

管理企業 SBM 安裝的設計變更

目錄	頁碼
摘要	1
SBM — 專為適應調整與整合而設計	1
SBM 程序應用程式開發	2
使用 SBM 的各項職責和人員配置	4
協調循環式長期開發工作	8
重點總結	14

SBM 是以程序為基礎的工作管理產品，讓企業能夠引導、追蹤和監控他們所做的工作。

摘要

本文件說明如何組織 OpenText™ Solutions Business Manager (SBM) 的維護與開發程序，在面對業務需求變動時，能更輕易地針對程序應用程式進行長期維護工作。

SBM — 專為適應調整與整合而設計

SBM 是以程序為基礎的工作管理產品，讓企業能夠引導、追蹤和監控他們所做的工作。雖然 SBM 可以單純只在部門層級使用，但是我們最成功的客戶發現這項產品適用於整個組織，為各式各樣的商業程序結構及可見度。例如，資訊科技組織經常使用 SBM 來管理他們的服務台。整個公司內的使用者使用 OpenText™ Request Center 提交申請單，接著 SBM 程序會引導該申請單在合適的部門按照程序逐一達成要求。新員工入職、福利以及休假申請單會交到人力資源部門；資訊科技接收電子郵件；而基礎架構的問題和支出報帳報表則交給財務部門。SBM 內建報告與稽核功能，負責監督程序的人員不但能瞭解更重大的趨勢，還可以向下鑽取至任何層級，來瞭解任何交易或申請單的詳細資料。

為了滿足客戶對動態組織的需求，我們在設計 SBM 時謹記著幾個基本前提。

變革是根本

變革是企業的根本。市場情況、商業目標和業務要求不斷變動，能不斷適應變化的企業才會成功。業務程序自動化的目標不僅是跟上變革的腳步，還要採取最佳化的方式持續改進。

SBM 是為了讓企業更快適應新出現和不斷變動的使用者要求而設計。我們利用大家熟悉的拖放介面來進程序變更，這個介面設計簡單易用，適合熟悉業務程序的非開發人員使用。

我們的目標是讓人們在參與要管理的例常程序時，能夠隨著新出現的要求以及發現效率不彰時調整和改造這些程序，以利達成現有要求。SBM 允許企業隨著變革情況以及新發現的業務要求去改造各項程序，而不是強迫企業去適應一個僵固的程序，程序變更因此變得更簡單。

系統也參與程序執行

資訊科技環境中有多種不同的工具和系統。程序自動化軟體必須能與目前這些系統搭配運作，而不是在旁邊一同運行而已。參與程序執行的不僅是人，還包括其他記錄系統。程序設計師能夠找到系統之間的交集，並將該交集構建到最高層級的程序設計中。SBM 就是為了實現這樣的整合而建立的。例如，您可以使用 SBM 的 REST Web 服務支援、SOAP (簡易物件存取通訊協定) Web 服務支援、協調化工作流程、AppScripts 甚或是我們的 C++ API 來彌補 SBM 與構成資訊科技環境的其他系統之間的差距。

若您向 OpenText 購買以 SBM 為基礎的解決方案，您的首要目標可能是改動現有的程序應用程式，以利配合您的企業環境之獨有情況。

SBM 程序應用程式開發

開發程序應用程式最好採用循環方式，由瞭解要自動化的程序之業務負責人密切參與。業務負責人瞭解哪些人參與相關程序、他們扮演的角色、必須收集的資料以及監督程序的人員需要檢查哪些報告。業務負責人也知道這些工作的宗旨並指導開發過程，以確保生產力、收益、品質、降低風險和可見度皆能達標。

若您向 OpenText™ 購買以 SBM 為基礎的解決方案，您的首要目標可能是改造目前的程序應用程式，以利配合您的企業環境之獨有情況。這種情況通常只需要簡單改動資料模型 (即變更在程序執行期間收集的資訊) 以及修改工作流程圖來反映程序的狀態和流程之差異。

