

# HP パフォーマンス・エンジニアリング・ ベスト・プラクティス・シリーズ

パフォーマンス・エンジニアおよびパフォーマンス・マネージャ  
向け

---

パフォーマンス・センター・オブ・エクセレンス・  
ベスト・プラクティス

ドキュメント・リリース日：2010年11月（英語版）

ソフトウェア・リリース日：2010年11月（英語版）

## ご注意

### 保証

HP の製品およびサービスの保証は、当該製品およびサービスに付属する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、HP はいかなる責任も負いません。ここに記載する情報は、予告なしに変更されることがあります。

### 権利の制限

機密性のあるコンピュータ・ソフトウェアです。これらを所有、使用、または複製するには、HP からの有効な使用許諾が必要です。商用コンピュータ・ソフトウェア、コンピュータ・ソフトウェアに関する文書類、および商用アイテムの技術データは、FAR 12.211 および 12.212 の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

### 著作権について

© Copyright 2010 Hewlett-Packard Development Company, L.P.

### 商標について

Microsoft®, Windows® は、Microsoft Corporation の米国登録商標です。

Oracle® は、カリフォルニア州レッドウッド市の Oracle Corporation の米国登録商標です。

## ドキュメントの更新情報

このガイドの表紙には、次の識別情報が記載されています。

- ソフトウェアのバージョン番号は、ソフトウェアのバージョンを示します。
  - ピリオドの前にある番号は、メジャー・リリース番号です。
  - ピリオドのすぐ後にある番号は、マイナー・リリース番号です。
  - ピリオドの後にある 2 番目の番号は、マイナー・マイナー・リリース番号です。
- ドキュメント・リリース日は、ドキュメントが更新されるたびに変更されます。
- ソフトウェア・リリース日は、このバージョンのソフトウェアのリリース期日を表します。

最新の更新のチェック、またはご使用のドキュメントが最新版かどうかのご確認には、次のサイトをご利用ください。

**<http://support.openview.hp.com/selfsolve/manuals>**

このサイトを利用するには、HP パスポートへの登録とサインインが必要です。HP パスポート ID の取得登録は、次の Web サイトから行なうことができます。

**<http://h20229.www2.hp.com/passport-registration.html>** (英語サイト)

または、HP Passport のログイン・ページの [New users - please register] リンクをクリックします。適切な製品サポート・サービスをお申し込みいただいたお客様は、最新版をご入手いただけます。詳細については、HP の営業担当にお問い合わせください。

## サポート

HP ソフトウェア・サポート Web サイトを参照してください。

**<http://support.openview.hp.com>**

この Web サイトでは、窓口の情報、製品、サービス、HP ソフトウェアが提供するサポートの詳細情報を提供します。

HP ソフトウェアのオンライン・ソフトウェア・サポートでは、セルフソルブ機能を提供しています。お客様の業務の管理に必要な対話型の技術支援ツールに素早く効果的にアクセスいただけます。HP ソフトウェア・サポート・サイトのサポート範囲は次のとおりです。

- 技術情報の検索
- サポート・ケースとエンハンスメント要求の登録とトラッキング
- ソフトウェア・パッチのダウンロード
- サポート契約の管理
- HP サポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェア・カスタマとの意見交換
- ソフトウェア・トレーニングの検索と登録

一部を除き、サポートのご利用には、HP パスポート・ユーザとしてご登録の上、ログインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。サポート・アクセス・レベルの詳細に関しては、次のサイトを参照してください。

**[http://support.openview.hp.com/access\\_level.jsp](http://support.openview.hp.com/access_level.jsp)**

HP Passport ID を登録するには、次のサイトにアクセスしてください。

**<http://h20229.www2.hp.com/passport-registration.html>** (英語サイト)

# 目次

はじめに .....	8
HP センター・オブ・エクセレンスについて .....	8
対象となる読者 .....	10
前提条件 .....	10
本書の構成 .....	11
フィードバック .....	11
<b>1 パフォーマンス CoE とは .....</b>	<b>12</b>
パフォーマンス・エンジニアリングの重要性 .....	12
CoE を採用する理由 .....	15
パフォーマンス・テストが十分な効果を発揮できない理由 .....	15
早急な取り組みが求められる理由 .....	16
CoE とは .....	18
CoE のフェーズ .....	22
組織の成熟度 .....	22
手動テスト .....	23
プロジェクト・テスト .....	23
標準化 .....	24
集中化 .....	24
パフォーマンス・オーソリティ .....	25
<b>2 CoE の構築 .....</b>	<b>27</b>
CoE 構築への 2 つのアプローチ .....	29
トップダウン .....	29
ボトムアップ .....	30
成果 .....	31
組織評価キット .....	32

概念実証.....	35
コミュニケーション.....	37
ツールの標準化.....	40
<b>3 CoE の運用.....</b>	<b>42</b>
プロジェクト計画.....	44
パフォーマンス・テスト戦略.....	44
パフォーマンス・テスト計画.....	44
目的.....	45
対象範囲.....	46
アプローチ.....	46
前提条件.....	47
依存関係.....	47
リスク.....	47
スケジュール.....	47
テスト・ケース.....	48
結果.....	49
不具合.....	49
ステータス・レポート.....	50
テスト実行レポート.....	51
定期レポート.....	52
サマリ・レポート.....	52
<b>4 CoE を構成するメンバ.....</b>	<b>54</b>
役割.....	54
CoE マネージャ.....	55
ソリューション設計者.....	56
テスト・プラクティス・リーダー.....	57
テスト・オートメーション・エンジニア.....	57
ビジネス・アナリスト.....	58
オペレーション・マネージャ.....	58
テスト・ラボ・スペシャリスト.....	59
負荷テストのスキル.....	60
アプリケーションに関する知識.....	62

5 結論 .....	64
A PCoE の質問.....	66

# はじめに

## はじめに

HP パフォーマンス・センター・オブ・エクセレンス・ベスト・プラクティスに  
ようこそ。

本書では、さまざまな組織に Performance Centers of Excellence を導入する際の概  
念、ガイドライン、事例を紹介します。

## HP センター・オブ・エクセレンスについて

HP は、自動パフォーマンス・テスト部門のリーダーです。自動化されたパ  
フォーマンス・テストでは、アプリケーション、アップグレード、パッチのデプ  
ロイメントに伴うリスクを軽減することを目的に、製品、人材、プロセスを活用  
します。特に、運用環境の作業負荷を事前にデプロイしたシステムに適用し、シ  
ステム・パフォーマンスとエンド・ユーザの作業状態を測定することを主な目的  
とします。パフォーマンス・テストを綿密に計画することによって、次のような  
点を解明できます。

- アプリケーションの応答時間は、対象ユーザが期待するレベルを満たし  
ているか。
- アプリケーションは、ユーザが想定するレベル、またはそれ以上のレベ  
ルの負荷に対応できるか。
- アプリケーションは、業務で求められるトランザクション数を処理でき  
るか。
- アプリケーションは、想定または想定しないユーザ負荷下で安定稼働で  
きるか。
- 運用開始日からすぐに、ユーザが満足するパフォーマンスを発揮できる  
か。

上記のような項目を明確に把握する方法として、自動パフォーマンス・テストで  
は業務上の条件の変化が及ぼす影響を数値化し、デプロイメントに伴うリスクを  
予測します。自動パフォーマンス・テストを効果的に行うことによって、豊富な

データに基づいてリリースの可否を判断し、システム・ダウンタイムや障害を回避することができます。

自動パフォーマンス・テストを最大限に活用する方法の1つが、センター・オブ・エクセレンスの社内構築です。センター・オブ・エクセレンス (CoE) モデルは、IT の運用を着実かつ継続的に改善する実践的な方法として、多くの IT 組織で採用されています。CoE モデルは、QA インフラストラクチャ、プロセス、ベスト・プラクティスを1つに集約したものです。CoE は、アプリケーションの管理および自動化のプラットフォームとなるだけでなく、アプリケーション品質を最大限に高める上で必要になるリーダーシップ、コンサルティング、サポートなどのサービス提供でも役立ちます。

また、CoE によって、組織内の誰もが品質に関する標準メトリックスと主要業績評価指標 (KPI) を参照することが可能になります。つまり、すべての関係者が必要な情報をいつでも取得できるようになり、ビジネス目標とアプリケーションの連携が実現されます。

**HP Performance Center** は、パフォーマンス・テストを自動化するツールであり、パフォーマンス CoE をそのまま実装することを可能にします。ユーザ組織全体をカバーするグローバルなクロス・エンタープライズ対応のパフォーマンス・テスト・ツールとして、ユーザ組織のインフラストラクチャにインストールされます。

- 地理的に離れた場所で負荷テストを実行したい場合でも、**Performance Center** を使用すれば、複数の負荷テスト・プロジェクトを同時に実行でき、テスト担当者が実際に足を運ぶ必要がなくなります。
- **Performance Center** は、負荷テストに対する内部のニーズを全て管理します。
- **Performance Center** には、リソースの割り当てやスケジュール設定など、大規模な負荷テスト・プロジェクトで必要になる機能がすべて含まれており、Web 経由での集中管理が可能です。
- **Performance Center** は、テスト・プロセスの合理化、リソース・コストの削減、操作効率の向上に役立ちます。
- **Performance Center** により、パフォーマンス・ボトルネックを特定できます。
- **Performance Center** により、テスト対象アプリケーションが対応可能なユーザ数を特定できます。これらの情報は、アプリケーションの負荷許容量を拡大する上で参考になります。

本書は、IT 組織の現状を把握し、社内でパフォーマンス・センター・オブ・エクセレンスを構築および管理するためのガイドブックです。本書のガイドラインは、実情に即したベスト・プラクティス・データの調査結果と、次に示す専門知識をベースに作成されています。

- HP オペレーティング・システム管理者
- HP プロフェッショナル・サービス組織
- 業界の専門家が作成した文書と書籍
- 多数のカスタマ・テスト組織

本書を参考にすることで、最初の構築プロセスを効率的に開始し、CoE の価値を最大限に引き出すことができます。

## 対象となる読者

本書は、次のユーザを対象としています。

- パフォーマンス CoE マネージャ
- パフォーマンス・エンジニア
- QA マネージャ

## 前提条件

本書を利用するためには、まずパフォーマンス・エンジニアまたは開発者の視点からパフォーマンスの評価対象を理解する必要があります。ソフトウェア開発ライフサイクル (SDLC) の主要フェーズに関する知識があるとよいでしょう。また、実際の IT 組織で実行される一般的なビジネス・プロセスに関する知識も必要です。

## 本書の構成

本書は、次の章で構成されています。

- はじめに
- パフォーマンス CoE とは
- CoE の構築
- CoE の運用
- CoE を構成するメンバ
- 結論
- 付録 – PCoE に関する質問

注：本書では、CoE はパフォーマンス・センター・オブ・エクセレンスを意味します。したがって、本書では CoE と PCoE の両方が使用されています。

## フィードバック

ご質問、コメント、効果的なベスト・プラクティスを次の電子メール・アドレスにお寄せください。

[lt\\_cust\\_feedback@hp.com](mailto:lt_cust_feedback@hp.com)

# 1 パフォーマンス CoE とは

パフォーマンス・センター・オブ・エクセレンスとは、成熟レベルを高めるためのアプローチであると同時に、その成果として誕生した IT 組織でもあります。パフォーマンス・テストにはさまざまな方法がありますが、手順に基づいて体系化および統合化することによって、従来よりも短い時間でさらに大きなビジネス成果を実現できることがわかっています。

## パフォーマンス・エンジニアリングの重要性

ソフトウェア・エンジニアリングは、ソフトウェア開発プロセスを管理する手順、手法、ツールであり、生産的な方法で高品質のソフトウェアを構築するための基盤となります。「ソフトウェア品質」には、機能性、使いやすさ、柔軟性、スケーラビリティ、セキュリティ、パフォーマンスなど、さまざまな評価基準があります。ほとんどのソフトウェア・エンジニアリング手法は、ソフトウェアが機能要件を満たしていることと、厳しいスケジュールと予算の範囲内で開発を行うことに重点を置いています。

現在、パフォーマンス要件は一段と管理が難しくなっている一方で、必要性も高まっています。しかし、分散システム環境にクライアント/サーバ型のソフトウェア・アーキテクチャを採用する動きが進むにつれて、従来のキューイング理論に基づく分析では対応しきれなくなっています。この傾向は、インターネット・サービスの登場でさらに加速しています。これは、インターネット・サービスでは「トランザクション」ベースではなく、「サブスクリプション」ベースで情報がユーザに送信されるためです。また、このようなソフトウェアとコンピュータ・システムの質の変化に伴って、ユーザ・コミュニティも変化しています。

