Forrester Total Economic Impact™ 調査

委託元：

Synopsys

プロジェクト責任者：
Liz Witherspoon

2016 年 9 月

Synopsys のソフトウェア・テストツール：Coverity および Defensics に関する Total Economic Impact™ レポート

FORRESTER®

目次

FORRESTER CONSULTING について

Forrester Consulting は、組織からの委託により第三者機関として客観的な調査を行い、これに基づくコンサルティングを提供することで事業の成功を支援しています。短期の戦略セッションから個別のご要望に応じた長期のプロジェクトまで、専門知識と経験が豊富な Forrester Consulting のリサーチ アナリストが直接お客様に対応し、それぞれのビジネスに関する課題について専門的な知見を提供いたします。詳細につきましては、forrester.com/consulting をご覧ください。

FORRESTER®

エグゼクティブ・サマリー

Forrester Consulting は Synopsys の委託を受け、Coverity および Defensics の導入による投資対効果 (投資利益率、ROI) について Total Economic Impact™ (総合的経済効果、TEI) の調査を実施しました。本調査の目的は、これらのツールへの移行によって企業にもたらされる経済効果の評価フレームワークを提供することです。

Coverity と Defensics を組み合わせて導入すると、不具合やセキュリティ脆弱性の対策費用を低減でき、製品発売後の問題発覚による大きな悪評や高額な対応費用が発生する事態を予防しやすくなります。また、テストツールは新製品の市場投入を迅速化するために効果的です。

今回のインタビュー対象企業では、次のようなベネフィットが得られました。

- **3 年間で約 950 万ドルの利益 (現在価値)**
- **開発フェーズ中、不具合/脆弱性の早期発見により、対策費用が 5 分の 1 に低減**
- **テスト・フェーズ中、不具合/脆弱性の早期発見により、対策費用が 2 分の 1 に低減**
- **新製品の市場投入に必要な期間が 25% 短縮**

Coverity および Defensics の導入によるベネフィット、費用、およびリスクについて十分に把握するために、Forrester は、これらのセキュリティ・テストツールを既に複数年運用した経験を持つ既存顧客企業 (以下、X 社) のテスト・エンジニア複数名にインタビュー調査を実施しました。Coverity は静的解析ツールです。ソースコードに含まれる品質上の不具合やセキュリティ脆弱性を、ソフトウェア開発ライフ・サイクル (SDLC) の早い段階で発見して修正し、対策の手間と費用を抑えることができます。Defensics は、テスト環境内で未知の脆弱性を動的にトリガーして検知するファジング・テストツールです。ゼロデイ攻撃に備える先制策として使用できます。いずれのツールも、製品が問題を抱えたまま発売されるのを防ぎ、本番環境内で高額な対応費用が発生する事態やセキュリティ攻撃のターゲットにされる事態を回避する目的で使用されます。また、両ツールとも、SDLC のうち複数の段階において、コードに含まれる不具合/脆弱性の対策関連費用を低減する効果を発揮します。

Coverity および Defensics の導入以前、X 社は、ソフトウェアのセキュリティ脆弱性の原因となり得る不具合を発見・修正するためのシステム的アプローチやツールを持っていませんでした。旧来のソフトウェア開発ライフ・サイクル・アプローチに頼り、テスト・スクリプトの開発・実行は品質保証テスト担当チームに任せていました。最終製品がインターネットからのアクセスを受け付ける必要がなかった頃は、この体制でも問題ありませんでした。しかし、モノのインターネット (IoT) の登場によって情勢は変わりました。製品は、ネットワーク上に存在する様々な新しいセキュリティ脆弱性の問題と向き合うことになったのです。それだけではありません。コンシューマー通信機器サプライヤー X 社の重要顧客である大手電気通信事業者が、契約要件を満たすテストツールの提供を要求してきました。X 社は、合わせて年間売上高の 35% を占める最重要顧客 2 社のために、契約要件に合ったテストツール群を用意する必要に迫られました。そこで Coverity と Defensics を採用したところ、X 社はファジング・テストツールおよび静的解析ツール群を短期間で実装することができました。それらのツールによって、もし見逃せば膨大な出費と大きな悪評が発生するような不具合やセキュリティ脆弱性を、即座に検知できるようになったのです。あるシステムインテグレーション・テスト・エンジニアは次のようにコメントしています。「従来は、お客様の施設で何かが起きない限り、当社製品のセキュリティ侵害など発生しようがありませんでした。ところが今は、インターネット接続経由で攻撃される可能性がありますから、当社製品もしっかりセキュリティ対策を施す必要があります。Synopsys は、当社のリスク・エクスポージャーを減らし、お客様との契約やビジネスを失わずに済むために役立っています」

不具合/脆弱性対策費用を低減し、市場投入を迅速化する SYNOPSYS

Forrester は、システムおよびインテグレーション・テスト・エンジニアのリーダー 5 人をインタビュー調査し、その結果に基づいてモデル組織を作成した後、費用分析を実施し、X 社における ROI、ベネフィット、および費用を算出しました (図 1 参照。数値はリスク調整済み)。[1]

Coverity と Defensics によって、X 社は、ソフトウェア開発フェーズとテスト・フェーズの両段階で、不具合/脆弱性の発見および対策に要する費用の低減を実現しました。また、開発チームとテスト・チームが行う新しいコード・ベースの追加や既存コード・ベースのメンテナンスについても、作業の費用効率が向上しました。さらに、会社のビジネス中断回避による費用低減効果と、コンシューマー製品の市場投入期間の短縮効果が現れました。

図 1
費用ベネフィット解析の結果 (3 年間、リスク調整済み)

資料： Forrester Research, Inc.

