# 在需求管理中套用使用案例

**Roger Oberg、Leslee Probasco 與 Maria Ericsson**

Rational Software 白皮書

TP 505（1.4 版）

Rational®
the software development company

# 目錄

如果您對需求管理很陌生或有點熟悉，而您對需求流程提升；改善很有興趣，則本白皮書提供的架構可供您開發自己的方法。

> **使用案例與軟體需求規格 (SRS)**
>
> 爲了使需求管理活動對讀者更具意義，作者從 Rational Software 的統一流程和業界標準統一建模語言中選擇了特定文件類型和其他需求管理構件，兩者都建議採用使用案例驅動的軟體工程流程。
>
> 因此，這裡說明用來指定軟體需求的使用案例驅動方法。這些活動也可以與更傳統的軟體需求規格（例如 IEEE 標準）一起使用，來取代或補充使用案例模型和使用案例。

## *流程存在期間內的軟體和系統開發*

對於大部分軟體和系統開發團隊而言，相較於過去無拘無束的日子，1990 年是一個流程密集的年代。測量和證明有效軟體開發流程的標準已引進並普遍流行。有關軟體開發流程的許多書籍和文章，以及商業流程建模和再造的相關資料也已發佈。越來越多的軟體工具更協助定義及套用有效的軟體開發流程。全球經濟對軟體的相依性在這十年當中加速成長，賦予開發流程能力並提升系統品質。

因此，我們該如何解釋目前軟體專案失敗的高意外率？爲何大部分（若非全部）軟體專案仍然遭受延遲、超出預算和品質問題之苦？當企業、國家經濟和日常生活越來越依賴我們所建置的系統時，我們該如何改進系統的品質？

答案不外乎應用在我們這一行的人力、工具和流程。需求管理通常是提出來作爲軟體開發現行問題的解決方案，但很少關注此規範作法上的改進。

本白皮書介紹有效需求管理流程的元素，並強調順利完成實作所要面臨的一些障礙。

需求管理同時適用於純軟體專案以及軟體只是其最後結果的一部分或完全不包含軟體的專案。爲了方便起見，本白皮書以下將使用「系統」的這個詞彙來表示它們其中之一或全部。然而，軟體開發不論是單獨存在或因爲與硬體結合而使需求管理複雜化，其抽象本質才是本白皮書的主要焦點。

## *爲何需要管理需求？*

簡單地說，負責管理需求的系統開發團隊會這麼做，是因爲他們希望專案能夠成功。符合其專案需求就是成功的定義。無法管理需求會減低符合這些目標的機率。

最近的證據也支持這樣的說法：

- Standish Group 從 1994 到 1997 的 CHAOS Reports 證實，專案失敗的最重要促成因素與需求有關。[1]
- 在 1997 年 12 月，Computer Industry Daily 對 Sequent Computer Systems, Inc. 的英美兩國 500 位 IT 主管進行一項研究，發現有 76% 的回應者在其職業生涯中都有專案完全失敗的經驗。專案失敗的最常提及原因是「變更使用者需求」。[2]

避免失敗爲管理需求的充份動機。增加成功專案的機率以及管理需求的其他好處也一樣具有誘導性。Standish Group 的 CHAOS 報告進一步證實，良好的需求管理才是專案成功最重要的因素。

## *什麼是需求？*

瞭解需求管理的第一個步驟就是對共同詞彙達成共識。Rational 將需求定義爲「所要建置之系統必須符合的條件或功能」。Institute of Electronics and Electrical Engineers (IEEE) 使用類似定義。

知名需求工程作者 Merlin Dorfman 和 Richard H. Thayer 提供一個相容但更精確的定義，此定義對軟體是特定的但不一定是限制。

「軟體需求可定義爲：

- 使用者要解決問題或達到目標所需的軟體功能。
- 系統或系統元件必須符合或擁有才能滿足合約、規格、標準或其他正式文件的軟體功能。」[3]

## *什麼是需求管理？*

因爲需求是所要建置之系統必須符合的，且符合某一組需求是專案成功或失敗的關鍵，因此，瞭解什麼是需求，寫下它們，組織它們，在它們變更的事件中追蹤它們，都是合理的作法。

換句話說，需求管理就是：

- 有計劃有步驟地誘導、組織和記載系統需求的方法，以及
- 客戶和專案小組之間對於系統的變更需求建立及維護共識的一種流程。

此定義類似 Dorfman 和 Thayer 以及 IEEE 對「軟體需求工程」的定義。需求工程包括軟體需求的誘導、分析、規格、驗證和管理，以「軟體需求管理」作爲所有這些相關活動的規劃和控制 [4]。所有這些活動會納入此處介紹及 Rational Software 所指導的需求管理的定義中。其差異主要在於選擇「管理」而非「工程」一詞。管理是所有相關活動更適當的說明，它正確強調追蹤變更的重要性，以維護關係人和專案小組之間的共識。

對於那些不熟悉「誘導」這個詞彙的人，它是定義爲一組活動，專案小組運用它們來誘導或探索關係人要求、決定要求背後的實際需求，以及達成系統要滿足這些需求所需具備的一組需求。

## *需求管理的問題*

那麼，要確保系統能夠達到其設定期望，在流程上有哪些困難？當執行實際專案時，困難就會出現。「圖 1」顯示 1996 年對開發人員、經理和品管人員的一份調查結果。它顯示有遇到最常提及的需求相關問題的回應者所佔的百分比。

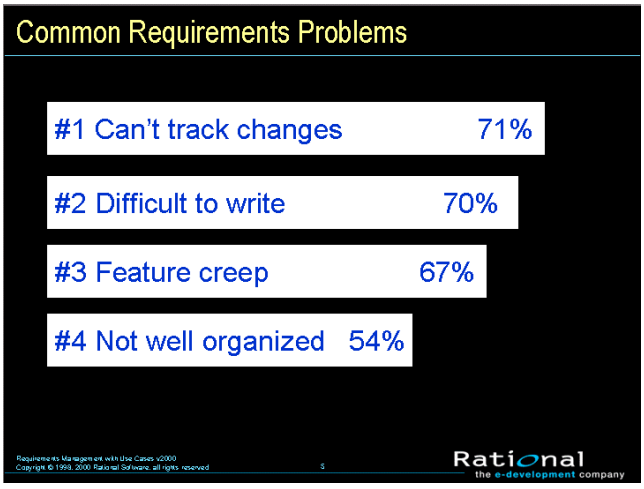


**圖 1 — 一般需求問題**

更完整的問題清單包括：

- 需求不一定明顯，而且有許多來源。
- 要用文字將需求表達清楚並不容易。
- 不同詳細資料層次有許多不同類型的需求。
- 如果不加以控制，需求的數目會變得難以管理。
- 需求彼此相關，也與流程的其他交付項以各種不同方式相關。
- 需求具有唯一內容或內容值。例如，它們不同樣重要，也不同樣容易滿足。
- 感興趣及負責的團體不少，這表示需求要由許多跨功能團隊來管理。
- 需求變更。
- 需求可能有時間性。

這些問題再加上需求管理和流程技巧不足，以及缺乏容易使用的工具，有許多團隊會覺得他們永遠無法妥善管理需求，而感到絕望。Rational Software 已發展出一套專案技術來指導團隊的需求管理技巧和流程。此外，Rational RequisitePro® 是一個可存取的工具，可達到有效需求管理自動化。

## *需求管理技巧*

爲了解決上述問題，Rational 鼓勵開發關鍵技巧。以下介紹這些技巧，它們看起來是按順序呈現，不過，在有效需求管理流程中，它們是以不同順序連續套用。這裡是按您要套用到新專案的第一個反覆的順序來呈現。

