

planning

Identifying system project  
Initiating system project

Analysis

System Requirement analysis  
System Requirement structuring

System  
├── data requirement  
├── process/functional  
└── Interface

ASP (Application Service Provider)

透過 IE 下載軟體到本地電腦安裝, 執行操作。

SaaS (Software As A Service)

透過 browser 到雲端直接操作並將 data 存在雲端

雛形法。

漸進式 (Incremental Model)

第一子系統的分析、設計、編碼、測試

↓  
操作前編譯重複

↓  
新子系統的分析、設計、編碼、測試

代 = 不伴大量投資, 初期的成功

← 整合新子系統至現有系統

CSP

1. mission statement

2. objective

3. competition

公司做事會透過 SWAP 做外在環境分析

ex. 藍海 - 冰池磨大魚

紅海 - 右池磨小魚

→ 分析 competitor

→ 行銷

ISP

(資訊部門) 要完全的分析資訊, 小是地計劃專案

CMMI (Capability Maturity Model Integrated) 能力成熟度模型

1. 最初步 (Initial) 開發流程混亂, 不可預測

2. 可重複的 (Repeatable) 流程被清楚化且有頻繁的互動

3. 定義層 (Defined) 軟體開發流程已正式化、標準化、清楚定義

4. 量化管理層 (Quantitatively Managed) 量度、量化、管理

5. 最佳化 (Optimizing) 創新構思及技術引導, 改善作業流程

SDLC = <sup>SDLC model</sup> <sup>→ JAD</sup>

專案確認和選擇 = 系統專案的優先順序; 相關系統的規劃; 活動所形成之資料網路  
 ↳ 專案是叫作兩規劃 = 專案的詳細步驟或工作計畫; 系統範圍內高階系統需求或  
 分析的規格; 指派團隊成員與其他資源; 系統修正或企業新  
 不可修改  
 ↳ 分析: 敘述現有系統並提出問題或凡會所在, 建議如何兩年決加以改善替換  
 現有系統; 解釋替代系統而選擇的修正方案  
 ↳ 設計, logic = 所有系統元件功能性; 詳細規格(資料、程序、輸入、輸出)  
 責任: =  
 ↳ 製作: 程式碼文件、訓練課程及支援能力  
 ↳ 維護: 新版本的取件、更新文件、訓練支援皆已更新

RAD (rapid application development) = 快速應用程式發展  
 需求規劃  
 ↳ 使用省設計  
 ↳ 架構  
 ↳ 結束

利用原型形法

extreme, RAD, JAD  
 ↳ 快速, 但只適用於小系統 or  
 造成很多問題, 造成許多小組去  
 估, 需較多人力

JAD (Joint Application Development) 聯合應用設計  
 主要目的: 為了由參與系統的重要人員同時蒐集到所有系統需求  
 (in 分析)

- 5 step - 1° 選定範圍  
 2° 設計 JAD 會議 → 問題存開放式  
 3° 準備 JAD 會議 → 問題結構 → 由上而下  
 4° 進行  
 5° JAD 後評價

原型形法 (System Prototyping) → 透過系統雛形使 user 了解系統 ex: 一對多

跟 SDLC 比較簡略的過去, 是開發流程  
 重複責任化 → 給 user 看 → 修改這一步  
 又適用於小系統

漸進式  
 一份制成本系統, 對系統  
 執行一部分功能

缺乏深入研及開發的工具

正规化 (Normalization Theory) → 资料分割的方法

目的: 避免资料重复, 相互矛盾, 相对来说, 查询时 pay 增加

2/3 1NF, 2NF, 3NF, 4NF, 5NF → 以 (树) 位相依性为依归分割  
Step.

未正规化的关联

↓ → 删除重复的资料

1NF = 所有记录中的项目都是基本值, 即删除重复资料

↓ → 删除部分功能相依性

2NF = 满足 1NF, 且单一非键属性完全相依于主 key

↓ → 删除传递相依性

3NF = 满足 2NF, 且单一非键属性非传递相依于主 key

树位相依: 乙树位必须搭配甲树位有意义 → 乙树位相依于甲树位

1NF 第一阶正规化 (First Normal Form)

role

1° table 内有主 key, 其他所有树位都相依于主 key

2° 单一树位只存单一值

3° table 内没有意义相同的多树位

→ 未符合 1NF 缺 → 1° 造成储存空间浪费

2° 降低存取效率

2NF

role

1° 符合 1NF

↑ 主 key 是多个树位时发生

2° 各树位而主 key 的没有部分相依 → 造成资料重复出现

→ 删除部分相依, 把部分相依的树位分割成另外的 table

3NF

role

1° 符合 2NF

↑ 树位并非直接相依, 而是借助于树位

↑ 要找各树位而主 key 的间接相依性, → table 中有没有

2° 各树位而主 key 没有间接相依 "而主 key 有关的相依性"

→ 删除间接相依 = 把间接相依的树位分割成另外的 table