# MET CS473/673 Quiz 3

**Due 10/19/2015**

**Your name: Ruojin Cao**

1. **How might you use a model of a system that already exists? Explain why it is not always necessary for such a system model to be complete and correct. Would the same be true if you were developing a model of a new system?**

a. The reasons to use a model of a system that already exists are:

1. To understand and document the operation and architecture of the existing system.
2. To decrease the cost and time when setting up the corresponding system model.
3. To act as the focus of discussion about changes to that system.
4. To inform the implementation of the system after discussion of changes.
5. System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. It only helps the analyst to understand the functionality of the system and models are used to communicate with customers. The aim of the model in such cases is usually to help you work on parts of the system so only these need to be modelled. If the model is used as a discussion focus, you are unlikely to ne interested in details and so can ignore parts of the system in the model.
6. This is true for models of new systems unless a model-based approach to development is taking place in which case a complete model is required. You may also need a complete model when there is a contractual requirement for such a model to be produced as part of the system documentation.
7. **Develop a sequence diagram showing the interactions involved when a student registers for a course in a university. Courses may have limited enrolment, so the registration process must include checks that places are available. Assume that the student accesses an electronic course catalog to find out about available courses.**

**Look up**

Student

Course Catalog

**Display**

confirm

Course

Check

availability

**register**

1. **You are a software engineering manager and your team proposes that model-driven engineering should be used to develop a new system. What factors should you take into account when deciding whether or not to introduce this new approach to software development?**

The factors have to be considered include:

1. Is a member of the team expertise in using UML or MDA?
2. The lifetime of the software you are developing.
3. Whether the tools are available in house or you have to purchase.
4. Are the tools supporting MDA good enough for the software being developed?
5. The long term benefits of using MDA.
6. Requirements for high performance.
7. Are all team members committed to the new approach?
8. **Imagine a situation where 2 developers are simultaneously modifying 3 different software components. What difficulties might arise when they try to merge the changes that they have made?**

When 2 developers make changes to the same components, there may be changes that are incompatible with each other. Sometimes the problems maybe worse because there may be dependencies between the components that are affected by the changes. One developer implements a change in component A which depends on a particular feature of component B. Mean while the other developer working on component B implement changes to component B which make the particular feature no longer exists. With more than 2 components, the problem exacerbates because of the chains of dependencies. It is difficult or impossible to detect the incompatibilities automatically.

1. **(Only graduate students need answer this question) Look carefully at how messages and mailboxes are represented in the email system that you use. Model the object classes that might be used in the system implementation to represent a mailbox and an e-mail message.**

**Mailbox**

**Path name**

**Name**

**Date**

**contactor**

**Messages**

**Unread messages**

**Flagged messages**

**Deleted messages**

**Delete message ()**

**Move message ()**

**Flag message ()**

**Save contactor ()**

**Copy message ()**

**Create ()**

**Rename ()**

**Delete ()**

**Messages:**

**Sender**

**Receivers**

**Subject**

**Date**

**cc list**

**return path**

**message info**

**message body**

**attachments**

**signature**

**message ID**

**routing number**

**spam info**

**read ()**

**reply ()**

**reply all ()**

**forward ()**

**send ()**

**draft ()**

**print ()**

**mark ()**