OpenText 建議您指定一位程序負責人，由他來收集參與執行程序者的回饋意見並確保 SBM 程序根據那些回饋而逐步發展。

開發新的程序應用程式

單純的自訂實作必須預先定義完整的要求，但是 SBM 則不同，您可以從簡單的程序開始引導參與者執行他們的工作，然後隨時間推移再反覆改進。

在開發期間，程序設計師用視覺方式定義：

1. 需收集、呈現以及由程序運用的資料。
2. 反映組織中工作流動情況的程序工作流程。
3. 參與執行程序者的角色和責任。
4. 參與執行程序者所看到的使用者介面。
5. 讓負責監控程序運作情況的人員洞察現況的報告。

當程序應用程式足夠成熟，並得到業務負責人的認可之後，便可以推出給最終使用者使用。

改進已推出的現有程序應用程式

有了 SBM，初始推出後並不是事情的終點，反而是持續改進的起點。OpenText 建議您指定一位程序負責人，由他來收集參與執行程序者的回饋意見並確保 SBM 程序根據那些回饋而逐步發展。

有些客戶利用論壇定期與參與執行程序者的代表會面，確保收集到他們的回饋意見且採取相關行動。在這個階段，我們可以利用回饋意見來改善自訂表單、使用表單動作將使用者介面行為自動化、使用 REST 網格來存取外部資料以及提供嵌入式報告來存取相關資料。出現更多報告需求時，您可以為那類資料的使用者提供更豐富的報告內容。沒有深厚的技術或程式設計技能的人員也可以執行這類變更。熟悉使用 SBM Composer 以及瞭解要用作模型的程序是唯一的先決條件。

例如，藉由使用者回饋意見，您發現使用者被迫捲動到轉換表單的底端才能到達某個必要欄位。所以，雖然導入 SBM 可能達成管理程序的功能性要求，我們卻發現了效率不彰的問題，這是可以解決也應該立即解決的事。只要使用 OpenText™ SBM Composer 將欄位拖曳到表單上方，便可以馬上修正這項疏失。

使用 SBM 的各項職責和人員配置

為了讓 SBM 發揮最大價值，必須由適合的人員進行安裝以利正確使用。為了快速地反覆改善程序，相關人員最重要的特色是熟悉要自動化的業務程序。變更週期太長使得程序無法及時因應業務變革。另一方面，對於整合以及建立和維護 SBM 系統的更技術性的層面，在開始開發之前應先謹慎定義相關要求，然後便可以用較長的開發週期進行導入且不需要與業務保持密切關係。此小節探討導入和管理 SBM 系統相關活動的各項職責。*

安裝與系統組態

一般由系統管理員執行 SBM 相關的安裝、升級和系統組態設定工作。這些責任通常落在負責維護公司的實體和軟體基礎架構的資訊科技人員身上。

SBM 管理

與系統管理員相反，SBM 管理員管理 SBM 程序的例常運作。SBM 管理員使用 SBM Application Administrator 管理 SBM 使用者和群組、權限、通知、專案和管理工作流程覆寫。使用者可以按照本身需求定義多種報告，而管理員通常根據參與執行程序者和觀察員的需求定義各種各樣的報告。管理員必須熟悉在 SBM 中導入的業務程序以及 SBM 用戶群的需求。

SBM 程序設計維護

與 SBM 管理員密切相關的其他職責包括使用 SBM Composer 逐步改進您設計的程序。SBM 管理員或設計師參加解決方案論壇，即可與使用者及程序負責人一起合作，瞭解不斷變動的要求並馬上導入能為參與執行程序者帶來最多好處的變更。因為有一個與程序有密切關係的人員協助您在短時間範圍內執行循環式設計變更，您便能善用 SBM「為變更而打造」的多項功能。

為了讓 SBM 發揮最大價值，必須由適合的人員進行安裝以利正確使用。

*取決於 SBM 系統的大小和複雜性，有些角色可以由數人擔任，或是相反地由一個人擔任一個以上的角色或是擔任一個 SBM 角色並肩負非 SBM 的責任。

SBM 中每個應用程式均有一個主表格，定義一個項目存在於程序中的生命週期裡收集到的資訊。當新要求出現時，程序設計師一般會修改資料模型，就是在主表格中新增欄位以收集更多資訊。