静的解析およびファジング・テストの定義

この調査結果の背景を理解していただくために、Synopsys のツールが実行する 2 種類のテストと、それらによって検知される不具合の種類について定義を示すことにします。

静的解析

「静的解析」とは、アプリケーションを実際に実行することなく、アプリケーション・コードを検査してセキュリティ上の潜在的な不具合を発見するテクノロジーです。この機能は、ソースコードの解析や実行可能バイナリの解析を行い、攻撃者の手口を検討することで実現されます。攻撃者は、自己の目的のために様々な手口を使って、データの侵害、プログラムのクラッシュ、アプリケーションの状態操作などを引き起こす可能性があります。[2]静的解析の特徴は次のように説明できます。

- **「ホワイトボックス」手法によってセキュリティ上の不具合を発見。** 静的解析ツールは、ソースコードを参照してアプリケーションの内部からセキュリティ上の弱点を見つけ出します。これと対照をなすのが、侵入テストなどの「ブラックボックス」手法です。ブラックボックスの場合はコードを参照せず、動作中のアプリケーションに対する攻撃者の行動を外部からシミュレートすることでアプリケーションをテストします。北米においては、エンタープライズ企業の半数近く (45%) が、アプリケーションのセキュリティ評価手段として、ホワイトボックス/ブラックボックスの両手法によるアプリケーション・セキュリティ・テストツールやスキャンツールを使用しています。
- **費用効率性に優れたコード不具合除去ツール。** ブラックボックス侵入テストと違い、静的解析ツールは開発段階でコードの不具合を取り除くために役立ちます。普通、この方法はリリース後のバグ修正よりも費用効率に優れています。エンタープライズ企業は、コード品質向上のインセンティブに基づいて静的解析ツールを採用しています。たとえば、コンプライアンス、セキュリティ要件などが静的解析ツール導入の促進要因です。

ファジング・テスト

「ファジング」とは、ソフトウェア断片に異常な入力をわざと送り込み、誤動作するかどうかを調べるプロセスです。個々の異常入力を「テストケース」といいます。誤動作はバグが発見されたことを意味し、そのバグを修正することでターゲット・ソフトウェアの堅牢性やセキュリティが改善されます。ターゲット・ソフトウェア断片に対してテストを実行するソフトウェアを「ファザー」といいます。ファザーとしての使いやすさを実現するには、記録機能、対応行動を示すレポート機能、修正作業のためにスムーズな誤動作再現プロセスを実現する機能などが必要です。

不具合とセキュリティ脆弱性

本調査において、「不具合」または「バグ」とは、ソフトウェアのあらゆる誤りや不備のうち、不正確な結果や予期しない結果の発生原因となるものを指す広い意味の用語です。セキュリティの「脆弱性」とは、攻撃者による情報盗用、クラッシュ、ソフトウェア乗っ取りなどを可能にする特定の不具合を意味します。不具合と脆弱性は、いずれもビジネスにとって明確なリスクであり、予防的手段を講じて事前に発見・修正することが最善の対応策であると考えられます。

