

# 利用 Rational Unified Process 達到 CMM 等級 2 與 3

Rational Software 白皮書

---

TP 174

## 目錄

摘要.....	2
簡介.....	2
等級 2, 可重複.....	3
要件管理.....	3
軟體專案規劃.....	5
軟體專案追蹤和監督.....	6
軟體轉包管理.....	6
軟體品質保證.....	7
軟體配置管理.....	7
等級 3, 已定義.....	8
組織流程焦點.....	8
組織流程定義.....	9
培訓計劃.....	9
整合性軟體管理.....	9
軟體產品工程.....	10
群組間的協調.....	10
同僚檢視.....	10
參照.....	11



## 摘要

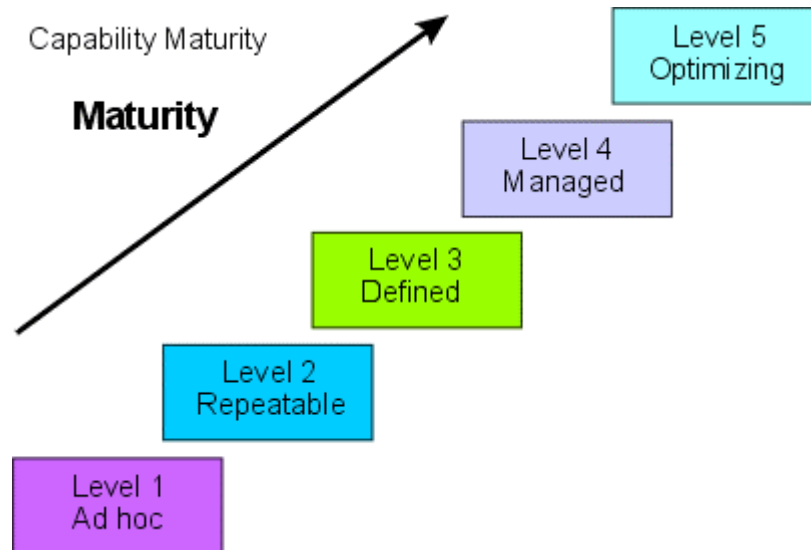
軟體工程協會 (Software Engineering Institute, SEI) 的能力成熟度模式 (Capability Maturity Model, CMM) 提供軟體流程成熟度的知名基準性能測試。CMM 已成為許多領域評量組織軟體流程成熟度的普及工具。本白皮書說明 Rational Unified Process 如何支援組織達到 CMM 等級 2 (可重複) 和等級 3 (已定義) 的軟體流程成熟度等級。

## 簡介

軟體工程協會 (SEI) 的能力成熟度模式 (CMM) 是一種架構，它說明有效軟體流程的元素；請參閱本文件結尾的 *References* [REF 1] 這一節。CMM 說明從特定的不成熟流程到成熟而有紀律的流程的改革之路。

CMM 涵蓋規劃、設計和管理軟體開發與維護的實作要領。這些重要實作要領增進組織的能力，達到成本、排程、功能和產品品質的目標。

CMM 有五個成熟度等級：等級 1 到等級 5。如下圖所示，每一個成熟度等級是由關鍵流程區 (KPA) 所組成，每一個 KPA 識別相關活動叢集。當這些相關活動共同執行時可達到一組目標，這對於建立該成熟度等級的流程能力很重要。



在等級 2 (可重複等級)，會建立管理軟體專案的原則和實作那些原則的程序。規劃和管理新專案是以類似專案的體驗作為基礎。達到等級 2 的目標是為了使軟體專案的有效管理流程制度化，讓組織能夠重複先前專案所開發出來的成功實作要領，不過專案實作的特定流程可能不同。有效流程的特徵是：已實作、記載、強制、訓練、評量，且能夠改進。

等級 2 組織內的專案已安裝基本軟體管理控制。實際專案的承諾會以在先前專案上觀測到的結果和現行專案的需求為依據。專案的軟體管理者負責追蹤軟體成本、排程和功能，並找出履行承諾時出現的問題。軟體需求和為了符合軟體需求而開發出來的工作產品有基準為依據，其整合性受到控制。有定義軟體專案標準，而組織要確保能夠確實遵守。軟體專案要搭配轉包商（如果有的話）來建立穩固的客戶供應商關係。

