

白皮書

Enterprise Analyzer
Enterprise Developer
Enterprise Sync
Enterprise Test Server

大型主機 DevOps 實作現況

從理論到現實 — 大型主機 DevOps 最佳實務指南

目錄

頁碼

關於本白皮書	2
協同作業.....	3
加速交付但品質不打折.....	3
效率	9
用更好的流程管理程式碼變更.....	9
彈性	13
擴充開發和測試產能同時控制成本.....	13
總結	18

成功志在必得。企業需要整個 IT 環節而非一小部份的投入，才能合力加速軟體交付，以便滿足客戶不斷變化的要求及維持競爭優勢。

雖然 IT 部門交付軟體沒有標準做法，但是大部分開發主管會同意，無論什麼樣的交付方式，加速交付創新的應用程式是務必達成的目標。常見動機包括保持競爭優勢、降低作業成本、增進 IT 對業務的支援，或是上述因素的不同組合。

既然上市時程作為一項明確的商業優勢比任何事情都更重要，但是資訊長念茲在茲的總是如何用更簡單的方法做更少的事並把事情做得更好。這些全是企業想要加速交付創新應用的重要原因。最近一項研究¹指出 45% 的 IT 專業人士計劃在企業組織中導入 DevOps 實務，也許正是出於上述考量。

新創和小型公司從一開始便採用提升效率的工具和現代化協同作業式工作實務，因為不受流程限制而且能彈性、快速地適應變化，因此發展的速度更快。不過，加速開發對於歷史悠久的老牌企業來說卻是極大挑戰。

複雜的技術以及各個團隊使用多種不同的開發流程是老牌企業的典型包袱，大部分大型主機用戶都有這項特徵。所以，您的組織可否讓新的交付概念適用於舊有的基礎架構，透過符合成本效益的方式提高效率？

答案是肯定的。並不是任何特定的業務模式或開發模型才能享受 DevOps 的成果。每一個組織皆可透過百分比遞增式的改進達成特定的業務驅力。要透過行動裝置和 Web 交付新服務？讓流程更精簡以利增加市佔率或擴大業務所及地區？要改善大型主機的效率和品質以利提高利潤？事情的優先程度可能各有不同，但是全部目標皆可望達成。

DevOps 所特有的平行開發、持續整合與交付、提早與頻繁測試之特色，原是從分散式運算環境中發展出來並以敏捷式開發實務為基礎。這些特色要如何移轉到大型主機環境及非採用 DevOps 概念設計的應用程式？因為我們對成功志在必得。企業需要整個 IT 環節而非一小部份的投入，才能合力加速軟體交付，以便滿足客戶不斷變化的要求及維持競爭優勢。

¹ www.computing.co.uk/ctg/news/2461474/more-firms-are-planning-adopting-and-successfully-implementing-devops-research

大型主機導入 DevOps 的成功關鍵在於迅速找出並排除大型主機應用程式交付週期中的重大瓶頸。當中三項主要難題包括開發與測試團隊之間缺少協同作業、開發與流程之整合效率不彰以及資源運用彈性不足以致無法依需求擴充測試產能。

關於本白皮書

DevOps 宣揚大使們經常談到「DevOps 夢想」，但是卻甚少提到實際的操作細節。本白皮書透過我們挑選的客戶案例，說明客戶如何發現並排除妨礙大型主機迅速交付應用程式的特定瓶頸。我們根據在全球各地協助大型主機客戶採取漸進式途徑並配合業務需求和預算以持久的速度達成目標的經驗來撰寫這份白皮書。使用個案的成就和實證已證明了這項策略相當有效。

這份白皮書的主軸就是說明客戶如何利用專為達成大型主機 DevOps 成果而設計 Micro Focus® Enterprise 解決方案組合所包含的產品。以下是本白皮書所參照之產品的簡要說明。

Enterprise Analyzer®²

智慧與分析工具，深入解析應用程式和業務流程。

Enterprise Developer®³

整合 Visual Studio 或 Eclipse 的現代化應用程式開發環境 (IDE)，簡化大型主機 COBOL 和 PL/I 開發。這是直接與大型主機整合的工具組，能改善遠端開發的效率。

Enterprise Sync®⁴

分散式軟體組態管理解決方案，協助企業組織擴充平行開發與持續整合的規模，同時保持大型主機原始碼的完整性及組態管理功能。與 Enterprise Developer 整合。

Enterprise Test Server®⁵

Windows 上的 IBM 大型主機應用程式測試環境。IT 組織利用這項產品在可擴充、低成本的硬體上順利測試大型主機應用程式的變更。

Micro Focus Enterprise 解決方案組合包括：

- Enterprise Analyzer
- Enterprise Developer
- Enterprise Sync
- Enterprise Test Server

² www.microfocus.com/products/enterprise-suite/enterprise-analyzer

³ www.microfocus.com/products/enterprise-suite/enterprise-developer

⁴ www.microfocus.com/products/enterprise-suite/enterprise-sync

⁵ www.microfocus.com/products/enterprise-suite/enterprise-test-server

利用直覺易用的現代化工具以及開發與測試整合式環境來鼓勵協同作業，可改善大型主機應用程式交付之效率達 40%。

我們在白皮書中分別說明促成大型主機成功導入 DevOps 的三大支柱，即協同作業、效率和彈性。現實世界全球各地客戶的成功案例將示範說明每一個支柱。我們不會明確公開客戶的名稱，但是他們的成就以及達成目標的方式，對於任何擁有相同的目標的人而言皆有說服力。