本調査において説明するベネフィットと費用

本調査では、Coverity および Defensics に関する以下のベネフィット、費用、将来価値について取り上げます。

- **ベネフィット**： インタビュー調査対象である X 社の場合は、以下のようなベネフィットが得られました (リスク調整済み数値)。
  - **静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — 既存コード・ベース**：8 つの既存コード・ベースに関して、コーディング・フェーズで発生した不具合/脆弱性対策費用は 386 万ドル低減されました (現在価値)。しかも、Coverity による誤検出率はわずか 15% 程度に抑えられました。
  - **静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス**：X 社は、テストツール Coverity を静的解析に使用することの主要なベネフィットとして、新しいコード・ベース内に見つかる不具合/脆弱性の継続的な対策費用や、既存コード・ベースのメンテナンス・費用を抑制できることを挙げています。これによる費用低減効果は 230 万ドルでした (現在価値)。
  - **ファジング・テストによる不具合/セキュリティ脆弱性対策費用低減効果**：X 社では、Defensics を初めてテスト実行したとき、本番運用を目前に控えた製品のファジング・テストで 40 件の不具合/脆弱性が見つかり、対策を施すことができました。それらの不具合の対策に要した人時は、リリース後フェーズに対策を行う場合と比較して 2 分の 1 でした。
  - **新製品の市場投入に要する期間の短縮効果**：ファジング・テストと静的解析の採用により、X 社では、製品リリース・サイクルを 12 か月から 8 か月に短縮し、その分のリソースを他の生産的な活動に振り向けることが可能になりました。
  - **セキュリティ・テスト体制の改善によるビジネス中断回避効果**：Defensics によるファジング・テストを導入したところ、X 社は、不具合/脆弱性を含んだ製品をリリースして評判を落とす事態を回避できました。Synopsys のテスト・スイートを採用する前には、そのような事態が実際に発生していました。
- **費用**：調査対象の X 社では、次のような費用が発生しました (リスク調整済み数値)。
  - **Coverity のソフトウェア・ライセンス料。**この費用は継続的に発生します。開発チームが作成したコード行数に基づいて、1 年ごとに Synopsys に支払われます。
  - **Defensics のソフトウェア・ライセンス料。**この費用は継続的に発生します。Defensics の使用ライセンス本数に基づいて、1 年ごとに Synopsys に支払われます。
  - **オンサイト・コンサルティング・サービス料。**これには、Coverity および Defensics の実装とインテグレーションに関する支援費用が含まれます。
- **柔軟性**：調査対象の X 社では、柔軟性に関して次のようなベネフィットが発生しました。
  - **再利用コードの不具合密度の低下**：コード・ライブラリを構築すると、そのコードは再利用されるようになります。X 社では、コード・ベース内の不具合密度が低下したことにより、エラーを含んだコードの再利用で発生する将来的な費用を回避できました。
  - **総合的なリスク・エクスポージャーの低減**：X 社に対して実際にセキュリティ攻撃が行われた形跡はないものの、セキュリティ脆弱性によって防御に隙が生まれる恐れは常にあります。Coverity と Defensics は、将来的なリスク・エクスポージャーの低減効果を発揮しています。

情報開示

本レポートは次の点に留意してご参照ください。

- 本調査は Synopsys からの委託により、Forrester Consulting が実施しました。本調査は比較分析を目的としたものではありません。
- Forrester は、他組織が得る潜在的な投資利益に関しては何の予測も行っておりません。読者の皆様には、このレポートで採用されているフレームワークの枠内で御社自身の見積もりを行い、Synopsys の Coverity および Defensics 製品に対する投資の妥当性を判断されることをお勧めします。
- Synopsys は本レポートの内容をレビューし、Forrester にフィードバックを提供しましたが、本調査および調査結果については Forrester がこれを編集・管理する権限を有し、調査結果と矛盾する変更や調査の趣旨が曖昧になるような変更は一切行っておりません。
- インタビュー調査は Synopsys から紹介された顧客に対して行いましたが、Synopsys はインタビュー調査に一切関与していません。

TEI のフレームワークと方法論

概要

Forrester は、インタビュー調査により得られた情報に基づき、Coverity および Defensics の導入を検討している組織向けに、Total Economic Impact (総合的経済効果、TEI) を評価するためのフレームワークを構築しました。このフレームワークの目的は、投資に関する意思決定に影響を与える、費用、ベネフィット、柔軟性、およびリスクを明らかにすることです。

調査方法

Forrester は多層的なアプローチによって、Coverity および Defensics が企業に及ぼす影響を評価し (図 2 を参照)、次のような段階を踏んで調査を行いました (図 4 参照)。

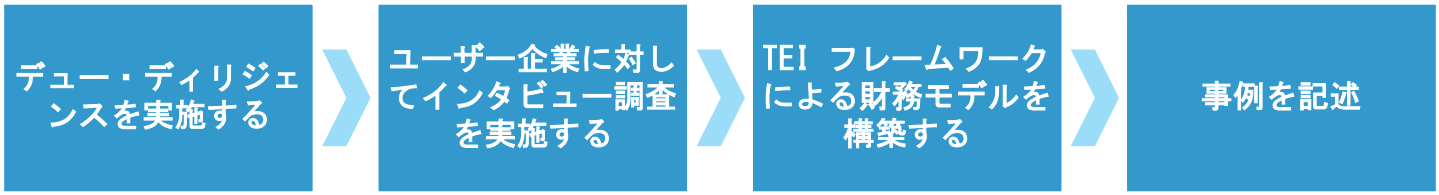
- Coverity および Defensics 製品の情報と、これらの製品の市場全体に関する情報を収集するため、Forrester のアナリストの協力を得て、Synopsys のマーケティング、営業、コンサルティング担当者に対してインタビュー調査を実施しました。
- Coverity および Defensics を現在使用している組織 1 社において、品質保証テストのリーダー複数人にインタビューし、費用、ベネフィット、およびリスクに関するデータを入手しました。
- 調査により収集したデータに基づき、TEI 法による財務モデルを構築しました。この財務モデルに、インタビューで得られた費用と利用価値のデータを追加しています。
- インタビュー対象組織から指摘された問題や懸念事項に基づいて、財務モデルにリスク調整を施しました。TEI 法ではリスク調整が非常に重要になります。Forrester はインタビュー対象の組織から費用とベネフィットの見積データを受け取っていますが、回答に大きなばらつきがある項目や、様々な外的要因が業績に影響したと考えられる項目も多いため、リスクを踏まえて費用とベネフィットの数字を一部調整する必要があります。リスク調整の詳細については各項で説明しています。