等級 2 組織的軟體流程能力可以「有紀律」三個字來作總結，因為軟體專案的規劃和追蹤穩定，而且可重複之前的成功經驗。專案的流程在專案管理系統的有效控制之下，以先前專案的效能作為基礎來遵循實際規劃。

等級 2 KPA 為：

- 要件管理

- 軟體專案規劃
- 軟體專案追蹤和監督
- 軟體轉包管理
- 軟體品質保證
- 軟體配置管理

在等級 3（已定義等級），會記載在組織內開發及維護軟體的標準流程，包括軟體工程和管理流程，且這些流程會整合成一體。標準流程在 CMM 內統稱為組織的標準軟體流程。組織使用（並適當地變更）等級 3 建立的流程，來幫助軟體管理者和技術人員更有效率地執行工作。組織利用有效軟體工程實作要領，將其軟體流程標準化。有一組人負責組織軟體流程活動；例如軟體工程師或 SEPG。實作組織培訓計劃，以確保技術人員和主管具備必要的知識和技能來履行指定的角色。

專案可修改組織標準軟體流程，開發它們自己定義的軟體流程，以彰顯專案特質。修改過的流程在 CMM 中稱為專案已定義的軟體流程。已定義的軟體流程包含一致、整合、定義完善的一組軟體工程和管理流程。定義完善的流程，其特色為具有整備準則、輸入、標準和執行工作的程序、驗證機制（例如同僚檢視）、輸出和完成準則。因為軟體流程定義完善，所以管理部門對所有專案的技術進度瞭如指掌。

等級 3 組織的軟體流程能力可以用標準和一致作總結，因為軟體工程和管理活動均穩定而且可重複。在已建立的產品系列中，成本、排程和功能均受到控制，並會追蹤軟體品質。此流程能力是以對於定義的軟體流程中的活動、角色和責任有全盤瞭解為基礎。

等級 3 KPA 為：

- 組織流程焦點
- 組織流程定義
- 培訓計劃
- 整合性軟體管理
- 軟體產品工程
- 群組間的協調
- 同僚檢視

本白皮書的每一節說明 Rational Unified Process 的特性、方法、程序和構件如何符合 KPA 目標。

本白皮書是為與達成 CMM 架構內的組織成熟度等級 2 和等級 3 有關的組織人員而撰寫的。

## 等級 2，可重複

---

### 要件管理

需求管理的目的是要在客戶和要解決客戶需求的軟體專案之間建立共識。與客戶的共識是規劃（「軟體專案規劃 KPA」所述）和管理（如「軟體專案追蹤和監督」所述）軟體專案的基礎。如何控制與客戶之間的關係，視下列有效變更控制流程而定，如「軟體配置管理」所述。

Rational Unified Process 的主要特性之一為*使用案例驅動*。使用案例代表有計劃有步驟地誘導、組織和溝通使用者需求的方法。它們提供記載功能需求的方式，作為專案開發、測試和反覆規劃的基礎。在 Rational Unified Process 中，使用案例會保存在使用案例模式中，從分析、測試到維護的專案生命週期內會一貫地參考它們。

