

Software  
Engineering

軟體工程

Software Engineering

李允中

Mc  
Graw  
Hill 美商麥格羅·希爾  
資訊科學 系列叢書

高立圖書有限公司

# CHAPTER 5

## 軟體專案 計畫與管理

# 5

## 大綱

- 專案管理概念
- 專案計畫書
- 專案範圍
- 專案時間排程
- 專案成本管理
- 人力資源
- 專案風險管理
- 專案監控



# 5

## 專案管理

- 為了有效率執行專案
- 與軟體開發生命週期作連結
- 專案生命週期
  - 初始(Initial)
  - 計畫(Planning)
  - 執行(Execution)
  - 結案(Closure)



# 5

## 專案計畫書

- 又稱為專案執行計畫書(Project Execution Plan, PEP)
- 通常包含了下列章節
  - 簡介(Introduction)
  - 專案組織(Organization)
  - 工作分解結構(Work Breakdown Structure, WBS)
  - 計畫時程(Schedule)
  - 資源需求分析(Resource)
  - 風險管理(Risk)
  - 專案監控機制(Monitor and Control)

## 5

## 專案範圍

- 對於一個成功的專案，精確的界定專案範圍是一項重要的工作
- 界定專案範圍，包含下列工作
  - 計畫初始(Project Initial)
  - 範圍規劃(Scope Planning)
  - 範圍定義(Scope Definition)
  - 範圍驗證(Scope Verification)
  - 變更控制(Change Control)

## 5

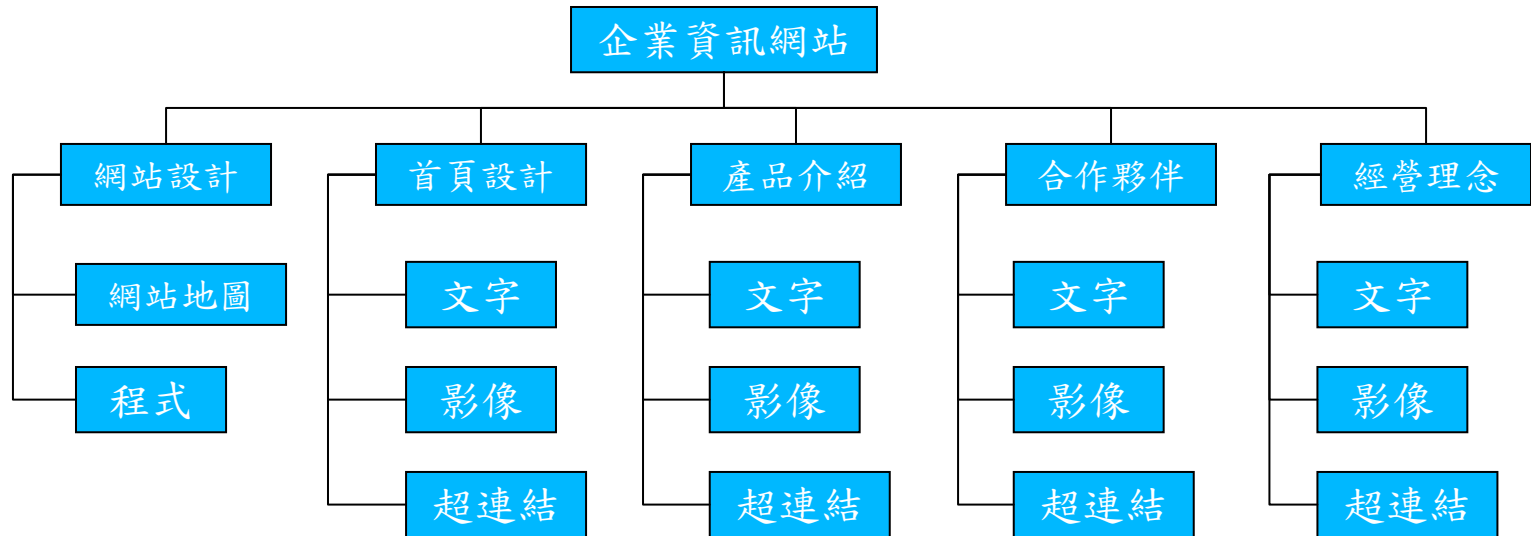
# 工作分解結構

- 工作分解結構通常用來定義
  - 各項分工產出
  - 規劃管理專案中各項時程、成本與異動
- 建立工作分解結構的方法
  - 利用發展指引
  - 類比法(Analogy Approach)
  - 由上而下法(Top-down Approach)
  - 由下而上法(Bottom-up Approach)



# 5

## 工作分解結構範例



# 5

## 變更控制

- 主要導致專案失敗之原因前三名 [1995, Standish Group]
  - 缺法使用者輸入
  - 不完整的需求及計畫
  - 變更性的需求及計畫
- 上述三項皆與專案範圍驗證與變更控制有著密切關係



