

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 2

## 需求工程

# 2

## 大綱

- 需求的種類
- 需求工程
  - 需求擷取
  - 需求分析
  - 需求規格化
  - 需求確認
- 需求管理
- 本章總結



## 2

# 需求的種類

- 使用者需求與系統需求
  - 使用者需求來自於使用者，通常是種比較抽象的想法，以企業目的為導向，並以口語陳述為主，讓開發者可以容易了解系統建置目標。以開發線上考試系統為例，使用者需求可能表達如下：
    - 教師可以線上出題。
    - 系統可以支援選擇、填充、問答等題型的線上考試。
    - 教師可以在線上閱卷。
    - 教師閱卷後，學生可以在線上直接看到成績與正確的解答。
  - 系統需求則比較明確，屬於系統導向的思考與分析，多以文字規格呈現。系統需求會更細部地考慮各項需求。例如，
    - 線上出題是否可以使用題庫？
    - 系統是否會自動偵測配分為100分？
    - 是先分配每大題的總分、還是先分配每一小題的分數？
    - 在系統效能方面，系統是否可以同時支援100人以上的考試、效能是否會太差？
    - 以及如何確保系統安全性等問題。
  - 系統需求通常會以軟體需求規格書的方式呈現，做為雙方建立合約的基礎。使用者需求雖然較不明確，無法做為合約的基礎，但可以導引系統需求的建立。

## 2

# 需求的種類

## ● 功能性需求與非功能性需求

- 功能性需求是指具體提出系統應該提供的服務項目，如：
  - 提供以考題題目為關鍵字的搜尋。
  - 提供排序。
  - 提供統計平均的功能。
- 系統是否具備這些功能性需求是非常明確的，只有兩種可能：有與沒有，不會有程度的差異。
- 非功能性需求則是強調對於系統品質的要求與限制，或者是說系統應該具備的特性，例如，可靠度、安全性等品質性的要求。如：
  - 必須提供良好的使用介面。
  - 必須提供快速的搜尋。
  - 所建立的系統必須容易維護。
  - 可以方便快速地建立試卷。

## 2

## 需求的種類

- 非功能性需求有兩個特點。
  - 通常相依於功能性需求
    - 例如，「搜尋的速度要快」，非功能性需求「快」相依於「搜尋」功能，如果「搜尋」都沒有做到，就無法進一步談到搜尋速度要快。
  - 通常是有程度的滿足，不像功能性需求那樣地明確。
    - 例如，我們如何定義「快速」？如何定義「容易維護」？如何定義「良好的使用介面」？
- 非功能性需求從需求提出者來看，可分為「使用者」與「開發者」兩類。
  - 使用者所關心的是
    - 系統的正确性、效率性及可靠度等；
  - 而開發者所關心的是
    - 系統的可維護性、可測試性、彈性及再使用性等，因為這些特性關係著系統是否易於開發及維護。

## 2

## 軟體品質

- 軟體品質可以分為以下三類
  - 產品操作
    - 正確性：系統的執行結果是否正確，是否符合需求。
    - 可靠性：系統是否總是正確地執行。
    - 效率性：系統是否有效率地執行。
    - 整合性：系統是否有正確地整合。
    - 可使用性：系統的使用者是否可以容易地使用該系統。
  - 產品開發
    - 可維護性：系統是否易於維護。
    - 可測試性：系統是否可測試，以確認其沒有錯誤，並符合規格。
    - 彈性：系統是否容易修改擴充。
  - 產品移交
    - 可移植性：系統是否容易移植到另一個系統。
    - 在利用性：在開發系統時，是否容易引用先前開發的模組。
    - 互助運作性：系統是否容易與其他系統互相整合運作。

## 2

## 需求工程

- 需求工程(Requirements Engineering) 主要是指透過一連串有系統、有步驟的方式，將待開發的系統需求從需求端擷取出來，並加以分析及建置，以供開發端依此開發。
- 需求建置的步驟，主要包含以下活動：
  - 需求擷取(Requirements Elicitation)、
  - 需求分析(Requirements Analysis)、
  - 需求規格化(Requirements Specification) 與
  - 需求確認(Requirements Validation)。

## 2

## 需求擷取

- 需求擷取的目的是獲得使用者對於系統的需求，其方法有面談、問卷、使用者觀察、研討會、腦力激盪、使用案例與雛形法。
  - 面談
    - 最直接取得需求的方法，系統分析師與需求單位負責人進行面對面對談，需求單位將使用者對系統的期望、限制與介面需求以口述的方式告訴系統分析師，以便能記錄需求，並得以面對面的與需求單位討論不清楚的部分，藉以釐清系統的功能與限制。面談的好處是可以直接溝通，對於初步了解系統需求是相當方便的。
  - 問卷
    - 因為每次的面談都相當耗時，除了面談的時間外，面談前的議程溝通通常也相當耗費時間，因此面談的次數通常不會太多。面談所獲得的需求僅能代表面談者的需求，無法反映多數使用者的需求。這個問題可以透過問卷的方式了解多數人對目前系統的觀感，對新系統的期望，同時可以獲得更客觀的需求。