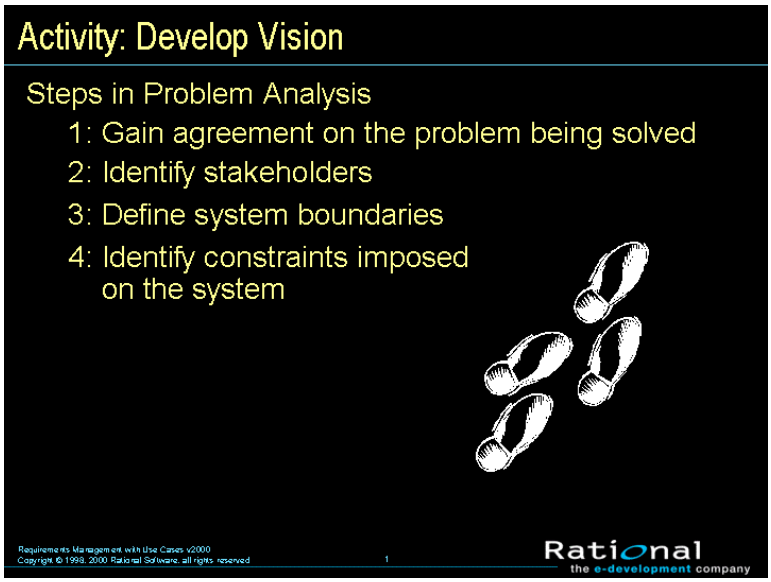


**圖 2 — 問題分析的步驟**

### 關鍵技巧 1：分析問題

執行問題分析，以瞭解商業問題、以初始共同工作人關係人需求爲目標及提議高階解決方案。這些推理和分析的行動將發現「問題背後的真正問題」。

在問題分析期間，找到真正問題和關係人，即代表取得共識。起始解決方案界限和限制是從技術和商業觀點來定義。如果適當，專案的企業案例會分析系統預期的投資報酬率。

### 關鍵技巧 2：瞭解關係人需求

需求有許多來源。它們可能來自對專案結果感興趣的人。客戶、夥伴、一般使用者和領域專家是需求的一些來源。管理部門、專案小組成員、企業政策和主管機關是另一些來源。

知道如何決定誰是來源、如何存取那些來源及如何從中誘導資訊，是很重要的。作爲此資訊主要來源的個人，在專案中稱之爲「關係人」。

如果您要開發一個在公司內部使用的資訊系統，則您的開發團隊可包括有一般使用者經驗和企業領域專案知識的人。通常您會在商業模型層級上而非系統層級上開始進行討論。如果您要開發的產品是要銷售到市場上，您可以擴大規模來利用行銷人員，以便更瞭解該市場的客戶需求。

誘導需求的技巧包括面談、腦力激盪、觀念塑型、問卷調查和競爭力分析。需求誘導的結果是以文字方式和以圖形方式描述的要求或需求的清單，這些要求或需求已有相對優先順序。

### 關鍵技巧 3：定義系統

定義系統表示將對於關係人需求的瞭解翻譯並組織成對於要建置之系統的明確說明。在系統定義階段初期，要決定需求的構成要件、文件格式、語言形式、需求程度、要求優先順序和預估人力、技術和管理風險，以及範圍。此活動有一部分包含初期原型和設計模型，它們與最重要的關係人要求直接相關。

我們使用「說明」而不是「文件」，以避免後者在一般使用上固有的已知限制。說明可能是書面文件、電子檔案、圖片或用來傳達除系統本身以外的系統需求的任何其他表示法。

系統定義的結果是使用自然語言和圖形的系統說明。後面幾節將提供一些建議的說明格式。

> **原則 55：在較正式的模型之前撰寫自然語言**
>
> 如果您先撰寫正式模型，則傾向於撰寫自然語言來說明模型而非解決方案系統。請看以下的範例：
>
> > 若要撥長途電話，使用者應該拿起電話。系統會以撥號音回應。使用者要撥 "9"。系統會以特別的撥號音回應。…
> >
> > 系統由四個狀態組成：閒置、撥號音、特殊撥號音和連線。若要從閒置狀態變成撥號音狀態，請拿起電話。若要從撥號音狀態變成特殊撥號音狀態，請撥 "9"。
>
> 請注意，在後面的範例中，文字對讀者一點幫助也沒有。
>
> —Alan M. Davis, 201 Principles of Software Development, 1995

### 關鍵技巧 4：管理系統的範圍

專案的範圍是由配置給它的一組需求來定義。管理專案範圍使其適合可用的資源（時間、人和金錢）是管理成功專案的關鍵。管理範圍是一個連續活動，需要反覆或漸進式開發，而將專案範圍分割成更多更小的可管理單元。

使用需求屬性（例如優先順序、人力和風險）作爲協商需求併入的基礎，這項技術在管理範圍上特別有用。把焦點放在屬性而非需求本身，有助於減輕協商的爭議性。

小組負責人接受協商技巧的訓練以及專案在組織中和在客戶端具有優秀人士，也很有幫助。產品和專案優秀人士應該有組織能力來拒絕超出可用資源的範圍變更，或展開資源來容納其他範圍。

### 關鍵技巧 5：修正系統定義

有一致同意的高階系統定義和一目了然的起始範圍之後，在更精確的系統定義中投入資源既可行又經濟。修正系統定義包括兩個主要考量：研擬高階系統定義更詳細的說明，並確認系統符合關係人的需求而且行爲方式一如所述。

說明通常是專案小組的重要參照資料。心中惦記著讀者才能提供最好的說明。常見的錯誤是用複雜定義來代表複雜建置，尤其讀者可能無法或不願意花時間去思考如何才能達成共識。這導致難以向專案小組內外的人說明系統的用途。您會發現需要爲不同讀者提供不同種類的說明。本白皮書包括詳細自然語言的建議格式、正式文字和圖形說明。建立說明格式之後，會在整個專案生命週期內繼續修正。

### 關鍵技巧 6：管理變更需求

不論您多麼仔細定義需求，它們還是會改變。事實上，有一些需求變更是好事！它表示團隊吸引關係人。調和變更需求是對團隊關係人靈敏度和操作彈性的測量—構成成功專案的團隊屬性。變更不是敵人；未管理的變更才是要害。

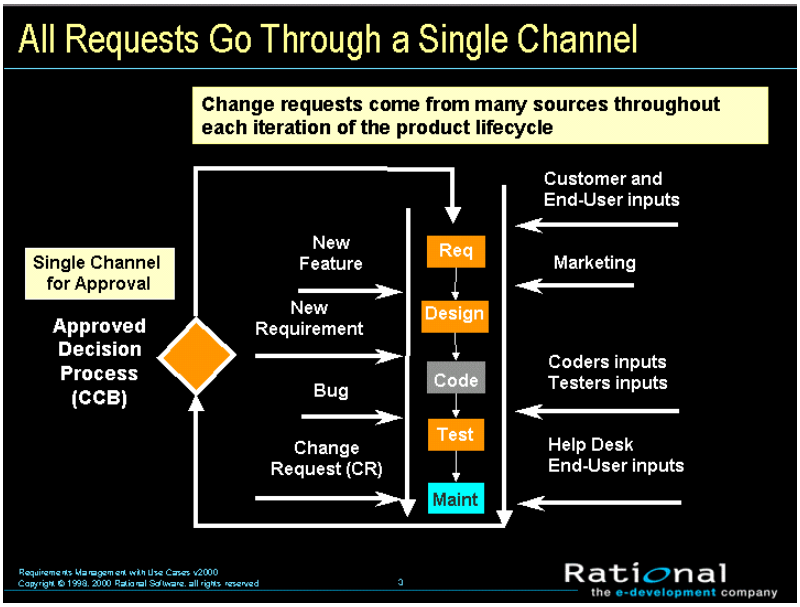


**圖 3—管理變更的流程**

變更的需求表示多少要花點時間實作特定特性，一項需求的變更可能會影響到其他需求。管理需求變更包括一些活動，例如建立基準線、追蹤每一項需求的歷程、決定要追蹤的重要相依關係、在相關項目之間建立可追蹤的關係，及維護版本控制。如「圖 3」所示，建立變更控制或核准流程也很重要，所有提出的變更需要由指定的團隊成員檢視。有時候，這個變更控制的單一管道稱爲變更控制委員群組 (Change Control Board, CCB)。