# 5

## 專案時間排程

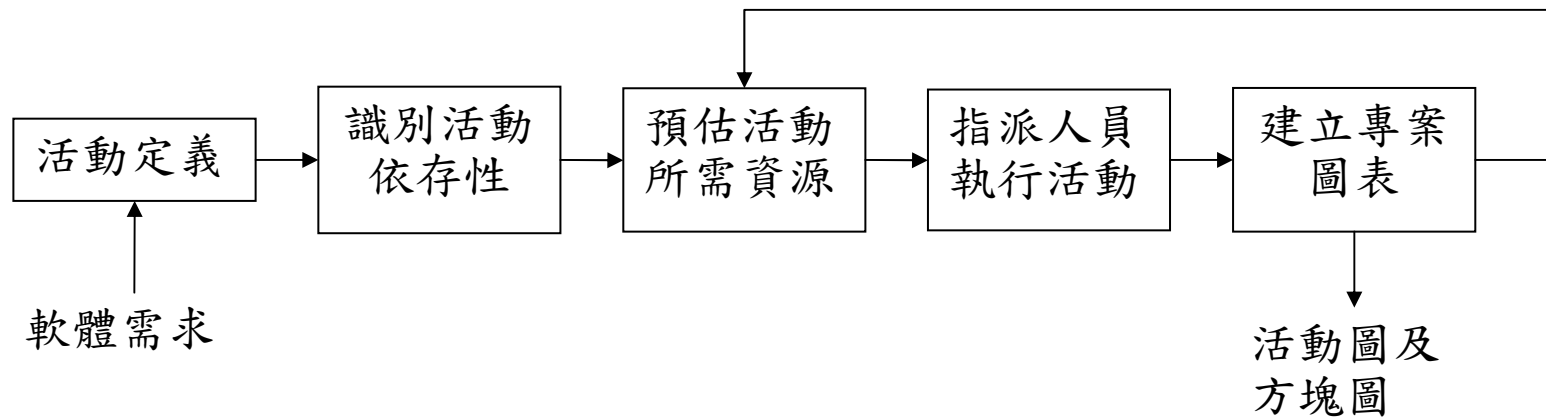
- 專案時間管理，主要包含了下列流程
  - 活動定義
  - 活動編排
  - 活動期間預估
  - 時程建立
  - 時程控制
- 以下就計畫與控制兩方面介紹



## 5

專案時程計畫<sub>1</sub>

- 專案的排程計畫，主要包含活動如下



# 5

## 專案時程計畫<sub>2</sub>

- 活動定義
  - 訂出更詳盡WBS，並提供工作相關描述
- 識別活動依存性
  - 瞭解活動間的從屬關係
    - 強迫依存性(Mandatory Dependency)
    - 隨意依存性(Discretionary)
    - 外部依存性(External Dependency)
  - 專案網路圖(Project Network Diagram)

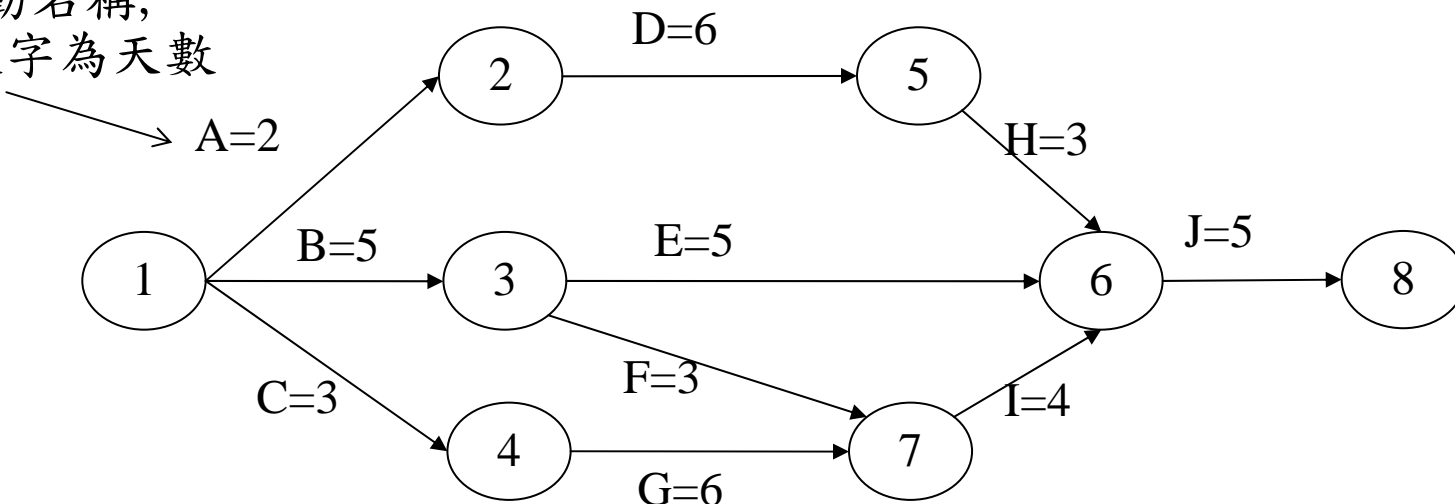


## 5

## 專案網路圖



A~J為活動名稱，  
其後的數字為天數



- 箭頭圖示法(Arrow Diagramming Method, ADM)
  - 或稱為活動箭頭圖示方法(Activity-on-Arrow, AOA)
  - 箭頭代表各項工作間的關聯性