## 2

# 需求擷取

## ● 使用者觀察

- 觀察使用者或需求單位目前的作業狀況。例如，開發校務系統，可以實地觀察目前校務運作的流程，了解整個校務的領域知識與作業流程後，定義出系統的需求。透過使用者觀察可以客觀、明確地了解需求單位現階段的作業方式，避免面談時需求端的口誤或分析師的誤解。使用者觀察的缺點是需要較長時間的觀察與分析，否則所觀察的作業流程可能僅是片面而不正確。

## ● 研討會

- 當系統的使用者來自不同的組織或單位，需要共同協商討論系統的功能架構時，則可以研討會的方式進行，收集各方的需求、限制與對系統的期待，當不同的單位需求發生衝突時，可以立刻進行協商討論，降低日後因為需求錯誤所造成的成本損失。研討會應由有經驗的主持人負責主持，主持人必須對該領域知識相當了解，且具備溝通協調能力，以協調不同單位的衝突需求。研討會前應有相當的準備工作，明確地定義會議主題與議程，以提高會議效率。

## 2

# 需求擷取

## ● 腦力激盪

- 適用於創意型的系統開發，這些系統可能沒有特定的使用者或是使用者的需求並不明確。例如，手機應用系統的開發，對於多數的手機使用者而言，最初的要求僅是可以收發電話，對手機並沒有多餘的要求。手機廠商透過腦力激盪，不斷地開發出記事本、股票查詢、天氣查詢、火車訂票等功能。

## ● 使用案例

- 透過情境思考，站在使用者操作系統的角度，思考系統該具備怎樣的機能，進而引領需求的開發與分析。任何系統都不會單獨存在，在大部分的情形下，系統是設計給人用的，而人「使用」系統的「案例」（或是情境），就稱為使用案例(Use Case)。使用案例在整個系統發展中占了極重要的角色，它不僅可以在需求階段幫忙擷取與管理需求，同時也是分析、設計、實作，甚至是測試階段的依據。

## 2

# 需求擷取

## 雛形法

- 系統開發端透過初步的雛形系統展示，以視覺化且直覺的方式來擷取使用者的需求。開發端先寫出系統雛形，展示給使用者看，使用者透過「修改建議」的方式來建構他對系統的需求。這種方式可以降低需求描述的門檻，也可避免許多口語或文字的模糊。
- 雛形法的應用可以區分成兩種類型：
  - 捨棄式雛形法(Throwaway Prototyping)；
    - 系統發展之前期，開發端以其所認知的系統需求，快速地發展出一套可執行的雛形系統，以便於讓需求端可以立即審視該系統需求是否與其認知有所誤差，需求一經確認，此雛形系統便不再使用。
    - 捨棄式雛形法的特色是開發端發展的雛形系統將會針對其較沒有把握的系統需求而設計，目的是用來與需求端做進一步的釐清需求之用。
  - 漸進式雛形法(Evolutionary Prototyping)。
    - 開發端會從最有把握且最為明確的需求開始發展雛形系統，此系統經與需求端確認無誤後，再進而擴增到其他的功能需求，如此一來，雛形系統便能隨著系統的開發演進而更加的完備，最後成為要遞交給需求端的成品。