多くの場合、システム・パフォーマンスは、システム・ライフサイクルのかなり後まで評価されません。最悪の場合、ユーザや顧客から応答速度に関する苦情が出て、やっとパフォーマンスの問題に気付くこともあります。これよりも若干よいケースであれば、デプロイメントの前にベンチマークが行われます。ただし、デプロイメントの直前なので、アプリケーションの設計、コーディング、テストはすべて完了しています。その結果、コンピューティングや通信のリソース・コストが増大してしまいます。

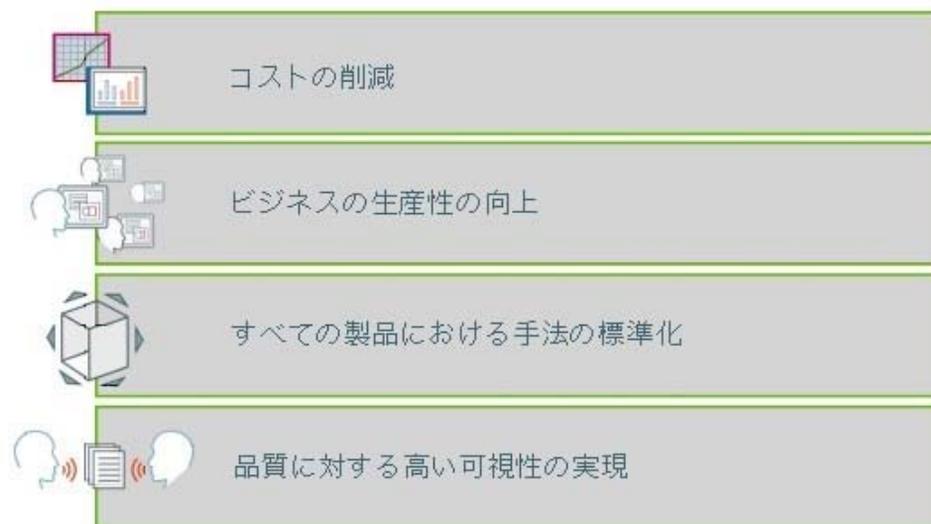
品質の高いアプリケーションが求められる一方で、ますます厳しくなるスケジュールでの開発とデプロイメント、組織の地理的な分散、アウトソーシング、スキルの高い人材の離職率の増加などにより、アプリケーションのテストはさらに難しい課題になっています。1つの事業単位に適切なスキル・セットがあったとしても、それが組織全体に波及することはありません。多くの企業のIT部門は、スキルを共有できず、共通のプロセスがなく、パフォーマンスを把握できないという状況に陥っています。

増大を続ける作業に限られた時間と予算内で対応し、複数のプロジェクトを同時進行しながら、地理的に分散したプロジェクト・チームを管理しなければならない状況の中で、多くの組織は、テスト管理手法と自動化されたテスト管理ツールを使って、テスト作業の一元管理、体系化、優先度の設定、ドキュメント作成を行っています。

本書では、このようなテスト管理を行う上での課題と利点を説明し、体系化されたテスト・プロセスを導入する実践的な方法を紹介します。

主要な顧客やアナリストが採用してきたアプローチの1つがパフォーマンス・センター・オブ・エクセレンス (CoE) モデルです。これは、品質、一貫性、効率、効果を高めることを目的にしたプロセスと標準化を、組織全体で推し進める論理的または物理的なエンティティを指します。

## CoE のビジネス価値



パフォーマンス CoE は、雑多なツールが使用されている、パフォーマンス・テストの効率が低い、分析結果の正確性に欠ける、といった問題を解決します。また、集中プラットフォームでパフォーマンスを一元管理することによって、アプ

リケーションの重要なパフォーマンス・パラメータを把握できる機能を提供し、誰もが必要な情報を取得でき、ビジネス目標に即したアプリケーションを提供できる環境を整備します。

## CoE を採用する理由

答えは簡単です。センター・オブ・エクセレンスが大きな価値をもたらすからです。組織の能力の大幅アップを図って設定目標またはそれ以上の成果につなげるという CoE の重要な特徴は、多数のアナリストのレポートで指摘されています。ガバナンス、サポート構造、ガイダンス、測定、スキルの共有がすでに実践されている組織では、CoE を推進しやすく、具体的なプロジェクト目標に注目することができます。CoE 構築の第一の目的は成果を上げることであり、これがなければ組織は潜在能力を最大限に発揮することはできません。

## パフォーマンス・テストが十分な効果を発揮できない理由

パフォーマンス・テストは特殊なスキル・セットであり、アプリケーション、ハードウェア、サードパーティ・システムに関する知識が必要です。具体的には、アーキテクチャの視点からアプリケーションの動作を理解し、モデリング・スキルと体系的なアプローチが求められます。すべてのテスト担当者がこのような条件を満たしているわけではなく、スキル・セットを完全に習得するには何年もかかります。また、テスト担当者がプロジェクトのテスト・グループごとに分かれて作業する環境では、すべてのアプリケーションを同じ基準をベースに確実にテストすることは難しくなります。さらに、新しいテクノロジーが登場した場合、グループによっては必要な専門知識がなく活用しきれない場合もあります。

ほとんどの場合、プロジェクト・チームがそれぞれ使いやすいツールを選び、独自の手順でパフォーマンス最適化を行っています。このような環境では、雑多なワークフロー、プラクティス、ツール、手法が混在することになり、文書化や統合も不十分なので、他のグループの状況を明確に把握することは不可能です。その結果、テスト効率が下がるだけでなく、コストが大幅に増大して手順が複雑になり、意欲の低下を招いたり、IT スタッフの能力開発の妨げにもなります。このように、開発およびデリバリのプロセスに含まれるそれぞれのサイロが独自のアプローチでパフォーマンスを最適化すると、コミュニケーションや連携に支障をきたし、最終的にはパフォーマンスを最適化できません。さらに、このように断片的な方法でテストを行うと、IT やビジネス管理の視点からテストを実施するという姿勢が失われ、プロジェクトの目的があいまいになってしまいます。

運用環境でのデプロイメントを計画する際に、多くの企業が開発プロセスの遅い段階でベンチマークを実施します。その結果、パフォーマンスや可用性の問題が解決されないまま、最適とはいえな状態でデプロイされることになります。このようなシステムのパフォーマンスを別のツールセットで改善しようとしてもうまくいきません。

## 早急な取り組みが求められる理由

次に示すように、アプリケーション・パフォーマンスは重要な要素であるにもかかわらず、責任を持って担当する部署がないという理由で見過ごされてきました。

非常に高度な機能設計が採用され、ビジネス要件を完全に満たしていても、パフォーマンスと信頼性が低ければ、エンド・ユーザにとっては役に立たないアプリケーションでしかありません。

残念ながら、ユーザの記憶に残るのは、最高のパフォーマンスを発揮したアプリケーションではなく、最悪のアプリケーションなのです。アプリケーションのパフォーマンスは、エンドユーザ・エクスペリエンスに直接的な影響を与え、結果として収益源に大きく影響することになります。したがって、アプリケーションのデプロイメントにおいて、パフォーマンス・テストは極めて重要なコンポーネントであり、アプリケーションが高い顧客満足を得られるか、それとも競合他社に乗り換えられてしまうのかを分ける鍵になります。

パフォーマンスに問題が発生すると、修正に時間と労力がかかるので、ユーザ・エクスペリエンスに大きな影響を与えます。パフォーマンスの問題は短時間で解決できるものばかりではなく、サービスを完全に停止しないと解決できないものもあります。

アプリケーションの処理速度と可用性を高める作業は複雑であり、長年の経験から培われた専門知識が必要です。また、SOA、仮想化、Web 2.0などの登場でプラットフォームは一段と複雑になっていることから、アプリケーションのパフォーマンス最適化に含まれるすべてのフェーズが非常に専門的になっています。このように複雑なプラットフォームでは、それぞれ独自のスキル・セット、ツール、フォーマット、プロトコル、プロセスが使用されます。その結果、プロジェクト・チームが独自のスキルとプロセスに基づいて負荷テスト、コードの最適化、使用率の分析、チューニング、キャパシティ・プランニングを行うことになり、IT部門内や事業部門で共有されることはありません。

さらに、企業をとりまく環境は、事業予算やIT予算の削減、企業分割、合併、買収、アウトソーシングなど変貌を続けており、状況はさらに深刻です。現代の企業は、リソースと資金をばらまいて現状維持を選択しながら、「従来通りのビジネス」というイメージを与え続けることはできません。

インターネットの普及と、エンド・ユーザの間で Web アプリケーションのパフォーマンスへの理解が深まったことを背景に、ユーザ・エクスペリエンスの向上は IT 部門が取り組むべき非常に大きな課題になっています。トランザクションの複雑化にともなって処理時間も長くなるにもかかわらず、トランザクションの完了までユーザが許容できる時間は従来よりも短くなっています。その方法として、ユーザは SLA 準拠を厳格に満たすことを要求します。通常、SLA では期待されるパフォーマンス・レベルが規定されており、これを達成しないと顧客満足度が低下し、ユーザがアプリケーションの使用を停止する事態にもなりかねません。

このようなリスクを軽減するのがパフォーマンス・テストであり、アプリケーションとアーキテクチャの処理能力を測定し、期待されるレベルのパフォーマンスを発揮できるかどうかを確認してからアプリケーションをユーザ・コミュニティに提供します。したがって、パフォーマンス・テストはオプションで実施するものではなく、必須の手順です。

## CoE とは

「センター・オブ・エクセレンス」という用語は必ずしも一般的な用語というわけではなく、企業によっては、「コンピテンシー・センター」、「セントラル・テスト」、「アプリケーション・テスト・センター」など、また単に「テスト・グループ」という用語を使用しています。その一方で、「センター・オブ・エクセレンス」という用語は、パフォーマンス・テストの分野だけでなくビジネス業界では非常によく使われています。以上を踏まえて、本書でも CoE という用語を使用します。

業界調査機関である **voke** が実施した調査において、PCoE は次のように定義されています。

パフォーマンス CoE とは、ビジネス要件を満たすためにアプリケーションのパフォーマンスを向上および検証することを目的とした社内のコンピテンシー・センターです。パフォーマンス CoE は、パフォーマンス最適化プロセス向けの管理および自動化プラットフォームに加えて、パフォーマンスの検証と最適化で CoE モデルが果たす重要な役割についての理解を深めるためのコンサルティング・サービス、サポート・サービス、リーダーシップ、提案を提供します。プロジェクト・チームは、パフォーマンス CoE を通じて、CoE で構築された専門知識、ツールセット、ベスト・プラクティスを活用できます。

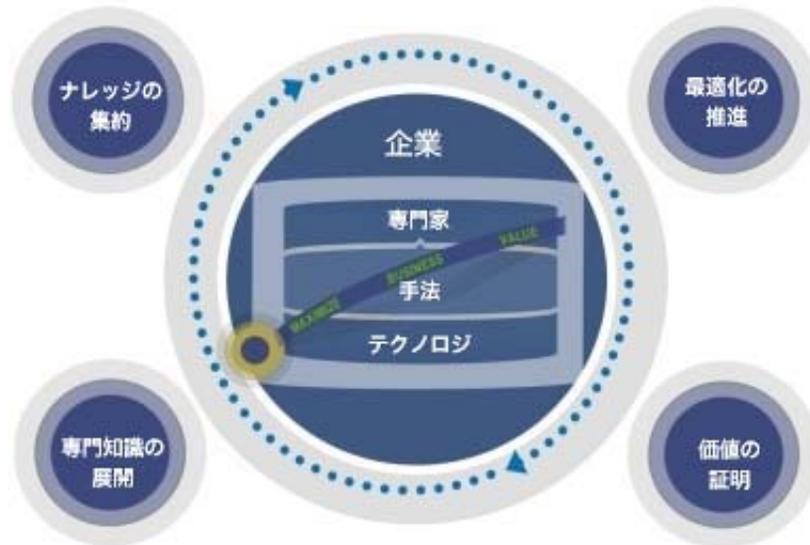
CoE モデルには、次のような利点があります。

- インフラストラクチャと専門知識の集約  
テスト・ラボ、ツール、スタッフ、ベスト・プラクティスを 1 つに統合することで、すべてのプロジェクト・チームが簡単に活用できる環境を整え、高価なリソースの重複を排除します（実際、スタッフの人数とシステム・ライセンスを低減できます）。構築される知識が増えるほど、組織全体のメリットも大きくなります。また、テスト品質もさらに高くなり、テストの実行と分析を効率化できます。
- プロセスにかかる時間の短縮と品質の向上  
ツールセットとプロセスの標準化は、プロセスの一貫性の強化、コスト効率アップ、実装にかかる時間の短縮、品質の向上をもたらします。スクリプト、モニタ・プロファイル、シナリオを個々のテスト・マシンやテスト・グループに分散するのではなく、すべてのデータを一元管理することによって、共有や再利用が簡単になります。また、テストの作成にかかる時間も短縮でき、テスト作業も効率化できます。
- ビジネス中心の視点  
パフォーマンス CoE モデルは、システムやコンポーネントの視点だけでなく、ビジネスとエンド・ユーザの視点からパフォーマンスを測定します。テ

