

Rational Unified Process – 更高 流程成熟度的啓用程式

Annie Kuntzmann – Combelles, Q-Labs France

Philippe Kruchten, Rational Software Canada

Rational Software 和 Q-Labs 白皮書

TP 178, 2/01

目錄

摘要.....	1
您的效率夠嗎？	1
改進的導覽圖	1
組織需要什麼來啟動 SPI 計劃？	2
CMM 模型 [6].....	2
IDEAL 方法 [4].....	2
流程.....	2
RATIONAL UNIFIED PROCESS.....	2
ISO 文件	3
針對 SPI 使用 RUP	3
RUP 符合專案層次的 CMM 需求	3
生命週期問題	4
取得專案願景及開發商業模型	5
組織和規劃專案.....	5
部署統一變更管理原則.....	6
流程選取和調整	7
風險管理.....	7
測量	8
RUP 符合組織層次的 CMM 需求	9
流程改進.....	9
在組織中實作流程和工具的步驟.....	9
流程實作專案可分成幾個階段	10
資源和技能.....	11
實用主義：RUP 如何入門.....	11
結論.....	14
作者的聯絡資訊.....	15
參照.....	16

摘要

本白皮書標示成熟組織 - 層次 3 開發單元 - 必須示範的主要概念，及 Rational Unified Process (RUP) 元件如何符合這些需求。

高專案成熟度和組織成熟度都做了處理。此外，第 4 節提供 RUP 入門的一些好點子，並報告在不同環境定義下的早期採用者所觀察到的一些主要好處。

您的效率夠嗎？

今天，公司必須持續搜尋能夠達到更高效能的方法。未來的挑戰是能夠在全球競爭。在全球市場競標時，這些共同徵兆變得越來越明顯：

- **效能的需求：**在驗證階段發現太多錯誤，爲了符合交付目標而將不穩定版本產品上市
- **效率的需求：**專案克服預算和規劃
- **失去市場佔有率：**競爭對手表現更好且交付更好的品質
- **缺乏足夠的資源：**人員更換率高和/或雇用新工程師非常困難
- **需要以最小的風險來整合各項發展的技術：**支援產品的技術必須遵照趨勢；團隊經驗不一定足夠應付挑戰

更白話地說，將已開發產品的每一面最佳化，已成為強烈的需求。

因此，假設開發產品的流程對結果有很大的影響，而這樣的假設被廣為接受的話，其中一個最常見的實務就是處理軟體流程來改進產品。

面對這種狀況的組織或許要採用下列解決方案才能成功。

解決方案的第一個部分是要瞭解市場的軟體價值。如果軟體繼續成為產品和專案的主要競爭優勢，則指定及實作它們的流程在一段時間後必須小心定義、記錄和最佳化。我們觀察到資深管理階層仍然相信，軟體製造問題而且太晚出現；且未執行軟體的加值分析。

解決方案的第二部分是投入足夠的時間和人力來解決流程的主要問題：

- 評定流程來識別改進機會
- 不要一切重頭來過：從別人的經驗中學習，加速突破
- 對任何專案要同時考量管理和技術觀點
- 管理以後需要的技能：學習、分享和成長

要使組織轉變成高生產力的團隊，有數個導覽圖可使用；您所做的其中一個最佳決策是：可以就改變，不能改變就接受，並盡量借用成功的軟體社群已證明有效的軟體。功能成熟度模型 (CMM) [6] 和 Rational Unified Process (RUP) [11] 是兩個廣被接受的健全工具的範例，它們可以朝著足夠的流程加快進度來開發產品。我們使用工具這個詞彙，因為這兩個元件是變更和改進軟體作法的有效支援，它們可讓您定義適合組織環境定義的最佳作法。

下一節說明這兩個元件—CMM 和 RUP—以及您在改進流程和已交付產品時必須考量的一些其他主要元件。

改進的導覽圖

在過去這十年，軟體社群變得很有流程意識，並瞭解軟體產品無法離開產生它們的流程而獨立評估。因此，有一些組織建立程式來改進其軟體開發流程，即所謂的「軟體流程改進 (SPI)」創新技術。在大部分情況下，軟體開發集結

成組織的一或多個單元或部門，且被視為獨立規範。當 SPI 計劃出現後，這種狀況有了改變，軟體團隊與組織的其他規範有了更好的整合。

組織需要什麼來啟動 SPI 計劃？

- 啟動計劃的刺激因素和協調改進活動的關鍵人物；換句話說，以願景陳述和目標來設定改進專案。
- 一種參照模型，用來評估專案和組織作法以及識別阻礙成功的任何遺漏組件。在 Software Engineering Institute (SEI) 所開發的 CMM 模型是數千個組織所使用的既定參照，而 SPICE 架構 (ISO 15504) [9] 則是很快成為 CMM 相容之正式標準的替代方案。
- 管理改進計劃來含括於成功結果的方法：在 SEI 定義的 IDEAL [4] 架構已證明它的實力。
- 爲了每一個專案的成功而改建的開發流程：RUP 是一個可被接受的現成解決方案。

CMM 模型 [6]

Software Engineering Institute (SEI) 的能力成熟度模式 (CMM) 是一種架構，它說明有效軟體流程 [6] 的元素。CMM 說明從特定的不成熟流程到成熟而有規範的流程的改革之路。它在一些主要流程區域中呈現建議的作法集，它們已展現來加強軟體開發與維護功能。CMM 指引開發人員如何獲得其開發和維護流程的控制權，及如何逐步形成軟體工程和管理才能的文化。

