



東海大學

資訊工程碩士論文

利用 Web Engineering 之 Modeling 加速更新效能

Use Web Engineering model to enhance the updating of  
web content efficiently

研究生：黃春木

指導教授：黃育仁 教授

朱延平 教授

中華民國一〇五年六月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 黃春木 所提之論文

利用 Web Engineering 之 Modeling 加速  
更新效能

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召集人

羅秉章

簽章

委

員

黃育仁

王仁澤

朱延平

指導教授

黃育仁

簽章

朱延平

中華民國 105 年 6 月 16 日

## 摘要

本研究是以 Web Engineering 相關理念及文獻，將其實際應用在工作實務層面，並以公司網站為本次研究的對象，依據使用者特性以及相關的網路應用系統技術來進行實作範例。以 Modeling 分析需求、設計規劃，搭配網路線上協作平台應用(Moqups)，結合敏捷開發方法(Agile Development) 實作過程，記錄步驟並加以說明。而透過數據統計的表現，確實有達到快速更新網站資訊的效能表現，足以證明透過 Web Engineering 在開發建構網站，對公司網站上的需求溝通及快速上線有著明顯助益。

關鍵字: Web Engineering, Modeling, Agile Development

## **Abstract**

Related Web Engineering ideas and literatures were practiced to this study. The subjects are web sites of businesses. The characteristics of users, and related network applications system technology are the practice model. The study recorded the steps and instructions of Modeling requirement analysis, design and plan, on-line network collaboration platform (Moqups), and the process of Agile Development. From the statistics data, It truly can speed up the effectiveness of update web site information. The study proved that developing the web sites with Web Engineering is very helpful to the communication and faster on-line.

Keyword : Web Engineering, Modeling, Agile Development

# 目次

摘要.....	1
Abstract.....	2
圖目錄.....	5
表目錄.....	7
<b>第一章 緒論.....</b>	<b>8</b>
第一節 研究背景與動機.....	8
第二節 研究目的.....	9
第三節 研究方式.....	9
<b>第二章 文獻探討.....</b>	<b>10</b>
第一節 網站工程.....	11
第二節 軟體工程.....	15
2-2.1 瀑布模型(Waterfall Model).....	17
2-2.2 快速原型模型(Rapid Prototype Model).....	18
2-2.3 敏捷式開發(Agile Development).....	20
第三節 UML 建模語言.....	23
2-3.1 循序圖(Sequence Diagram).....	23
2-3.1 類別圖(Class Diagram).....	24
2-3.1 案例圖(Use Case Diagram).....	25
第四節 網站的應用.....	27
<b>第三章 研究方向.....</b>	<b>29</b>
第一節 研究方法.....	29
第二節 研究限制.....	32
第三節 研究與實作.....	33
3-2.1 重置廣告區塊.....	33
3-2.2 更新流程圖.....	34
3-2.3 使用者案例分析.....	36

3-2.4 協作平台應用 Moqups.....	42
<b>第四章 實作結果 .....</b>	<b>55</b>
第一節 效能表現.....	55
第二節 協作平台表現.....	58
<b>第五章 結論.....</b>	<b>59</b>
第一節 研究貢獻.....	59
第二節 研究建議.....	59
<b>參考文獻 .....</b>	<b>60</b>
<b>附件一 .....</b>	<b>63</b>
<b>附件二.....</b>	<b>64</b>
<b>附件三.....</b>	<b>65</b>

## 圖目錄

圖 1: 網站基礎開發流程(Development process for Web-based systems)[11].....	11
圖 2: Standish Group 2013 統計圖 .....	13
圖 3: 軟體工程的四個影響[28] .....	15
圖 4: 瀑布模型 .....	18
圖 5: 快速原型模型 .....	19
圖 6: 敏捷開發方法 .....	21
圖 7: 敏捷開發方法(Agile Development).....	22
圖 8: 循序示意圖 .....	24
圖 9: Class Diagram 示意圖 .....	25
圖 10: Use Case Diagram 示意圖 .....	26
圖 11: Development 核心精神 .....	28
圖 12: Frameset Design.....	30
圖 13: 重置廣告區塊.....	31
圖 14: 工作流程.....	32
圖 15: 廣告區塊 CSS 碼.....	34
圖 16: 工作流程圖.....	35
圖 17: The Web Engineering Process .....	36
圖 18: Use Case Diagram.....	37
圖 19: 主管 Use Case.....	39
圖 20: 客服 Use Case.....	40
圖 21: 設計人員 Use Case.....	42
圖 22: Moqups 網站.....	43
圖 23: 廣告初稿.....	44
圖 24: Moqups 協作平台 .....	45
圖 25: 版面設定.....	46

圖 26: 載入圖檔.....	47
圖 27: 圖檔分享.....	47
圖 28: 分享設定.....	48
圖 29: 分享邀請.....	49
圖 30: 輸入修改意見.....	50
圖 31: 修改重製.....	51
圖 32: 修改上傳.....	52
圖 33: 主管確認畫面.....	53
圖 34: 確認訊息.....	53
圖 35: 未規劃前工作進度表.....	56
圖 36 規劃後工作進度表.....	56
圖 37 效能表現圖表.....	57
圖 38 協作平台上修改記錄.....	58



## 表目錄

表 1: 軟體工程四大影響因素.....	14
表 2: 軟體開發架構.....	15
表 3: 軟體過程模型特點.....	16
表 4: 網站類型.....	28
表 5: 效能表現數據.....	57

# 第一章 緒論

## 第一節 研究背景與動機

自 1990 年時，英國伯納斯-李(Berners Lee) 在瑞士提出並開發了全球資訊網(World Wide Web)後，相繼應用於網路上的軟體，也如雨後春筍般應運而生。讓網站的開發變的相對簡單容易。

而網路應用系統包羅萬象，始終脫離不了一個最重要的元素：內容。網路應用系統的內涵可以是各式各樣的，包括：文字、多媒體等。一般新聞網站提供的是文字的內涵，最近流行的影音分享網站提供的則是多媒體的內涵；除此之外，還有線上版的 Word，內涵則是提供了共同編輯協作檔。網站中的內涵則是網路應用系統中最重要的因素；也可以說網路應用系統是以檔為中心[1]。所以網路應用系統對於內涵品質及更新的要求是非常高的，如網站的優惠訊息，能在短時間內，將訊息發佈在網站上，並掌握到最即時的反饋，做出反應與處理。

目前大部份網站系統開發，總是缺乏較嚴格及有系統的方式，大多數的應用開發和管理，程度上都是依賴開發人員個人的知識和經驗，造成的後果就是開發的網站系統多為低質量的產品[2]。而傳統的軟體工程(Software Engineering)[3]方法並不能適應網站設計的要求，而網站工程(Web Engineering) [4]的需求就變的相對重要。

在網頁技術快速的演進下，已不再是靜態展現資訊而已，而是需要與瀏覽者友善且快速的互動，而如何有系統的應用 Web Engineering[5]在網頁上的開發，變得格外重要。Web Engineering 提供了開發網路應用系統的相關概念，而每年的 www 國際研究會中，也都會針對全球資訊網站中議題進行討論。

由於寬頻的普及，造成了資訊需要被快速的取得，在一個網站的開發中，敏捷式開發(Agile Development)[6,7,8]概念的重要性更加明顯，網頁如何在最短時間內將資訊做出反應，都是需要被關注的議題。

因此在本研究中，主要聚焦放在網站透過 Web Engineering 有系統的規劃，重新應用在網站的作業流程上，以敏捷式開發的概念，以人為主軸，找出需求加以規劃設計，並記錄步驟，並在最後透過標準化的制度規範，將網站當作一個工程概念來開發基礎建設，在繁雜的程式裡，可以同時兼顧網站品質、開發時間及成本。

## 第二節 研究目的

本研究將以實作的方式，應用在公司的促銷廣告推出上線的流程上，藉由相關文獻對 Web Engineering 的參考及探討，應用到實際規劃上，將目前公司網站在推出廣告耗時且重複動作太多的部份，在建立明確的 SOP 後，達到改善討論次數、廣告上線時間及網站頁面一致性，亦可在工作上，讓新進人員可以做到無縫銜接工作，做為本研究主要的目的，在現在的工作流程上，有所掌控及助益。

## 第三節 研究方式

在本研究中，將公司網站進行以下三個步驟規劃：

### 一、系統模組化

利用 Web Engineering 中的 Modeling[9,10]概念，將網站版面重新規劃，將廣告要顯示的區塊規格化、數字化，便於日後製作網站。

### 二、需求分析

透過 UML 工具繪製出需求層面，藉由 Use Case 圖讓整個作業流程，有更明確的工作動向。

### 三、敏捷式開發應用

敏捷式開發就是不斷的開發並立即加以修正，為了便於討論，利用線上的協作系統 Moqups，在即時互動下，可以做出最直接的反應。

## 第二章 文獻探討

在 Pressman 的研究裡，將 Web Engineering 的流程分為五個階段，如圖 17 中看到五個重要流程步驟，依序為溝通(Communication)、計畫(Planning)、塑模(Modeling)、建構(Construction)及交件(Deployment)，每個階段中有主要的執行項目及達成目標，透過這樣的流程安排，可以有系統的將網站開發過程加以區分階段進行，每個階段都是循序漸進，讓開發過程有個很好的導引方向。

在網站的開發中，與客戶的溝通(Communication)是最重要的一個環節，能清楚瞭解客戶建構網站的主要目的與需求，才能完成符合客戶所需要的網站設計，當在瞭解客戶的想法及需求後，就需要開始計畫(Planning)，在這個階段裡，將會針對開發團隊的建立、分工與時程表、風險評估與管理等未來開發網站的內部準備，做一套很明確的規劃及說明，而在塑模(Modeling)這個階段裡，則是利用透過溝通所得到的相關資訊加以分析，構思了網站上的設計藍圖樣貌，在建構(Construction)這個階段裡，就開始運用到網站的實際作業中，程式的開發及不斷的測試，在測試過程中找到問題並加以解決，而這個部份亦可加入客戶端的測試數據，而非開發系統者自行的測試，最後一個階段則是交件(Deployment)，經過上述四個嚴密的工作流程，到了交付的階段，這個階段不代表結束，而是要持續觀察客戶使用後的反饋，並加以處理及應變，且做好每次的問題處理。

網站開發過程中，需要有一套完善的系統規劃程式，並導入一個搭配網站型態的方案來執行，在本研究中就是利用 Web Engineering 為網站規劃的原則，並在實際開發時應用敏捷開發(Agile Development)，在實作時有個準則依歸，以期能建立長期的品質控管和網站完整性。

根據一些實務經驗建構成功的網站，建議先有一個開發流程如圖 1 所示。這個過程有助於擷取需求，提昇整合來自不同領域、促進各會員參與溝通開發程式、支援連續演進與維護、協助管理資訊內容，並有助於成功管理複雜性的開發過程[11]。

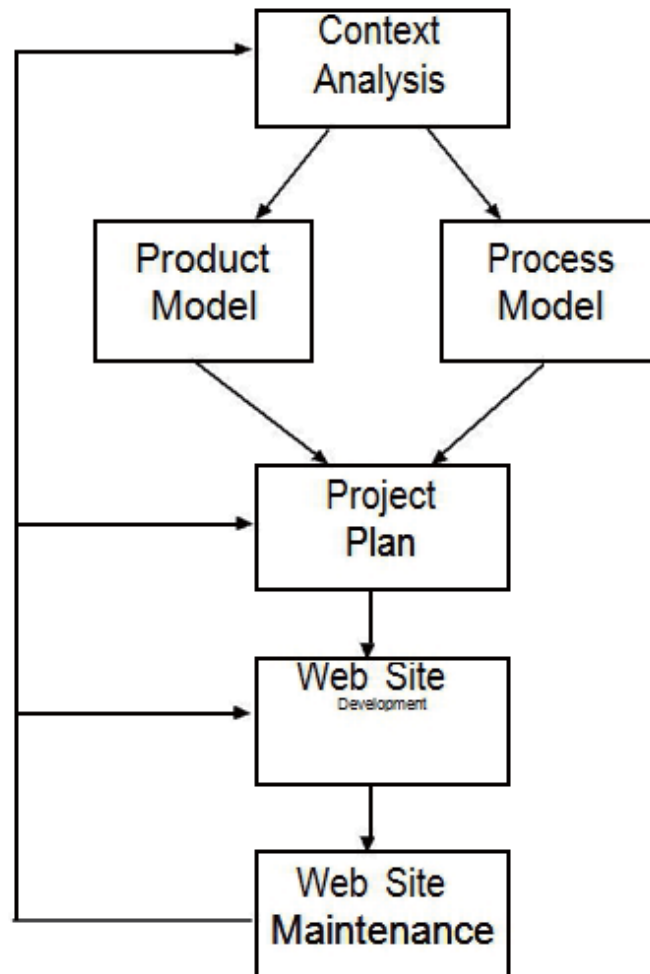


圖 1: 網站基礎開發流程(Development process for Web-based systems)[11]

## 第一節 網站工程

Web Engineering 這個名詞，緣起在每年舉辦一次的 WWW 國際研討會，在這個研討會裡主要探討全球資訊網中的各式各樣議題，包括一些技術與非技術的；Web Engineering 是屬於技術的應用層面，接受相關各種工程技術發展成果的研究論文，並於研討會上發表。

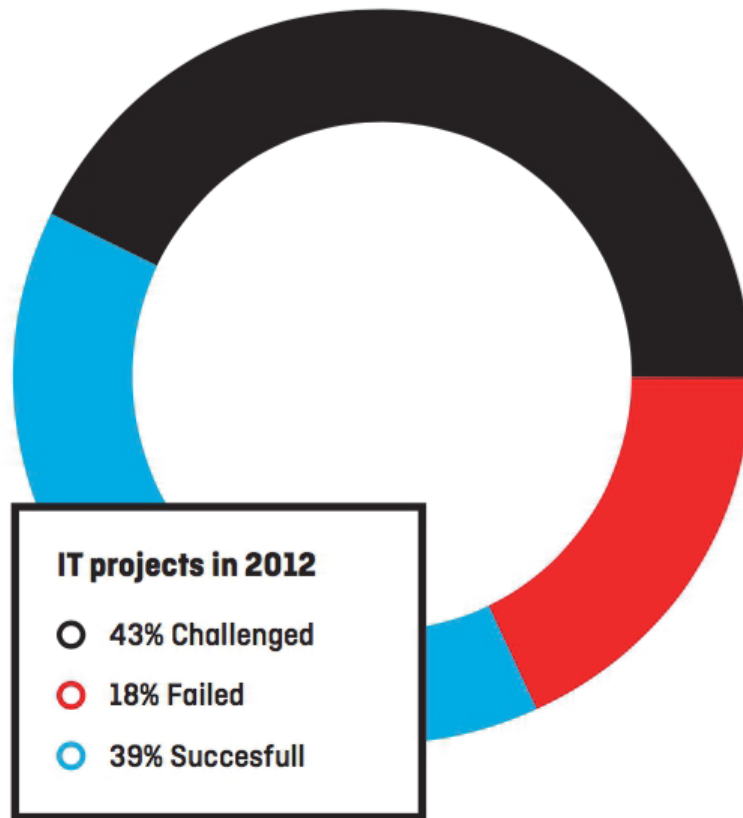
然而什麼是 Web Engineering，可簡單的說明它是針對網站開發做管理與控制的一種模式，而建立 Web Engineering 的主要目標就是讓開發的網站更有效率、且有高質量及能降低風險。網站開發中，常常有著太多因素造成網站失敗收場，然而這大多是無法如期交付、不符合客戶預期需求及超出原先的開發預算，而分析其主要失敗原因，大致區分如下 [12]：