## 2

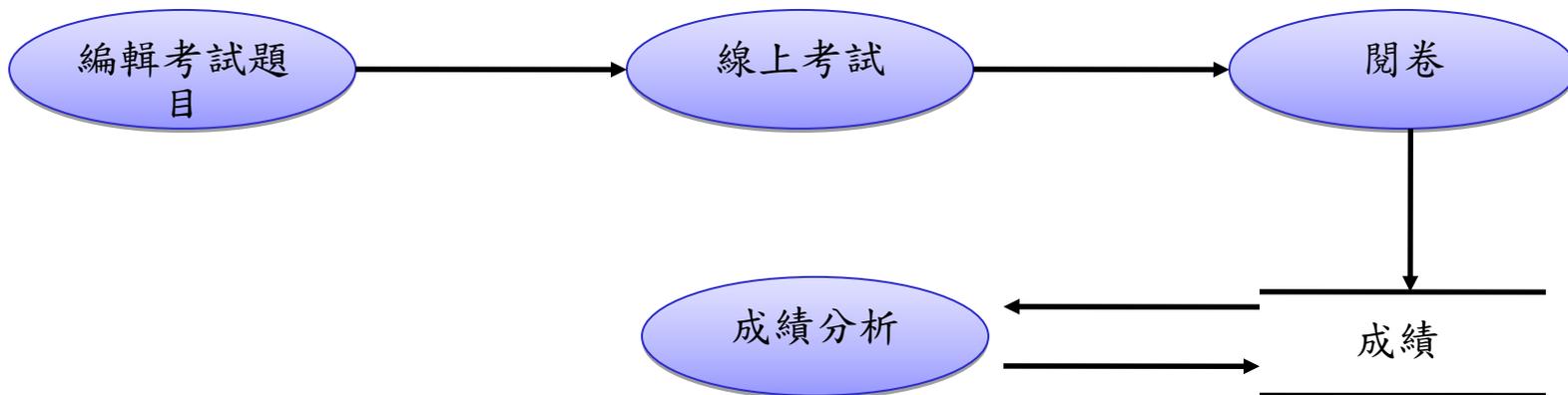
# 需求分析

- 需求分析的主要目的在分析需求是否正確、完整、沒有衝突。
  - 需求是否正確
    - 主要檢查所提的需求是否可行、是否符合企業目標、是否滿足多數期望等。
  - 需求是否完整
    - 檢查需求端所提出的需求是否完整，有沒有遺漏的地方？
  - 衝突性需求分析
    - 主要考慮需求之間是否有相互矛盾、衝突的地方，發生衝突時應該如何解決？
- 系統化的分析
  - 資料流程的分析、實體關連的分析、狀態行為的分析、企業流程的分析等。

## 2

## 資料流程分析

- 資料流程分析是以系統功能面的角度切入，藉由描述該系統之資料於各個功能程序之間的轉變情況，來剖析系統所具備的各項功能，並利用資料流程圖(Data Flow Diagram, DFD)來進行資料流程的分析與記錄。
- 資料流程圖是用來分析資料在系統中如何運算處理，例如，在線上考試系統中，其資料流程圖如下：



## 2

## 實體關連分析

- 實體關連分析是以系統資料面的角度切入，描述系統具有哪些實體資料(Entity)，以及各個實體資料之間的關連性、結構性。透過實體關連分析圖(Entity Relationship Diagram)，可以幫助釐清系統資料彼此之間的複雜關係，有助於後續的資料庫設計。
  - 實體在語言的文法上通常都是以名詞來表示，代表資料、觀念、事物。
  - 關係在文法上通常都是以動詞來呈現。
  - 實體本身具有抽象化的概念，我們可以將實體區分成實體型態(Entity Type)以及實體實例(Entity Instance)。
    - 例如：軟體工程系統分析題庫便是題庫實體型態的一個實例。