# 5

## 專案時程計畫<sub>3</sub>

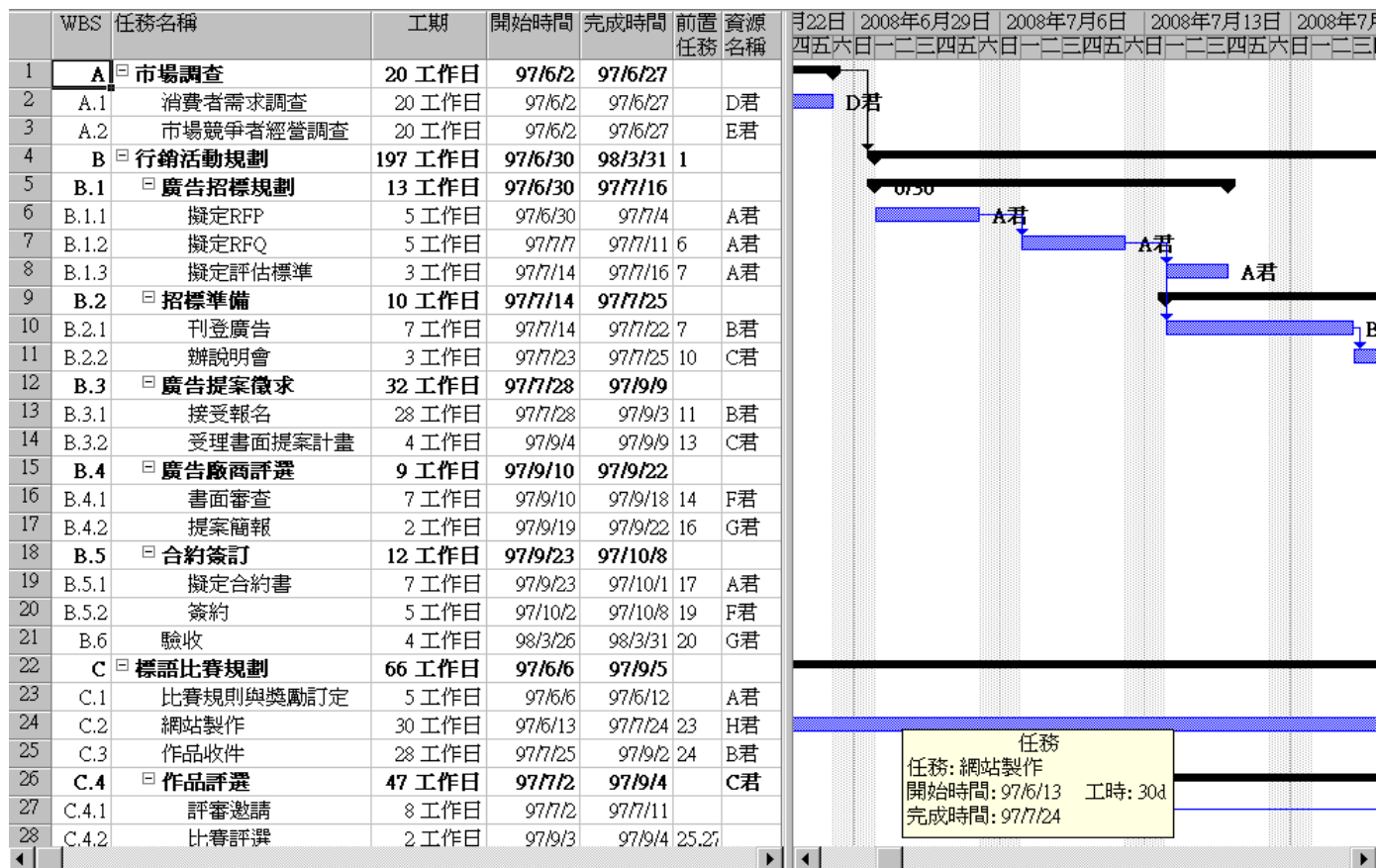
- 建立時程表
  - 針對時間管理程序的產出綜合整理
  - 可以看出專案的起始與結束時間
  - 通常會使用甘特圖作為呈現專案時間資訊的工具
  - 使用要徑分析的技術，以分析及監控專案的時程