Synopsys 製品の影響をモデル化する作業には、4 つの TEI 基本要素(ベネフィット、費用、柔軟性、リスク) が使われています。

IT 投資の費用ベネフィット解析/ROI 解析については、組織内でも高度なテクニックを駆使するようになってきましたが、Forrester の TEI 法は、導入の意思決定がどのような経済効果につながるかを総合的に把握することを目的とした、非常に有効な方法です。TEI 法の詳細については、付録 A をご覧ください。

図 2

TEI 法



資料： Forrester Research, Inc.

分析結果

インタビュー調査対象組織の詳しい状況

本調査のインタビュー対象となった組織 (X 社) は、エンターテインメントおよび通信のテクノロジー企業です。TV とインターネットのサービス・プロバイダーやコンテンツ・プロバイダー向けに、ハードウェア、ソフトウェア、サービスを提供しています。取扱製品は、セットトップボックス、デジタル・ビデオ、インターネット・プロトコル TV (IPTV) 配信システムと、クラウド/ネットワーク/家庭でコンシューマーへのリーチを実現するケーブル機材です。幅広い製品群を手がけ、事業規模が数十億ドルにもなるグローバル企業ですが、一方では、最大の顧客わずか 2 社に対するビジネスが総収益の 35% を占めています。このリスクは、同社が米国証券取引委員会に提出する 10K レポートに記載され、本調査のインタビューでも指摘されている問題です。2 社との大規模な契約に何か望ましくないことが発生すれば、X 社の収益は大幅に減少しかねません。同社の機材にセキュリティ脆弱性が見つかり、ブランド・イメージが低下する事態になれば、そうした顧客との関係にも大きく影響する恐れがあります。同社は、10K レポートの「リスク」に関する項で次のように述べています。「当社製品に不具合が含まれる場合、それらは業績に重大な影響を及ぼす恐れがある。当社製品の多くは、ハードウェアおよびソフトウェアの両コンポーネントを含んだ複雑なテクノロジー製品である。期待される動作を予想外に妨げるようなバグがソフトウェアに含まれることは異例ではなく、特に早期バージョンにおいては一般的である」

本調査においては、X 社内の品質保証リーダー 4 人を対象として合計 3 回のインタビュー調査を実施しました。対象者とインタビューの内容は次のとおりです。

- セットトップボックスのテストを担当する主任システムおよびインテグレーション・テスト・エンジニア。顧客から求められる要件や結果を踏まえて、テストに関する調査を行い、テストの内容を考案することが主な役割です。テストの作成後には、指揮下のテスト担当者チームのためにテスト実行計画を策定します。チームは、様々な勤務地にいる 15 人のテスト担当者で構成され、Defensics を使って 3 つの製品のテストを実施します。実際のユーザー環境には約 300 万台の製品が設置されています。
- 4 種類ある DSL 製品のテストを担当する、シニア QA エンジニアおよびシステム・テスト担当スタッフ・エンジニア。DSL 製品には毎年 4 ～ 5 回のリリースがあります。製品全体に関するエンドツーエンドのシステム・テストをこのグループで受け持ち、ボックスに搭載されたアプリケーションおよびファイアウォール機能のテストと、第 7 層のテストを実施します。
- すべてのオープン・ソフトウェアとプロプライエタリ・ソフトウェアに関する DevOps およびリリース・エンジニアリング担当のシニア・マネージャー。8 つのコード・ベースに含まれる約 1,100 万行のコードをテストするチームの管理者です。

> 「150 万台のルーターが再起動したとき、お客様は、まず攻撃を疑いました。ルーターが予定外に奇妙な挙動を始めて停止し、5 分間にわたって、150 万戸の家庭でネットワークに接続できない状況が発生したのです。原因は技術部門が実施したルーターのアップグレード作業だと判明しましたが、私たちは、お客様のネットワークがネットワーク外の者によって簡単に止められる恐れがあるという現実を思い知らされました」
>
> ~ DevOps およびリリース・エンジニアリング担当シニア・マネージャー

インタビュー調査の要点

X 社は、攻撃を受けたような (実際には違った) 予定外の事態を経験したことで、最重要顧客との関係を維持するには本格的なセキュリティ・テストツールが必要であることを認識しました。