- 一、 忽視客戶不斷變化的需求；
- 二、 沒有保留歷史檔做為決策參考；
- 三、 忽視監督項目進度；
- 四、 忽視不斷測試與修改；
- 五、 沒有導入專業的開發管理模組，全靠主觀決策。

來自一份 Standish Group 在 2013 年發佈的統計數據，很清楚的告訴我們在 2012 這一整年中，來自 IT 的專案開發中，只有 39% 是成功的案例，在圖 2 的統計圖可以明確看出，在他們的分析中，失敗的因素各有不同，歸納出三個主要的因素[30]：

- 一、 預算：大部份出資的創業公司，對於他們本身要開發系統，實際花費往往超出當初的預算，因為會隨著需求增加更多的功能，卻不願追加預算。
- 二、 缺乏溝通、透明度：這算是最嚴重的失敗主軸，客戶及開發者兩者之間無法取得良好的溝通管道，是最快速導致失敗的主要原因。
- 三、 拒絕改變：當開發的軟體在不斷更新及重新定向時，若此時無法接受新的改變，將會導致專案中存在著中斷開發的相關問題。

因此有些學者認為是一種軟體蕭條現象而不是軟體危機問題，主要是在沒辦法精準預測時間及成本。而同樣的 Web Engineering 的 Web-based System 也有同樣的問題存在，大型且複雜 Web-based 系統及應用，想要成功開發，Web 開發者需要配製一套規律開發程式、健全方法、良好開發工具及允許開發步驟都為指導方針[13]。



**SOURCE:**  
CHAOS Manifesto 2013, The Standish Group 2013.

圖 2: Standish Group 2013 統計圖

資料來源：Standish Group

隨著網站技術的越加成熟，綜觀上面所列的種種失敗原因，需求的被發現及溝通上的順暢，是一個開發網站裡需要被放大重視的，更可看出 Web Engineering 應用是無法忽略的技術與發展。因為它是一種系統、量化方法(理念、方法、技術、工具)的應用程式，以高規格的需求分析、設計、實施、測試、運行和高品質網站應用維護。

Web Engineering 是為了成功開發大型 Web based 系統，採取完整且主動性開發方式，在企業、教育、保健及政府都有將傳統系統移植至 Web 環境的趨勢，採用 Web Engineering 相關方法，Web Engineering 是新興的規程，具備理論及實務特性，包括：研究者、開發者、學術界及使用者來探討及研究[13]。

Web 應用範圍各有不同，從小規模的企業、到大規模企業的官方網路和企業內部網路。多年來 Web 應用的範圍也從簡單變得越來越複雜，例如 Slashdot 網站、奧運網站接收到每分鐘數以萬計點擊率、或是在詞彙的動態特性資訊、使用多媒體或其他方式等，他們可能會提供大量的動態資訊，儲存在多媒體格式的檔案中(圖形、影像、視訊) [14]。

Web Engineering 也如同 Software Engineering 一樣的架構，在品質上也有一定的要求，因為品質也包括了日後在維修的考量，在圖 3 中很明白的說到在整個開發過程中，有四個影響因素是一個重要的基礎，表 1 將內容文字化，讓我們可更加清楚其重要性 [28]。

表 1: 軟體工程四大影響因素[28]

工具	支援的程式與方法
方法	提供如何使用的技術支援
流程	技術層級整合
品質	促進持續改善流程

從這些影響的因素中，清楚的明白首先需要有好的開發規劃，才能擁有好的成品品質，套用在開發的過程裡，是很重要的理念。Web Engineering 也是有著相同執著，能將網站明確劃分出需要及準則，在執行的過程裡，運用各種方法及模組，結合各種工具的應用，讓整個結構慢慢形成一個健全的網站。





圖 3: 軟體工程的四個影響[28]

而一個網站開發，首要是能控制開發的時程，能如期交付系統，並在交付後提供可維護及修改的彈性空間，還要能降低風險並提高品質，而交付的系統更是需要使用者所認同的，這樣的環環相扣，需要有別於傳統的網站開發程式，網站基礎應用 Web Engineering 來進行規劃開發，將會在開發流程上得到完善的管控。

## 第二節 軟體工程

軟體工程在 1968 被提出，而其定義，以 IEEE 1993 年為說明：「為一系統化的、透過訓練有紀律的，及可量化的應用方式，來著手處理軟體發展，運作及維護。」，換句話說，它的核心目標，就是以系統化、規格化及量化的原則和方法，進行軟體的開發及維護，嘗試使用一些經驗證過而且可行的方式，讓整個軟體開發過程，控制在可接受的時間及預算內，完成高品質的軟體系統。

在表 2 中，將軟體開發的架構大致做一個簡單的分析，可做為在開發應用時，明確定位出開發方向及需求。

表 2: 軟體開發架構

資料來源：軟體工程([irw.ncut.edu.tw/peterju/se.html](http://irw.ncut.edu.tw/peterju/se.html))

單機架構(1-Tier)	展示層，商業邏輯層，資料層都在單機上處理，適用於文字處理，個人資料處理(PIM)等單機架構。
主從架構(Client/Server,2-Tier)	將資料層分離出來，儲存到資料庫伺服器，適用於多人存取資料的環境。
分散式架構(N-Tier)	將展示層，商業邏輯層(放在 AP Server)，資料層(放在 Database Server)都各自獨立，適用於平台不同，網際網路的環境。
網路服務(Web Service)	將整個網際網路視為區域網路甚或是作業系統般。

在譚慶平、毛新軍及董威等人出版的軟體工程實踐教程(2009)，如表 3 中，很清楚的表列出軟體工程中常用到的開發時的過程模型，可以依據表格中，找出適用的軟體開發模型，才能讓整個作業流程順暢。

表 3: 軟體過程模型特點

資料來源：軟體工程([wiki.mbalib.com/zh-tw/瀑布模型](http://wiki.mbalib.com/zh-tw/瀑布模型))

模型名稱	技術特點	適用範圍
瀑布模型	簡單，分階段，階段間存在因果關係，各個階段完成後都有評審，允許反饋，不支持用戶參與，要求預先確定需求	需求易於完善定義且不易變更的軟體系統
快速原型模型	不要求需求預先完備定義，支持用戶參與，支援需求的漸進式完善和確認，能夠適應用戶需求的變化	需求複雜、難以確定、動態變化的軟體系統
增量模型	軟體產品是被增量式地一塊塊開發的，允許開發活動並行和重疊	技術風險較大、用戶需求較為穩定的軟體系統

模型名稱	技術特點	適用範圍
反覆運算模型	不要求一次性地開發出完整的軟體系統，將軟體開發視為一個逐步獲取用戶需求、完善軟體產品的過程	需求難以確定、不斷變更的軟體系統
螺旋模型	結合瀑布模型、快速原型模型和反覆運算模型的思想，並引進了風險分析活動	需求難以獲取和確定、軟體開發風險較大的軟體系統
RUP	可改造、擴展和剪裁：可以對它進行設計、開發、維護和發佈；強調反覆運算開發	複雜和需求難以獲取和確定的軟體系統；軟體開發項目組擁有豐富的軟體開發和管理經驗

針對本研究中採用到且相似的案例，加以述說比較，並不一一列舉陳述表 3 中各模型，排除較不適合的流程模型，就以瀑布模型(Waterfall Model)及快速原型模型(Rapid Prototype Model)做一個比較說明。

### 2-2.1 瀑布模型(Waterfall Model)

溫斯頓·羅伊斯(Winston Royce)提出了著名的“瀑布模型”(1970)，主要有六個基本的活動項目，流程是由上而下順序作業，就好像瀑布一樣，依序有軟體規劃、需求分析、架構設計、程式編寫、系統測試及最後的系統運作與維護，如圖 4 瀑布模型。

其核心思想是利用順序方式來呈現，將所要的功能實現與設計分開，方便在開發時的分工協作上，在前一個步驟完成後，就可以將動作由下個程式接收，每次只要關注目前工作的階段，故在這樣的開發過程模型，有幾個較為明顯的缺點：[32]

- 一、在項目各個階段之間極少有反饋。
- 二、只有在項目生命週期的後期才能看到結果。
- 三、通過過多的強制完成日期和里程碑來跟蹤各個項目階段。

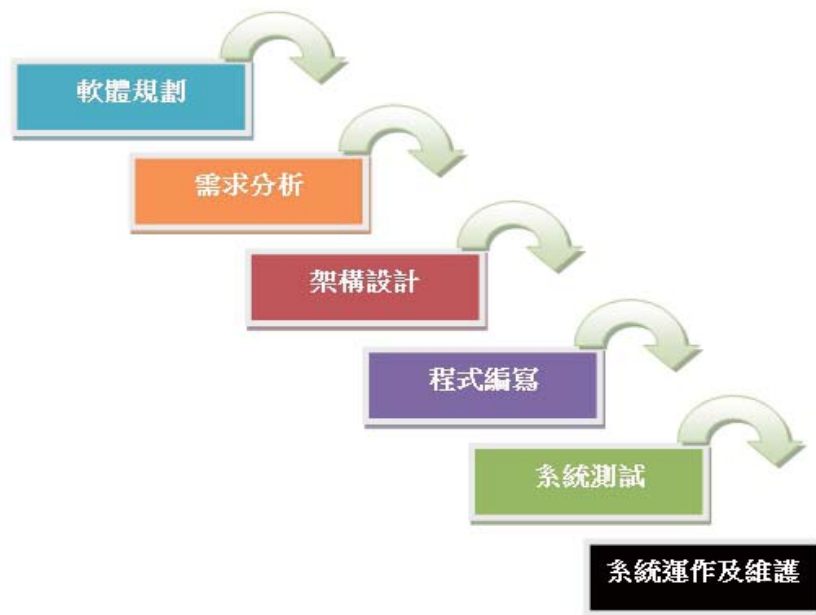


圖 4: 瀑布模型

可以明確看到在這樣的開發過程裡，少了開發者與需求者即時的反饋動作，加速開發軟體的時間，卻有可能不是需求者所需要的，而這樣的開發流程模型，對於本研究中的網站，需要經常性的變化並即時，這種瀑布模型開發，並不是很好的模式，故不以這個模式做為本研究中的應用模型。

### 2-2.2 快速原型模型(Rapid Prototype Model)

它是在開發真實系統之前，構造一個原型，在該原型的基礎上，逐漸完成整個系統的開發工作，圖 5 為快速原型模型的流程圖。例如：客戶需要一個 ATM 機軟體，可以先設計一個僅包含刷卡、密碼檢測、數據輸入和帳單列印的原型軟體提供給客戶，此時還不包括網路處理與資料庫存取以及數據應急、故障處理等服務。快速原型模型的第一步驟是建造一個快速原型，實現客戶或未來的用戶與系統的交互，用戶或客戶對原型進行評價，進一步細化待開發軟體的需求。通過逐步調整原型使其滿足客戶的要求，開發

人員可以確定客戶的真正需求是什麼；第二步驟則在第一步驟的基礎上，開發客戶滿意的軟體產品[33]。

快速原型的關鍵在於盡可能快速地建造出軟體原型，一旦確定了客戶的真正需求，所建造的原型將被丟棄。因此，原型系統的內部結構並不重要，重要的是必須迅速建立原型，隨之迅速修改原型，以反映客戶的需求[34]。

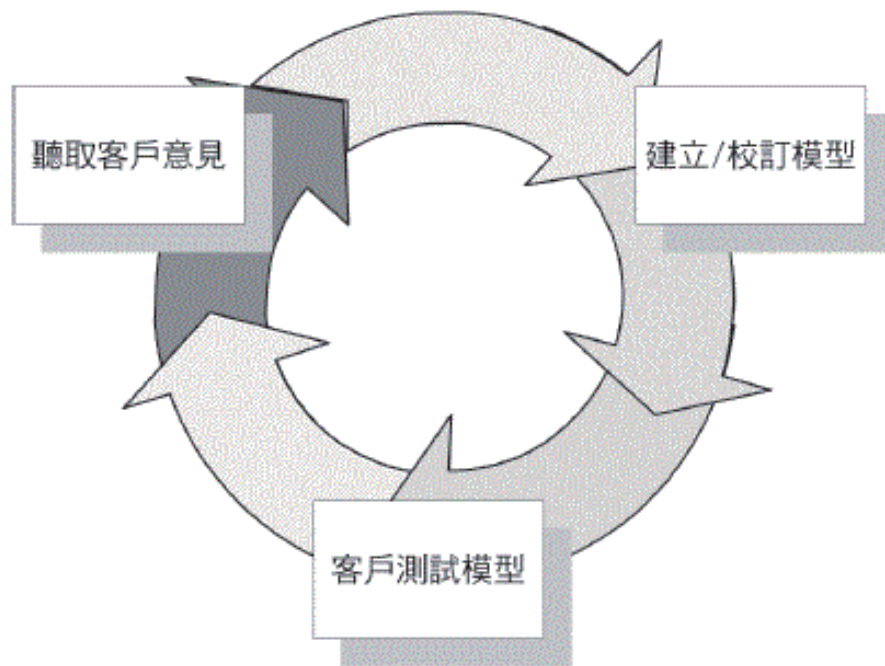


圖 5: 快速原型模型

資料來源：糊”私人天地”塗([cht.tw/x/ps11r](http://cht.tw/x/ps11r))

而這樣的模型是最早能與使用者達到共識與參與感，也是能有良好的溝通學習的機制，讓每次使用者給的反饋中，能快速改善，並能將反覆的週期縮短。

但這樣的模型，在本研究的使用上，於快速溝通上有達到需求，但並不需要去重製一個原型，在成本上也較不適用，這適合在網站開發的初期，若像本研究中的網站，只是需要針對訊息的快速更新，並不合用這樣的模型。

### 2-2.3 敏捷式開發(Agile Development)

由於上述幾個模型，並不實用在本研究中，在隨著網站開發的迅速發展，開發網站的要求也相對提高不少，為了滿足必須在快速且維持品質的思維下，又能在現有的網站環境裡，解決目前開發時所產生效率不彰的情況。其中敏捷開發方法(Agile Development)就是為解決這樣的困擾而生。

敏捷開發方法是應對快速變化需求的一種開發能力，是依據核心、反覆運算、循序漸進的開發方式。主要精神為確認需求、細分功能、測試、整合、反覆運算、重構。充分的理解客戶的需求，將需求細分成各個小功能，在每個小功能的開發都先經過縝密的測試，以確保寫出來的程式是沒有問題的。這樣的切割成小區塊的需求開發，有助於能快速反應及修正。圖 6 繪出彼此小功能間相互關聯圖。

接著進行整合測試，若是整合出問題，便可快速找出是哪個小功能出了問題，且能立即迅速修正。整合完成確認沒問題後，再確認是否有其它需要調整的部分，進行反覆運算。最後才進行程式的重構，在不改變介面的狀況下，對程式碼進行優化的動作，不只是速度方面，也盡量讓程式碼達到簡潔、可擴展。在於將需求細分與整合的部分。因為已經將功能細分，所以如果整合的過程中覺得有那些部分不夠完善或者客戶想要調整，便可以快速直接的將部分功能替換掉[31]。

本研究中，重構觀念就是在這樣的環境下，將原來的流程加以結構化，在不動到主系統的情況下，如何去改變流程及做法，讓整個開發運作敏捷快速許多。同樣在很多開發專案中，若能如同敏捷式開發一般，將每個階段劃分出來，在每一個區塊裡就能很清楚得知需求元素，該開發的功能也就能快速的發佈。



