PreEmptive Solutions

アプリケーション インストルメンテーション：*ビジネスの改善、迅速化、収益増加のための処方箋*

2008 年 7 月 15 日

はじめに

このホワイトペーパーはアプリケーション インストルメンテーションの定義を、単なる品質管理技術というものから、ビジネスのパフォーマンスを改善し収益を増加させるためのもの、と改めます。この拡大された解釈において、ISV、システム インテグレーター、および企業は、アプリケーション開発の投資に対する収益を著しく増加させることができます。

アプリケーション開発におけるインストルメンテーションとは、アプリケーション実行時の挙動の監視と、ある種のインシデント改善を自動化するロジックを含めることを意味します。インストルメンテーションはこれまで、開発者や運用管理者がアプリケーションのパフォーマンス、信頼性、拡張性、セキュリティを監視するために使われてきました。インストルメンテーションはさらに、CRM、BI、BPM サービスと統合することにより、企業のリスク低減、営業サイクルと導入曲線の短縮、IT ガバナンスの自動化をすることができ、供給側と消費側双方の、アプリケーション ROI（投資利益率）を劇的に改善します。

ステークホルダー 組織	エンジニアリング利用以外のインストルメンテーションの使用例
独立系ソフトウェア ベンダー（ISV）	• ベータ版、評価版、製品導入における機能レベルでのトラッキング
付加価値販売業者（VAR）	• 見込み客への活動とリンクした、ソフトウェア評価プロセスの管理 • アプリケーションとプラットホームの使用状況に応じた、積極的な顧客サポート
システム インテグレーター（SI）	• ソフトウェア導入とビジネス プロセスの成果を結びつける、ビジネス インテリジェンス（BI）およびパフォーマンス管理（BPM）ソリューション
企業ユーザー	• アプリケーションと Web サービスのポートフォリオ管理 • 運用と IT ガバナンス統制のエンコードおよび自動化

表 1：典型的な品質やパフォーマンス以外でのインストルメンテーションの使用例

アプリケーション ライフサイクルにおけるインストルメンテーション

インストルメンテーションのロジックはソース コードの内部に含めることも、ビルドされたバイナリ（ソース コードがコンパイルされた後）に挿入することもできます。このホワイトペーパーで示している機能、能力、および利点は、実行可能なバイナリをビルドした後にインストルメンテーションのロジックを挿入する、後者の場合について紹介しています。

図 1：ビルド後（または導入後）にインストルメンテーションを行うアプリケーション ライフサイクル

インストルメンテーションは、ソース コードにアクセスすることなく、ビルド フェーズの後に挿入を行うことで、利用することできます。この技術は、現場の個別の実行可能モジュールに対しても適用することができます。

ビルド後インストルメンテーションのユニークな利点

プロジェクトの要件収集の段階においては、どのようなソフトウェア開発ライフサイクル（SDLC）手法を使うかにかかわらず、膨大な数のビジネス上の要件は認知されず、また場合によっては知られずに終わります。たとえば以下のようなものがあります。

- 複数の配置サイトにまたがる、IT ガバナンス ポリシーのテストと施行（認知されていないだけではなく、他と矛盾するものもある）
- SDLC の異なるフェーズにおけるアプリケーション インスタンスの失効ポリシー（例：コンセプト検証、ベータ版、評価版、製品版）
- 複数のビジネス インテリジェンス（BI）またはビジネス プロセス マネジメント（BPM）システムの利用とデータ統合

実行中のアプリケーションのある時点での挙動を可視化する（および左右する）というビジネス上の大きな可能性は、店舗販売でのポイント オブ セールス（POS）ソリューションの導入による変革や、Web コマースにおける Web 分析に似ています。

ビルド後インストルメンテーションの使用例

ビルド後インストルメンテーションでは、ビルド後の任意の時点で、個別のアプリケーション インスタンスに対して必要なロジックを選択して挿入することにより、以下および類似する "独立した" ビジネス要件を扱うことができます。