## 2

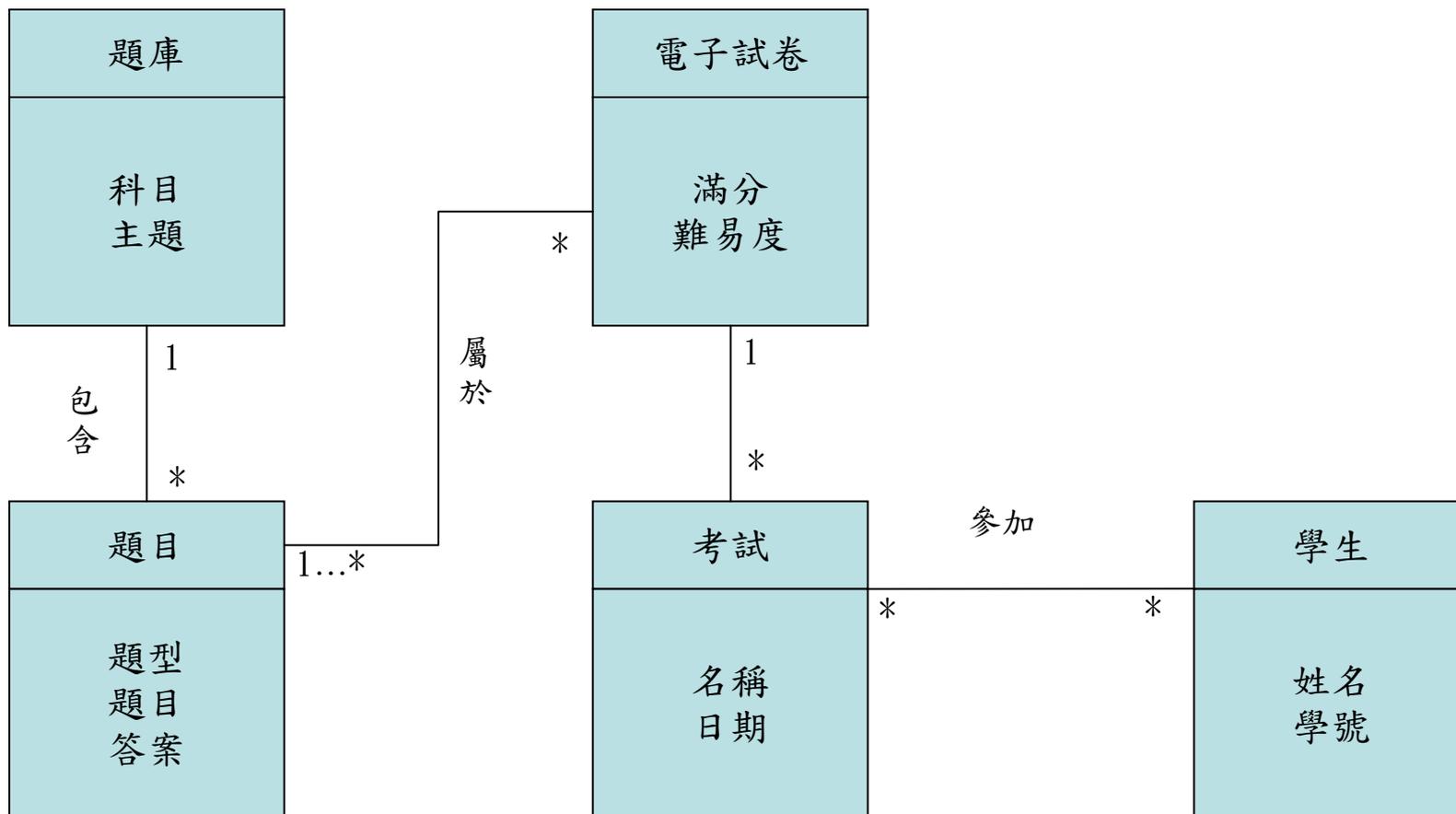
## 實體關連分析

- 實體內有許多的屬性(Attribute)，透過這些屬性的描述來表示該實體。例如，
  - ✦ 題庫的屬性有科目（如：軟體工程）與主題（如：系統分析），用以表示該題庫的範圍；
  - ✦ 又如每一場考試應該有其考試名稱（如：期中考）和考試時間，所以我們可以將「考試」當成一個實體，而將「考試名稱」與「考試時間」當成其屬性。
  - ✦ 實體本身可以透過關鍵屬性(Key Attribute)來描繪特定的實體，透過給定實體型態中關鍵屬性的值，可以判斷出其所指為哪一個實體實例。
    - ✦ 關鍵屬性其具備有唯一性(Unique)以及不變性(Unchanging)的特性。

## 2

## 實體關連分析圖

## ● 線上考試系統實體關連分析圖



## 2

## 擴充實體關連分析圖

- 擴充實體關連分析圖(Extended Entity Relationship Diagram)，其針對傳統的實體關連分析圖進行擴充，主要目的是要能夠透過實體關連分析將資料庫裡實體的邏輯內容定義出來。
- 擴充的地方主要分成兩個部分：
  - 針對每個實體的屬性進行擴充
    - 當定義屬性時，需要再針對屬性的主要鍵值(Primary Key)、外部索引鍵(Foreign Key)、屬性資料型態(Type)、寬度(Width)以及可否允許空值(Null OK)進行描述。

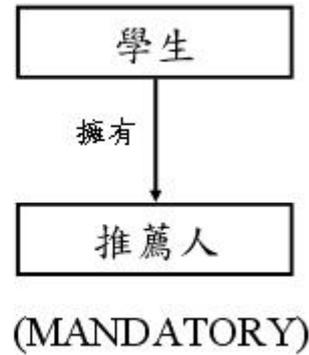
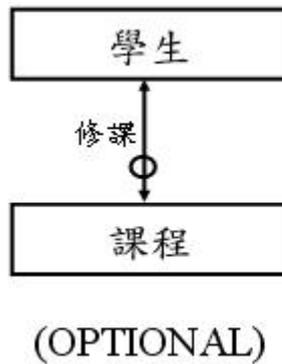
學生實體屬性定義

ATTRIBUTE	PRIMARY KEY	FOREIGN KEY	NULL OK	TYPE	WIDTH
STUDENT_ID	X			ID	9
SURNAME			X	CHAR	20
NAME				CHAR	20
PROGRAM_ID		X		ID	9