IDEAL 方法 [4]

軟體流程改進是一種有系統的、協同的及長期的方法，可逐步形成組織和執行軟體工作的方式。改進方法包括 IDEAL 方法，它是 SEI 所定義的 SPI 的整合方法。IDEAL 識別五個階段：起始、分析、建立、行動和操縱。每一個階段以一個特定活動爲中心：

- 指定要實現或支援的商業目標（**起始**）。
- 識別組織與相關標準或參照模型有關的現行狀態（**分析**）。
- 開發計劃來實作選擇的方法（**建立**）。
- 集合一切可用的事物來建立組織需求專用的「最佳猜測」解決方案一例如，現有的工具、流程、知識和技能一並使解決方案就定位（**行動**）。
- 總結關於用來實作 IDEAL 的流程所獲的經驗（**操縱**）。

流程

有些組織沒有流程；有些組織基於其經驗有流程。有一些現成的流程可用，例如 OPEN 和 Rational Unified Process (RUP)。OMG Group 可能很快就會提出一般流程模型。「流程」這個詞彙主要使用於 Software Engineering Community，但關於軟體流程元件的假設則非常異類。現在我們來看 SEI 提出的定義：人們用來開發及維護軟體和相關產品的一組活動、方法、作法和轉換。流程是確保專案符合其定義目標的一個主要元素。

值得注意的是，此定義包含活動、技術（即方法和工具）和人員。這三個元件同等重要。

Rational Unified Process

RUP 是早已存在的流程架構的範例之一，得益於悠久的專案經驗 [10, 11]。

RUP 藉由快速開發系統的起始版本來定義其架構，強調應儘早處理高風險區域。它沒有在專案的初始階段假設一組固定的需求，但容許您隨著專案的發展而修正需求。它可以預期及容納變更。流程不把焦點放在文件或形式上，它有助於許多與軟體開發相關聯的瑣碎作業的自動化。主要焦點留在軟體產品本身及其品質上，並依據它滿足一般使用者和它符合投資報酬率的程度來衡量。

它相當一般化，可以在大小和應用領域上，針對各式各樣的軟體產品和專案加以調整，而且以三個區域為中心：人員、流程和工具或方法 [10,11]

ISO 文件

最後，臨門一腳，我們必須提到 ISO 文件，例如 ISO 9001、ISO 12207、ISO 15504 (SPICE) [9]，它們是組織與其他組織比較其軟體開發作法時可用的參考。它們形成其他上述元件相容的一般架構。

ISO 15504（亦稱為 SPICE）是分析軟體流程的另一個參照模型。它假設許多完全實現的評量模型（CMM 就是其中之一）和許多評量方法（由 SEI 定義的方法就是其中之一）可以對映到 ISO 15504 的基準部分。

在本白皮書其餘章節，我們將焦點放在 CMM，它是軟體標準社群的既定標準。CMM 是一個完全實現的模型，支援數百個軟體流程改進創新技術。所討論的一切均符合 SPICE。

針對 SPI 使用 RUP

考量上一節列出的主要元素—CMM 參照模型、管理改進計劃直到結束的方法 (IDEAL) 和提高競爭力及面對軟體全球挑戰的調適性開發流程—我們現在將 RUP 概念對映到 CMM 需求，並仔細考慮 RUP 達到特定軟體功能的潛力。軟體功能確實表示組織的下列任何一項：

- 成功的專案在正確時間的正確決策
- 在商業中成功及存活
- 冒險及控制專案輸出
- 取得產品的正確品質
- 整合產品中最先進的軟體技術

軟體功能是之前這些陳述更準確的混合體，但有些概念必須被視為軟體功能的一部分：

- 在專案層次：
 - 生命週期問題：**專案生命週期不限於產品的實作和維護。有一個管理階層視景，處理專案的業務、財務和策略。
 - 流程選取和調整：**現成的和/或反映過去經驗的標準流程無法達到各類專案的目標。為特定專案定義足夠的開發流程才能產生效率。
 - 風險管理：**任何專案都會面臨風險。能幹的組織可預測風險，並決定重新配置專案，使發生風險的影響降至最低。
 - 測量：**在軟體組織中，長期進步的一個關鍵是收集歷程資料來分析軟體品質和生產力。收集每一個專案的部分歷程資料大有幫助。關於人力、排程、實作的程式大小和功能等資料，以及問題計數，形成規劃未來專案及改進可預測性的一個穩固基礎。預測效能是成熟度的一個徵兆。
- 在組織層次 - 視為專案的集合：
 - 流程改進：**功能意指組織能夠從過去經驗中學習，尤其是從他人的錯誤中學習，並且可以將這些所獲得的經驗轉換成流程進展。唯有已定義測量來量化改進時，改進反覆週期才會完整。
 - 資源和技能：**組織的能力與應用流程及開發產品的人員的能力緊密耦合。

RUP 符合專案層次的 CMM 需求

CMM 說明達到特定成熟度層次的組織中的專案狀態。

等級 2 組織內的專案已安裝基本軟體管理控制。實際專案承諾以在先前專案上觀測到的結果和現行專案的需求為依據。軟體專案經理追蹤軟體成本、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，其完整性受到管制。有定義軟體專案標準，且組織確保能夠確實遵守這些標準。軟體專案搭配轉包商來建立穩固的客戶供應商關係。

在已定義等級（等級 3），有記載在組織內開發及維護軟體的標準流程。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。