圖 6: 敏捷開發方法

資料來源：FineArt([www.fineart-tech.com](http://www.fineart-tech.com))

敏捷開發方法(Agile Development) 是一種以人為核心、反覆運算、循序漸進的開發方法[20]。原則上敏捷式開發主要的精神在於較短的開發循環（建立在反覆式開發方式上）以及漸進式開發與交付。換句話來說，專案的成果，包含計畫、各類的需求細節、設計等都會隨著專案的進行而漸漸完整，而非在一開始就將所有的計畫與需求擬定完成。2001 年時支持敏捷式開發的社群組成了 Agile Alliance(<https://www.agilealliance.org/>)，並且發表了 Agile 宣言與原則[21]。

敏捷式的 12 原則(The Agile 12 Principles)[22]中，依據每一個原則代表的精神，套用在網站的開發，將討論的週期縮短，又可快速交付的開發環境要求，團隊式開發的運用上，敏捷式開發是適用的。

在敏捷開發方法(Agile Development)中，軟體項目的建構被切分成多個子項目，各個子項目的成果都經過測試，具備集成和可運行的特徵。換言之，就是把一個大項目分為多個相互聯繫，但也可獨立運行的小項目，並分別完成，在此過程中軟體一直處於可使用狀態[20]，如圖 7 為敏捷式開發(Agile Development)。

Web Engineering 將網站如工程般來制定時，需要有一個正確的工法來施工，而敏捷開發方法(Agile Development)的快速開發及溝通，正好符合本研究中目前工作所需，應用導入在公司網站開發。滿足即時性需求，提昇服務品質，提高維護速度，延長網站生命週期，這些都是非常重要的意義。

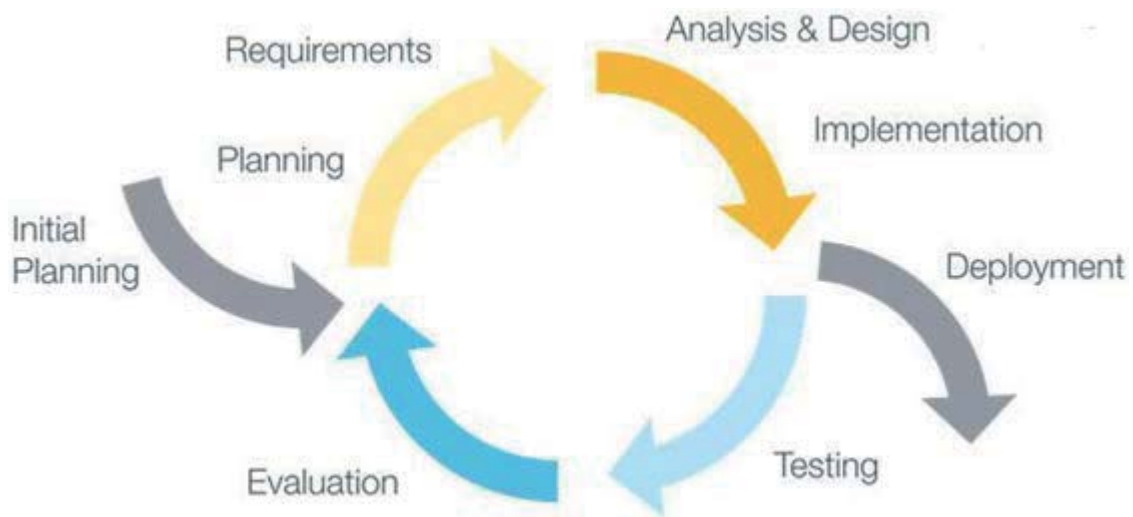


圖 7: 敏捷開發方法(Agile Development)

資料來源：網頁教學網([www.webjx.com](http://www.webjx.com))

敏捷開發方法(Agile Development)目前成為 Web 開發應用程式的一個標準，而 Model-Driven Web Engineering (MDWE) 方法是提高生產效率，故在本研究裡應用敏捷式開發中利用 Moqups 這個網站，做為開發時的一個溝通管道，利用這套線上協同作業的工具網站快速且即時的版面修改記錄，讓開發者及主管能進行即時溝通，並做出快速開發的目的。



### 第三節 UML 建模語言

西元 1997 年，物件管理協會（Object Management Group；OMG）提出了統一建模語言（UML），其中一項的目的就是提供軟體開發社群使用穩定且通用的設計語言，也可以用來開發並建構電腦應用程式，UML 帶給 IT 技術人員長久以來所需要的一個統一標準建模標記法。利用 UML，IT 技術人員能夠閱讀、傳達系統組織架構還有設計圖面，就像是建築工人多年以來使用建築物的藍圖施工一樣[15]。有利網站開發的順利運作。

當在規劃及開發大型網站時，有一個很重要的部份，就是網站開發的流程圖，在資訊系統愈來愈往大規模化、複雜化的發展趨勢下，演化促成了統一塑模語言（UML, Unified Modeling Language）的誕生。UML 建立在物件導向設計、與非專開放式精神的塑模規約語言，適用各種物件導向軟體開發應用[16]。

當我們將要開發的網站，利用 UML 這一個可圖形化的塑模語言，讓需求能更有條理性的組織，並與功能性做出區隔，是一個最佳的工具，也是網站開發時設計思維的手段。而 UML 工具裡，可找到適合自己網站型態的 UML 類別，除了本研究會使用的案例圖(Use Case Diagram)，這裡介紹 UML 常用到的三劍客：

#### 2-3.1 循序圖(Sequence Diagram)

透過時間軸的概念，可以控制到程式發生的先後順序。捕獲單一案例圖的行為，同時顯示在特定用例的時間框架中的對象以及這些對象之間傳遞的消息。Sequence Diagram 並不顯示對象之間的關係[17]。

Sequence Diagram 有兩個象限：垂直象限依照訊息/呼叫發生的時間順序，來描述訊息/呼叫的先後次序；水準象限則描述一個物件實體（instances）傳送訊息給哪一個物件實體[15]。圖 8 為一家宅配物流公司的 Sequence Diagram，明確看到每個階段所需要完成的任務及要傳送的資訊。

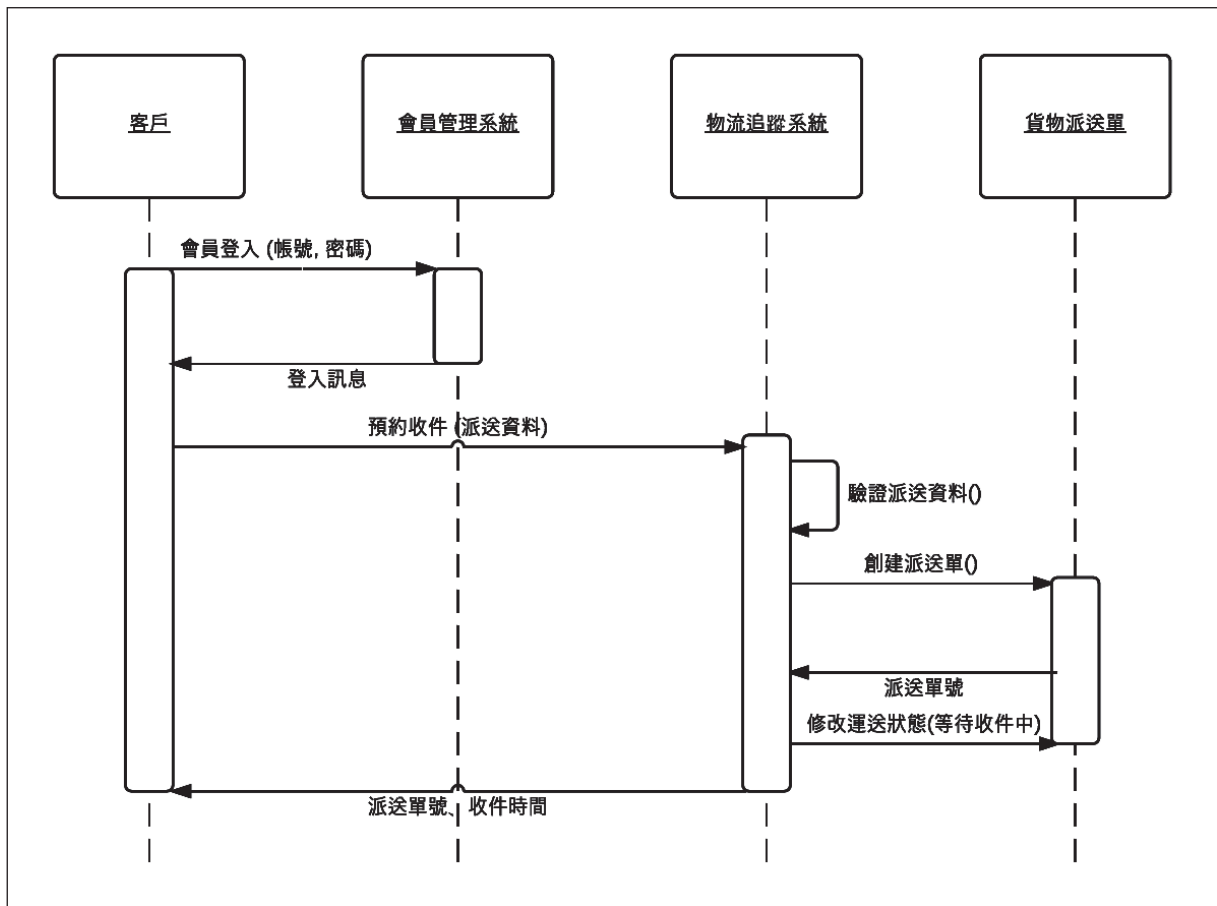


圖 8: 循序示意圖

### 2-3.1 類別圖(Class Diagram)

Class Diagram 是設計人最不想去畫的一款圖，但它卻是 UML 中重要的一環，它可以表達系統內部的靜態結構，而這些系統內部靜態結構會影響日後能否接收到多元化變動[18]。

Class Diagram 描述不同的個體（人、事物和資料）相互的關係；換句話說，它表示了系統的靜態結構。類別圖能夠用來表示邏輯類別，通常也就是組織裡面的業務人員所談論的東西：搖滾樂團、CD、收音機；或是貸款、住宅抵押、汽車貸款和利率。Class Diagram 也能夠用來表示實作的類別，也就是一般程式設計師所要處理的物件[15]。圖 9 中為一宅配物流公司中的 Class Diagram，在每一個格子中分

三個區域，分別為類別的名稱、屬性及方法，若能善用並確實規劃好 Class Diagram，有助於在開發軟體的思路。

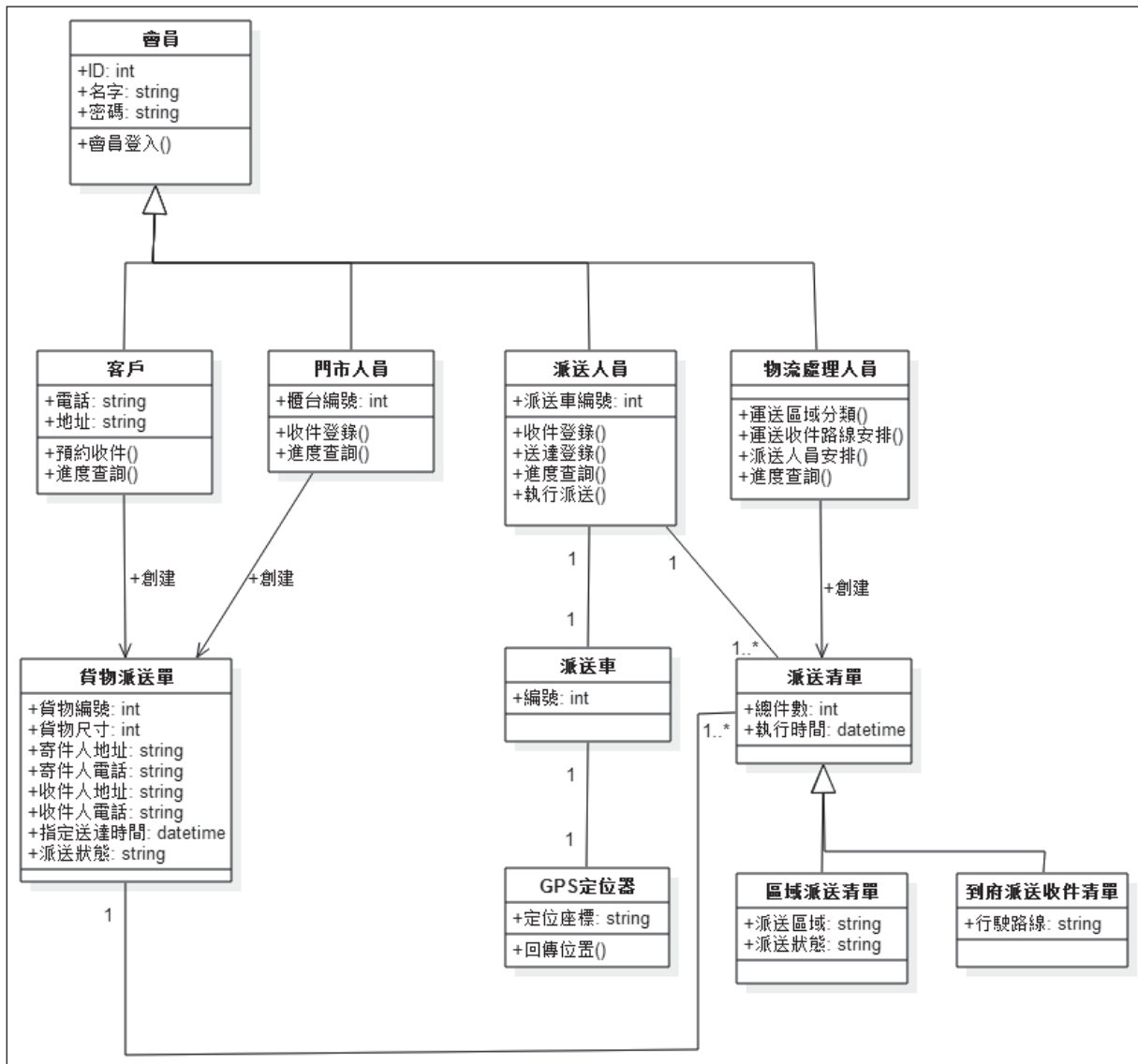


圖 9: Class Diagram 示意圖

### 2-3.1 案例圖(Use Case Diagram)

在一個網站開發時，需要繪製出 Use Case Diagram，原因是在於可以用鳥瞰的視野來看系統的全貌，很容易可以看界定系統的範圍、區分系統內與外[19]，在本