在工程內容中擷取需求的 Rational Unified Process 構件如下：

- 由使用案例和使用案例資料包組成的使用案例模式

- 非功能性增補規格
- 使用案例模式調查
- 使用案例報告
- 名詞解釋

正如在管理內容中所使用的，說明使用案例和要開發的實務（需求）的 Rational Unified Process 構件如下：

- 反覆計劃
- 整合建置計劃
- 專案計劃
- 軟體開發計劃

所有這些構件都有基準，並受到變更管理紀律的約束。

**目標 1：配置給軟體的系統需求受到控制，以建立軟體工程和管理用途的基準。**

Rational Unified Process 主張對所有發展中的構件進行配置控制，不過，「正式」基準則對應到下列里程碑。

- 生命週期目標里程碑（開始階段）
- 生命週期架構里程碑（詳細闡述階段）
- 起始作業功能里程碑（建構階段）
- 產品版本里程碑（轉變階段）

就其本身而論，Rational Unified Process 在需求、管理、追蹤和基準設定的協議上與 CMM 一致。

**目標 2：軟體計劃、產品和活動與已配置給軟體的系統需求保持一致。**

強調這個 CMM 目標是爲了確保所交付的系統符合使用者需求。Rational Unified Process 以兩種方式幫助組織達到此目標：

- 使用案例方法確保使用者需求獲得瞭解和擷取。擷取之後，需求就會流向不同「視覺化」Rational Unified Process 模式（使用案例、設計、實作和測試），以確保一致性和遵循性。
- 受管制的反覆漸進式開發方法是一種減輕風險的方法，能夠提早瞭解和探索專案的風險，而敢於再度面臨風險。每一個進行中的反覆，經由持續整合新增的功能，可提早揭露風險。若使用傳統瀑布式方法，這些風險可能要等到開發生命週期後期才會被發現。透過需求重定範圍或提示其他策略性變更，因而能早期識別風險，這樣對於管理專案有直接的益處。

Rational Unified Process 管理文件如下：

- 商業案例
- 軟體開發計劃
- 測量計劃
- 風險清單
- 專案計劃
- 反覆計劃
- 反覆評量和狀態評量

有效變更控制和管理是另一個 Rational Unified Process 特性，可確保軟體的開發符合指定的、追蹤的和配置的需求。Rational Unified Process 主張每一個專案要建立「變更管制委員會」(Change Control Board, CCB)，對於所提議變更的範圍和影響（預算上、技術上或排程上）或在開發過程中發現的問題進行仲裁。為了協助 CCB 運作，Rational Unified Process 建議使用強大的配置管理和版本控制工具/環境。

### 軟體專案規劃

軟體專案規劃的目的是為了建立合理的計劃來執行軟體工程和管理軟體專案。這些計劃是管理軟體專案的必要項目，一如軟體專案追蹤和監督一節所述。若無實際可行的計劃，就無法實作有效的專案管理。

#### 目標 1：記載軟體預估，以使用於計劃和追蹤軟體專案。

Rational Unified Process 的目標之一是要確保各方期望能夠同步和一致。這個目標是透過整個專案生命週期內的定期評量加以確保，並記載於狀態評量報告中。此報告需要追蹤資源上的資料（人員配置和財務狀況）、前十大風險、透過測量值判斷的技術進度和主要里程碑結果。

Rational Unified Process 使用下列測量值類別：

- 進度（程式碼行、類別數、每一個反覆的功能點、重做）
- 穩定性（重做類型、需求或實作變化性）
- 可改造性（重做成本）
- 模組性（重做影響的範圍）
- 品質（問題發現率、密度、繼承程度、重做指示器）
- 成熟度（每次失敗的測試時數）
- 資源支出設定檔（計劃支出與實際支出）

#### 目標 2：計劃與記載軟體專案活動和承諾。

擷取專案計劃和承諾的 Rational Unified Process 文件如下：

- 商業案例
- 軟體開發計劃
- 測量計劃
- 風險清單
- 專案計劃
- 反覆計劃
- 反覆評量
- 狀態評量

#### 目標 3：受影響的群組和個人同意他們對軟體專案的相關承諾。

在 Rational Unified Process 中，軟體開發計劃定義專案的整體計劃；反覆計劃詳細定義在反覆中要完成的目的。Rational Unified Process 所需的「反覆計劃檢視」向所有關係人顯現反覆計劃，允許在反覆開始之前得以形成共識。從同意的反覆計劃中，專案管理者製作一系列工作訂單，向受影響的專案團隊和個人詳細溝通反覆計劃的目的。專案管理者與受影響的人員對這些工作訂單達成協議，使反覆得以進行。

## 軟體專案追蹤和監督

軟體專案追蹤和監督的用途是爲了建立對實際進度的足夠可見性，使管理者在軟體專案的效能與軟體計劃大相逕庭時可以採取有效的行動。

### 目標 1：依據軟體計劃來追蹤實際結果和效能。

如軟體專案規劃一節所述，Rational Unified Process 有數個專案計劃等級和一個狀態評量報告，這份報告是爲了評定計劃效能與實際效能而產生的。爲了特定里程碑而產生的這份報告，由專案管理者負責。