## *重要需求概念*

若要將需求管理技巧運用到專案中，專案中每個人瞭解某些需求管理概念是有幫助的。它們包括：

- 需求類型
- 跨功能團隊
- 可追蹤性
- 多維屬性
- 變更歷程

### 需求類型

系統越大越複雜，需求的類型越多。需求類型只是需求的類別。若能識別需求類型，團隊就可以將大量需求組織成更多有意義的、可管理的群組。在專案中建立不同類型的需求有助於團隊成員將變更的要求加以分類，使溝通更加明確。

通常，一種需求類型可分割或分解成其他類型。商業規則和願景陳述是高階需求的類型，團隊可從中衍生使用者需求、特性和產品需求類型。使用案例和其他形式的建模驅動設計需求，它們可分解成軟體需求，並在分析和設計模型中表示。測試需求是從軟體需求中衍生的，可分解成特定測試程序。若給定的專案中有數百、數千、甚至數萬個需求實例，將需求分類成類型，可使專案更容易管理。

### 跨功能團隊

測試或應用程式建模之類的流程可以在單一商業群組中管理，需求管理和它們不同，凡可以將其專業知識貢獻到開發流程的人都與需求管理有關。它應該包括代表客戶和商業期望的人。開發經理、產品經理、分析師、系統工程師、甚至客戶都應該參與其中。需求團隊也應該包括那些建立系統解決方案的人 — 工程師、建構師、設計師、程式設計師、技術文件作者和其他技術文章投稿人。測試者和其他品管 (QA) 人員應該算是重要的團隊成員。

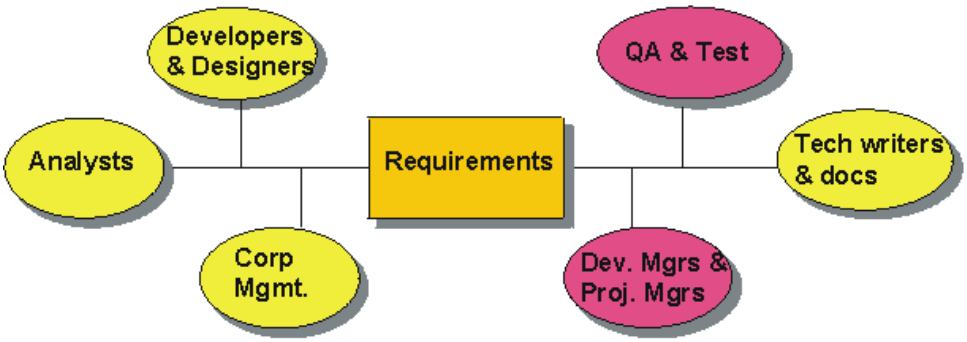


**圖 4 — 跨功能需求團隊**

通常，編寫和維護需求類型的責任可按功能範圍配置，進一步促成更佳的大型專案管理。需求管理的跨功能本質是規範更具挑戰的一個層面。

## 可追蹤性

如需求類型的說明中所暗示，需求不能有單獨存在的單一表達。關係人要求與爲了滿足它們而提出的產品特性相關。就功能和非功能行爲而論，產品特性與指定特性的個別需求相關。測試案例與它們確認及驗證的需求相關。需求可與其他需求相依或彼此互斥。若團隊要決定變更的影響以及系統符合期望感到有信心，必須瞭解、記錄和維護這些可追蹤性關係。可追蹤性是在需求管理中實作的其中一個最困難的概念，它是要調和變更不可或缺的。建立明確需求類型及納入跨功能參與，可使可追蹤性更容易實作和維護。如需需求可追蹤性的不同策略的相關資訊，請參閱白皮書「以使用案例管理需求的可追蹤性策略」[5]。

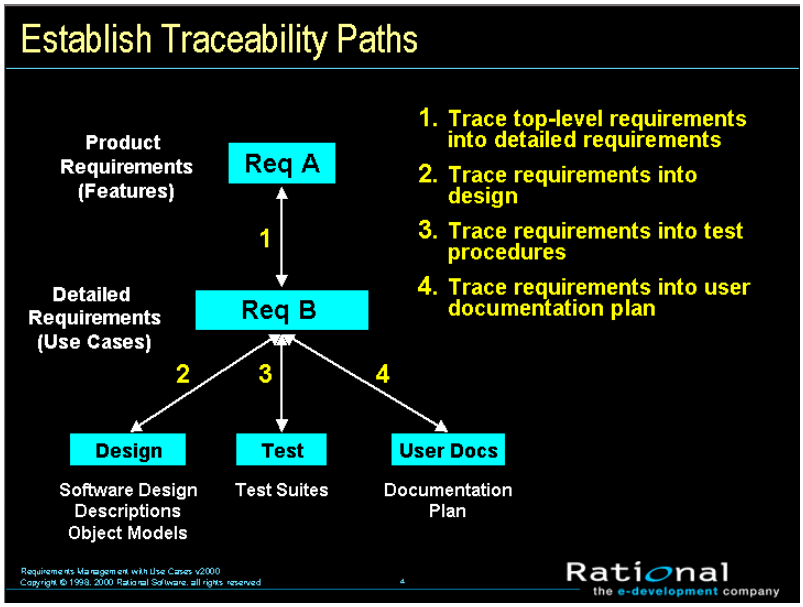


**圖 5 — 需求可追蹤性**

## 多維屬性

每一個需求類型都有屬性，每一個個別需求都有不同的屬性值。例如，需求可指派優先順序、由來源和基本理由來識別、授權給功能範圍內的特定小組、給予困難度指派，或與系統特定反覆相關聯。爲了加以說明，「圖 6」顯示在 Rational RequisitePro 需求管理工具的「學習專案」中的「特性需求類型」的屬性。如畫面標題所暗示，需求類型和每一種類型的屬性是針對整個專案而定義，確保跨團隊使用的一致性。

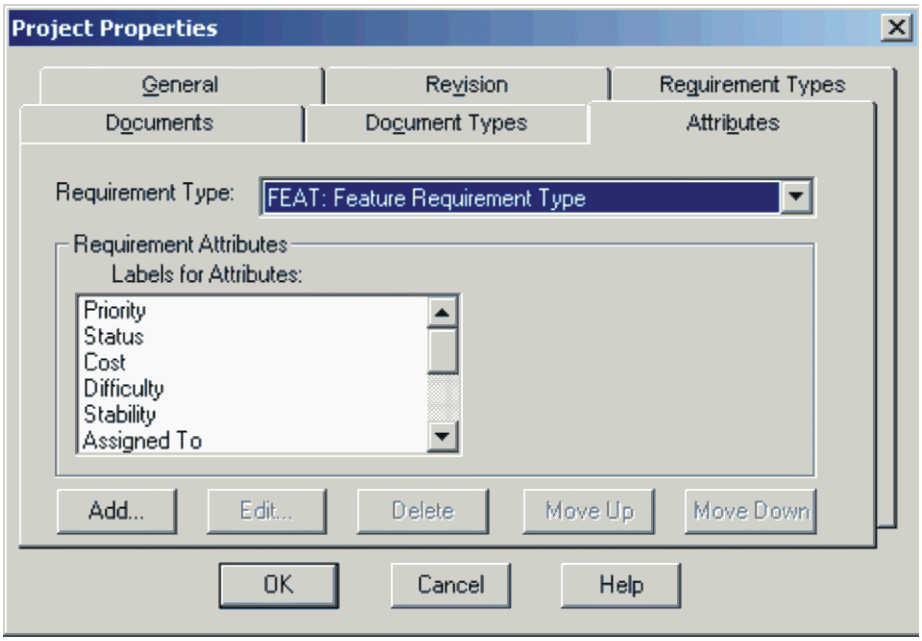


**圖 6 — 定義特性的需求屬性**

在「圖 7」，特性需求的實例是針對 RequisitePro 中的特定專案而顯示。請注意，即使沒有顯示每一項需求的完整文字，我們也可以從它的屬性值當中充份瞭解每一項需求。在此情況下，其優先順序和困難度—當然是由不同團隊成員指定—將幫助團隊開始將專案範圍限定爲可用的資源和時間，並將關係人優先順序和反映在困難度屬性值的人力概估列入考慮。在更詳細的需求類型中，優先順序和人力屬性可能有更特定的值（例如預估時間、程式碼行等等），來進一步修正範圍。需求的這個多維層面由不同類型的需求合成—各有其自己的屬性—對於組織大量需求及管理專案整體範圍很重要。

