

# RUP 的十大要項—有效開發 流程的要項

Leslee Probasco

Rational Software 白皮書

---

TP177, 9/00

## 目錄

摘要.....	1
請求幫忙 .....	1
詳細資料太多 .....	1
以要項為焦點.....	1
開始於 簡短清單.....	2
您的清單上應該有些什麼呢？ 這要看情況而定。 .....	2
優先建置架構.....	2
RUP 的十大要項 .....	3
願景 — 開發願景 .....	4
計劃 — 計劃的管理 .....	4
風險 — 識別和減輕風險 .....	4
問題 — 指派和追蹤問題 .....	5
企業案例 — 檢查企業案例 .....	5
架構 — 設計元件架構 .....	5
產品 — 漸進式建置和測試產品 .....	5
評估 — 定期評定結果 .....	5
變更要求 — 管理和控制變更 .....	6
使用者支援 — 為使用者提供協助 .....	6
需求是什麼？ .....	6
測試方面如何？ .....	6
摘要：套用十大要項 .....	6
對於小型專案 .....	7
對於成長的專案 .....	7
對於成熟的專案小組 .....	7

## 摘要

---

若要有效套用軟體開發流程，例如 Rational Unified Process<sup>1</sup>（即所謂的 RUP），一定要先瞭解它的主要目標，為何每一個都很重要，及它們如何一起運作來幫助開發團隊產生高品質產品，以符合關係人「真正的」需求。

## 請求幫忙

---

有一天晚上，我們鄰居 Randy 過來請我幫忙：他正準備週末和教會的朋友一同去露營和遠足，但無法決定該使用何種裝備<sup>1</sup> 為遠足打包行李。他已經有一些衣物和設備，但不確定是否應該多買一些。他想要借我的裝備清單？

我的裝備清單？是的。似乎他來拜訪的那一天我剛好在準備徒步旅行和爬山。他知道我在帶領和參與野外旅行很有經驗，而且對於我能夠參考自行編製的裝備和衣物清單，快速而有效率地決定哪些物品該放入空間有限的背包裡，印象深刻。他能不能借我的裝備清單？

當然可以，我歡迎他向我借裝備清單 — 但我認為這對他幫助不大。為什麼呢？難道我不認為他需要的一切都在這份清單上？

事實上，我很確定他需要的一切至少在我的清單上都有代表的物品，但不止是可能而已，他需要用於這次特殊的旅行的真正物品可能根本不在這份清單上。例如，不只他的鞋子大小比我大很多，而且毫無疑問地，他喜歡吃的食物（和數量）和我選擇攜帶的食物不一樣。

## 詳細資料太多

---

事實是，我的戶外裝備清單上就列出幾百個物品，涵蓋許多不同的遠足類型，從徒步旅行和爬山，到滑雪、穿雪鞋走路、攀冰和泛舟 — 也涵蓋旅行天數，從簡單一日遊，到多日遠征都有。而且，如果我是組織的領隊，那麼我攜帶的物品（例如旅行登記名單和棄權書）跟我只是帶領幾個好朋友從事野外活動會有所不同。我的主要考量是他無法看完清單上的一大堆物品，然後猜出他這次相當短的旅遊真正需要的東西。

Randy 絕望而苦惱不已！他一定來不及準備好這次的徒步旅行。他如何知道打包的內容？他從上週就開始打包 — 他在網際網路上搜尋了很久，買了超炫的新馬靴、夾克和衣物 — 他的背包已經裝不下了！

