

國立屏東大學 資訊工程學系 物件導向軟體工程 (last updated September 6, 2021)

1. 軟體工程概述

軟體工程(software engineering)顧名思義就是「製作軟體(software)的工程(engineering)方法(methodologies)用以確保高品質的軟體產品能夠在有限的資源、預算與時間內開發完成。本章將從軟體的定義開始，針對軟體及軟體工程相關的主體加以簡介，讓同學能夠在開始學習前有足夠的基礎。

1.1 軟體(Software)

相較於有形的「硬體(hardware)軟體(software)則是無形的。舉例來說Nintendo Switch遊戲機以及其相關的搖桿與健身環等就是硬體裝置；而「瑪利歐賽車」、「動物森友會」、「健身環大冒險」等遊戲就是軟體。同樣的道理，也可以套用在音樂產業，音樂作品是軟體，用以播放作品的音響設備則是硬體。硬體與軟體必須互相配合，才能發揮功能或提供使用者相關的服務。例如單有遊戲主機，而無遊戲光碟；或是單有音響主機，卻沒有音樂光碟。反之，單有遊戲或音樂光碟，但卻沒有對應的執行或播放硬體，也都是沒有意義的。

<note> 用來存放軟體或是音樂作品的光碟則被稱為載具(carrier) </note>

本課程所關注的軟體，專指在電腦系統上運作的軟體，又稱為電腦軟體(computer software)當然，電腦系統是一個泛稱，並不侷限於桌上型或筆記型電腦，還包含了伺服器主機、智慧型手機、平板電腦，甚至是前述的家用電視遊樂器等各式裝置。電腦軟體可以被視為是一些電腦程式(Computer Program)的集合，當其被加以執行時可以提供特定的功能或服務；至於一個電腦程式又可以被視為是一些程式碼(code)的集合，用以完成特定的功能。除此之外，與電腦程式相關的資料以及說明文件(操作與安裝說明等)都應該被視為是軟體的一部份。綜合上述，我們將「軟體」定義為：為了在電腦系統上執行以達成特定目的一組電腦程式、資料及其文件的組合。

1.1.1 軟體的特性

基本上，軟體是無形的(intangible)它沒有一般商品存在的耗損(wear out)問題，不過它可能存在效能退化(deteriorate)的問題（你有沒有遇過隨著時間的推移，軟體執行效率退化的情況？例如因為累積的資料過多，造成需要處理的時間愈來愈長）。另一方面，軟體是極為容易再製的(reproduce)不管製作多少份複本，其成本不會隨之增加（只要一個cp的指令即可完成）。一般而言，軟體的成本集中在其開發階段，一旦開發完成，其再製成本極低，不過後續的軟體維護也必須投入相當的人力與物力。整體而言，軟體產業可算是一個勞力密集(labor-intensive)的產業，不像其它的製造業，其製作或開發的過程很難加以自動化。

1.1.2 軟體的分類

通常軟體可以依據用途與功能區分為系統軟體(system software)與應用軟體(application software)兩大類，例如作業系統(operating system)程式語言的編譯器(compiler)與裝置驅動程式(device driver)等用以控制或操控硬體的軟體；至於應用軟體則是以提供使用者服務為主要目的，依用途又可再區分為文書處理軟體、會計軟體、即時通訊軟體、簡報軟體與遊戲軟體等多個類別。

另一方面，我們也可以依照軟體使用者的不同，將軟體可以分為客製型軟體(customized software)與通用型軟體(generic software)其中客製型軟體是針對特定使用者、特定的用途所加以製作，通常此類軟體只有少數的複本存在；非客製型的軟體則被稱為「通用型軟體(generic software)它是由軟體廠商針對特

定市場所開發的軟體產品(product)通常在市場上公開向所有使用者販售，有時又稱為「商用成品(commerical off-the shelf COTS)」消費者購買後可以不經修改而直接使用(或是再付費進行小幅度的修改)。這兩種軟體的差別在於，客製型軟體是針對明確已知的使用者及其需求客製化設計開發，至於通用型軟體則是由軟體開發者預想使用者可能的需求而加以設計開發；簡言之，這兩者的差別就如同量身定製的西裝與大量生產的成衣西裝一樣，穿起量身定製的西裝當然合身帥氣，可以卻價格高昂；相對之下，成衣的價格則便宜親民多了，不過只能在現有的尺碼上做出選擇，有時可能還要再花點錢修改一下才能穿上身！

本課程受限於18週的課堂時間，將以通用型的軟體開發為主要演示，但同學們在分組開發軟體時，可依喜好在客製型軟體與通用型軟體間進行選擇。

1.2 軟體工程(Software Engineering)