スト・システムの集中化によって、マシンあたりのダウンタイム短縮や24時間365日のテスト体制など、システムをフル活用できます。

- 実現可能な目標

パフォーマンス CoE を構築することは、実現可能な目標を設定することにつながります。既存のリソースをベースに達成できる小さな目標から始めて、成果を確認しながら段階的に高い目標を設定していきます。経営陣や IT スタッフによって CoE の価値が認知されてきたら、リソース、サービス、機能を順次拡張していきます。CoE は、正しい方法で実装すれば必ず成果を上げることができます。



一般的に、パフォーマンス CoE は、総運用コスト (TCO) を低減しながらパフォーマンスと可用性を高めることができ、人材を最大限に活用します。

CIO からよく出る質問の 1 つに、「パフォーマンス CoE に移行することによって ROI はどのような影響を受けるか」というものがあります。アプリケーション・デリバリ全体を効率化する上で、パフォーマンス・テストはボトルネックだと考えられがちですが、重要なのは、テスト組織の初期状態によってどのような ROI も変動するということです。

CoE の要素やテスト・グループの統合が効率化に貢献するのかどうか、という点は議論の余地があるところですが、CoE による改善点の多くは組織構造の変更だけに起因するものではなく、ベスト・プラクティスの蓄積や組織全体の行動の変化をもたらす成果です。

テストの目的は問題点を見つけ出すことです。したがって、「プロジェクトの最後になってバグが見つかるから」という理由でテストをボトルネックだとみなすのは誤りです。たとえば、アプリケーション要件の不備が原因でバグが発生している場合には、パフォーマンス CoE では解決できません。

どのような組織でも、パフォーマンス CoE を構築するのに遅すぎるということはありません。本書は、テストを集中統合していない組織や、このような手法をまだ導入していない組織を対象に作成されています。

本書では、組織の現状を把握し、目標を設定してから、それを達成するために必要な手順を実装します。

## CoE のフェーズ

パフォーマンス CoE モデルは、簡単に実現できる実用的なモデルです。投資を最小限に抑えた小規模なシステムからスタートし、段階的に投資を増やしていくことができます。そして経営陣や IT スタッフによって CoE の価値が認知されてきたら、リソース、サービス、機能を順次拡張していきます。CoE は、正しい方法で実装すれば必ず目に見える成果を達成することができます。

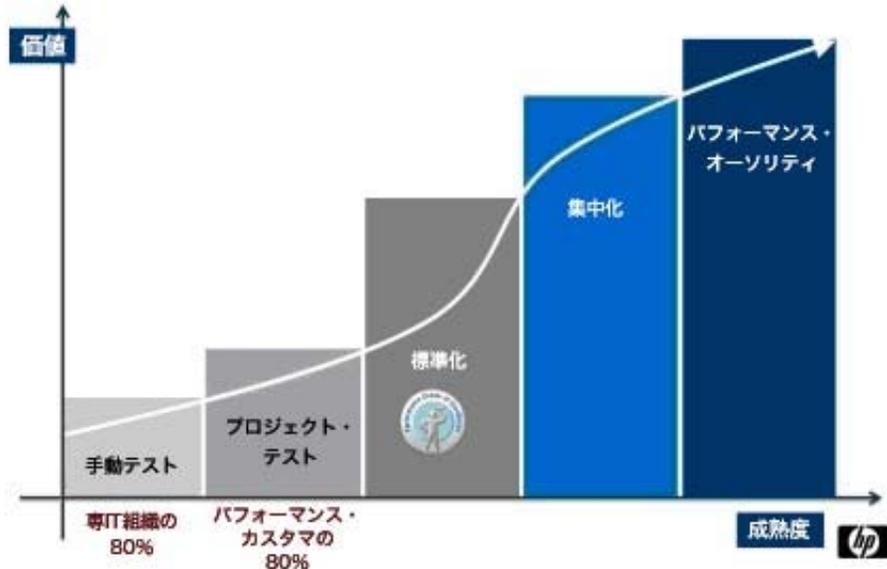
企業がパフォーマンス CoE 構築を検討するきっかけになるのは、パフォーマンスに重大な問題が発生した場合、プロジェクトで問題が発生している場合、コスト削減が必要な場合などがあり、ほとんどは部門レベルの規模での導入が検討されます。たとえば、テスト対象アプリケーションがますます複雑になり、専門知識で対応できる範囲が縮小していく状況に対応する手段として、パフォーマンス・スペシャリストが CoE の検討を始めるケースや、ハードウェア統合やライセンスの再利用などビジネス上の要因や、応答時間が遅いという事業部門からの苦情がきっかけになり、テスト・グループの管理部門が CoE 構築を指示するケースがあります。

このように CoE が開始される理由はさまざまですが、CoE はハードウェアおよびソフトウェア資産の統合からパフォーマンス・オーソリティまで、フェーズごとに進んでいきます。各アプリケーションが運用環境へと移行するまでの間、フェーズを追って専門スキルの構築と共有が進み、ベスト・プラクティスがドキュメント化され、それに従ってテストが実施されます。

## 組織の成熟度

IT 企業には、同じものは1つとしてありません。ただし、パフォーマンス・テストという分野で考えれば、どの企業にも管理方法にいくつかの共通点があります。成熟度の点で、ほとんどの企業が同じパスを進んでいきます。高い成熟度に到達する企業もありますが、労力をあまり必要としない快適なレベルで止まってしまう企業もあります。PCoE は継続的に構築を続けるプロセスなので、企業の現在の成熟度を理解し、テスト・グループがどの位置にあるかを評価することが重要です。生産性を最大限に引き出すという目標に向かって取り組みを始めたばかりのチームでも、現状を評価することによって、プロセスの全体像をつかみ、それを元に次のステップのプランニングを行うことができます。

## パフォーマンス・テスト・チームの進展



では次に、組織の成熟度と、CoE での位置付けについて詳しく説明します。

### 手動テスト

これは、最も基本的なタイプのテストです。このタイプのテストでは自動化は行われず、アプリケーション・パフォーマンスは単発的に検証されます。ビジネスに対するアカウントビリティはなく、運用環境ではシステム・パフォーマンスに問題が発生しています。本格的なパフォーマンス・ツールを使用しない IT 組織やテストがビジネスに及ぼす成果を測定しない IT 組織の大部分がこの状態になります。

### プロジェクト・テスト

このタイプのテストは、テストの本格的な自動化に向けた最初のステップになります。一般的に、部門別に特化したアプリケーションを、それぞれの部門がテストします。各部門は独自にツールとライセンスを購入し、テスト用のインフラストラクチャを調達します。部門がそれぞれ専門の分野を担当することもあります。たとえば、J2EE アプリケーション専門の部門や、常駐データベース専門の部門などがありますが、このような専門知識が部門間で共有されることはありません。ほとんどの場合、ビジネス・ユーザと直接的

なつながりを持っていない開発者がテストを実施するため、テストの現場では、ビジネス目標やアプリケーションの重大度は詳細に把握されていません。

このような状態でテストを行うと、次のような結果になります。

- インフラストラクチャやテスト・ツールのライセンスを再利用できない
- 人材やスキルを再利用できない
- プラクティスとビジネス目標の整合性がとれない

## 標準化

本格的なパフォーマンス CoE へと続くフェーズです。組織がハードウェア資産とソフトウェア・ライセンスを集約しようとするのは、パフォーマンス・テスト用に運用環境に近い環境を構築および保持するためには非常に大きなコストがかかることがわかっているためです。業務部門間でテスト・ラボが重複していると、このコストはさらに大きくなります。一方で、運用環境を的確に反映したパフォーマンス・テスト環境がないと、致命的な結果につながります。

パフォーマンス CoE は、パフォーマンス・テストで使用するソフトウェアとハードウェアの統合から始めて、資産の有効活用と管理および保守コストの低減からコスト削減へとつなげます。このフェーズで CoE は、部門別のサービス組織から、業務部門の枠組みを超えてインフラストラクチャ管理とユーザ・サポートを担うリソースへと転換します。

この時点の組織では、パフォーマンス最適化エンジニアのスキルセットの開発と知識共有の重要性はまだ認識されていません。パフォーマンス CoE モデルの利点がプロジェクト・チームと経営陣から認知されるに従い、CoE はサービス対象をパフォーマンスへと徐々に広げていくことができます。

## 集中化

このフェーズでは、パフォーマンス・テスト・チームがソフトウェア開発およびデプロイメント・ライフサイクルのさまざまな段階で深く関与していきます。

このフェーズでは、パフォーマンス CoE チームは、プランニング、スクリプトの記述、テストの実行など、実際のパフォーマンス・テスト・サービスの提供を始めます。これまでの事業部門はパフォーマンス・テスト、業界のベスト・プラクティスおよびプロセスについては限られた知識しか持っていませんが、CoE モデルを導入することによって、パフォーマンスの専門家の経験や提案を活用することができます。

このフェーズはサービス・ユティリティ・モデルと呼ばれることもあり、CoEを導入した多くの組織では広く使用されています。CoEの構築は、パフォーマンス・エンジニアリングに関する専門的で貴重な知識を育成するチャンスです。このフェーズでは、事業部門の枠組みを超えた協力体制、さまざまなテクノロジーを駆使した問題解決、アーキテクチャの連携に関する深い理解が可能になります。その結果、さらに有能な人材を引きつけることができ、定着率も高まります。

## パフォーマンス・オーソリティ

CoEは、成熟度が最も高いパフォーマンス・オーソリティへと進化します。CoEはアプリケーションの開発、デプロイメント、運用に欠かせない要素となり、パフォーマンスとコスト効率を重視する企業カルチャーの形成に貢献します。パフォーマンス・オーソリティ・モデルでは、すべてのアプリケーションが、パフォーマンス向上を主眼とする一貫した包括的なプロセスを経てから、運用環境へと移行します。

パフォーマンス・オーソリティには、次のような基本原則があります。その一部は、CoE構築の早い段階で実施されます。

### — サポート

対象分野について、パフォーマンス CoE は事業部門をサポートします。具体的には、サービスや非技術分野の専門家を提供します。

### — ガイダンス

一般的なガイダンスのアプローチとして、標準、手法、ベスト・プラクティス、ツール、知識リポジトリなどを提供します。

### — 知識の共有

知識の共有を促進するために、トレーニングや認定、スキルの評価、チームの構築、役割の編成などを行います。

### — 測定

CoEは、出力メトリックスに基づいて、価値を実証し成果物の妥当性を示します。測定は、CIOとビジネスの可視性の基盤となります。

## — ガバナンス

リソース（資金や人材など）の有効活用と潜在能力の発揮は、**CoE**の重要な機能です。そのためには、最も大きな価値を生み出すプロジェクトに投資し、サービス提供に合わせて経済効果を拡張していく必要があります。さらに、**CoE**が効果を発揮するためには、企業内の他の関係者との調整も必要です。

このような基本原則を正しく実施することによって効果的な**CoE**構築が可能になり、全体的な顧客満足や、サービス・レベル・アグリーメント（**SLA**）の達成またはそれ以上の成果を実現することができます。さらにこれが**ROI**の向上へとつながり、パフォーマンス・オーソリティが広く認知されることとなります。

## 2 CoE の構築

本章では、パフォーマンス・センター・オブ・エクセレンスのプランニングと、現在の IT 組織内で CoE エンティティを構築する手順に関するベスト・プラクティスを紹介します。

パフォーマンス・テストは、アーキテクチャに負荷をかけた状態で全体的なパフォーマンスを測定するプロセスです。次の質問への回答から、パフォーマンス・テストで適用するリソースと、テスト組織の全体的なキャパシティが決まります。