生命週期問題

在 CMM，生命週期問題涵蓋於多個主要流程區域 (KPA)。第一次在組織提出 CMM 概念時，大家通常會誤解「軟體流程」這個詞彙，而把它比作「軟體生命週期」。專案生命週期包含商業和金融問題及開發問題。等級 2 主要流程區域和一些等級 3 主要流程區域與專案生命週期相關。它們是：

- 需求管理目標 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線。
- 軟體專案規劃目標 1：記錄軟體評估值，來規劃和追蹤軟體專案。
- 軟體專案規劃目標 2：計劃與記載軟體專案活動和承諾。
- 軟體專案追蹤目標 1：依據軟體計劃來追蹤實際結果和效能。
- 軟體專案追蹤目標 2：受影響的群組和個人同意軟體承諾的變更。
- 軟體配置管理目標 1：識別、控制及提供已選取的軟體工作產品。
- 軟體配置管理目標 2：控制已識別之軟體工作產品的變更。
- 軟體配置管理目標 3：將軟體基準的狀態和內容通知受影響的群組和個人。
- 軟體產品工程目標 1：定義、整合和一貫執行軟體工程作業來產生軟體。
- 軟體產品工程目標 2：軟體產品彼此一致。

換句話說，CMM 需要：

1. 有隱含地定義一條基準線，來考量要開發的應用程式的商業環境，及設定專案需求的優先順序，並做出與專案的功能和非功能內容相關的所有必要決策。
2. 有建立專案計劃，且需要設定實際而可靠的活動、角色和責任、里程碑和構件的清單。專案計劃需考慮風險。
3. 專案計劃受到監督，萬一特性內容或專案環境/組織有很大的偏差時，專案小組要作出反應。
4. 專案的任何變更會加以識別及分析。做出進一步的決策並在基準線和專案計劃中加以報告。
5. 選取和套用方法和工具，以確保專案的最佳效能。

Rational Unified Process 在初始階段的主要作業中實現這些需求，如下所示：

- 取得專案願景並比對之前清單的第一個項目來開發商業模型。
- 組織和規劃專案
- 評估可能的風險。（這個項目與之前清單中的第二個項目有關）。
- 部署「統一變更管理」原則，它類似之前清單中的第三和第四個項目。

初始階段的優先目標是要在專案的生命週期目標上達到所有關係人意見一致。初始階段的重要性主要是針對新的開發工作，在專案進行之前，必須先處理該階段重大的商業和需求風險。對於聚焦於現有系統的增強功能的專案，其初始階段較短，但仍然成為焦點，以確保專案值得進行而且行得通。在初始階段結束時，會檢查專案的生命週期目標，並做出要繼續專案或取消專案的決策。

要檢視的重要構件如下：

- 專案願景
- 企業案例
- 風險清單（請參閱風險管理這一節）
- 軟體開發計劃
- 反覆計劃
- 開發案例（請參閱「流程選取和調整」這一節）

這些構件滿足本節一開始所列出的 CMM 主要流程區域的數個目標。

取得專案願景及開發商業模型

商業模型從工作者的內部觀點來定義商業使用案例。此模型定義在企業中工作的人及其處理和使用的事物—商業的類別和物件—應如何彼此相關，包括靜態和動態，以產生預期的結果。此模型也強調在商業區域執行的角色，及其作用中的責任。模型類別的目標需要能夠執行所有商業使用案例。

Rational Unified Process 根據商業模型來識別一個叫作「開發願景」的作業，它有下列目的：

- 對需要解決的問題達到共識
- 識別系統的關係人。
- 定義系統的界限。
- 說明系統的主要特性。

組織和規劃專案

就像軟體流程受到專案特性的影響一樣，專案組織也是如此。這裡所呈現以及「圖 1」所說明的預設結構，必須加以調整，才能反映出以下列出的那些因素的影響：

- 商業環境定義
- 軟體開發人力大小
- 創新程度
- 應用類型
- 現行開發流程
- 組織因素
- 技術和管理複雜性

這些因素在 RUP 中當作流程判別式來考量，它們影響專案結構的選擇。

專案結構主要是透過下列兩者定義：

1. 每一個反覆的長度
2. 反覆的數目

反覆是比較完整的迷你專案，歷經所有主要工作流程，在大部分情況下會產生可執行但尚未完成的系統，我們稱之為版本。

判斷反覆的數目時，可能有許多變化，視風險、大小和複雜性而定。

如果產品是專門爲了某個全新的領域而提供的，您可能需要在初始階段新增一些反覆來合併一些概念，展現客戶或一般使用者的橫斷面的不同原型，或對某項要求做出完整的回應，來當作提議。

