**Question 1**

* **They used a new, unfamiliar and unsuitable technology**. The CIO and project leader had used Java before but it was for a simple e-commerce application, not for a complex medical claim processing system. As most of the software developers on the team were quite unfamiliar with the new Java , they are familiar with Oracle technology, it affected their productivity, slowing down the work. In addition, Java did not have good database connectivity at the time, not suitable, they are aware of this but they choose to ignore them and this seriously affected their project quality (i.e failing to connect to DB). Furthermore, the requirements for medical claim processing logic is much more complicated than e-commerce applications logic.
* Using **JAVA technology that did not have tools to support fast prototyping**. This was a critical error as rapid prototyping is an important component that is required in Rapid Application Development (RAD)/Joint Application Development (JAD). Java technology cannot produce UI prototypes in one day. It took more than one day to produce the UI prototype, around 2 to 3 weeks, thus things were delayed. This resulted in the software developers struggling to produce the GUI and the team was then unable to show the user interfaces to the users and obtain the user requirements and feedback on time. This resulted in the team becoming seriously unmotivated and left the team, only the PM knows JAVA, has to do everything alone.
* **Using the wrong process model which is RAD to build the medical claim processing**, RAD/JAD is suitable for the small scale project such as simple ecommerce applications they developed before. However, for medical claim processing systems whose requirements are much more complicated than ecommerce applications, use spiral models that allow for incremental developments (build it part by part iteratively) and risk management activities can be performed to identify and resolve all the possible risks occurring during the project development.
* **Introducing more formal requirements and sign-offs.** This was introduced too late in the project, where the condition of the project had deteriorated beyond the point of salvaging. In addition, the software developers were still struggling with the new Java and Oracle technology which would have made it difficult for them to produce the work on time.
* **Outsourcing the existing payment subsystem**. However, this became a complicated task that required extra 6 months to manage the arrangement, thus the project was delayed for another 6 months, initially the project was estimated to be completed in 6 months, but ended up taking 12 months.
* **Study the requirements of the medical claim processing systems** first, the processing requirements of the systems are much more complicated than the e-commerce requirements, RAD or JAD is not recommended here, use spiral model is much more appropriate as the spiral model involves risk assessment activities which will try to identify and resolve all the possible risks during the project development. Be aware that e-commerce applications are different from medical claim processing systems that require more studies on the processing logics.
* **Choose the technology that are suitable for the systems requirements**, technology that can provides a scalable database connectivity for the medical claim processing systems, and choose the technology that they are familiar with or provide them the necessary training on the selected Java technology, use of CASE tools are recommended here, especially to produce the UI prototypes faster.
* **Adopt spiral model to build the system part by part iteratively**, each iteration model makes use of the prototype to finalize the user requirements part by part, the spiral model enables us to perform the risk assessments for each iteration also, thus early precautions measures can be taken to avoid any delays on the project.
* **The project manager should review the project plan regularly** so that he is able to discover the problem of the project early and take necessary corrective action to solve the issue. This can help the project manager to avoid the problem occurring before he notices that. Reviewing the project plan can help the project manager to make sure the project is developed on schedule and also avoid overuse of resources.

Syllabus answer

* To **use Oracle Forms and PL/SQL**. All the software developers were familiar with this technology and therefore, this would boost their morale rate/ motivation , productivity and also ensure their ability to deliver the required work products on time. In addition, this technology had tools to support prototyping that is required in the RAD/JAD approach. If they are not familiar, send them for training, and get the senior to coach the junior staff.
* **The project manager should delegate management of the outsourcer to another person**. This is as the same few people were doing all the work as well as managing the outsourcing and dealing with other side issues that came along. This would free the team to focus on the development of the current project.
* **Break the system into modules**, each of which could be managed as an RAD project. The modules would not need formal requirements or heavy documentation, because each would be a real RAD project: small and quick. This would ensure the timely and successful completion of the modules and thus the overall project.

**Question 2**

1. **Software size estimates may be significantly low.** E-commerce applications are smaller and simpler than medical claims systems.

* **Category**: Product size, which refers to the risks associated with the overall size of the software to be built.
* **Impact** : Larger and complex projects make it more difficult to come up with a reliable estimation on the time and resources required to complete the project. This would then translate to a greater probability that schedule delays and budget overruns will occur. In this case the medical claim processing is much more complicated than e-commerce applications , for example the processing logic to calculate the claims vs the processing logic to calculate the total orders.

1. **Staff turnover will be high**

* **Category**: Staff size and experience, which refers to the risks associated with the overall technical and project experience of the software engineers who will do the work.
* **Impact** : If the staff are inexperienced in the particular technology used to develop the system and do not get sufficient training, it will result in lower productivity levels. In addition, the increasing frustration with the project due to their struggles with the technology may lead them to resign. Both of these will cause schedule delays. In this case most of the Java developers stop and quit before the project is completed.

b.