下列小節探討利用 SBM Composer 執行的各類設計工作以及個別需要的技能等級。

工作流程設計

工作流程界定一個項目在參與一項程序時通過整個組織的路徑。以最抽象的定義而言，它是一組靜止點 (狀態) 以及讓項目移動至各種狀態的轉換。有效狀態表示它的擁有者目前負責處理該項目。使用者可以將項目轉換至各種狀態，將責任交託給另一位使用者。我們可以定義決定節點，它會使用商業規則根據所收集的資訊決定一個項目的傳遞路線。工作流程的各種內容會根據檢視項目的使用者之角色，控制轉換的存取權以及資料可見度。程序設計師可以透過新增或刪除狀態或轉換來修改一個工作流程；改變管理決策行為的業務規則；或是改變資料在工作流程、狀態或轉換層級的可見度。隨著時間推移，這些都是程序演變的典型變更，只要對業務程序有基本瞭解，再利用 SBM Composer 即可做到。

資料設計

SBM 中每個應用程式均有一個主表格，定義一個項目存在於程序中的生命週期裡收集到的資訊。當新要求出現時，程序設計師一般會修改資料模型，就是在主表格中新增欄位以收集更多資訊。例如，若程序設計師必須在資料模型中新增一個**迫切性**欄位，他們會從「表格」組合列表中拖曳一個「選擇」欄位到主表格中，將它命名為「迫切性」，然後在欄位的內容中新增數值「低」、「中」、「高」和「緊急」以供選選。使用欄位的「表單」索引標籤上的**新增至表單...**功能，指定相關欄位和相對位置，即可輕鬆在任何自訂表單中增加新欄位。如同工作流程變更一樣，資料設計的變更也相對簡單，不需要特殊的技術或程式設計技能即可完成。

視覺設計

除非您定義一個自訂表單，否則 SBM 會自動產生狀態和轉換的「快速表單」。若您使用快速表單，便無法在欄位上使用諸如**在表單上展開整列**以及**高權限區段**等內容來控制表單上欄位的版面配置。您可以變更工作流程、狀態或轉換內容的欄位權限順序來控制欄位的出現順序，而變更狀態的轉換內容則會改變轉換按鈕的順序。

雖然快速表單也許適用於建立系統原型或是不常使用的狀態或轉換，但是自訂表單有更大的彈性可決定資料呈現方式；能夠定義表單上的有效行為；以及允許使用嵌入式報告、HTML 框架和 REST Web 服務進行簡易整合。使用自訂表單時，您可以將檢視窗和資料輸入表單最佳化，讓最終使用者享受更好的服務。

設計自訂表單並不難。OpenText™ SBM Composer 視覺化表單編輯器讓您利用熟悉的拖放技巧新增和移動欄位。您可以用快速表單為基礎建立自訂表單或是從頭設計表單，新增欄位、容器、動作按鍵、影像、連結、報告、REST 網格等。使用自訂表單時，您可以全權控制表單上轉換按鍵的位置以及所有工具列和標題區的動作與顯示。

下一層級的視覺化設計則包括在表單中新增行為。例如，若您想要按照嚴重性自動選擇詳細資料索引標籤，您可以新增這樣的表單動作：

當

載入表單時

或嚴重性欄位的數值變更時

若

嚴重性欄位是「緊急」

則

啟動**詳細資料**索引標籤

否則

啟動**標準**索引標籤

可用來建立動態表單動作的表單動作數量極多而且各不相同，涵蓋大部份事件、情況以及您想在表單上執行的許多動作。至於我們未提及的情況，請參考商業規則。而且作為備案，您可以編寫 JavaScript 程式碼片段來導入更複雜的情況。同樣地，若表單未提供您想執行的動作，您可以編寫簡短的 JavaScript 程式碼片段，當出現某個事件且達到相應的條件時便執行一個動作。

