

以 WBS 為基礎的品質量測模式提昇軟體專案可變動性

賴森堂

實踐大學資訊科技與管理學系

E-mail: stlai@mail.usc.edu.tw

摘要

軟體開發過程中，專案計畫將持續遭受需求變更、錯誤修訂、技術與環境演進以及資源調整等事件的衝擊，計畫一旦無法適時變更與調整，勢必遭遇許多困境，最後導致專案的失敗。WBS (Work Breakdown Structure)是規劃與制訂專案計畫的關鍵，WBS 必須具備高可維護品質，專案計畫才能克服各種變動事件的挑戰，有效降低軟體開發風險。本文針對軟體專案計畫的變更事件進行探討，並剖析與蒐集 WBS 維護品質影響因子，進而提出一套 WBS 為基礎的品質量測(WBS-based Quality Measurement; *WBSQM*) 模式，協助找出 WBS 維護品質問題與缺失，進而衍生出修正與改善作業，具體提昇軟體專案可變動性且降低軟體開發風險。

關鍵詞：品質量測模式、WBS、維護品質、軟體專案、軟體開發

以 WBS 為基礎的品質量測模式提昇軟體專案可變動性

壹、緒論

軟體專案管理計畫(Software Project Management Plan; SPMP)主要著重於資源、人力與時程等項目的規劃(Schach 2008)，而這些規劃項目則取決於完整的需求分析與精確的成本估算，能夠精確估算出軟體的成本，才能具體且有效達成經費投入、時程安排、資源配置與品質管控等規劃項目(Boehm 1981; Pressman 2010)。SPMP 是軟體開發過程中，必備的關鍵文件，經營的運用、時程的掌控、資源的規劃以及品質的管理都是依據計畫書的規劃內容來執行，當軟體系統的需求規格確定後，第一版的 SPMP 也隨之誕生，不過，SPMP 必須配合軟體系統的開發作業，持續不斷的進行調整與變更，新版本的 SPMP 將不斷被提出，直到軟體專案結束或被終止。下面四項關鍵事件是促使 SPMP 必須不斷調整與變更的主要原因：

- (1) 開發初期，分析人員蒐集與分析的資料不夠完整，導致錯誤、不一致的需求規格。
- (2) 資訊技術與運作環境的演進，造成計畫書規劃的內容不符合環境的需求。
- (3) 使用單位針對已確認的系統需求規格，提出需求新增、修改或刪除等要求。
- (4) 資源規劃項目可能遭遇人員流動、延期交貨或設備故障等狀況的影響。

依據分析軟體開發專案的知名顧問公司 Standish Group 對 9236 軟體專案，於 2004 年的剖析報告中，指出只有 29% 的軟體專案是成功的，18% 於執行過程中被取消的，而剩餘的 53% 是屬於延後完成、超出預算或是功能未能滿足需求等狀況(Hayes 2004)。探究失敗軟體專案的主要原因就是專案計畫變動過於頻繁，軟體開發過程中，許多衝擊軟體專案運作的變動事件將陸續發生，迫使軟體專案所規劃的項目必須不斷變動，更動後的專案規劃項目卻無法及時且有效的反映在預算規劃、開發時程、產品品質及資源配置等重要項目上，使得每次的計畫更動就拉開規劃項目與軟體開發作業之間的距離，漸漸導致專案計畫完全無法有效控管預算執行、開發時程、產品品質及資源配置等項目的運作，最後成為失敗專案(Boehm 1991; Boehm 1989; Fairley 1994; Schach 2008)。為此，要能夠應付更動頻繁的專案計畫，SPMP 必須具備高度的可變動性(Changeability)，方能有效降低計畫變動所帶來的風險。

WBS (Work Breakdown Structure)是規劃與制訂大型專案計畫的關鍵核心，WBS 必須具備高可維護品質，專案計畫才能克服各種變動事件的挑戰與衝擊。在 SPMP 擬訂過程中，全面蒐集 WBS 可維護品質影響因子，且深入剖析 WBS 維護品質潛在的問題與缺失，及時提出修正作業及改善活動，將可大幅降低計畫變動的失敗風險。量化的量測機制是改善軟體品質的關鍵(Weller 1994)，本文針對衝擊專案計畫修訂與調整的事件進行探討，剖析修訂過程的品質影響因子，且以查核表的審查與檢視作業，蒐集且量化品質影響因子，進而提出一套 WBS 為基礎的品質量測(WBS-based Quality Measurement; WBSQM)模式，以協助找出 WBS 於變動維護作業的問題與缺失，且衍生出問題與缺失的改善作業，以具體提昇 WBS 的可維護性且降低軟體開發風險。本文第二節將說明軟體規劃過程中，WBS 協助完成的重要任務，且深入探討造成軟體專案高失敗率的主要

變更事件。量化品質特性是標示管理計畫書問題與缺失的關鍵，第三節將剖析影響 WBS 可維護性的關鍵品質特性及其影響因子。不同的影響因子隱含在管理計畫書各個不同的層面中，第四節將討論如何有效且具體的蒐集各個層面的品質影響因子以及影響因子的正規化方式。第五節將以線性結合模式為基礎，提出一套 WBSQM 模式，協助標示缺乏可維護性的問題與缺失，進而衍生出改善措施及修正作業，具體降低軟體專案潛在的變動風險。第六節將說明以 WBSQM 模式提昇專案計畫可變動性、降低專案開發風險等優勢，且針對本文主題作個結論。

