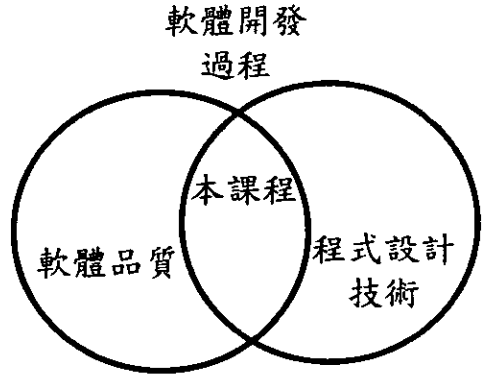
7.4 軟體發展技術

7.4.1 課程設計理念

前言：今日市面的應用軟體功能強大，數十至數百萬行的程式比比皆是，需告團隊合作方可在指定期間內完成，因此有必要訓練學生熟悉軟體文件標準及寫作技巧、軟體測試與除錯技術等等，以培育其製作高品質的大型軟體模組能力。。



教育目標：

1. 使學生了解軟體生命周期、及軟體工程之由來、目的及範圍。
2. 使學生瞭解軟體品質的意義與指標，與提升軟體品質的一些技術。
3. 訓練學生熟悉程式偵錯與測試技術，及運用進階的(OOP)程式開發技術開發複雜軟體。
4. 訓練學生使用 UML 製作軟體文件的能力。
5. 訓練學生瞭解軟體發展各階授工作內容與軟體製作階段之所需技能。

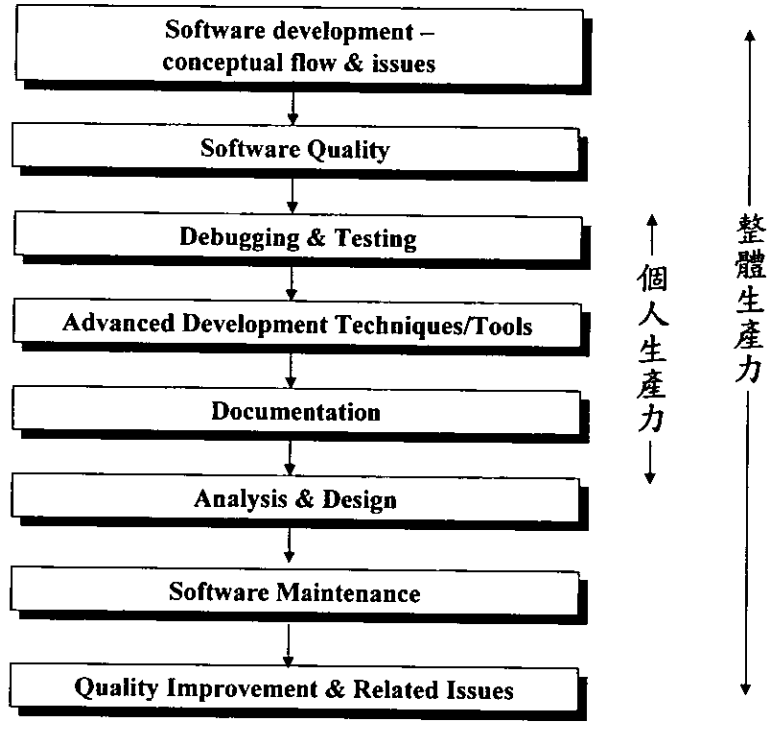
課程內容：

(1) 軟體品質基本觀念與軟體開發相關程序
(2) 軟體工程概論
(3) 軟體測試策略與技術
(4) 軟體文件種類與製作
(5) UML
(6) 軟體分析，設計維護及其他相關議題

設計構想與進行方式：

本課程強調的是程式設計師在整個軟體系統開發過程中，所扮演的角色及其相對應具備的知識與技能。希望學員完成此課程後，能夠在設計程式時，可以兼顧軟體品質，以最有效的方式完成設計程式的任務，解決問題。希望微觀(micro view)的從單一程式設計師設計單一程式的經驗與軟體品質的要求出發，進而涵蓋宏觀(macro view)的整個軟體系統的開發與品質，來強調程式設計師、軟體品質、軟體發發技術與工具的關係。課程的基本精神在於讓

學生在做中學(learning by doing)。課程進行時以學生以前設計開發的程式為對照經驗，用以驗證各單元的內容。並適度加入專題或較大型的程式系統開發，以強化學習成效。