## 5

## 甘特圖

- 標準的專案時程資訊呈現格式
  - 標示各項活動的起始與結束日期，並以日曆的格式呈現出來

- 甘特圖通常不會表達出活動之間的依存性



## 5

# 甘特圖里程碑

- 專案里程碑
  - 為專案中相當重要的事件，用以確保在特定的時程專案有完成必要的產出
- 可採用SMART標準來定義里程碑
  - 在SMART標準中，里程碑應該具備下列五項特色
    - 具體性(Specific)
    - 可測量性(Measurable)
    - 可指定性(Assignable)
    - 實際的(Realistic)
    - 具備時間架構(Time-Framed)
  - e.g. 發佈計畫beta測試版本
    - 可執行而且具備時間架構的活動

# 5

## 要徑分析方法<sub>1</sub>

- 要徑分析方法 (Critical Path Method, CPM)
  - 用來預測整體專案期間的專案網路分析技術
- 要徑
  - 指一連串的活動，藉此可以決定專案得以完成的最早時間
  - 專案網路圖中最長的一條路徑
  - 路徑中的寬裕時間或是浮動時間最少

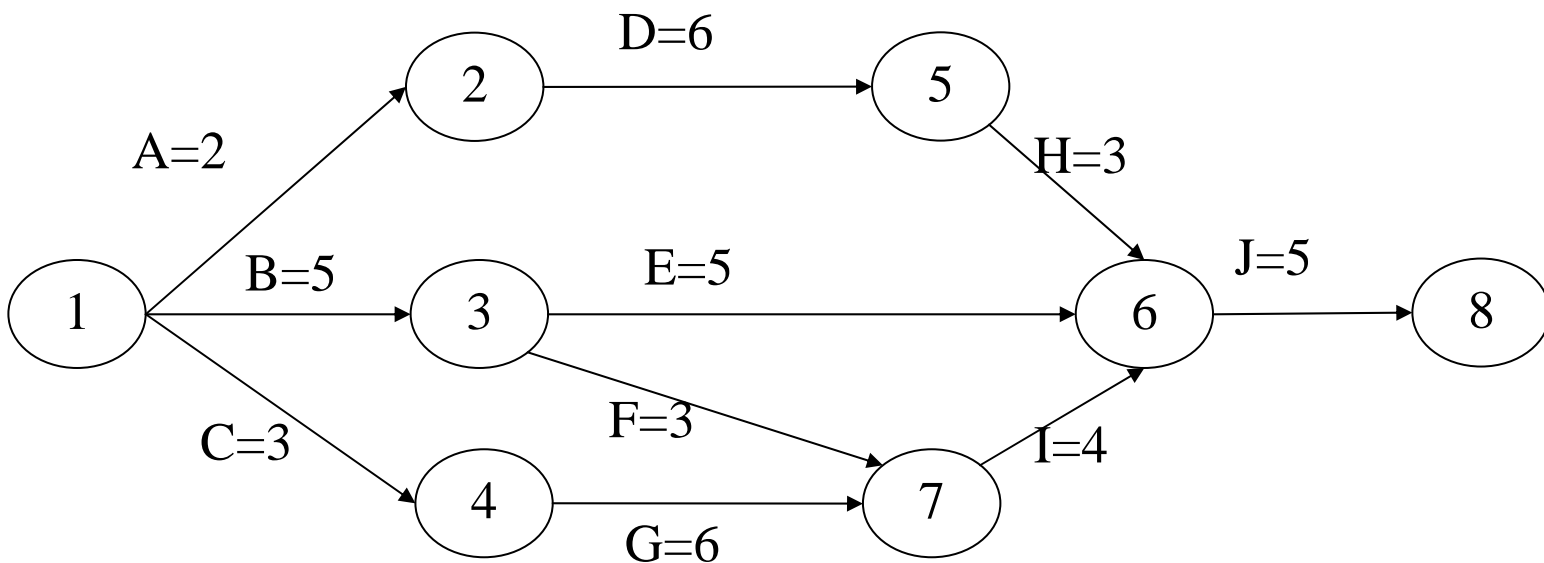


## 5

要徑分析方法<sub>2</sub>

- 浮動時間(Float)
  - 又稱為寬裕時間(Slack)
  - 活動中可能發生的延遲時間
  - 對於整體專案的完成時間並不會造成實質上延遲的影響
  - 分為
    - 總浮動時間(Total Float)
      - 不延遲專案結束的前提下，能延遲一項任務最早開始時間的時間量
    - 自由浮動時間(Free Float)
      - 在不延遲下一項任務最早開始時間的前提下，能延遲一項任務開始的時間量

# 專案要徑範例



路徑 1: A-D-H-J 長度 =  $2 + 6 + 3 + 5 = 16$ 天

路徑 2: B-E-J 長度 =  $5 + 5 + 5 = 15$ 天

路徑 3: B-F-I-J 長度 =  $5 + 3 + 4 + 5 = 17$ 天

路徑 4: C-G-I-J 長度 =  $3 + 6 + 4 + 5 = 18$ 天

所以此網路圖的要徑即為路徑4(C-G-I-J)