協同作業

加速交付但品質不打折

DevOps 本質上是交付應用程式與服務的一種方法，其中的文化重視整個交付鏈的時間運用和貢獻。因為軟體發佈更加頻繁了，所以業務線、開發、測試與 IT 營運團隊之間必須緊密配合並進行協同作業。

若 DevOps 的重點是促進部門之間的溝通，那麼開發部門的成員若無協同作業的文化，如何能成功？可惜的是，這反而是常見的情況。大型主機團隊通常是孤立的，因為他們使用大型主機專用開發工具，而且慣用的實務做法既過時又太複雜，即使是同輩的開發人員亦有同感。

在理論上，DevOps 能夠清除開發團隊之間這些陳年障礙，而把理論付諸實現正是困難所在。塑造「協同作業文化」聽起來很簡單。只要跟隨知名企業⁶的做法就好；打破每個小房間的隔間牆，當陽光照射在全新、寬廣的開放式樓面上，協同作業文化便會由此茁壯成長嗎？

根據研究人員 Vinesh Oommen 指出，一項有關開放式辦公室的研究⁷顯示：「90% 的研究結果指出，人們認為在開放式辦公室工作帶來了負面效果，因為開放式辦公室造成極大的壓力、衝突、高血壓和極高的員工流動率」。

⁶ www.forbes.com/sites/aliciaadamczyk/2015/03/31/inside-facebooks-incredible-new-frank-gehry-designed-headquarters-mpk20/#5aaa257b416e
⁷ www.intheblack.com/articles/2016/02/01/open-plan-offices-work-for-your-boss-but-not-for-you

解開「協同作業文化」之謎

Micro Focus 的客戶把牆推倒了 (非字面意義的牆)，也因此解開了協同作業文化之謎。利用直覺易用的現代化工具以及開發與測試整合式環境來鼓勵協同作業，可改善大型主機應用程式交付之效率達 40%。下列使用案例取自一家大型銀行與信用卡發卡機構，這家機構與不同的境外服務供應商以及公司內部的 z/OS 大型主機開發、測試與營運團隊一起合作。以下是他們的故事：

挑戰

為了更快並在較短的週期內交付新功能，組織必須讓開發流程更精簡。然而，不同的開發團隊擁有各自的開發流程，他們使用傳統的 ISPF 進行應用程式的分析、變更和測試。此外，在他們的發佈管理環境 Changeman ZMF 中，變更和組態管理設定並未一致套用。這樣一來便無法加速完成企業所要求的應用程式變更。

解決方案

該公司使用 Micro Focus 大型主機應用程式分析工具及其整合式開發環境來縮短交付週期並變得更加敏捷。為了建立簡單的標準化流程，讓每個人使用相同、一致且可重複的方法作業，該企業組織導入了 Micro Focus Enterprise Developer。開發人員在這樣的整合式開發環境 (IDE) 中善用現代化工具來改善生產力。整個部門使用通用的 IDE，各個團隊之間便更容易進行協同作業，因為大家都看得到程式碼變更。

為了建立簡單的標準化流程，讓每個人使用相同、一致且可重複的方法作業，該企業組織導入了 Micro Focus Enterprise Developer。

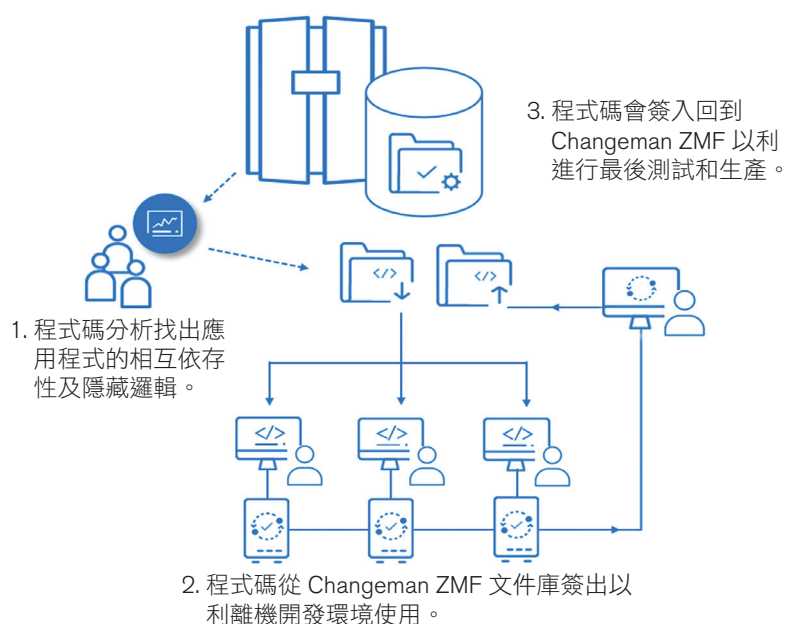


圖 1

現代化開發工具允許整個開發團隊看得到必要的程式碼變更，藉此改善協同作業、生產力和品質。

完成所有變更和測試以後，原始碼元素會再簽入 Changeman ZMF。

為了更容易找出所要的變更，作為大型主機應用程式分析工具的 Micro Focus Enterprise Analyzer 會分析大型主機應用程式空間中的對應關係。分析師和開發人員可藉此發現應用程式的相互依存性和隱藏的應用程式邏輯。在變更應用程式時，分析工具會自動找出所有受影響的原始碼元素，然後從對應的 Changeman ZMF 文件庫中簽出，再簽入離機的 Enterprise Developer 來執行作業。

如圖 2 所示，一旦在本機變更並編譯程式碼之後，開發人員便執行除錯程序檔並進行單元測試來驗證變更，並確保同一時間在相同環境中修復的其他變更也可以正常運作且不會影響其它程式碼。完成所有變更和測試以後，原始碼元素會再簽入 Changeman ZMF。