- パフォーマンス・テストの実施頻度
- パフォーマンス・テストの品質基準の厳しさ
- パフォーマンスの問題が運用環境で発生した場合のリスクと損害の大きさ
- パフォーマンス・テストに投入できる技術リソースの量と頻度

以上の質問の回答から、アプリケーション・チームまたは会社全体がパフォーマンス・テストを継続して実施すべき重要なプロセスと考え、そのために組織内のリソースを投入する用意があることが確認できた場合には、パフォーマンス・センター・オブ・エクセレンス (PCoE) の構築を真剣に検討してください。

これまでに説明したように、パフォーマンス CoE モデルは、既存のリソースを活用して小規模からスタートでき、具体的なメリットをすぐに実現することができます。その後、ビジネスで成果を上げていることがプロジェクト・チームや経営陣から認知されるに従って、さらに高い成熟度へとサービスのレベルアップを図ることが可能です。

PCoE は、パフォーマンス・テストのベスト・プラクティスの推進、パフォーマンス・テスト・プロセスの定義と導入、運用チームでの展開、組織としての戦略的な方向性の決定をサポートします。

PCoE モデルは、パフォーマンス・スペシャリストのグループとパフォーマンス・テスト・コンポーネントをすべてのプロジェクトで活用する体制をサポートします。パフォーマンス・テスト・スペシャリストは、各プロジェクト・チームの教育とリソース補充を通じて、一貫性のある体制作りを行います。これにより、人員の配置変更を最小限にとどめながら、プロジェクトの認知徹底を実現します。

PCoE は、リソースの効率活用の方法として、管理とパフォーマンス・テストのベスト・プラクティスをプロジェクト全体で共有するための基礎作りを行います。PCoE は、テスト計画、テスト・ケース、障害レポートなどパフォーマンス・テストの成果物の再利用を推進します。また、プロジェクトごとに作成される成果物の保管庫を管理し、将来的なプロジェクトに対してフレームワークやサンプルとして提供します。

パフォーマンス CoE は、次のようなパフォーマンス目標の達成をサポートします。

- 新しいコード、新しいソフトウェア、新しいハードウェア構成を運用環境に導入する場合、容量の制限、同時実行の問題、パフォーマンス・ボトルネックなどを事前に把握できるので、安心して移行できます。
- アプリケーションのチューニングや全体的なリソース要件の縮小によって、既存のコードのパフォーマンスが向上し、全体的な応答時間を短縮できます。
- アーキテクチャのパフォーマンス・プロファイルやスケーラビリティの特徴を理解することを通じて、キャパシティ・プランニング/管理を効率化します。

## CoE 構築への 2 つのアプローチ

一般的な組織では、次の 2 つの方向からパフォーマンス CoE 構築の必要性が認識されます。

- トップダウン：経営陣の意思決定として CoE を構築
- ボトムアップ：テスト・コミュニティがテスト環境の統合を緊急課題として検討

HP が一部の顧客を対象に実施した調査および業界の調査データでは、トップダウンの方がボトムアップよりもはるかに多いことが示されています。

### トップダウン

トップダウン・アプローチでは、アプリケーション開発部門のバイス・プレジデントまたはそれに相当する職務担当者がテスト・グループに対して、テストのリソースと知識の育成および集約を目的に、プロセス指向のセンター構築に関する提案書の作成を依頼します。このようなイニシアチブは、ミッション・クリティカルなアプリケーションのパフォーマンスが低いという苦情が、CIO や CEO に近いビジネス・マネージャから何回か報告されたことがきっかけになるケースが多いといえます。また、顧客満足の向上、蓄積された知識の活用、ハードウェアおよびソフトウェア・リソースの ROI の向上など、CoE が発揮できる効果を社外コンサルタントが提案するケースもあります。さらに、CIO またはアプリケーション部門 VP がパフォーマンス・ツール・ベンダから CoE の情報を得るケースや、業界エキスポ・イベントで CoE 構築による成功事例を紹介されるケースもあります。

いずれの場合も経営陣は、パフォーマンス CoE の目的、最初の実施対象、運用開始予定日を明確に示す必要があります。まず最初に、イニシアチブのミッション・ステートメントを定義することをお勧めします。これまでのミッション・ステートメントといえ、コンサルタントが作成を支援するものも含めて実体のないものばかりでしたが、ここでは役に立つミッション・ステートメントを作成します。そのためには、計画と実践を対応付ける必要があります。日々の業務で行う作業をテスト・チームのミッションとして考えます。

テスト・グループは、この情報を元に、作業内容、スケジュール、レポートの手順、予測される問題点、期待される利点などの詳細情報をまとめた提案書を作成します。提案書の作成、検討、導入のプロセスに対する経営陣の関心の高さが、プロジェクトの成否を決める鍵になります。

## ボトムアップ

一部の組織では、実力を持つテスト・グループが、開発チーム間での連携不足、業務に関する知識不足、貴重なハードウェアやソフトウェア・リソースが効率的に活用されていない状態について問題提起をし、パフォーマンス CoE の構築を提案することがあります。

テストの効率化や顧客満足の上昇が目標である点はトップダウン・アプローチと同じですが、CoE 構築へのアプローチに若干相違があります。通常のボトムアップのアプローチでは、ビジネス・ケースの作成、エグゼクティブ・スポンサーの獲得、承認のための条件設定という作業が伴います。

次に示すプロジェクト・フェーズの計画と実行が非常に重要です。

- 期待される価値の設定

CoE を構築することは、組織のビジネス目標を確認する作業です。組織内の主要な関係者との話し合いにおいて、CoE の対象範囲を決定します。

- 現在のテスト・モデルの確認

改善の効果は、ベースライン・モデルに基づいて評価されます。将来的にテスト・センターの効果を測定できるように、既存のテスト・プロセスを分析して文書化します。

- 導入計画の作成

対象範囲が決定したら、インフラストラクチャ要件、ツールセット、スタッフを含めた具体的な計画を作成します。テスト CoE を導入する際に適用する標準、ポリシー、手順を明確に定義します。

- CoE サービスのセットアップ

多くの事業部門（LOB）が品質保証にテスト CoE を活用できるようになります。

- 結果の評価

事前に定義しておいた期待値や目標に基づいて、CoE の成果を評価します。必要に応じて改善点を提案します。

- ベスト・プラクティスと手法

さまざまな知識分野でコア・コンピテンシーを構築し、標準化されたテンプレート、規格、ガイドライン、チェックリストとして提示します。

## 成果

ここで行う手順はすべて、パフォーマンス・センター・オブ・エクセレンスの構築というビジネス目標を明確に認識することを主な目的とします。具体的には、「パフォーマンスの検証にはどのような問題点があるか」という質問への回答を考えることによって、直接的なアクションの実施、変更対象の定義、進捗を測定するメトリックスの選択などを行います。

ここでは、PCoE の構築に成功したある HP ユーザが設定した目標を紹介します。

- ベスト・プラクティスの推進

負荷テストに関する知識とベスト・プラクティスを集約するグループを作ります。すべての IT で共通のフレームワークと手法を定義します。

- ツールの一元管理

すべてのアプリケーションの問題の連絡窓口を 1 つにまとめます。更新とパッチの適用も一元管理します。ライセンス管理を効率化します。

- ツール、リソース、機器の活用

人材、プロセス、テクノロジーを企業全体で共有することにより、コスト削減を推進します。

- 負荷テストの対象アプリケーションの増加

負荷テストの対象を広げます

- IT リスクの軽減

パフォーマンスに確信を得た状態で、アプリケーションを運用環境に移行します。

## 組織評価キット

ビジネス目標が明確になったら、評価フェーズに進みます。ここでは、パフォーマンス CoE の構築と前のステップで定義したビジネス目標の達成にあたり、組織の準備が整っているかどうかを査定します。また、パフォーマンス CoE が組織にもたらす効果のベースラインを、期待される ROI で算出します。

CoE を使いこなすレベルや、CoE が企業戦略において発揮する重要性は企業によって異なります。組織がどのような段階を経てパフォーマンス・センター・オブ・エクセレンスを構築するのかわかりやすく示しているのが成熟度モデルです。このようなモデルの中でも、最もよく知られているのが能力成熟度モデル（CMM または CMMI）でしょう。

CMM モデルでは、組織の成熟度を 5 つのレベルに分類していますが、このレベルは前の章で紹介したレベルと高い類似性があります。

- 初期（無秩序、正式でない、その場限り、大きな決断） - 新しいプロセスの使用を開始した時点です。本書の「手動テスト」に相当します。

通常の場合、初期段階では CoE の必要性は認識されていません。知識やスキルは、特定の事業部門や個人に内在しています。

- 反復可能（管理された状態、文書化、プロセスの統制） - プロセスが繰り返し活用されます。本書の「プロジェクト・テスト」に相当します。

組織は反復可能な段階へと移行し、CoE はプロジェクト・チームの資産として認識され始めます。プロジェクト・チームは、プロジェクト中心の視点に基づいて、サポートの必要性和、高いレベルのスキルを取得できるセンターの必要性を認識します。プロジェクトを超えた協力体制作り、計画と対象範囲の設定、開発を担当しているアプリケーションの監視を始めます。ただし、この段階では CoE は限られた成果しか達成できません。企業の大部分は、この成熟度以上のレベルに分類されます。

- 定義（規定，統合） - プロセスは，標準的なビジネス・プロセスとして定義/確認されています。本書の「標準化」に相当します。

CoE の次のステップでは，コンピテンシーを高めるための標準やプラクティスを定義および文書化します。この段階では，wiki 形式でベスト・プラクティスを集め，リスクと品質の積極的な管理を始めます。基本的なトレーニングの編成，ハードウェアおよびソフトウェア・リソースの統合，組織内にあるコンピテンシーの積極的な活用も行われます。成熟した CoE で求められる作業を一部実施せず，上位レベルに進まない組織も多数あります。最初のいくつかの成熟度レベルで満足するケースがほとんどであり，さらに上位のレベルへと進むためには強力な協力体制，つまり経営陣のレベルでの強力なコミットメントが必要になります。

- 管理（戦略的，定量化） - ベスト・プラクティスが共有され，プロセスの管理や測定が実施されます。本書の「一元管理」に相当します。

戦略的で管理された CoE を構築するためには，経営陣の支援が必要です。この段階では，個々の部門ではなく組織全体へと視点が広がっています。この段階のテスト組織は，全社的な目標や計画のサポート，企業スコア・カードとの統合，サービスを利用する組織全体でのイニシアチブの積極的な管理を開始します。正式なキャリア・パスが多数提供され，CoE がサポートするコンピテンシーで能力開発や教育が提供されると，パフォーマンス CoE は真価を発揮します。

- 最適化（継続的な改善） - プロセスを継続的に最適化/改善します。本書の「パフォーマンス・オーソリティ」に相当します。

最適化された CoE は，ますます大きな価値を企業に提供します。このレベルでは，CoE は組織全体で認知されています。CoE のメンバは企業目標を自分の目標として取り組み，このような目標達成を目指すプロジェクトを確実に成功へと導きます。評価に基づいて，目標達成に必要なコンピテンシーを活用し，着実に向上します。CoE は，企業スコアカードとその成果に責任を持ちます。監査，スキル評価，認定は，さらにこのような能力を強化します。

以上の内容を踏まえて、テスト組織の成熟度と現状を測定するには、どのような方法を使えばよいでしょうか。ここでは、最初の手順として、現在のソフトウェア・エンジニアリングと管理プラクティスの査定をお勧めします。この査定を行うことにより、ソフトウェア・テストのプラクティスの長所と短所を詳細に示すレポートを作成できます。この査定では、次の作業を行います。

- 経営陣によるディスカッション
- 質問（付録 A を参照してください）
- フィードバック
- 構造化されたインタビュー
- 分析
- アクション・プラン

会社のビジネス目標とアプリケーションのパフォーマンス目標の整合性は、CoE 構築では重要な役割を果たします。IT 管理部門は、ビジネス目標が相互に与える影響を理解することで、目標の達成度を測定するためにはどのような KPI セットが必要なかを特定できます。このメトリックスは、パフォーマンス CoE の成否を判断する条件を定義するだけでなく、パフォーマンス CoE の経済効果を判定する基準にもなります。たとえば、アプリケーション・パフォーマンスの向上が顧客サービスのコスト低減に対して、どのように直接的な影響を与えるのかを把握できます。

