



IBM Application Discovery によるソフトウェア品質の向上

3 ステップ・アプローチ

目次:

- 1 ソフトウェア品質管理とはなにか?
 - 3 ソフトウェア品質管理を組織に組み込むための3ステップ・アプローチ
 - 4 ステップ1-見える化
 - 5 ステップ2-改善
 - 6 ステップ3-評価
 - 7 提言
-

新たなビジネス価値を創出するためにソフトウェアの品質管理を実行していくことは、今日の多くの大企業にとって不可欠です。

従来から、品質管理に関する取り組みやプロセスの多くは、特定の規制や法的準拠を満たすことが目的であり、IT部門におけるコスト、サービス、迅速性を改善させることが目的ではありませんでした。

当初から、もしくは長期にわたって劣化した低いソフトウェア品質は、テクノロジー活用における損失の主な原因となりかねません。この損失を低減することは、今日の厳しい経済情勢において、多くの組織の重要な課題です。

「現実のビジネス」の中で、ソフトウェア・アプリケーションやその品質がどう位置づけられているか理解および管理するためには、すべてのコンポーネントの構造を深く理解する必要があります。エンドー・ツー・エンドのビジネス・プロセスを実行するためにそれらかどのように相互接続されているかを解析することによって、ITシステム全体を通して、ビジネスにあり方に即したソフトウェア品質を保証できるようになります。

IBM® Application Discovery and Delivery Intelligence の3ステップ・アプローチでは、ソフトウェア品質管理を実行するためにはなにが必要か、実行することから何が達成できるかについて実践的な視点から見ていきます。

ソフトウェア品質管理とはなにか?

ビジネス・ソフトウェア・アプリケーションの品質管理は、プロセスをサポートし自動化するツールの発達と、そのコスト効果の向上、実装が現実的になってきたことを受けて、「主観的」なものから「客観的」なものに急



速に移行しています。しかし、各コード・コンポーネントを単純に解析、評価することだけでは、それ自体は開発作業の品質の向上には有効だとしても、以下の意味のビジネス価値を本質的に提供するとは限りません:

- サービス・コスト
- 運用上のパフォーマンス
- サービス提供
- 保全性

さらに、アプリケーション・ソフトウェアを最新に維持するための累積コストは、ソフトウェアの低品質や複雑さによって大幅に増加するため、ソフトウェア品質が低いことが組織の技術的損失も引き上げます。

ソフトウェア品質の向上とは、IT バリュー・チェーン全体にわたる評価、改善、および管理を意味し、これは、実際には、エンドツーエンド・ビジネス・プロセスを提供する複数のアプリケーション・スイートです。

今日の世界で一般的なビジネス・プロセスを考えてみましょう。それは、以下の点で IT 境界にまたがっている可能性があります:

- ソフトウェア・パッケージ
- アプリケーション言語
- 技術環境

また、複雑なマッピングで、複雑なインターフェイスまたは統合ソフトウェア・レイヤーによって、境界をまたいでいる可能性が高くなります。たとえば銀行システムにおいては、Java 言語で書かれたクライアント・フロントエンドは、メインフレーム上の COBOL 言語で書かれ、コアバンキング・パッケージとリンクしているクレジット処理と連携しています。(図 1 参照)。

