

「完整版PPT」版權申明及使用說明

□本PPT為《物件導向系統分析與設計—結合MDA與UML》一書之「完整版PPT」（編號：978-957-511-275-2），本版之授權範圍僅限智勝文化之教師資源專區及指定用書之教師個人使用，遂不得任意將本版內容上傳非智勝文化之教師資源專區，或以任何形式重製（舉凡複製、掃描檔案等行為）、公開散佈、傳播給他人（含學生等），以免觸犯著作權法而受有刑事責任，請尊重本書作者之智慧財產權，切勿以身試法。

第一章 資訊系統開發概論

內容大綱

□ 本章主要讓學生學習下列重要概念：

1.1 導論

1.2 資訊系統開發環境

1.3 資訊系統開發的問題

1.4 結論

教學目標

□ 詳讀本章，你至少能瞭解：

- 資訊系統開發環境。
- 系統分析與設計的發展趨勢。
- 系統開發的問題及解決之道。

1.1 導論(1/4)

- 為有效管理與應用資訊，以支援組織之經營管理和決策需求，各資訊系統 (Information System, IS) 應運而生，常見的資訊系統包括：
- 交易處理系統
 - 管理資訊系統
 - 企業資源規劃系統
 - 決策支援系統
 - 專家系統
 - 高階主管資訊系統等

1.1 導論(2/4)

- 雖這些資訊系統有其不同特性，開發策略與技術也不盡相同，但系統開發過程仍可大略歸納出一些基本而共同的步驟或階段。如：
- 較單純之系統：可分為需求分析、系統分析與設計、系統實施三個階段 (如圖1-1a)。
- 較複雜之系統：則可分為七個階段 (如圖1-1b) 或更多階段。

1.1 導論(3/4)

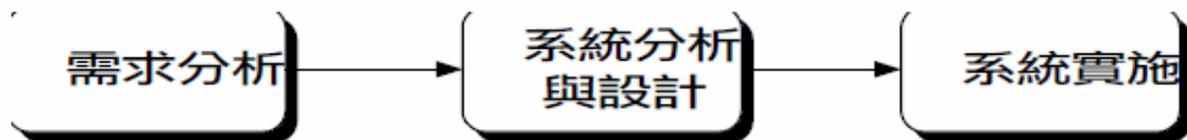


圖1-1a、三階段資訊系統開發過程

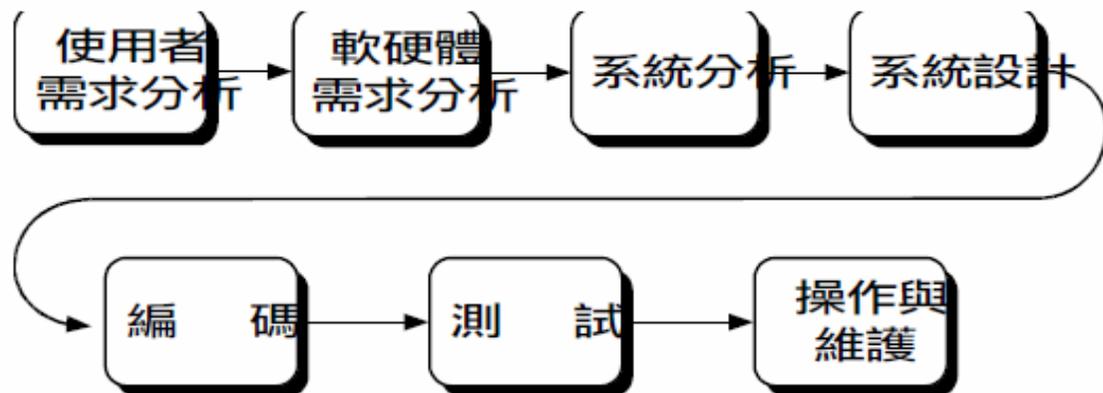


圖1-1b、七階段資訊系統開發過程

1.1 導論 (4/4)