# 5

## 縮短專案時程的技巧

- 減少在要徑上的活動所需要的時間
  - 擠壓(Crashing)
    - 調整成本與時程的方式，在成本增加最小的情況下，將時程進行最大程度的壓縮
  - 快速追蹤(Fast Tracking)
    - 將平日可能採取順序執行的活動，改以平行執行方式取代

# 5

## 專案時程控制

- 時程變數控制技術
  - 實際時程的檢討
  - 處理人的議題
    - 專案經理控制時程變數的領導統御技巧
      - 授權、獎勵、紀律、協商



## 5

# 專案成本管理

- 工作主要包含
  - 資源規劃
    - 可以透過工作分解結構(WBS)來找尋完成該項工作所需要的資源
  - 成本預估
    - 估算專案的人力與花費
  - 成本預算
    - 整體成本分配至各個工作中，建立績效衡量的基準
  - 成本控制
    - 控制專案預算的變數，可透過盈餘管理(Earned Value Management, EVM)的方式來進行

## 5

# 成本預估

- 成本預估是專案成本管理的主要依據
- 成本預估的方法
  - 專家意見(Expert Opinion)
  - 類比(Analogy)法
  - 分解(Decomposition)法
  - PERT模型
    - 根據最壞狀況(下限推估)、最佳狀況(上限推估)以及最可能狀況進行推估
    - 工作量 = (下限推估 + 4 最可能的推估 + 上限推估) / 6
  - 數學模型(Mathematical Models)
    - COCOMO建構成本模型
    - Rayleigh曲線模型
    - Albrecht的功能點(Function Point)模型

## 5

## Albrecht功能點模型

- 根據系統早期的可視功能並配合權重的方式來建立
- 模型建構的起點為判斷在系統中的下列各項目數量：
  - 外部輸入：來自不同的應用程式之資料或是使用者的輸入，如檔案名稱及選單選取
  - 外部輸出：以不同的報表或訊息方式呈現給使用者
  - 使用者查詢：需要有回應的互動輸出入
  - 外部檔案：處理針對其他系統之所有機器可讀取之介面
  - 內部檔案：系統中的檔案

## 5

## 複雜度對應表

項目	簡單	平均	複雜
外部輸入	3	4	6
外部輸出	4	5	7
使用者查詢	3	4	6
外部檔案	7	10	15
內部檔案	5	7	10

- 未調整功能計算(Unadjusted Function Count, UFC)

$$UFC = \sum item_i w_i$$

- 針對技術複雜度因子(Technical Complexity Factor: TCF)進行考量

$$FP = UFC * TCF$$



## 5

Albrecht的技術複雜度因子  
(TCF)

- TCF的計算公式如下

$$TCF = 0.65 + 0.1 \sum_{i=1}^{14} f_i$$

- 每一個因子的範圍以0~5的尺度進行分類，0表示無關，5表示重要的

可靠的備份和還原	數據通訊
分散功能	效能
依賴設定	線上資料輸入
操作性的使用	線上更新
複雜的介面	複雜處理
重複使用性	安裝的簡易性
多台主機	功能的改變

## 5

## 範例

- 假設某個系統，其規格項目分別為
- A=外部輸入的數目=2
- B=外部輸出的數目=4
- C=查詢的數目=0
- D=外部檔案的數目=1
- E=內部檔案的數目=1
  
- 根據複雜度對應表：
- 若以平均等級的複雜度來看，可以計算出UFC：
  - $UFC = 4A + 5B + 4C + 10D + 7E = 8 + 20 + 0 + 10 + 7 = 45$
- 若以簡單等級的複雜度來看，則會得出UFC：
  - $UFC = 3A + 4B + 3C + 7D + 5E = 6 + 16 + 0 + 7 + 5 = 34$
- 若改為複雜等級的話，則可以得到UFC為65的結果。
- 一個FP需要3個工作人天的話
  - 此專案所預估需要的人天數為102到195

## 5

# COCOMO模型<sub>1</sub>

- 建構成本模型(COnstructive COst MOdel, COCOMO)模型
  - 依據原始程式碼行數或功能數的數量做為參數
  - COCOMO是Barry Boehm[Boe1997]所研發
  - 根據不同的預估詳細程度分為三個等級
    - 基本、中等及詳細
  - 會區分為三個種類的專案類型
    - 組織的(Organic)：主要為已經有相當程度了解(Well-Understood)的應用軟體系統，並且由小型的團隊進行開發
    - 半獨立的(Semi-Detached)：較為複雜的專案，且團隊成員對於發展相關系統的經驗有限
    - 嵌入的(Embedded)：複雜的專案，且所開發的軟體需高度整合軟體硬體及其存取操作程序。