Rational Unified Process 主要里程碑對應一個階段的結束（開始、詳細闡述、建構或轉換），並已妥善指定完成準則。請在階段內的每一個反覆結束時檢視次要里程碑的機會，並作爲決策點和未來方向的學習課程。

例如：詳細闡述階段的目標是要分析問題領域、建立健全的架構基石、發展專案計劃和排除專案的最高風險元素。做出架構決策之前，必須先瞭解整個系統。這意味著將一些限制列入考慮之後，大部份使用案例可描述爲：增補需求。爲了驗證架構，以實作系統的方式來示範架構選項和執行重要的使用案例。

在詳細闡述階段結束時，會檢查詳細系統目標和範圍，以及架構選擇和主要風險的解決之道。當實際結果和效能與軟體計劃大相逕庭時，會採取更正動作並設法完成。

風險清單是一個 Rational Unified Process 構件，它提供專案所有已知風險的概觀，並作爲規劃和專案評量的輸入。每一個風險會從它的影響和爲了降低問題風險而行使的延續計劃方面加以說明。風險清單是因應商業案例而研擬的，形成專案「做」與「不做」決策的基礎。在專案生命週期內會保留一份風險清單。

### 目標 2：當實際結果和效能與軟體計劃大相逕庭時，會採取更正動作並設法完成。

Rational Unified Process 需要在兩個等級上追蹤效能：專案管理者將使用已收集的標準來產生狀態評量，作爲例行監督和專案控制的一部分。狀態評量中所確定的問題會根據專案的問題解決計劃加以處理，可能是直接由專案管理員（利用工作訂單）處理，如果是範圍較廣的問題，則需透過「變更要求」。此外，每一個反覆的結果視「反覆評量和檢視」而定。這樣可讓先前的反覆經驗引導後續反覆的規劃，以及透過「變更要求」管理更正動作。

### 目標 3：受影響的群組和個人同意軟體承諾的變更。

如 Rational Unified Process 中所述，受管制的反覆漸進式開發流程可確保關係人定期得知專案進度和持續追蹤專案所需的變更。提出的變更由「變更管制委員會」(Change Control Board, CCB) 檢視，以確保這些變更實際可行，可以納入整體專案排程中。

## 軟體轉包管理

轉包管理的目的是要選取合格的軟體轉包商並有效加以管理。它結合了對基本管理控制的需求管理、軟體專案規劃和軟體專案追蹤和監督的考量，以及軟體品質保證和軟體配置管理的必要協調，並適當地將此控制套用至轉包商。

### 目標 1：主要承包商選取合格的軟體轉包商。

### 目標 2：主要承包商和軟體轉包商同意彼此的承諾。

### 目標 3：主要承包商和軟體轉包商保持不間斷的溝通。

### 目標 4：主要承包商根據轉包商的承諾來追蹤軟體轉包商的實際結果和效能。

這些目標超出 Rational Unified Process 的現行範圍之外，隸屬於組織的權責。

Rational Unified Process 並未特別處理轉包的問題，已假設其工具、技術和機制會流向轉包商，使流程均一。

所有轉包決策應該記載於商業案例中。遵循與主要承包商相同開發計劃的轉包商也會參與相同技術交換、主要里程碑和狀態評量。



## 軟體品質保證

軟體品質保證的目的是要讓管理者對軟體專案使用的流程和建置的產品有適當的瞭解。軟體品質保證是大部份軟體工程和管理流程整體的一部份。

Rational Unified Process 將品質視為所有專案人員的群體責任，就其本身而言，並未在任何組織內具體化。

### 目標 1：規劃軟體品質保證活動。

規劃軟體品質保證作業是組織的責任。然而，Rational Unified Process 有一些屬性有助於形成有效專案品質保證計劃。

每一個 Rational Unified Process 里程碑有特定的完成準則，可作為審核的基礎。在 Rational Unified Process 內，有個別的檢視作業。每一個檢視有一組相關聯的核對點，代表在進入下列活動之前需要通過的閘道。