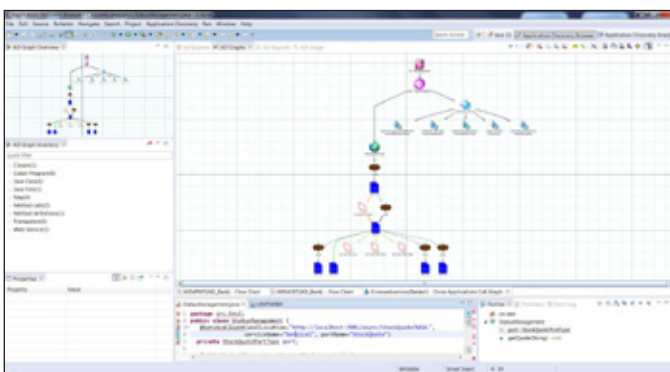


図 1: Application Discovery による、バンキング・アプリケーションのクロスアプリケーション呼び出しグラフ

アプリケーション・ソフトウェア品質とは、システムの機能全体を構築する高品質なコード・コンポーネントとそれらの優れた連携です。

コード・コンポーネントがその組み合わせ全体に結合されたとき、整合性と品質を保証できる IT 開発チームはほとんど存在しません。

品質上の問題の約 80 % は統合テストまで検出されず、これを修復するには、多くの場合、コストと時間がかかります。

一貫性のある品質指標を適用し、異なるアプリケーション、コンポーネント、環境すべてにわたって管理することは非常に困難です。事実、納品期間を短縮し、最終成果物の品質を改善するためには、慎重な管理が必須となるアジャイル開発メソッドロジーの採用が必要となります。

プロセスを組み込み、IT 組織全体に自動化されたツールを統合することによって、IT 部門は、品質標準が規定され、アプリケーション・ライフサイクル全体を通して適用、管理されていることが確認でき、そのライフサイクルの中でソフトウェア品質管理のための適切なタイミングに適切な情報が提供されます。

「CIO と IT 管理チームは、クリティカルなアプリケーション資産の整合性の管理について、企業内の他の部署と新しい種類の対話を始める必要があります。」¹

IT 部門	ソフトウェア品質管理を適用する重要分野
アーキテクチャ	アーキテクチャ標準、アプリケーションアーキテクチャ、統合アーキテクチャ技術プラットフォーム、計画のアップグレード、IT戦略
設計	アプリケーション設計、影響評価、コーディング標準、統合、Common Services 設計
開発	コーディング標準、コード品質、統合、単体テスト、不具合修正、改善
テスト	コードカバレッジ統合テスト、UAT、回帰テスト
サポート	アプリケーション保守、フォールト解析、アプリケーションパフォーマンス監視
管理	アプリケーションポートフォリオ管理、変更影響分析、サービス管理、プログラム/プロジェクト管理

ライフサイクル全体を通してソフトウェア品質を管理することにより、以下を保証します:

- ソフトウェアの品質問題を早期に特定し、解決する
- ソフトウェアの品質を、複雑性やコストなどの主要指標を利用して監視
- 改善を長期にわたって追跡
- アウトソースされた納品を、高レベルの品質で管理
- 納品されたアプリケーションが堅牢でその目的に適している
- 保守と改善を、より短期間に低予算で実行
- 技術的損失を把握し、その結果低減が可能となる

これにより、IT 組織はアプリケーション開発と保守予算（一般的に IT 予算全体のおよそ半分）から、以下の点で目に見えるビジネス・バリューを証明できます:

- アプリケーション停止、コンプライアンス問題、セキュリティ脅威からの潜在的なリスクの低減
- アジャイル開発プロセスへの移行の正当性実証
- 貴重なリソースを、単純なアプリケーション保守や不具合修正ではなく、ビジネス・バリューのためのアプリケーション改善のために再配当
- 既存のアプリケーション資産を最大限に活用

「アプリケーション開発および保守を最適化することにより、コストを 50 % 以上削減可能」²

ソフトウェア品質管理を組織に組み込むための 3 ステップ・アプローチ

ソフトウェア品質管理プロセスおよびツールを IT 組織内で成功裏に組織化するには、一貫性があり、わかりやすいアプローチを、各ステップで同じツールセットを使用して適用する必要があります。これにより、インベントリ、構造的品質と長期にわたる改善に関する「唯一の真実」が提供されます。これは、最新のものではない、または品質標準を開発者が適用するには十分に詳細な情報を提供しないことが多い設計情報の主観的な評価ではなく、実際のコード・コンポーネントの客観的な分析を基に行われる必要があります。