- 早期資訊系統主要應用於工程、科學或會計等專業領域，系統開發者與使用者常為同一人或兩者關係密切。
 - 當時，資訊系統之開發常憑開發人員的經驗和喜好進行，未依照系統開發方法，因此系統大多無詳細文件紀錄，並造成後續系統維護、整合和擴充上的困難。
 - 後來，資訊系統普及且系統複雜度提高，促成專業軟體公司的成立與套裝軟體的發展。
 - 當系統開發者和使用者分開後，更需要一套方法來銜接兩者，因此系統分析與設計之重要性大幅提升。

1.2 資訊系統開發環境 (1/2)

- 資訊系統開發環境涉及層面廣範，
 - 包括資訊系統種類、開發人員、建置策略、開發模式、程式語言、分析與設計的理論、技術和工具等。
 - 此外，外在環境之科技、社會、教育、文化、政府政策與法規等因素，也直接或間接地影響系統分析與設計（如圖1-2）。
- 瞭解資訊系統之開發環境，可提供開發者一個宏觀的視野，有助於系統分析與設計工作的進行。

1.2 資訊系統開發環境 (2/2)

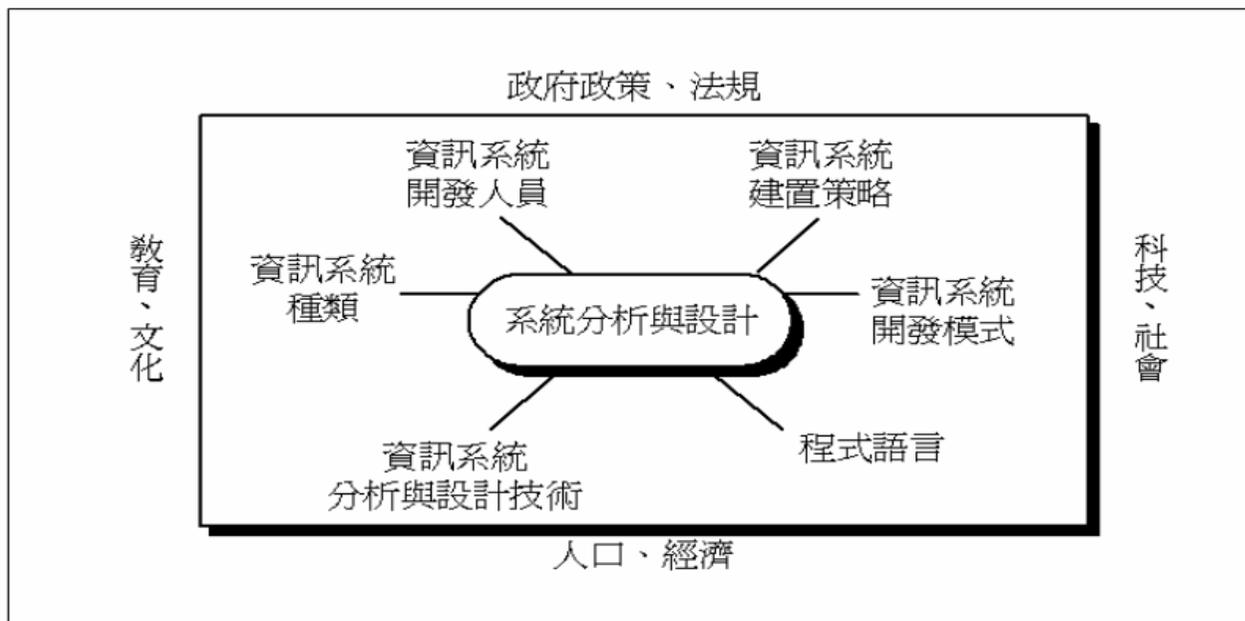


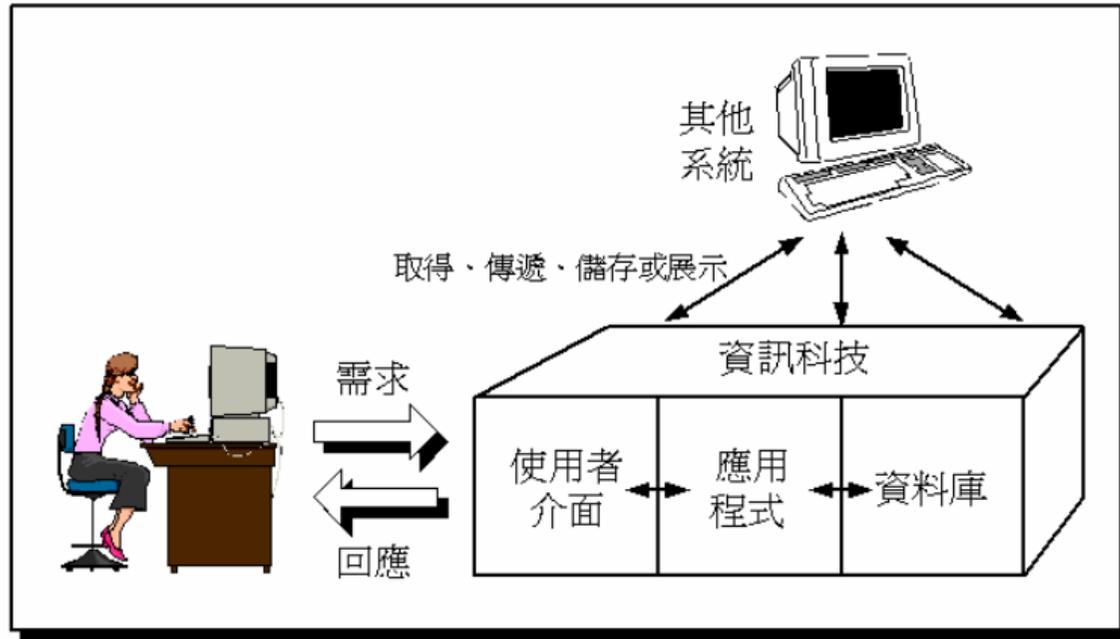
圖 1-2 資訊系統開發環境及系統分析與設計

1.2.1 資訊系統的種類 (1/3)

- 資訊系統由使用者介面 (User Interface)、應用程式(Application) 與資料庫 (Database) 等元件有組織地結合，用以管理 (如擷取、傳輸、儲存、處理與展示等) 資料、資訊或知識，並為某些活動 (如企業流程或問題解決 (Problem Solving) 等) 提供知識 (含資料、資訊等) 或服務。