貳、WBS 的重要性與軟體專案失敗的關鍵

大型且複雜的軟體專案完成 WBS 後，仍舊必須面臨變動事件的挑戰，WBS 無法配合軟體系統開發作業持續修訂與調整，將導致軟體專案失敗的關鍵。

一、WBS 的重要性

軟體專案計畫是依據使用單位對於軟體開發的經費預算、時程要求、產品品質需求以及與資源配置等項目，擬訂出軟體開發步驟、階段時程排定、組織架構及資源規劃等，初版的 SPMP 於需求分析完成後被提出，用以確保軟體專案開發作業，能夠依計畫書所擬訂的項目逐一進行，且滿足使用單位所提出的各項需求。為了具體且有效的擬訂需求項目步驟、工作時程排定、組織層級架構及各項資源配置等計畫內容，大型且複雜的軟體專案必須進行適度的劃分且切割，才能將計畫任務單純化、步驟明朗化、成本公式化及資源實質化，工作分解結構(WBS)是大型且複雜的軟體專案進行工作規劃的關鍵作業，WBS 是一套階層式圖示的產品導向方法(Mantel, Jr. 2011)，將專案產品進行完善的細分與切割，細分後的工作項目必須具備高度可管理的特性，以利於後續相關作業的規劃，順利達成工作項目的時間安排、成本估算、資源配置、方法與技術引用及風險管理等。WBS 分解後的工作項應該具備明確任務、具體範疇、與其他工作項的先後關係、與其他工作項的引用關係、易於擷取的估算參數等特質，說明如下：

- 複雜專案分解成可管理的工作項：專案計畫經過 WBS 多層次分解後的低階子工作項，大幅降低其複雜性，且提昇子工作項之間的相互運作關係，具體增強子工作項可管理的特性，有效協助專案計畫後續整合性規劃。
- 易於排定工作項的順序：細分後的工作項目具備明確的任務與目標，對於低階層次的子工作項或高階層次的工作項目都能夠適切的安排工作時程與先後順利，以協助專案計畫資源與預算的配置規劃。
- 易於管控工作項的品質：單純化的子工作項可以蒐集運作過程中的相關數據，也利於擷取特性的品質影響因子，可以適時監督與管理子工作項與工作項目的作業效率與相關品質。
- 易於估算工作項的成本：軟體成本估算的方式非常多樣化，如 Function Point, COCOMO 模式、專家判定法則及類比法等(Boehm 1981; Schach2008)，無論採取那一種成本的估算方式，單純化且任務明確的子工作項都可以有效提高估算的精確度，此外估算出的成本費用也可以順利轉換成時程、資源、品質與預算等估算值。

WBS 的工作項最後採取由下而上的整合步驟，逐層將工作項彙集成整體軟體專案

的開發經費、時程、品質、建構以及資源等五個管理子計畫(參閱圖 1 所示)，各個管理子計畫以 WBS 為基礎，配合開發過程的各種變動衝擊，及時且完整的修訂與調整計畫內容。

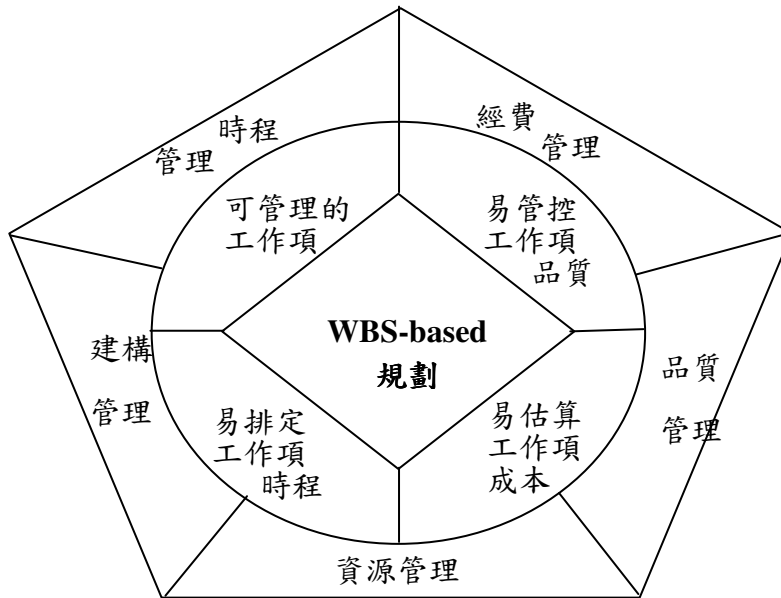


圖 1 以 WBS-based 規劃核心的軟體專案計畫架構圖

二、軟體專案失敗的關鍵因素

依據 Standish Group 近十年對軟體專案進行大規模的調查，發現軟體專案的成功率只有三成左右，而失敗的專案中，則有 80% 是遭受時程延後、預算不足及品質不符需求等問題的困擾(賴森堂 2005; Hayes 2004; Eveleens and Verhoef 2010)。軟體系統面臨的開發高度風險主要來至專案時程延後、預算不足及品質不符需求等狀況，而形成高風險的原因則與下面四個關鍵事件息息相關：

- (1) 開發初期，系統分析與需求分析人員蒐集或分析的資料不夠完善，導致系統需求規格書與使用單位實際需求之間存在許多不正確、不完整與不一致的落差。這些落差若無法及時修正，需求規格的問題與缺失將隨著開發作業遞延至後續階段且持續擴大，當使用單位於檢視作業發現問題與缺失時，更正要求勢必會被提出，隨之造成開發時程遞延、預算須追加及品質不符需求等狀況，間接導致軟體專案失敗風險。
- (2) 軟體開發過程中，經常受到新技術的發表或運作環境的改變，使得計畫書原先所規劃的內容無法完全符合新技術或新環境的運作。這些不符合現況的規劃項目若無法及時且有效更正，勢必影響後續階段的作業，進而提高軟體專案的失敗風險。
- (3) 使用單位在開發初期所提出的各項需求規格，經常隨著開發作業的進展而要求更動，使用單位提出的更動內容包括需求項目的新增、修改或刪除等作業，這些更動要求對於後續階段作業將形成重大的衝擊，SPMP 若無法配合更動內容及時變更與調整，軟體專案勢必會發生時程延後、預算不足或品質不符需求等狀況，而傾向失敗之途。
- (4) 計畫執行的各個階段都需要配置適當的資源，包括開發人員、硬體設備、軟體工具以及運作環境等，資源配置可能遭遇人員流動、設施延期交貨、設備故障或老舊更