状況

インターネット/クラウド・ベースのコネクテッド・デバイスを製造することの本質的なリスクに対処するため、X 社は、ソフトウェア開発等セキュリティ・テストに対するアプローチの見直しに取り組むことにしました。もともとはインターネットに接続しない機器を提供するビジネスでやってきた会社ですが、エンターテインメント・システムが続々とインターネットのストリーミングに対応する流れの中、従来の考え方はもはや通用しません。150 万台のルーターに突然の再起動が発生したとき、その現実はいやおうなく明らかになりました。本物のセキュリティ攻撃ではなかったとはいえ、社内の上級エンジニアや、当該製品と顧客アカウントの営業担当を総動員して原因を探る作業を余儀なくされたのです。結局、リブートは臨時のソフトウェア・アップデート作業によるものだったと判明しました。しかし、この出来事の強いショックは顧客との関係に大きな影響を及ぼし、後々にも長い余波を残しました。ファジング・テストツールを早急に導入することが必要不可欠であるという点で顧客と合意した X 社は、以下のような考え方でソリューションを模索しました。

- リリースの前に製品をテストできること。たとえ個別のコード・コンポーネントが既に完成していてもテストの対象にすること。
- 検知可能なあらゆる種類の不具合と脆弱性を発見できること。しかも、最新の技法とセキュリティ・テストのノウハウを活用して目的を達成でき、確保が難しく人件費がかさむ優秀な人材を社内に抱える必要がないこと。
- 不具合やセキュリティ脆弱性の発見・対策が遅れると大きな費用が発生するため、大半の不具合/脆弱性をソフトウェア開発の早い段階で発見し、対策ができること。
- 大手電気通信事業者とのビジネスを獲得するにあたって求められる要件や基準を満たせること。顧客の要件において、テスト用セキュリティ・ツールキットの使用は必須とされる。顧客も Coverity や Defensics のようなツールを採用しているため、ソフトウェア・テストに関して、購入する製品のサプライヤーにも同等の品質基準を要求している。

問題解決

X 社は、10 年ほど前からソフトウェア開発ライフ・サイクルの要素として Coverity を使用していました。それに加え、2 年前にコンシューマー向けルーターの再起動が 150 万台で発生したことを受け、Defensics を実装しました。2 つのテストツールが同社内に導入された経緯は次のようなものです。

- X 社では、かなり前から Coverity による静的解析を採用しています。あるチームが 2006 年に導入したことがきっかけで、他のチームに広まっていき、Coverity の利用は全社のコード・ベースにまで拡大しました。現在、社内には 8 つのコード・ベースが存在しています。1 つのコード・ベースに含まれるコードの規模は約 150 万行です。
- X 社が Defensics を使い始めたのは 2 年ほど前です。採用の理由は、Defensics が数社の重要顧客で既に使われていたことと、開発済みコードの不具合/脆弱性テストに最も早く対応したファジング・テストツール製品の一つであったことです。
- Defensics のライセンスを購入したことが社内に知れわたるにつれ、異なる製品のチームにも Defensics の利用が広がっていきました。追随したチームの多くは、テストツール選定のために RFP (提案依頼書) を起こしたわけではなく、他のテスト・チームから Defensics を紹介され、社内の良い評価を聞いた上で採用を決めています。

> 「最大のお客様がテストツールを要件に盛り込み、ツールの使用を求めてきたときに、対応をお断りするのは巨大なリスクを伴うことです。別の大手ベンダーが割り込んできて“うちでは使っています”と言われたら、その案件を奪われかねません」
>
> ~ システム・テスト担当スタッフ・エンジニア