設計自訂表單並不難。
SBM Composer 視覺化表單編輯器讓您利用熟悉的拖放技巧新增和移動欄位。

在部門的系統中使用過 SBM 以後，我們獲得最大成功的客戶變得更有信心為組織內各式各樣的程序建立新的程序應用程式。

程序設計師不需要任何特殊技能或程式設計知識即可完成幾乎所有自訂表單的工作。您可以聘請視覺設計師為使用者打造更好的視覺體驗，但是大多數時候沒有這樣的必要。

SBM 程序設計之創建

建立程序應用程式的工作與前述的維護工作相似，但是程序設計師與程序負責人要合作預想及設計整體的程序應用程式，而不是對現有程序進行逐步變更。在部門的系統中使用過 SBM 以後，我們獲得最大成功的客戶變得更有信心為組織內各式各樣的程序建立新的程序應用程式。他們發現了一道手動程序，會因為自動化、控管、稽核和報告而獲益。他們與程序負責人合作，定義程序的步驟和轉換、要收集的資料以及參與者的角色。他們接著使用 SBM 迅速建立程序應用程式來解決業務問題。大部份工作在 SBM Composer 的應用程式編輯器中完成，提供常見的譬喻和易用介面來設計程序。程序設計師可以在表格編輯器中設計資料模型、使用工作流程編輯器安排工作流程和指派角色將程序自動化，並且立刻在開發環境中讓程序開始運作。接著他們便可以反覆改善其設計，直到它成為最低可行性的產品，適合推出給最終使用者使用。

負責建立和維護程序設計的人員可能擁有商業分析師的頭銜而且要熟悉 SBM Composer。程序設計師不需要特殊的軟體工程或開發人員訓練，但是應該具備使用商業生產力工具如 Microsoft Excel 和 PowerPoint 的相應技能。

設計與導入整合

相對於建立與改進程序的工作，與外部系統整合是耗時費力的工作，不但要仔細規劃，而且導入過程複雜，還得進行周密的測試。複雜程度取決於整合的類型及所採用的協力廠商產品。在最簡單的情況下，可以使用自訂表單上相對簡單的 REST 網格來完成整合，只需要瞭解協力廠商 REST Web 服務介面而且知曉如何在 SBM Composer 中建置自訂表單即可。如果外部系統支援 SOAP (簡易物件存取通訊協定) Web 服務而且深入至業務邏輯的層面，那麼導入整合時可能要開發 SBM Orchestrations，這是比較複雜的任務。而在最複雜的情況下，可能需要撰寫程式才能存取外部系統的資料。

儘管 OpenText 盡力讓整合外部系統的作業變得更簡單，但是對 SBM 客戶而言這通常是最複雜的任務。如果所整合的系統在業務層級具有豐富的 SOAP (簡易物件存取通訊協定) 和 REST Web 服務介面，則不需要使用 SBM Orchestrations 撰寫程式和在表單上整合 REST 即可完成這樣的整合。然而，即便是如此，這項極複雜的工作仍不是商業分析師在行的事。雖然整合開發人員所需要的技能取決於他要處理的任務，不過大多與軟體工程師所具備的技能相差不遠。

因為整合是時間較長、花費較多精力的工作而且必須仔細規劃，它的開發過程不必如同前述的漸進式程序變更一般與現行業務緊密連結。我們可運用 SBM 的功能例如與應用程式無關的協調化來區隔這些開發工作，讓開發工作在完成時可以交付給程序開發人員，並由他們使用。

您只需要瞭解業務程序
並保持密切關係以及熟
悉使用 SBM Composer
和 Application
Administrator。

協調循環式長期開發工作

如上所述，使用 SBM 進行開發有兩種截然不同的必要工作。大部份工作與管理和視覺設計相關，不需要深厚的技術或程式設計知識即可完成。您只需要瞭解業務程序並保持密切關係以及熟悉使用 SBM Composer 和 Application Administrator。另一方面，與協調化和整合相關的工作通常需要運用不同程度的技術能力。這類工作比較複雜，往往要花費較長時間，而且需要比循環式維護更加謹慎。接下來您該如何透過長期的工作去協調，讓您的 SBM 程序接受這個快步調、反覆進行的適應性變更？本小節會說明數種方法以及 SBM 針對這個問題特別設計的相應功能。

雖然讓長期工作所使用的介面保持穩定可能會簡化該工作之整合，但是讓 SBM 程序保持流暢且不斷隨著使用者和業務要求而變動才是最重要的事。

循環式變更

一般而言只要該工作定義明確而且已有適當封裝，不需要仔細協調長期的工作，循環式變更也能繼續進行，如下所述。例如，若是所開發的協調化倚賴目前應用程式的某些方面，例如擁有特定欄位、狀態和轉換，則完成循環式變更後這些應該保持不變。雖然讓長期工作所使用的介面保持穩定可能會簡化該工作之整合，但是讓 SBM 程序保持流暢且不斷隨著使用者和業務要求而變動才是最重要的事。經過適當封裝後，便可以快速又輕鬆地把長期工作納入程序中。