換等狀況影響，必須重新規劃與調整。

這四項關鍵事件的衝擊是軟體開發過程中無法避免或排除的，對於專案已完成規劃的工作項目勢必造成不同程度的影響，更是專案失敗的主因。因此，為了降低失敗專案的風險，WBS 必須具備高度的可維護品質，配合軟體系統開發作業遭遇的變動事件，持續不斷的進行修訂與調整，直到軟體專案結束或被終止。有關不完善的需求分析、技術與環境的演進、使用單位變更需求及資源配置調整等四項關鍵的變更因素，造成專案計畫的經費管理、時程管理、品質管理、建構管理與資源管理等規劃項目受到的影響程度(參閱表 1 所示)說明如下：

- 不完善的需求分析：一般都屬於需求項目被遺漏，SPMP 為彌補被遺漏的需求項目，在經費、時程及品質等方面的規劃管理受到的衝擊較大，至於建構與資源的規劃管理受到的衝擊較有限。
- 技術與環境的演進：導入新的開發技術或運作環境，需要花費額外的購置成本及人員訓練費用，開發人員也必須花一段時間進行試用與熟悉，因此對於經費、時程等方面的規劃管理影響較大。
- 使用單位變更需求：在開發過程中，使用單位無預期的提需求變更提案，勢必對於已完成的開發文件與規劃項目造成影響，經費、時程及品質等方方面的規劃管理受到需求變更的衝擊較大，必須花較多時間重新調整，以配合變更需求的相關活動。
- 資源配置調整：計畫變動的影響，時程必須往後遞延，造成人員流失、組織縮編，或是設備延期交貨、故障、更新等狀況影響，為了避免時程延誤過長，迫使資源必須重新配置，其中又以經費、時程及資源等方方面的規劃管理受到較大的衝擊。

表 1 變動事件造成專案規劃項目的影響程度

變動事件 \ 受影響項目	經費管理	時程管理	品質管理	建構管理	資源管理
不完善的需求分析	***	***	***	*	**
技術與環境的演進	**	**	**	*	*
使用單位變更需求	***	***	***	**	**
資源配置調整	**	**	*	*	***

*註：*** 影響很大、** 影響中等、* 影響較小

參、WBS 維護品質的影響因子

SPMP 是軟體專案規劃與開發的重要依據，軟體開發過程中所出現的各種變動事件，經常是導致專案失敗的關鍵，其主要原因就是 SPMP 不能配合變動事件進行及時修訂與調整，使得 SPMP 無法持續且有效引導變更後的軟體專案之運作，因而導致專案的失敗。WBS 維護品質對於 SPMP 可變動性具有高度影響力，高可變動性的 SPMP 能夠適時且具體的配合變動事件進行修訂與調整，持續引導變更後軟體專案的運作。強化 WBS 維護品質是降低專案失敗風險的關鍵，依據 McCall 的因子模式論述(Galin 2004)與 ISO 9126 品質架構(ISO 9126 1988)，可以得知軟體品質特性是具有層次分明的結構，可變動性在 ISO 9126 品質架構中，是屬於軟體可維護性的一項重要指標(ISO 9126

1988)，軟體產品或文件能夠進行完善變動或修改，最基本的要求就是產品或文件必須具備正確性、完整性、一致性及可讀性等四項基層品質。軟體設計階段要求設計文件必須具備高模組性(Modularity)，主要目的就是為了提昇軟體的可維護性，將高模組性融入的 WBS 中，同樣可以降低修改過程的漣漪效應，避免牽一髮而動全身的修改困境。減少修改過程經常會出現的遺漏現象，可以改善修改後的文件或產品出現不完整與不一致的內容，可追溯性(Traceability)也是軟體可維護性的一項關鍵指標，WBS 中所擬訂的各個工作項目，若能以交互參考引用表明確定義各管理項目之間的相互引用關係，一旦高層次規劃項目需要變動或修改，就可以完整且快速的對映出其他需要修改的工作項目與內容，達到修改作業的完整性與一致性。影響 WBS 維護品質可以從基本品質特性(Basic Quality)、模組性、可追溯性及再估算等四個層面進行探討：

(1) WBS 規劃文件必須具備基本品質特性以提昇其可維護性，基本品質特性涵蓋正確性(Correctness)、完整性(Completeness)、一致性(Consistency)及可讀性(Readability)等四項基層品質，以下針對四項基層品質說明之：

- 正確性：正確的 WBS 規劃文件是基本的品質要求，卻很重要，錯誤過多 WBS 文件不僅將影響後續調整與修改活動的效率，更會造成修改後的 WBS 存在更多的問題與缺失，持續衝擊後續的開發作業。
- 完整性：任何開發文件與規劃文件都必須具備高度的完整性，特別是 WBS 規劃文件，不完整的 WBS 文件一旦進行調整與修改活動，將擴大不完整的現象，對於後續開發作業造成難以預期的影響。
- 一致性：為了協助 WBS 有效且及時達成修訂與調整的能力，WBS 規劃文件相互之間的引用內容必須具備高度的一致性，才能確保 WBS 修訂後的品質。
- 可讀性：具備高可讀性的 WBS 文件，才能有效協助軟體專案進行及時修訂調整與修改，WBS 規劃文件應該依照指引與格式要撰寫求，且融入簡潔、明確等可讀性的特質。

(2) 模組性是 WBS 強化可維護能力的重要特性，模組性涵蓋耦合度(Coupling)、凝聚力(Cohesion)及複雜度(Complexity)等三項基層品質，以下針對三項基層品質說明之：