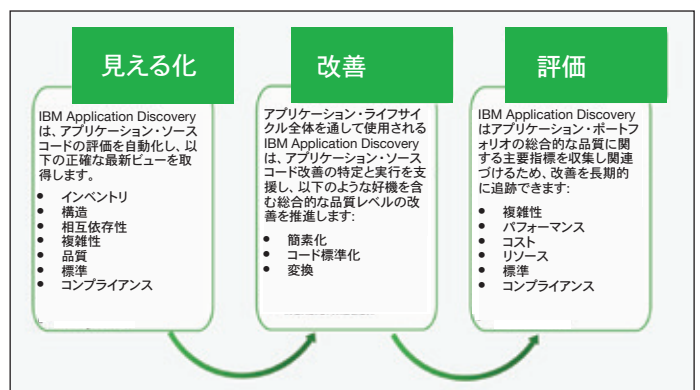


図 2: IBM Application Discovery Solution のソフトウェア品質管理のための 3 ステップ・アプローチ

多くの組織で見られる、詳細情報の欠けたインベントリや構造情報では、IT 管理担当者は、ソフトウェア品質がコスト、リソース、そしてサービスにどのような影響を与えるか十分認識できません。

改善のための変更を詳細に評価することを前提に行うことで、変更のベースラインを規定し、アプリケーション・ランドスケープ全体のアーキテクチャ上、および構造的に品質を改善し、納品品質を改善することができます。

その後のソフトウェア品質は、長期にわたって追跡された、明確に定義された主な指標を提供する正確な情報に依存し、(a) 改善が達成されているかどうか、(b) 組織にはっきりとした価値を提供しているかどうかを評価できます。

10 年以上も厳しいアプリケーション保守予算、その後の多くの組織内でのアプリケーション保守と改善活動が停滞している中、エンドツーエンドのソフトウェア品質の改善でこれらの問題に効果的に対処できることを認識することは、解析およびレポートの自動化ツールを基盤として、プロセスの各段階で正確な情報にアクセスできることが今まで以上に重要であることを意味します。

「ユーザー・エンゲージメント、および静的および動的解析を介したアーキテクチャおよびコード品質に対する可視性の必要性、そして内部および外部リソースのパフォーマンスを評価する効果的な指標が、ソフトウェア品質管理の自動化を推し進めています。」³

ステップ 1 – 見える化

多くの大企業で、アプリケーション・ソフトウェア・ポートフォリオに対する考えは、たずねた相手によって異なります。これは、その観点に大きく影響を受けています。開発者はコード・コンポーネントの視点から、アーキテクトはランドスケープ/標準全体の視点から、プロジェクト・マネージャーはコストやリソースの視点から、そして最も重要なビジネス・ユーザーは、提供されるサービス・レベルと機能性の観点から見ています。

アプリケーション・ソフトウェア・ポートフォリオ全体の品質について共通の基準を持つことは、以下がない限り最善でも容易でもない、最悪の場合は IT 幹部の悪夢となります：

- 明確に規定されたプロセス
- 合意を得た指標
- データ収集を自動化する、統合されたツールのスイート (およびこれを最新の状態に維持すること)
- 簡単に使用できるレポート/追跡メカニズム

戦略的なアプリケーション・ライフサイクル計画とポートフォリオ管理の実行は、アプリケーション・ポートフォリオの見える化第 1 段階であるインベントリ化抜きには不可能です。これには以下が含まれます：

- すべての技術プラットフォームおよび言語
- すべてのアプリケーション・コンポーネント
- すべての相互依存性

Report summary			
TOTALS			
Total BackEnd Function Points	52	Total number of statements	1184
Total modules number	116	Total number of calls	108
Total number of includes	19	Total number of accessed files	1
Total number of MSDs	3	Total number of programs (C/C++)	26
AVERAGES			
BackEnd Function Points per program	0.2	Number of statements per program	45.5
Number of modules per program	4.8	Number of calls per program	4.2
Number of includes per program	0.7	Number of accessed files per program	<0.1
Number of MSDs per program	0.1		

図 3: Application Discovery のインベントリ・レポート機能

これは、実際に展開されたアプリケーション・ソース・コードから抽出され、不完全であったり最新でない可能性のある設計書からではなく、実施の状況を反映する必要があります。

保守人員は、コードを理解するために 47% の時間を費やしている⁴

次の段階の「評価」は、インベントリに基づいた次のレベルの情報です。これは、業界標準またはカスタム指標を使ったアプリケーションとそのコンポーネントのより詳細な解析であり、アプリケーション・ソフトウェア品質の最初のビューです。通常、これには以下が含まれます：