研究中更可以看到使用案例中所要表達的層次(Level)，如主管、客服與網站設計人員的相關層級。

圖 10 為一宅配物流公司 Use Case Diagram，主要是幫助開發者能明確看清系統的需求所在，包括參與者間的互動關係及程式的處理過程。

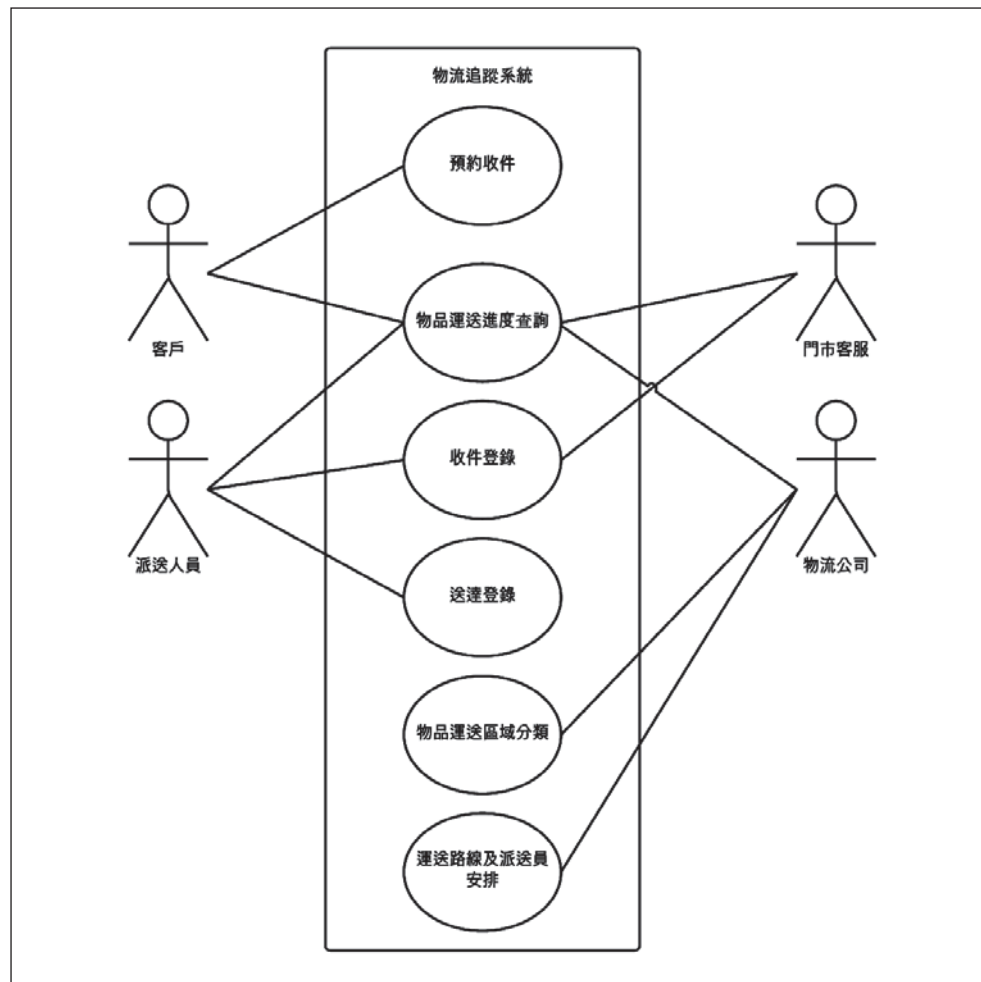


圖 10: Use Case Diagram 示意圖

在開發系統時，不用 UML 同樣可以開發系統，但在整體的結構以及在日後維護及更新上，相信會變的更加的困難及耗時，找到對的類型來繪製 UML 圖，它會是一個很好的需求記錄，讓整個開發有一個明確的目標，以及如何去做的導向。故在本研究中所

採用的是 Use Case Diagram，讓每個參與者知道需求點在哪裡，亦可改善重複重工的事，整個工作時程相信也會獲得良好的改善。

而在本研究中，將採用 Use Case Diagram 做為實作時的方式，主要是要將目前廣告上線的工作時程快速化，才會採用 Use Case Diagram，它可以很清楚分析出每個參與者的工作流程，剔除一些重複性動作及不必要的步驟，讓工時得到更有效率的發揮。

Use Case Diagram 既能分析出商業處理流程又能透析出系統需求的有效方法，並且在繪製過程中也是較簡單和容易掌握。而且 Use Case Diagram 使需求在實現決定的過程中變的更加容易，且也簡化了整個流程。

#### 第四節 網站的應用

通常我們瀏覽網頁時，常因不同目的而使用不同類型的網站。當前網站應用程式的複雜性與範圍相差很大：從簡單個人小型網站到大型企業應用網際網路和企業內外部網的服務。基於網站的應用不同，大致可分為七個類別在表 4 中[23]。

透過不同類型的大致分類，發現類型的不同，在開發網站時所使用的開發模式就會有所不同，如本研究中公司網站是屬於資訊型態的網站，就是要有快速將最近資料反應在網站上，讓使用者看到最新的訊息資訊，而在敏捷式開發的過程中，我們所運用的網站(Moqups)，就是屬於表 4 中的協同作業型，故在開發時對網站的定位也是相當重要，才能找到相對應用的開發程式，做到事半功倍的成效。

在本研究中的網站，使用敏捷開發方法(Agile Development)方式，將快速開發過程聚焦的協作平台系統，在快速的開發及不斷的溝通與持續的修正的過程裡，顧及成本、時間及品質，收到不錯成效，利用這機動性的規劃開發系統，可在發現問題的同時，馬上進行修改並改善工作。

敏捷開發法是從軟體產業衍生出來的團隊工作方式，對敏捷專案管理的重要性不可言喻，將各部門在這個團隊裡工作、配合內容，可以明確的用 Use Case 做出條列式分析，發現在設計一個促銷廣告案裡，有著即時的價值及時效，而在敏捷開發方法(Agile

Development)核心精神裡有著「價值驅動」(value-driven)、急迫性(urgent)及需求變化快(volatile)，有著相同的需求及應用，如圖 11 為表示 Development 核心精神。

如何快速的在高急迫性(urgent)及需求變化快(volatile)中，將我們的促銷廣告在最短時間內推出，並在不斷快速開發及修正時，做到利用最短時間完成促銷廣告 On Line，並創造出更多的廣告經濟效益。

表 4: 網站類型[23]

類別	範例
資訊型	線上報紙、 產品目錄、 通訊、 服務手冊、 線上分類廣告、 線上電子書
互動型	登記表格， 演示文稿自訂的資訊， 線上遊戲
事務性	電子購物、 訂購商品和服務， 網上銀行
工作流程型	線上規劃和調度系統， 庫存管理、 狀態監測
協同作業型	分散式創作系統， 協同設計工具
線上社區市場型	聊天群， 推薦產品或服務， 線上市場， 線上拍賣的推薦系統
入口網站型	電子商場、 線上仲介



圖 11: Development 核心精神

## 第三章 研究方向

本研究將以公司資訊網站為實例，利用 Web Engineering 重新檢視規劃網站，並以 Modeling 做出分析需求，UML 繪製出 Use Case 使用圖例，將參與促銷廣告相關人員的關係明確標示出來，最後在敏捷式開發(Agile Development)來進行開發與交付，並在實做過程中，透過不斷討論及意見交流，即時的做出修改與異動。

### 第一節 研究方法

公司網站初期，為求快速上線，網站設計人員，使用直覺式的 HTML<frameset></frameset>格式做為資訊顯示框架設定，如圖 12 所示，但其格式在各家的 Web Browser 皆會有不同的呈現方式，甚至也有不支援其框架的部份屬性，導致無法呈現或畫面的不完整，而在當時的處理方式，則是要求使用者使用指定的 Web Browser，這樣的限制對使用者操作並不是很好的方式。

當時公司資訊網站隨時都會有些即時公告及優惠相關的訊息廣告圖，常因呈現的位置及方式，耗費較多時間在溝通上，圖案的大小也沒有一定的格式，讓網頁設計及美工人員在作業上造成諸多困擾。亦由於這樣的規劃與設計，讓後續接手的網站人員，需要重新檢視程式及語法，無法在短時間內就能立即銜接上工作進度。

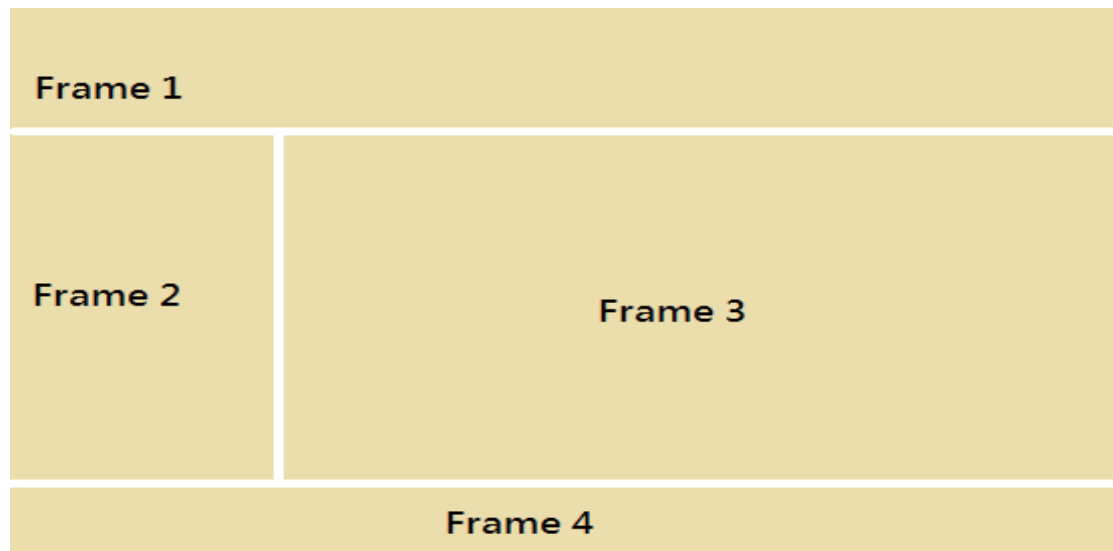


圖 12: Frameset Design

基於網站的機動及異動性高，將廣告的區塊固定在網頁的的某一個位置，如圖 13 中可看到，搭配 CSS 樣式表及 Java Script，重置廣告展示的方式，也因為這些語法是跨平台都可以使用，故不會因為使用者用 Browser 時，顯示的位置及畫面無法正常展現。也可以較多應用在廣告區塊的部份。

其實使用 table 也是能做到這樣的效果，但它的的機動性就沒有 CSS 來的靈活，利用 CSS+DIV 我們就能將這個廣告區塊定義下來，接下來的工作就是將工作流程重新架構起來，就可以提高作業的順暢度了。



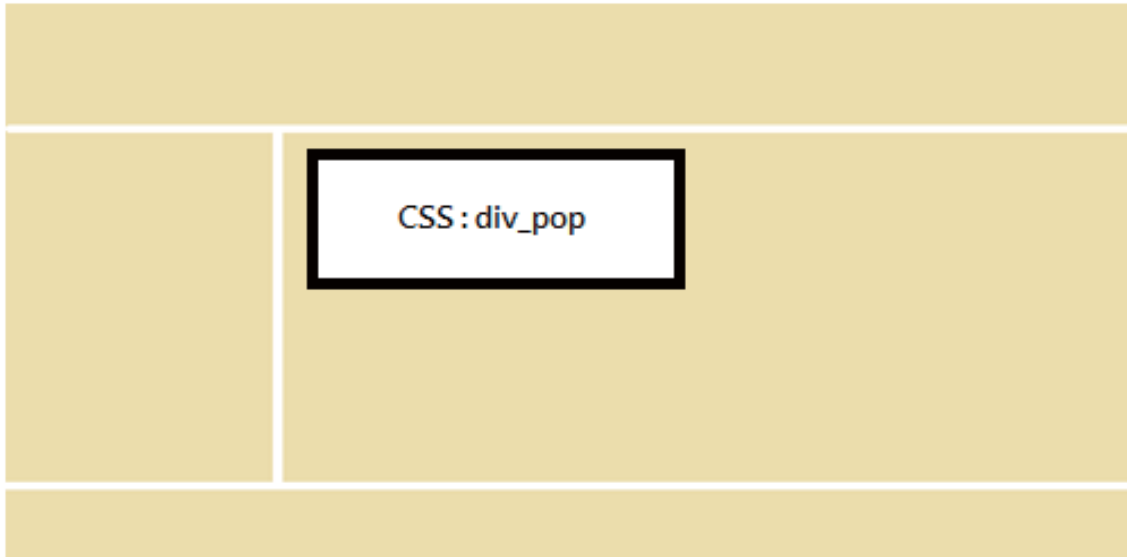


圖 13: 重置廣告區塊

當有些功能被要求加入，卻因當時沒有很好的網站規劃，導致每次要新增一個功能或選項，都需要花費時間去檢視 Program Code，重複性動作太多，亦增加了開發人員的工作量。在每一次變動或申請時，都需要填寫網頁修改申請表格，表格如附件一，往返的檔程式浪費太多時間，若能克服減少這些重複性工作，則必需將網站網頁有系統的模組化，方能快速而正確反應在網站的開發上。所以需要去檢視目前的工作流程，找到目前需要改善及加強的地方，或是可以減少重複動作的步驟，本研究將利用 Web Engineering 規劃去改善，如圖 14 就是公司在未進行 Modeling 時的作業流程，很多時間都是費時在來回各部門間的討論作品以完成一個廣告促銷案。

在一個規劃制度、規格化網站架構下，將網頁配置區隔靜態及動態呈現方式，設置一個固定版面又醒目的區塊，用以放置最新公告及傳達訊息的圖檔。而本研究對象網站，在這區塊就是目前需要即時去改善的部份，將制定規格數據，在網站開發或美編圖檔時能有所依循，非一開始網站初期隨性的配置呈現。

一個沒規劃完善的網站，常因一個簡單訊息的呈現，及個人的喜好觀點不同，主管或相關人員與網站開發人員意見相左等，造成在網頁配置溝通上曠日費時，這是經常會

遇到的情況，而為了有效解決這樣的問題，就要著手探究網站工程理論之規劃流程，使其在最短時間內達到最大功效。

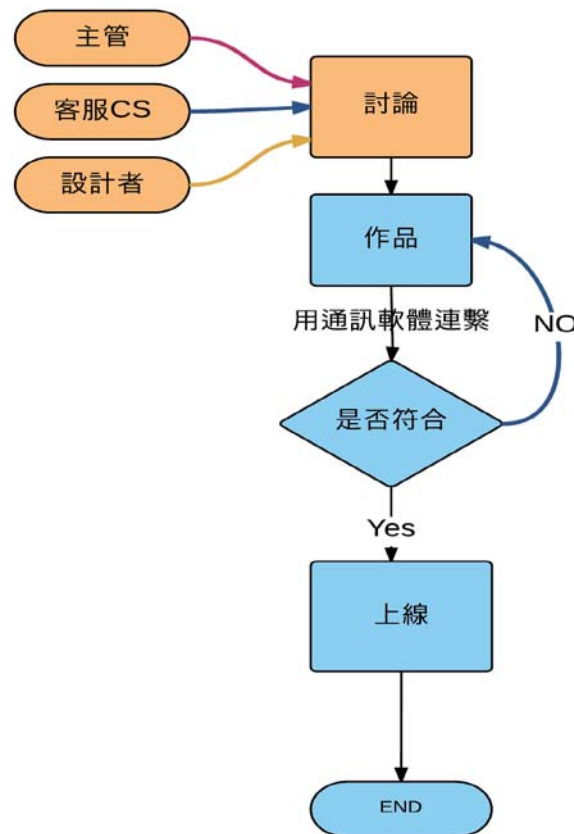


圖 14: 工作流程

## 第二節 研究限制

而本研究實作案例，為已開發完成的網站，故在理論的適應性上有著其局限，無法套用所有網站開發中來應用，若一個全新網站開發時應用到 Web Engineering，則需依循其前述的五階段逐一建構網站，並非如本研究中只利用 Web Engineering 中的 Modeling 一個階段，此即為本論文研究中的限制。