- 耦合度：WBS 分解後的工作項目應該盡可能減少與其他工作項目之間的關聯性，以提昇工作項目的獨立性，且降低 WBS 進行修訂與調整的漣漪效應。
- 凝聚力：WBS 分解過程中所細分的子工作項，應將具備高度相關性的子工作項彙集在一起，且排除低相關性的子工作項，以提昇工作項目的整合能力，增強 WBS 進行調整與修改的效率與完整性。
- 複雜度：WBS 分解後的工作項目進行規劃、定義與描繪時，應該適度進行簡化作業，將不易理解或過於複雜的項目再細分為子工作項，有效提昇規劃項目的明確性、單純化與易於瞭解，以降低工作項目目的複雜度，進而減少 WBS 進行修訂與調整的阻礙與時間花費。

(3) 為了快速且完整的修訂 WBS 的變更內容，WBS 必須具備高度的可追溯性，WBS 可追溯性可以從單層次與跨層次的工作項目排定關係，及單層次與跨層次的工作項目交

互關係進行蒐集，以下針對四項基層品質說明之：

- 單層次工作項目排定關係：WBS 中歸屬同一層次的工作項目，應該有一套完善的機制描述其間的先後關係，詳實記載跨層次工作項目的排定關係，以利於單層次工作項目的修訂與調整。
- 跨層次工作項目排定關係：歸屬不同層次的 WBS 工作項目，應該有一套明確且完整的工作排程方式，詳實記載跨層次工作項目的排定關係，以協助變更事件後，跨層次工作項目的修訂與調整。
- 單層次工作項目交互關係：WBS 規劃文件必須配合各種變更事件進行調整與修改，單層次工作項目交互關係是確保調整與修改後同一層次文件的完整性與一致性。
- 跨層次工作項目交互關係：跨層次工作項目交互關係是確保調整與修改後不同層次文件的完整性與一致性。

(4) 修訂或調整後的 WBS 工作項目，必須針對各種費用進行重新評估，以確保成本、時程、品質及資源等數據的正確性，以下針對影響重新評估作業的三項基層品質說明之：

- 參數資料取得方式：費用重新估算所採用或引用的參數資料應該具有明確性且易於規劃文件擷取。
- 明確簡潔估算公式：費用重新估算所採用的估算公式，應該盡可能具備簡單化、高精確度、易於瞭解且可剖析等特性。
- 其他費用轉換方式：估算公式一般只計算其中一項費用(如開發成本)，透過單純、明確且一致的費用轉換方式，可以快速且正確的轉換成其他的費用數據。

影響 WBS 可維護性的四個層面品質特性一共涵蓋了十四項基層品質，這十四項基層品質與軟體專案規劃項目存在相互的影響關係，深入剖析相互的關係可以協助可維護性影響因子的蒐集與評估，圖 2 即為影響 WBS 可維護性的四項關鍵品質架構圖。

肆、WBS 維護品質影響因子的蒐集與正規化

不同的影響因子隱含在 WBS 規劃文件不同的層面中，如何有效且具體的蒐集各個層面的可維護品質影響因子，是 WBS-based 品質量測與評估的重要依據。

一、WBS 品質影響因子的蒐集方式

從表 1 的專案變更事件與規劃項目受影響程度表，及圖 2 的四項關鍵特性及十四項基層品質之架構圖中，可以瞭解維護品質影響因子是隱藏在管理計畫書的各個層次中，這些層面的影響因子可以透過內部人員的文件審查以及使用者參與的正式檢視(Formal Inspection)(Pressman 2010; Schach 2008)等活動中，蒐集且獲得重要資訊。軟體系統在開發過程中，確認與驗證(Verification and Validation)是確保產品符合使用單位需求且達到一定品質的關鍵作業(Pressman 2010; Schach 2008)，每一階段或重要步驟完成前，都必須透過檢視與審查活動來確保產品需求符合度與融入的品質，以查核表協助檢視或審查活動找出產品的問題與缺失是一種可行性與有用性的方式。其中，查核表(Checklists)的

設計與制定，可以協助品質管制人員蒐集更完整且正確的資料(Galin 2004)。查核項目是協助品質管制人員能夠快速且確實找出 WBS 維護作業問題與缺失的關鍵，規劃完善的查核項目可以引導品質管制人員以最短的時間蒐集重要的維護品質影響因子，進而協助找出潛在的問題與缺失，且引導專案規劃人員明確的改正方向。不過，查核項目的內容並不是固定不變的，必須配合開發的方法、軟體專案的應用領域、人員的經驗及品質層次的要求等項目進行適度的調整，才能夠協助品質管制人員，確實找出 WBS 維護作業品質潛在的問題與缺失。