但是都還沒開始打包食物呢！對了，他不是也需要帶水嗎？萬一出事時該怎麼辦？萬一迷路該怎麼辦？或是有人受了傷？

## 以要項為焦點

---

當我將 Randy 放入背包裡的物品加以分類時，一下子就看出他並沒有同等看待野外旅遊所需要的重要物品。

你有「十大要項」嗎？我問他。十大要項？是哪些？

就是這個，Randy，我剛好有你需要的這份清單。我取出一張白紙，在最上面寫著：十大要項。我在上面寫了 10 個物品：

1. 地圖
2. 指南針
3. 太陽眼鏡和防曬乳
4. 備用衣物

---

<sup>1</sup> 裝備是登山客和其他熱衷戶外活動者使用的一個俚語，用來表示任何一種設備、工具、衣物、鞋類或各種運動使用的其他構件。如果您認為 RUP 有很多構件，請參考：[www.mgear.com](http://www.mgear.com)。

5. 備用食物和水
6. 頭燈
7. 急救箱
8. 點火器
9. 火柴
10. 小刀<sup>2</sup>

## 開始於簡短清單

---

就這樣？對！這是這樣！如果您從這 10 個元素開始，並確定有涵蓋每一個區域，則其他東西將依需要而逐漸明白。當然，這些「要項」每一個都可以向上或向下調整，視旅遊性質而定，但是開始於「簡短清單」，然後再依需要擴充它，比開始於長清單，然後再決定什麼不要，容易多了。

當您有了經驗之後，就可以逐步形成自己的「完整清單」，它會隨時間而變得有意義。此時，您再看看我的清單和別人的清單，可幫助您瞭解該如何擴充自己的清單。但是，不會有兩人的清單看起來完全一樣（除非他們作弊！）

我記得幾年前我第一次開始從事登山運動時，不管在準備何種類型的旅遊或打算去幾天，我都是參考這份清單。如果你問任何有經驗的登山者「您有十大要項嗎？」，他們大部分都知道您在問什麼。實際上，不同群組或個人可能有不同的「十大要項」清單，但其實都是一樣的。在每一次的旅遊中，我打包的實際物品可能完全不同，但我還是會說我有「十大要項」。

## 您的清單上應該有些什麼呢？這要看情況而定。

---

那麼，這和 RUP 有什麼相關呢？通常，當我幫助專案團隊將 RUP 中的許多元素分類時，我聽到的問題有：「我要如何將所有這些項目分類，然後決定我的專案需要些什麼？」我需要這個嗎？RUP 不是只針對大型專案嗎？

大部分回答是（我們最愛的其中一個回答）：看情況而定”

我決定自己真正想要的是「RUP 的十大要項」清單，我可以將它提供給其他在搜尋流程準則的朋友—就像我幫我的朋友 Randy 一樣。這份清單可作為決定任何專案要包含的正確元素的合理起點，不管是一日遊（非常小的專案）、要過夜的徒步旅行、滑雪或開小艇（在不同領域的中型專案）或是聖母峰探險（超大型重要專案），都可以套用。其構想是要將焦點放在我所謂的「RUP 十大要項」—或任何真正有效的軟體流程。這又返回到我們「最佳作法」的概念<sup>3</sup> 在軟體開發中。

## 優先建置架構

---

我在許多專案中看到的一般問題，是它們通常太著重於特定區域，以致於在確定它們瞭解涉及製作高品質產品的整個流程生命週期的「關鍵」元素之前，即陷入該特定區域之詳細資料的泥沼而動彈不得。

於是，當專案落後時，他們會說：「看吧，我就說過，需求管理只會拖慢我們的速度而已！」您可以將先前陳述中的「需求管理」取代成「使用案例」、「收集專案測量值」、「使用配置管理」、「使用問題追蹤工具」、「召開狀態會議」或任何您熟悉的一些詞彙。