循環式變更的典型程序在《企業採用 SBM 方法上線》定位報告中有詳細說明。

長期開發的工作

在 SBM 執行長期開發工作時，主要目標是將您的工作結構化，讓他們透過簡單、定義明確的介面與相關的程序應用程式相互作用。他們應該封裝起來而不要暴露複雜的內涵。這些開發工作可以在 SBM 內部或外部進行，而且一般程序設計師通常不具備相關技能，或者他們對此並不在行。參與這些工作的開發人員需要一個獨立於主要 SBM 系統之外的開發用 SBM 系統執行個體，用來執行循環開發、預備和上線。

SBM 外部的開發

這類開發的例子是建立一個 REST 服務揭露來自協力廠商業務線軟體的資料，而該資料可以顯示在 SBM 自訂表單 REST 網格上。依情況而定，其中可能包括編寫一個與外部系統的應用程式介面 (API) 相互作用程式碼模組以及服務代管。同樣地，也可能包含建立和代管 SOAP (簡易物件存取通訊協定) Web 服務，這項服務會揭露一個外部系統詳細的業務功能，適合在 SBM Orchestration 中使用或是用作轉換時的 Web 服務動作。我們需要一個開發用 SBM 系統，才能確認所開發的介面符合 SBM 的要求。依上述範例的情況，可能意味著要使用 SBM 來測試正在開發的 REST 和 SOAP (簡易物件存取通訊協定) Web 服務。按照其性質，這類開發工作會被封裝起來，因為它對 SBM 而言是定義明確的高層級介面。

在 SBM 內部開發

這個開發類別就是使用 SBM 提供的各種技術整合科技。例如開發協調化流程，提供適當的錯誤處理和補償功能。開發中會呼叫 SBM C API 的複雜程序檔或軟體也屬於這個類別。這些工作通常與 SBM 外部的開發工作一起進行。例如，外部系統可能僅揭露層級極低的應用程式介面 (API)。這個應用程式介面 (API) 必須由業務層級的 Web 服務包裹在外，接著即可由 SBM Orchestrations 使用。

SBM 開發封裝技巧

應用程式工作流程中轉換的各項動作擁有最多與已封裝開發工作互動的接合點。例如，同步工作流程、異步工作流程、AppScripts 和 Web 服務全是由轉換時的「動作」所啟動。其他主要整合點是最終使用者看到的表單上的實時整合。下列小節說明不同類型的整合並探討建立合適區隔和封裝的開發慣例，以利後續與主要 SBM 系統的整合更順利。

一般考量 — 內部名稱：以下探討的許多整合技巧使用 SBM Web 服務、AppScript 或 C API 與 SBM 應用程式內的工作流程互動。由於欄位名稱可能會變更，與欄位互動的所有 Web 服務應當使用欄位資料庫名稱，這個名稱在初始部署後即不可變更。工作流程、狀態和轉換的所有參照均應使用其合格的內部名稱，因為那些項目的名稱內容可能模糊不清，或是當應用程式為因應變更需求而修改時會有變動。

協調化流程：此小節會考慮四種類型的協調化流程。若您在長期開發工作中開發一個協調化流程，可能的話，您應該在另一個程序應用程式中開發該協調化流程，讓後來的整合更便利。接下來探討不同類型的協調化流程。

- **同步協調化工作流程：**這類協調化工作流程是在 SBM 應用程式中由轉換動作呼叫出來的。項目資料對應到協調化工作流程的輸入並由協調化工作流程的輸出更新。同步協調化流程必須與呼叫它的應用程式存在於同一個程序應用程式中。因此，在開發這類協調化流程之後必須使用 SBM Composer 的比對/合併功能將它與主要 SBM 系統上的程序應用程式合併。同步協調化工作流程的規格應定義同步協調化流程的輸入和輸出，而且這些欄位在開發期間要維持不變。