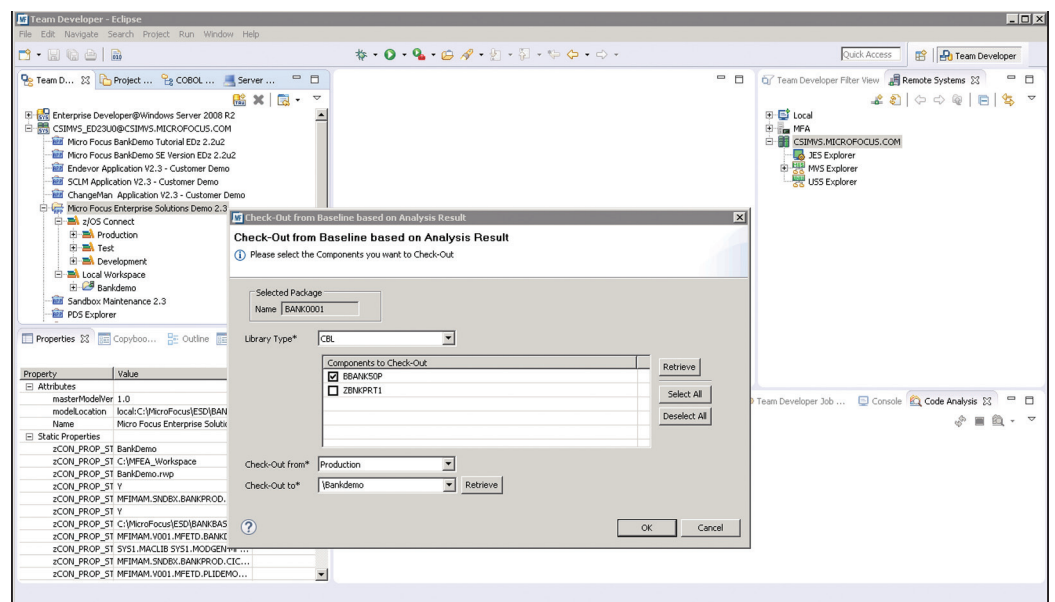


圖 2

Enterprise Analyzer 與 Enterprise Developer 整合後帶來快速、即時的搜尋和探查功能，開發人員得輕鬆地迅速找到需要變更的受影響原始碼行。

成果

- 把耗時、容易出錯的多項活動包裹成可在現代化 IDE 中執行的簡單易用的動作，改善了開發人員的生產力。
- Enterprise Analyzer 和 Enterprise Developer 所營造的協同作業環境讓程式碼變更變得顯而易見，改善程式碼的可信度和品質。
- 不必再等待大型主機單元測試的資源或是重設測試個案資料了，Enterprise Developer 為開發人員提供 Windows 上的本機 z/OS 子系統以及一組本機測試資料。

大型主機只要採納現代化的開發工具，便能大幅改善各個開發團隊之間的協同作業效率，並達成符合 DevOps 水準的可靠度和交付速度。

移除創新的絆腳石

協同作業的優點並不「只是」加快交付速度的生產力優勢而已。Forrester 指出，「最終的目標並不是更快地交付舊的程式碼，而是除去妨礙核心產品與流程之創新的絆腳石。」(Kurt Bittner 與 Robert Stroud，「Digital Transformation Needs Mainframe DevOps」(數位轉型需要大型主機 DevOps)⁸ Forrester 報告，2016)

對許多客戶而言，大型主機「記錄系統」資源與分散且虛擬化的「互動系統」資源之間必須相互配合，方能促成核心產品與服務之創新。新式的行動裝置與雲端型應用程式必須與核心大型主機應用程式搭配運作才能交付出優秀的創意。遊戲設計師暨作家 Jane McGonigal⁹ 評論道：「協同作業的重點並不是達成目標或結合眾人力量；它的目標是要合力創造出非一人之力所能成就的事。」她說明協同作業需要三種協力方式：合作、協調和共同創造。最重要的是共同創造，因為協同作業務必要透過共同創造來達成。

那麼，真實世界中要如何實踐呢？這個故事會重點說明歐洲的託管服務供應商如何利用統一的開發環境來培養不同團隊之間共同創造的文化。他們藉此讓本身的大型主機應用程式服務創造出新的價值，超過 16 個國家的公營事業與私營企業客戶皆受惠。

「協同作業的重點並不是達成目標或結合眾人力量；它的目標是要合力創造出非一人之力所能成就的事。」

—Jane McGonigal

⁸ www.forrester.com/report/Digital+Transformation+Needs+Mainframe+DevOps/-/E-RES134941

⁹ www.janemcgonigal.com/meet-me/

利用 Enterprise Developer 導入現代化大型主機開發環境是邁向大型主機 DevOps 的第一個重要步驟。

挑戰

服務供應商必須改善執行 COBOL、CICS、DB2 和 JCL 的傳統大型主機團隊以及在敏捷式、持續整合的環境中工作的 Java 團隊兩者在開發和測試極複雜的大型主機應用程式的生產力。若要存取大型主機才能進行編譯和測試，這樣不但限制甚多且效率不佳；而大型主機工具對於專案時間表的影響尤大。這些挑戰使得優先安排、規劃和交付客戶的強化功能要求更難實現。

解決方案

利用 Enterprise Developer 導入現代化大型主機開發環境是邁向大型主機 DevOps 的第一個重要步驟。服務供應商可藉此將傳統大型主機 COBOL 開發成功移轉到 Java 開發團隊所使用的敏捷式流程。建立一個持續整合 COBOL 應用程式的平台，可確保一天數次將變更併入應用程式基線。

