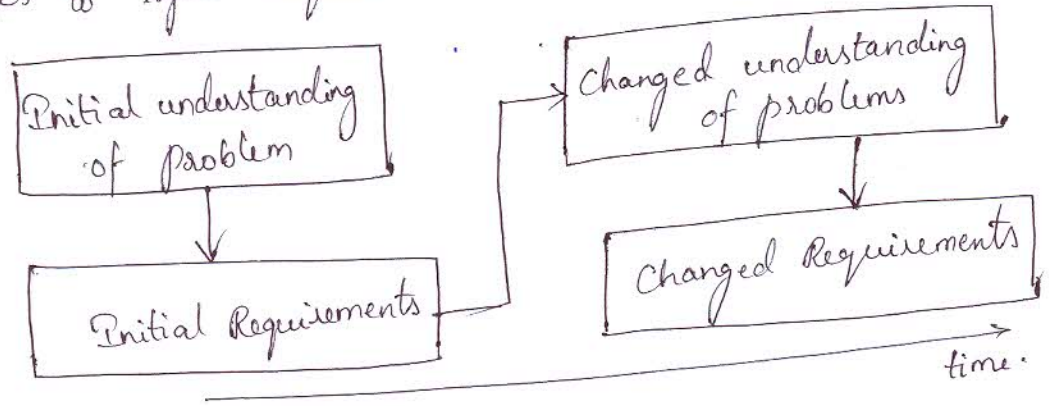
Change is inevitable because:-

i) Business and technical environment of the system may change after installation like new H/w, change in business priorities, new legislation etc

ii) System customers and system users may be different which may impose organizational and budgetary constraints.

iii) As number of users increase, requirements and priorities may be conflicting or contradictory.

Requirements management is the process of understanding and controlling changes to system requirements.



Requirements Specification :-

Requirement specification is the process of writing down the user and system requirements in a requirements document, which ideally should be ~~class~~ clear, unambiguous, easy to understand complete and consistent.

User requirements may use following notations :-

ways of writing a system requirements specification :-

Notation	Description.
1. Natural language sentences	→ Requirement are written using numbered sentences in natural language. Each sentence should express one requirement.
2. Structured natural language	→ Requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
3. Design description language	→ uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of system.
4. Graphical notations	→ Graphical models, supplemented by text annotations. Eg:- UML use case, sequence diagrams.
5. Mathematical specifications	→ Based on mathematical concepts such as finite state machines or sets. Difficult to understand by customers.