1. **Software size estimates may be significantly low.**

* Adopt a software process model that considers risk management and develops iteratively and incrementally to cope with the size and complexity of the software to be developed. Such models typically perform size estimation at the beginning of each iteration. This is *risk reduction* as precautions are taken to reduce the probability of the risk. I recommended the spiral model here.

1. **Staff turnover will be high**

* Meet with current staff to determine causes of turn over and mitigate those causes where possible before the project starts. Implement measures to ensure continuity when people leave (e.g. work product standards). This is *risk mitigation* because these steps ensure that the impact of the risk is lessened.

c. Risk monitoring

1. **Software size estimate may be significantly low**

* Monitor the error/FP of the completed components or modules as an indication of complexity.
* Monitor the FP/month for the team. This can be used for future estimations.
* Monitor staff views by obtaining regular feedback from the team members about the project.

1. **For staff turnover will be high**, risk monitoring activities could include:

* Monitoring the general attitude of team members based on project pressures.
* Monitoring the job market for software development-related jobs.
* Monitoring the health of the relationships among the team members.

**Question 3**

1. **Performance efficiency**

* Ideally, the new medical claims processing system would eradicate duplicate payments, reduce the billing turnaround time from a month to days, and ensure that no backlog of claims exists. This is important as duplicate payments will result in increased costs, slow turnaround time would cause customer dissatisfaction and the backlog would likely inflict greater stress on the staff, which may in turn cause lower morale and in the worst case staff turnover.
* Examples include:
  + **Time behaviour**: billing turnaround time
  + **Throughput rates:** to avoid backlogs
  + **Capacity**: Being able to handle a large number of claim transactions
* In addition, the system should enable millions of ringgit a year in further savings by allowing flexible contacting and discounting with providers for the best deals.

2. **Functional suitability**

* This is important in order to ensure that the operations related to medical claims processing is able to be carried out smoothly as this would impact existing customer satisfaction and their reputation, which in turn plays an important part in their acquisition of new customers.
* Examples include:
  + **Functional completeness**: which is important for ensuring the set of functions covers all the specified tasks and user objectives.
  + **Functional correctness**: the degree to which the system provides correct results with the required degree of precision.
  + **Functional appropriateness**: the function facilitates the accomplishment of specified tasks and objectives, for example by presenting the user with only the necessary steps to complete the tasks.

b.

Goal : **Better user experience**

Question 1 - Is the user interface design intuitive?

Metrics:

* Duration in minutes to submit a medical claim for a first time user.
* The number of errors/medical claim submission.
* Rate of user satisfaction in terms of the ease in using the system.

Question 2 - Is communication with the customer to obtain feedback sufficient and effective?

Metrics:

* Number of prototype demos with customers per month to obtain feedback
* Rate of customer satisfaction in terms of the frequency of communication.
* The number of different modes of communication used.

**Question 4**

**Business critical system**

* For example: Customer relationship management (CRM) system of a company. This system manages all the company's relationships and interactions with customers and potential customers.
* The security dimension is the most important as any breach of security, for example the exposure of customers’ personal details, will result in a loss of trust towards the company and subsequently may cause customer churn.

**Safety-critical systems**

* For example : Air traffic controllers monitor the location of aircraft in their assigned airspace by radar and communicate with the pilots by radio. To prevent collisions, these systems enforce traffic separation rules, which ensure each aircraft maintains a minimum amount of empty space around it at all times.
* The safety dimension is the most important because a system failure may result in a collision, which in turn may cause injury or loss of life of the passengers and crew or bystanders in the area of the crash. In addition, it may also cause environmental damage to the area of the crash.

**Mission-critical system**

* For example : A baggage handling system is a type of conveyor system installed in airports that transports checked luggage from ticket counters to areas where the bags can be loaded onto airplanes.
* For such a system, the reliability dimension is the most important because system failure would result in not being able to deliver the services as specified. This would cause confusion and inconvenience as flight passengers’ baggage are lost, and would cause the company to have a bad reputation among travelers

b.

**Version management**

* It's important because it keeps track of the multiple versions of system components and ensures that changes made to components by different developers do not interfere with each other.
* For example, one of the developers changed the SST rate in the POS systems from 3% to 6%. It's important to keep track of which version is using 3% and which version is using 6% here. Some systems are still based on 3% for the older transactions and some systems use 6% SST for the current transactions.

**Change management**

* It's important because it keeps track of requests for changes to the software from customers and developers, working out the costs and impact of changes, and deciding if the changes should be implemented or not.
* For example, the change request form should be provided and approved by the committee teams first to study the feasibility of the change in the systems (i.e technical feasibility, economic feasibility, etc), the impact of the changes (i.e may cause a delay in the project).