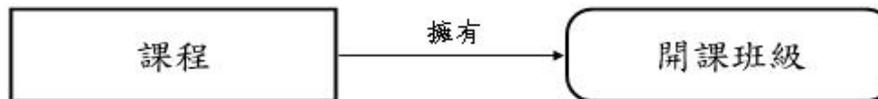
## 2

## 擴充實體關連分析圖

- 針對實體彼此間關係的意義進行擴充。
  - 當描述實體與實體之間的關係時，新增加了選擇性關係(Optionality)、生存相依關係(Existence Dependency)以及抽象關係(Abstraction)此三種關係。
  - 選擇性關係範例：下圖左表示學生可能可以不修課。而右圖的例子於是解讀成，學生必須擁有推薦人。



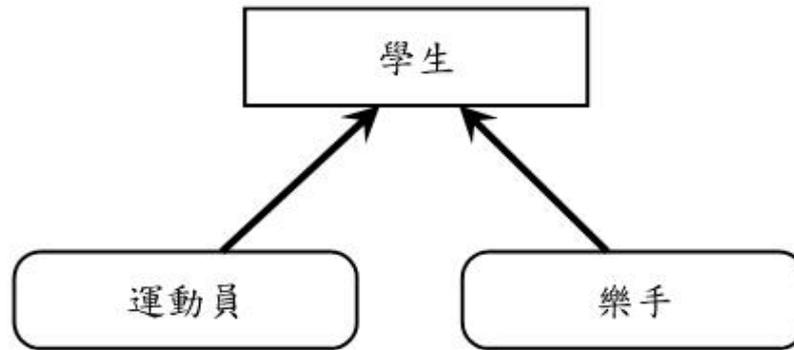
- 生存相依關係範例：下圖表示課程實例會擁有許多開課班級，而每個開課班級的實例之存在是必須依賴於此課程實例的存在與否。



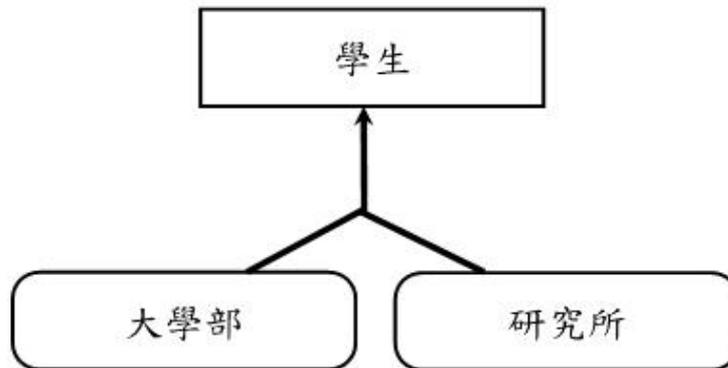
## 2

## 擴充實體關連分析圖

- 抽象關係之重疊關係範例：下圖解讀成一位學生可以是運動員，也可以同時是樂手的身分。



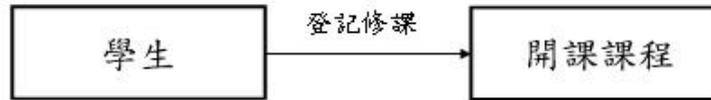
- 抽象關係之互斥關係範例：下圖解讀成一位學生，其不是大學部學生就是研究所學生。