本課程分成 8 個單元，共 48 小時：

Unit 1: Software development – conceptual flow & issues（3 hr）

- 簡介軟體系統開發過程，從單一程式設計師設計單一程式到多人參與的大型開發案可成會遭遇的問題，引導學生瞭解軟體工程的需求與核心問題，並從中瞭解程式設計師的定位。

Unit 2: Software Quality （3 hr）

- 介紹軟體品質的定義及需求，並介紹影響軟體品質不良的可能環節。同時請學生用本單元的經驗分析以前自己開發的程式的軟體品質。

Unit 3: Debugging & Testing （9 hr）

- 介紹經由 debugging 與 testing 的方式來強化軟體品質的過程與技巧。介紹各種可以提升程式設計師個人生產力的 debugging 技巧與工具，及軟體進行 testing 時的各種要求與標準。同時請學生用本單元的技巧與準則分析以前自己開發的程式的軟體品質。

Unit 4: Advanced Development Techniques/Tools （9 hr）

- 介紹數種可以提升程式設計師個人生產力的程式設計技巧與工具，例如新的技術標準(如 C#, .Net, OOP, XP 等)。強調 OOP 的進階技術，以銜接以前程式設計課程。同時請學生用本單元的技巧與準則進行幾個相關的練習。

Unit 5: Documentation （12 hr）

- 介紹文件撰寫在軟體開發工程中扮演的重要性，並特別強調與程式設計師在開

發程式的過程中有關的文件製作。利用 UML 此一工具，讓學生用本單元的技巧與準則進行以前自己開發的程式的文件撰寫。

Unit 6: Analysis & Design（6 hr）

- 簡介軟體系統的設計與分析過程。利用 UML 讓學生學會系統分析與設計相關的工作產出（如 use cases, sequence diagram, collaboration diagram, etc.）。

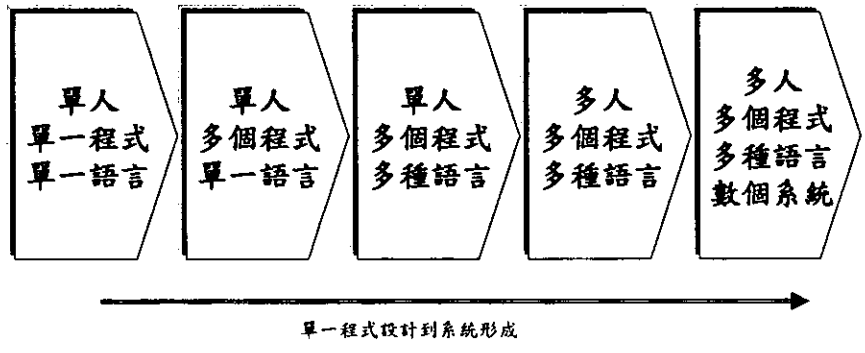
Unit 7: Software Maintenance （3 hr）

- 介紹軟體系統(或程式)完成後如何維護，內容涵蓋的是與程式設計師在開發時應該注意的相關知識與技巧。介紹軟體維護的工作種類與內容及管理程序等，如何將軟體設計的易於安裝、維護與更新，並介紹例如 CVS 此類的軟體版本控制方式。同時請學生用本單元的技巧與準則調整以前自己開發的程式的安裝與管理方式。

Unit 8: 軟體品質改善相關議題（3 hr）

- 由軟體系統開發的不同階層（個人、工作小組、組織）的觀點簡介軟體開發程序與軟體品質改善相關議題，如 PSP、TSP、CMMI 等。

各單元進行的順序的關係如下圖所示：



7.4.2 課程大綱

Core Knowledge	Programming Skill Correspondence	Hours
1. Software development – conceptual flow & issues ■ Concept & Definition ■ Software crisis (not on schedule, meet budget and requirements) ■ A programmer with one (big) program → a team of programmers with one big and complicated software system ◆ The need for SOP (standard operating procedure) for software development ◆ The needs of Documentation, process, management (of people and software) Why? ■ Life-cycle of software development ◆ Requirement phase, Specification phase, Design phase, implementation phase, test & integration phase, maintenance phase, retirement phase ◆ The tasks performed in each phase ◆ The requirement and possible errors in each phase ■ Software engineering: ◆ Why? ◆ definitions, scope ◆ the needs ◆ objective: method→process→tool ■ The role of programmers in software engineering ■ How/Why can SE improve the productivity of programmers and the quality of software?		3
2. The concept of software quality (SQ) ■ Concept, Definition & examples ■ How to justify & evaluate the quality of software ◆ Several indicators ◆ By one programmer v.s. by a team of programmers ◆ The flow/process that effect SQ ◆ The factors that effect SQ in the development of software by programmers ■ SQA(Software Quality Assurance) and V&V ■ SQA and V&V—methods and evaluation	■ Define and evaluate the quality of the programs that you have ever done.	3