這位客戶利用 Enterprise Analyzer 迅速找出應用程式需要變更的範圍並瞭解其複雜度，讓開發團隊可以更準確地估算為客戶提供優質服務所需的時間和資源。

成果

- 在共同創造和管理混合型應用程式時，分析和開發工具所提供的可見度和持續控管有助於協同作業、維持品質和創新。
- 管理完整的大型主機應用程式組合速度更快。
- 開發和測試時減少對大型主機的依賴降低了大型主機 MIPS 耗用量，所節省下來的可以轉移給用戶端或是重新投入到其他大型主機生產工作負載中。
- 能夠更加頻繁進行測試，改善了應用程式品質。
- 增加協同作業的機會，改善了開發人員的技能，也讓新進人員入職更順利，這項優點也有利於提供新的顧問諮詢服務。

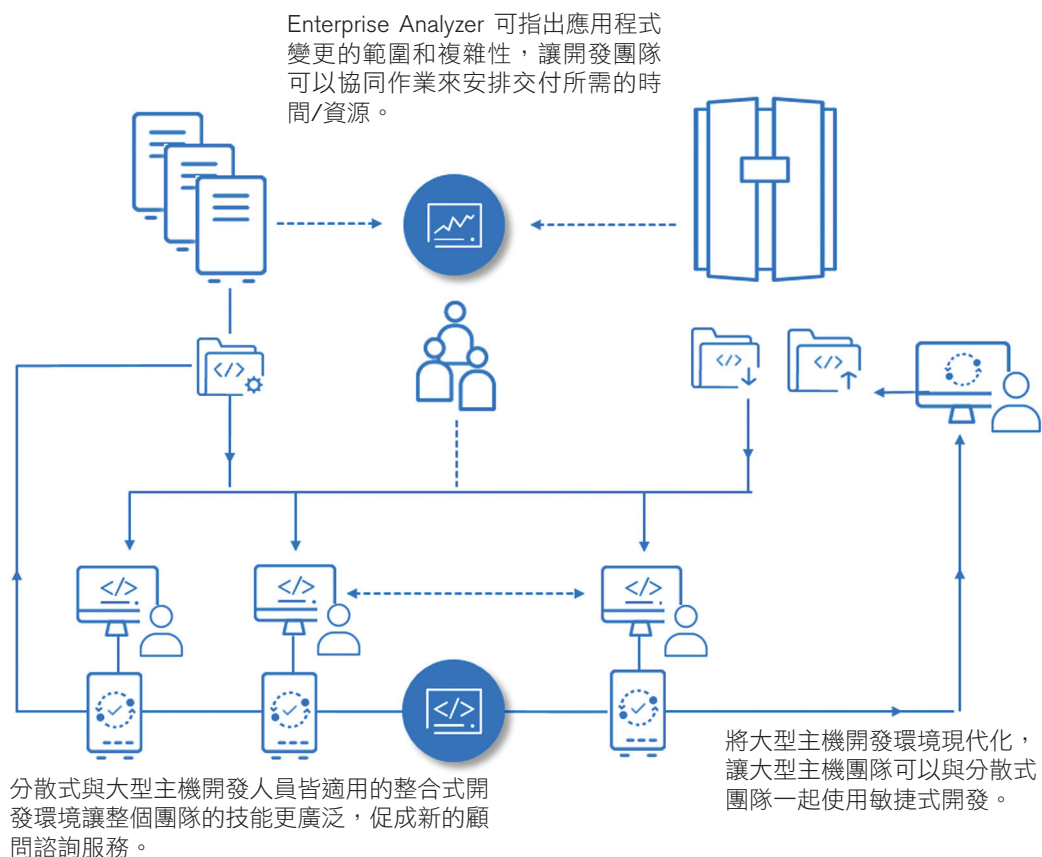


圖 3

Micro Focus Enterprise Developer 是現代化的 IDE，將分散式和大型主機開發團隊團結在一起。

能夠進行 DevOps 層級的協同作業

若 DevOps 的目標是更快達成更好的業務成果，加強運用您的核心業務流程和記錄系統，那麼大型主機必須在分散式和大型主機開發團隊兩者之間採行協同作業的文化。

若 DevOps 的目標是更快達成更好的業務成果，加強運用您的核心業務流程和記錄系統，那麼大型主機必須在分散式和大型主機開發團隊兩者之間採行協同作業的文化。

希望更快交付創新應用的大型主機用戶受制於線性工作流程的時間限制，無法為多個開發團隊提供足夠的支援以利同時交付多個軟體版本。

DevOps 層級的協同作業：自我評估

- 您是否覺得與大型主機相關的活動無法與其他分散式 (互動) 系統的活動配合？
- 各個團隊之間的溝通是否順利，或變更要求是否變成「雞同鴨講」的狀況？
- 您的 Java 或 .Net 開發人員可否存取大型主機原始碼來建置混合型應用程式或服務？
- 您是否完全瞭解大型主機環境而能理解變更對於相互依存性有何影響？

效率

用更好的流程管理程式碼變更

由於大型主機已存在數十年，大型主機開發環境中內嵌的技術和流程、部署執行時期及組態管理皆採用舊方法。從概念到交付給客戶為止，大型主機應用程式開發與發佈流程皆無連續性，進度緩慢又容易發生錯誤。希望更快交付創新應用的大型主機用戶受制於線性工作流程的時間限制，無法為多個開發團隊提供足夠的支援以利同時交付多個軟體版本。