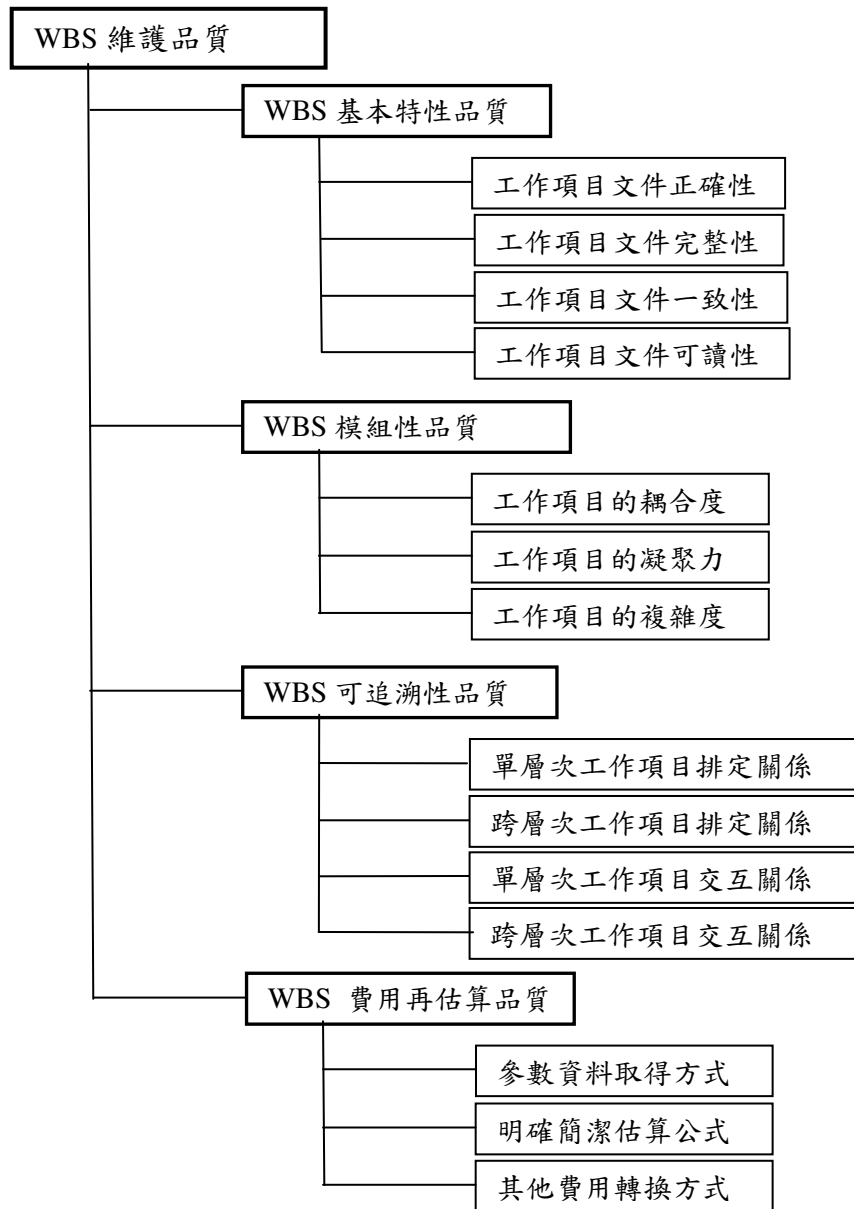


圖 2 影響 WBS 維護品質的四項關鍵特性架構圖

維護品質影響因子是多樣化的，系統分析與專案計畫規劃過程中，記錄的歷史數據也是影響因子蒐集的重心，大量的歷史數據則不是透過查核表單可以協助蒐集的，因此統計分析工具成為蒐集歷史數據的重要方式。從過去的經驗、歷史的數據以及專家的知識可推論出重要的計畫變動風險與因果關係，也是評斷計畫變動導致專案失敗的關聯

性，更可以剖析出計畫變動受到不會層級品質特性影響的強度。

二、品質影響因子正規化

個別或低階的維護品質因子無法具體表現其影響力，必須將個別或低階的影響因子適度的結合才能現維護品質因子的影響力，因此高階且具體的品質項目量測值都是由一些低階的維護品質因子結合而成(Boehm 1981; Conte 1986, Lai 2000; Lai 1998)，其中每個查核項目的評分就相當於一低階影響因子量度值，這些低階量度值依其影響度及發生率，又可設定成不同的參數或權位值，接著再配合特定的結合公式就可計算出高階品質項目的量測值。利用分析工具所蒐集到的影響因子量度值，也可以依同樣的方式計算出特定品質項目的量測值，因為個別的量度只能評量某項品質的單項特質，為了量測整體維護品質，必須將個別的量度做適當的結合。量度結合的方式可以分為線性結合(Linear Combination) (Conte 1986; Fenton 1991; Lai 2000; Lai 1998) 與非線性結合(Nonlinear Combination) (Boehm 1981; Conte 1986; Fenton 1991)，其中線性結合的特質包括：思維型態較容易被接受、公式化的步驟較單純、具備較高的調整彈性及適合資料蒐集量較少之應用等。非線性結合的特質包括：能夠較精確的表現量度結合的效果、公式化的步驟較複雜且需要工具的支援、需要資深的統計技術人員的協助及適合資料蒐集量較少之應用等，線性結合與非線性結合主要特質比較內容請參閱表 2。Boehm 所提出的 COCOMO 成本估算模式 (Boehm 1981) 就是採用非線性結合，太複雜是使用過此模式人員的共同感受。量化品質特性是持續改善軟體重要品質的關鍵作業，考量量度結合方的實用性、修改彈性與簡單性等特質，本文選擇以線性結合方式來建立品質量測模式。不同層面的影響因子有不同的量度表達方式，因此在進行線性結合之前，必須將相關的風險因子量度值正規化(Normalization)，正規化的作業必須考量影響因子的發生機率、影響程度及蒐集到的量度值，優質的影響因子量度值正規化後趨近於 1，劣質的影響因子量度值正規化後趨近於 0，再配合事先設定的權位值及結合公式可以將低階且具高度相關性的量度結合成基層及高階的特性項目量測值。

表 2 線性結合與非線性結合主要特質比較表

度結合模式 特質	線性結合	非線性結合
精確度	中等	高
公式化	步驟單純	步驟複雜
配合人員	資深軟體開發人員	資深統計技術人員
資料蒐集量	少	多

伍、品質量測模式與 WBS 品質改善方式

本節將提出一套 WBS-based 品質量測模式，在專案規劃過程中，協助找出潛在的高變動風險，進而衍生出矯正作業與改善活動，以具體降低軟體專案開發風險。

一、品質量測模式

影響 WBS 維護品質的四個關鍵因素分別為工作項目的基本品質特性、模組性、可追溯性及再估算品質等，各個關鍵品質的影響指標則是由一些低階的維護品質因子所組成，透過線性結合公式，可以將高度相關性的基層品質因子結合成基層品質量度，這些基層品質量度可以進一步結合成高階項目量測值，最後再將高階項目量測值加以結合，而成為 WBS 維護品質量測指標，以下分四個步驟說明。