評価フェーズでは、組織全体としての変更に関する戦略を作成し、この変更を実施するためのロードマップを作成します。また、変更内容を規定化するために必要なサービスも選択します。

さらに評価フェーズでは、次のようなプロセスを統合します。

- パフォーマンス CoE の組織構造の設計
- パフォーマンス CoE 構築が組織に与える影響の概要
- 既存のハードウェア、ソフトウェア、人的リソースのインベントリ
- 追加が必要なリソースの洗い出し
- 全体目標の定義
- 高レベルなマイルストーンとスケジュールの決定
- パフォーマンス CoE から期待できる成果をコスト削減とプロセス改善の点から算出

## 概念実証

パフォーマンス CoE 構築を成功させる重要なステップの1つが、概念実証プロジェクトです。このステップはパイロットと呼ばれることもあり、次のような目的があります。

- 設計フェーズで形成した概念を検証
- CoE 構築がもたらす最初の価値を実証
- センター・オブ・エクセレンスの社内での認知度を向上

パイロットの実施では、パフォーマンス CoE が最初に取り組む対象を決定し、実装内容を明確に定義します。パフォーマンス CoE の対象の例としては、アプリケーション開発などアプリケーション・ライフサイクル内にある特定の段階を選択し、その段階のサポートに必要なテクノロジーとリソースを実装します。また、キャパシティ・プランニングなど、特定のアクティビティを対象にすることもできます。さらには、特定の業務単位を対象とし、そこでパフォーマンス CoE のプロトタイプを作成してから、他の領域で展開していく方法もあります。

パイロット段階の準備では、できるだけ多くの側面から CoE を捉える必要がありますが、対象範囲は小さな規模にとどめてください。たとえば、特定の LOB を対象にパイロットを実施する場合、この LOB のビジネス・アナリストと開発者と協力して次の作業を行います。

- パイロットのプランニング
- 関係者の洗い出し
- 既存のテスト用インフラストラクチャ・ハードウェアの評価
- 使用しているソフトウェア・テスト・ツールのインベントリ確認とツールの選択
- 重要性和優先度に基づいてアプリケーション・リストを作成し、2～3種類のアプリケーションを概念実証の対象として選択
- 明確なパフォーマンス目標を定義
- 提案された CoE のテスト・スペシャリストと LOB の代表者でパイロット・チームを結成
- プロジェクトのスケジュール作成とデリバリ・ポイントの特定
- 各ステップの進捗状況を文書化

パイロットの対象として J2EE, .Net, SAP など特定のテクノロジーを選択した場合には、その分野の専門家がパイロットに参加し結果の測定を担当することが重要です。

一般的なパイロット・プロジェクトは2~3か月の期間、集中的に実施され、大規模な CoE の運用開始に向けて複数の要素の改善が継続して行われます。CoE が小規模であっても、LOB の管理者やアプリケーション部門の部長、さらに CIO など、経営陣のサポートは欠かせません。進捗状況と問題点を報告する際に、経営陣からのサポートを定期的に確認してください。

パイロット・プロセスの最後に、パイロット・チームのメンバと顧客（テスト対象アプリケーションのビジネス・アナリスト、開発者、エンド・ユーザ）とで分析セッションを行うことをお勧めします。経営陣からフィードバックを得ることで、不要なミスを防ぎ、CoE を効率的に進めていくことができます。この分析セッションには、可能であれば経営陣の参加を依頼してください。いずれの場合も、プロセスの内容と所見をサマリ・レポートにまとめ、これをマスター計画として使用します。

# コミュニケーション

パフォーマンス・センター・オブ・エクセレンスを構築する過程で、最も重要であるにも関わらず見過ごされがちな要素の1つがコミュニケーションです。作業にはビジネス、ファイナンス、ITなど組織内の各部門から複数の担当者が参加するので、関係者間のコミュニケーションが不可欠です。

関係者に十分な情報を与えることはどのようなプロジェクトでも重要な作業ですが、CoEの構築では特に重視する必要があります。プロジェクト実施期間中、常に経営陣のサポートを得ることの重要性についてはすでに説明しました。このようなサポートを得る方法として、経営陣向けにレポートを定期的に作成します。レポートには、関係者の職務ごとに、それぞれの関連データを盛り込む必要があります。

ほとんどの場合、パフォーマンス・テスト担当者には次のような弱点があります。

- 関係者がどのような情報を欲しがっているのかがわからない
- 不要な情報や関連性のない情報を削除できない（決して情報量が多ければよいというものではありません）
- 技術的な問題がビジネスに及ぼす影響を説明できない
- データをわかりやすく説明できない

パフォーマンス・センター・オブ・エクセレンスの認知度を高め、その価値を効果的に伝えるためには、社内の関係者に提示する情報の準備に真剣に取り組む必要があります。次の項では、「すべきこと」と「してはならないこと」を関係者の役割ごとにまとめています。

- 経営陣
  - すべきこと
    - 運用開始日の遅れにつながるような問題を報告
    - 運用開始後に発生する可能性のある問題を報告
  - してはならないこと
    - 自分で解決できる問題点を報告する
    - 技術的な内容や実行したテストの詳細を列挙する

- プロジェクト・マネージャ
  - すべきこと
    - プロジェクトのスケジュールに関するリスクを報告する
    - プロジェクトの予算に関するリスクを報告する
    - ビジネス部門から承認を得る上で障害になるような問題があれば注意を促す
    - 予定通り完了したタスクのステータスを報告する
    - プロジェクトのマイルストーンと他のチームが与える影響を提示する
  - してはならないこと
    - 発生した問題をほとんど報告しない
    - 技術的な内容や実行したテストの詳細を列挙する
- アプリケーション設計者
  - すべきこと
    - テストの正しいアプローチを提示する
    - システムまたは機能の設計やハードウェア要件の見直しが必要になるような所見があれば注意を促す
    - 機能目標とパフォーマンス目標の達成を踏まえたアプリケーションの能力を報告する
  - してはならないこと
    - 予算やリソースの制約を伝える
    - スケジュールの衝突を列挙する

関係者の職務は組織によって異なる場合があるので、報告内容を職務ごとに固定的に決める必要はありません。

報告は、次の一般的なルールに従って行います。

- レポートを受け取る人の職務に合わせてカスタマイズします
- パフォーマンス・ツールで作成されたレポートをそのまま送付せず、レポートをレビューし、不要なデータを削除します
- 最も重要な内容を最初に記載し、その後で詳細な内容を説明します

- 大量のデータ結果だけでは説得力がありません。結論と分析、パフォーマンス向上のための提案事項を盛り込みます
- レポートから応答時間のデータをそのまま貼り付けるのではなく、主要なビジネス・トランザクションの応答時間だけを報告します
- CoE の運用効率を示すメトリックスを使用します
- 報告書の作成も重要ですが、プロジェクトの完了時にミーティングを招集してプロジェクトの結果が正しく理解されていることを確認します
- テスト・チームが達成した成果を宣伝し、CoE のメリットをいつでも説明できるように準備しておきます

## ツールの標準化

本書でこれまでに説明したように、完全なパフォーマンス CoE を構築するための最初のステップの 1 つがツールの標準化です。非常に基本的なレベルの作業として、それぞれの事業部門で使用されているハードウェア・リソースとソフトウェア・テスト・ツールのインベントリを作成し、標準化の土台を探します。このインベントリから、さまざまなことがわかります。

ソフトウェアは、会社が数量割引で購入するため、ほとんどのプロジェクトで同じテスト・ソフトウェアが使用されます。このような環境では、ツールに関する知識が蓄積されていくというメリットはありますが、それぞれの事業部門の目的に合った機能をソフトウェアが備えているかどうかを確認する必要があります。また、テスト、チューニング、ボトルネックの特定、キャパシティ・プランニング、診断といったパフォーマンス最適化の主要なプロセスを自動化できるアプリケーション群が提供されていることを確認してください。

一方、インベントリ・チェックの結果、たとえば複数のベンダから数種類のソフトウェア・パッケージが購入されているが、機能面でどれも優劣付けがたいケースなどがあります。このような場合には、市販されているパフォーマンス・テスト・ソフトウェアの比較分析を行い、入れ替えを行います。

経済的な理由から既存のツールセットをそのまま使用するとしても、ライセンスを一元管理することをお勧めします。これにより、ライセンスを効率的に使用でき、コストも削減できます。

HP Performance Center ソフトウェアは、次のような機能を備えており、パフォーマンス・ソフトウェア・パッケージとして最適な製品です。

- 一貫性があり、測定と再利用が可能な負荷テストを 1 箇所で集中管理
- 実際のユーザを多数の仮想ユーザで置き換え、あらゆる層でパフォーマンス・ボトルネックを特定
- あらゆるパフォーマンス・テスト資産の管理とテストのスケジュール作成を 1 つの Web インタフェースで実行

HP Diagnostics ソフトウェアと組み合わせることで使用することにより、J2EE、.NET、Oracle、PeopleSoft、SAP、Siebel など専門的なテクノロジーを使った製品で発生するパフォーマンスの問題をコード・レベルまで徹底的に究明します

HP Performance Center は、運用環境とテスト環境向けの監視ソリューションである HP SiteScope のインストール・パッケージに付属し、ライセンスが供与されています。

ハードウェア・リソースについては、ほとんど(あるいはすべて)のサーバ、ルータ、その他ネットワーク機器を1箇所にまとめることにより、これまで分散していたチームに協力体制が生まれ、厳しいテスト・スケジュールへの集中的な取り組みにも貢献します。テスト環境は、それぞれのテスト・グループが製品のデプロイメント領域ごとに確保するのではなく、全体的なデプロイメント・スケジュールに基づいて、必要に応じて提供されます。この方法では、ハードウェアの有効活用だけでなく、テスト環境全体のリソース・コストを低減できます。

以上をまとめると、ツールの標準化には次のようなメリットがあります。

- 使用率の向上
  - すべてのプロジェクトが負荷テストを利用できるようになります
  - 可視化により、負荷テストがいつでも使用できるようになります
  - 資産へのアクセスが柔軟になることで、使用率が高まります
- コスト削減
  - ライセンス・プール・モデルにより、取得コストを低減できます
  - インフラストラクチャを中央に統合することで、メンテナンス・コストを低減できます
  - リソースを中央に統合し、サポート窓口がすべてのハードウェアやソフトウェアのサポートを担当することにより、サポート・コストを低減できます
  - トレーニングの統合により、人件費を低減できます

## 3 CoE の運用

どのようなソフトウェア・テスト・フェーズでも、主に次の成果物が必要になります。

- パフォーマンス・テスト戦略
- パフォーマンス・テスト計画
- テスト・ケース
- 不具合リスト
- ステータス・レポート

パフォーマンス・テスト・チームは、上記の成果物に基づいて、プランニング、テスト、不具合修正、ステータス・レポートというプロセスを進めていきます。上記の成果物は絶対的なものではないので、テスト組織それぞれでどのような項目が成果物として適切かを検討するとよいでしょう。

このフェーズでよくみられるのは、「物理的な」ドキュメント一式を作成しなければならない、という誤解です。物理的なドキュメントやドキュメント一式を作成しなくても、上記の作業を実行できるアプリケーションが提供されています。このフェーズの目的は、必要な内容を使いやすく一貫性のあるフレームワークでできるだけ簡潔にまとめることにあります。

個々の成果物について詳細に説明する前に、パフォーマンス CoE のスタッフが成果物を作成する際に陥りやすい問題点について説明します（[コミュニケーション](#)の項を参照してください）。

パフォーマンス・エンジニアは、「テストを実際に担当していない関係者には、テストの概念を理解することは難しい」ということを念頭に置く必要があります。一般的に、テスト担当者以外は、テストに使用されるツールセットの詳細を理解しているわけではなく、テスト結果と、ソフトウェアの信頼性と品質に自信を持って販売できるかどうか、という点にしか関心がありません。したがって、クライアント、関係者、開発者、マネージャとのやりとりでは、ソフトウェア・テストについてあまり知識がないという点を踏まえてコミュニケーションを行う必要があります。

次に、パフォーマンス・テストのプロセスに関するドキュメント作成と、関係者とのコミュニケーションに役立つヒントを紹介します。

- コミュニケーションの対象となる相手を特定します
- 相手がどのような情報を欲しがっているのかを把握します
- 不要な情報は提供しません
- 簡潔かつ明確なコミュニケーションを心掛け、わかりにくい表現は避けます
- ドキュメントの最後まで目を通さない場合に備えて、重要な部分を強調します
- アクション項目を明確に示します