如果產品又大又複雜，而也已開發一段很長的時間，您需要規劃在建構階段擁有三個以上的反覆。

在專案生命期間，組織將發展爲支援專案計劃所擷取的工作分析結構。這顯示於「圖 1」[7]。

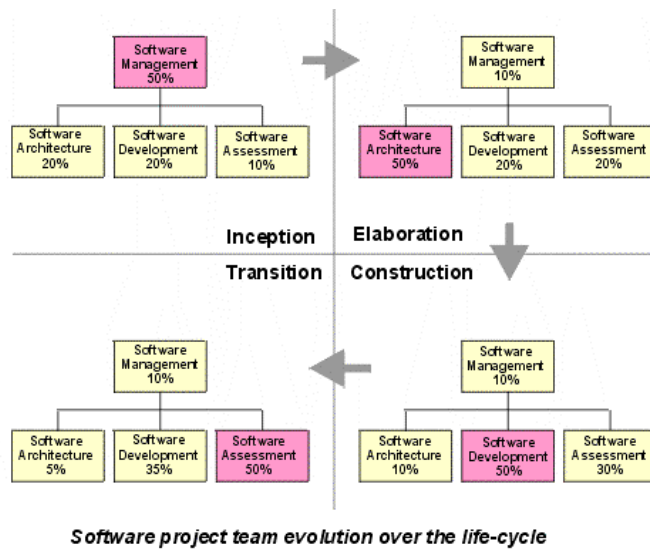


圖 1：軟體專案小組發展

此發展在每一個階段強調不同的活動集：

- 初始階段團隊：組織聚焦於規劃，加上其他團隊充分的支援，以確保計劃能夠代表所有看法一致。
- 詳述團隊：以架構作爲焦點的組織，其專案驅動力在於軟體架構團隊，必要時，該團隊會得到軟體開發和軟體評量團隊的支援，以達到穩定的架構基準線。
- 建構團隊：一個和諧的組織，其大部分活動落在軟體開發和軟體評量團隊身上。
- 轉換團隊：以客戶爲焦點的組織，其使用意見驅動部署活動。

在此發展期間，團隊之間的調動可確保知識和能力保留在專案內。

部署統一變更管理原則

「統一變更管理 (UCM)」是 Rational Software 在軟體系統開發中管理變更（從需求到版本）的方法。UCM 跨越開發生命週期，定義如何管理變更的需求、設計模型、文件、元件、測試案例和程式碼。

UCM 模型的主要觀點在於它統一用來規劃及追蹤專案進度和經歷變更的構件的活動。UCM 模型由流程和工具一起實現。Rational ClearCase® 和 Rational ClearQuest® 是 UCM 的基礎技術。ClearCase 管理軟體專案產生的所有構件，包括

系統構件和專案管理構件。ClearQuest 管理專案的作業、問題和增強功能的要求（一般稱為活動），並提供追蹤專案進度所需的繪圖和報告工具。

流程選取和調整

成熟度等級 3 包括三個以整體組織為目標的 KPA，和四個以專案組織、管理和工程為目標的 KPA。管理和工程活動的軟體流程會加以記錄、標準化及整合成一個全組織的軟體流程。所有專案需使用記錄的和核准的組織流程版本來開發及維護軟體。組織流程焦點 (OPF)、組織流程定義 (OPD) 和培訓計劃 (TP) KPA 可確保組織能夠在等級 2 識別專案的最佳作法，並將它們記錄為組織的標準。它們也把焦點放在技能管理上，主要是依據專案識別的需求、從過去專案中所獲得的經驗，和組織任務及未來願景。然而，如果沒有準則為特定專案調整標準流程，以達到成功、管理風險和改進效能，則成熟度不能滿足等級 3 的需求。

因此，對於大部分主評審員和 CMM 社群，等級 3 以整合軟體管理 (ISM) KPA 為特徵，其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。此定義流程是標準流程最適合的自訂作業，幫助符合客戶需求和限制、市場需求和商業策略。

ISM 由 CMM 模型中的兩個目標來定義。第一個模型包含調整，第二個流程需要專案管理。ISM 的活動 10：專案的軟體風險根據一個已記錄的程序加以識別、評估、記錄和管理，這是等級 3 組織的第二個關鍵指標。等級 2 需要風險管理，但等級 3 組織的成就在於更能預期風險，及示範企業家決策系統。下面的「風險管理」一節會詳述這些層面。

在 RUP，開發案例和環境規範的主要概念達到這些需求。開發案例是 Rational Unified Process 產品的靜態配置；換句話說，它是軟體工程的商業流程，針對特定專案、產品和組織而調整。「開發案例」的焦點在於執行內容和執行方法，並提供應遵守的流程概觀，讓專案的每一個人都可以瞭解。

Rational Unified Process 列出流程的組成要素，這些要素在給定的開發案例中可能要修改、自訂、新增或廢止。

- **規範**
軟體專案很少完全略過一個主要規範—例如分析和設計、實作等等
- **構件**
專案更可能因為它們必須產生、更新和交付的構件而不同。
- **作業**
作業不同，至少有兩個原因：使用構件作為輸入的作業，和產生或更新使用構件作為輸出的作業，因為這些構件修改而受到影響。尤其，如果有些構件或構件中的一些資訊元素不再需要，則可以廢止或大幅修改相對應的步驟。也可以修改作業來引進與特定應用程式領域相關的特定技術、方法和工具，或一些開發專門技術，例如設計步驟、程式語言、自動產生程式碼工具、測量技術等等。

流程工程師負責配置流程、決定開發流程的外貌、在開發組織（團隊、專案或公司）中「安裝」開發案例，及指導開發人員如何使用它。

當流程工程師設定開發案例之後，專案經理就可以將它實例化，並針對給定的專案執行它。這通常叫作流程制定。

當流程展開時，可從流程本身獲取經驗，流程工程師會以這些經驗作為改進流程的意見。

風險管理

風險管理在 CMM 中涵蓋很廣，主要是在等級 2，等級 3 比較深入。會列出特定的風險活動，以滿足「主要流程區域軟體專案規劃」、「軟體專案追蹤和監督」及「整合軟體管理」目標。CMM 需求遵循許多組織奉行的作法：在