面對平行開發活動所造成的難題，分支 (branching)、合併 (merging) 與標記 (labelling) 不但是最佳實務做法，在 DevOps 實務上也一直被用作控制版本的工具。但是大型主機開發環境與其 SCM 工具密不可分，而 SCM 工具的新改良功能卻無法觸及這些環境。開發人員必須手動合併簡單與複雜的程式碼行。這樣一來便造成版次不連續的問題，因此必須重新作業，讓開發人員備感挫折。所以挑戰也就顯而易見：要支援平行開發並且用簡單又快速的方式管理、協調和策劃多個變更流，但不要耗用更多大型主機資源。

接下來的使用案例將說明一家大型的全球保險公司旗下的全資子公司如何一邊在大型主機上執行該公司支援多個品牌的核心保單應用程式，一邊達成持續整合的目標。應用程式組合包含 COBOL、CICS、Batch、DB2、SAS、REXX 和 Web 服務並透過 CA Endevor Software Change Manager (SCM) 平台維護和管理。

挑戰

企業為了把變更加速交付到核心應用程式而採納了平行開發的做法。當程式碼要統合成單一路徑並透過 CA Endevor 進入生產時便出現了難題，即交付瓶頸。這個環境無法用簡單的方法延伸，以利滿足大規模進行平行開發的需求。於是，企業迫於使用容易發生錯誤的手動方式置入程式碼變更，這種情況在大規模、長時間運行的大型主機應用程式功能強化過程中尤為常見。

解決方案

保險公司搭配使用 Enterprise Developer 與 Enterprise Sync。CA Endevor 所維護的應用程式原始碼會聯同原始碼版本控制一起自動同步化到分散式環境，在該環境中的開發人員使用更有效率的工具管理平行開發流的原始碼變更。這個解決方案執行超過三個月並分成三階段進行，表示 Micro Focus 分散式工具必須與悠久的大型主機既有流程搭配運作。

在第一階段，開發人員從舊式的大型主機開發工具轉變成現代化的 Eclipse 平台同級技術，可直接存取大型主機上的原始碼。這樣一來，當開發團隊進入第二階段時，改採現代化工具的過程便會更順利。CA Endevor 的原始碼管理功能會藉由 Enterprise Sync 延伸到分散式環境。開發團隊可利用原始碼複製在分散式環境中設置多個同時執行的原始碼基線，並與 CA Endevor 的平行開發活動自動同步化。如今，開發人員運用現代化的原始碼控制管理工具有效地管理程式碼變更、合併與改造。他們的工作效率更高，平行開發流的管理效果更好且更有彈性。

這個解決方案執行超過三個月並分成三階段進行，表示 Micro Focus 分散式工具必須與悠久的大型主機既有流程搭配運作。

為開發人員提供現代化且有效率的開發工具，讓保險公司更快導入程式碼變更且更有信心。

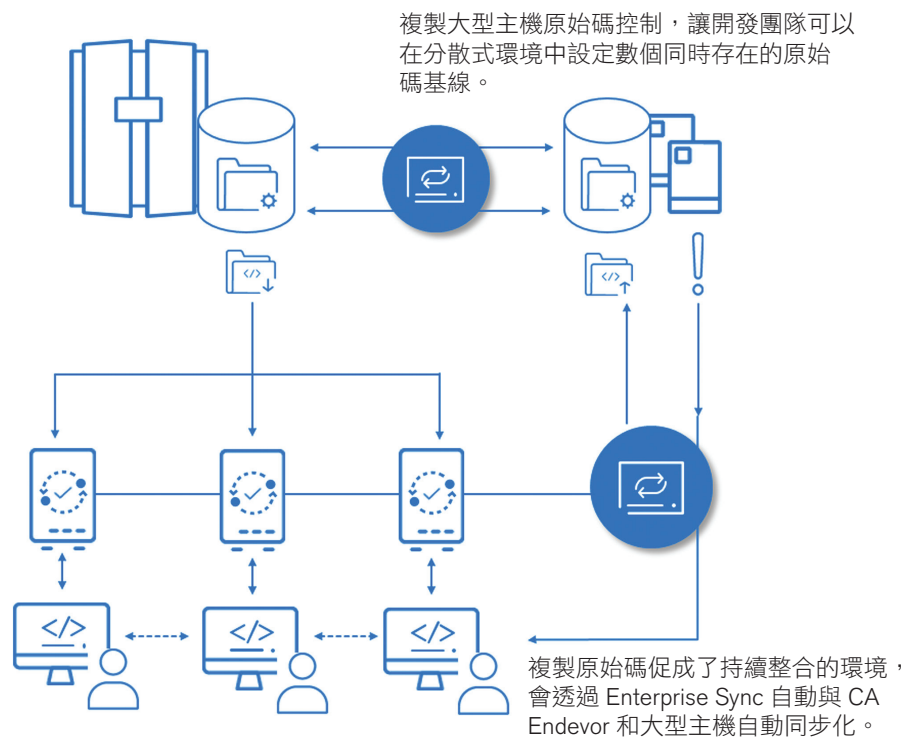
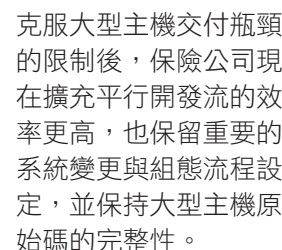


圖 4

將大型主機原始碼版本控制與分散式環境同步化，開發人員可以使用現代化工具自動偵測程式碼衝突

成果

- 為開發人員提供現代化和有效率的開發工具，讓保險公司更快導入程式碼變更且更有信心。
- 程式碼變更的可見度更高，所以在整個開發週期內更容易及早發現品質問題。
- 藉由大型主機原始碼控制與分散式環境同步化，開發人員得以使用現代化工具自動偵測程式碼衝突、查看差異並利用工具輔助進行合併。把變更置入程式碼的速度更快，出錯機會也比較少。