- 亦即資訊系統由適用於特定問題領域 (Problem Domain) 之資訊科技，及資料庫、應用程式和使用者介面等元件所組成，用以管理知識和支援使用者之作業、服務或知識需求。

1.2.1 資訊系統的種類 (2/3)

圖 1-3 資訊系統之一般性架構及其環境



1.2.1 資訊系統的種類 (3/3)

- 資訊系統已成為企業必備之工具，常見的資訊系統有：
 - 交易處理系統 (Transaction Processing System)
 - 管理資訊系統 (Management Information System)
 - 決策支援系統 (Decision Support System)
 - 高階主管資訊系統 (Executive Information System)
 - 專家系統 (Expert System)
 - 企業資源規劃 (Enterprise Resource Planning, ERP) 系統

交易處理系統

□ 交易處理系統 (Transaction Processing System) 亦稱資料處理系統 (Data Processing System, DPS)

- 主要目的是將大量的交易處理自動化。
- 此系統主要有兩功能：
 - 交易記錄之保存
 - 交易表單之產生

實例

□ POS (Point of Sale) 系統之前檯系統、加油站之加油作業與收銀系統、金融機構之櫃檯系統皆屬交易處理系統。

管理資訊系統

- 管理資訊系統 (Management Information System, MIS)
 - 主要目的為提供不同層級的管理者，有關組織營運狀況但摘述程度不同之報表，這些報表格式已事先訂定，也就是這些資料之處理與報表產生大多為結構化的。
 - 該系統主要有兩功能：
 - 交易資料之記錄保存
 - 摘述性報表之產生