軟體工程(Software Engineering)一詞，首見於1968年的NATO的Software Engineering Conference.¹⁾²⁾，用以面對當時已注意到的軟體危機 – 在當時絕大多數的軟體專案都面臨開發成本不斷追加，以及不斷推遲交貨時間的困境。不像傳統的製造業或營造業，已存在許多自動化或是專案管理的方法，軟體產業極需好的方法在有限的時間與預算內，開發出具有高品質的軟體。

用更簡單的話來說，軟體工程就是關於從無到有產生軟體的過程；而這個過程希望能夠有工程、科學化的方法，讓軟體的開發能夠如同其它領域的商品開發一樣，採用有效的方法來確保軟體開發的效率、品質以及成本的控管。為了達到此一目的，軟體工程應該被視為是一種專業技能，可以應用在軟體的設計、開發、實作以及修改、維護等過程，並希望能達成軟體開發的品質保證、降低成本、易於維護，同時也希望透過軟體工程的方法來縮短開發的時程³⁾。同時，軟體工程也被期望能向許多其它領域一樣，能夠提供包含分析、設計、評量、實作、測試、維護與精進的工程方法⁴⁾。這些所謂的工程方法必須是易於理解、管理與應用的方法，同時也被期待是易於再製且能夠具有標準化程序的方法。用更簡單的話來說，軟體工程就是採用系統化、規章化、可量測的方法，來進行軟體的開發與維護⁵⁾。

軟體的開發過程，也可以視為是從抽象化到具體化的過程，從一開始較高抽象層次的系統分析、設計，到逐漸具體化的程式碼撰寫與測試等過程。同時在這些過程中，我們必須建構出完善的工程方法來獲致具有經濟效益、且能夠在提供穩定與效率的高品質軟體⁶⁾。為了回應這些需求，目前的主流做法之一就是將軟體工程視為是建立與使用模型的過程⁷⁾。模型必須是明確且易於理解的，在軟體開發的過程當中針對不同抽象層次建立並使用這些模型，就可以在真正進行軟體開發前將最終軟體的樣貌先加以呈現。其中需求與使用者界面模型，有助於軟體開發團隊與使用者進行相關的討論；更具體的設計模型，則可以用來描述實際需要開發的各項軟體組件之功能與設計準則；建置模型則可以應用在軟體的配置與維護等方面的工作。本課程後續也將就這些不同的模型加以討論，並以UML模型為主要的探討對象。

1.2.1 軟體工程的目標

軟體工程的主要目標可列示如下：

- 使用適當的方法來描繪使用者的需求；
- 確保軟體的品質；
- 用以開發小型或大型、簡單或複雜的軟體產品；
- 縮短開發時程、降低開發成本；
- 進行軟體開發的風險管理；
- 用以在不同的方案中，尋求適合的解決方案；
- 支援軟體的精進。

1.2.2 軟體工程的範疇

如前所述，以最簡單的方式來說，軟體工程就是從無到有將軟體開發出來的過程，其相關的範疇(與簡要內容)如下：

- 分析(Analysis)[]釐清使用者需求(搞清楚到底要開發什麼樣的軟體！)
- 規格化(Specification):依需求制定規格(寫明到底要開發什麼功能！)
- 設計(Design)[]制定實現需求的方式(想清楚該用什麼方法來實現所要開發的功能)
- 開發(Development)/實作(Implementation)[]實現設計構想(寫程式來完成這些功能)
- 測試(Testing)[]測試功能是否正確、是否滿足使用者需求?
- 建置(deployment)[]建置軟體運行的軟硬體環境，並加以安裝
- 維護(Maintenance)[]維持軟體系統長期運作

1.2.3 物件導向軟體工程

發展於90年代初期的物件導向(object-oriented)概念與相關技術，現今已普及成為當代軟體開發的主流方法；軟體工程領域也廣泛地使用物件導向技術來進行各項活動，並稱為物件導向軟體工程(object-oriented software engineering)[]相關的活動包含：

- 物件導向分析(Object-oriented analysis, OOA)
- 物件導向設計(Object-oriented design, OOD)
- 物件導向程式設計(Object-oriented programming, OOP)
 - 使用物件導向程式語言(Object-oriented programming language, OOPL)進程式之設計，例如使用Java, C++, C#, Object C/C++等程式語言

本課程後續也將把重點放在物件導向軟體工程的相關活動與技術介紹。

1.3 軟體專案管理(Software Project Management)

現在的軟體開發愈趨大型化、複雜化，往往不再是一個人或少數人可以獨立完成的，所以軟體工程也與軟體團隊的管理緊密相關。專案(project)⁸⁾係指一項暫時性的工作，其目的在於創造出特定的產品或服務。專案通常具有以下特性⁹⁾ []