分析結果

インタビュー調査の結果、X 社では Coverity および Defensics の導入から以下の効果が得られたことが明らかになりました。

- **開発プロセスの早い段階でコードのエラーを発見し、かつ、誤検出率を低く抑えることができる。**最大のベネフィットは、対策費用が大きくかさむ段階にまで不具合/脆弱性を残さず、ソフトウェア開発ライフ・サイクルの早い段階でエラーが見つかるようになったことです。多数のコード・ベース (8 系統あり、今後も増加予定) と大量のコード (コード・ベースごとに 150 万行程度) を取り扱う開発者が不具合/脆弱性を正確・迅速に発見できるようになったことで、発見がソフトウェア開発ライフ・サイクルの遅い段階になった場合の労働時間の発生による高いコストが抑制されました。
- **コード・ベースの品質、特に再利用コードの品質が向上する。**X 社では、Coverity と Defensics を使うようになったことで、開発者の作成するコードの品質が上がりました。社内ではコードの再利用が広く行われており、多くの製品が共通のコード・ベースに依拠しています。このため、たとえば 30 件の不具合/脆弱性が発覚せずに残存していると、不具合コードの再利用によって 300 件を超えるバグの発生につながる恐れがあります。Coverity と Defensics は、対策を要するエラーの特定に役立つだけでなく、コード・ベース全体の品質向上に貢献しています。これは、両ツールで実現されたワークフローと、社内で使われる他の開発ツールとのインテグレーションによって現れた効果です。
- **稼働現場での障害発生リスク (およびそれに伴う費用) が低減される。**セキュリティ・リスクを抱えた組み込みソフトウェアがコンシューマーの手元に届いてしまうことは大きな問題であり、X 社は、そうした事態の防止策を求める重圧を強く感じていました。予定外の再起動が残した忘れられない記憶の強さは、同社作成の 10K レポート (公開企業に義務づけられた提出書類) にリスクとして明記されていることからもうかがい知れます。運用開始後に障害対策を行う場合、非常に人件費の高い技術系/営業系リソースを即座に確保して集中的な対応に当たらせる必要があるため、大きな費用が発生します。それだけでなく、提供したインターネット対応デバイスを使う側のエンド・コンシューマーにも多大なリスクを負わせることになります。リスクの内容は以下のようなものです。
  - ハードウェアの障害で数百万台のデバイスを交換することが必要になり、会社の財務に多大な損失が生じる恐れがある。
  - 通信用製品のコンシューマーやエンド・ユーザーに関する機密情報が漏れて、脅威となりかねない、悪意を持つ可能性がある者の手に渡る恐れがある。その結果、個人情報や財務情報の紛失・悪用、損害などが発生する恐れがある。
  - 社会の注目を集める大きなセキュリティ問題が発生すると、最大級の顧客 (一般コンシューマーの知名度が高い通信事業者) のブランド・イメージに傷がつく恐れがある。その結果、年間売上高の 35% を占める非常に大規模な契約を失う恐れがある。
- **新製品の市場投入に要する期間が短縮される。**Coverity と Defensics は、いずれも運用に投入される前の段階でソフトウェアの不具合/脆弱性を見つけるツールです。これによって、不具合/脆弱性の発見・対策に要する費用の低減効果に加え、市場への製品リリースの迅速化を実現できる効果が得られました。
- **セキュリティのノウハウがほとんどないエンジニアでもテストを実施できる。**Defensics を導入した結果、セキュリティの専門知識を持たない人 (品質保証エンジニアなど) でも、不具合/セキュリティ脆弱性を発見するネットワーク・テストツールの実行に必要な基本スキルを簡単に習得できるようになりました。したがって、セキュリティに詳しい人材の時間を確保してテストを実施させるための待ち時間や、セキュリティに特化したスキルセットを持つ人材を社外から雇用することが不要になりました。

「時間の節約効果が出るのは、このツールが自動的に動いてくれるからです。実行ボタンのクリック一つで膨大な数のテストケースを自動実行でき、何日、何週間、何か月かかろうと、途中で人間が操作する必要はありません。このテストを 1 件ずつ手動実行することを考えたら、途方もない時間を節約できていることになります。おかげで、テストに以前ほど大きな労力を割く必要がなくなりました。クリック一つでテストを開始し、裏で走らせたまま、別の作業をこなせるからです」

~ システム・テスト担当スタッフ・エンジニア

ベネフィット

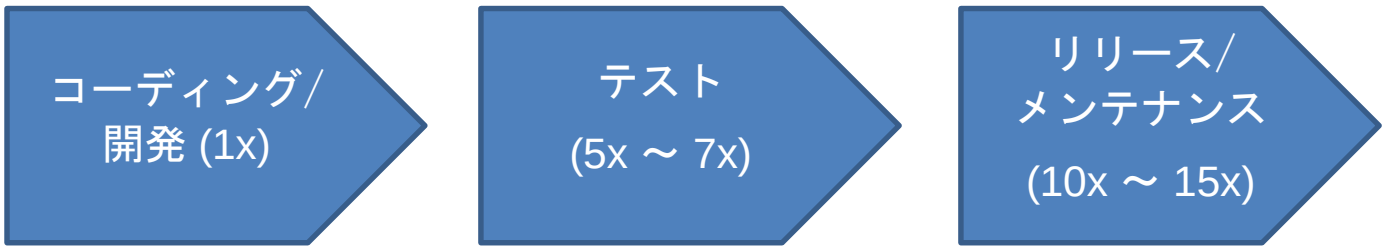
インタビュー調査対象の X 社では、Coverity と Defensics を併用してテストを実施したところ、以下のような定量的ベネフィットが得られました。

- 静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — 既存コード・ベース
- 静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス
- ファジング・テストによる不具合/セキュリティ脆弱性対策費用低減効果
- 新製品の市場投入に要する期間の短縮効果
- セキュリティ・テスト体制の改善によるビジネス中断回避効果

ソフトウェアの不具合対策に関する相対的費用

X 社で生じたベネフィットの価値を見積もるには、ソフトウェア開発ライフ・サイクル (SDLC) の各種段階でソフトウェア不具合対策を行う場合の相対的費用を見積もる必要があります。[3]説明を簡潔にするため、ここでは SDLC をコーディング/開発、テスト、リリース/メンテナンスの 3 段階に分けて考えます (図 3 を参照)。設計フェーズや複数段階のテスト・フェーズを区別する複雑なモデルが存在することも Forrester は認識していますが、本レポートではシンプルなモデルを採用します。

図 3
セキュリティ脆弱性対策に関する SDLC 各段階の相対費用



資料：Forrester Research, Inc.