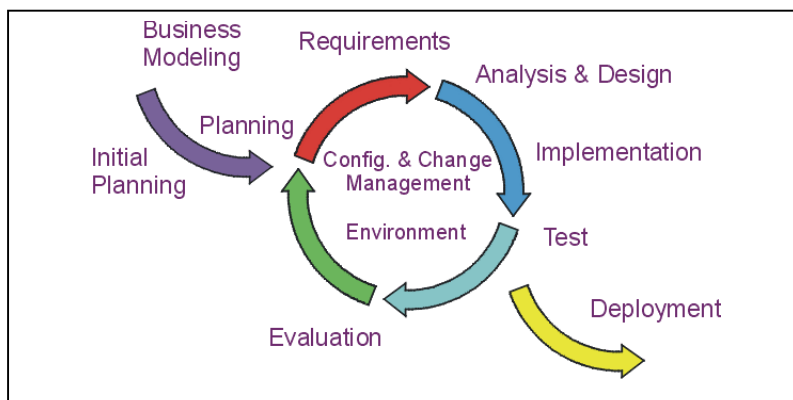
---

<sup>2</sup> 若要知道「十大要項」清單的完整分析，我推薦您看一本好書：*Mountaineering – The Freedom of the Hills*, 6<sup>th</sup> Edition, The Mountaineers, Seattle, WA, 1997. pp. 35-41。

<sup>3</sup> Rational 在過去幾年所偵查到的「最佳作法」包括：(1) 反覆式開發 (2) 管理需求 (3) 使用元件架構 (4) 視覺化模型 (5) 驗證品質和 (6) 控制變更。

我相信，在決定將焦點放在某特定問題區域之前，採取更有系統的和完整的方法，確定流程的關鍵元素已就位（所謂的架構），這樣會更有效率。

一旦品質軟體流程的這個架構就定位之後，我相信專案就可以有效率地聚焦在被認為是問題主因的特定區域上（我承認，通常需要管理在清單中排名第一<sup>4</sup>）。不過請記住，RUP 方法的這項選取是要以識別和優先化專案風險及決定那些已識別的風險的早期減輕策略作為基礎。



## RUP 的十大要項

那麼，什麼是「RUP 的十大要項」？下列清單描述我認為專案必須具備的最基本項目集（如果它們真的有遵循 Rational Unified Process 的要項的話）：

1. 願景
2. 規劃
3. 風險
4. 議題
5. 企業案例
6. 架構
7. 產品
8. 評估
9. 變更要求
10. 使用者支援

現在我們個別查看這些項目，看看它們在 RUP 的位置，並試著瞭解為何每一項會成為我的要項「簡短清單」。

---

<sup>4</sup>由於需求管理是「功能成熟度模型 (CMM)」的其中一個起始「關鍵流程區域」，因此，這並不是那些決定要改進其軟體開發流程的人不太可能發生的情況。

## 願景 — 開發願景

有明確的願景是開發符合關係人真正需求的產品的關鍵。<sup>5</sup>

願景擷取 RUP 中的 *需求規範* 的要項：分析問題、瞭解關係人需求、定義系統及管理變更的需求。

願景為更詳細的技術需求提供高階（有時是契約性的）基礎。願景擷取極高階需求和設計限制，讓讀者瞭解要開發的系統。它提供專案核准流程的輸入，因此與企業案例息息相關。它傳達與專案相關的基本「為什麼和是什麼」，是要驗證所有未來決策的一個量規。

願景的內容應該要回答下列問題：

- 重要詞彙是什麼？（名詞解釋）
- 我們試圖解決的問題是什麼？（問題陳述）
- 誰是關係人？誰是使用者？他們的需求是什麼？
- 產品特性是什麼？
- 功能需求是什麼？（使用案例）
- 非功能需求是什麼？
- 設計限制是什麼？

## 計劃 — 計劃的管理

產品會和產品計劃一樣好。<sup>6</sup>

在 RUP，「軟體開發計劃 (SDP)」收集管理專案需要的所有資訊。它可能含括一些在初始階段期間開發的個別構件，在整個專案中會維護它們。

SDP 是用來規劃專案排程及資源需求，並依據排程追蹤進度。它處理的區域有：專案組織、排程（專案計劃、反覆計劃、資源、工具）、需求管理計劃、配置管理計劃、問題解決計劃、QA 計劃、測試計劃、測試案例、評估計劃和產品驗收計劃。