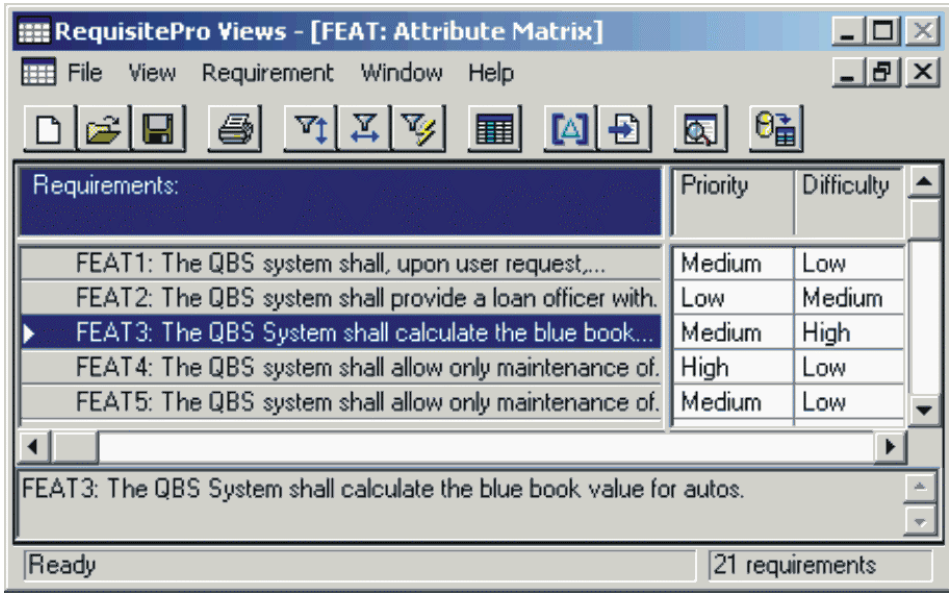


**圖 7 — 設定每一項需求的屬性值**

### 變更歷程

個別需求和需求集合都有歷程，它們在經過一段時間後變得有意義。變更是無可避免的好事，這是爲了與千變萬化的環境及新興技術齊步並進。記錄專案需求的版本可讓小組負責人擷取變更專案的原因，例如新的系統版本。瞭解需求的集合可能與軟體特定版本相關聯，可讓您以漸進方式管理變更，降低風險並提高符合里程碑的機率。當個別需求浮現時，一定要瞭解其歷程：變更的內容、原因、時間，甚至由誰授權。

## *使需求管理可以運作*

需求管理運用上述關鍵技巧和概念來識別問題並成功地解決問題。

爲了建置真正符合客戶需求的系統，專案小組必須先定義系統要解決的問題。下一步，團隊必須識別關係人，並從中誘導、描述及優先順序化企業和使用者需求。在這一組高階期望或需求中，有一組產品或系統特性應該得到共識。

詳細軟體需求需要以客戶和開發團隊都可以瞭解的形式撰寫。我們發現使用客戶的語言來描述這些軟體需求，是獲得其瞭解和同意的最有效方式。這些詳細的軟體需求會作爲系統設計規格的輸入，以及用於實作和驗證所需的測試計劃和程序。軟體需求也應該驅動起始使用者文件規劃和設計。

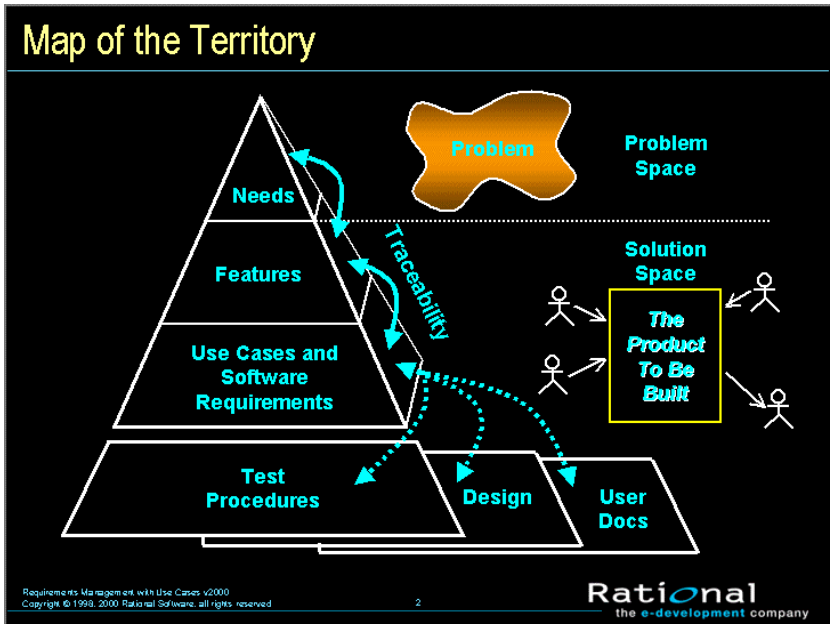


**圖 8 — 需求管理結構的概觀**

若要有所幫助，專案小組應該：

- 同意專案的一般詞彙
- 開發現系統的願景，來描述系統要解決的問題以及它的主要特性。
- 在至少五個重要區域中誘導關係人的需求：功能、可用性、可靠性、效能和支援性
- 決定要使用的需求類型

- 選取每一個需求類型的屬性和值
- 選擇用來描述需求的格式
- 識別將編寫、提出或只檢視一或多個需求類型的團隊成員
- 決定需要的可追蹤性
- 建立提出、檢視和解決需求變更的程序
- 開發機制來追蹤需求歷程
- 建立團隊成員和管理的進度和狀態報告

這些重要的需求管理活動與業界、開發方法或需求工具無關。它們也很有彈性，可以在最嚴苛最快速的應用程式開發環境中產生有效的需求管理。

**關於文件的二三事**

在文件中描述需求的決定應該三思而行。一方面，寫作是被廣泛接受的溝通形式，對大多數的人而言，也是很自然會去做的事。在另一方面，專案的目標是要產生系統，而不是文件。

根據一般常識和經驗法則，不是要決定要不要記錄需求，而不是要決定如何記錄需求。文件範本提供需求管理的一致格式。Rational RequisitePro 提供這些範本，以及在文件內連結需求與包含所有專案需求的資料庫的其他特性。此獨特特性可在自然狀態下記錄需求，而且在關聯式資料庫中可以更容易存取和管理需求。

## *需求管理：活動*

需求管理可遵循無限個網域特定路徑。下列方法指定六個詳細活動，它們套用到每一個主要需求管理技巧中，但也可以套用到任何網域。

下列活動圖是來自 Rational Unified Process [6] 需求規範。這些活動是從角色、作業和構件（輸入或輸出）方面來表達，如「圖 9」的總結所述。本白皮書的文字簡短描述每一個活動，希望能刺激您在改進需求管理流程上的想法和興趣。如需 Rational Unified Process 的相關資訊，請參閱 www.ibm.com/software/rational。