SBM Orchestrations 分為四種類型：

- 同步協調化工作流程
- 相依於應用程式的異步協調化工作流程
- 獨立於應用程式之外的異步協調化工作流程
- 外部啟動的異步協調化工作流程

與應用程式無關的異步協調化工作流程是最常見的由 SBM 應用程式啟動的協調化。

- **相依於應用程式的異步協調化工作流程：**這是經由 SBM 應用程式啟動的最常見的協調化流程。使用轉換動作引發一個事件即可執行協調化流程。事件資料對應於該應用程式的主表格中所包含的資料。這些協調化流程可以呼叫外部系統和 SBM 應用程式的 Web 服務。一如往常，在開發協調化流程期間，您應該謹慎地讓應用程式和這個協調化流程之間的介面維持不變。您應確保應用程式中無回覆的事件協調化連結已勾選「鎖定定義」核取方塊，以便主表格的欄位重新命名後，與協調化流程之間的介面保持不變。SBM 2009 R4.02 及以上提供這個機制。

這些協調化流程與呼叫它的應用程式不必存在於同一個程序應用程式中。要在另一個程序應用程式建立相容的協調化流程，請將名為無回覆的事件的「協調化連結」從啟動的應用程式匯出，然後在另一個空白的程序應用程式中，加入一個協調化流程。在 Application Explorer 的「應用程式連結」項目之下，按一下滑鼠右鍵並選擇新增事件定義…功能表項目。在出現的對話方塊中，選擇用事件定義建立的檔案選項，瀏覽至您匯出的事件定義，然後便可以繼續開發。一如既往，您需要有一個 SBM 系統來引發要測試的事件。

- **與應用程式無關的異步協調化工作流程：**這些是包裹外部系統的異步協調化工作流程，而且可由多個 SBM 應用程式重複利用。這些工作流程的事件介面獨立於應用程式之外，而且由協調化開發人員定義，以利對外部系統做出充足的指示。您可以把這些工作流程想像成定義連接至外部系統的可重複利用的通用介面，其中每個工作流程代表一個指令。使用轉換時的一個動作即可從 SBM 應用程式呼叫這些工作流程。他們與相依於應用程式的異步協調化工作流程不同，因為應用程式的項目資料可對映至事件介面，而相依於應用程式的事件介面由應用程式的項目資料自動定義。

要建立獨立於應用程式之外的異步協調化工作流程，您得建立新的程序應用程式並增加一個協調化流程，請在 Application Explorer 的程序應用程式上按一下滑鼠右鍵然後選擇新增 -> 協調化功能表選項。在 Application Explorer 的「應用程式連結」項目之下，按一下滑鼠右鍵並選擇新增事件定義…功能表項目。在出現的對話方塊中，選擇建立新的自訂事件定義選項；接著定義「物件類型」和「事件類型」。事件管理員會使用這些數值來判斷某個事件被引發時要執行哪個協調化流程。現在請定義在事件中傳遞的自訂資料。這些數值將是由這個事件運作的協調化工作流程的輸入值。

- 要引發由這個應用程式連結定義的事件，請先匯出該事件，方法是按一下**匯出事件定義**... 按鈕並把它儲存成 .mtd 檔。現在，在初始啟動的應用程式中，在 Application Explorer 的**協調化連結**之下的**外部事件**項目上按一下滑鼠右鍵，然後選擇**新增外部事件**... 功能表項目，瀏覽至事件檔案並把它匯入到應用程式。最後，要使用這個事件檔，請建立一個轉換時執行的動作，選擇協調化工作流程，並且在精靈首頁的第 2 步中，按一下**本機事件並變更為外部事件**。結束精靈的操作，選擇您剛才匯入的外部事件，然後按一下資料對映，將應用程式的項目資料對映到事件的輸入。唯有 SBM 10.1 及以上提供這個機制。
- **外部啟動的異步協調化工作流程**：這是由外部系統所引發的事件所啟動的異步協調化工作流程。這些協調化工作流程的事件介面是根據外部系統的要求予以定義。這些協調化工作流程通常稱為 SBM Web 服務，用來操作特定應用程式的項目。
為了建立可以從外部系統啟動的協調化流程，請按照上述相同的步驟建立一個協調化流程並定義新的自訂事件定義。我們不再匯出事件定義，而是按一下「**匯出外部事件 WSDL**」按鈕來建立一個 WSDL 檔，用來指定適合引發事件的 Web 服務。詳細資訊和說明請參閱相關文件。

其他進階開發技巧：其他類型的進階開發包括 AppScript 開發和 C API 開發。儘管許多 SBM 安裝實例利用這些介面，您若是能夠利用表單動作、Web 服務和協調化流程來完成您的工作，那麼您的系統會更容易維護。