在簡單的專案中，這些每一個可能只包括一兩個句子。例如，CM 計劃可能這樣陳述：在每一天結束時，專案目錄結構的內容會壓縮，並複製到舊的含標籤的 Zip 磁碟上，並標明版本號碼，再放入中央檔案櫃。

軟體開發計劃本身的格式不像從事產生它的活動和想法那麼重要。因此，我並不在乎它看起來像什麼 – 或您用來建置它的工具是什麼。就像 Dwight D. Eisenhower 所說的：計劃本身不重要，規劃才是重點。

重要項目號碼 2、3、4、5 和 8 擷取 RUP 中的 *專案管理規範* 的要項：構想一個新專案；評估範圍和風險；監督及控制專案；規劃及評估每一個反覆和階段。

## 風險 — 識別和減輕風險

RUP 的重要規則就是要在專案初期識別及反擊最高風險的項目。風險清單打算捕捉專案要成功干冒之已知風險。它依遞減的優先順序識別會導致重大負面結果的事件。

在每一個風險之外，應該有一個計劃來減輕風險。這是規劃專案活動的焦點，而且是組織反覆的基礎。

---

<sup>5</sup> 其目標是要準時在預算內開發高品質產品，來符合關係人的真正需求。— *管理軟體需求*，Dean Leffingwell & Don Widrig, Addison-Wesley Longman, January, 2000。

<sup>6</sup> Johnson Space Center Shuttle Software Group, "They Write the Right Stuff", Charles Fishman, *Fastcompany*, Issue 6, p. 95, December, 1996.

## 問題 — 指派和追蹤問題

連續開放與直接從進行中活動衍生的目標資料的溝通，逐漸形成的產品配置在任何專案中佔有重要地位。在 RUP，這是透過一般狀態評估來完成，它提供處理、溝通和解析管理問題、技術問題和專案風險的機制。除了識別問題之外，應該指派每一個問題一個到期日，和一個負責解決的負責人。而且應該定期追蹤，必要時也要更新。

這些專案即時資料提供管理階層注意的活動訊號。雖然此期間可能不同，但強制功能需要擷取專案歷程及決心移除約束進度的任何路障或瓶頸。

## 企業案例 — 檢查企業案例

企業案例提供必要的資訊，從企業觀點判斷此專案是否值得投入。

企業案例的主要目的是要開發一個實現專案願景的經濟計劃。開發出來的企業案例是用來對專案所提供的投資報酬率 (ROI) 進行精確評量。它提供專案的正當理由及建立它的經濟限制。它提供有關專案價值的資訊給經濟決策者，並用來判斷專案是否應該積極推展。

此說明不應深入探索到問題的細節，而應該針對為何需要本產品提出一個令人信服的理由。然而，它必須簡短一點，好讓所有專案小組成員容易瞭解和記住。在重要里程碑上，重新檢查企業案例，看看預期的報酬和成本估計是否依然正確，及專案是否應該繼續下去。

## 架構 — 設計元件架構

在 Rational Unified Process 中，軟體系統的架構（在某一時時間點上）是系統透過介面互動的系統重要元件的組織或結構，內含由連續小元件和介面組成的元件。主要片段是什麼？它們如何搭配在一起？

RUP 提供方法的、有系統的方式來設計、開發和驗證軟體架構。這是 RUP 中的*分析和設計規範*的「要項」：定義候選的架構、修正架構、分析行為及設計系統元件。

若要陳述及推論軟體架構，您必須先定義架構表示法，即一種描述架構重要層面的方式。在 RUP，此說明是擷取到軟體架構文件中，該文件以多重檢視畫面來呈現架構。

每一個架構視圖處理一些特定的考量，是開發流程中的這些關係人特有的考量：一般使用者、設計師、經理、系統工程師、維護者等等。這是作為架構師與其他專案小組成員之間溝通有關在專案中所做的架構上重要決策的媒介。