實例

- POS系統之後檯系統屬於管理資訊系統。

決策支援系統

□ 決策支援系統 (Decision Support System, DSS)

- 主要目的為支援決策者，提升其決策效率(Efficiency) 與效能(Effectiveness)。
- 決策支援系統主要支援半結構化或非結構化之決策活動，其特徵有：
 - 能以即興 (Ad Hoc)、自訂性或標準化的方式分析資料與產生報表。
 - 能直接與決策者互動。

實例

□ POS 之後檯系統，除了固定式資料查詢、處理、分析與報表產生外，若還能與使用者互動，並依其需求擷取、分析與展示資訊，則該系統可稱為DSS。

高階主管資訊系統

- 高階主管資訊系統 (Executive Information System, EIS) ，亦稱高階主管支援系統 (Executive Support System, ESS)
 - 針對高階主管之資訊需求而設計。
 - 希望高階主管能即時由系統中獲得其所需之關鍵資訊，不需透過第三中介者整理或轉達。
 - EIS 可過濾、摘述關鍵資訊，亦可將資訊以多種方式作圖形化的展示 (如儀表版、長條圖)，也得以由上而下之方式擷取資訊並進行分解。
 - 一般而言，EIS 之特徵與DSS 相同，比DSS 有更多限制，故高階主管資訊系統可視為是DSS 的一種特例。

專家系統

□ 專家系統 (Expert System, ES)

- 初期發展目的是用以模仿人類專家解決特定問題之能力，並希望專家系統所提供之答案或建議可達人類專家水準。
- 專家系統包含三個主要元件：
 1. 使用者介面：為專家系統與使用者交談之機制。
 2. 推理機：為專家系統依使用者需求由知識庫推論結果或建議之機制。
 3. 知識庫：為系統儲存專家知識之處。
- 現今，專家系統不再強調取代專家，而是支援專家，因此專家系統也漸成為決策支援系統之另一種形式。

企業資源規劃系統

□ 企業資源規劃 (Enterprise ResourcePlanning, ERP) 系統

- 為一種將多項企業功能整合在一起的模組化與架構化套裝資訊系統，藉由及時整合與掌握企業分散於各地的資源，提供最佳流程典範 (Best Practice)，以降低企業營運成本並提升客戶服務滿意度。
- 狹義的觀點視ERP 系統為整合企業內部資源規劃之系統。
- 廣義的觀點視ERP 系統為整合企業內外部資源規劃之系統，例如企業上下游之供應鏈管理等。

表1-1 資訊系統種類及特性

資訊系統種類	資訊系統特性
交易處理系統	<ul style="list-style-type: none"> ➢ 將大量的交易處理自動化 ➢ 處理程序與資訊需求非常結構化
管理資訊系統	<ul style="list-style-type: none"> ➢ 提供不同層級之管理者摘述程度不同之報表 ➢ 料的處理與報表的產生亦多為結構化
決策支援系統	<ul style="list-style-type: none"> ➢ 支援決策者半結構化或非結構化之決策 ➢ 提升決策之效率與效能
高階主管資訊系統	<ul style="list-style-type: none"> ➢ 支援高階主管即時瞭解所需之關鍵資訊 ➢ 可過濾、摘述關鍵資訊 ➢ 可將資訊以多種方式作圖形化的展示
專家系統	<ul style="list-style-type: none"> ➢ 針對特定應用範圍或領域，集合不同專家知識而成的系統 ➢ 希望專家系統提供之答案或建議可達到人類專家的水準
企業資源規劃系統	<ul style="list-style-type: none"> ➢ 將多項企業功能整合在一起 ➢ 可即時整合與規劃企業分散於各地資源 ➢ 提供最佳的流程典範