作為獨立於主要 SBM 系統之外的開發工作，所有這些開發類型最關注的事情就是接合點皆已清楚明瞭且穩定。SBM 10.1 引進的「**內部名稱**」特性讓這件事比從前簡單許多。在這之前，任何 SBM 工作流程模型的參照，包括工作流程、狀態和轉換皆以名稱為據。由於表單上看得見狀態和轉換名稱，平常的維護和工作流程的演變可能會因改變名稱而破壞了整合。現在，使用 AppScript、Web 服務 API 或 C API 參照這些項目時可以使用內部名稱，在程序應用程式初次發行以後便維持不變。

將長期變更合併到開發、預備和上線系統中

長期專案開發完成時，您必須把這個工作整合到主要 SBM 系統。

長期專案開發完成時，
您必須把這個工作整合
到主要 SBM 系統。

您可以使用 Composer 的比對與合併功能，把您在另一個系統完成的任何開發與主要 SBM 系統的程序應用程式合併起來。

協調化流程

對於在另一個程序應用程式中開發的協調化流程而言，這個程序相當簡單。若是由外部系統啟動的協調化流程，您只需要部署包含該協調化流程的程序應用程式，它便會在外部系統引發事件時執行。至於其他異步協調化類型，您只需要修改執行呼叫的應用程式來引發呼叫協調化流程的事件。若是使用外部事件定義協調化流程，您必須將執行呼叫的應用程式的資料對映到協調化流程的輸入。若協調化是根據應用程式資料模型予以定義（相依於應用程式的協調化），請在轉換時執行的動作中引發事件。

對於同步協調化流程，或是您保留在相同的程序應用程式中作為相關應用程式的協調化流程，您必須使用 Composer 的比對與合併工具，將開發程序應用程式藍圖的協調化流程及其「協調化連結」複製到主要 SBM 系統中的程序應用程式，然後在您要完成工作的轉換中呼叫協調化流程。

合併 APPSCRIPTS

不同的程序應用程式之間可以用檔案的形式複製 AppScripts。只要與它互動的 AppScript 和應用程式之間的介面保持穩定（例如因為使用內部名稱），它在複製到主要系統之後仍能繼續運作。

合併其他變更

最後，您可以使用 Composer 的比對與合併功能，把您在另一個系統完成的任何開發與主要 SBM 系統的程序應用程式合併起來。您可以使用該功能，將您在不同版本的程序應用程式所做的任何變更複製到主要系統的程序應用程式。幾乎所有類型的設計元件都可以使用這個功能，包括協調化、表單、工作流程、AppScripts、規則、角色、欄位和輔助表格。若某個設計元素有所變更，經比對產生的報告會指出差異之處，並用視覺化方式在您熟悉的 SBM Composer 環境中突顯其差異。

雖然您可以利用這個技巧將任何分開的開發工作整合到主要系統中，您應該設法把任何長期專案封裝在分別的設計元素之中，即可作為一個單元移動到主要系統中，不需要辛苦地將現有物件的不同內容分類整理。

重點總結

享受 SBM 所帶來的效益，讓程序能夠不斷適應企業的變化，而不是強迫企業配合僵固的程序。

使用 SBM 即可用輕鬆又直覺的方式適應大多數變更並且隨時間推移不斷改進程序。SBM 是專為精通業務的非技術人員而設計，讓他們能夠進行循環式程序變更。請讓這些人員與程序保持密切關係。

指派一項任務給程序負責人，讓他負責定期召開論壇，請參與執行程序者派代表參加，包括提交者、履行者和程序觀察者，連同程序負責人一起。這個論壇應探討人們如何運用解決方案並瞭解它能改善哪些事情。根據團隊提供的回饋意見反覆改進程序。

有些工作，例如整合，讓開發人員以長期專案的方法完成的話效果最佳。將這些工作封裝起來，既可進行獨立開發，也可以與主要 SBM 系統輕鬆結合。

不要把能夠為組織立即帶來效益的小型專案與較大型的開發工作網綁在一起。這麼做會讓 SBM「為改變而打造」的價值主張有所減損。

如需更多資訊，請瀏覽
www.opentext.com

與我們交流