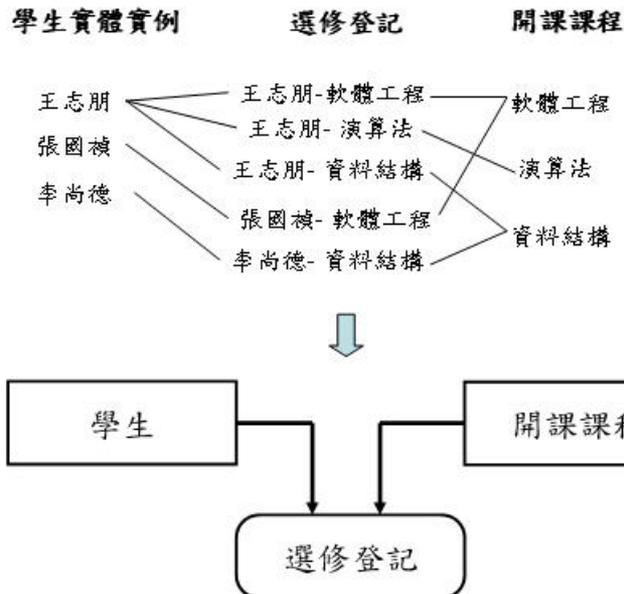
## 2

## 擴充實體關連分析圖

- 以相交實體來針對多對多的關係加以精煉
  - 多對多實體關係存在著無法表達特定實體實例之間的對應關係之問題。



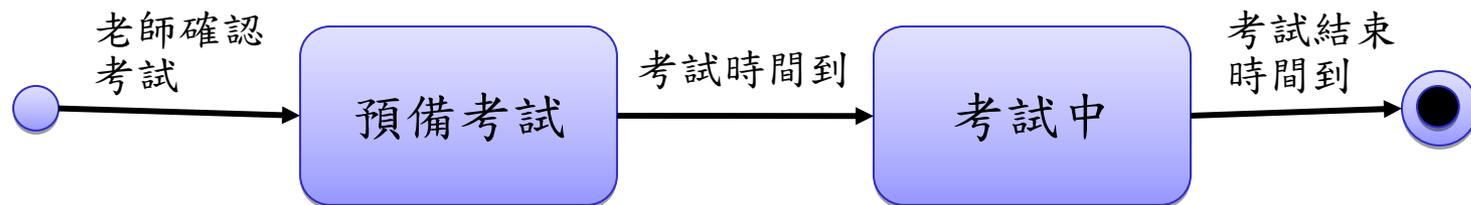
- 系統分析人員可以利用相交實體(Intersection Entity)來負責記錄在多對多的實體關係下，特定實體實例之間的對應關係。



## 2

# 狀態行為分析

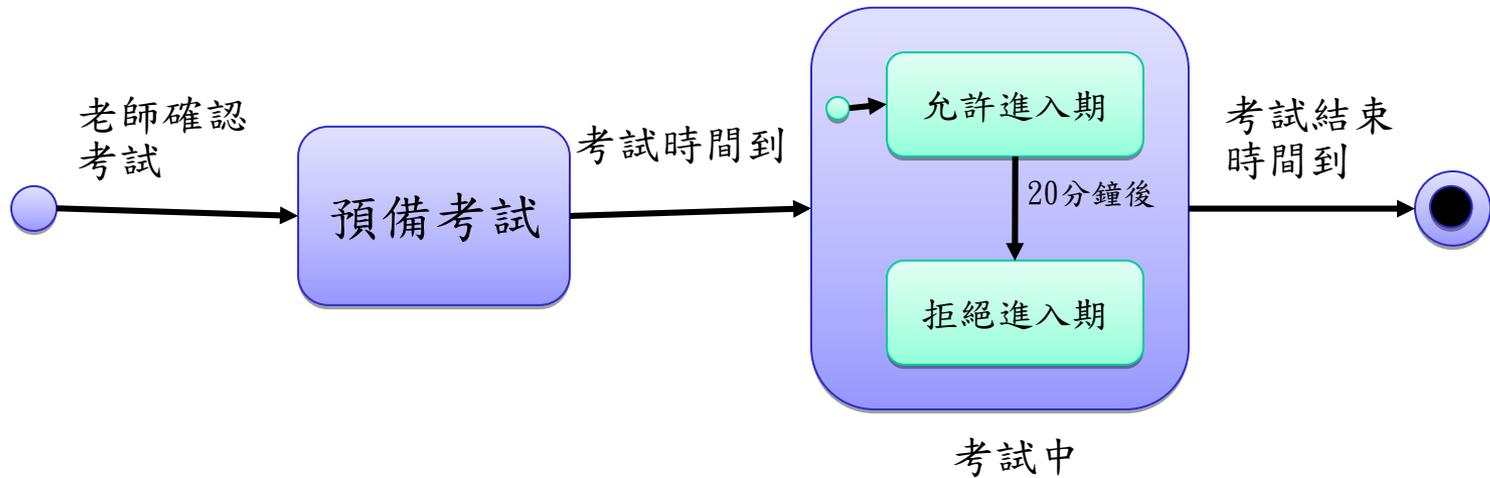
- 狀態行為分析是以系統行為面(System Behavior)之角度切入，藉由描述系統運作時其所處之狀態如何因應外界事件發生而導致系統狀態之轉變，進而產生相關之反應動作，以此來剖析系統之運作行為，此分析方法將產出系統之狀態轉換圖(State Transition Diagram)。
  - 例如，系統的行為可以用「受到某事件的刺激後，進而做出某反應」來表達，而其中的「反應」又可以狀態轉換以及轉換狀態後會執行何種反應動作來表達。
    - 例圖為線上考試系統簡易狀態轉換圖