- 有預定的目標與時程
- 具有獨特性
- 是暫時性的工作
- 是資源的整合
- 成敗受外在因素影響甚大
- 變動性大

所謂的軟體專案(Software Project)即為以開發軟體為主要目的的專案。就軟體產業而言，不論是客製型的軟體或是通用型軟體，其開發通常都被視為是一項軟體專案。每一項軟體專案都必須針對有限的資源(包含人力、設備、時程等)進行的管理，相關工作稱為軟體專案管理(software project management)[]當然也不是每一項軟體專案，都是以開發新軟體為目的，有些軟體專案是針對既有的軟體進行改善或維護等工作；又例如有的軟體專案是為了找出既有軟體的缺失並加以修正，又或者有的軟體專案是為了將既有的系統進行修改，使其能適用於更新版本的作業系統、資料庫系統、或是新的商業模式或法規。另外有些軟體專案則是為了幫既有的軟體增加或修改功能、有時候是由開發人員針對既有的功能進行新的設計，以提升軟體的效能、或透過模組的重新設計提升軟體的可維護性。

1.3.1 相關人員(Stakeholders)

軟體專案的相關人員(stakeholders)包含：

- 使用者(Users):軟體的使用者。
- 客戶(Customers):支付或購買軟體的人。
- 軟體開發的相關人員:
 - 系統分析師
 - 軟體工程師
 - 程式設計師
 - 測試工程師
 - 維護工程師
- 專案經理(Project Manager[PM])負責軟體專案成敗的決策者、領導者。
- 軟體的銷售與發行人員

1.3.2 軟體專案團隊(Software Project Team)

負責開發軟體專案的相關人員將被視為(或實際上即為)一個團隊，如何在團隊間適切地分工、溝通合作，也成為軟體工程相關領域的重點之一。主要的困難在於以下數點：

- 如何將整個軟體切割為適切的組件，讓團隊成員得以分工合作，同時各自完成的軟體組件也必須確保能夠正確地組合運作。
- 最終完成的軟體必須具有一定的品質。
- 軟體的開發必須滿足經費、時間等成本限制。
- 軟體開發的過程中，其可運用的各項資源往往都是有限的。
- 軟體開發的過程中，還必須確保所投入的各項成本都能夠帶來適切的助益。
- 有時軟體的開發還必須與其它團隊在成本與時程上競爭。
- 不正確的成本控管往往會造成軟體專案的失敗，但成本的控管有許多不可預知的因素。

1.4 簡易繪圖軟體開發

本課程以一套繪圖軟體為例，實際演示使用軟體工程方法開發軟體的過程。課程所開發的繪圖軟體，可以讓使用者使用基本的筆及圖形(例如矩形、橢圓形等)元件進行繪圖。希望透過此實際個案的示範，幫助同學理解軟體工程的相關活動及設計與實作方法。

1)

P. Naur and B. Randell, (Eds.), "Software Engineering: Report of a conference sponsored by the NATO Science Committee", Garmisch, Germany, 7-11 Oct. 1968

2)

B. Randell and J.N. Buxton, (Eds.), "Software Engineering Techniques: Report of a conference sponsored by the NATO Science Committee", Rome, Italy, 27-31 Oct. 1969

3) 4)

Phillip A. Laplante, "What Every Engineer Should Know about Software Engineering", ISBN 9780849372285, 2007.

5)

Alain Abran, James W. Moore, Pierre Bourque, Robert Dupuis, Pierre Bourque and Robert Dupuis, "Guide to the Software Engineering Body of Knowledge", 2004 Version, IEEE Computer Society, ISBN 0-7695-2330-7, 2004.

6)

P. Naur and B. Randell, (Eds.), "Software Engineering: Report of a conference sponsored by the NATO Science Committee", Garmisch, Germany, 7-11 Oct. 1968.

7)

I. Jacobson, M. Ericsson, and A. Jacobson, "The Object Advantage: Business Process Reengineering With Object Technology", Addison-Wesley Professional, ISBN: 978-0201422894, September 30, 1994.

8)

William R. Duncan, "A Guide to the Project Management Body of Knowledge", Project Management Institute, 1996.

⁹⁾

林信惠, 黃明祥, 王文良, "軟體專案管理", 初版, 智勝文化, 2002.

From:

<https://junwu.nptu.edu.tw/dokuwiki/> - Jun Wu的教學網頁

國立屏東大學資訊工程學系

CSIE, NPTU

Total: 173168

Permanent link:

<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=se2021:introduction>



Last update: **2024/09/05 13:54**