- 複雑性 (機能ポイント、McCabe、保守性指標、ヒューリスティックなど)
- 品質 (デッド・コード、失敗したジョブなど)
- コスト (週間開発コスト、予定コスト 対 実際のコスト比較など)

これにより、アプリケーション・ソフトウェア品質の概要図が形成され、以下による改善の優先分野を規定できます：

- 簡素化
- 改善
- アップグレード
- 最新化
- 変更
- クラウドへの移行

これは、コスト、時間、リソースの面で、変更影響の見積もりも裏付ける必要があります。

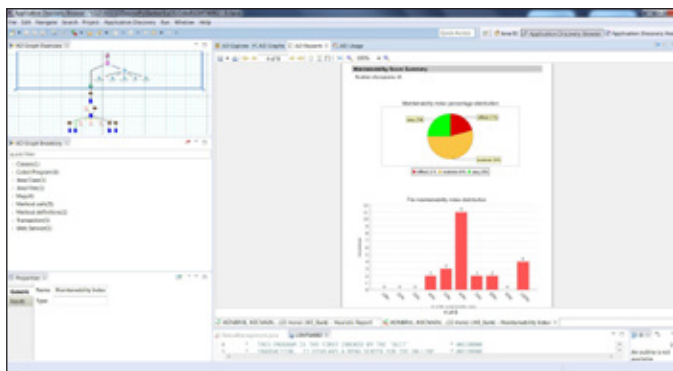


図 4: Application Discovery 健全性評価

ステップ 2 – 改善

開発およびテスト担当者を巻き込んで、個別、およびアプリケーション・ソフトウェア・ポートフォリオ全体でアプリケーションを改善する最善の方法を決定します。次に、開発コンテキストで改善を適用し、アプリケーションのプラットフォーム、言語およびバージョンにわたる変更の複雑性を処理するには、統率と正確な解析情報および自動化されたツールが必要になります。開発チームとテスト・チームは共通のレポジトリを必要とし、これは、開発、テストおよび実稼働環境にわたるアプリケーション・コード・コンポーネントの常に最新情報に更新されたビューに加え、職務に応じたロールベースのビューを提供します。

共通のツールキットを使用することで、開発者は、組織の特定の以下のものに常に準拠できます：

- 特に大規模な基幹システムでのアーキテクチャ上の制約
- コード標準化
- 共通のサービスとインターフェイス・メカニズムの使用
- アップデート・メカニズムを含むデータの一貫性のある使用
- コンプライアンス規制

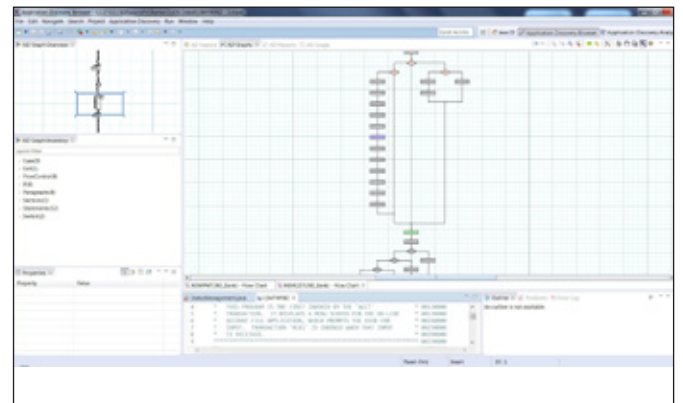


図 5: Application Discovery ブラウザー

「使用箇所」タイプの解析を実行し、影響を評価できることで、開発者は影響を受けたコンポーネントに集中することができ、統合テスト時にのみ構造的な欠陥が判明するリスクが低減できます。

どのアプリケーション・コンポーネントが変更によって影響を受けるかを理解できることで、テスターはより包括的なテスト計画を策定し、コード・カバレッジ・データを解析するおとにより、計画を最適化できます。