- (1) 基本品質量測(Basic Quality Measurement; BQM)：進行基本品質量測之前，必須先分別以線性結合方式將正確性、完整性、一致性及可讀性等四組低階影響因子加以結合，產生正確性、完整性、一致性及可讀性等四項基層品質量度，再透過公式(1)，將四項基層品質量度結合成高階基本品質量測值，結合公式如下所示：

BQM: Basic Quality Measurement

CrM: Correctness Metric

W_{cr}: Weight of CrM

CmM: Completeness Metric

W_{cm}: Weight of CmM

CnM: Consistency Metric

W_{cn}: Weight of CnM

RdM: Readability Metric

W_{rd}: Weight of RdM

$$BQM = W_{cr} * CrM + W_{cm} * CmM + W_{cn} * CnM + W_{rd} * RdM$$

$$W_{cr} + W_{cm} + W_{cn} + W_{rd} = 1 \quad (1)$$

- (2) 模組性量測值(Modularity Measurement; MM)：進行模組性量測之前，必須先分別以線性結合方式將凝聚力、耦合度及複雜度等三組基層品質影響因子加以結合，產生凝聚力、耦合度及複雜度等三項基層品質量度，再透過公式(2)，將三項基層品質量度結合成模組性量測值，結合公式如下所示：

MM: Modularity Measurement

CpM: Coupling Metric

W_{cp}: Weight of CpM

ChM: Cohesion Metric

W_{ch}: Weight of ChM

CxM: Complexity Metric

W_{cx}: Weight of CxM

$$MM = W_{cp} * CpM + W_{ch} * ChM + W_{cx} * CxM \quad W_{cp} + W_{ch} + W_{cx} = 1 \quad (2)$$

- (3) 可追溯性量測值(Traceability Measurement; TM)：進行可追溯性量測之前，必須先分別以線性結合方式將 WBS 單層與跨層次工作項排定關係及單層與跨層次工作項交互關係之品質影響因子加以結合，產生單層與跨層次工作項排定關係及單層與跨層次工作項交互關係等四項基層品質量度，再透過公式(3)，將四項基層品質量度結合成可追溯性量測值，結合公式如下所示：

TM: Traceability Measurement

SLSR: Single Layer Scheduling Relationship

W_{slsr}: Weight of SLSR

ALSR: Across Layer Scheduling Relationship

W_{alsr}: Weight of ALSR

SLCRR: Single Layer Cross-Reference Relationship

W_{slcrr}: Weight of SLCRR

ALCRR: Across Layer Cross-Reference Relationship

W_{alcrr}: Weight of ALCRR

$$TM = W_{slsr} * SLSR + W_{alsr} * ALSR + W_{slcrr} * SLCRR + W_{alcrr} * ALCRR$$

$$W_{slsr} + W_{alsr} + W_{slcrr} + W_{alcrr} = 1 \quad (3)$$

- (4) 再估算品質量測值(Re-estimation Quality Measurement; RQM)：進行再估算品質量測之前，必須先分別以線性結合方式將費用估算的參數資料、簡潔公式及轉換方式之品質影響因子加以結合，產生參數資料、簡潔公式及轉換方式等三項基層品質量度，再透過公式(4)，將三項基層品質量度結合成再估算品質量測值，結合公式如下所示：

RQM: Re-estimation Quality Measurement

PDM: Parameter Data Metric

W_{pdm}: Weight of PDM

CFM: Clear Formula Metric

W_{cfm}: Weight of CFM

TMM: Transformation Manner Metric

W_{tmm}: Weight of TMM

$$RQM = W_{pdm} * PDM + W_{cfm} * CFM + W_{tmm} * TMM$$

$$W_{pdm} + W_{cfm} + W_{tmm} = 1 \quad (4)$$

- (5) 最後透過公式(5)，將基本品質、模組性、可追溯性及再估算品質等四項品質量測值結合成 WBS-based 品質量測指標(Indicator of WBS-based Quality Measurement)，結合公式如下所示：

IWQM: Indicator of WBS-based Quality Measurement

BQM: Basic Quality Measurement

W_{bqm}: Weight of BQM

MM: Modularity Measurement

W_{mm}: Weight of MM

TM: Traceability Measurement

W_{tm}: Weight of TM

RQM: Re-estimation Quality Measurement

W_{rqm}: Weight of RQM

$$IWQM = W_{bqm} * BQM + W_{mm} * MM + W_{tm} * TM + W_{rqm} * RQM$$

$$W_{bqm} + W_{mm} + W_{tm} + W_{rqm} = 1 \quad (5)$$

本品質量測模式一共匯集了 14 組基層 Web-based 維護品質影響因子，且透過線性結合模式產生 14 項基層品質量度及 4 項高階品質量測值，經過三個層次的量度結合作業，最後產生 WBS-based 品質量測指標，稱此量測作業模式為 WBS-based 品質量測模式 (WBS-base Quality Measurement; WBSQM Model)，其架構如圖 3 所示。

二、品質量測模式衍生的改善作業

WBSQM 模式所估算出的 WBS 維護品質量測指標，是協助找出 WBS 維護品質潛在問題與缺失的依據，透過維護品質低階影響因子、基層品質特性及高階品質特性等三層次，結合而成的維護品質量測值是找出 WBS 潛在問題與缺失的關鍵。量測值是一種相對性的指標，維護品質量測值小於 0.5 就可以視為「過低」的範圍，不過隨著軟體專案屬性的差異，「過低」的範圍也可以進行調整，如大型的軟體專案、開發人員眾多或需求不明確的專案等，變動的機會大幅增加，維護品質量測值「過低」的範圍就應該調整成小於 0.6 或 0.7。因此，當維護品質量測值落在「過低」的範圍，便可以從 WBSQM 模式中的結合公式，判斷出相關的品質影響因子，再從量化的影響因子剖析出對映的專案計畫之工作項目，進而找出 WBS 注入的高變動風險問題與缺失，依據問題與缺失可以提出具體的修訂作業與改善活動。以下即針對 WBS 維護品質「潛在問題與缺失」所提出的改善法則：