1.2.2 資訊系統開發相關人員之角色 (1/3)

- 資訊系統開發之相關人員包括
 - 系統分析師 (System Analyst)
 - 程式設計師 (Programmer)
 - 終端使用者 (End User)
 - 終端使用者之高層主管 (Manager)
 - 資訊系統經理 (IS Manager)
 - 資料庫管理者 (Database Administrator)
 - 其他技師 (Technician)

1.2.2 資訊系統開發相關人員之角色 (2/3)

□ 系統分析師

- 使用者需求擷取與分析。
 - 進一步將需求轉換成資訊技術、企業處理與知識等元件，並將之有組織地結合。

□ 程式設計師

- 主要依分析與設計藍圖設計出程式，轉換成電腦可執行的指令，及建立資料庫、測試與安裝系統等。

1.2.2 資訊系統開發相關人員之角色 (3/3)

□ 終端使用者

- 為問題領域之專家，但可能並非資訊科技方面之專家。在系統分析與設計過程中，主要扮演提供使用者需求與企業知識之角色。
- 終端使用者應能主動、積極地參與系統開發，並盡可能完整與清楚地表達需求和知識。

□ 企業經理

- 為終端使用者之高層主管 (如部門領導者或公司執行長)，雖無許多時間參與系統之開發，但在系統開發過程中扮演非常重要的角色。
- 如承諾對系統開發所需財力、人力、時間等資源之投入及提出組織政策與限制等。
- 由於企業經理之決策權與企業經營的專業知識，因此能對資訊系統專案發展設定一般性的需求與限制。系統分析師須與其保持良好的溝通，瞭解對方之期望和優先順序，以及組織內的限制和資源。

1.2.3 資訊系統建置策略 (1/2)

- 資訊系統建置策略指新資訊系統之建立與舊資訊系統之修改、擴充或更新時所採取之方式。資訊系統建置策略分為三種：
 - 由公司內部自行完成。
 - 由公司外部取得。
 - 其他方式。

- 若由公司內部完成：
 - 終端使用者自建 (End User Computing)。
 - 由公司資訊部門和相關部門人員組成任務編組開發。

1.2.3 資訊系統建置策略 (2/2)

□ 若由公司外部取得：

- 委外開發 (Outsourcing)。
- 購買現成套裝軟體 (Application Package)。
- 引進同業之系統。
- 採取租賃方式。

□ 其他方式：

- 上述各種策略之組合。
- 由部分同業聯合共同找資訊公司開發等。

1.2.4 資訊系統開發模式

□ 系統開發模式主要考量

- 開發過程分成哪些階段；每階段如何進行及包含哪些工作內容等議題。

□ 常用之資訊系統開發模式有八種：

1. 瀑布模式 (Waterfall Model)
2. 雛型模式 (Prototyping Model)
3. 漸增模式 (Incremental Model)
4. 螺旋模式 (Spiral Model)
5. 同步模式 (Concurrent Model)
6. RUP模式 (Rational Unified Process Model)
7. 敏捷軟體開發 (Agile Software Development)
8. MDA (Model Driven Architecture) 軟體發展生命週期

1.2.5 程式語言 (1/3)

- 系統分析與設計之方法與技術和程式語言之功能與特性息息相關
 - 從歷史發展觀點來看，程式語言已從1950年代起的機器碼、組合語言、程序性程式語言，演變到現今的物件導向程式語言 (參考表1-2，Kleppe et al.,2003)。
 - 結構化系統分析與設計的技術於1960年代末期興起，當時也正是程序性程式語言興起的年代。

1.2.5 程式語言 (2/3)

表1-2. 程式語言的歷史

年代	當時的程式語言
1950	機器碼
1950-1965	組合語言
1965-1985	程序性程式語言
1985-現在	物件導向程式語言
現在-	下一波為何？

1.2.5 程式語言 (3/3)