故本研究結果的適用對象，為在現有網站中需做局部重新規劃，且這區塊是需要有著即時更新需求，本研究的成效數據適用在這樣的範圍。

### 第三節 研究與實作

基於以上因素，本研究以實際應用在工作上的效能改善經驗，網站在開發過程中，搭配應用 Web Engineering 系統規劃，讓系統開發有著完善的管理成效。

#### 3-2.1 重置廣告區塊

在有了整個廣告開發的流程圖後，首要修改就是要將原本隨性擺放的廣告位置固定下來，其檔案格式、尺寸大小，都要將其數位化，在程式撰寫時亦方便管理，這次是將廣告區塊利用 CSS 來做樣式管理，若日後有需要編輯圖檔大小，只需修改 CSS 這個檔案即可，且具有共用性，若別的頁面有需要用到這個格式，亦可直接套用，無需重寫。如圖 15: 為廣告區塊的 CSS 碼，方便在做交接時，在整個製作時的動線會很流暢。

```
0 10 20 30 40 50 60 70
1  html { font-size:13px;}
2  body { text-align:center; -webkit-text-size-adjust:none;background-color:#f
3  * { margin:0; padding:0; list-style:none; }
4  i, em, b { font-style:normal; font-weight:normal; }
5  img{width:100%;vertical-align:top;}
6  input,textarea,select{-webkit-appearance:none;border:0;border-radius:0;back
7
8  /*圖案及文字標題*/
9  .ico_title{text-align:left;font-size:24px;color:#333333;border-bottom:1px s
10 .ico_title:after{content:"";width:100%;border-top:1px solid #E3E3E3;positio
11 .ico_title i{margin-right:10px;}
12
13 /*即時廣告區*/
14 #div_pop{
15     width:500px;
16     height:220px;
17     background-color:#FFFFFF;
18     position:absolute;
19     top:20px;
20     left:40px;
21 }
22 }
```

圖 15: 廣告區塊 CSS 碼

### 3-2.2 更新流程圖

本研究以公司內部資訊網站為例，如圖 16 中，在透過 Web Engineering 論點為基礎，將網站即時系統當作一個主軸規劃，如何在最快的時間內將其資訊上線，使用者能在短時間內得到最新的促銷資訊，並讓網站設計人員能在有準則依循下進行開發工作。

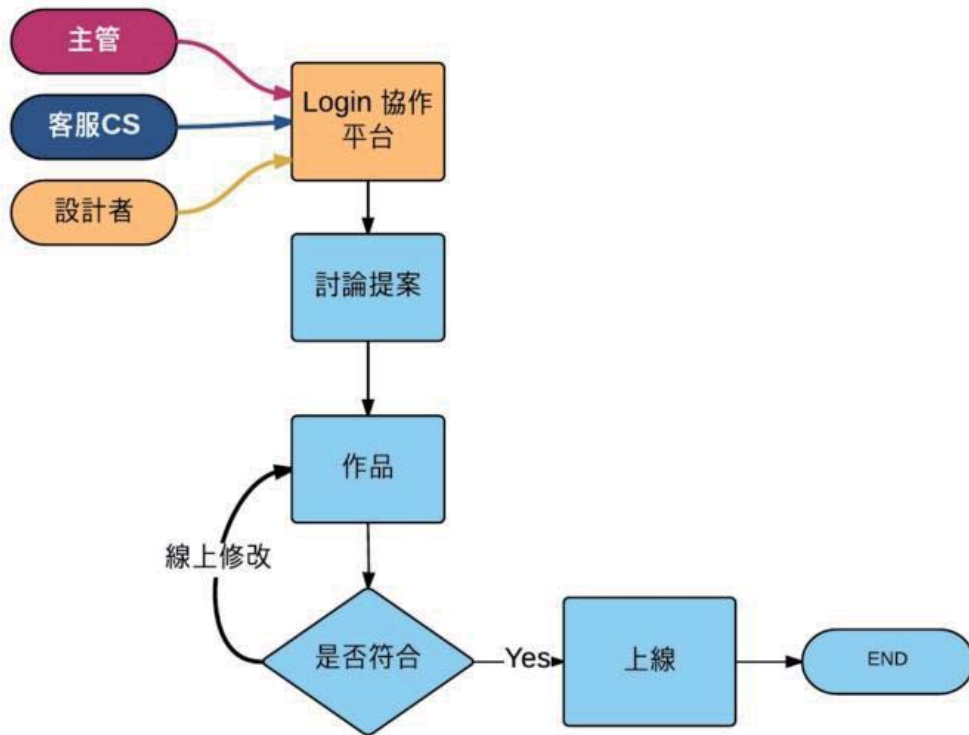


圖 16: 工作流程圖

以系統化、科學化來處理的方法，稱做工程學(Engineering)。為了讓網站能便於管理及快速反應工作流程，將網站以 Web Engineering 的方式進行規劃，搭配 Modeling 建模理念，如圖 17 為 The Web Engineering Process。

Modeling 中的兩個重要準則是 Analysis 及 Design[24]，針對使用者對整個網站中的預期需求，及對網站功能角度來探討，瞭解長處、功能及限定的部份，最後可協助完成 Context Analysis 作業檔[13]。分析出需求再加以設計，將以往隨性化的設計方式，能在一個統一基準原則下進行開發與設計，並以本研究做為實作目標，如何利用 Web Engineering 的 Modeling 的規劃，使其反應在即時促銷資訊，在官方網站中快速呈現。

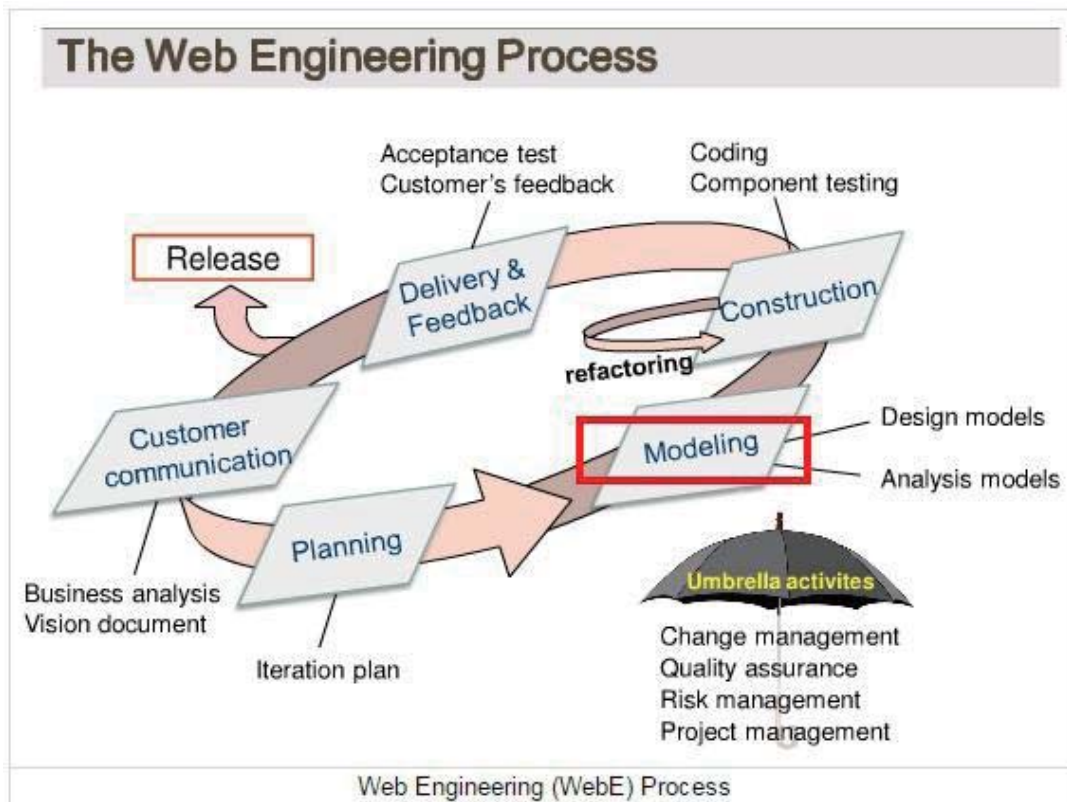


圖 17: The Web Engineering Process

資料來源: Web Development - Project 3

### 3-2.3 使用者案例分析

本研究改善的案子裡，首先會透過 Analysis Models 分析目前參與者的需求及想法，導入 Use Case Diagram[25]找出參與者間的相互關係，繪製出每個參與者需求規格，而這樣明確流程架構，有助於網站快速開發的節奏，圖 18 表示參與者 Use Case 表。

在圖 18 的 Use Case 圖表裡，將本研究中，相關參與者表列出來，並用線條拉出所關聯事項，可看出三者之間最終目標，就是要讓本檔的促銷資訊在整合之下，得到快速的處理方案，讓網站能在最短的時間內看到成果，省略重複步驟過程，預期會快速反應在網頁製作。

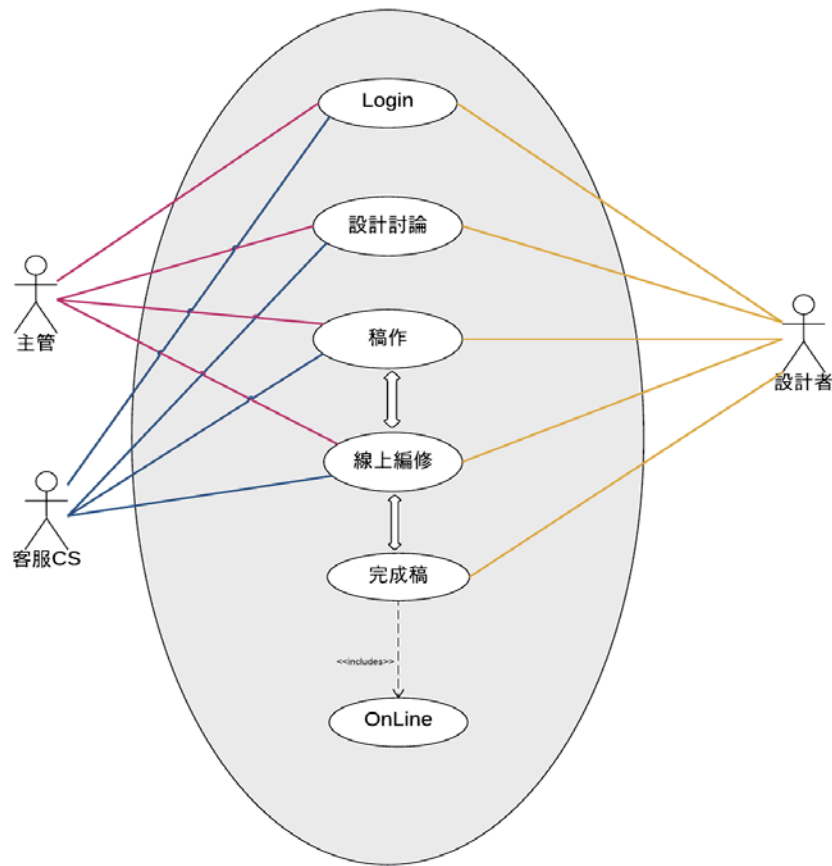


圖 18: Use Case Diagram

本研究中，將參與的人分為主管、客服人員及網頁設計人員，將工作事項逐一列出，而由主管督導整個專案運作，因為一個工作流程的順暢與否，是需要主管瞭解並參與，並由他來統籌人員的配合協調，所以主管在整個研究中，扮演著舉足輕重的角色。

客服人員在本研究中的名單裡，在於他們是直接面對客戶，無論是在電話中或是網站線上回覆客戶問題，為需求規格中第一線的重要人員，因此需要他們在本檔次的促銷專案裡，以使用者來檢視廣告文宣，是否能讓客戶接受。

而設計人員在會議中收集主管及客服人員所提供的資訊後，便能進入設計階段，且完成文宣的初稿圖，利用線上的協作平台進行溝通，讓主管及客服人員看到設計出來的文宣

圖稿，並針對其提出相關修改或確認，網頁設計人員可看到來自主管或客服人員即時反饋。

以下就針對這三個相關參與人員的層面，利用 UML[26,27]相關工具繪製 Use Case，並對工作項目加以各別說明。

### 3-2.3.1 主管 Use Case

針對主管需求規格而繪製，圖 19 為主管之 Use Case。

步驟一、由主管召開廣告會議：主管通知並召集本次參與會議的客服相關人員及網頁設計人員，一起商討本次促銷會議，廣告主題相關內容，並指示各部門對本次會議之決議內容進行瞭解及必要協助。

步驟二、提出本檔促銷廣告及相關議題：針對本次會議中推出的促銷廣告，提出想法及訴求，在網頁想要呈現的大致輪廓，並聽取客服人員彙總使用者反應經驗，提供做為文案需求規格參考依據。

步驟三、協作平台：利用網路上提供的協作平台，可以讓網頁設計人員將初稿圖上傳到協作平台，主管只要登入本身的 Google 帳號，即可看到網頁設計人員上傳的初稿圖。

步驟四、提出修改：當主管檢視過初稿圖時，若對設計有所意見及要求，可將修改部份直接線上輸入修改的文字說明，當網頁設計人員得知欲修改內容後，就能再次將修正過的圖稿，重新上傳到協作平台。

步驟五、主管會再對平台上的完成稿做審定，若還有意見或問題，就重複步驟四動作，直到主管確認後，由網頁設計人員將促銷廣告完成稿，上傳到公司資訊網站正式上線，此時，主管需指示客服人員做好應對演練，針對本次促銷內容對客戶進行明確陳述。



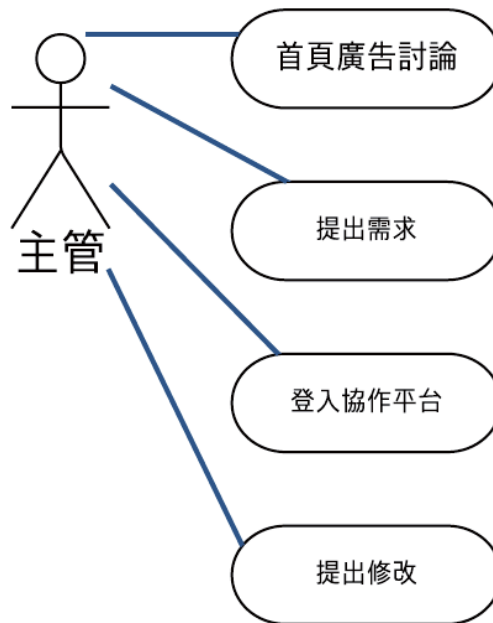


圖 19: 主管 Use Case

### 3-2.3.2 客服 Use Case

針對客服人員需求規格繪製，圖 20 為客服 Use Case。

步驟一、進入促銷會議討論：由主管指定客服相關人員，進入本次的會議程式，藉由客服人員是第一線直接面對使用者，也是最能瞭解顧客的問題，並站在使用者的角度，審視本次的促銷文宣，能否引起客戶的共鳴及參與誘因。

步驟二、提出需求：客服人員是使用者的重要橋樑，最接近使用者層面，也較能提出接近使用者需求心態的方案，怎樣的優惠及促銷方式是使用者想要看到，而將這樣的需求化為文字，提供給網頁設計人員在文宣上的參考，並針對本次促銷廣告，演練面對客戶的應對說詞。

步驟三、協作平台：同主管相同，利用網路上提供的協作平台，讓網頁設計者將文宣初稿圖上傳到協作平台，登入客服的 Google 帳號時，便可看到網頁設計人員上傳的初稿圖。