## 5

COCOMO模型<sub>2</sub>

- COCOMO的基本格式是以每千行原始碼(KLOC)為基礎
- 根據三種不同種類的專案類型分別有不同的公式
  - 組織形式的專案： $\text{Effort} = 2.4 * (\text{KLOC})^{1.5}$
  - 半獨立形式的專案： $\text{Effort} = 3.0 * (\text{KLOC})^{1.12}$
  - 嵌入形式的專案： $\text{Effort} = 3.6 * (\text{KLOC})^{1.20}$
- 開發排程，以M（月）來表示，根據三種不同的專案類型
  - 組織形式的專案： $M = 2.5 * \text{Effort}^{0.38}$
  - 半獨立形式的專案： $M = 2.5 * \text{Effort}^{0.35}$
  - 嵌入形式的專案： $M = 2.5 * \text{Effort}^{0.32}$
- COCOMO模型
  - 可以快速得到對於專案成本大略的預估
  - 但基本模式(Basic)的COCOMO模型並沒有將硬體的限制、人員素質的差異、使用的工具及技術等等不同專案屬性考慮在內

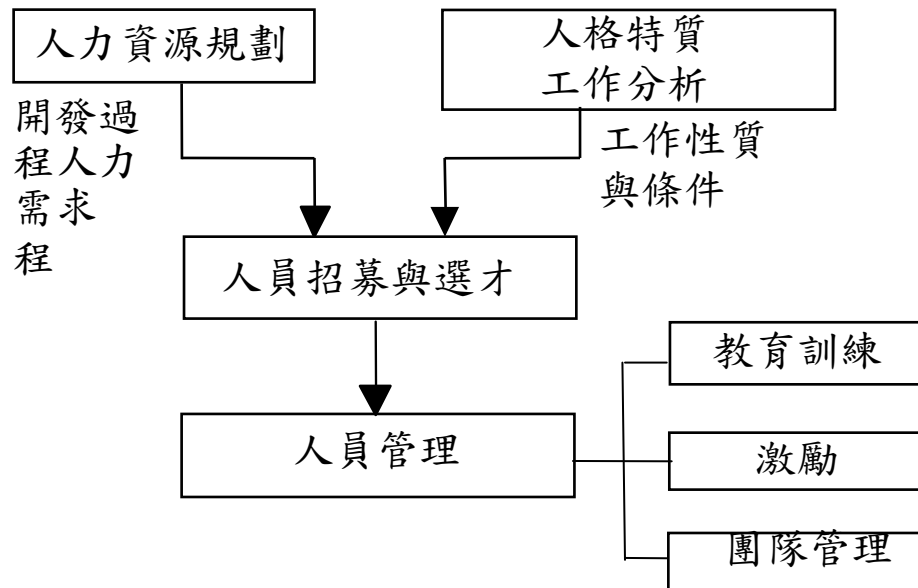
# 5

## 成本預算與控制

- 成本預算是將專案成本預估分配到個別的工作項目上
  - 工作項目則是依據專案的工作分解結構WBS所擬定
- 成本預算表基本包含
  - 預算項目
  - 預估成本
  - 說明

# 5

## 人力資源管理



- 人員管理有4個關鍵因素
  - 一視同仁
  - 尊重
  - 包容
  - 誠實

## 5

# 團隊管理

- ◆ 影響團隊工作的因素有下列幾點
  - 團隊組成
  - 團隊向心力
    - ◆ 具有向心力的團隊具有下列幾個優點
      - 可發展出團隊的品質標準
      - 成員緊密的一同工作
      - 成員知道彼此的工作內容
      - 實踐無私的程式設計方式
    - ◆ 向心力很強的團隊有時候會遇到下列兩個問題
      - 不理性地對抗領導的改變
      - 團體思考(Group Think)
  - 團隊組織
  - 團隊溝通
    - ◆ 影響有效溝通的主要因素
      - 團隊人數
      - 團隊結構
      - 團隊組成
      - 工作環境

## 5

# 專案風險管理<sub>1</sub>

- 風險
  - 潛在的問題且足以阻礙專案進行
- 有效的風險管理
  - 事先規劃一連串的程序以進行風險辨識、風險評估、風險分級等
- 與專案管理的關係
  - 專案管理應制定風險計畫，研究風險解決方法，進行風險控制活動，並且不斷地進行風險監督直到專案結束
- 專案風險管理的困難
  - 風險是針對未來的事件
  - 風險的發生及造成的損失均不確定
  - 風險一旦發生，將對專案造成重大或致命性的影響