- 結構化技術配合程序性程式語言的特性與分治原理
 - 創造出結構化系統分析與設計概念，改善軟體之結構性與重用性，使結構化系統分析與設計成為1970～1980年代的主流。
- 軟體工程人員發現，**資料與處理在系統運作過程中息息相關**，兩者應緊密結合。
 - 程式語言功能不斷提升，物件導向程式語言因而興起，由1980年代末期後便逐漸取代程序性程式語言。
 - 之後，**配合物件導向程式語言特性的物件導向系統分析與設計**，其相關語言、方法與技術應運而生，成為今日主流。

1.2.6 系統分析與設計理論、技術和工具 (1/7)

- 系統分析與設計是一系列有組織之處理程序，目的為將需求轉換成有組織的使用者介面、應用程式與資料庫等元件。
 - 系統分析與設計的技術可大概分為：結構化、物件導向與元件導向。

- 結構化技術主要用於系統開發過程之分析與設計階段。
 - 結構化技術將企業流程與資料分開處理。
 - 以結構化塑模工具幫助系統分析師進行資訊系統之描述與驗證。

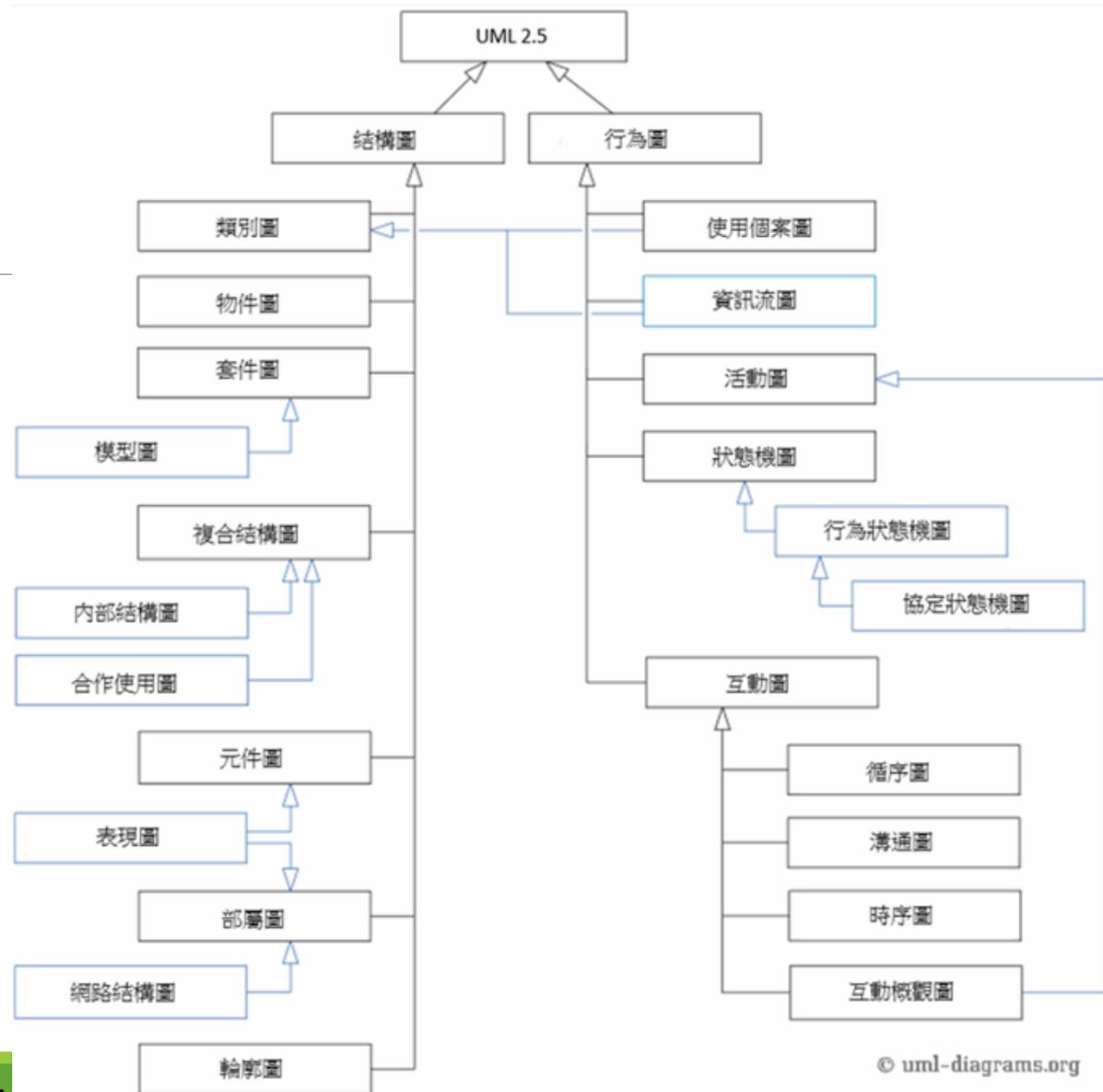
1.2.6系統分析與設計理論、技術和工具(2/7)