# 2

## 狀態行為分析

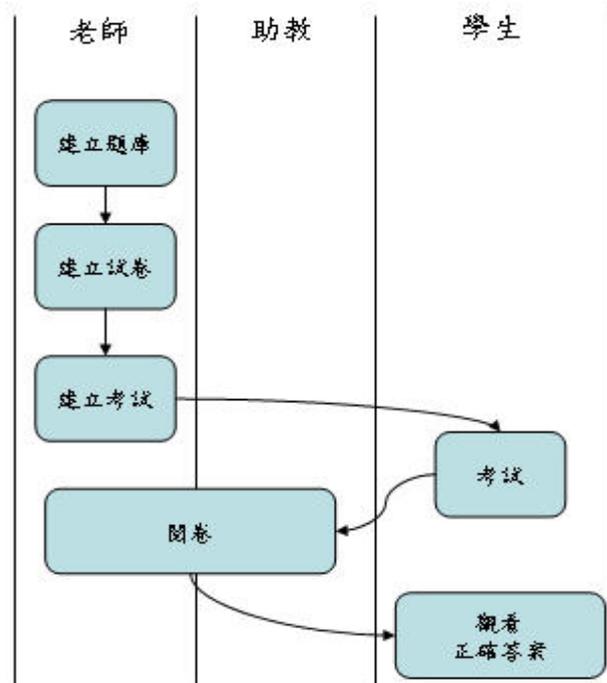
- 例圖為線上考試系統複合狀態轉換圖



## 2

## 企業流程分析

- 企業流程分析在於分析系統的作業流程，可用企業流程模組 (Business Process Model) 表示。
  - 這裡的「企業」並非只能表達企業的流程，而是強調企業通常有許多不同的組織，所以流程的表達應該考慮各種不同的組織或角色。
  - 例圖為線上考試系統的企業流程模組



## 2

## 需求規格化

- 需求擷取所獲得的需求是屬於使用者需求，有可能不明確、不完整，或相互矛盾，透過需求分析後，將使用者需求轉換為系統需求。接下來將進行需求規格化，寫成具體的文件，以做為後續設計的依據，也做為雙方的合約基礎。
  - 下表為線上考試系統之軟體需求規格書範例

1. 簡介	4. 試卷管理子系統
2. 整體性描述	4.1 外部介面需求
3. 題庫管理子系統	4.2 功能需求
3.1 外部介面需求	.....
3.2 功能需求	5. 考試管理子系統
3.3 效能需求	.....
3.4 設計限制	6. 閱卷管理子系統
3.5 軟體系統特性	7. 成績分析子系統

## 2

## 軟體需求規格書格式範例

- 依據IEEE 830建議的軟體需求規格，軟體需求規格書應包含如下表之項目：

軟體需求規格格式	說明
1. 簡介	
1.1 系統目的	- 系統建置的目的、主要使用者。
1.2 系統範圍	- 概述包含的功能與不包含的功能。
1.3 名詞解釋	- 包含縮寫等名詞解釋。
1.4 參考	- 說明此文件的參考資料。
1.5 文件概述	- 說明本 SRS 的架構與內容。
2. 整體性描述	- 說明系統的環境、與其他系統的介面。若系統僅是產品的一部分，則需說明它們之間的關係、介面。其他的介面包含使用者介面與硬體介面等。
2.1 系統環境	
2.1.1 系統介面	- 簡略敘述系統主要的功能。
2.1.2 使用者介面	- 描述使用的特性與使用系統的目的。
2.1.3 硬體介面	- 包含法令、搭配的軟硬體限制、可靠度等限制。
2.1.4 軟體介面	
2.2 產品功能	- 描述系統的假設。當假設不成立時，此文件可能需要修改。
2.3 使用者特性	- 條列與外部界接的介面限制。
2.4 限制	- 條列系統應提供的功能。
2.5 假設	- 條列系統應滿足的非功能性需求。
3. 細部功能說明	- 條列設計限制，以供後續設計參考。
3.1 外部介面需求	- 條列此軟體系統的特性。
3.2 功能需求	
3.3 效能需求	
3.4 設計限制	
3.5 軟體系統特性	
附錄	

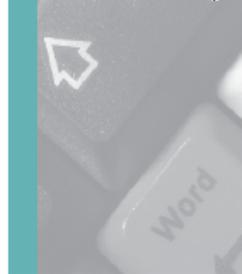
## 2

## 需求確認

- 需求確認(Requirements Validation)主要在確認所定義的需求規格是否正確，是否符合使用者的本意。另一個重點是確認系統是否為可驗證(Verifiable)的，這會影響到日後的驗收測試。
  - 何謂可驗證之需求？
    - 假設規格書中提到像「系統必須容易操作」這樣的需求，顯然是無法驗證的，我們必須進一步的建立它的驗證方法，例如，「系統必須容易操作」可以分解成幾個子項：
      - 老師容易建立試卷。
      - 老師可以用拖曳的方式，將題庫的題目直接拉到試卷中建立試卷。
      - 學生容易作答。
      - 針對選擇題，學生可以直接點選選項，不需以鍵盤輸入選項代碼。
    - 上述之子項的功能很明確，因此可以直接地測試而無須做進一步地分解。
    - 又如：「系統必須防止學生作弊」的需求過於抽象而無法驗證，需要再進一步地加以分解。
      - 則「**建立測試案例**」是很好的需求確認技巧，用以確認需求的可驗證性。