技術チームは、異なる技術プラットフォームにわたって情報を交換しアプリケーション・ソフトウェア改善を実現できます。これにより、変更を並行して実行し、リリースのためにパッケージ化できるため、統合、リスクベースの回帰、およびユーザー・アクセプタンス・テストのためのより正確な基礎が提供されます。

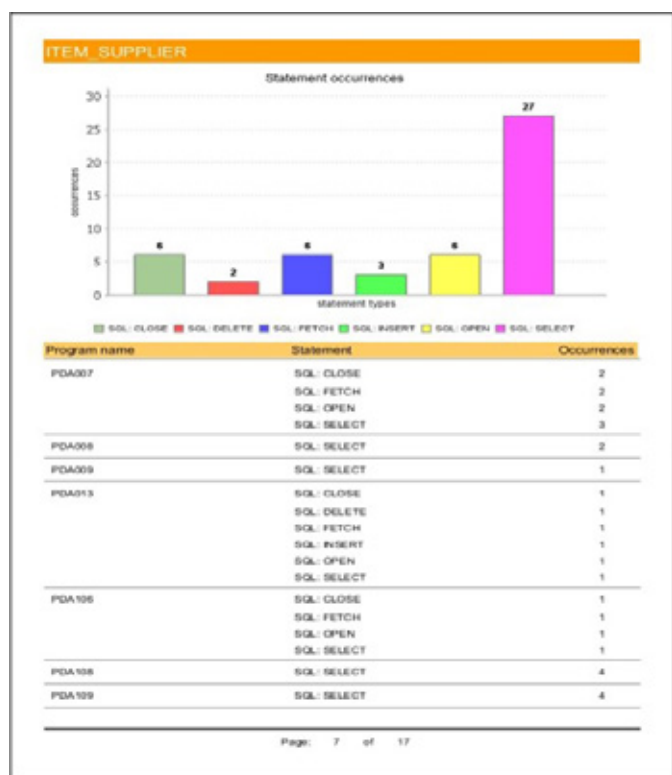


図 6: 「使用箇所」解析

変更を基にしたコード解析への自動化された更新を使用することにより、開発、テストおよび実稼働環境にわたる品質および整合性指標を簡単に規定することも可能になります。これは特に、コア・アプリケーション保守と改善活動が、単一または複数のプロバイダーにアウトソースされている組織での品質の維持に有効です。

Report summary		
Rule	Failure(s)	Weight
Limit IF Blocks	52	52
Limit Nested IFs	24	24
TOTALS	76	76
Failure(s)		
Name	Rule(s) failed	Occurrence(s)
ACCT01	1	7
MJ00090	1	5
MJ04280	1	6
MJ04290	1	4

図 7: コーディング・ルールへの準拠性を監視

また、これにより、アプリケーションを所有する IT 組織は、アウトソース・プロバイダーについて以下を検証することが可能になります:

- アーキテクチャ標準に適合している
- コーディング標準に適合している
- 品質の改善を図っている

これらのツールと技法を制度化することで、IT 部門は、未来頭痛の種となるレガシーではなく、今日の「ソフトウェア資産」を提供することができます。

アプリケーションの構造的品質問題が根本原因となって、サービス停止、欠陥、パフォーマンス低下およびセキュリティ違反が発生することが多くあります。

自動化された深い理解と影響解析能力なしには、検出はほぼ不可能です。

ステップ 3 – 評価

アプリケーション・ソフトウェア品質を評価するために、アプリケーション・コード・コンポーネントの技術的な深い知識のある IT マネージャーに頼る必要はなくなります。これにより、不完全で最新版でない設計書に頼る必要性も低くなり、人事異動によってアプリケーションに関する知識全体が失われるという最悪のシナリオを考える必要性もなくなります。

自動化された解析レポート・ツールと、解放および権限付与ステップで生成され、コードに品質改善が加えられると同時に自動的に更新される解析結果の一元的リポジトリを基に、さまざまなレベルの IT マネージャーは、これらの改善が実際にどのようなビジネス価値を生み出しているかを認識できます。