等級 2，專案一般會定期識別及評估風險。但主動性很低。在等級 2，風險管理尚未全面涵蓋/獲得瞭解，而且專案延續方面和早期識別風險方面也有問題。

相反地，等級 3 的組織會識別、評估及減輕風險，且團隊證明更好的預感。在大部分先進的團隊中，會使用定量測量來做決策。

從我們在 CMM 式的評量和「軟體流程改進」上的豐富經驗，可得知軟體單位在管理風險時面臨的難題。儘管一切努力都投入專案控制和測量上，還是發現了一個弱點。

使用 RUP 和反覆式方法對於成功管理專案風險很有幫助。整個 Rational Unified Process 是由「風險管理」驅動。

風險驅動反覆計劃；反覆是以處理特定風險（嘗試限制或減少風險）為中心來規劃。會定期檢視「風險清單」，以評估風險減輕策略的有效性，藉此驅動專案計劃和後續反覆計劃的修訂。

管理風險的關鍵在於不要等到風險成形（變成問題或失敗）才決定要怎麼對付它。就像橫越大陸的飛行航道若有一點點變更，對飛機著陸的位置就會有很大的影響，比真相大白之後再來大掃除，提早管理風險通常可付出較少的成本和痛苦代價。

這裡識別風險的四個類別：

1. 資源風險
 - 組織
 - 資金
 - 人員
 - 時間
2. 商業風險
3. 技術風險
 - 範圍風險
 - 技術風險
 - 外部相依關係風險
4. 排程風險

「風險管理計劃」和「風險清單」是 RUP 的「專案管理」規範所識別的兩個構件。它們建議您識別及評估風險。

測量

測量是 CMM 的一般特性。從等級 2 到等級 5，每一個主要流程區域都有一般特性，指出作法何時制度化。測量很重要，在 SW-CMM 的新版本中——也就是內嵌到 CMMI 架構的版本：整合性 CMM 系統工程 + 軟體工程——每一個主要流程區域有一個以測量為基礎的目標。

CMM 的關鍵是要測量流程，以決定它們在等級 2 的恰當性和在等級 3 的有效性。從報告到分析和按照測量行事，這樣的轉變通常很難，但也是專案更成熟的明白徵兆。早期以量化方式管理專案的小成就讓人接受和瞭解測量的好處。

除了這些一般需求之外，有些 KPA 有特定的測量需求，例如「軟體專案規劃」、「軟體專案追蹤和監督」及「整合軟體管理」。這些與專案資料有關，它們啓用專案評估和專案控制。

RUP 提供使用測量值的準則。測量計劃和測量構件必須在初始階段產生。

專案測量構件儲存專案的測量資料。當測量一產生或變成可用時，就會保持最新測量資料。它也包含衍生的測量值（從初始資料計算而來），也應該儲存關於如何取得衍生測量值的資訊，例如程序和演算法。專案狀態的報告一例如，朝向目標的進度（功能、品質等等）、支出和其他資源取用一是利用專案測量來產生。使用 Rational Project Dashboard 方法可以頻繁甚至明顯而連續地顯示專案狀態，例如，自動化軟體資料收集代理程式提供專案狀態的即時顯示。

Rational 文件編輯一個起始的測量集，它很容易入門。專案某些層面的測量包括：

- 大小和複雜性的進度
- 需求或實作、大小或複雜性的變更率的穩定性
- 變更範圍的模組性
- 錯誤數目和類型的品質
- 錯誤頻率的成熟度
- 專案支出與計劃支出的資源

RUP 符合組織層次的 CMM 需求

等級 2 組織的軟體流程能力可以「有規範」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。

在等級 3（已定義等級），有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，且這些流程整合成一致標準。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）在等級 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。

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。

流程改進

此功能意指組織能夠從過去經驗中學習，尤其是從他人的錯誤中學習，並且可以將這些所獲得的經驗轉換成流程進展。唯有已定義測量來量化改進時，改進反覆週期才會完整。

當組織的成熟度改進時，其標準流程也跟著變更。從等級 2 進入等級 3 意味著專案內所有好的作法都已經制度化，而且有評估流程來幫助識別專案的最佳作法，這些將記錄在「組織標準軟體流程 (OSSP)」中。這些是「組織流程焦點 (OPF)」主要流程區域的需求。然後會依據從組織各專案獲得的經驗來修正 OSSP。因為專案使用相同標準，所以所獲得的經驗庫容易累積，且 OSSP 本身可以從不同經驗中受益。

RUP 環境規範闡述類似的方法。「在組織中實作流程」的概念說明您在組織層次執行什麼動作，以便在開發組織中實作流程和工具。

可以使用 RUP 的四個階段來說明如何在軟體開發組織中實作新流程：初始階段、詳述、建構和轉換。

在組織中實作流程和工具的步驟

CMM 的 OSSP 是組織的新流程，其定義可遵循 RUP 的四個階段。

是否要以必要的調整來開發全組織的開發環境，讓每一個軟體開發專案都可以使用，這個決定是關鍵所在，但必須有某種成熟度層次。

如果您決定要開發全組織環境，您必須起始一個專案來開發組織的開發環境。如果您決定要起始這樣的專案，必須明確表示此專案小組將與軟體開發專案小組密切合作。RUP 也建議您將它視為特定專案來處理。如此又履行了 CMM 需求。

