

**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
INSTITUTE OF DISTANCE LEARNING
DEPARTMENT OF COMPUTER SCIENCE
END OF FIRST SEMESTER EXAM, DEC. 2014
CSM 495: INTRODUCTION TO SOFTWARE ENGINEERING**

(Attempt all questions) (70 Marks Total) TIME ALLOWED: 2 HOURS

Student ID Number:

(Instruction for students and Invigilators)

This question paper should be returned together with the answer booklet

1. Discuss the three key characteristics of the engineering of web-based software engineering? (6 marks)
2. Explain why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process. (4 marks)
3. Explain why change is inevitable in the development of complex systems? And discuss two ways of dealing with change? (7 marks)
4. Based on your experience with a bank ATM, draw an activity diagram that models the data processing involved when a customer withdraws cash from the machine. (10 marks)
5. Explain why design conflicts might arise when designing an architecture for which both availability and security requirements are the most important non-functional requirements. (8 marks)
6. Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:
 - A system to control anti-lock braking in a car
 - A virtual reality system to support software maintenance
 - A university accounting system that replaces an existing system
 - An interactive travel planning system that helps users plan journeys with the lowest environmental impact(16 marks)

7.

- I. Define what is meant by software process model? Briefly discuss three examples of software process models indicating when each might be used. (7 Marks)
- II. As software engineer consultant, discuss *the type of system*, and explain (with the aid of a suitable diagram), *the stages of the software process model* you would apply to develop the software system for the *case study below*. (12 Marks)

INSULIN DELIVERY SYSTEM

Insulin pump system is a medical system that simulates the operation of the **pancreas** (an internal organ). The software operating this system, collects information from a sensor, and controls a pump that delivers a controlled dose of insulin to a patient.

The system is used by diabetic patients. Diabetes is a relatively common condition where the human pancreas is unable to produce sufficient quantities of a hormone called **insulin**. Insulin metabolises glucose (i.e. sugar) in the blood. The conventional treatment of diabetes usually requires a diabetic patient to regularly inject genetically engineered insulin into their body. Thus, diabetics measure their blood sugar levels using an external meter and then calculate the dose of insulin to inject.

The problem with this treatment is that, the level of insulin required by a patient does not just depend on their blood glucose level but also on the time of their last insulin injection.

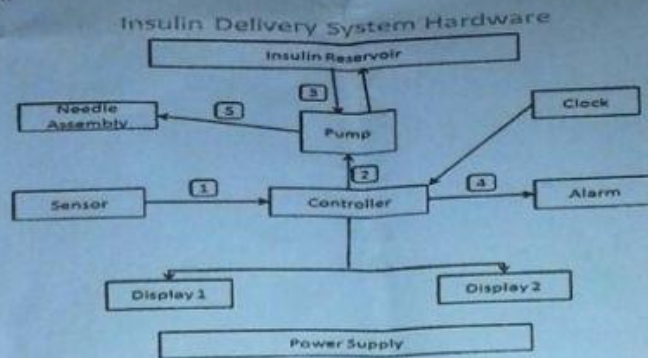
This method of treatment can lead to *very low levels of blood glucose* (if there is too much of insulin in a patient's body), or *very high levels of blood sugar* (if there is too little insulin in a patient's body). *Low blood glucose* is, in the short term, a more serious condition as it can result in temporary brain malfunctioning, unconsciousness, and death. In a long term however, *continual high levels of blood glucose* can lead to eye damage, kidney damage, and heart problems.

Current advances in developing miniaturised sensors have meant that it is now possible to develop a system for *automated insulin delivery* (i.e. **automated insulin delivery systems**).

These systems monitor a diabetes patient's blood sugar levels and deliver an appropriate dose of insulin when required. *Insulin delivery systems like this already exist at hospitals for the treatment of diabetes patients.* It may however be very useful to many diabetic patients to have such system in a device that is permanently attached to their body.

A software-controlled insulin delivery system might be developed by

- 1) Using a micro-sensor attached to the patient to measure some blood parameters that is proportional to the sugar level.
 - 2) This data is then sent to a controller that controls the insulin pump
 - 3) The controller computes the sugar level and the amount of insulin that is needed
 - 4) The controller alert the patient of treatment via an alarm
 - 5) The controller then sends signals to a miniaturised pump to deliver the insulin to the patient via a permanent needle attached to their body.
- The figure below shows the hardware component and organisation of the insulin delivery system.

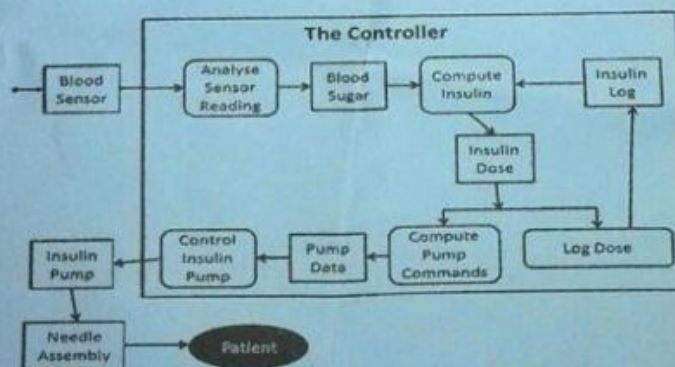


How it works:

- 1) The blood sensor measures the electrical conductivity of the blood under different conditions. The value(s) obtained are then related to the patient's blood sugar level.
- 2) The insulin pump delivers one unit of insulin in response to a single pulse from the controller. Thus, to deliver 10 units of insulin, the controller sends 10 pulses to the pump.

The figure below is a Unified Modelling Language (UML) activity model that illustrates how the software transforms an input blood sugar levels to a sequence of commands that drives the insulin pump.

Activity model of the insulin delivery system



1.

a) Why are iterations usually limited when the waterfall model is used?
(5 Marks)

b) Outline and explain 5 different types of software application. Give an example software application system of each type
(10 Marks)

c) Distinguish between the waterfall model and incremental model and give an example each of a suitable software application that each might be used to develop
(8 Marks)

d) Explain why design conflicts might arise when designing an architecture for which both availability and security requirements are the most important non-functional requirements.
(7 Marks)

2.

a) Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development each of the following systems:

- A system to control anti-lock braking in a car
- A virtual reality system to support software maintenance
- A university accounting system that replaces an existing system
- An interactive travel planning system that helps users plan journeys with the lowest environmental impact

(2 marks each)

b) Briefly explain the principal stages of the requirements engineering process with a suitable diagram?
(12 marks)

3. LUTH is requesting for an automated Hospital Management System to replace its existing manual paper based system. The new system is to control the following information: Patient information, bed availability, consulting and operating room schedules, and patient invoices among others. These services are to be provided in an efficient, cost effective manner with the goal of reducing time and making efficient use of LUTH resources. The new system should be available at all time to LUTH staff on and off site.

- a) Draw an appropriate UML use case diagram for the requested system (5 marks)
- b) Briefly explain the stages of your choice of an appropriate generic software process model that you would use to manage the development of the requested system for LUTH. (5 marks)
- c) State and justify the *type(s) of software system* you would develop for the requested system by LUTH. (5 marks)
- d) Giving reasons for your answer, Identify the requested systems key non-functional attributes (5 marks)