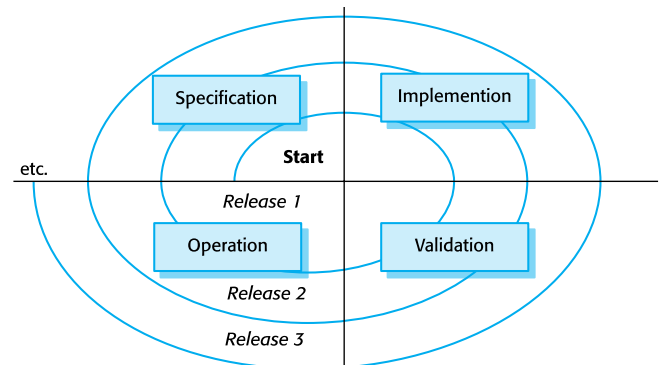
Chapter 9 – Software Evolution(軟體演進)

* Software development and evolution can be thought of as an integrated, iterative process that can be represented using a spiral model.

軟體開發和演進可以被認為是一個可以使用螺旋模型表示的整體的迭代過程。

* + A spiral model of development and evolution  
    一個發展與演化的螺旋模型  
    

Specification - 規格

Implemention - 實現

Validation - 驗證

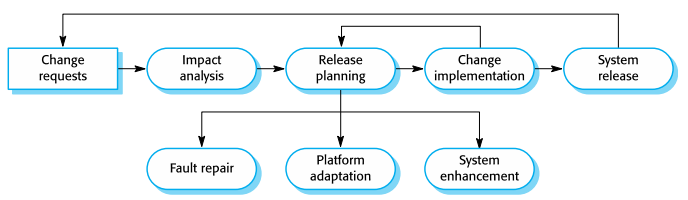
Operation - 運作

* For custom systems, the costs of software maintenance usually exceed the software development costs.

對於客製化的系統，軟體維護成本通常超過軟體開發成本。

* The process of software evolution is driven by requests for changes and includes change impact analysis, release planning and change implementation.

軟體演進過程由變更請求驅動，包括變更影響分析，發布計劃和變更實施。



The software evolution process

* Legacy systems are older software systems, developed using obsolete software and hardware technologies, that remain useful for a business

傳統系統是較舊的軟體系統，使用過時的軟體與硬體技術開發，對商業仍然有用。

* It is often cheaper and less risky to maintain a legacy system than to develop a replacement system using modern technology.

與使用現代技術開發替代系統相比，維護遺留系統通常更便宜且風 險更低。

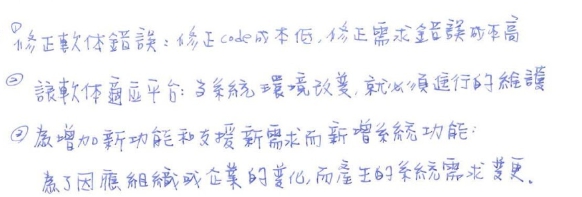
* The business value of a legacy system and the quality of the application should be assessed to help decide if a system should be replaced, transformed or maintained.

應該評估遺留系統的商業價值和應用程式的質量，以幫助確定是否應更換，轉換或維護系統。

* Briefly describe three types of software maintenance.(期末有考)

There are 3 types of software maintenance, namely bug fixing, modifying software to work in a new environment, and implementing new or changed requirements.

有三種類型的軟體維護，Bug修復、修改軟體以便在新的環境中工作以及實現新的或修改需求等。



* Software re-engineering is concerned with restructuring and re-documenting software to make it easier to understand and change

軟體再造工程涉及重組和重新記錄軟體，以便於理解和修改。

Briefly describe the differences between reengineering and refactoring in software maintenance (期末有考)

Software re-engineering 再造工程 是根本性地重新思考及徹底重新設計，以達成在苛刻的當代度量標準，諸如成本、品質、服務及速度上的戲劇性改進

* Refactoring, making program changes that preserve functionality, is a form of preventative maintenance.

Refactoring 重構 是在不改變程式的外在行為的前提之下，改變程式內部的結構以提升設計品質。

* Bad smells in program code
  + 重複的程式碼
  + 過長的函式
  + 過長的參數
  + 過大的class
  + Switch 敘述
  + 註解
  + 暫時欄位