## 5

專案風險管理<sub>2</sub>

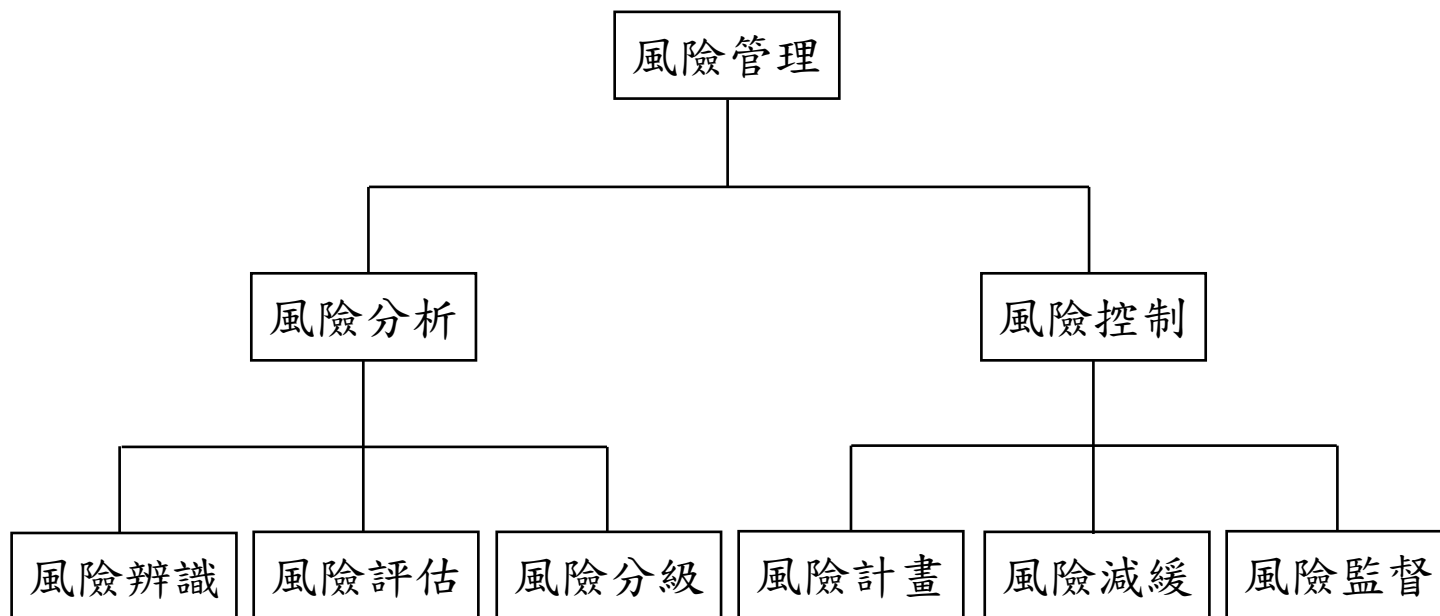
- 風險包含機率(Probability)與損失(Loss)二項變數

$$R = \sum_{i=1}^n P_i \times L_i$$

- R是風險，Pi是第i項風險發生的機率，Li是第i項風險發生的損失，n是風險數
- 螺旋式模型
  - 首次將風險分析因素加入軟體開發模式
  - 每一次的風險管理象限都會產生一個「做與不做」的決定，若專案的風險太大則必須中止軟體開發

# 5

## 風險管理架構圖



## 5

# 風險分析

- 風險辨識(Risk Identification)
  - 找出危害專案的風險事項
    - 方法有分解法、個案參照、腦力激盪、檢查表、理論、經驗等
- 風險評估
  - 估計風險項目發生的機率與風險發生後可能造成的損失
  - 影響風險評估的主觀因素包括
    - 處理風險的態度、資訊解讀的誤差、資訊易得性的偏差、資訊熟悉性的偏差、專業經驗的偏差
  - 風險評估的技術
    - 加權積分法、團體會商、德菲法、腦力激盪、面談、SWOT、核對清單、假設分析、因果圖、流程圖等
- 風險分級
  - 設定風險參考水準(Risk Referent Level) )以設定風險截斷線(Cutoff Line)
  - 高於風險參考水準的風險事項優先管理
  - 未達到風險參考水準者僅以適度的觀察，而不列入風險管理活動中
- 風險降低比 = (風險管理前之風險 - 風險管理後之風險) / 風險降低時花費成本

## 5

# 風險控制

- 風險計畫
  - 針對已納入風險管理之每一風險，擬定所採用的風險控制技術、評估成本效益並且制定風險監督計畫等管理步驟
- 風險減緩
  - 降低風險機率、減少損失幅度所採取的風險解決行動
    - 如風險預防、風險解決、風險避免、風險分散、風險移轉
- 風險監督
  - 持續性的工作，直到專案結束為止
  - 風險監督的方法
    - 監督最優先的十項風險事項
    - 里程碑監督

## 5

## 專案監控

- 專案監督與控制為比較、分析專案實際進度與專案規劃進度的差異，進而評估可能的備選方案，並且採取必要的矯正行動
- 專案監督
  - 不斷地蒐集資料、正確地分析評估以避免誤差，並且建立一合理的評估制度
  - 專案經理對專案進行監督，可能產生的問題
    - 容易使用主觀的直覺來衡量專案進度。
    - 不同背景的人員評估差異大。
    - 專案成員害怕受罰而呈報不實進度。
    - 不能即時發現問題。
    - 被視為小問題而未處理，以致延遲專案完工時間。
    - 選擇不當的衡量工具。

## 5

## 專案執行狀態

	工期超前	工期落後
預算剩餘	專案執行有效率	專案執行落後
預算超支	增加人員或加班趕工	專案出現重大問題或效率不彰

- 單從預算S曲線圖或時程甘特圖都不能完整地監督專案進度
  - 盈餘管理(Earned Value Management, EVM)，同時整合預算與時程並加以監督，以更完整地描述專案進度執行的成效