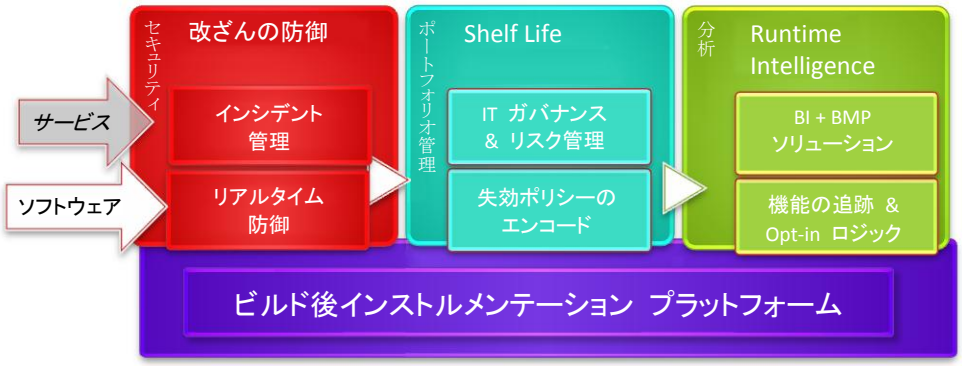


図 2：ビルド後インストルメンテーション プラットホームの機能概要

図 2 は、ビルド後インストルメンテーション特性のユニークな例を紹介しています。インストルメンテーションに、以下の処理を行うソフトウェアの挿入が追加されたことに注目してください。

a) 新しいランタイム アプリケーションの挙動をエンコード

b) 従来のインストルメンテーションによる監視を行うだけでなく、ソフトウェア サービスに対し実行時データを送信

改ざん防止

アプリケーション ステークホルダーは以下により、難読化や他のコード保護技術を補完できます。

- アプリケーションの運用目的および IT ガバナンスの制約用途にあったカスタム ロジックを使って、アプリケーションの改ざんをリアルタイムに検知し防止するランタイム ソフトウェア
- サプライヤ、企業ユーザーの双方に安全な信号を送り、改ざんインシデント管理を開始するためのソフトウェア サービス

Shelf Life

アプリケーション ステークホルダーは、アプリケーション内に使用失効時の独自の挙動をエンコードすることにより以下のコンポーネントを有効にします。

- サイトごとの異なる挙動で使用期限を自身で規制するランタイム ソフトウェア
- IT 運用チーム、サポート、営業管理などにアプリケーションのインベントリ管理を提供するソフトウェア サービス

Runtime Intelligence サービス

アプリケーション ステークホルダーは以下のものを挿入できます。

- 機能レベルでのトラッキング、ユーザー プロファイリング、実行時の環境の統計、法律で定められたオプトイン(またはオプトアウト)機能を生成するためのランタイム ソフトウェア
- アプリケーションの導入と挙動をほぼリアルタイムに可視化するソフトウェア サービス。このインテリジェンス サービスは、CRM、BI、ERP、および IT 管理ソリューションと統合することができます。

インストルメンテーション プラットホーム評価基準

ビルド後インストルメンテーションによって、アプリケーションの役割と価値を一変させる可能性を大げさに述べることはできません。しかし、成功を確実なものにするためには、依存関係と運用要件を全面的に考慮することが重要です。

ビルド後インストルメンテーション

IL(中間言語)の直接操作は .NET や Java ではサポートされていますが、依然としてある種の技術と専門知識を要する特殊な作業です。成功を確実なものとし、最高の品質とパフォーマンス標準を確保するために、次の項目は必須または強く推奨されます。

IL 適正、最適化、リンク

- インストルメンテーションのプラットホームは、サポートされるすべてのバージョンと同様にすべての種類(モバイル、デスクトップ、web、マイクロ... をターゲットとする)のマネージ コード フレームワークをサポートする必要があります。
- インストルメンテーションはコードを追加します。また多くの場合ファイルも追加するため、既存の実行可能ファイル(DLL や JAR など)にコードをリンク(または連携)する能力、また、できあがった実行可能ファイルを刈り込む(または小型化する)能力も高く推奨されます。

IDE 統合

- Visual Studio 2003～2010 のすべてのバージョン、Eclipse、その他のサポート
- MSBuild、ANT、その他との統合
- コマンド行インターフェース
- スタンドアローン UI
- カスタム属性および注釈が使用できること