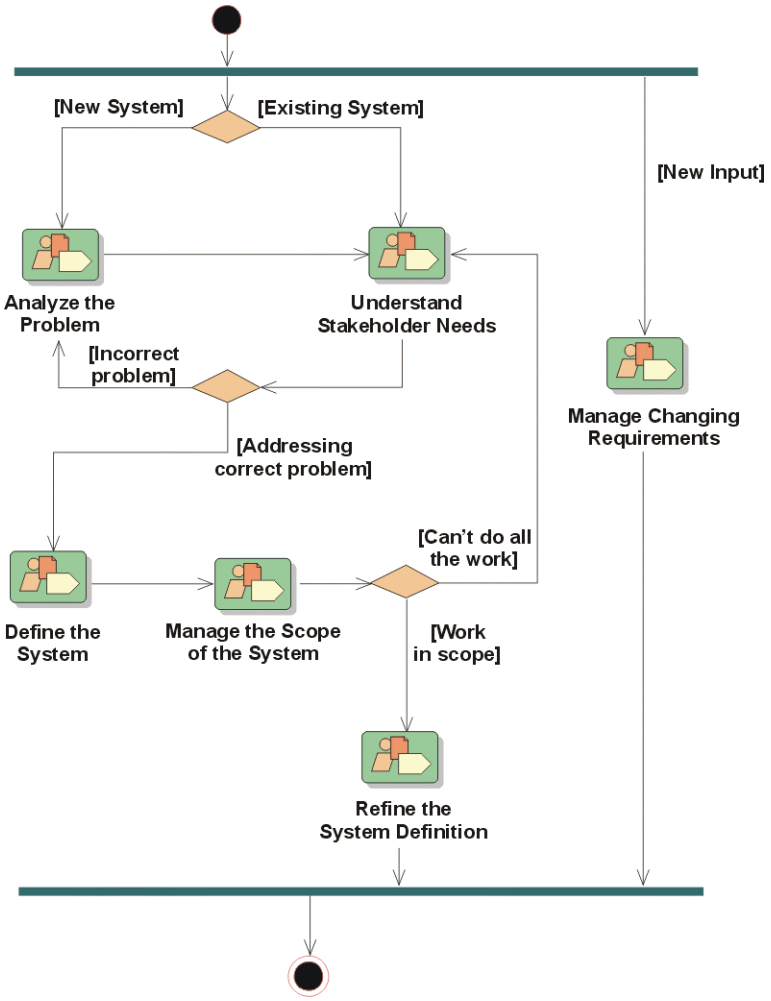


**圖 9 — Rational Unified Process 中的需求工作流程**

## 活動：分析問題

在問題分析中，主要活動是開發專案的願景。此活動的輸出是一份願景文件，它識別高階使用者或客戶對於所要建置之系統的視圖。願景表達的起始需求是系統要解決大部份嚴重問題及符合主要關係人需求時必須擁有的主要特性。系統分析師在此活動中扮演主要角色。系統分析師應該有問題領域專業知識，並瞭解問題，而且應該能夠說明他認爲可以解決問題的流程。需要不同專案關係人的積極參與，而且應該考量所有相關的關係人要求。

若要開始管理相依關係作業，應該指派特性的屬性，例如有理數、相對值或優先順序和要求的來源。在發展願景時，分析師會識別可能的使用案例的使用者和系統（參與者）。參與者是使用案例模型的第一個元素，它將定義系統的功能和非功能技術需求。

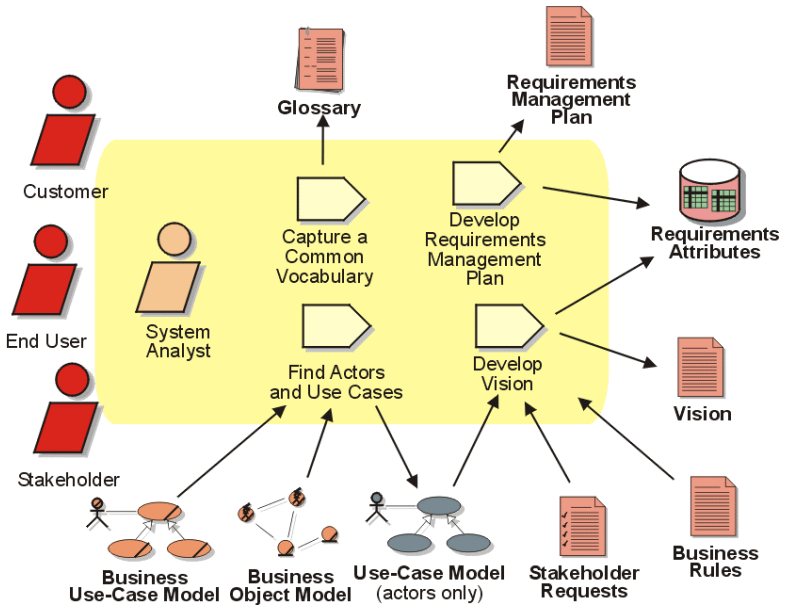


**圖 10 — 分析問題**

*活動：分析問題*

開始：發現問題的一或多個關係人將起始此活動。

在開發團隊中的系統分析師可促成一個階段作業，協助起始階段的關係人描述他們想要解決的問題。對於所發現問題的簡要陳述取得共識是很重要的。下表顯示**問題陳述**的主要元素：

| | |
|---|---|
| **問題** | 定義問題 |
| **影響關係人** | 列出受問題影響的關係人 |
| **問題的影響** | 說明問題的影響 |
| **成功的解決方案應包括** | 列出成功解決方案的一些主要優點 |

問題陳述簡要說明專案的目的。問題分析刺激進一步調查所有關係人要求及起始企業案例，包括令人信服的優點和概估成本。與定義問題陳述相比，團隊也應該編譯名詞解釋，追蹤常用詞彙及對其定義取得共識。

**使用案例模型的簡介**

使用案例模型由參與者、使用案例及其關係組成。參與者代表必須與系統交換資訊的一切，包括一般所謂的使用者在內。當參與者使用系統時，系統會執行使用案例。理想的使用案例是一個交易序列，它會產生參與者值的可量測結果。使用案例的集合是系統的完整功能。

—Jacobson I., Christerson M., Jonsson P., Overgaard G., Object-Oriented Software Engineering – A Use Case Driven Approach, Addison Wesley – ACM Press, 1992

問題分析也識別主要系統參與者。參與者是指系統或任何其他會與其交換資訊的系統的使用者。在此階段，問題分析應該簡短識別幾個顯著的方法，讓參與者與系統互動。說明應該傾向商業流程而非系統行爲。例如，預算計劃可

以讓某一種類型的參與者「建立部門預算」，讓另一個參與者能夠「統一部門預算」。系統分析師稍後可將它們分成其他的使用案例，與特定的系統行爲更有意義地搭配在一起。例如，「建立部門預算」會導致這樣的系統使用案例：匯入試算表資訊、建立預算視圖。

上述問題分析階段作業通常不止執行一次，可能由不同的關係人執行，並混合了內部開發團隊階段作業。要求與關係人開會的系統分析師將帶領一個與開發團隊之間的階段作業，來展望問題的技術解決方案、從起始關係人輸入衍生特性，並起草願景說明，這是所要建置之系統的第一個定義。爲了幫助起始階段的關係人瞭解提出的解決方案，系統分析師可使用建模工具或手繪圖片技巧來增補願景說明。

在許多時間點上會諮詢起始階段的關係人，協助修正問題說明，並限制可能之解決方案的數目和範圍。關係人和系統分析師將協商主要特性的優先順序，並對於開發它們所需的資源和人力取得共識，藉以管理此活動中的相依關係。雖然優先順序和人力或資源預估無可避免會改變，但提早管理相依關係可建立一個重要模式，連續使用在整個開發生命週期中。這是範圍管理的要素及專案成功的早期預報器。

在歷經幾次起草之後，願景到達團隊必須決定是否要投入其他需求誘導的時間點。同時，企業案例核准流程也個別起始。雖然本白皮書未進一步說明，但企業案例會描述下列每一項：

- 環境定義（產品領域、市場和範圍）
- 技術方法
- 管理方法（排程、風險、成功的客觀測量）
- 財務預測

### 活動：瞭解關係人需求

如果起始問題分析證明其他投資是正當的，則「瞭解關係人需求」活動即正式開始。主要工作是誘導關係人要求。主要結果是優先順序化的關係人需求和特性的集合，它啓用願景文件的修正，並且更充分的瞭解需求屬性。同時，在此活動期間，您可以從系統的使用案例和參與者方面開始討論系統。另一個重要結果是更新的名詞解釋，可促進團隊成員之間的共同詞彙。