5

Enterprise Sync 自動偵測程式碼衝突並利用輔助工具執行合併

克服大型主機交付瓶頸的限制後，保險公司現在擴充平行開發流的效率更高，也保留重要的系統變更與組態流程設定，並保持大型主機原始碼的完整性。

能夠達成 DevOps 層級的效率

大型主機組織要有效擴充至 DevOps 層級的效率，即管理變更的速度更快，省卻耗時或高成本的重新作業，就應該利用現代化的開發工具。Micro Focus 工具把大型主機原始碼複製到分散式 SCM 平台。開發團隊利用現代化工具即可透過協同作業更快發佈更多軟體版本，沒有任何限制。開發團隊可自由地平行作業，而且對於所發佈的軟體信心滿滿。

這項獨特的低風險方法確保程式碼變更會自動與大型主機的軟體變更與組態管理器 (SCCM) 同步化。這樣做可以確保大型主機 SCCM 環境始終是主要的記錄系統。

傳統大型主機交付方法在規劃以及資源配置時採用循序的流水式作業。

DevOps 層級的效率：自我評估

- 您的軟體交付速度是否足夠應付業務需求？
- 您的開發人員可否使用平行開發流程來交付大型主機應用程式嗎？
- 您可以實行持續整合嗎？您的個別變更是否提早整合，以避免在交付週期的後期發生更大的問題？
- 您的開發團隊在至生產為止的過程中是否需要手動重新撰寫程式碼來管理變更？
- 對企業來說，達成更高層級的大型主機效率或品質是否重要？

彈性

擴充開發和測試產能同時控制成本

傳統大型主機交付方法在規劃以及資源配置時採用循序的流水式作業。珍貴的大型主機資源，特別是機器和人員，皆有時間限制，這是一個既缺乏彈性也難以擴充的僵化架構。錯過了交付里程碑就等於長時間延遲，直至大型主機資源再次可供開發和測試使用為止。這家亞太地區大型綜合保險公司當初面臨的就是這樣的情況。他們的 IBM 大型主機上裝載著非常多種技術，包括 COBOL、Batch、CICS、IMSTM/DB、DB2、MQ、CA Endevor SCM。有數個品牌和公司使用其核心的保單應用程式。以下是他們的故事：

挑戰

企業對於新應用程式功能的需求增加了三倍，面臨必須加快發佈新版本的壓力。他們的大型主機沒有產能容納更多的測試環境，所以無法快速交付出好的品質，也沒有辦法購買更多的大型主機基礎架構來支援增加測試。

解決方案

解決方案就是使用 Micro Focus Enterprise Developer 以及 Micro Focus Enterprise Test Server 之下運行的四個分散式測試區來導入一個大型主機應用程式交付模型。結果如何？增加了測試產能之後，開發團隊不必依賴大型主機資源也能夠更快速地測試應用程式變更，結果也更可靠。

由 Micro Focus 啟動流程意味著當開發人員執行程式碼變更時，或是作為預定流程的一部分，它會自動建置一個完整的測試環境。

1. CA Endevor 內指定層級的原始碼會同步化到執行 Enterprise Developer 的 Windows 組建機器 (Build Machine) 上的原始碼庫。
2. 組建機器利用 CA Endevor SCM 處理器群組的中繼資料來推動變更流程。應用程式載入模組的組建方式與在大型主機上相同 一本質上就是在雲端模擬大型主機測試環境。
3. 組建好的載入模組接著隨著測試資料和區域組態一起發佈到產出工件儲存庫。
4. 這些產出工件接著會自動部署到執行 Enterprise Test Server 的 Windows 型伺服器的雲端執行個體，以便啟動一個測試區。
5. 這樣做會觸發 IBM 測試自動化工具的自動測試程序檔。那些程序檔會執行一套自動化功能與整合的測試情境。
6. 獲得結果之後會交給品管團隊進行研究。雲端測試伺服器的執行個體之後便會停用。