# プロジェクト計画

## パフォーマンス・テスト戦略

テスト・ドキュメントの中で最初に作成されるのは、パフォーマンス・テスト戦略です。パフォーマンス・テスト戦略は、一般的に大きな組織による大規模プロジェクトで作成されるものですが、パフォーマンス・テストを進めていく上で全体的なアプローチを決めるときに役立ちます。このドキュメントでは、パフォーマンス・テスト・チームがテスト・フェーズで実施する作業の中で、スケジュール、リソース調達、予算編成に影響を与える内容を「おおまかに」示します。また、テスト・サイクルに関する次のような質問に対する回答も盛り込むのが理想的です。

- 他のテスト・フェーズとの関係で、パフォーマンス・テスト・フェーズをどのタイミングで実施するのか
- パフォーマンス・テストはどのような環境で実行するのか
- テストにはどのようなツールを使用するのか
- テスト・チームはどの程度の規模か
- テスト・チームには、どのような人のサポートが必要か
- テスト・フェーズは、他のフェーズからどのような影響を受けるか
- どのような条件でフェーズが開始および終了するか

## パフォーマンス・テスト計画

パフォーマンス・テスト計画では、テストの目的、対象範囲、アプローチ、前提条件、依存条件、リスク、スケジュールなど、テスト・フェーズに関連する内容を記載する必要があります。また多くの場合、ソフトウェア・テスト・フェーズ、テスト手法、テストの方法など、テスト作業に関する一般的な情報も記載します。以上の内容は、「ベスト・プラクティス」のリポジトリに含めることをお勧めします。これにより、重複した情報や矛盾した情報を報告してしまう状況を回避でき、実施すべきタスク（テスト作業のプランニング）を中心としたテスト計画を作成できます。ほとんどの場合、テスト計画は次の質問の回答を元に作成されま

- テスト環境と運用環境にはどのような相違点があるか

- テスト環境と運用環境において、パフォーマンス・テストの結果はどのように解釈されるか
- どの程度の量のトランザクションを使用するか
- どの程度のトランザクション応答時間（TRT）を十分なレベルだとみなすか
- データの作成と環境の整備をどのようなスケジュールで進めるか
- いつスクリプト開発を開始するか
- どのような方法でテストを実施するか
- テストは何サイクル実施する必要があるか
- 分析とレポート作成はどのようなタイミングで、どのような方法で行うか

次の項では、テスト計画について詳しく説明します。

## 目的

実行中のテストの目的を明確に示し、ソフトウェア・テスト・チームとデプロイメントに関わる組織による理解を徹底します。ここでは、「アプリケーション全体」をテストする、というようにあいまいな内容ではなく、具体的な目的を示す必要があります。テストの主な目標を現在のリリースの目標に関連付けて示してください。たとえば、電子商取引システムの場合、今回のリリースの目的が支払機能の追加であれば、テスト目標（ミッション・ステートメントとも呼ばれます）は次のようになります。

拡張された支払機能が全体的な仕様通りに動作することと、対象範囲となる既存の機能を検証すること。

テスト目標には、「なぜ」テストを実施するのかという理由を示します。「なに」をテストするのかという詳細な内容は、テスト計画の対象範囲に記載されます。ここでも、一般的に適用できるテスト目標は、「ベスト・プラクティス」のリポジトリに記載しておいてください。これは、どのようなテストにも共通する一般的な目標であり、回帰テスト・ケースの拡張、新しい要件の追加、テスト・ケースの自動化、既存のテスト・ケースの更新などがあります。

## 対象範囲

「目的」の項では、「アプリケーション全体のテスト」というようなステートメントでは実質的な内容を示していない、ということを説明しました。テスト対象には、テストを行う項目（対象範囲）と、テストを行わない項目（対象外）を一覧で示します。

- 対象範囲

テストの実施対象となるシステム・コンポーネントです。「対象範囲」には、要件、機能領域、サブシステム、ビジネス機能など、テスト組織やデプロイメントに関わるその他組織がテスト対象として明確に指定した項目が含まれます。したがって、テスト計画のこの部分では、「なにをテストするか」という質問に対する回答を記述することになります。つまり、今回のテストの実施対象となるシステム領域を記述します。

- 対象外

テストを行わないシステム・コンポーネントは、「対象外」と明示する必要があります。ただし、対象外のシステム・コンポーネントはテスト・ケースに含まれないというだけで、実行や調査の対象から外されるわけではありません。「なにをテストしないか」という質問の回答が、テスト計画のこの部分になります。強調されることはあまりありませんが、テスト計画のこの部分からリスク・ベースのスケジューリングが始まります。このスケジューリングは、すべての組織が行う必要があります、テストしなくてもよいシステム・コンポーネントはどれか、という質問をベースに考えます。この質問については、テスト計画のアプローチの項で説明します。

## アプローチ

本項では、今回のテスト・フェーズにおいて、アプリケーションに適用するテスト・アクティビティについて説明します。「対象範囲」のシステムでテストを実行する方法と、テスト対象から外れる部分のリスクを軽減する要因を紹介します。

アプローチとは、実行すべき作業を詳細に記載したテスト・スケジュールだと考えることができます。アプローチでは、テスト対象のシステム・コンポーネントとテスト方法を明確に示する必要があります。たとえば、環境テスト、インタフェース・テスト、並列テスト、回帰テスト、寿命テスト、ストレスおよび負荷テストなど、現在のテストに適用されるアプローチです。次に、アプローチの使用が妥当かどうかを、リスクの視点から評価します。

## 前提条件

前提条件とは、事実、ステートメント、他のチームによる期待など、テスト・チームが想定する内容を指します。前提条件は、テスト・フェーズごとに文書化し、これをベースにテスト・アプローチを作成します。したがって、不適切な内容が前提条件に含まれていると、テストに悪影響を与えます。前提条件には、環境を問わず、どのリリースにも適用される共通の条件があります。このような共通の前提条件は、「ベスト・プラクティス」のリポジットリに記載しておいてください。前提条件を文書化する際には、現在のテストだけに適用される内容と、共通の前提条件の中でも特に現在の状況で重要な内容を含めます。

## 依存関係

依存関係とは、テストを進めて行く上で、完了しなければならないイベントまたはマイルストーンを指します。このような依存項目は、テスト・スケジュールに記載します。本項では、テストで重要な意味を持つイベントまたはマイルストーンと、それがテスト・スケジュールに与える影響またはリスクを紹介します。

## リスク

リスクとは、テストに悪影響を与える可能性のある要素です。リスクを項目別に洗い出し、テストに対する潜在的な影響を特定します。プロジェクト計画でテストへの影響がすでに明確に示されているリスクについては、ここで繰り返す必要はありません。

## スケジュール

テスト・スケジュールは、いつ誰がテスト作業を行うかを示したものです。ここでは、テスト計画のベースになった情報と、使用可能なリソース・プールに基づいてテスト・スケジュールを決定します。これまで実施したテスト作業での経験と、現在のテスト目標から、できるだけ正確なテスト・スケジュールを作成してください。プランニングおよびスケジューリングの作成と管理をサポートするツールがいくつか提供されています。たとえば **HP Performance Center** には高度なスケジューリング機能が付属しており、1つのプロジェクト内または複数のプロジェクトでテスト・リソースを効率的に共有することができます。

## テスト・ケース

テスト・ケースとは、テスト・ケース設計を正式に実装したものです。テスト・ケースまたはテスト・ケース・セットの目的は、テスト対象システムの不具合を見つけ出すことにあります。テスト・ケースは、現在のテスト・サイクルだけでなく将来にわたって使用できるような方法で文書化する必要があります。最低限の情報として、作成者、名前、説明、ステップ、実行結果、ステータスをテスト・ケースごとに記載してください。

- テスト・ケースの名前

テストの機能領域と目的など、テスト・ケースの特徴を示す名前またはタイトルです。共通の命名規則を使用すると、名前が重複することがなくなり、再利用も簡単になります。命名規則は、「ベスト・プラクティス」のリポジトリに記載しておくことをお勧めします。

- テスト・ケースの説明

テスト・ケースで実行するビジネス・イベントのシーケンスを明確に示します。アプリケーション領域全体をテストする場合は複数のテスト・ケースが使用されることが多いので、1つまたは複数のテスト・ケースに同じ説明を付加することも可能です。

- テスト・ケースのステップ

テスト・ケースのステップでは、操作手順、データ、イベントなど、ステップの実行に必要な内容を明確に示します。共通の方法に基づいてわかりやすく示すことによって、統一を図ることができ、再利用しやすくなります。テスト・ケースの設計で効果を発揮する方法の1つがキーワードであり、手動と自動のいずれのテスト・ケースにも適用できます。

- 予想される結果

予想される結果とは、検証や確認が必要なテスト・ケース・ステップの実行において想定される動作です。たとえば、画面のポップアップ、データの更新、変更内容の表示、イベント、システム・トランザクションなど、テスト・システム・ステップが実行されたときに発生する動作があります。

- ステータス

実行可能かどうかなど、テスト・ケースの操作上の状態です。

# 結果

パフォーマンス・テストが完了したら、次の手順として、監視および診断フェーズで収集したデータを分析します。テスト結果は、不具合レポートやステータス・レポートなどさまざまな形式で示されます。

## 不具合

テストの主な目的は、運用環境にリリースを移行する前にアプリケーションの不具合を見つけ出すことにあります。また不具合は、プロジェクト・チームがテスト・チームの実績を評価するときの、唯一の判断材料だと考えることができます。不具合は、修正プロセスで役立つような方法で文書化します。最低限の情報として、作成者、名前、説明、重大度、影響が発生する領域、ステータスを不具合ごとに記載してください。

- 不具合の名前

機能領域と状態など、不具合の特徴を示す名前またはタイトルです。

- 不具合の説明

不具合が発生したイベント・シーケンスを明確に示します。可能であれば、エラーの画面ショットまたはプリントアウトを添えます。ビジネスに与える影響を、技術的な専門用語を使わずにわかりやすく記述します。

- 関係者が不具合の重大度や優先度を判断できるように、発生頻度や可能性を示します。

- 再現方法

不具合の修正を担当するトリアージ・チームと開発者が不具合を再現できるように、十分に詳細な説明を記述します。

- 不具合の重大度

不具合の重大度は、テスト・フェーズ、不具合がテストに与える影響、運用環境で不具合が発生した場合のビジネスへのリスクに基づいて判断されます。

- 影響を受ける領域

この領域は、システムの機能コンポーネントまたは機能領域ごとに示されますが、両方が使用されることもあります。

パフォーマンス・テストで検出される不具合は、機能テストで検出される不具合とは次のような点で異なります。

- 不具合には、明示的な要件がないことが多い
- 開発チームだけでは再現できないことが多い
- パフォーマンスの不具合には専門的な特徴があるので、技術に関して詳細な知識を持っていないマネージャには理解しにくい（注：不具合修正を担当する開発者を任命するマネージャを指します）。

ほとんどの場合、不具合の報告にはバグ追跡ソフトウェア・ツールが使用されます。その1つが **HP Application Lifecycle Management (ALM)** , **Quality Center Edition** です。このツールは上記のような情報の入力を備え、**HP ALM, Performance Center Edition** とのシームレスな統合が可能です。

## ステータス・レポート

パフォーマンス・センター・オブ・エクセレンス・チームのメンバは、さまざまなレポートを作成します。作成されるレポートのタイプは組織によってさまざまですが、テスト結果を定期的（日次、週次、プロジェクト・サイクルの終了時）に文書化することをお勧めします。

ステータス・レポートには、テスト目標、対象範囲、スケジュールの現在のマイルストーンを中心とした内容を記載します。このような項目をステータス・レポートの最初に記載し、その後、レポート期間中に達成した成果や目標、次のレポート期間で実行予定の内容を記載するとよいでしょう。テストに直接的な影響を及ぼすことがわかっているリスクがある場合、特に、あるシステム領域でのテストの中断につながるような重大なリスクがあれば明記します。

- レポート期間

現在のステータス・レポートの作成対象となる期間です。過去のステータス・レポートの中でレビューが必要になるものがあれば、それもレポートに含めます。

- 現在のテスト対象

テスト対象となるシステム・コンポーネント（ハードウェア、ソフトウェア、ミドルウェアなど）を「対象範囲」、テストを実行しないコンポーネントを「対象外」として明記します。

- スケジュールのマイルストーン

現在のレポート期間中に取り組んだマイルストーンと、現在のステータスを記載します。実施が予定されていたにも関わらず、現在のレポート期間には実施しなかったマイルストーンは、リスクとして明記する必要があります。