□物件導向之分析與設計理論、技術

- 應用模組化系統理論&OOT進行SA&D。例如：
 - 將企業流程與資料封裝成物件合併處理。
 - 另一特徵為繼承的概念，亦即當物件類別之層級結構建立後，低層物件可繼承上層物件之性質，以減少因需求改變而造成之相關程式修改。
- 以統一塑模語言進行SA&D結果之塑模。

1.2.6系統分析與設計理論、技術和工具(3/7)

□ 統一塑模語言 (Unified Modeling Language, UML) 共使用二十二種圖進行塑模 (UML 2.5)



1.2.6 系統分析與設計理論、技術和工具(4/7)

- 系統是由開發者直接用程式語言撰寫出來的，或由系統開發者使用產生器組裝出來的，或是直接導入裝軟體(如ERP系統)。
- 不管哪一種方式，都需要系統分析與設計，我們分別將之稱為針對
 1. 程式碼的分析與設計。
 2. 產生器的分析與設計。
 3. 套裝軟體的分析與設計。

1.2.6系統分析與設計理論、技術和工具(5/7)

□針對程式碼的分析與設計

- 會受所用之程式語言與系統架構影響
 - 若用程序性的程式語言來設計模組架構的系統
 - 可用結構化的系統分析與設計技術。
 - 若用物件導向的程式語言設計物件導向的系統
 - 可用物件導向的系統分析與設計技術。
 - 若用物件導向的程式語言 (搭配元件) 設計元件導向的系統
 - 可用元件導向的系統分析與設計技術。

1.2.6系統分析與設計理論、技術和工具(6/7)

□針對產生器的分析與設計

- 會受所用之產生器影響
 - 所用之產生器若為可組裝出模組化的應用系統，則須依該產生器之特性進行系統分析與設計。再依該系統分析與設計之產出逐步操作產生器，以組裝最終之應用系統。
 - 所用之產生器若為可組裝出物件導向或元件導向的應用系統，也須依該產生器之特性進行系統分析與設計。

1.2.6系統分析與設計理論、技術和工具(7/7)

□針對套裝軟體的分析與設計

- 會受所導入之套裝軟體影響，但若不考慮個別套裝軟體，而由較高階的導入方法與步驟來看，與套裝軟體的關聯就很小。
- 系統導入過程與系統分析與設計的相關工作包含：
 - 需求塑模：主要是瞭解使用者、組織或企業需求，並將需求以需求塑模工具表達；亦須將所要導入軟體之功能、可能的處理程序與資料等以相同的需求塑模工具表達。
 - 需求對應與差異性分析：此活動主要瞭解套裝軟體與需求匹配程度與確切的部位。