支援核心功能與整合作業的測試如今在分散式測試環境中的執行速度比大型主機更快 50%。

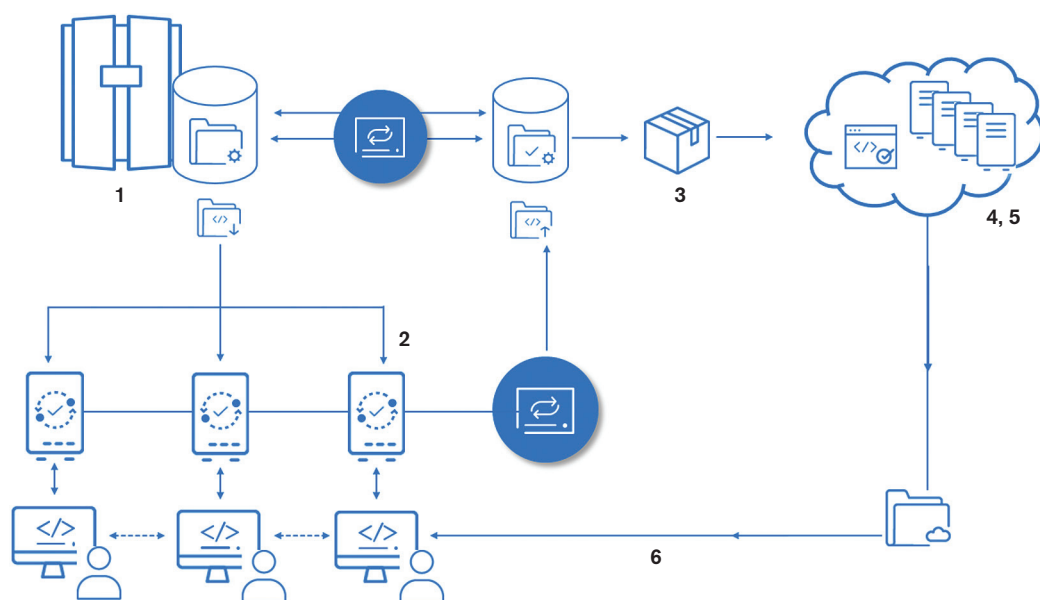


圖 6

支援核心功能與整合作業的測試如今在分散式測試環境中的執行速度比大型主機更快 50%。

Enterprise Test Server 提供簡單佈建且「按需供應」的測試環境，不會耗用珍貴的大型主機資源。有需要時測試環境隨時可用，這樣的彈性和延展性正是實現大型主機 DevOps 所需要的。

成果

- 新測試區的自動佈建作業從 6 週縮減為 2 小時，而對大型主機測試產能的需求減少了，對開支亦有正面影響。
- 支援核心功能與整合作業的測試如今在分散式測試環境中的執行速度比大型主機更快 50%。
- 按需求執行的測試能更快獲得結果回饋並減少尋找和修正問題的時間，讓品管團隊更有信心。
- 現在，測試伺服器的產能已有擴充，可在 Amazon Web Services (AWS) 執行個體上執行 35 個虛擬大型主機測試環境。新增更多測試環境的佈建作業僅需 2.5 小時，其中每個環境包含超過 30 個應用程式。
- 大型主機邁向 DevOps 的下一個漸進式步驟是停用大型主機上的整合與測試區，讓這些珍貴的資源可以貢獻給生產工作負載。

模擬大型主機環境以利提早測試

當平行開發開始擴充規模，企業也必須迅速測試新功能。

增加大型主機的測試產能可能頗有難度；通常測試資源的數量有限，而生產工作負載會優先使用。將早期測試週期卸載到低成本的商用平台，可解決大型主機的資源瓶頸並加速應用程式交付週期。

Enterprise Test Server 提供簡單佈建且「按需供應」的測試環境，不會耗用珍貴的大型主機資源。有需要時測試環境隨時可用，這樣的彈性和延展性正是實現大型主機 DevOps 所需要的。這家採用多種技術 (包括 COBOL、IMS DB/DC、Assembler、DB2、MQ 和 CA Endevor) 的英國跨國銀行，就是這項優點的見證者。

挑戰

所有英國銀行皆需遵循「結構改善方案」(Structural Reform Programme)，這是 2014 年開始變更實施的法規之一。為了滿足其中的特定要求，應用程式開發與維護必須更加「敏捷」，也意味著應用程式測試必須更頻繁。

非功能性測試團隊存取大型主機測試區的機會極少。當他們要解決產能問題以及達成企業對上市時程的要求時，就得在測試週期中採用自訂化、簡略的例行測試程序。可惜的是這些暫代性例行程序提高了生產失敗的風險。

解決方案

解決方案是建立持續同步化程序以及組成大型主機應用程式測試環境的大型主機元件之「鏡像複製」。該鏡射儲存庫包含大型主機原始碼、資料庫中繼資料、資料和基礎架構組態，整個軟體供應鏈上所有相關的層級全部複製下來。

該鏡射儲存庫包含大型主機原始碼、資料庫中繼資料、資料和基礎架構組態，整個軟體供應鏈上所有相關的層級全部複製下來。

鏡射儲存庫儲存在 Enterprise Developer 上，而這是支撐一個全自動測試環境的 Windows 主機型組建伺服器。

Jenkins 是基於 Java、可獨立運作的持續組建工具，可控制測試環境的佈建流程。透過 Jenkins 佈建設定檔入口網站輸入的參數會套用到相關的基礎架構範本。所產生的組態檔案定義了在分散式架構上執行的大型主機模擬測試環境。使用 Micro Focus 組建伺服器執行自動化組建程序來生產和部署供測試使用的可執行檔。

接著使用 Enterprise Developer 的資料庫工具從鏡射基線匯入資料，無論是 VSAM 檔案、IBM DB2 或 IMS 資料庫。

最後在 Enterprise Test Server 之下啟動一個大型主機執行個體並傳送通知到測試團隊，表示應用程式測試環境已準備就緒。該銀行發現，測試環境也可以指定用途，僅針對接受測試的使用案例所需要的可執行檔和資料。Enterprise Sync 在持續同步化程序中，從 Enterprise Analyzer 所創建和維護的儲存庫中擷取出應用程式智慧並套用，即可完成指定目的測試。