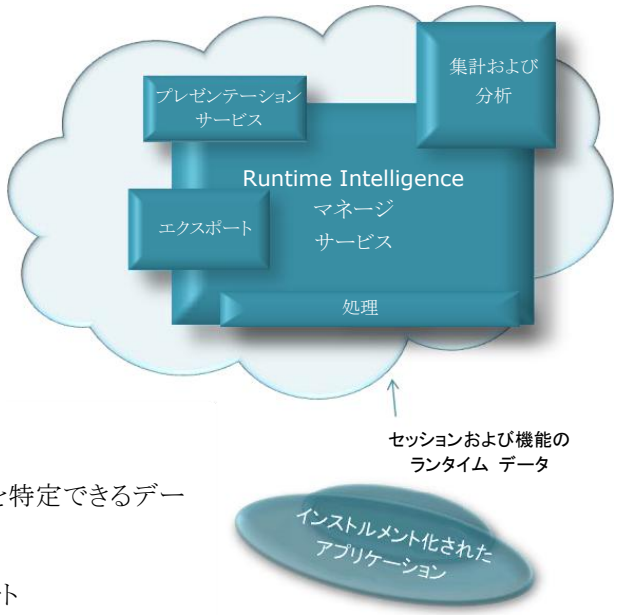
エンドポイント アーキテクチャ

データ パケットの受信は簡単で単純な作業に思えますが、今日の潜在的に敵対するコンピューター環境においてスケーラブルかつマルチテナントをサポートするエンドポイントを構築するためには、多大なエンジニアリング努力を必要とします。

アーキテクチャ、スケーラビリティ、セキュリティ

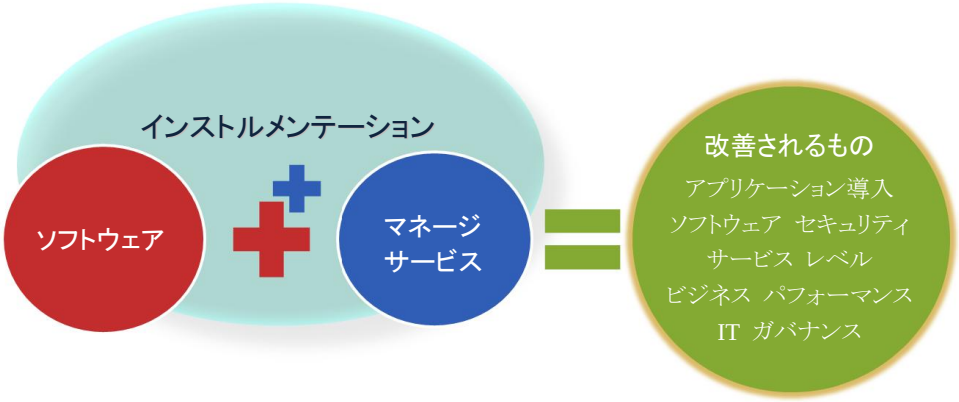
持続可能な商業レベルのエンドポイントを提供するためには、次の機能について考慮する必要があります。

- サプライヤとユーザーの両方に、共通の Runtime Intelligence へのアクセスをサポートするマルチテナント
- パフォーマンスへの影響を最小限にするための "撃ったらそのまま"(Fire & Forget)データ転送
- 個人情報保護の法令に準拠するために、個人を特定できるデータを転送しない
- 安全な標準転送を行うための SOAP/SSL パケット
- パフォーマンス最適化のためのキャッシュされたメッセージング



まとめ

マネージ コードとクラウド コンピューティングの融合は、アプリケーション インストルメンテーションが単に監視をするものではなく、アプリケーションの挙動を拡張して、その価値を増加させ、保護するために効果的に適用できるものであるという風土を作り上げています。



あなたのビジネスが販売、インテグレータ、あるいは単にアプリケーションに依存しているだけであっても、ビルド後インストルメンテーションは即時に、実質的な利益をもたらします。さらに、インストルメンテーションはソース コードではなく実行可能ファイルに適用されます。これらの特性は進行中の開発に限らず、これまでに書かれたすべてのマネージ コード アプリケーションに適用可能なのです。

PreEmptive Solutions について

PreEmptive Solutions は、Dotfuscator および DashO のインストルメンテーションと難読化製品ファミリ、および Runtime Intelligence アプリケーション分析サービスを製造しています。PreEmptive Solutions は、アプリケーションのセキュリティおよび分析を開発ライフサイクルに組み入れる最良のソリューションを提供することができます。

3,000 を超える企業のお客様と 100 カ国以上における 40,000 の登録インストールに対して、また、Microsoft 社の 6,000,000 を超える Visual Studio シートに対して、PreEmptive Solutions は、ソース コードの保護や、アプリケーション セキュリティ、分析、および収益化に真剣に取り組んでいるあらゆる組織にとって明確な選択肢となっています。