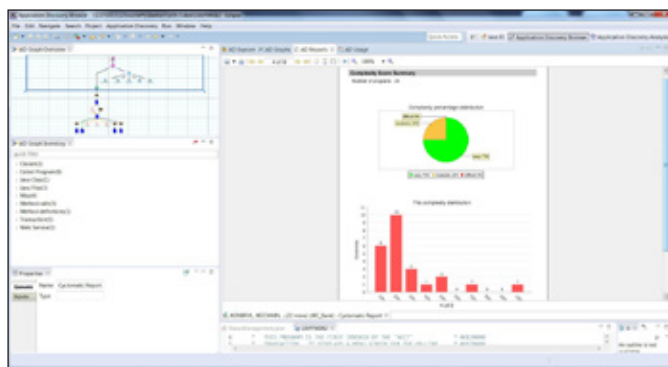


図 8: Application Discovery のレポート自動化機能

以下のような、運用およびコード・カバレッジ指標を使って品質および複雑性に関する主要なアプリケーション・ソフトウェア指標を組み合わせ、相互に関連づけることによって:

- 使用頻度
- リソースの使用状況および
- パフォーマンス

現在の状態だけでなく、長期的に IT 組織が改善しているかどうかを理解できます。これにより、以下が実現します:

- 新しいアプリケーション・プロジェクトとプログラムへの投資の優先順位付け
- 時間、コストおよびリソースに関するプロジェクト確実性の改善
- 保守からビジネス改善への IT 予算の移行
- 提供されるバリューに関するビジネスと IT 間の対話の改善

当然ながら、アプリケーション・ソフトウェア品質の管理は継続的なプロセスで維持する必要があります。アプリケーション・ソフトウェア品質管理のための堅牢な専用ツールを使って成熟した管理プロセスを運用する組織は、事業組織に明らかなバリューを提示します。

長期的に積極的に品質管理を監視および管理することは、IT マネージャーがアプリケーション再作業を減らし、納品までの期間を短縮し、パフォーマンスをより正確に予想する上で不可欠なプロセスです。

提言

組織内でのアプリケーション・ソフトウェア品質の低下の原因がなんであれ、それを修正するコストは膨大です。

品質の脆弱性は主に以下の 2 つが原因です:

- 明確なアーキテクチャ標準およびコーディング標準の欠落、またはそれへの違反とそれに伴う構造欠陥
- リリースからリリースへ既存のコード問題が引き継がれた結果、新しいアプリケーションと正しく統合できなかったり、以前には露呈しなかった欠陥が判明する

堅牢なソフトウェア品質管理プロセスを実装し、過去のミスを修正する機能を提供し、再発を予防する補完ツールキットを使用することによって、過去の多くの過ちや誤った意思決定が繰り返されることがなくなります。

ソフトウェア品質管理を改善する対策を今日行い、IBM Application Discovery の 3 ステップ・アプローチを使用することで技術損失と将来の開発および保守コストを大幅に削減できます:

1. アプリケーション・ソフトウェア・ポートフォリオの見える化
 - a. すべてのアプリケーションとコンポーネントおよび相互依存性を理解することによるベースラインの確立
 - b. 改善の優先順位を付ける
 - c. 改善の目標を設定し、指標を規定する
 - d. 改善を計画し、コストを現実的に評価する

2.アプリケーション・ソフトウェアの品質改善に開発チームとテスト・チームを巻き込む

- a. 可能な限り簡素化と最新化を図る
- b. できない、またはよりよい解決策がある箇所を転換する

3.ソフトウェア品質を継続的に評価する

- a. 長期的な改善を追跡/監視
- b. ビジネス・ニーズに指標を調整および適応させる
- c. 特定のコードではなく、アプリケーション・ソフトウェア・ポートフォリオ全体を視野に入れる

高品質ソフトウェアとは、高品質コード以上のものです。

自動化されたツールとテクノロジーを使用して、ソフトウェア品質を評価、改善、管理することでアプリケーション開発コストと保守コストを徹行 50 % 削減できます。⁵

このソフトウェア品質管理への 3 ステップ・アプローチは、IBM Application Discovery Solution によって自動化および支援されることを全体に設計されています (図 9 参照)。