流程實作專案分成幾個階段，在每一個階段會執行所有四個步驟，直到專案準備就緒，且整個組織已部署及成功地使用了流程和工具。

流程實作專案可分成幾個階段

四個階段處理：

- 階段 1：向贊助商推銷流程實作專案。
- 階段 2：處理主要風險。
- 階段 3：完成一切一開發案例的範本、準則、範例已備妥，且培訓系列課程也就緒。
- 階段 4：將它部署到整個組織。

這些階段可分別命名為初始階段、詳述、建構和轉換，因為它們是針對使用 Rational Unified Process 的軟體開發專案。

每次標準流程爲了提高效能而必須發展時，就會依據在特定專案所獲得的經驗或根據特定技術來定義相同專案階段。

RUP 也定義「管理組織變更」的概念，它是「軟體流程改進」的整體背景。成功實作流程變更的建議如下：

- 在組織的不同層次識別變更代理程式。
- 請以一些小小的、合理的、可測量的步驟來規劃變更。
- 使用適合組織層次的基準語言來溝通變更。

這些建議類似引起它們的 IDEAL 方法所提出的那些建議。

最後，RUP 定義一個特定參與者，叫作「顧問」。顧問負責教授及指導專案小組瞭解需求及需求的時機。指導的一般方式如下：

- 研討會領導者
有些活動最好以群組方式執行；例如，在使用案例建模期間尋找參與者和使用案例。在這類活動中，有流程專家作為建模領導者很有用。
- 流程專家
流程專家是專案的駐點支援人員。流程專家的工作是要幫助開發人員好好運用流程和模型。
- 專案經理支援
流程專家可幫助專案經理規劃及帶領專案。有時候專案經理對於有問題的流程不太有經驗，或完全沒有經驗。
- 審查人員有成本效益的傳輸知識方式是請一位流程專家審查每一個階段的結果。流程審查人員在審查一個專案所執行的任何流程調整時也很有幫助。

這個顧問扮演的角色是否與 CMM 中的 SEPG (Software Engineering Process Group) 不同呢？不，它是同類型的角色。

資源和技能

在組織層次，CMM 識別「培訓計劃」主要流程區域，它若命名為「技能管理」更好，因為那才是真正的考量。成熟的組織必須能夠識別及規劃中長期的技能需求，並設法讓它們能夠適時發揮。

「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，軟體專案應該識別必要的技能，並提供必要的培訓。在進行專案人員配置時，RUP 中的管理規範會考量這些問題。

培訓計劃主要流程區域有三個目標：

- 目標 1：規劃培訓活動。
- 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。

RUP 的幾個方面滿足這些需求。精確識別角色，並定義其能力。角色通常由個人或在團隊工作的一小組人來扮演。專案小組成員通常扮演許多不同的角色；就像一個人可以戴好幾頂帽子一樣，一個人也可以執行許多不同的角色。

角色不是個人；相反地，它們說明個人在企業中如何表現以及這些個人應該承擔的責任。

雖然大部分角色是由組織內的個人所扮演，但開發組織以外的人也扮演重要角色；例如專案或開發中產品的關係人的角色。

每一種角色是由成員應展現的技能和知識周密地定義。根據這些定義，組織可以得知要有效率地扮演這個角色需要哪些缺少的技能和培訓特性。

RUP 準則可形成培訓用的最佳規則集。課程開發人員是所列出之角色類別的其中之一。

RUP 唯一未全面涵蓋但卻是 CMM 所必要的，那就是識別、規劃和提供培訓來開發未來新技能。假設組織的策略是要從現在起一年內以電子商務作為其活動的中心；這個目標必須進一步分解，而且可能影響所執行的開發類型和所使用的環境類型。團隊必須為這項變更作好準備，且必須從外部定義、開發或取得必要的培訓。等級 3 的培訓主要流程區域需要仔細處理、追蹤和記錄這些問題。

- 本白皮書的第三節提供一些成熟度設定檔與 RUP 特性的詳細對映。本白皮書原本可以將 CMM 參考手冊的每一個目標與 RUP 的對應準則有系統地並列，但我們故意不這麼做，相反地，我們選擇把焦點放在高成熟度的主要徵兆上。
- 軟體品質在 RUP 內處理的方式，一方面是透過測試活動，另一方面則透過「開發品質保證計劃」作業。然而，我們在示範中強調我們對於高成熟度真正含意的經驗和觀察：等級 3 組織與等級 1 和較低的等級 2 組織相比，差別在哪裡。
- 下一節是關於這些合併技術的早期採用者的一些意見。

實用主義：RUP 如何入門

我們最常被問到的問題是：以前有人這麼做嗎？

每當向經理和從事此行業的人提出上述概念時，總是被問到這個問題。變得更成熟、更加流程導向、能應付風險和成功，任何軟體企業都會被此吸引。然而，在移動的環境定義中，真的會有快速變更的需求和需要應用未受過訓練的技術之團隊嗎？

Rational 客戶報導包括 Computing Devices International 的經驗談：Computing Devices 是一家領先的防禦電子和資訊公司。資訊繼續成為幾乎任何產業的主要競爭優勢。對於 Computing Devices International 而言，資訊不只是資訊—其任務是要提供端對端解決方案，隨時隨地隨需應變傳遞重要資訊。

它們需要確定高品質解決方案能夠一致且如期地交付。其經驗的主要益處包括下列各項：

- 開發流程透過反覆式設計方法加速進行
- 開發時間從 3 年縮短為 19 個月
- 確保可靠而準時交付的能力
- 客戶滿意度提高
- 開發成本減少 33%

Computing Devices 為客戶開發的更準確系統，高度仰賴其設計符合嚴格需求的軟體。Computing Devices 率先將注意力轉向撰寫安全重要軟體的流程。