1.2.7 外在環境

□ 外在環境指資訊系統開發所面對的組織、社會、文化、教育、科技、法規、政策等因素，這些因素直接或間接影響到資訊系統開發的成效。

- 資訊密集行業如金融業；強調決策速度之環境如科技產業；勇於嘗試新科技的組織文化、人員教育程度較高的組織、重視智慧財產權的社會價值觀等特性，都有利資訊系統的發展。
- 科技的成熟度、科技取得的方便性、科技人員的普及性、法規的要求或同業的協定，如金融機構之跨行連線等，也是有利的不在環境因素。
- 此外，社會媒體的宣導、政府的政策推動、智慧財產權的保護、資訊教育的推廣等，均有利資訊系統開發。

1.3 資訊系統開發問題(1/6)

- 目前系統開發仍存在某些問題，使超過30%的軟體專案因無法完成而被迫取消；超過50%的專案雖可完成，但完成時之成本幾乎為當初估價的兩倍。這些問題包括：
- **生產率問題**：程式編輯仍是勞力密集的工作。
 - **可攜性問題**：應用新技術時，許多工作必須重進行。
 - **互通性問題**：系統並非僅由單一技術建構，而是以不同技術建構不同子系統後，再進行子系統間之互動。
 - **維護與文件問題**：需求持續地更改。

1.3 資訊系統開發的問題 (2/6)

□ 生產率問題

- 典型的軟體開發流程包括需求分析、系統分析與設計、細部設計、程式編輯、測試、部署等步驟，且前三項步驟在進行時均會產出書面文件，包括以文字和圖片描述需求及介面，如UML中的使用個案圖、活動圖、循序圖與類別圖等。
- 開始進行程式編輯時，前三項步驟產出之書面文件開始失去價值。
- 且隨著程式編輯工作的進展，這些書面文件與程式間之關聯性也漸漸消失，尤其當系統不斷修改，兩者間之距離也愈來愈大。

1.3 資訊系統開發的問題 (3/6)

□ 可攜性問題

- 公司遵循新技術開發軟體或系統，對舊平台而言，雖有某些技術可解決文件或平台互通問題，但原先之軟硬體工具供應商卻可能已轉向新技術，不再支援舊技術。
- 人們因快速學習新技術，導致舊技術的投資失去價值甚至變得無用。
- 各種技術本身會因更新而有不同版本，且新版本並不保證與舊版本相容，軟硬體工具供應商常僅支援最近的一、兩個版本。

1.3 資訊系統開發的問題 (4/6)

□ 互通性問題

- 很少有軟體系統單獨存在，大部分系統皆需與其他系統互動。如Web-based 系統中，使用者端應用系統執行網頁瀏覽器時，須存取後端系統的資料。
- 由於各元件須與其他元件互動，因此也產生跨平台互通性的需求。

1.3 資訊系統開發的問題 (5/6)

□ 維護與文件問題

- 大部分系統開發者認為，於系統開發期間製作文件將浪費時間與降低產能。
- 但文件與程式並非毫無關聯，且所開發之系統於完成後，常須被更改與維護，因此製作文件也是其重要的工作之一。

1.3 資訊系統開發的問題 (6/6)

- 系統開發過程中，完整的分析與設計文件皆能使相關人員迅速且清楚地瞭解現有系統之目標與限制、可完成之工作內容、作業時主要的資訊傳遞、介面元件的設計和所使用的資料等。
- 應用模式驅動結構 (MDA)結合模組化系統理論、OOT與CASE Tool等，進行需求分析及系統分析與設計，再由兩者產出之高階文件與CASE Tool等進行後續程式碼的開發，有助於直接或間接解決上述問題。

1.4 結論

- 資訊系統開發為有系統、有組織的活動，過程中需考量系統開發之環境背景，如瞭解資訊系統種類、開發人員、建置策略、開發模式、程式語言及分析與設計技術等，並做最適當的選擇。
- 在選擇時，也應考量組織之限制、政策、人員之資訊科技能力、資金、規模、對系統需求之迫切性、組織對資訊倚賴之程度與重要性等。
- 資訊人員應對上述問題有充分地瞭解，才能有效地掌握系統開發之進行。