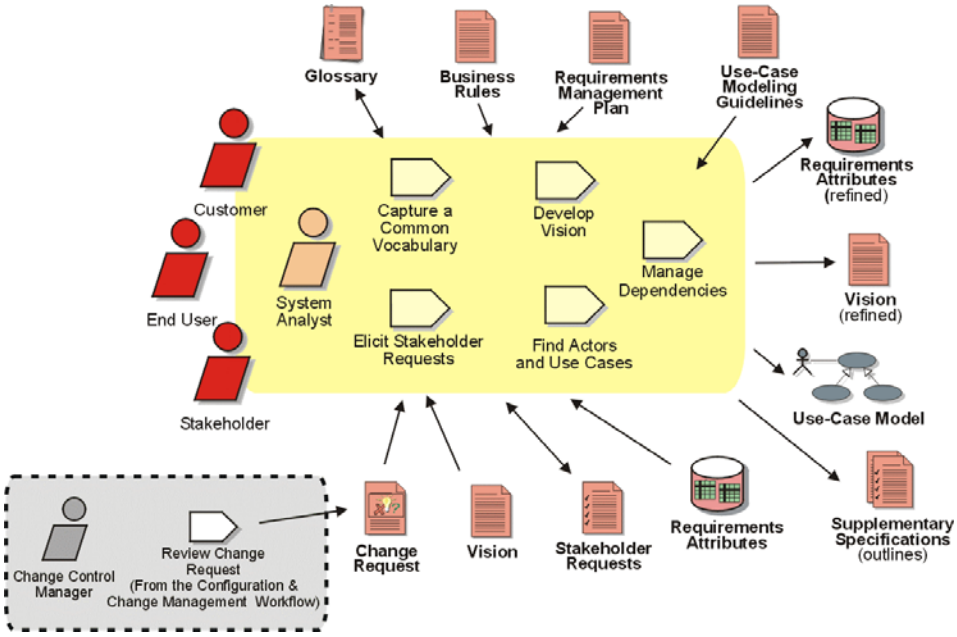


**圖 11 — 瞭解關係人需求**

*活動：瞭解關係人需求*

系統分析師和主要關係人識別其他關係人、誘導其要求，並透過面談、研討會、分鏡腳本、商業流程使用案例和其他技巧，來決定主要需求和特性。有一或多個系統分析師會促成這些階段作業。這些需求研討會是最有用的誘導技巧之一。此流程包括使用者、服務中心人員、企業主、測試人員和對提出的專案結果有利害關係的其他人在內。關係人要求通常語意不明、重疊、甚至多餘。除了正式誘導結果之外，關係人要求可以用格式完善的文件、瑕疵和來自資料庫的加強要求，或電子郵件和群組軟體作業執行緒來表達。系統分析師記錄、分類和優先順序化已識別的主要關係人需求。

**瞭解關係人需求：「滿意的客戶」由此開始**

關係人要求儘量以要求的關係人的語言和格式來擷取。後續的需求類型通常是由流程教導和技術經驗豐富的專案小組成員來編寫，關係人要求則不同，它們通常表達不夠清楚。它們會重複或重疊。它們可以在任何地方表達，從紙條到增強要求的資料庫都可以。

分析師（或代表分析師角色的團隊）必須檢視所有要求，詮釋、分組或是重新輸入（不必重寫），以及將它們轉換成願景文件中的主要關係人需求和系統特性。視開發環境中套用的精確程度和工具的可用性而定，可應用部分或全部關係人要求、需求和特性之間的可追蹤性，協助關係人瞭解其要求和需求在決定系統的需求上佔了多少份量。

運用「瞭解關係人需求活動」來展現對誘導要求和滿足關係人需求的嚴重關切，對於建立關係人對開發團隊能力的信心很重要。

基於更充分的瞭解關係人需求，開發團隊中的系統分析師修正願景文件時，特別留意闡述「產品定位陳述」。此陳述以兩三個句子建立專案的價值，令人信服。此陳述應包括預定使用者、它解決的問題、它提供的好處，以及被它取代的競爭對手。所有團隊成員都應該瞭解此專案主題。系統分析師也更新名詞解釋來促成大家對詞彙的共識。

主要關係人在多個時間點諮詢，以協商因瞭解關係人需求而衍生的新特性的優先順序，並對於要開發它們所需的資源和人力瞭解其最新資訊。和問題分析一樣，在此活動中管理相依關係有助於管理範圍。它也建立關係人要求、需求和系統特性之間的可追蹤性，使關係人可以確定其輸入有列入考量。

## 活動：定義系統

前兩個需求活動是「分析問題」和「瞭解關係人需求」，它們建立主要系統定義的初期反覆，包括願景文件中指定的特性、使用案例模型的第一個概要以及起始需求屬性。下一個活動是「定義系統」，它以特性定義的修正、新參與者的增加、使用案例和增補規格，來完成高階系統需求的說明。

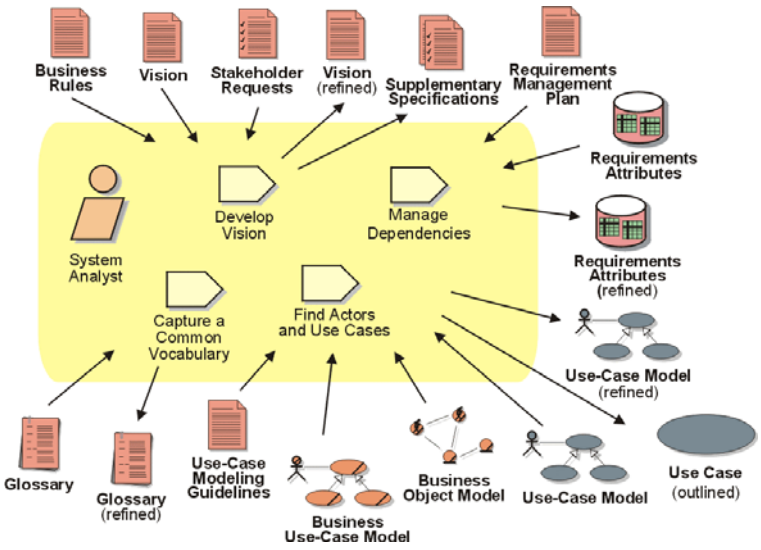


**圖 12 — 定義系統**

*活動：定義系統*

名詞解釋已更新，可反映出目前對於用來描述使用案例模型和增補規格中擷取的特性和需求的詞彙的瞭解。

系統分析師使用已修正願景中所定義的特性集來衍生及描述使用案例，使用案例闡述使用者對系統預期行爲的觀點。使用案例模型作爲客戶、使用者和系統開發人員之間對於已選取的特性在系統中如何運作的合約。它幫助系統開發人員設定實際期望和目標，並幫助客戶和使用者驗證系統是否符合這些期望。

部分系統需求並不適合使用案例。系統分析師會在增補規格中描述這些需求。許多非功能需求，例如可用性、可靠性、效能和支援性需求通常在這裡結束。另請注意，這些類型的許多非功能需求是單一使用案例特有的。使用案例指定元最好將這些需求放在使用案例規格本身（請參閱「修正系統」活動），而將增補規格留給全系統非功能需求。

在此活動中，系統分析師建立和設定增補需求的屬性（例如優先順序和相關使用案例）。此外，系統分析師會新增和更新起始的和新的使用案例的屬性值。

最後，系統分析師會將重要使用者需求和重要特性追蹤到增補規格所描述的相關使用案例和需求，來管理相依關係。

### 活動：管理系統的範圍

在識別特性層級需求、描述大部分參與者、使用案例和增補規格所指定的其他需求之後，系統分析師應該收集及儘量精確地指定一些值給需求屬性，例如優先順序、人力、成本和風險。這樣可以更充分的瞭解如何決定系統版本的起始範圍，也可以讓建構師識別在架構上很重要的使用案例。

由專案和開發管理並列開發的反覆計劃，會先出現在此活動中：管理系統的範圍。反覆計劃即所謂的開發計劃，它定義爲該版本規劃的反覆數目和頻率。初期反覆應該規劃範圍內的最高風險元素。

「管理範圍」活動的其他重要輸出包括軟體架構文件的起始反覆和一個已修訂的願景，此願景反映出分析師和主要關係人對系統功能和專案資源更多的瞭解。

如前所述，和企業案例一樣，也是反覆計劃的第一個問題，軟體架構文件並不是需求管理的一個構件，不過它卻是相關的，而且是 Rational Unified Process 的一部分。它並不是本白皮書的主題。