組織內的不同業務單位使用各種不同的工具，這些工具是基於結構式設計或「瀑布式」方法。因此，開發流程緩慢且缺乏一致的方法。這導致錯過截止日期、成本更高，且軟體品質不一定符合其高標準。Computing Devices 瞭解他們需要採取行動—而且要快。

Skandia-IT 是 Rational 資料夾的另一個範例。在該組織中，有 9 個大型保險系統在 12 個月內準時或提前交付。

他們以最好的開發流程吸引開發人員和顧問們，加速新人才的招募，運用使用案例、模型系統和直接開發誘導需求來確保客戶滿意，並快速開發彈性的 3-層架構，將舊版系統包裝起來作為元件使用。這些是 Skandia-IT 克服的一些難題。

Skandia-IT 的經理解釋：我們改變整個流程有一些原因。最重要的是我們重組整個企業。我們改變開發流程的同時，也從一個產品導向的組織變成一個客戶導向的組織。不論是透過電話或全球資訊網 (WWW)，我們的新系統讓客戶有自己處理許多事情的能力。我們在 Skandia 的員工可以投入更多時間來成為保險專員，而不必學習如何使用該系統。節省下的時間轉而投資在客戶關係的維持上。

Skandia-IT 大量投資在新流程的安裝上。這是一項成功的投資。更好的品質、可預測的交付時間和副作用，現在要招募有經驗的專案領導者和開發人員比以前更加容易，這個事實就是它的副作用。

還有其他公司的範例，其焦點放在流程上並開始使用 RUP 來加速部署。Q-Labs 與 Lysis 合作，Lysis 是一家成長快速的公司，專門開發網際網路廣播用的軟體。在一個新運作的公司環境中，把焦點放在流程成熟度的進階層次的流程技術和方法並不合適。這種組織主要是步調快、反應快和創新，而出版產品的速度需求與最大流程控制權需求之間的對比引發這樣的問題：流程的主題可以容納觀點的差異到何種程度？

這家公司的特徵是：

- 他們是一家快速成長的軟體公司。
- 他們發展速度更快（不到 6 個月）。
- 他們使用領先的物件技術來交付更好的產品。
- 他們可適應機動的市場。

這個環境與其他軟體業並無太大不同，只不過時間表完全不同，可預測性成爲生死存亡的關鍵，團隊工作必須快速而徹底地改變。學習時間幾乎不存在，且最好是從現有的、公認的作法開始。Lysis 就是這麼做的，他們同時採用 CMM 來引導其成長和使用 RUP 作爲其安裝流程的背景。

定義軟體作法的需求，無庸置疑。它提供瞭解、溝通、執行和管理。它也形成與其他內部和外部組織的關係公式化的基礎。以 Lysis 爲例，與行銷和業務部以及與可能的軟體轉包商之間的關係是它的目標。已定義的流程必須保持彈性：人員、基礎架構、發行時間表和採用的技術，都有可能快速變化。開發流程需要容納它們。需要有快速即時的回應。建議採用具有簡短反覆週期、風險導向出入口和建構式管理的反覆式流程。在產品系列各元件之間重複流程的能力是必要的。流程定義有助於在公司成長時幫助通知加入公司的新進人員。

在 CMM 中，流程定義（成熟度等級 3）跟在流程可重複性（成熟度等級 2）之後，但在這樣的公司裡，從 RUP 之類的標準流程開始，有助於提升和建立可重複性，並藉此帶來重大利益。

Lysis 採取的方法是先從依據 CMM 需求來評估開發作法開始。這樣可導致對程式設計、物件技術和工具環境最感興趣的團隊正確買入流程導向。根據觀察，定義流程的行動計劃是以強烈的時間限制來建立。由最高管理階層設定來支援 SPI 的商業目標，包括：

- 縮短開發流程，以便成爲目標市場的第一名
- 加速從專案模式轉換到產品模式
- 管理公司成長
- 滿足大眾
- 保持新運作公司的反應力
- 改進可預測性和可見性（對於 IPO，乃必要的季度結果）
- 產品品質的定量改進
- 管理成本和潛在的利潤
- 客戶滿意度最佳化
- 識別和管理與夥伴的介面（針對工作的再轉包部分）

SPI 計劃本身是使用 RUP 架構建立的專案，其中：

- 初始階段 = 評定
- 詳述 = 專案規劃 + Web 型儲存庫定義和原型化
- 建構 = 定義流程 + 實作和工具
- 轉換 = 試驗專案 + 部署

有定義工作套件來處理 CMM 主要流程區域。每一個都有目標、要執行的活動、可交付項和檢視流程。例如，軟體規劃的目標是：

動作的目標：

- 記錄軟體規劃的現行良好作法
- 定義專案的一般組織設定
- 定義角色和規劃的責任
- 定義專案層次的確定流程
- 爲一組標準專案流程定義一般專案生命週期和調整準則
- 定義標準工作分析結構 (WBS)
- 列出可交付項的標準清單

- 定義開發計劃的範本和準則
- 撰寫專案規劃流程
- 定義規劃所需的技能和培訓

選擇以 RUP 開始可加速一些動作，並可確保組織從健全的經驗中受益。SPI 創新技術尚未完成，但已觀察到一些可見的成功因素：新產品的開發有更好的結構；有計劃而且經常重看計劃。採取更正動作，軟體團隊以外的可見性也大幅改進。