該解決方案為大型主機應用程式開發團隊奠下了採納 DevOps 實務的基礎，包括敏捷式開發、持續整合和持續部署，才能應付企業應用程式開發對於速度的要求。

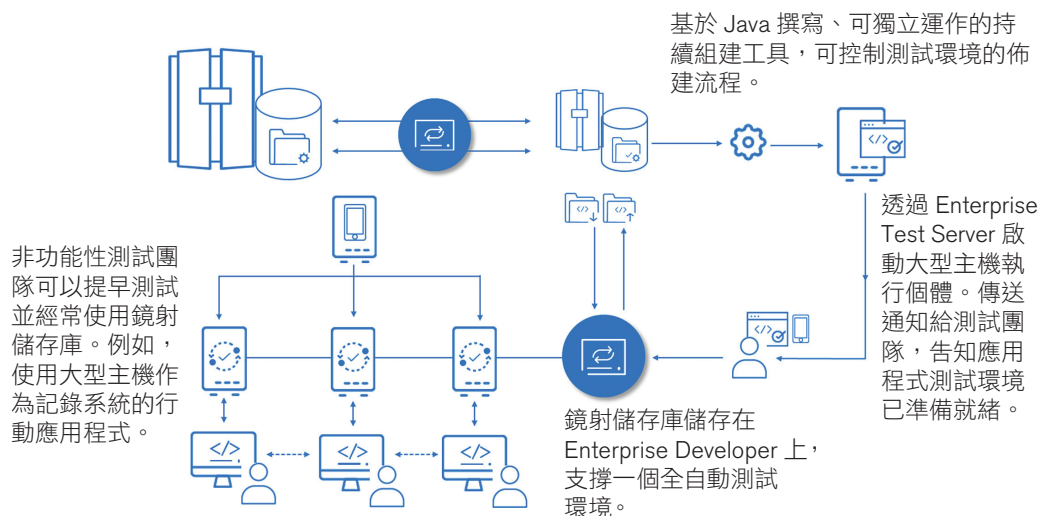


圖 7

大型主機元件的鏡射測試環境包含大型主機原始碼、資料庫中繼資料、資料和基礎架構組態，整個軟體供應鏈上所有相關的層級全部複製下來。

成果

- 非功能性測試團隊可利用按需提供的測試環境加速測試前端業務應用程式，因為他們不再受限於大型主機測試產能資源的問題。
- 因為可以擴充 Enterprise Test Server 來滿足測試需求，所以按需提供的測試環境取代了簡略的例行測試程序。測試品質因此有了改善，減少了應用程式交付週期的後期測試階段中所發現的瑕疵數量。
- 該解決方案為大型主機應用程式開發團隊奠下了採納 DevOps 實務的基礎，包括敏捷式開發、持續整合和持續部署，才能應付企業應用程式開發對於速度的要求。

能夠達成 DevOps 層級的彈性

由於大型主機會把資源優先分派給生產工作負載，所以大型主機鮮少把產能專門保留給開發和測試使用。大型主機環境增加新產能來支援 DevOps 層級的開發與測試實務會增加企業的成本，也有可能帶來另一項瓶頸。大型主機環境適用的低風險、分散式測試工具可以幫助有產能限制的大型主機團隊改善應用程式交付的彈性並維持品質。

DevOps 層級的彈性：自我評估

- 您是否可以輕鬆擴充開發和測試產能，同時控制大型主機成本？
- 大型主機資源可用性是否僅允許您在一日中的特定時間或在一週的固定期限內測試程式碼變更？
- 您是否有意運用虛擬化或雲端環境中自動化測試佈建的優勢？

讓 IT 能夠迅速應變同時保留核心業務流程與資料，不但是雙贏的策略，在財務上也是務實的做法。

總結

儘管大型主機的時間、測試資源、人員皆不足，組織還要抑制大型主機的成本，開發人員卻被要求完成更多任務並承受著巨大壓力。縮短上市時程是改善交付問題的重大優勢，尤其是在企業領導人認定大型主機僵化又過時的時候。

讓 IT 能夠迅速應變同時保留核心業務流程與資料，不但是雙贏的策略，在財務上也是務實的做法。其他替代方案，例如重新撰寫應用程式，不但成本高又曠日費時，隨之而來的大型主機智慧財產損失更會減損企業交付獨特服務的能力。大家傾向於選擇讓大型主機加速運作。這會需要更高的彈性、跨團隊的互動參與，而現代化的整合式工具更是不可或缺。

雖然本文所提及的客戶各有不同的設置方式，但是他們分別在效率、協同作業和彈性方面獲得大幅度的改善，或是三者皆獲改善。他們有充分的理由宣稱已達成大型主機 DevOps，或確定效能提升已達相近的水準。

要為您的企業物色最合適的方案，必須先找出加速交付可以創造最高業務價值的領域。對此您的組織與市場上的其他對手可能有不同的看法。可能是利用有效的開發工具和流程。或者，您可能認為簡化新進團隊成員的入職流程是務必達成的策略目標。現今 IT 營運必須迅速因應各種變化，大部份 IT 部門都會同意這個看法。

一個執行效果良好的大型主機 DevOps 策略所省下的成本和提升的效率，大多數企業都能善用。

如同最近的 Forrester 白皮書所參照的內容顯示，數位轉型與大型主機 DevOps 的目標十分相近。謹慎考慮進行應用程式現代化的公司與評估自身流程及 IT 基礎架構以利採行大型主機 DevOps 的公司，將會採取相似的步驟。兩者皆可預期關鍵領域的效能會大幅改善。

我們將 Micro Focus 客戶追求大型主機 DevOps 層級之效率的經驗，概要整理後寫成這份白皮書。在追求改善協同作業、效率和彈性的過程中，您的策略優先目標是主要的驅力，其中包括我們提過的維持或增加市佔率、降低成本、確認 IT 能支援更廣大的商業目標，或是三者皆有。一個執行效果良好的大型主機 DevOps 策略所省下的成本和提升的效率，大多數企業都能善用。

但是每個人都站在相同的起跑點。價值分析服務¹⁰是我們提供的免費諮詢與審核服務，歡迎您來洽詢。我們透過這項服務來評估您的 IT 基礎架構是否適合採行大型主機 DevOps 並為您規劃應用程式現代化。我們的白皮書可以告訴您，從理論到現實之間，其實僅需克服一連串可達成的循序式步驟。

¹⁰ www.microfocus.com/promo/the-value-profile-service



台灣
+886 2 2376 0000

Micro Focus
總公司
英國
+44 (0) 1635 565200

www.microfocus.com

www.microfocus.com