## 5

盈餘管理<sub>1</sub>

- 計畫價值 (Planned value, PV)
  - 又稱為規劃工作的預算成本 (Budgeted Cost of Scheduled Work, BCSW)，亦即經過認可預計在特定時間內花費在某項活動的總成本
- 實際成本 (Actual Cost, AC)
  - 又稱為執行工作的實際成本 (Actual Cost of Work Performed, ACWP)，亦即在某段時間內完成某項工作所產生的總直接成本與間接成本
- 盈餘 (Earned Value, EV)
  - 又稱為執行工作的預算成本 (Budgeted Cost of Work Performed, BCWP)，亦即工作的實際完成比例乘以計畫價值

## 5

盈餘管理<sub>2</sub>

- 成本差異(Cost Variance, CV) = EV - AC
  - 成本差異是用以判斷預算是否超支
    - CV值為正：預算在控制範圍內
    - CV值為負：表示預算已經超支
- 時程差異 (Schedule Variance, SV) = EV - PV
  - 時程差異是用以決定實際的工作時程與規劃的時程之差
    - SV為正：工作時程超前原訂計畫
    - SV為負：工作時程落後計畫

## 盈餘管理分析結果

SV(+), SPI(&gt;1)

SV(-), (SPI&lt;1)

CV(+), CPI(&gt;1)

工期提前、工作效率高，是最佳情況

工期落後、工作效率高，可慢慢追上進度

CV(-), CPI(&lt;1)

工期提前、預算超支，可能因為趕工緣故

工期落後、工作效率低，最糟的情況



## 5

盈餘管理<sub>3</sub>

- 成本績效指標(Cost Performed Index, CPI)
  - $CPI = EV/AC$
- 時程績效指標(Schedule Performed Index, SPI)
  - $SPI = EV/PV$
- 預估專案完成時所需要的成本(Estimate At Completion, EAC)
  - $EAC = (AC/PV) \times \text{總成本}$
- 盈餘管理可以使用以下幾種方法來衡量工作完成進度
  - 百分比法：以工作完成的百分比計算。
  - 50/50法則：一開始算50%，完成後再計算另外50%。
  - 0/100法則：100%完成後才計算。
  - 過去的經驗。
  - 設定里程碑。

## 5

專案控制<sub>1</sub>

- 專案控制的步驟
  - 設立比較基準，例如歷史資料、規範、專家顧問
  - 蒐集所需的資料，運用專案監督方法，以正確衡量專案執行進度或狀況
  - 比較實際與規劃的差異
  - 製作專案狀態報告
  - 實施修正行動
- 改善專案控制成效的作法
  - 邀請有專業經驗的人員參與
  - 確實掌握外包資源與進度
  - 研究制定簡潔、合理的控制流程與作法
  - 整合時程與預算、統一專案進度衡量標準
  - 運用決策分析技術，修正中止準則或修正行動
  - 配合使用適合的工具軟體，增進監督控制的效能

## 5

專案控制<sub>2</sub>

- 90%現象
  - 專案規劃初期無法仔細規劃所有細節及一些未考慮到小問題往往成為整合階段延誤的主因。因上述無法預期的問題，導致專案原先計劃的10%進度，最後完成的時間卻花了專案一半的時間
- 專案控制機制
  - 正式控制
    - 行為控制
    - 結果控制
  - 非正式控制
    - 群組控制
    - 自我控制

# 5

## 專案其他計畫

- 一個軟體專案的計畫與管理，除了前面的專案規劃項目外，還包含其他軟體開發重要工作事項的計畫管理
  - 建構管理計畫
  - 品質保證計畫
  - 度量分析計畫

## 5

## 本章總結

- 專案計畫的目的為發展描述專案活動的計畫書。在發展專案計畫書的過程中，需要和所有專案利害關係人適度溝通，取得承諾，才能更有效地掌握專案進度與資源。之後更需根據專案的進度與變化作必要的調整。專案的進度與變化，則需仰賴專案經理依據專案計畫書與專案監控方法進行專案的監控管理。
- 本章的重點在介紹專案管理及其工作內容，
  - 專案計畫書的撰寫，包含專案範圍、活動與時間排程、成本預估、專案資源管理、風險分析
  - 專案執行監督與控制

## 5

## 作業練習 (Exercises)

1. 請為下列專案製作工作分解結構
  - 打造一棟寵物住家
  - 新生迎新活動計畫
  - 為學校設計一套資訊系統，工作分解結構內的工作項目最少需分為三個層級，並記錄碰到的困難
2. 試著根據所執行的專案的工作分解結構繪製甘特圖
3. 請設計一個專案，使用AOA網路圖繪製該專案的活動關係，並指出圖中的所有路徑及要徑。
4. 訪問一位專案經理，請教他對於專案生命週期的管理方式及看法，並撰寫成報告。