Rational Software 報告關於 E-corporation 的類似案例。E-corporation 是一個 European Interactive Architect，它引導全世界的公司探索數位商務新領域。E-corporation 結合策略、藝術和軟體技巧，來構想、建立和實作網際網路和電子商業活動。

部署 RUP 的 E-corporation 的主要優點是：

- 團隊成員之間有更好的溝通，使工作效率提高。
- 更精確地滿足客戶的期望。
- 有提供符合高品質和高速需求的架構。
- 反覆式開發的風險降低，並使用一個以專家知識和業界認可的最佳作法為基礎的公認流程。

E-corporation 對所有專案使用一個共同流程所得到的最大好處，就是有效地重複使用元件、經驗和活動。當組織從不同專案獲得經驗後，Rational Unified Process 可以不斷做細部調整。因為所有活動都是遵循一致的結構和語言來記錄，所以，很多經驗可以從先前的專案中獲得並傳給新的專案。

E-corporation 打算在所有地點的所有專案中推出 Rational Unified Process，進一步發揮其團隊的一致性。Rational Unified Process 賦予其開發團隊應用公認原則的能力，使所有工作最佳化、簡化溝通，並保證符合所有需求的高品質解決方案。

這些範例說明不同組織如何結合 Rational Unified Process 和 CMM 準則來發現其好處。我們目前所合作的其他組織，把 RUP 視為通用結構，並將它適合的組織流程資產移入其中。

結論

我們所描述有關 RUP 和 CMM 的對映，同樣適用於 ISO 15504。這兩項參照的基礎概念是類似的，而我們所標示來說明成熟度設定檔的主要因素支援 ISO 15504 的功能等級。

本白皮書已闡明 RUP 概念符合高度成熟組織的目標的程度。因此，採用 RUP 是成功達到 CMM 需求的一個方式。

Telcordia Technologies 最近發行的一篇文章，對企業的高成熟度提供一些定量結果。Telcordia 最近被評定為成熟度等級 5，它報告如下：

- 自從 SPI 創新技術開始以來，現場錯誤密度的減少已達 94%。
- 從 1995 年開始，有超過 98% 的軟體版本準時交付。
- 測試程式碼行的成本減少了 64%。

Telcordia 的品質之旅花了 6 年才達到其成熟度等級。

對於在電子商業和電子商務領域的軟體密集公司，花五六年的工夫將流程最佳化，即時交付恐怕等不及了。因此，要達到維持競爭力所需的成熟度，唯一可能的情况是：

- 瞭解軟體流程的價值
- 採用現有的架構，例如 RUP
- 調整該架構，使適合市場環境的特性
- 估量它對企業的影響
- 學習和豐富自訂的流程模型

因為 RUP 模型是通用的，所以注意不要迷失方向。因此，Rational 識別「RUP 要點」，包括在採用此模型時要成為焦點的那些層面 [11, 12]。可使用預設 RUP，但不必在第一次實作每一項。重要特性如下：

- 企業案例
- 排程
- 願景文件
- 風險
- 架構
- 變更要求及如何處理問題
- 測試
- 軟體產品本身
- 使用者支援文件
- 專案評量

執行 RUP 本身並不是目標；它幫助解決群組和團隊的痛處，並幫助達到有競爭力的目標。

作者的聯絡資訊

Annie Kuntzmann-Combelles，執行副總
Q-Labs France
28 Villa Baudran 94742 Arcueil cedex
電話 +33 (0)1 49 08 58 00
akc@objectif.fr

Philippe Kruchten, Rational Fellow
Rational Software Canada
pbk@rational.com

參照

- [1] K. Pulford, A. Kuntzmann-Combelles, and S. Shirlaw, 1995. *A Quantitative Approach to Software Management—The AMI Handbook*. Addison Wesley Longman.
- [2] Barry W. Boehm, 1996, 鰐anchoring the Software Process, 鰐EEE Software, July 1996, pp.73-82.
- [3] Philippe Kruchten, 1996. 鰐Rational Development Process, 鰐rossTalk, 9 (7), July 1996, p.11-16.
- [4] Robert McFeeley, 1996. *IDEAL: A User's Guide for Software Process Improvement*. Software Engineering Institute, Pittsburgh, PA, CMU/SEI-96-HB-001.
- [5] Steve McConnell, 1997. *Software Project Survival Guide*. Redmond, WA: Microsoft Press.
- [6] Mark Paulk, et al. 1993. *Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute, Pittsburgh, PA SEI-93-TR-024.
- [7] Walker Royce, 1998. *Software Project Management: A Unified Framework*. Addison Wesley Longman.
- [8] Jas Madhur et al, 1998. *Reaching CMM Levels 2 and 3 with the Rational Unified Process* white paper, Rational Software.
- [9] Alec Dorling et al., 1999. *SPICE, the Theory & Practice of Software Process Improvement*, IEEE Computer Society.
- [10] Philippe Kruchten, 2000. *The Rational Unified Process—An Introduction, 2nd ed.*, Addison Wesley Longman.
- [11] *Rational Unified Process, version 2000.02.10*, Rational Software Corporation
- [12] Leslee Probasco, 2000. 鰐he Ten Essentials of RUP, *The Rational Edge*, December 2000, <http://www.therationaledge.com>.



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

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

Rational、Rational 標誌、Rational Unified Process 和 Rational Rose 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。