この分析では以下の想定条件を設定しました。

- コーディング/開発フェーズでは初期のソフトウェア開発作業が実行されます。このフェーズで発生する費用を、不具合/セキュリティ脆弱性対策費用の相対評価のベース値とします。たとえば、このフェーズで不具合/脆弱性対策作業に 5 人時を要する場合、5 人時を 1x とします。
- テスト・フェーズで不具合/脆弱性対策を行う場合に発生する費用の大きさは、コーディング・フェーズの 5 ～ 7 倍となります。この倍率は、実施するテストの種類や SDLC 内の段階によって変動します。
- リリース/メンテナンス・フェーズは、ポストリリース、つまり、ソフトウェアがリリースされて顧客に提供された後の段階です。このフェーズでソフトウェア不具合/脆弱性対策を行う場合に発生する費用は、コーディング・フェーズの 10 ～ 15 倍になると考えられます。
- リリース/メンテナンス・フェーズで不具合/脆弱性対策を行う場合に発生する費用をテスト・フェーズと比較すると、2 ～ 3 倍の大きさになります。

インタビュー調査の結果、X 社が不具合/脆弱性対策に要する平均的な労力の大きさには、不具合の性質や、テスト・サイクル内のどのような段階で発見されたかによって大幅な違いが出ることがわかりました。この分析においては、コーディング/開発フェーズで見つかった不具合/脆弱性の対策に要する平均費用を 5 人時、リリース/メンテナンス・フェーズで見つかった不具合/脆弱性の対策に要する平均費用を 50 人時と仮定しました。人時の費用は、開発者およびテスト担当者について、また、ソフトウェアのコンパイル/ビルド作業の費用について発生します。ソフトウェアを顧客にリリースした後に不具合/脆弱性が見つかった場合は、フィールド・エンジニア、アカウント担当チーム、技術サポート・エンジニアが対応に要する人件費も対策費用に含まれることになります。

静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — 既存コード・ベース

X 社は、テストツール Coverity を静的解析に使用することの主要なベネフィットとして、不具合/脆弱性の対策費用を抑制できることを挙げています。現在、社内には 8 つのコード・ベースが存在しています。1 つのコード・ベースに含まれるコードの規模は 150 万行を超え、数十種類ものコンパイラーを併用しています。また、コード・ベースの約 30% は、各種製品に共通の再利用コードです。セキュアなソフトウェア開発ライフ・サイクルの早い段階でエラーを発見することにより、同社では不具合対策関連の費用低減が可能になりました。また、Coverity の誤検出率は約 15% と低いため、旧来のテスト手法を採用していた頃よりも短時間で確実にエラーを特定し、迅速に不具合/脆弱性の対策を施せるようになりました。Coverity に備わっている自動化機能、他の開発ツールとのインテグレーション機能、内蔵ワークフロー機能により、開発者は不具合/脆弱性を簡単かつ迅速に見つけて修正できるようになりました。

Coverity の導入によって、X 社ではテスト段階ではなくコーディング段階で不具合/脆弱性を発見できるようになり、不具合/脆弱性 1 件の対策に要する時間が平均 25 時間から 5 時間へと短縮されました。控えめに見積もってもコード 1,000 行につき 1 件の不具合/脆弱性が存在するため、Coverity は従来採用されていた手法なら見落とされていたであろう不具合/脆弱性を、コーディング/ユニット・テスト段階で 8 つのコード・ベースの中で誤検出を除き 5,406 件も発見している計算になります。コーディング/開発段階で発見された不具合/脆弱性 1 件につき、開発者が修正作業に要している時間は平均 5 時間程度です。ただし、実際の所要時間は不具合/脆弱性の種類によって大きく異なり、場合によっては時間単位でなく数日にわたることもありました。また、開発者の作業時間確保の制約から、Coverity の導入 1 年目には、発見したすべての不具合/脆弱性に対策を施すことができませんでした。そこで、このモデルでは、1 年目には優先度と重大度の高い不具合/脆弱性のみ対策作業の対象になるものと想定しました (全件のうち 30% で見積もり)。2 年目には、発見された 5,406 件の不具合/脆弱性のうち 50% (優先順位が中程度と見なされるもの) に対策が施され、3 年目に残り 20% のクリーニングが完了するものとしました。このモデルでは、不具合/脆弱性の早期対策によって節約された人的リソースのすべてが生産的に活用されるとは限らないことを踏まえ、生産性を 75% として合計に反映させています。開発者の時間あたり平均賃金を約 60 ドルとすると、Coverity 静的解析ツールによって節約されたセキュリティ脆弱性対策費用の合計は、3 年間で約 486 万ドルになります。あるソフトウェア・エンジニアリング担当マネージャーは次のように語っています。「能動的な対策は、受け身の対策よりも簡単かつ低費用で行えます。私たちは発想を変え、常に先手を打つ方針に切り替えました。それがブランド・イメージを守ることになるからです」