- リスク

リスクとは、現在のテストに悪影響を与える可能性のある要因です。テストに現在影響を与えているリスクの一覧と、テストにどのような影響を与えるのかを示します。

次のような理由から、共有ドライブなどを使って、関係者がレポートや未加工データにアクセスできる環境を作ることをお勧めします。

- 「自分で分析できる」という自由度を与えることができます
- 「すべてが開示されている」という信頼を得ることができます
- Excel など使い慣れたツールでデータを加工できるので、テストへの参加を促すことができます

## テスト実行レポート

これは、テストの実行後に作成される情報提供のための通知であり、一般的に Word ドキュメントではなく電子メールが使用されます。このレポートは技術部門の関係者を対象としたものであり、CoE チームが参照用に保管します。

このレポートでは、次に示すようにテスト実行に関して重要性の高い内容を記載します。

- テストの目的
- 構成の変更（該当する場合）
- テストがビジネスに与える影響
- 不具合のチューニングまたは調査で見つかったボトルネックの根本原因
- 結果ファイルとシナリオ・ファイルの参照

テスト実行に関するコメントを記録して関係者に送付することには、テスト担当者が結果の分析に真剣に取り組むという効果があります。

## 定期レポート

定期的なステータス・レポートは必須ではありませんが、作成をお勧めします。日次、週次、月次などの定期レポートはプロジェクト・マネージャ向けに作成されることが多く、次のような情報が含まれます。

- スケジュールの追跡方法
- スケジュール、予算、プロジェクト・リソース調達に影響を与える可能性のある問題
- 今回のレポート期間中に実施した内容の概要
- 来週の作業予定
- すでに解決した問題のリストは不要
- 短時間で目を通せるように、不要な情報は記載しない

## サマリ・レポート

システム・パフォーマンスと安定性についてまとめた最も重要度の高いレポートであり、プロジェクトまたはテスト・サイクルの最後に作成されます。専門的な内容や高度な内容も一部含まれていますが、すべての関係者に一部ずつ配布されます。

このレポートでは、テスト・サイクルについて次の内容が網羅されます。

- 実行したテストと合否  
パフォーマンス・テスト・サイクルの大部分では、テスト結果は単に「合格」と「不合格」に分類できない複雑な判定になります。テストの主な目的は、「ライブ稼働」した後の動作を予測することにあります。
- 主要 KPI の明示  
これには、応答時間、エラー、トランザクション・ボリューム、システム・リソースの使用率などがあります。
- 対象範囲の変更  
テスト計画で示した対象範囲と最終的なテスト作業の範囲が異なる場合には、その内容を明示します
- 未解決の問題

テスト判定に影響を与える唯一の結果が不具合です。したがって、重大度の高い不具合とビジネスに与える影響を一覧にし、可能であれば優先度ごとに分類します。

#### — AUT の設定

バージョン、主な設定、ファイルの場所など、すべての関連情報を、対象範囲と伴にレポートに記載します

サマリ・レポートは、次のガイドラインに従って作成します。

- 未加工のテスト結果だけでなく、結論と分析の前提条件も併せて記載します
- テストで使用した最終的な設定を記載します。ただし、チューニングに至るすべてのステップを記載する必要はありません
- テスト・データのソースとステータス、準備作業を、過度に詳細にならない程度に記載します
- 結論には、重要度の高い応答時間のみを記載し、残りは付録にまとめます

ガイドラインに従って関係者の関心がある情報をまとめたレポートが配布されても、ほとんどの場合、読まれるのはサマリの部分だけで、最後まで目が通されることはありません。したがって、サマリ・レポートの作成後にミーティングを招集し、重要なテスト結果や結論を PowerPoint などのビジュアル・ツールを使って説明することをお勧めします。これにより、実施したパフォーマンス・テストへの理解を深め、承認を得ることができます。

## 4 CoE を構成するメンバ

パフォーマンス・センター・オブ・エクセレンス構築の重要な目的の1つに、負荷テスト、スクリプト作成、さまざまなアプリケーション・テクノロジーや分野の専門知識の蓄積と活用を通じて、人材を最大限に活用することがあります。

PCoEは、テスト・スペシャリストとテスト・コンポーネントで構成されるグループであり、繰り返し活用できる知識、技術、手法、リソースの提供を可能にします。テスト・スペシャリストは、各プロジェクト・チームのスタッフを教育し、足りない部分を補強します。また、作成された成果物を管理し、将来的なプロジェクトに対してフレームワークやサンプルとして提供します。CoEは、共有可能なベスト・プラクティスの基盤作りを行います。また、テスト計画、テスト・ケース、テストの自動化、不具合など、テストの成果物の再利用を推進します。このプロセスを通じて再利用可能なコンポーネントが増え、テスト作業の効率化へとつながります。

### 役割

組織の規模、構造、成熟度、業種によって異なりますが、パフォーマンス・センター・オブ・エクセレンスはいくつかの部門とチームに分割されます。ほとんどの場合、主に次のような役割で構成されます。

- CoE マネージャ
- ソリューション設計者
- テスト・プラクティス・リーダー
- テスト・オートメーション・エンジニア
- ビジネス・アナリスト
- オペレーション・マネージャ
- テスト・ラボ・スペシャリスト

上記は役割と職務であり、リソースではありません。したがって、PCoE 構造内にある複数の役割を1人に割り当てることもできます。

このリストには環境スペシャリストが含まれていますが、PCoEに関連した作業のみを担当します。プロジェクトや組織全体のレベルの作業や役割（設定管理など）は含まれません。

## CoE マネージャ

CoE マネージャの役割は、テスト・チームを効果的に統率することにあります。そのためには、マネージャとしての一般的なリーダーシップに加えて、テストの実施やテスト・プロセスを効率的に実装する方法を理解する必要があります。したがって、テスト・プロセスを効率的に管理、実装、保守できる人材がマネージャとして適しています。堅牢なコミュニケーション機能とコスト効率の高いテスト・フレームワークをサポートするテスト・インフラストラクチャの整備も必要です。

職務：

- センター・オブ・エクセレンスを推進する
- PCoE が組織構造の中で果たす役割の重要性を理解および推進する
- リリースまたはデリバリの視点でテストの対象範囲を定義する
- テスト手法の使用方針を徹底する
- サービス・レベル・アグリーメント（SLA）の確立と達成をサポートする
- SLA に関する知識をプロジェクト・チームに提供する
- テストで発生した問題の解決に向けたエスカレーションを行う
- PCoE の効果を測定する KPI を実装し、継続的に作成する
- テストの計画作成、デプロイメント、管理を実行する
- テスト実施に必要な次のテスト資産を管理し、継続的に更新する：
  - チーム・メンバ
  - テスト・ツール
  - テスト・プロセス

- CoE の人材と予算を管理する
- チーム・メンバのトレーニングを実施する
- パフォーマンス・テストのスキルを持ったスタッフを確保する

テスト・チームの管理や統率は、IT 業界で最も難しい職務の 1 つです。なぜならば、チームには人材が不足していて、ツールや予算も十分ではない場合が多いからです。このような状況では、主要なテスト・スタッフの士気と定着率を高めることが重要です。次に、このような課題に取り組む上でのヒントをいくつか紹介します。

- スケジュールの変更に合わせてテスト計画の対象範囲を更新する
- テスト・チームとプロジェクト管理チームの両方で、期待される項目を明確に設定する
- 開発者とプロジェクト・マネージャ間のコミュニケーションを常に円滑に行う
- テスト・チームが果たす役割の重要度と貢献度を高める
- センター・オブ・エクセレンスの各チーム・メンバに明確な役割とキャリアパスを提供する
- PCoE の ROI を測定および報告する
- コストではなく投資（ROI）の点からテスト費用を説明する

## ソリューション設計者

テスト・ソリューションの設計者には、テスト・アーキテクチャの視点から PCoE のビジョンを構築し、整備していくという役割があります。これは、センター・オブ・エクセレンスの中でも最も重要な役割であり、複数の業種でこれまでにプロジェクト管理、テスト・プラクティス・リーダー、テスト・オートメーション・エンジニアとしての実績を持つシニア・クラスのテスト担当者が適任です。テスト・ソリューション設計者は、テスト全体の効率化に役立つツール群、プロセス、手順を選択して統合します。テスト設計者は、CoE マネージャ、テスト・プラクティス・リーダー、シニアクラスのテスト・オートメーション・エンジニア、ソフトウェア・ベンダ、品質保証チームとの密接な協力体制に基づいて、テストのフレームワークを構築し、実装します。

職務：

- テスト・アーキテクチャの視点からのビジョンを管理および保守する
- テスト・アーキテクチャに関する短期的目標と長期的目標を設定する
- 上記の目標達成に必要なソフトウェア・テスト・ツールを選択する
- ツール、プロセス、手法を1つの構造へと統合する
- テスト・フレームワークとテンプレートをプロジェクトに提供する
- PCoE アーキテクチャの進展を促進する

## テスト・プラクティス・リーダー

テスト・プラクティス・リーダーには、テストの設計と実施の効率化を図るプロセス、手順、テンプレートを提供するという役割があります。対象となる範囲には、テスト設計、テストの作成、テストの実行、テスト結果の収集、不具合のレポート手順、テスト・カバレッジの分析が含まれます。

職務：

- テスト手法に関するトレーニングを行う
- QA との連携を通じて、テスト手法の継続的な改善を図る
- 手法の実践的な使い方について指導する
- テスト・プロジェクトの編成について指導する
- テスト戦略を評価し、承認に関するアドバイスを行う
- 技術的なテスト・プロセス手法が適切にカバーされていることを確認する

## テスト・オートメーション・エンジニア

テスト・オートメーション・エンジニアには、効率的なテスト自動化ソリューションの設計、構築、テスト、デプロイメントという役割があります。このような役割を果たすために、自動化テクノロジーを採用して、テスト組織が掲げる短期的および長期的な目標の達成に取り組みます。その目的は、最小限のコードやスクリプトを使って、テスト作業をできるだけ自動化することであり、テスト・カバレッジではなく、テスト作業の最適化が作業の中心になります。自動化の第1

候補は、テスト・リソースを大量に消費する手動テスト・ケースや手動のテスト準備プロセスです。

職務：

- テスト・ツールの実践的な活用スキルを身につける
- テスト・ツールの使用方法に関するトレーニングをエンド・ユーザに実施する
- テスト・ツールを管理する
- テストのアップグレードや新しいツールに関する情報を提供する
- パフォーマンス、負荷、キャパシティなどさまざまなタイプのテストを実施する

## ビジネス・アナリスト

ビジネス・アナリスト（BA）は非技術分野の専門家（SME）とも呼ばれ、PCoEの階層構造に所属する場合と所属しない場合があります。ほとんどの企業では、パフォーマンス・センター・オブ・エクセレンスは社内組織の1つであり、ビジネス・アナリストは社内または事業部門ごとに独立した1つのエンティティとして存在します。PCoE組織にビジネス・アナリストが所属していない場合、PCoEはBA/SMEとの緊密な連携を通じてテスト対象となるビジネス・プロセスを明確に理解する必要があります。PCoEがアウトソーシングされた社外組織である場合には、PCoE内部にBAが必要になります。このBAは、クライアントとの円滑なコミュニケーションを図り、PCoE内でのビジネス担当窓口として、ビジネス目標が明確に理解され、テストがクライアントの要件を満たしていることを確認します。

職務：

- 特定の業種の現状や動向に関する深い知識を持つ
- トレンドに応じたプロセスに対応できるように手法を適合させる
- テスト・プロセスやテスト・フェーズについて、業界の動向を考慮したアドバイスを提供する

## オペレーション・マネージャ

オペレーション・マネージャには、ソフトウェア、ハードウェア、人材といったPCoEリソースの使用スケジュールを作成するという役割があります。オペレーション・マネージャは、CoEマネージャ、ソリューション設計者、ラボ・スペ

シャリストとの緊密な協力体制を通じて、想定済みまたは想定外のリソースの重複に対処できるようスケジュールを調整します。オペレーション・マネージャは、リソース・マトリックスを実装し、事業部門およびIT部門に所属するすべてのPCoEパートナーに周知徹底する必要があります。これは、複数のクライアントやパートナーが共通のリソース・プールを共有する場合には特に重要です。

職務：

- 人材や環境などプロジェクトで使用するリソースを共有する方法をスケジュールリングする
- 計画、ケース、データ、スケジュール、スクリプトなど、再利用可能なテスト・ビジネス・コンポーネントを管理する
- テスト・スケジュールを公表する

