MODULE OVERVIEW

ICT2106 – Software Design

TEACHING STAFF

Module Coordinator

Dr. Alex Q. Chen
E-mail: Alex.Q.Chen@SingaporeTech.edu.sg
Tel: +65 6592 1401

Module Co-coordinator

Dr. Gaurav Kapoor
E-mail: Gaurav.Kapoor@SingaporeTech.edu.sg
Tel: +65 6592 1356

LEARNING OUTCOMES

After completing this module, you should be able to:

1. Comprehend commonly used software architecture designs and their impacts on software performance.
2. Apply the appropriate software architecture styles and design patterns to design a moderately complex software system.
3. Translate a software architecture design into appropriate patterns, reusable software components, and classes.
4. Implement the detailed design for components of a software system and specify the internal interfaces between components.
5. Evaluate designs against non-functional requirements of a software system.
6. Analyse and use appropriate tools to support the software design process.

MODULE FORMAT

Lecture	Friday 9:00am – 12:00pm SIT@NYP, SR6G	• content delivery • worked exercises • tutorials • announcements	3 hours/week
Lab P1	Friday 4:00pm – 6:00pm SIT@NYP, SR5C	• consultation • practice exercises • project work	2 hours/week
Lab P2	Friday 2:00pm – 4:00pm SIT@NYP, SR5C		
Self-Study		(including assignments)	10 hours/week

SCHEDULE

Week	Topic
1	Introduction to Software Design
2	Principles of Software Architecture
3-6	Patterns and Styles of Software Architecture
7	RECESS
8	Principles of Detailed Design
9-10	Design Patterns
11	Construction Design
12	Design Evaluation
13	Module Revision

This schedule is subject to minor alterations.

ASSESSMENT

Item	Description	Due	Weight
Project (Group)	- Design and implement one module of a large application - Document the design - Integrate it into the application	Weeks 4, 6 & 9, 13	30%
Assignments (Individual)	- Research topics related to software design or submit coded solutions to software design problems. - Assessment may require submission of students' own work and their peer reviews of other students.	TBC	10%
Labs (Individual)	- Weekly design and implementation exercises - Assessment may require submission of students' own work and their peer reviews of other students.	Weekly	10%
Quizzes (Individual)	- Written tests (or online quizzes) of knowledge of software design	TBC	10%
Final Exam (Individual)	- Written test covering all material	Exam Period	40%

This schedule is subject to minor alterations.

SOFTWARE DESIGN

This module focuses on the DESIGN of large and complex software systems:

- principles of software design and evaluation of designs
- software architecture
- design patterns
- software design notation and tools

What is Complexity?

Anything related to the structure of a software system that makes it hard to understand and modify the system.

Complexity can take many forms. It might:

- be hard to understand how a piece of code works
- take a lot of effort to implement a small improvement
- not be clear which parts of the system must be modified to make the improvement
- be difficult to fix one bug without introducing another

If a software system is hard to understand and modify, then it is complicated.

If a software system is easy to understand and modify, then it is simple.

Have you come across or worked on a complex software before?

Can you recall how it feels like working on the project?

TEXT BOOKS

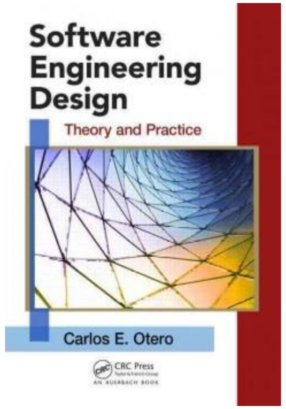
Martin Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley, Boston, Massachussetts, 2003.

- electronic access via Safari Text Books Online
- http://libproxy.singaporetech.edu.sg/login?url=http://proquestcombo.safaribooksonline.com/?uiCode=&xmlId=0321127420

The Addison-Wesley Signature Series
Patterns of Enterprise Application Architecture
Martin Fowler

Carlos E. Otero, *Software Engineering Design: Theory and Practice*, CRC Press, Boca Raton, Florida, 2012.

- print versions available for short loans in the SIT Library QA76.758 Ote 2012



TOOLS

The programming language used is C#.