## 產品 — 漸進式建置和測試產品

在 RUP，*實作和測試規範*的「要項」是要以漸進方式編碼、建置及測試系統的元件，並在初始階段之後的每一個反覆結束時提供可執行版本。

在詳述階段結束時，可使用架構原型進行評估；必要的話，這也包括使用者介面原型。遍佈於建構階段每一個反覆中的元件，會整合成可執行且通過測試的建置，然後再發展成最後產品。

此要項元素的關鍵就是一組整合的測試活動，它們伴隨產品的建置 - 以及進行中的配置管理和檢視活動。

## 評估 — 定期評定結果

反覆評定擷取反覆的結果、符合評估準則的程度、所獲得的經驗以及要實作的流程變更。

反覆評定是反覆式方法的重要構件。根據專案的範圍和風險及反覆的本質，其範圍涵蓋簡單的示範記錄與完整的正式測試檢視記錄的結果。

這裡的關鍵是要聚焦在流程問題以及產品問題：您落後越快，必須趕上的時間就越多。

## 變更要求 — 管理和控制變更

*配置和變更管理規範*的「要項」是要在專案生命週期內發生變更時管理和控制專案的範圍，同時又能維護其目標，即同時考量所有關係人需求和儘量符合其需求。

當第一個原型出現在使用者面前（通常甚至更早），就會要求變更。（生命的必然性之一！）爲了控制那些變更並有效管理專案的範圍和關係人的期望，任何開發構件的一切變更必須透過「變更要求」提出，並以一致的流程來管理。

「變更要求」是用來記錄和追蹤問題、加強功能要求和其他類型的產品變更要求。「變更要求」的好處是它們提供決策的記錄，並且由於其評量流程，確保可能的變更所造成的影響都能夠被所有專案小組成員所瞭解。「變更要求」對於管理專案的範圍及評定所提出之變更的影響很重要。

## 使用者支援 — 爲使用者提供協助

至少，這應該包括使用手冊（可能是透過線上說明實作），也可以包括安裝手冊和版本注意事項。根據產品的複雜性而定，可能也需要訓練資料，和隨附於任何產品套裝的資料清單。

在 RUP，*部署規範*的「要項」是要包裝及交付產品，並附上任何必要資料，來協助一般使用者學習、使用、操作和維護產品。

## 需求是什麼？

---

有些人看了我的要項清單之後，對於我的選擇強烈表達不同意見。例如，「需求」適合此圖片的什麼位置？這不是比「願景」更好的選擇嗎？需求當然很重要，不是嗎？問的很好。我認爲基本上這就是所謂的第一要項 – 而對於希望清單上有「需求」的那些人而言，您可以將它放在自己清單的這個位置。（記住，每一個專案小組應該提供自己的清單。我的十大要項清單只是要作爲進一步討論的起點。）

不過，根據我的經驗，有時候我會問專案小組（尤其是內部專案）：您的需求是什麼？得到的回應是：實際上我們並沒有需求。一開始，這個回答令我驚訝（我有軍事航空的背景）！他們怎麼可能毫無需求？後來我發現，當他們聽到「需求」這兩個字時，心中所想的是一組外界加諸的「應該」敘述句，說他們「必須」符合，或是拒絕專案 – 而他們真的沒有這些東西！他們可能參與研究和開發，產品的需求是在專案中逐漸形成的。

現在，如果我延續該回應，說：好吧，產品的願景是什麼？那麼，他們的眼睛就會爲之一亮 – 我們輕易地通過上面 A 到 G 的每一個問題 – 此後需求也就跟著來。這最常發生在協同的環境中，而不是契約型專案，後者的需求是探索出來的，而不是勒令強加的。

## 測試方面如何？

---

毫無疑問，你們當中有些比較敏銳的人也注意到，我並沒有將「測試」併入作爲 RUP 的其中一個「要項」。爲什麼呢？我真的相信商業模型對於有效軟體開發流程只是選用性的，但測試則不一樣，我絕不相信測試是選用性的。我把測試視爲一組整合活動，隨著產品的設計和建置而發生 – 和進行中的配置管理及檢視活動非常類似 – 碰巧的是，就和 IEEE 1074 軟體流程標準的方式一樣。