## テスト・ラボ・スペシャリスト

テスト・ラボ・スペシャリストには、PCoEのハードウェアと一部のソフトウェアが、予定されたすべてのテスト作業をサポートしていることを確認する役割があります。

職務：

- アプリケーションとシステムを含めたハードウェア、ソフトウェア、ネットワークを管理する
- 技術的なキャパシティ・プランニングを実施する
- パフォーマンス、負荷、キャパシティなどさまざまなタイプのテストをサポートする
- 技術サポート・グループとの主なインタフェースとしての役割を果たす
- 技術的な知識をプロジェクト・チームに提供する
- 再利用可能なテクニカル・テスト・コンポーネントを管理する
- テスト中に技術的な環境を監視する

PCoEスタッフの多くが、上記の役割のいくつかを果たします。

## 負荷テストのスキル

効果的なテストを実施することによって、市場で販売を開始する前、または社内のビジネス部門にサポートを提供する前に、製品の状態や製品の拡張に関して正確な情報を提供できるようになります。これはビジネスでの大きなメリットであり、実装前の適切な対処や、実装後に提供するサポートのプランニングが可能になります。

ただし、負荷テストを適切に設計および実施しないと、テストを効果的に行うことはできません。パフォーマンス・テストは特殊なスキル・セットであり、アプリケーション、ハードウェア、サードパーティ・システムに関する知識が必要です。具体的には、アーキテクチャの視点からアプリケーションの動作を理解することや、モデリング・スキルと体系的なアプローチが必要になります。すべてのテスト担当者がこのような条件を満たしているわけではなく、スキル・セットを完全に習得するには何年もかかります。

したがって、パフォーマンス・スペシャリストを適切な方法でトレーニングし、有能な人材を確保することが、パフォーマンス・センター・オブ・エクセレンスで最も重要な課題の1つになります。

この課題に対する最初の取り組みが、包括的な教育を継続的に実施することです。ここでは、PCoEの中核となる役割であるテスト・オートメーション・エンジニアについて、学習パスのサンプルを紹介します。

- 基本的な知識
  - プログラミング：1種類以上のオブジェクト指向言語、またはCに関する知識
  - オペレーティング・システムのチューニングのスキル
  - ネットワーク分析とチューニングに関する専門知識
  - データベース分析とチューニングに関する技能
  - システムをアーキテクチャの観点から理解する能力
  - インフラストラクチャのボトルネックを監視する能力
  - ツールに依存せずに、システム・ボトルネックを検出および対処する能力
- 負荷テストに関する技能
  - ソフトウェア・テスト・ツールに依存せずに、テストを効率的に実施する能力

- テストを包括的に捉え、ソフトウェア・テストとソフトウェア以外のテストで適用されている不適切なテスト・プラクティスを洗い出す能力
- ソフトウェアを使用せずに、パフォーマンス・テストを実施する能力
- 技術的な知識を持つ対象と持たない対象にテスト結果をわかりやすく報告する能力（[コミュニケーション](#)を参照）
- より高度なレベル
  - ベンダが提供する認定コースへの参加。HP Services が提供するトレーニング・コースでは、テストを成功させるために必要になるほとんどの内容についてしっかり学習できます。
  - 指導者のサポートを得ながら専門知識の蓄積を開始。ジュニア・テストから、テスト担当者、1つまたは複数のプロジェクト・リーダーへ
  - 他のメンバの指導者

パフォーマンス CoE では、スペシャリストを確保する方法として、IT プロフェッショナルに魅力的な新しいキャリア・パスを提供します。これにより組織は、有能な人材を獲得したり流出を防ぐことができます。そのためにも PCoE マネージャは、チームとの意思疎通を図り、視野を広げ、自己学習を奨励し、報酬による定期的な実績評価を行っていく必要があります。

# アプリケーションに関する知識

これまで、センター・オブ・エクセレンス内のそれぞれの役割が行う職務について、CoE マネージャからパフォーマンス・エンジニアまで順番に説明しました。このような職務を行う能力に加えて、CoE のメンバには負荷テストの専門スキルが必要です。

スペシャリストとしての基本的な知識だけでなく、テスト対象アプリケーション (AUT) , そこで使用されるテクノロジー, API, UI などを理解する必要があります。

- テクノロジー

バグは、テクノロジー固有のバグなど、システム・テクノロジーの影響を受ける場合が多いため、パフォーマンス・エンジニアはシステムがどのように構築されているのかを理解する必要があります。そのためには、使用されているプログラム言語、システム・アーキテクチャ、オペレーティング・システム機能、ネットワーク、プレゼンテーション層、データベースの機能と実装などに関する知識が必要です。

- アプリケーション

システム的设计ではなく、システムの動作に起因するバグもあります。このようなバグの修正では、正しい結果を予測する作業は特に難しくなります。したがって、システムで扱われているビジネス上の処理、技術的な処理、科学的な処理について理解する必要があります。

テクノロジーとアプリケーションの両方の知識が必要であることは明らかであり、このような知識の有無が、パフォーマンス検証プロジェクトの成否を分ける鍵になることがあります。たとえば、業界で広く普及しているアプリケーション・サーバ上で稼働する J2EE アプリケーションが AUT である場合、テスト・チームには Java の専門家が絶対に必要になります。Java の専門家は、Java 特有のボトルネックを特定し、Java プロファイラを使ってあまり知られていないシステム・パラメータを調整するなど、Java の分野で実力を発揮します。

また、データベース・パフォーマンス・スペシャリストの知識は、ほぼすべての CoE チームで必要になります。新しいアプリケーションのほとんどは、SQL を直接呼び出したり、データ・アクセス層をカプセル化するアプリケーション・サーバ・オブジェクトを使って、リレーショナル・データベースに格納されたデータを使用します。データベース関連の問題検出は非常に難しい作業ですが、センター・オブ・エクセレンスでは必ず必要になるスキルです。Oracle, IBM DB2, MS SQL Server などのデータベース管理者は常に需要がありますが、特にパフォーマンスの問題に対応できるスキルを持った人材は非常に少数です。幸運

にもこのような人材を確保できたとしたら、チームのメンバーとして保持できるように魅力的なキャリア・パスを提示することをお勧めします。

チーム内で一般的な知識の蓄積を始めたとしても、このような知識は、ERP や CRM などの複雑なエンタープライズ・システムに十分対処できるレベルではありません。このようなシステムでは、アプリケーション・アーキテクチャ全体の知識に加えて、特定のモジュール、アプリケーション設定、実装など AUT の動作に大きく影響する要素の知識が求められます。

プロジェクト・マネージャは時として、能力を超えた対応を求められる場面に遭遇します。このような場合には、スタッフの増員ではなく、知識を増強して対処する必要があります。そのためには、いわゆるコンサルタントを採用します。さまざまなタイプのコンサルタントがありますが、CoE のコンサルタントに求められる資質は「専門知識」です。

一般的に、CoE のチームとして最適な人材を確保することが、プロジェクトの成否の鍵になります。継続的なトレーニング、やりがいのある仕事への挑戦、専門家向けのフォーラムやイベントへの参加、実績評価などを通じて、CoE 全体の知識レベルを上げる取り組みを行ってください。

## 5 結論

センター・オブ・エクセレンスは、コンピテンシー・センターとも呼ばれ、長い歴史があるものの、これまでは J2EE や .Net など特定のテクノロジーに特化したものでした。パフォーマンス・センター・オブ・エクセレンスは、パフォーマンスの検証を中心に、テストの効率化を目指すプロセスやツールを取り扱います。

PCoE モデルに基づいて、テスト・スペシャリストと技術的なテスト・コンポーネントで構成されるグループが構築され、あらゆるプロジェクトにおいてテストの知識、手法、リソースとして活用されます。テスト・スペシャリストは、各プロジェクト・チームのスタッフを教育し、足りない部分を補強します。これによって、最小限のスタッフと機器で、テスト・カバレッジを最大限にまで広げることができます。

PCoE を設置および運用すると、あらゆる投資の効果を短期的および長期的に向上することができます。また、組織内のテスト・プロセスの成熟度を高めることもできます。このように成熟したパフォーマンス・センター・オブ・エクセレンスには、次のような利点があります。

- 環境的なリソース、知識、人材を複数のプロジェクトで活用できます。
- すべてのプロジェクト・チームがテスト・スペシャリストのスキルを活用できます。
- すべてのプロジェクトでテストの統一性を確保できます。
- 結果の再利用が可能になります。
- 最終製品の品質を向上できます。
- プロセスの改善に役立つ主要なメトリックスと分析を構築できます。
- テスト組織のビジネス価値向上に役立つリスクベースのテスト、要件のトレーサビリティ、知識管理を行う手法を採用できます。
- テストのスケジュールと予算の達成をサポートします。
- 人材や技術リソースの共有を通じて、プロジェクト全体でテスト予算を削減できます。
- 予測可能で正確なテスト作業を実現します。

このフレームワークでは、人材、プロセス、テクノロジーという戦略的資産を統合することで、堅牢なガバナンス・モデルが形成され、効率化、投資収益率の最大化、ソフトウェア品質の大幅な向上を実現します。さらに CoE フレームワーク

の実装で実現されたコストの削減分は、競争力を強化する新しいソリューションや製品へと再投資されます。

業界アナリスト会社である **voke** が実施した調査については[すでに紹介](#)しましたが、PCoE モデルが継続的な牽引力を発揮し、顧客満足を提供していることが示されています。**voke** は、次のようなアクションを推奨しています。

- 賛同を得る：経営陣からの強力なサポートを得る
- 事業部門とのコラボレーション：技術的な視点ではなくビジネスの視点でコミュニケーションを図る
- 組織に合わせてアプローチをカスタマイズ：組織はそれぞれ異なる特徴を持っている
- スキル獲得：人材の確保が第一
- テクノロジーの獲得：適切なツールと教育が鍵になる
- 顧客満足中心の考え方：お客様（社内の顧客も含む）を最優先に考える
- パイロットで結果を実証：少しずつ段階的に進める
- 結果を売り込む：自分自身の宣伝広報担当になる
- CoE スタッフの士気を高める：人材こそがすべて

顧客は競争力とパフォーマンスを同一視：人材とリソースの有効活用を通じて顧客ニーズへ迅速に対応する

## A PCoE の質問

ここでは、パフォーマンス・センター・オブ・エクセレンスの構築と実施に含まれるさまざまな段階において、顧客調査で使用できる質問をまとめました。この質問の回答をもとにビジネスの成熟度を自己評価できます。

1. 組織の全体的な構造を示すチャートを作成し、パフォーマンス・チームの位置を示してください。
2. パフォーマンス・テスト組織の構造について説明してください。この組織内に地理的に離れた部門があれば、役割を説明してください。また、アウトソーシングが活用されている場合には、どのような目的で活用されていますか。
3. この組織にはどのような責任がありますか。
4. 現在のテスト組織は、パフォーマンス CoE とみなすことができますか。
  - a. PCoE をどのように定義しますか。
  - b. CoE の構築プロジェクトを開始したのは誰ですか。
  - c. 現在、パフォーマンス・テストの対象になっていないプロジェクトはありますか。また、その理由を説明してください。
  - d. 負荷テスト、監視、分析、診断などに標準ツールセットを使用しましたか。使用した場合、どのようなツールを使用しましたか。
  - e. 標準ツールを使用しないプロジェクトはありますか。
  - f. 開発ライフサイクルの中で、どのフェーズからプロジェクトに参加しますか。
  - g. パフォーマンス・テストでは共通の手法を使用しますか。そのような手法は複数ありますか。
  - h. この組織では、製品を運用環境にリリースする前のパフォーマンス検証は必須だと考えられていますか。
5. すべての LOB で共通の標準的な方法がありますか。それとも、LOB ごとに異なる方法が採用されていますか。異なる場合、主な相違点を教えてください。
6. 各 LOB に対してどのようなサービスを提供していますか（パフォーマンス検証、パフォーマンス最適化、キャパシティ・プランニング、作業負荷分析など）。

7. PCoE と LOB の関係について、正式な対応付けはありますか。SLA は実施されていますか。
8. パフォーマンスを検証する場合、正式なレポートまたは結果の提出が要求されますか。
9. パフォーマンス・プロジェクトの終了後、どのような作業を行いますか。開発チームとのミーティングなどがありますか。
10. CoE を実施する上での大きな問題点を、技術面と管理面から説明してください。
11. CoE の管理と運用について、今後 2～3 年の目標を教えてください。