根據經驗，成功管理範圍的關鍵在於有將仔細考慮過的屬性值指派給關係人需求、特性、使用案例和增補規格所指定的其他需求；還有與代表關係人之間定期、開放而誠懇的互動。

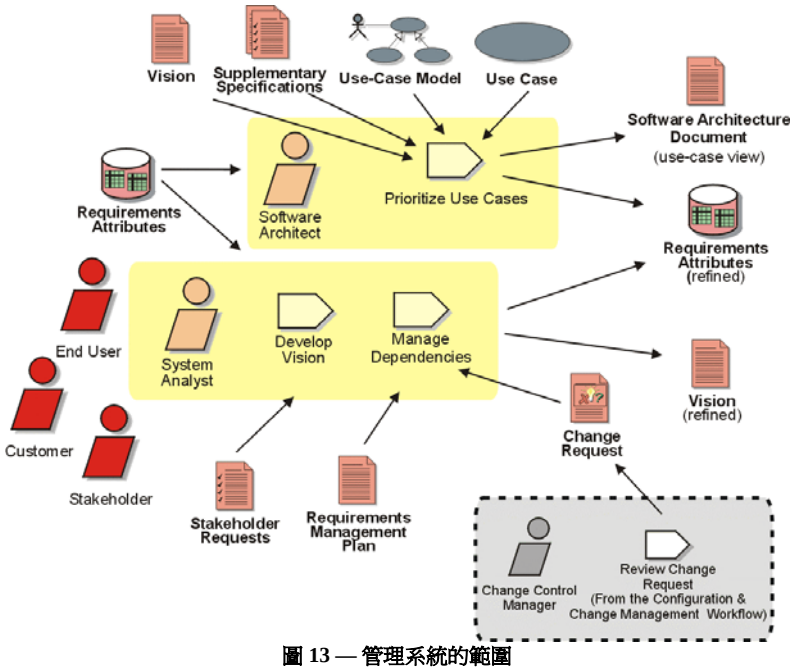


**圖 13 — 管理系統的範圍**

*活動：管理系統的範圍*

架構師依其風險涵蓋面、架構重要性和架構涵蓋面，將使用案例優先順序化。雖然系統可以由許多使用案例和增補規格需求來定義，但對於理想的系統架構而言，只有一個使用案例子集才是重要的。利用優先順序化的使用案例，架構師修正反覆或開發計劃，並在 Rational Rose 之類的工具中，爲系統架構的使用案例視圖建模。

系統分析師在願景中修正特性的需求屬性，來管理相依關係。他們也在使用案例和增補規格中修正需求。系統分析師確定關係人需求、特性、使用案例需求和增補規格需求有適當的可追蹤性存在。「適當的可追蹤性」這個詞彙是謹慎的。請參閱本白皮書後面的「可追蹤性」中的嵌入文字。

此步驟是整個專案中最重要的步驟之一。這是第一次對於所提出系統有這麼大幅度的相關知識可用，而能夠對需求、專案資源和交付日期做出嚴肅的承諾。同時，也必須瞭解，這些需求將隨著知識增加的深度而變更。如果在前三個活動中已管理相依關係，此步驟就容易多了，未來的變更也會簡單多了。

在專案的整個生命週期內，隨著狀況和環境的變更，會重新造訪此活動許多次，因爲系統分析師必須與主要關係人協商已修訂的專案範圍和願景。管理專案的範圍以符合可用的資源，此作業若要成功，關係人和開發團隊成員必須把這個步驟當作自然進度 – 而不是對使用者期望設下埋伏，或試圖向組織勒索更多時間和金錢。此活動需要在專案的主要里程碑重複執行，以評估對系統及其問題的最新理解是否需要變更範圍。已確定的需求、預算和截止期限很難變更，但深入瞭解優先順序化的使用案例、增補規格和初期系統反覆，無可避免地會造成範圍再審。

同樣地，在到達修正階段之前，當發生變更時，在整個專案生命週期內，專案小組必須從事慣常的範圍管理，這點很重要。代表關係人必須瞭解及相信，在越來越困難的範圍協商期間，將認真考慮其優先順序和興趣。到了修正系統需求時，只有重要需求還需要協商或修改。除非已建立有效範圍管理習慣，否則專案注定成爲「死亡之旅」– 無可救藥地過度限定範圍的專案，將面臨無情地延宕和成本超支。

**活動：修正系統定義**

此活動的詳細資料朝向「修正系統定義」，其假設系統層級使用案例已有概要，且參與者至少已簡短描述過。透過管理專案範圍，願景中的特性已重新優先順序化，而被認爲可在非常固定的預算和日期達成。此活動的結果是更深入的瞭解系統功能，它們是以詳細的使用案例、已修訂的增補規格和系統本身的初期反覆來表達。

顯然，並非所有系統都有使用者介面，而且並非所有初期反覆都包含 GUI 元素。我們在這裡使用它們，只是要作爲初期反覆的一個範例而已。其他範例包括原型、模型和分鏡腳本。

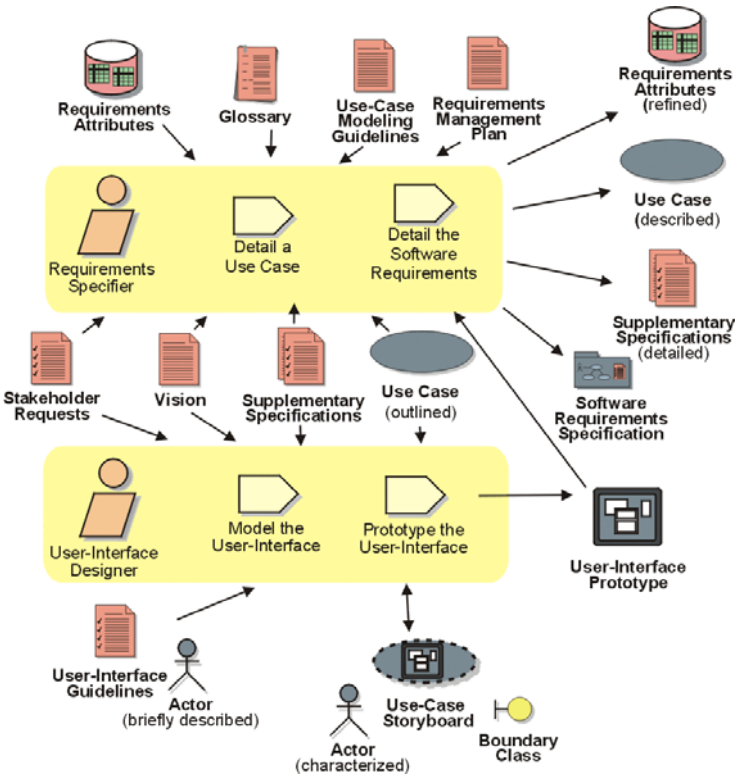


**圖 14 — 修正系統定義**

*活動：修正系統定義*

使用案例指定元詳述事件流程、前置和後置條件，及每一個使用案例的其他文字內容。若要花最少的工夫以及加強可讀性，建議使用標準文件格式或使用案例規格範本，來擷取關於每一個使用案例的文字資訊。建立經過深思熟慮的使用案例規格，對系統品質很重要。此規格開發需要充分瞭解與使用案例相關的關係人需求和特性。最好有幾個專案小組成員（例如軟體工程師）參與建立使用案例。

同時，使用案例指定元也以非使用案例特定的其他需求來修訂增補規格。

使用者介面設計師爲系統的使用者介面建模和塑型。此工作與使用案例的發展非常有關。

使用案例指定元和系統分析師爲每一個更充分瞭解的需求修訂人力、成本、風險和其他屬性值。

此系統定義修正的結果將提交至「管理範圍」活動的另一回合。當您更瞭解系統之後，您會想要變更優先順序。當然，必要的話，系統版本的範圍也需要加以檢視和修正。

### 活動：管理變更需求

當發生變更時—而且這是無可避免的—需要在專案整個生命週期內持續運用「管理變更需求」活動，如「管理系統的範圍」所討論的。此活動的輸出會造成每一個構件修正，因而需要所有專案小組成員和關係人之間有效率地溝通。