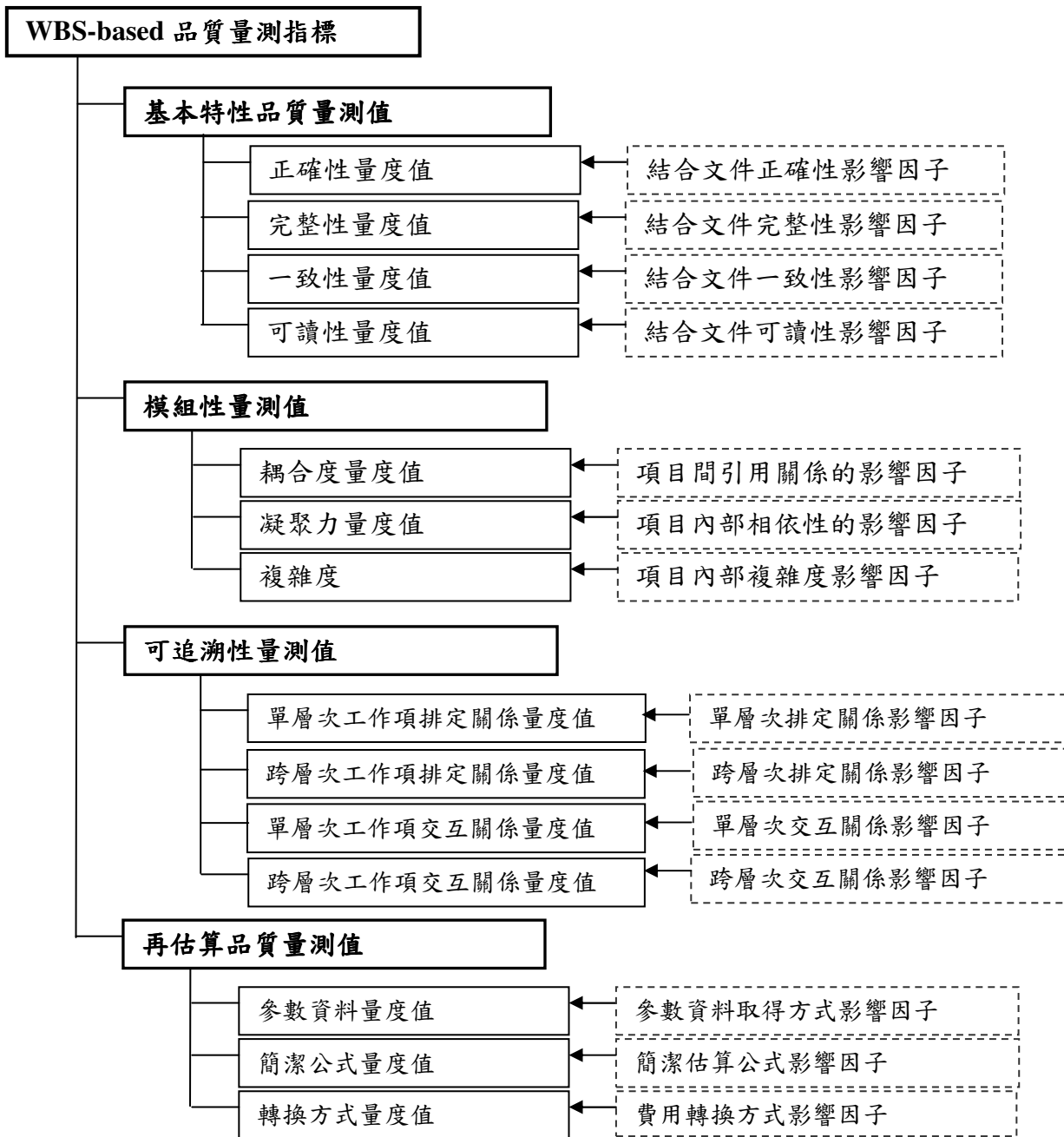


圖 3 WBS-based 品質測模式架構圖

<Rule 1> IF 「WBS 維護品質」量測指標被判定為過低等級

THEN 進一步分析「基本特性品質」、「模組性」、「可追溯性」及「再評估品質」等量測值是否落在「過低」狀態，且透過結合公式(5)，對照找出屬於過低狀態的品質特性量測值。

<Rule 2> IF 「基本特性品質」量測值屬於「過低」狀態

THEN 可以進一步分析出那幾項基層之基本特性品質不良造成「基本特性品質」量測值「過低」狀態中，且透過結合公式(1)，對照找出基本特性品質量度值所對應的正確性、完整性、一致性及可讀性等品質影響因

子，再由影響因子配合找出潛在的問題與缺失，進行修正與改善等措施。

<Rule 3> IF 「模組性」量測值屬於「過低」狀態
THEN 可以進一步分析出那幾項基層品質不良造成「模組性」量測值落在「過低」狀態中，且透過結合公式(2)，對照找出基層特性量度值所對應的凝聚力、耦合度及複雜度等品質影響因子，再由影響因子配合找出潛在的問題與缺失，進行修正與改善等措施。

<Rule 4> IF 「可追溯性」量測值屬於「過低」狀態
THEN 可以進一步分析出那幾項品質特性的品質造成「可追溯性」量測值落在「過低」狀態中，且透過結合公式(3)，對照找出基層特性量度值所對應的 WBS 單層與跨層次工作項排定關係及單層與跨層次工作項交互關係等影響因子，再由影響因子配合找出面潛在的問題與缺失，進行修正與改善等措施。