## 2

## 需求確認

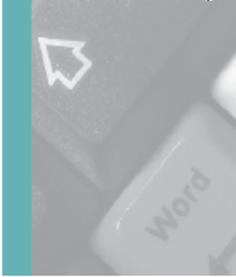


### ● 規格書審查

- 透過軟體需求規格書審查來進行需求確認，是最直接、也最方便的方式。其流程如下：
  - 完成軟體需求規格書後，商討規格書審查的時間及參與審查的同仁及專家。
  - 於審查前將規格書送給審查人員。
  - 審查人員審查規格書，並對疑問之處加上注解。
  - 審查會議召開，由文件的建立者逐項報告需求內容，由審查人員進行討論。
  - 會議結束前，由主席或報告者總結需要釐清或修改的部分，並決定是否進行再一次的審查。
- 審查會議參與的角色包含：
  - 主席：控制會議的進度、決定會議進行的方式、引導問題的討論。
  - 記錄員：避免讓報告者同時記錄要修改的內容，以免延誤會議的進行。
  - 審查人員：對該領域或系統熟悉的專家。
  - 報告者：需求規格書的報告者，通常也是撰寫者。

## 2

## 需求確認



- 雛形展示

- 如果是採用雛形法開發方式，則可以用雛形展示來確認系統。透過直接的系統展示，使用者可以很明確的知道開發端的方向是否正確。

- 需求確認之重要性

- 許多單位常常會忽略需求確認的動作，習慣一開始就立即進入程式撰寫，等到系統開發出來後，才驚覺不是使用者所要的，於是需要再花很長的時間改寫。需求確認雖然一開始會花一些時間，但對維持整體進度而言是相當有成效的。

## 2

## 需求管理

- 即使我們將需求分析做的很徹底，使用者還是會經常在開發的過程中提出變更。開發端不能毫無節制地接受需求變更，也不能一味地拒絕變更。只要能制訂及遵循變更的程序，做好需求變更的管理，便可產生雙贏的局面。變更分析並不是件容易的事情。當系統需求、設計及程式碼很多、很複雜時，開發端如何知道一個需求的變更會影響哪些項目？為能有效率地進行變更分析，通常都會建立需求追溯表 (Requirements Traceability Matrix, RTM)，以表達需求與其他項目之間的關係。
  - 需求追溯可以分為：
    - 水平追溯(Horizontal Traceability)：記錄需求與需求之間的關係；
    - 垂直追溯(Vertical Traceability)：追溯需求的來源以及它相對應的設計與實作。

## 2

## 建立需求追溯表

- 例如，系統需求「可拖曳的題庫題目」（需求編號R102-2），其來源來自使用者需求「方便快捷的建立試卷」（需求編號U203），而其設計描述於設計文件SDD之2.3.1小節，實作於模組DraggableItem.java，使用測試案例TC101測試，則需求追溯表可建立如下表：

需求編號	使用者需求	設計	實作	測試
R102-2	U203	SDD 2.3.1	DraggableItem.java	TC101
R102-3	U205	SDD 3.2	Test.java	TC304
.....	.....	.....	.....	.....

## 2

## 本章總結

- 需求工程是一項重要的工作：即便我們的設計能力很好、產品的功能很多，如果做出使用者覺得不實用或不好用的產品，仍算是一個失敗的軟體工程。為此，我們需要知道：
  - 需求的特色與分類
    - 使用者需求與系統需求。
    - 功能性與非功能性。
  - 需求擷取方法
  - 需求分析方法
  - 需求驗證的方法與需求管理的方法。
- 需求分析與驗證可視為一種「早期除錯」(Early Debugging)
  - 減少後期重工的可能性。
- 唯一不變的，是事事會變
  - 一套需求管控的流程與機制，可以不讓需求的變更，混亂我們的系統設計、延宕專案的進度、及降低專案的實質利潤。

## 2

作業練習 (Exercises)<sub>1</sub>

1. 考慮一個線上購物系統，列出五個功能性需求與非功能性需求。
2. 如同題1，請將非功能性需求分解成可以測試的需求。
3. 需求工程包含哪些活動，其一般的流程為何？
4. 以線上購物系統為例，說明使用者需求與系統需求之不同。
5. 當我們進行需求分析時，應該分析哪些項目？

## 2

作業練習 (Exercises)<sub>2</sub>

6. 以線上購物系統為例，分析其企業流程模組，必要時做出你的假設。
7. 以線上購物系統為例，分析其實體關連圖，必要時做出你的假設。
8. 以線上購物系統為例，分析其狀態轉移圖，必要時做出你的假設。
9. 以線上購物系統為例，分析其資料流程圖，必要時做出你的假設。
10. 需求變更的管理流程為何？