如果您是很敏銳的人，那麼您一定注意到測試其實是隨附於產品的建置中（要項編號 7），而測試結果對於驗證和評估（要項編號 8）非常重要。對於軟體開發而言，Rational Unified Process 的反覆式方法的要項，是要持續建置、測試和評估產品的可執行版本，以便顯露問題，並儘早解決風險和問題。

## 摘要：套用十大要項

---

那麼，探索「RUP 的十大要項」對生活會有什麼改變？



## 對於小型專案

首先，如果有人像我的朋友 Randy 一樣，其實是過來請我提出建議，他想知道如何靠自己或一個沒有經驗的小團隊來使用 RUP 建置一個簡單產品，只是要瞭解流程而已，現在我自己有一份「十大要項」清單可跟他分享，而不是交給他 RUP 的全部功能和 Rational 工具組，令他不知所措！

事實上，沒有專用工具支援也可以完成這十大要項！專案記事本裡有一節專門探討十大要項，它其實是管理專案一個很好的起點。我發現一開始在管理、優先化及追蹤我的 RUP 個人應用程式的變更要求時，便利貼很有價值！

## 對於成長的專案

當然，這些套用十大要項的簡單方法，隨著專案大小和複雜性增加，很快就變成難以管理，因此，想要尋求 Rational 工具的可能客戶的人，會在套用「RUP 的十大要項」的那些人當中找到可能成為客戶的人。

當然，一旦他們決定要深入每一個區域而非只是在表層，他們最後也想要完整的 RUP，以及 Rational 工具的支援（如果您有盡責的話！）然而，我還是鼓勵您以「RUP 的十大要項」搭配我們的「最佳作法」作為引導，而不是以工具來引導。

## 對於成熟的專案小組

對於較成熟的專案小組，「十大要項」可幫助提供快速方法來評定其流程的關鍵元素是否平衡，並識別改善最有利的區域。

在所有專案中，一定要探索萬一任何要項被忽略時會發生什麼情況 – 如果您沒有該要項，什麼會失敗。例如：

- 沒有願景？您可能失去自己該往何處去的線索，並因為繞路而容易心煩意亂。
- 沒有計劃？您無法追蹤進度。
- 沒有風險清單？您現在可能聚焦在錯誤問題上，從現在開始的 5 個月內，可能誤踩在料想不到的地雷上而爆炸。
- 沒有問題清單？不用說明，這些是成功的路障。
- 沒有企業案例？您干冒在專案上損失時間和金錢的風險；它可能被取消或完全失敗。
- 沒有架構？您可能無法處理發生的通訊、同步化和資料存取問題；以及調整比例和效能的問題。
- 沒有產品（原型）？開始撰寫程式碼；只累積書面作業的話，您收穫不大；要在客戶面前拿出實用的東西。
- 沒有評估？不要逃避現實。面對事實很重要。您實際上離截止時間多近？離品質的目標或預算多近？
- 沒有變更要求？如何追蹤這些？
- 沒有使用者支援？當使用者有疑問或想不出要如何使用產品時會發生什麼情況？取得幫忙有多容易？

你看，現在就是在幫忙你！

為何是「十大」要項？沒有特別原因，只不過我不想要超過十（數不了那麼多！）如果只有「九大要項」或更小的數字更好，不過我們似乎無法刪掉任何一個。猜測把它們稱作要項，對不對？至少對我是如此！我鼓勵您使用這個作為專案群組的起點：您的要項是什麼？

如果您跟我一樣，您可能會想出一些回馬槍或字首語來幫助您記住自己的「十大要項」清單。例如，為了要記錄成 4 個音節，我使用 V-PRI-BAPE-CU—當然，因為我把要項數保留為 10 個，所以，我可以偷偷地使用手指計算它們，而不必參考 PalmPilot 上的清單！（現在您知道我的秘密了！）



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。