Rational Unified Process 為需要檢視構件的人提供指引。例如，由設計師所執行之「使用案例分析」的結果，需要由一位獨立的建構師、設計師、使用案例設計師和設計評論家來檢視。有了已定義的 Rational Unified Process 和構件檢視準則之後，與產品品質有關的客觀團體就比較容易評定流程的遵循性和開發標準與準則的相符度。

### 目標 2：客觀驗證軟體產品和活動對適當標準、程序和需求的遵循性。

可透過挑選組織的品管人員來達到此目標。不過，Rational Unified Process 所提供的必要檢視核對清單和文件範本，可套用作為專案標準。

### 目標 3：將軟體品質保證活動和結果通知受影響的群組和個人。

如軟體專案規劃所述，Rational Unified Process 的目標之一是要確保各方期望能夠同步和一致。除了品質審核結果的輸入之外，Rational Unified Process 需要資源的報告（人員配置和財務狀況）、前十大風險、透過測量值判斷的技術進度和主要里程碑結果。Rational Unified Process 測量值計劃提供收集下列測量值的準則：

- 進度（程式碼行、類別、每一個反覆的功能點）
- 穩定性（重做類型、變化性）
- 可改造性（重做成本）
- 模組性（重做影響的範圍）
- 品質（問題發現率、密度、繼承程度）
- 成熟度（每次失敗的測試時數）
- 支出設定檔（計劃支出與實際支出）

### 目標 4：無法在軟體專案內解決的不依循問題交由資深管理者處理。

這超出 Rational Unified Process 的範圍之外，是組織的責任。然而，Rational Unified Process 所述的變更控制流程將啟用一種機制，來記載不依循問題並提升到解決層次。

## 軟體配置管理

軟體配置管理的目的是要在專案的軟體生命週期內建立及維護軟體專案產品的完整性。軟體配置管理是大部份軟體工程和管理流程整體的一部份。

### 目標 1：規劃軟體配置管理活動。

如 Rational Unified Process 中所述，優良的配置管理是受管制的反覆漸進式開發方法中的一個重要元素。由於軟體是分階段形成，所以之前開發工作的軟體版本能否供後續開發使用非常重要。規劃如何在每一個階段產生可確實運作的軟體，是 Rational Unified Process 的核心工作。

Rational Unified Process 有兩個主要工具來定義如何維護專案軟體開發資產及如何整合它們。

- 配置管理計劃
- 整合建置計劃

在開始階段起始的配置管理計劃說明下列各項：

- 管理軟體版本化和處理
- 儲存給定的 Rational Unified Process 模式並將它們分成配置項目
- 使用變更控制方法來管理變更和版本

整合建置計劃提供要建置的配置項目及要在給定的反覆中整合它們的順序的詳細資料。

#### 目標 2：識別、控制及提供已選取的軟體工作產品。

Rational Unified Process 配置管理計劃需要配置控制和管理流程的說明，以確保確實識別、控制及提供工作產品。

#### 目標 3：控制已識別之軟體工作產品的變更。

Rational Unified Process 主張專案要維護一個「變更管制委員會」(Change Control Board, CCB)，且具有「變更管理系統」來適當地管理、估計成本、追蹤和實作變更要求。

#### 目標 4：將軟體基準的狀態和內容通知受影響的群組和個人。

Rational Unified Process 主張以電子格式維護需求、設計和實作基準及彼此之間的可追溯性。專案控制的不同等級仲裁基準的一切變更。例如，「變更管制委員會」(Change Control Board, CCB) 會考量需求等級變更的影響。範圍較小的設計和實作變更，會在技術權限的適當等級加以檢視。在配置管理計劃和軟體開發計劃中，會說明核准、控制等級及其溝通方式。

## 等級 3，已定義

---

### 組織流程焦點

組織程序焦點的用途是要建立組織對軟體流程活動的責任，來提升組織的整體軟體流程能力。組織流程焦點活動的主要結果是一組軟體流程資產，「組織流程定義」中將有所說明。這些資產由軟體專案使用，如「整合性軟體管理」中所述。