■ For programmer: bug-free; debugging & testing ■ For software: meet the requirements ◆ Code review vs SQA: why & how ◆ Documentation vs SQA: why & what & how		
3. Debugging & Testing ■ Debugging: Concept, Definition & examples ◆ 除錯技術：Bugs Management 概念、除錯方法、除錯工具之觀念與使用 ◆ Bug finding/allocation ◆ Bug management ◆ Bug fixing ---------- (3hr) ■ Testing: ◆ 軟體測試功能 ◆ 軟體測試種類：功能性測試、模組化測試、安全性測試，整合測試（Graph-based 測試, etc.）、等價劃分、邊界值分析、比較測試、GUI 測試，主從系統測試、即時系統測試 ◆ 軟體測試策略：白箱測試、黑箱測試 ◆ 軟體測試步驟：測試計畫之撰寫、測試計畫之執行，單元測試、整合測試、驗證測試、系統測試 ---------- (4hr) ■ Test cases ◆ Generation & Preparation of test cases: how and where ◆ Correctness/Completeness/soundness ◆ 測試文件：測試計畫文件、測試案例 ---------- (3hr)	■ According to your programming experience ◆ Summarize the types, reasons, and solutions of bugs that you ever met ◆ List the debugging techniques that you ever did ◆ Analyze the effectiveness of new debugging techniques when being applied to your programs ◆ Test the programs that you have ever developed	9
4. Advanced Development Techniques/tools ■ 模組化程式設計 ■ 程式設計程序、模組分割、元件化程式設計 ■ New (latest) technologies in programming/system development ◆ OOP、XP、C#、.net、etc. ---------- (2hr) ■ Advanced techniques in OOP ◆ 物件導向的進階技術：more about 類別、物件與屬性、套件與介面、關聯、繼承與聚集、多形與連結 ◆ Polymorphism ---------- (2hr) ◆ Method Overriding/Overloading ---------- (2hr)	1-2 Homework Assignments	9

◆ Java Beam ---------- (1hr) ◆ N-tier applications with Java ---------- (2hr)		
5. Documentation (especially the ones related to programming by one or many programmers) ■ The needs, concept & definition, types of documentation ■ Documentation for development, management, and usage & deployment of software →強調與 programmer 有關的 ■ 虛擬碼、Coding Convention ■ 軟體文件種類、軟體文件格式 ---------- (3hr) ■ UML-based documentation： ◆ Concept, Definition & examples ◆ Basic notations in UML ---------- (6hr) ◆ Write your own and refer to others ◆ 使用案例圖、類別圖與物件圖、順序圖與合作圖、狀態圖與活動圖、元件圖與佈署圖、界面、合作、擴充機制、框架與樣式 ---------- (3hr) ■ ISO/IEC 12207, 15504 文件規範簡介	■ Denoting the programs that you have ever developed in UML (all documents related to programmer) ■ Read the documents of your partner ■ Documentation for your programs 實作：**Rational Rose** UML for existing software	12
6. Analysis & Design ■ Concept, Definition & examples ■ Analysis: 軟體系統開發人員應具備之軟體系統分析相關議題 ■ 軟體分析的工作內容 ■ 需求分析、領域分析 ■ UML for software analysis ■ 物件導向分析方法：情節為主之分析方法、使用案例之用法、使用案例規格、類別圖之推導，sequence diagram, collaboration diagram 之推導 ■ Design: 軟體系統開發人員應具備之軟體系統設計相關議題 ■ 軟體設計的工作內容--架構設計、元件設計、細部設計 ■ UML for software development (設計規格書：設計規格書項目、設計規格書撰寫) ■ Corresponding to Unit-5: The needs of documentation (architecture, components, details)	Sequence Diagram Collaboration Diagram UML for requirements of the outsides	6
7. Maintenance of software ■ Concept, Definition & examples ■ 軟體維護	■ According to your programming experience ◆ Maintenance	3

◆ 種類與步驟 ◆ 軟體可維護性、軟體維護程序、軟體維護評量、軟體維護計劃、軟體維護管理、軟體維護記錄 ■ 軟體組態管理 (configuration management)： ◆ 軟體組態程序管理、軟體組態識別、軟體組態控制、軟體組態稽核、軟體組態報告、軟體交付管理 ■ CVS(Current Versions System)	◆ Configuration management ◆ Upgrading & version control	
8. 軟體品質改善相關議題 ■ Concept, Definition & examples ■ 階層式軟體品質改善 ■ A person→group/team→organization: different requirements and methods ■ Personal: PSP(Personal Software Process) ■ Grouped: TSP(Team Software Process) ■ Organizational; CMMI(Capability Maturity Model Integration) ■ 軟體工程的未來趨勢		3
Total: 3*16=48		48