步驟四、提出修改：步驟如同主管，針對上傳的文宣初稿圖，提出意見及看法，也可將想要修改部份，直接線上輸入想要修改文字敘述。

步驟五、透過這樣線上協同作業，客服負責人員不用離開工作崗位，直接利用電腦網頁的協作平台進行溝通，省下來回設計部門討論的時間。

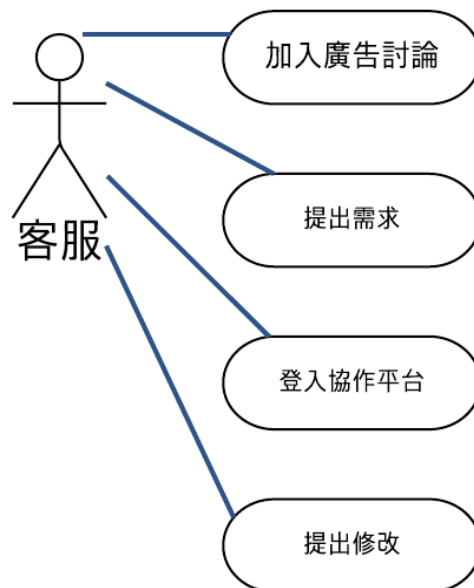


圖 20: 客服 Use Case

### 3-2.3.3 廣告設計人員 Use Case

針對廣告設計人員需求規格所繪製，圖 21 為設計者 Use Case。

- 步驟一、進入促銷會議討論：由主管召開此會議，網頁設計人員進入會議議程，先聽取主管針對此會議促銷主軸，綜合客服部門的相關建議，做為本次製作促銷廣告內容的參考依據。
- 步驟二、需求評估：在聽取主管對本次促銷廣告主軸及客服人員建議後，在本次促銷方案有了明確的需求認知，使其設計及開發時有更明確的方向，再結合本身的設計理念想法，可快速完成促銷初稿圖的成形。
- 步驟三、協作平台：將設計好的促銷完成稿圖上傳到協作平台後，通知主管及參與的客服人員登入到協作平台，並請他們針對本檔文宣設計上有需要修改的部份，直接在協作平台上輸入修改及確認的文字敘述。
- 步驟四、修改重新審核：若有進行修改時，網頁設計人員就依據協作平台上的格式，再對促銷內容加以修改，完成後再次將完成稿圖重新上傳，並再次通知主管及參與的客服人員登入協作平台進行確認。
- 步驟五、定稿上線：當主管及參與的客服人員都確認無誤後，這時網頁設計人員將確認的促銷廣告圖上傳到公司的官方網站，通知主管及客服人員，促銷活動正式開始，並請客服人員隨時留意並通報顧客的反應及意見，隨時可對促銷廣告進行修正。

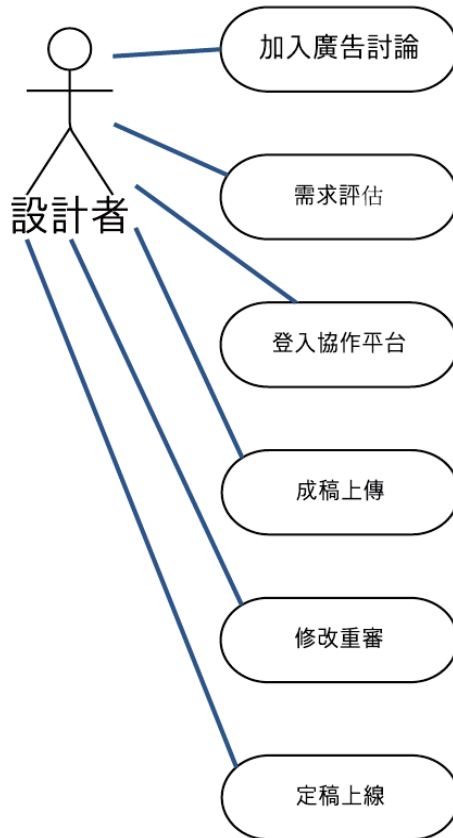


圖 21: 設計人員 Use Case

### 3-2.4 協作平台應用Moqups

基於 Agile Development 快速開發核心精神，我們利用線上的協作平台 Moqups，在本研究工作上擔任一個即時溝通中介系統。

Moqups 是一款以 HTML5 撰寫用來設計 WEB 框架、模型框線或是 UI 概念的線上服務，它內建了超過 60 套手繪的 SVG 範本及特性，Moqups 的所有物件皆已模組化，使用者只需要自由拉動拖曳組合排列，即可完成一款精緻的線框 UI 設計圖[29]。使其在開發過程中，有一個很便利的溝通平台，快速讓使用者知道設計人員的開發雛形，可在當下提出看法及修改建議，故在本研究中將以 Moqups 為溝通平台，便於多方的線上討論，如圖 22 表示 Moqups 網站。

經由各個單元流程及工作分析後，我們瞭解到如何縮短討論時間及來回各部門間的次數，需要有一個中介系統協助，可同步檢視網頁設計人員上傳最新完成的文宣圖檔，網路上有著許多線上的協作平台，都可讓我們免費加以利用，無需再自行架設協作平台系統，可以節省主機維護及人員的開銷。

本研究裡的協作平台是使用 Moqups，是一套方便好用的線上 Web UI 設計工具，用來快速產生網頁雛型，將大致網站完成後的樣貌，展示給客戶瞭解，並進行意見的雙向溝通。

在本研究的協作平台，只應用到 Moqups 部份功能，我們利用上傳圖檔的功能，將網頁設計人員的促銷廣告完成稿，在上傳圖檔後，利用 Share 功能，分享給主管及客服相關人員，他們利用彼此部門的 Google 帳號登入後，就可以看到網站設計人員上傳的促銷圖檔，若要修改可直接線上進行，輸入要修改內容文字敘述，在網頁設計人員看到要修改的部份，即可馬上進行修改作業，再將修改完成的圖稿，再次上傳請相關人員校審無誤後確認定稿，就能上傳本檔次的促銷廣告到公司官方網站。

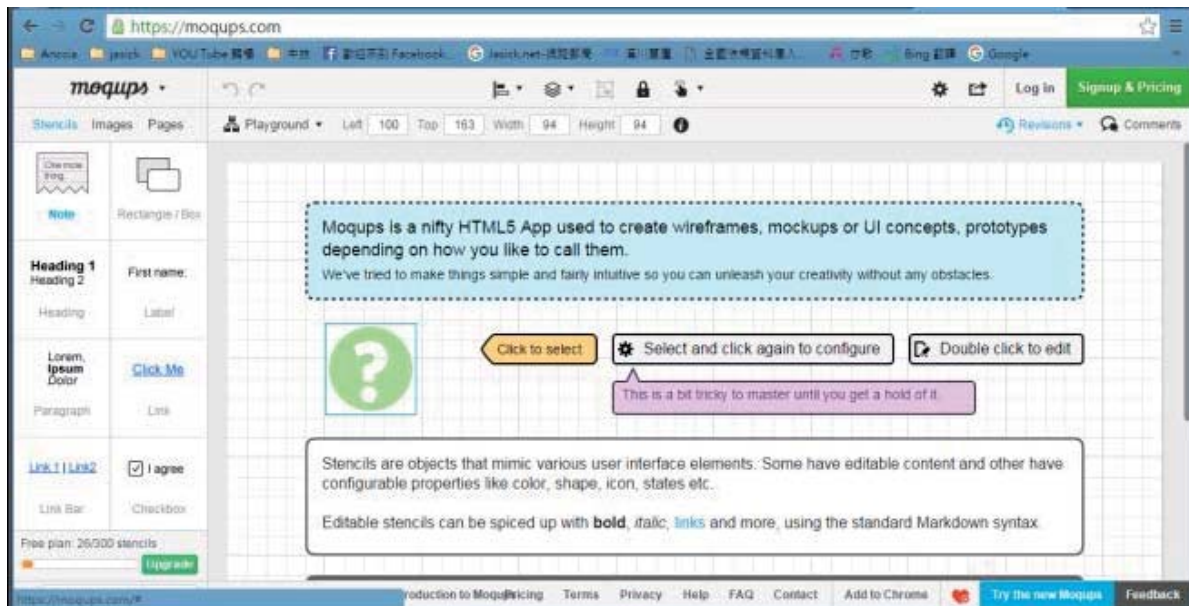


圖 22: Moqups 網站

資料來源：Moqups 網站(Moqups.com)

在經過本檔次的促銷廣告會議後，網頁設計人員將蒐集到的資料，快速的進行構思及設計，而接下來就利用一個實作案例，說明如何利用敏捷開發方法(Agile Development)的核心精神，並在快速開發過程中，不斷反覆討論加以修正問題，讓這廣告開發，在較短的時間內完成並上線。

### 3-2.4.1 步驟一：廣告初稿建立

網頁設計人員利用相關軟體完成本檔次的促銷廣告圖檔，將本次討論的廣告內容，在蒐集完成之後，馬上進入了實作階段，將相關文案在廣告設計稿中明確表達出來，並依先前制定好的廣告圖檔規格，完成促銷廣告初稿，如圖 23 為廣告初稿畫面。



圖 23: 廣告初稿

### 3-2.4.2 步驟二：登入到協作平台(Moqups)

公司使用的協作平台為 Moqups，如圖 24 Moqups 協作平台。平台登入方式，是搭配各部門的 Google 帳號為主，個別登入到同一個廣告專案中，由網頁設計人員發出協同要求信件到主管及客服的 Google 信箱，當主管及客服登入到 Moqups 協作平台時，便可看到由網頁設計人員上傳的廣告圖初稿，主管及客服人員可以在此對廣告促銷圖檔提出修改意見或認同。

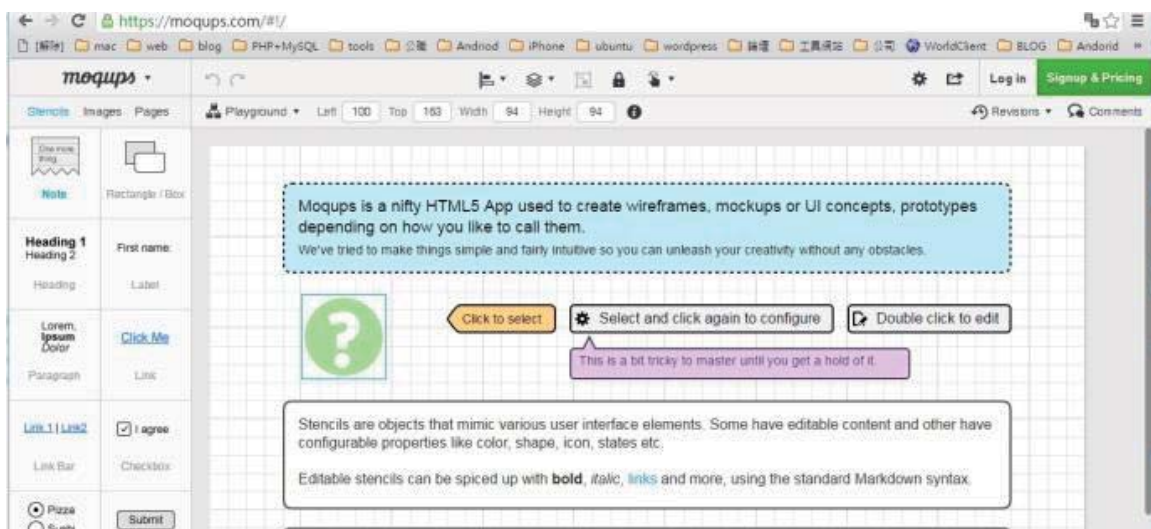


圖 24: Moqups 協作平台

### 3-2.4.3 步驟三：版面設定

網頁設計人員登入協助平台後，將本次會議中要製作的廣告版面尺寸，(本研究中的規格為寬 500 高 220 圖元)，在協作平台的【設定】裡，輸入圖片尺寸圖元，在上傳並載入圖檔後，便可完整檢視原尺寸的廣告促銷圖檔，如圖 25 版面設定畫面。

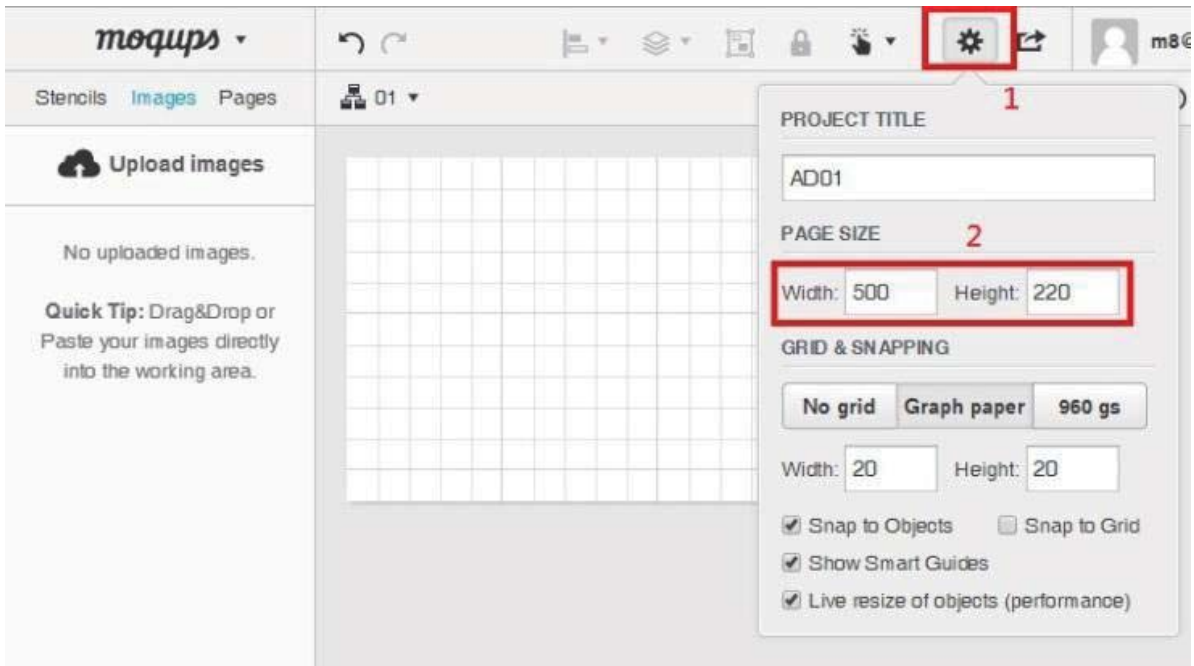


圖 25: 版面設定

#### 3-2.4.4 步驟四：載入廣告及分享

在Moqups 協作平台中,右上角顯示為目前登入者資訊,畫面左側功能表中【Images】分頁,點擊並將促銷廣告初稿圖檔載入,如圖 26 為載入圖檔畫面。

在載入後,左側下方則會看到載入後圖檔預覽圖示,雙擊圖案後,就會顯示在右邊版面區塊,呈現促銷廣告圖稿完整尺寸,並點選頁面中的右上方的【Share】功能選項,將這張促銷廣告圖稿分享給主管及客服人員的 Google 帳號,如圖 27 為圖檔分享畫面。

設定【Share】時,只需要填入相關分享者人員的 E-Mail 帳號即可,在 Moqups 預設 Google 帳號為免註冊帳號,只要利用 Google 帳號,就可免除重新註冊平台帳號,直接登入協作平台使用。並將這促銷廣告圖的使用權限設定為【Can Edit】,讓其他相關人員登入到這協作平台時,能有進行修改的權限,如圖 28 為分享設定畫面。