#### 目標 1：透過組織協調軟體流程開發和改進活動。

Rational Unified Process 是一種反覆流程，它取決於透過一些反覆重新制定同一個已定義的流程。流程制定的這種重複性質和在每一個階段和反覆中得到的狀態測量值的評量，提供微調每一個連續反覆流程的機會。

#### 目標 2：以相對於流程標準的方式識別軟體流程的優點和弱點。

Rational Unified Process 代表一個整體軟體開發流程，可基於任何類型的專案的有效運用而量身訂作。在環境紀律中有提供如何量身訂作 Rational Unified Process 的指引。除了技術和管理複雜度之外，還有一些流程區別元件將決定在專案中使用的流程「形式」，它們是：

- 商業背景（合約、純理論或內部）
- 軟體開發規模
- 創新程度
- 應用類型

### 目標 3：規劃組織等級流程開發和改進活動。

等級 3 目標完全取決於採用的組織。

### 組織流程定義

「組織流程定義」的目的是要開發和維護一組可用的軟體流程資產，透過專案改進流程效能，為組織提供長期累積利益的基礎。這些資產提供穩定的基礎，可透過培訓（如「培訓計劃」中所述）之類的機制加以制度化。

### 目標 1：開發及維護組織的標準軟體流程。

Rational Unified Process 可提供起點，作為組織的基準軟體開發流程，並加以發展、量身訂作和維護。

### 目標 2：收集、檢視及提供與軟體專案使用的組織標準軟體流程相關的資訊。

採用 Rational Unified Process 的組織需要支援此目標。

### 培訓計劃

「培訓計劃」的目的是要開發個人技能和知識，使他們能夠實際而有效地扮演其角色。培訓是組織的責任，但是當軟體專案的需求與眾不同時，軟體專案應該識別它們所需要的技能，並提供必要的培訓。

### 目標 1：規劃培訓活動。

只有採用 Rational Unified Process 的組織才能夠達成此目標。然而，Rational Unified Process 是「產業最佳作法」知識庫，能提供有關如何執行各種軟體開發活動的準則、概念和詳細逐步說明。因此，Rational Unified Process 本身就是培訓資料的不錯來源。

Rational Unified Process 相關支援課程包括：

- 「Rational Unified Process 概觀」，有需求、分析和設計、實作、架構、流程配置、管理、工具和物件導向簡介等單元。
- 有使用案例的需求管理 (RMUC)
- 物件導向專案管理 (OOPM)
- 物件導向設計和分析 (OOAD)
- 自動管理軟體品質
- 配置管理
- 軟體架構和反覆流程

### 目標 2：提供培訓來發展所需技能和知識，以扮演軟體管理和技術角色。

### 目標 3：軟體工程群組和軟體相關群組中的個人接受必要的培訓課程以扮演好其角色。

採用 Rational Unified Process 的組織需要達成這些培訓計劃的目標。不過，Rational Unified Process 有提供課程範圍，如上一節所述。

### 整合性軟體管理

「整合性軟體管理」的目的是要將軟體工程和管理活動整合成一致定義的軟體流程，它是從組織的標準軟體流程和相關流程資產量身訂作而成，如組織流程定義一節所述。量身訂作需根據專案的商業環境和技術需求而定，如「軟體產品工程」所述。「整合性軟體管理」是從等級 2 的「軟體專案規劃」和「軟體專案追蹤和監督」引申出來。

### 目標 1：專案已定義的軟體流程是組織標準軟體流程的量身訂作版本。

根據 Rational Unified Process *環境紀律*，Rational Unified Process 的標準交付是可設定的，並可重定範圍供不同類型的專案使用。

### 目標 2：根據專案已定義的軟體流程來規劃和管理專案。

採用 Rational Unified Process 的組織需要達成此目標。

## 軟體產品工程

「軟體產品工程」的目的是要一貫執行定義完善的設計流程，整合所有軟體工程活動，實際而有效地產生正確而一致的軟體產品。「軟體產品工程」說明專案的技術活動；例如，需求分析、設計、撰寫程式碼和測試。

### 目標 1：定義、整合和一貫執行軟體工程作業來產生軟體。

根據必要的專案規劃構件的背景，Rational Unified Process 作業和每一個角色需求的定義可確保已定義、配置和完成這些作業。Rational Unified Process 內含的反覆漸進式開發流程可快速證明軟體開發團隊的效率，並提供終端產品的評量。