- similar to C++ and Java; you are expected to pick up C# quickly and independently

The project will require .NET Core, ASP.NET Core and Git.

- .NET Core and ASP.NET Core : https://www.microsoft.com/net/core
- Git: https://www.git-scm.org

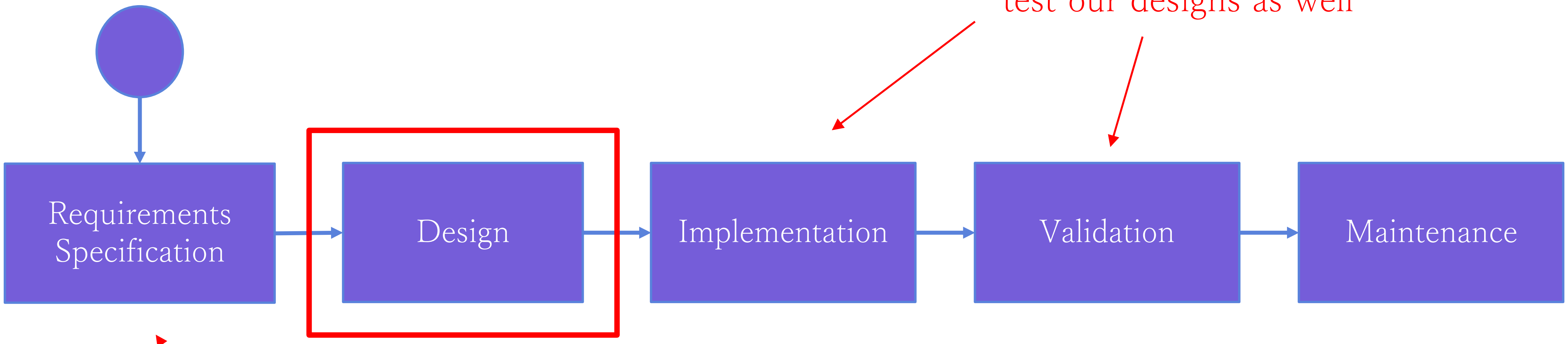
PRIOR KNOWLEDGE

To succeed in this subject, you will need to have knowledge of:

- principles of software engineering (ICT2101)
- object-oriented programming (ICT1009)
- some Web programming (ICT1004)
  - the major project will be a Web application

PRIOR KNOWLEDGE – SOFTWARE ENGINEERING

Software design is part of the software development lifecycle.



We will implement and test our designs as well

Requirements will be covered concurrently in ICT2108

(adapted from ICT2101 Introduction to Software Engineering)

PRIOR KNOWLEDGE – OBJECT ORIENTED PROGRAMMING

You should know how to recognise and use:

- classes and objects
- aggregation and composition
- inheritance
- abstraction, interfaces and polymorphism

We will use the Unified Modelling Language (UML) for describing designs:

- class and object diagrams (ICT1009 and ICT2101)
- sequence and communication diagrams (ICT2101)
- other diagrams will be introduced in ICT2106

PROJECT TEAMS MEMBERS

https://docs.google.com/forms/d/e/1FAIpQLSe8VNi_c1rQWHLO-bZTuul-4afmD8SxVbyBYyw_9ZPB8liOig/viewform?usp=sf_link

Deadline: Wednesday, 15 January 2020 at 1 pm

PROJECT TEAMS

Form your team

6 members each team

Everyone:

- must be a member of a team
- can only be a member of one team

Since you can select your team members:

- we expect better team cohesion
- you must be willing to adopt teamless members if your team is less than 6 members

PROJECT & MODULES

- One module per team only
- Every module can be done only by one team
- Discuss with the other teams in your Lab group to finalise your project modules
- You are allowed to redefine the modules suggested or create new modules

PROJECT PROTOTYPE 1 RUBRIC

Overall Architecture Design	33%
Components and Class Diagrams	25%
Report Writing	15%
Prototype	12%
Teamwork & Planning	15%
Total	100%

On top of the Project Prototype 1's rubric will be your peer evaluation that will affect your final marks for this component