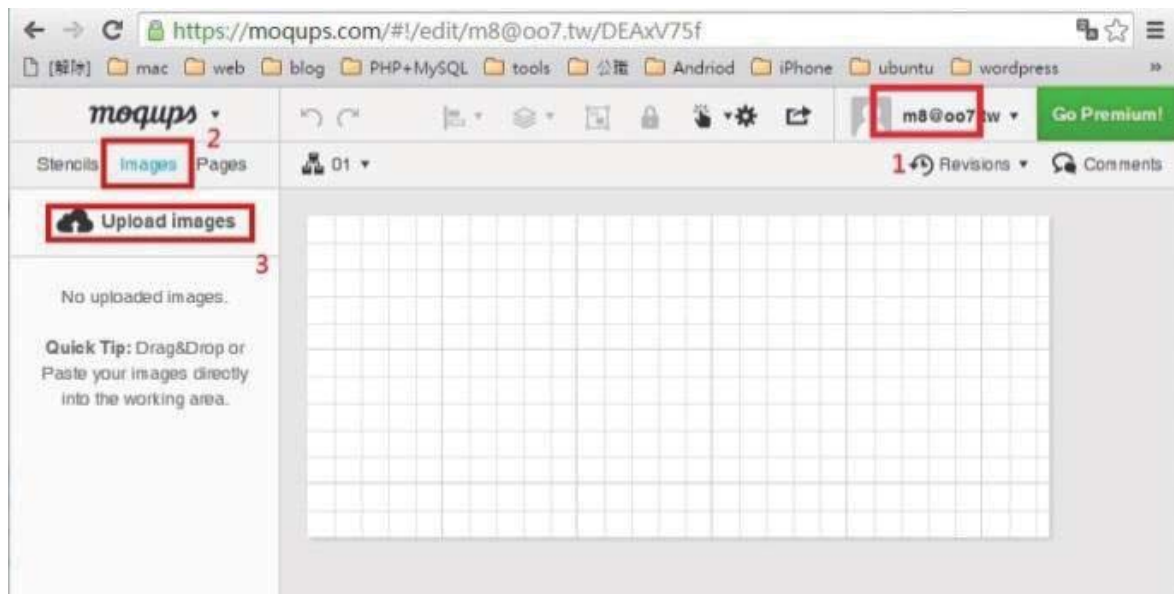


圖 26: 載入圖檔

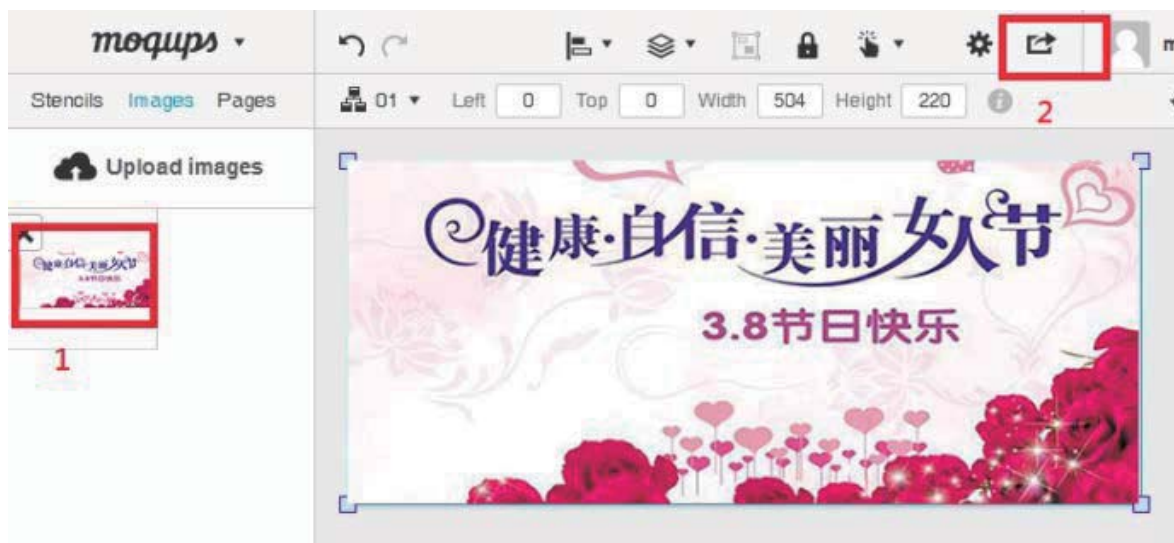


圖 27: 圖檔分享

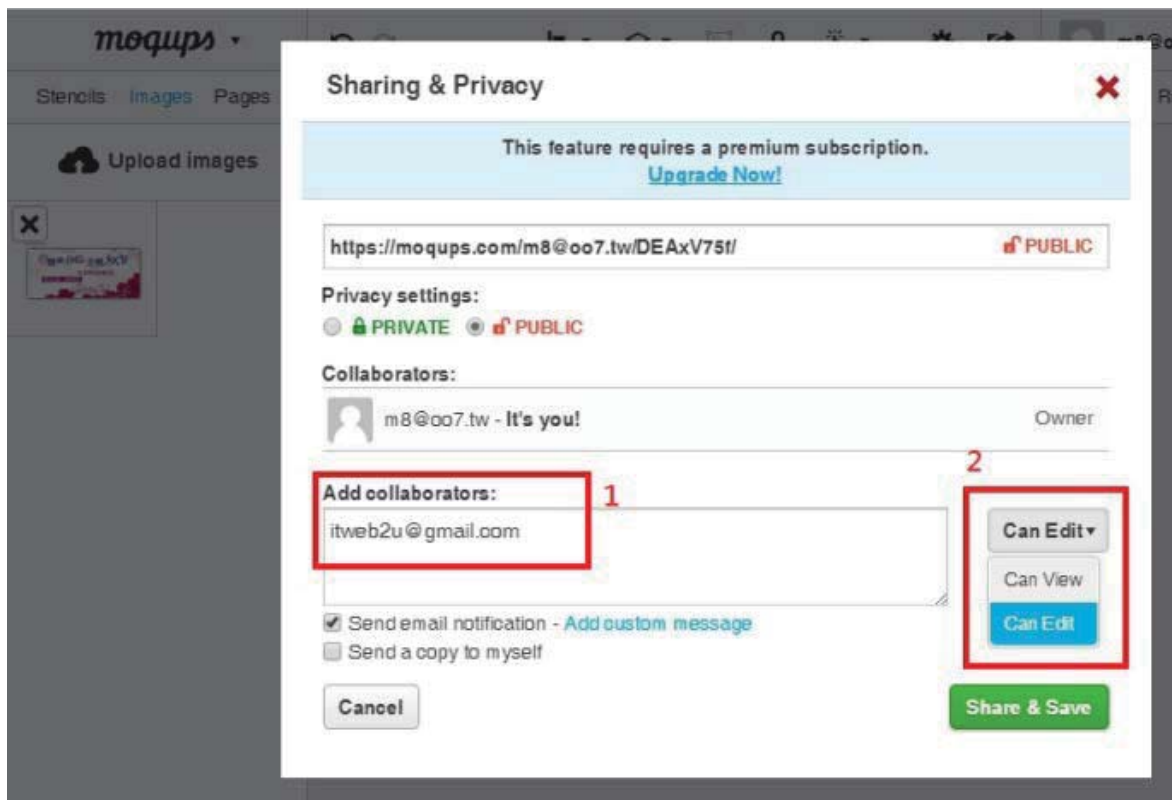


圖 28: 分享設定

### 3-2.4.5 步驟五：分享邀請

當網頁設計人員在協作平台發出分享邀請後，相關人員會在 Google 信箱收到邀請的連結，當相關人員在登入 Moqups 協作平台後，並點擊信箱中的邀請連結就可以被導引開啟視窗，並在畫面中看到由設計人員上傳的促銷廣告圖稿，如圖 29 為分享邀請畫面。

本研究實作中的帳號 m8 為網頁設計人員的帳號，發了一封協作邀請到主管的 itweb2u 帳號中，主管將利用這個連結，接受並開啟協作平台中的圖案。



圖 29: 分享邀請

#### 3-2.4.6 步驟六：即時編修與備註

主管或客服相關人員登入後，會同步看到與網頁設計人員相同畫面，若對文宣內容有任何問題，即可利用 Moqups 協作平台上的工具列中的文字輸入功能，在線上直接輸入修改意見或認同，如圖 30 為輸入修改意見畫面。

網頁設計人員在協作平台畫面中，即時同步看到要修改或改善的要求，就可馬上在自己工作平台上進行同步的修改動作。

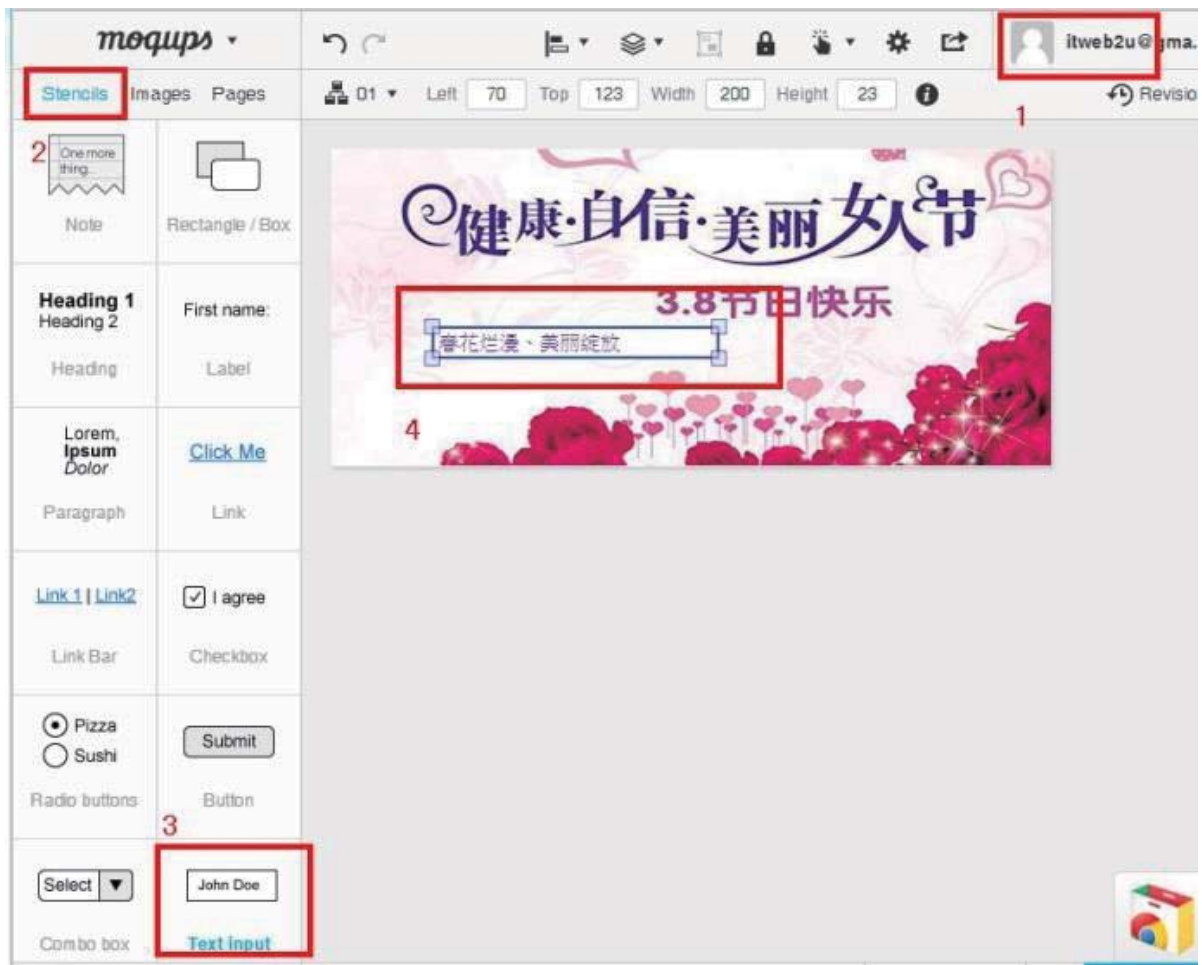


圖 30: 輸入修改意見

到此步驟，本研究建立一個使用者間的檢核點，讓使用中的主管及客服人員，就工作流程運作加以說明，其中他們皆表示很明顯是在反應的陳述上，不用往返網頁設計人員的部門溝通，可以直接在自己工作崗位上參與討論，主管在外地更可利用筆電或平板在線上看到實際成品並加以審閱，只要將需要修改的部份，記錄在協作平台上，三方都可以同時看到，在時間的控管上有了更有效的展現。

### 3-2.4.7 步驟七：重新上傳修改稿

網頁設計人員，針對修改的部份，在網頁設計人員的工作平台上進行重製作業，當確認無誤時，再次上傳到 Moqups 協作平台上，如圖 31 為修改重製畫面。

當重新上傳後，可以看到左側的預覽圖裡會呈現剛上傳最新版本的完成稿，在雙擊後就會在右側廣告版面出現，並可以利用這樣的方式，做為檢視修改前後的不同之處，也是一種修改記錄，如圖 32 為修改上傳畫面。



圖 31: 修改重製



圖 32: 修改上傳

在這步驟中，設計人員只要線上協同作業畫面中，看到修改的要求，可立即進行修改製作，這在以往是被動的情況下，可以掌控到流程的進行，可以在重新上傳後，讓參與者看到修改後的圖案，不再只是文字或電話上的溝通，較接近參與者的需求，相對也能減少等待時間。

#### 3-2.4.8 步驟八：重新上傳修改稿

當主管及客服人員看到更新版本，還是認為有需要再修改時，就重複步驟六的方式，加入要修改意見的文字敘述，若主管及客服人員都對本次的促銷圖稿認同時，則可在線上進行確認，如圖 33 為主管確認畫面。

網頁設計人員透過 Moqups 協作平台同步看到主管及客服人員，對於本次修改版本進行再修改或確認敘述，如圖 34 為確認訊息畫面。

若是確認無誤時，網頁設計人員即可將促銷廣告圖稿，上傳至公司官方網站，正式將這個活動推出。



圖 33: 主管確認畫面



圖 34: 確認訊息

經過這個一個流程下來，主管及 CS 表示在使用上都能感受到整體掌控上，相較之前在互相等待回應進度的情況下，流程進行得更為順暢，在整體的運作也較之前來的有準則效率，有效的改善彼此間的溝通及縮短作業時間，並加上有修改的相關記錄存在，讓參與的人都可以有查詢的記錄功能存在。



## 第四章 實作結果

### 第一節 效能表現

在專案工作進行中，皆利用如附件二中的表格，來記錄追蹤工作進度，而圖 35 就是在早期未做任何系統規劃下，完成工作的時間記錄表，一個專案大致都花費皆需在 5 至 6 的工作時數，在與 CS 部門及主管間來回，並等待主管回覆，部門間都需來回設計人員部門做確定工作，花費了較多時間。

### 工作進度記錄表

工作名稱	年終促銷案	專案日期	104/12/23
參與人	Jojo、CS、Jasick		
工作內容			
12/25 日前將廣告案推出			
日期	進度記錄	簽章	
12/23 13:29	初稿成完	Jasick	
12/23 15:47	CS 要求字體加大	asu	
12/23 16:22	Jojo 廣告尺寸再大一點	Jojo	
12/23 16:30	修改稿完成	Jasick	
12/23 18:08	主管確認	Jojo	
12/23 18:22	上線	Jasick	