### 目標 2：軟體產品彼此一致。

環境維護設計模式之間的可追溯性，例如使用案例模式、設計模式、程式碼和執行檔元件。

## 群組間的協調

「群組間的協調」的目的是要建立一種方式，讓軟體工程群組能夠積極參與其他設計群組，使專案更能夠實際而有效地滿足客戶的需求。「群組間的協調」是「整合性軟體管理」中延伸到軟體工程之外的科技整合層面；不只應該整合軟體流程，而且應該協調和控制軟體工程群組與其他群組的互動。

### 目標 1：所有受影響的群組都同意客戶的需求。

使用案例可作為需求擷取和透過其他「正式」需求指定方法的基礎，它的一個實質好處就是使用案例可以很快地由相關關係人瞭解。因此，擷取使用案例需求的 Rational Unified Process 方法，表示所有關係人可以同意要完成的事項。這可以透過此流程進一步完成，並反映在作為軟體開發基礎的模式和檢視中。

### 目標 2：受影響的群組同意設計群組之間的承諾。

採用 Rational Unified Process 的組織需要達成此目標。然而，Rational Unified Process 視覺化模式有助於瞭解產品開發每一個階段（從需求擷取到部署）的必要項目。「Rational Unified Process 變更和配置管理」流程可確保提出的變更經過適當評估並傳達給所有關係人。

設計群組可識別、追蹤和解決群組間的問題。Rational Unified Process 反覆漸進式開發流程，透過持續整合所有已開發的軟體，協助提早偵測軟體問題。與一些團隊所開發之軟體的整合問題，可作為提出和解決跨團隊問題的「開放園地」。Rational Unified Process 問題和變更要求流程支援此一想法，它提供正式機制來擷取、追蹤和解決專案開發問題。

## 同僚檢視

同僚檢視的目的是要提早並有效消除軟體工作產品的問題。它的一個重要的必然效果就是更瞭解軟體工作產品和可避免的問題。同僚檢視是在「軟體產品工程」中叫用的重要而有效的設計方法。

### 目標 1：規劃同僚檢視活動。

如「等級 2」的「品質保證」目標中所述，Rational Unified Process 內的每一個作業有個別的檢視作業。

由於提早偵測問題可降低整體成本，因此 Rational Unified Process 主張對所有構件（尤其是重要構件）「提早並經常地」進行同僚檢視。Rational Unified Process 提供要在每一個階段和每一個模式內檢視的顯著特性的核對清單。

## 目標 2：識別並消除軟體工作產品中的問題。

Rational Unified Process 構件檢閱人員必須判斷該構件是否備妥供下一個開發階段使用。如果構件不符合檢視「通過」準則，根據 Rational Unified Process 測量值計劃，需要擷取下列詳細資料：

- 穩定性（重做類型、變化性）
- 可改造性（重做成本）
- 模組性（重做影響的範圍）
- 品質（問題發現率、密度、繼承程度）
- 成熟度（每次失敗的測試時數）
- 支出設定檔（計劃支出與實際支出）

## 參照

---

[REF1] 「能力成熟度模式的重要實作要領 - 1.1 版」，由 Carnegie Mellon University 軟體工程協會 (Software Engineering Institute, SEI) 的 Mark C. Paulk 及其他人所寫。



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](mailto:info@rational.com)  
網址：[www.rational.com](http://www.rational.com)  
國際辦事處：[www.rational.com/worldwide](http://www.rational.com/worldwide)

Rational、Rational 標誌和 Rational Unified Process 是 Rational Software Corporation 在美國和/或其他國家的註冊商標。  
Microsoft、Microsoft Windows、Microsoft Visual Studio、Microsoft Word、Microsoft Project、Visual C++ 和 Visual Basic 是 Microsoft Corporation 的商標或註冊商標。所有其他名稱爲其他公司的商標或註冊商標，只做識別用途。ALL RIGHTS RESERVED. Made in the U.S.A.

© Copyright 2002 Rational Software Corporation.  
如有變更，恕不另行通知。