<Rule 5> IF 「再評估品質」量測值屬於「過低」狀態
THEN 可以進一步分析出那幾項品質特性的品質造成「再評估品質」量測值落在「過低」狀態中，且透過結合公式(4)，對照找出基層特性量度值所對應的參數資料取得方式、估算公式及費用轉換方式等影響因子，再由影響因子配合找出潛在的問題與缺失，進行修正與改善等措施。

以 WBS-based 品質量測為基礎的維護品質改善活動，可以標示出 WBS 維護品質存在的問題與缺失，再配合及時矯正作業與改善活動，可以具體提昇 WBS 維護品質，對於變動頻繁的軟體專案風險應可獲得大幅的降低，進而有效的提高軟體專案的成功率。

陸、結論

軟體系統開發過程中，受到開發人員的疏失、資訊技術與環境的演進、使用者的變更要求以及資源配置的調整等多樣因素的影響，專案計畫必須持續修訂與調整，計畫一旦無法及時應付各種變動要求，勢必難以引導軟體專案持續運作，進而導致專案的失敗。WBS 為軟體專案計畫的核心，出現各種變更事件，首當其衝的就是 WBS 規劃文件，為了降低變更事件帶來的風險，WBS 必須具備高度維護能力。為此，本文探討且蒐集 WBS 維護品質之影響因子，且深入剖析出衝擊專案計畫可變動能力的潛在問題與缺失，及時提出修正作業與改善活動，將可大幅降低計畫變動導致專案失敗的風險。本文討論引發專案計畫變動的四項關鍵事件，且以查核表的審查與檢視作業，蒐集且量化 WBS 維護品質影響因子，進而提出一套 *WBSQM* (WBS-based Quality Measurement) 模式，以量化的機制找出影響 WBS 維護作業的品質問題與缺失，進而衍生出修正作業與改善活動，以具體提昇 WBS 維護品質且降低軟體計畫的變動風險。本文提出的 *WBSQM* 模式更有下面四項優勢：

- (1) 以量化與量測機制及早發現 WBS 維護品質的問題與缺失。
- (2) 透過修正作業與改善活動及時矯正 WBS 維護品質的問題與缺失。
- (3) 持續進行 WBS 維護品質的改善作業，直到軟體專案結束或被終止。

(4) 品質量測結合公式具備簡化公式、易於標示問題、調整彈性與延伸作業等特質。

誌謝

本論文接受 100 學年度國科會研究計畫(計畫編號：NSC100-2221-E-158-007)之補助。

參考文獻

1. 賴森堂, 2005, 『輔助軟體專案管理的多層次風險量測模式』, 第一屆台灣軟體工程研討會, 中央大學。
2. Boehm, B. W., "Software risk management: Principles and practices," *IEEE Software*, 8(1), 1991, pp.32-41.
3. Boehm, B. W., *Tutorial: Software Risk Management*, IEEE CS Press, Los Alamitos, Calif, 1989.
4. Boehm, B.W., "A Spiral Model of Software Development and Enhancement," *IEEE Computer*, 25(5), 1988, pp.61-72
5. Boehm, B.W., *Software Engineering Economics*, Prentice-Hall, New Jersey, 1981.
6. Conte, S. D., H. E. Dunsmore, and V. Y. Shen, *Software Engineering Metrics and Models*, Benjamin/Cummings, Menlo Park, 1986.
7. Eveleens, J. L. and C. Verhoef, "The Rise and Fall of the Chaos Report Figures," *IEEE Software*, vol. 27, no. 1 , 2010, pp. 30-36.
8. Fairley, R., "Risk management for Software Projects," *IEEE Software*, 11(3), 1994.
9. Fenton, N.E. , *Software Metrics - A Rigorous Approach*, Chapman & Hall, 1991.
10. Galin, Daniel, *Software Quality Assurance – From theory to implementation*, Pearson Education Limited, England, 2004.
11. Hayes F., "Chaos is Back," *Computerworld*, Normember, 2004.
12. IEEE std. 1058-1998, *IEEE Standard for Software Project Management Plans*, IEEE, New York, 1998.
13. ISO/IEC: FCD 9126-1.2: *Information Technology – Software Product Quality. Part I: Quality Model*, 1988.
14. Lai, S. T., "A Quality Measurement Model for Software Maintenance," *Second World Congress on Software Quality (2WCSQ)*, Japan, 2000.
15. Lai, S. T. and C. C. Yang, "A Software Metric Combination Model for Software Reuse," *1998 Asia-Pacific Software Engineering Conference*, Taiwan, Taipei, 1988.
16. Mantel, Jr., S.J. etc. "Project Management in Practice", John Wiley & Sons, 2011.
17. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, New York, 2010.
18. Schach, S. R., *Object-Oriented Software Engineering*, McGraw-Hill Companies, 2008.
19. Weller, E.F., "Using Metrics to Manage Software Projects," *IEEE Computer*, 27(9), 1994, pp.27-24.

A WBS-based Quality Measurement Model for Improving Software Project Changeability

Sen-Tarng Lai

Department of Information Technology and Management, Shih Chien University

Email: stlai@mail.usc.edu.tw

Abstract

In software development, project plan will be impacted by change events which include requirements specification correction, information technology and environment evolution, requirement change and resource adjustment. Project plan revision unable to promptly and correctly achieve, software development will encounter the difficult position and cause to the project failure. In order to reduce project failure risk, project plan must have high changeability to promptly cope with the impact of change events. WBS is the critical kernel of software project plan. Therefore, this paper discusses the change events which cause the WBS modification and revision. Analyze and collect impact factors which affect WBS revision and adjustment, and propose a WBS-based Quality Measurement (*WBSQM*) model. Applying *WBSQM* model, WBS maintenance quality problems and defects can be identified. And derive the revision and improvement operations to enhance the changeability of project plan and reduce software development risk.

Keywords: WBS, measurement model, changeability, software project, software development