IBM Application Discovery を使用してソフトウェア品質管理を実行することで、顧客のアプリケーション保守および開発活動コストが 30 % 以上削減できました。⁶



図 9: IBM Application Discovery Solution のソフトウェア品質管理のための 3 ステップ・アプローチ

z Systems について

IBM z13 (z13) の詳細については、日本IBM 営業担当員または IBM ビジネス・パートナーにお問い合わせいただくか、次の Web サイトをご覧ください。 ibm.com/systems/z13

また、IBM グローバル・ファイナンスは、ビジネス・ニーズに合った費用対効果の高い戦略的 IT ソリューションの調達を支援します。IBM グローバル・ファイナンスは、信用審査で承認されたお客様のために、IT 融資ソリューションをビジネス要件に合わせてカスタマイズし、効果的なキャッシュ管理を可能にして総所有コストを改善します。IBM グローバル・ファイナンスは、重要な IT 投資資金を調達しビジネスを前進させる上で、スマートな選択となります。詳細については、次の Web ページをご覧ください: ibm.com/financing



© Copyright IBM Corporation 2017

IBM Systems and Technology Group
Route 100
Somers, NY 10589

Produced in the United States of America
October 2016

IBM、IBM ロゴ、ibm.com、および z Systems は、世界の多く国で登録された International Business Machines Corporation の商標です。他の製品名および サービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、「著作権と商標情報」
ibm.com/legal/copytrade.shtml でご覧いただけます。

Java およびすべての Java 関連の商標とロゴは、Oracle やその関連会社の商標または登録商標です。

本資料は最初の発行日の時点の内容であり、予告なしに変更される場合があります。本資料に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、プログラム、またはサービスについては、日本 IBM の営業担当員にお問い合わせください。

本資料の情報は「現状のまま」提供され、商品性、特定目的への適合性に対する保証、および非侵害の保証または条件を含め、いかなる明示的または黙示的な保証も行いません。

日本 IBM 製品は日本 IBM 所定の契約書の条項に基づき保証されます。

IBM の未来の方向性および意図に関するすべての声明は、事前の告知なく変更または撤回されることがあります。

確実なセキュリティ体制への取り組みについて:IT システム・セキュリティには、社内外からの不適切なアクセスに対する予防、検出、対応によってシステムと情報を保護する対策が伴います。不正なアクセスにより、情報が改ざん、破壊、または不正利用される可能性があり、システムへのダメージや他者への攻撃といったシステムの悪用が生じることがあります。IT システムまたは製品によってセキュリティ対策が万全になると考えることは危険であり、1 つの製品またはセキュリティ対策で不正アクセスを完全に有効に防ぐことはできません。IBM のシステムと製品は、包括的なセキュリティ・アプローチの一部として設計されています。そのため、運用手順を追加することがどうしても必要となり、効果を最大限に高めるには、他のシステム、製品、サービスが必要になることがあります。IBM は、システムと製品が他者による悪意のある行為または不正行為から免れることを保証するものではありません。



リサイクルにご協力お願いします

1 Kyte, Andy, "Measure and Manage Your IT Debt," Gartner, Source: RAS Core Research Note G00205265, 9 August 2010; http://imagesrv.gartner.com/media-products/pdf/cast_software/gartner2.pdf

2 "Gartner Says Optimizing Application Development and Maintenance Can Cut Costs by More Than 50 Percent", Gartner Application, Architecture, Development & Integration Summit 2014, May 19-20 in London, U.K. and July 21-22 in Sydney, Australia, <http://www.gartner.com/newsroom/id/2711017>

3 Worldwide Software Quality Analysis and Measurement 2015-2018 Forecast and 2013 Vendor Shares: Code Visibility Enables Quality with Security to Drive Market Growth, Global Information Premium Market Research Reports, 1/16/2015, <https://www.giiresearch.com/report/id220931-worldwide-software-quality-analysis-measurement.html>

4 Mary and Tom Poppendieck, "Implementing Lean Software Development: From Concept to Cash", 9/17/2006.

5 Gartner op. cit, 2014.

6 Percentage number based on surveys conducted by ezSource, now z Systems.