在此活動中，我們建立受到需求活動影響的其他構件。需求的變更自然會影響在分析和設計活動中呈現的系統模型。需求變更也會影響爲了驗證需求的適當實作而建立的測試。如前面範例所示，這些構件是 Rational Unified Process 的一部分，但並不是本白皮書的主題。在管理相依關係的流程中所識別的可追蹤性關係是瞭解這些影響的關鍵。

**可追蹤性**

在需求欄位中，很多是由可追蹤性組成。其中有許多爲每一個相關的規格、測試、模型元素和最後程式碼檔案提升追蹤個別客戶需求的長處。當然，有些可追蹤性是成功需求變更管理的關鍵。

然而，我們要預先警告您，所有形式的可追蹤性都需要在專案生命週期內加以設定和維護。和所有其他投資一樣，視您的特定情況而定，可追蹤性也有消失返回點。本白皮書強調不同類型的需求之間的追蹤價值。這是最佳起點，可利用工具自動化，例如 Rational RequisitePro、Rational Rose、Rational SoDA 和 Rational TeamTest。我們相信，您會發現一些需求可追蹤性層級是理想的投資。

如需需求可追蹤性的不同策略的相關資訊，請參閱白皮書「以使用案例管理需求的可追蹤性策略」[6]。

「管理變更需求」的另一個重要概念是需求歷程追蹤。藉由擷取需求變更的本質和基本理由，檢視者（軟體專案小組中受此變更影響的任何人）將得到所需的資訊來適當地回應變更。

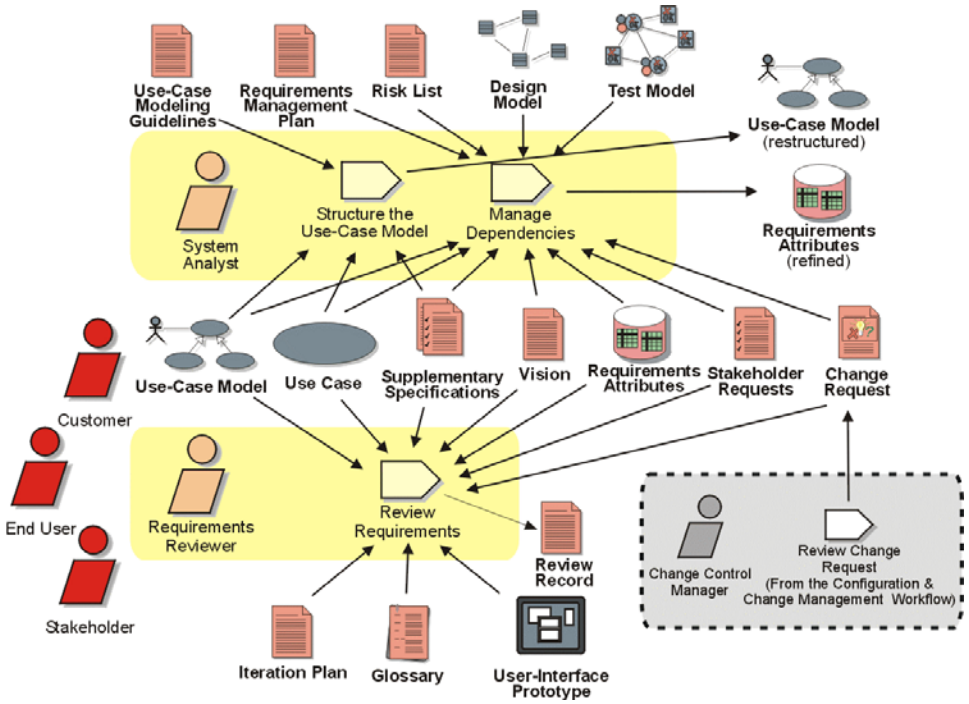


**圖 15 — 管理變更需求**

*活動：管理變更需求*

需求變更的要求可由任何關係人或專案小組成員，基於不同原因而起始。所有變更要求 (CR)，包括對需求或加強功能要求和瑕疵的變更，應該全部透過相同的「變更要求管理 (CRM)」流程來傳送。至少，應該包括在集中式資料庫系統中記載和追蹤要求，而且應該由中央檢視權限來執行檢視。如需這種 CRM 流程的詳細資料，請參閱 Rational Unified Process 的其他章節。

系統分析師應協調檢視活動，最好由 CCB（變更控制委員會）進行，收集和檢視所有變更要求，並將它們分類如下：

- 不影響需求的實作問題
- 對某種類型的現有需求的修正
- 加強功能要求

分類之後，對需求提出的變更已指派屬性和值，如其他需求活動中所描述。

在檢視變更要求時，系統分析師會向 CCB 呈現提出的優先順序化需求變更，CCB 是由代表關係人和專案小組成員所組成。超出資源的範圍修正應該遭到拒絕，或提升到關係人代表的層級，他們能夠核准對日期和預算承諾的必要變更。

CCB 核准或拒絕需求的變更。

系統分析師將需求變更傳達給需求指定元，或直接變更願景、使用案例、增補規格文件或其他需求構件中的需求。

需求檢視者（開發人員、測試人員、經理和其他專案小組成員）以檢視需求歷程的方式，來評估變更對其工作需求的影響。最後，他們實作變更，並對他們有權限的相關需求做出適當的變更。

## *摘要*

管理需求並不是新的要求。因此，先前的資訊爲何現在值得考慮？

首先，如果專案未經常滿足客戶、符合截止日期且保持在預算以內，那麼您就有理由重新考慮開發方法。如果這麼做時，您認爲需求相關問題暗中破壞您的開發工作，您就有理由考慮更好的需求管理作法。

第二，本白皮書摘要列出的需求管理作法包含數千種集體經驗，和一些個人的寶貴意見，這些人花了好幾年的時間與客戶埋首於需求管理中。我們建議對其貢獻的總覽—和以 Rational Unified Process 更徹底地呈現它們—足以代表需求管理的「最佳作法」。您不可能再找到這麼好的建議了。

作者想要對 Dr. Ivar Jacobson 和 Dean Leffingwell、Dr. Alan Davis、Ed Yourdon 及 Elemer Magaziner 等人的直接和間接貢獻表達感謝之意。最重要的是，我們十分感激 Rational Software 的客戶在數百個開發專案中應用及改良這些作法。

最後，在過去這兩年，Rational Software 是製作有效軟體開發解決方案的企業領導者，它接受需求管理的挑戰，並隨著此困難工作的自動化工具問世而嶄露頭角。需求管理的長期普遍問題是可以解決的。到最後，這可能是目前在需求管理中展開卓越練習的最佳理由。

如需知道上述概念的完整討論，請參閱 Dean Leffingwell 針對「管理軟體需求」所寫的一本佳作 [7]。

## *參照*

[1] CHAOS, The Standish Group International, Inc., Dennis, MA, 1994, 1997.

[2] Computer Industry Daily, December 12, 1997.

[3] Dorfman, M. and R. Thayer, *Software Engineering*, IEEE Computer Society Press, Los Alamitos, CA, 1997 pp. 79.

[4] Dorfman, M. and R. Thayer, *Software Engineering*, IEEE Computer Society Press, Los Alamitos, CA, 1997 pp. 80.

[5] Spence, Ian and Leslee Probasco, *Traceability Strategies for Managing Requirements with Use Cases*, White Paper, Rational Software Corporation, 1998.

[6] Rational Unified Process , Rational Software Corporation, Cupertino, CA, 1999.

[7] Leffingwell, Dean and Don Widrig, *Managing Software Requirements — A Unified Approach*, Addison-Wesley, 2000.

Rational
the software development company

兩個總公司：

Rational Software
18880 Homestead Road
Cupertino, CA 95014
電話：(408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
電話：(781) 676-2400

免付費專線：(800) 728-1212
電子郵件：info@rational.com
網址：www.rational.com
國際辦事處：www.rational.com/worldwide