圖 35: 未規劃前工作進度表

而利用有系統的規劃，將目前網站上的促銷廣告完成步驟逐步記錄，再應用 Web Engineering 以工程模組化方式來規劃開發，在整個作業流程運作上有著明顯的改變，如作業流程上的討論次數及時間，也讓開發者能快速反應在網站的呈現。附件三則是應用在這裡記錄整個工作進度時間表，而圖 36 則是將整個製作時所花費的時間，用明確圖表標示出來，有利與圖 35 中的工作進度時間表做一比較。

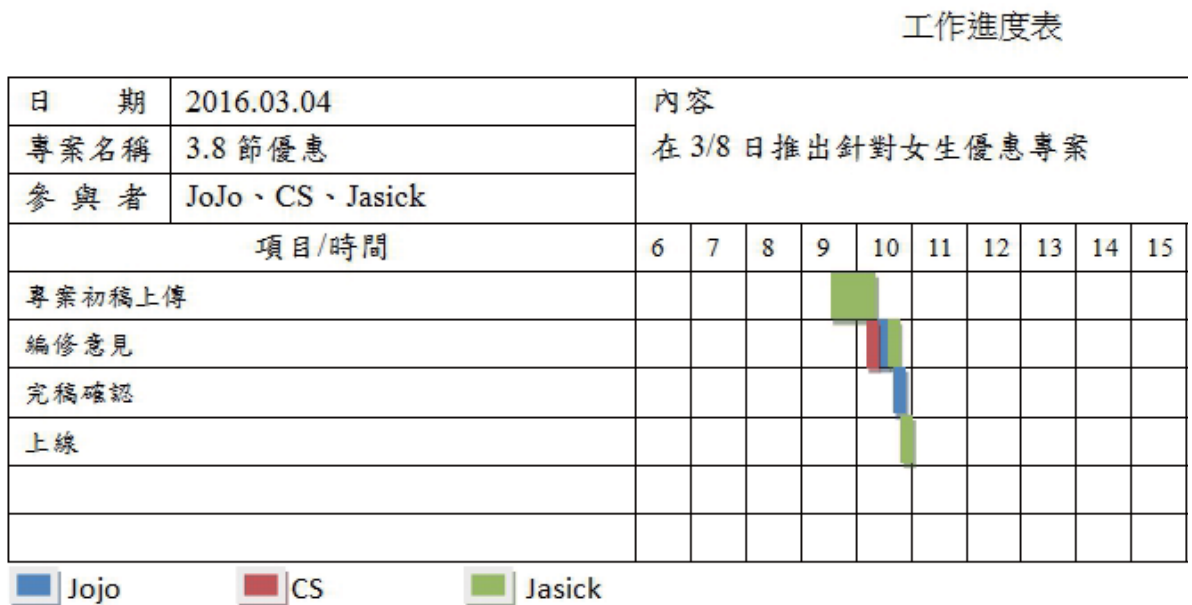


圖 36 規劃後工作進度表

根據這樣的數據，明確的說明，透過將網站開發，以工程化的有效規劃，是能達到快速開發及維護的目的，也可讓日後接手的工作人員，能儘速的進入工作領域中，簡化了交接上的程式。在整體表現上，無論在討論及反應的次數，乃至到完成上線的時間，都有很明顯的改善，除了有效讓網站廣告上線的時程縮短，也讓彼此溝通的時間更為省時，這是表現效能的最好說明。

表 5 中效能表現數據，是在未應用到 Web Engineering 規劃時，在完成廣告製作所耗費的時間及次數，並比較在加入 Web Engineering 工程化模組化後，完成作業的時間及次數，如圖 37 為效能表現圖。

表 5: 效能表現數據

	未模組	模組化
版面調整-顯示	3	1
版面調整-大小	6	1
字體文案調整	11	7
上線完成(h)	5	2

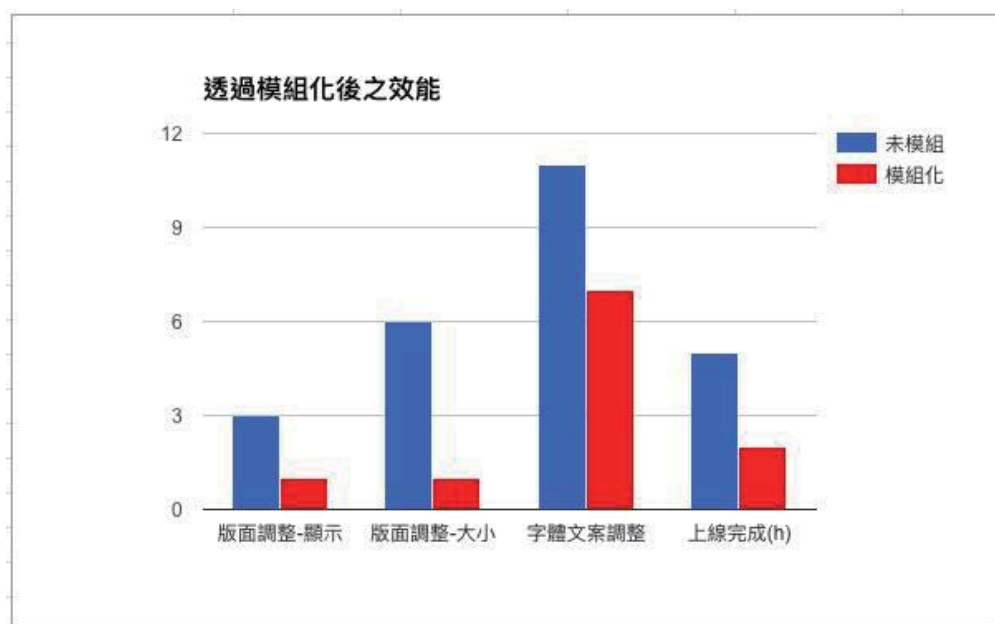


圖 37 效能表現圖表

## 第二節 協作平台表現

利用 Moqups 在協作平台上的應用，有付費及免費版本，本研究中應用是以免費版本為實作環境，在相關討論及互動上皆順手，唯在修改的版本控制上，時間戳記 (Timestamp) 是沒有的，若在這協作系統上能有這個選項，相信在時間上的控管會有更好的數據，日後在尋找類似的協作平台時，這會是一個需要考量的重要元素，如圖 38 為協作平台上修改的歷史記錄。



圖 38 協作平台上修改記錄

## 第五章 結論

### 第一節 研究貢獻

經本研究實作結果數據來看，經由 Web Engineering 有系統規劃，在日後網站的管理及更新，皆能有出色的效能表現，而建模開發模式，在整個系統也較具規格化，而在設計上的溝通解決方案裡，適時運用協作平台的功能性，讓各相關人員之間的溝通達到即時快速，也讓整個工作流程更加順暢，明顯縮短工時，進而提升了工作的成效。相較之前要完成一個促銷廣告案到上線，現在可以縮短到二至三個小時的時間。

本研究在實務上的做法，為了讓日後承接的設計人員，在一個具模組化依據下，能在短時間內掌控到工作內容，這也是 Web Engineering 在本研究擔任的重要主軸，而一個良好的 Web Engineering 系統規劃，更能彰顯在實務工作中，有效達到預期的效能表現。

在未來展望能透過這次的實作過程，讓 Web Engineering 在網站開發時被重視，因為太多網站開發到最後都以失敗收場，就是沒有正視網站開發的需求及後續支援，所以開發者與客戶之間的溝通，也都是要正視的問題，然本研究期許在喚起開發網站被重視的同時，對 Web Engineering 能有所涉獵，別讓網站開發只是一時之作，而是網站得以長遠性規劃，應對未來每一次挑戰。

### 第二節 研究建議

在本研究實作中，因為基於成本考量，利用現成的 Moqups 協作平台，雖有著更新的修改記錄，卻無時間戳記，故需利用手動去記錄完成的時間，若日後能在更新記錄中有時間戳記，相信在整个專案中，會得到更完善的時間管控，更可避免因人工作業上的人為因素，在論文的貢獻也更有參考價值。

## 參考文獻

- [1] 吳信輝, "Web Engineerin 簡介 (下) ," [Online]. Available: [http://newsletter.ascc.sinica.edu.tw/news/read\\_news.php?nid=1339](http://newsletter.ascc.sinica.edu.tw/news/read_news.php?nid=1339). [Accessed 18 9 2015].
- [2] 萬建成 & 鹿旭東, "Web 應用開發方法研究," [Online]. Available: <http://etc.edu.cn/articledigest38/web.htm>. [Accessed 11 4 2016].
- [3] "軟體工程(Software Engineering; SE) ," [Online]. Available: <http://irw.ncut.edu.tw/peterju/se.html>. [Accessed 11 4 2016].
- [4] G. Cormack, G. Griffiths, B. D. Hebborn, M. A. Lockyer and B. J. Oates, "Web Engineering: Methods and tools for education", in *SSGRR 2002s (Scuola Superiore G Reiss Romoli 2002 Summer Conference)*, L'Aquila, Italy, 2002.
- [5] G. Reif, H. Gall, and M. Jazayeri, "WEESA: Web engineering for semantic Web applications," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 722-729.
- [6] S. Ambler, *Agile modeling: effective practices for extreme programming and the unified process*: John Wiley & Sons, 2002.
- [7] K. Schwaber and M. Beedle, "Agilè Software Development with Scrum," 2002,
- [8] J. Highsmith III, *Adaptive Software Development*: Dorset House Publishing, 1996.
- [9] V. Jain and A. Kolambkar, "Modeling Web Attachment Storage for Web Applications," in *2014 21st Asia-Pacific Software Engineering Conference*, 2014, pp. 98-102.
- [10] H. Hiisilä and M. Kujala, "Combining Process Modeling and Requirements Engineering: An Experience Report," in *2015 IEEE 17th Conference on Business Informatics*, 2015, pp. 242-249.
- [11] S. Murugesan and A. Ginige, *Web engineering: Introduction and perspectives* vol. 1, 2005.
- [12] 阿捷, "專注微軟平台的商業智慧解決方案," [Online]. Available: <http://www.cnblogs.com/aspnetx/archive/2006/05/25/408583.html>. [Accessed 12 1 2016].
- [13] 賴森堂, "以量測模式提昇 Web 應用軟體的安全性," [Online]. Available: [http://sun.cis.scu.edu.tw/~epaper/newspaper/0013/speaking\\_971211.htm](http://sun.cis.scu.edu.tw/~epaper/newspaper/0013/speaking_971211.htm). [Accessed 14 2 2016].
- [14] S. Murugesan, Y. Deshpande, S. Hansen, and A. Ginige, "Web engineering: A new discipline for development of web-based systems," in *Web Engineering*, ed: Springer, 2001, pp. 3-13.
- [15] B. Donald, "UML 入門：統一建模語言入門," [Online]. Available: <http://www.dotspac.idv.tw/Jyemii/umlcolumn/articles/umlwriting/UMLBasics/UMLBasics.htm>. [Accessed 28 1 2016].

- [16] W. Sean TX, "UML, Unified Modeling Language," [Online]. Available: [http://tx.liberal.ntu.edu.tw/InfoMgt/Jx/IS\\_IM/uml.htm](http://tx.liberal.ntu.edu.tw/InfoMgt/Jx/IS_IM/uml.htm). [Accessed 16 1 2016].
- [17] 中科永聯高級技術培訓中心, "Sequence Diagram," 19 9 2013. [Online]. Available: <http://www.twword.com/wiki/Sequence%20Diagram>. [Accessed 17 1 2016].
- [18] iThome, "UML 類別圖的風格指南 (上)," 2 3 2009. [Online]. Available: <http://www.ithome.com.tw/node/53692>. [Accessed 17 1 2016].
- [19] W. Kenming, "從鳥瞰的觀點看 Use Case Diagram," 27 3 2005. [Online]. Available: [http://www.kenming.idv.tw/af\\_esyc\\_dc\\_es\\_er\\_c\\_use\\_case\\_diagram](http://www.kenming.idv.tw/af_esyc_dc_es_er_c_use_case_diagram). [Accessed 18 1 2016].
- [20] A. Qumer and B. Henderson-Sellers, "An evaluation of the degree of agility in six agile methods and its applicability for method engineering," *Information and software technology*, vol. 50, pp. 280-295, 2008.
- [21] L. Terence, "ProjectUp 專案管理生活思維," 8 1 2015. [Online]. Available: <https://www.projectup.net/article/view/id/15726>. [Accessed 9 1 2016].
- [22] "敏捷軟體的 12 個原則," [Online]. Available: <http://agilemanifesto.org/iso/zhcht/manifesto.html>. [Accessed 9 1 2016].
- [23] S. Murugesan and A. Ginige, "Web Engineering: An Introduction," [Online]. Available: <https://www.computer.org/csdl/mags/mu/2001/01/u1014.pdf>. [Accessed 11 1 2016].
- [24] T. Weilkiens, "Systems Engineering with SysML/UML: modeling, analysis, design," *Morgan Kaufmann*, 2008.
- [25] M. Grechanik, K. S. McKinley, and D. E. Perry, "Recovering and using use-case-diagram-to-source-code traceability links," in *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, 2007, pp. 95-104.
- [26] H.-E. Eriksson and M. Penker, "Business modeling with UML," *Business patterns at work*, 2000.
- [27] B. P. Douglass, "Real-Time UML: Developing Efficient Objects for Embedded Systems," 1997.
- [28] R. S. Pressman and L. David, "Web Engineering: A Practitioner's Approach," *McGraw-Hill Higher Education*, 2009.
- [29] SR 數位設計, "7 款超強大的線上 UI 設計工具 - 設計師必備!!!," 15 5 2013. [Online]. Available: <http://yesdesigning.com/thread-117-1-1.html>. [Accessed 19 1 2016].
- [30] S. Michael, "How to Hire the Right App Developers," [Online]. Available: <http://clearcode.cc/2014/08/hire-right-app-developers/>. [Accessed 21 1 2016].
- [31] 黃文奇, "專案管理使用敏捷開發," FineArt, [Online]. Available: <http://www.fineart.com.tw/index.php/ch/94-fineart-express/fineartinsight/361-veitnam-s-p-30898>. [Accessed 23 1 2016].
- [32] MBAlib.com, "瀑布模型," [Online]. Available: <http://wiki.mbalib.com>. [Accessed 13 2 2016].

- [33] MBAlib.com, "快速原型模型," [Online]. Available: <http://wiki.mbalib.com>. [Accessed 13 2 2016].
- [34] OSChina.Net, "常見軟件開發模型及其優缺點," 21 11 2010. [Online]. Available: <http://my.oschina.net/xyxld/blog/10147>. [Accessed 18 2 2016].



附件一

### 網站廣告建置、修改申請單

申請單位				
申請日期				
目的				
申 請 事 項	擬 辦 事 項			
<input type="checkbox"/> 內容修正 <input type="checkbox"/> 內容更新 <input type="checkbox"/> 版面修正 <input type="checkbox"/> 版面更新 <input type="checkbox"/> 網頁增建 <input type="checkbox"/> 網頁刪除 <input type="checkbox"/> 其他				
修 改 內 容 說 明				
填表人或 申請人	申請單位 二級主管	申請單位 一級主管	資訊室 承辦人員	資訊室 主任

附件二

### 工作進度記錄表

工作名稱		專案日期	
參與人			
工作內容			
日期	進度記錄		簽核

工作進度表

日期	內容																								
		6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5
專案名稱																									
參與者																									
	項目/時間																								

 Jojo

 CS

 Jasick