発見される不具合/脆弱性の種類は、対策が簡単なものから難しいものまで様々です。また、後の段階ではなくコーディング段階において作り出される不具合/脆弱性の割合も、場合によって異なります。そうした違いを考慮し、このベネフィットについては 5% 減のリスク調整を施して計算しました。静的解析による不具合/セキュリティ脆弱性対策費用の節約効果について、3 年間のベネフィットの合計を計算すると、460 万ドル強という数値が得られます (リスク調整済み)。詳細については、「リスク」の項をご覧ください。

表 1
静的解析による不具合/セキュリティ脆弱性対策費用低減効果

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
A1	1 つのコード・ベースに含まれるコード行数			1,500,000 行	1,500,000 行	1,500,000 行
A2	Coverity を使ってテストされるコード・ベースの数			8	8	8
A3	コード 1,000 行に含まれる不具合/脆弱性の平均件数	1 は平均値。実際の値は、コード開発作業内の段階、適用分野や製品ライン、コーディング規則によって異なる		1	1	1
A4	全コード・ベース中で Coverity により発見される不具合/脆弱性の件数	コード・ベース 1 つに含まれる不具合/脆弱性の平均件数 * コード・ベース数 * 再テスト率		12,000 件	12,000 件	12,000 件
A5	従来手法のコーディング/ユニット・テスト段階では見落とされるが、Coverity の採用によってコーディング/ユニット・テスト段階で発見可能な不具合/脆弱性の割合	資料： NIST		53%	53%	53%
A6	従来手法のテストであれば通常は見落とされる、コーディング段階で対策可能な不具合/脆弱性の総数 (誤検出を除く)	誤検出率、すなわち、発見されたエラーのうち実際には不具合/脆弱性でないものの割合 = 15%		5,406 件		
A7	利用可能なリソースを使って 1 年間にクリーニング可能なコード・ベースの割合	仮定：1 年目には不具合/脆弱性の 30% (最高の優先度/重大度) が対策される。2 年目には 50% (中級の優先度/重大度) が対策される。3 年目には 20% (低い優先度/重大度) が対策される		30%	50%	20%
A8	全コード・ベース中で 1 年間に対策される不具合/脆弱性の件数	注記：所要時間は、発見される不具合/脆弱性の対応難度によって異なる		1,621.8 件	2,703 件	1,081.2 件
A9	コーディング/ユニット・テスト段階の不具合/脆弱性対策に必要な人時	資料： 顧客 注記：所要時間は、発見される不具合/脆弱性の対応難度によって異なる		5 時間	5 時間	5 時間
A10	Coverity による SDLC 早期の不具合/脆弱性対策で生じた所要時間の差分 (節約分)	後のテスト・フェーズで行われる不具合/脆弱性対策については、5 倍の人時が必要になると仮定		20 時間	20 時間	20 時間

A11	エンジニアの時間あたり平均賃金			$60	$60	$60
A12	有効利用される生産性の割合	75% とする。節約される時間が生産的な作業に 100% 使われるわけではないと仮定		75%	75%	75%
At	静的解析による不具合/脆弱性対策費用低減効果 — 既存コード・ベース	A8*A10*A11*A12	$0	$1,459,620	$2,432,700	$973,080
	リスク調整率	↓10%				
Atr	**静的解析による不具合/脆弱性対策費用低減効果 — 既存コード・ベース (リスク調整済み)**		**$0**	**$1,386,639**	**$2,311,065**	**$924,426**

資料： Forrester Research, Inc.

静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス

X 社は、テストツール Coverity を静的解析に使用することの主要なベネフィットとして、新しいコード・ベース内に見つかる不具合/脆弱性の継続的な対策費用や、既存コード・ベースのメンテナンス費用を抑制できることを挙げています。組み込みハードウェア製品のメーカーである同社は、製品ラインアップの拡充や既存製品のソフトウェア・アップデートを随時続けています。その一環として、2 年目には 2 つのコード・ベースを新設し、コード規模を合計 300 万行に拡大しました。また、新規コードの追加だけでなく、8 つの既存コード・ベースについても回避できない変更や変動が発生し、静的解析の追加実行が必要になりました (20% で見積もり)。3 年にわたってコード・ベースの拡充とメンテナンスを続ける中、X 社は導入の価値やベネフィットを継続的に享受することができました。

コード再利用の効果により、2 年目に新規コード・ベースに関して必要になったテストの量は 70% のみでした。また、Coverity の誤検出率は約 15% と低いため、旧来のテスト手法を採用していた頃よりも短時間で確実にエラーを特定し、迅速に不具合/脆弱性の対策を施せるようになりました。モデルでは、新しいコード・ベースにも既存コード・ベースと同様のパターンが見られると仮定し、従来のテスト手法では見落とされるが Coverity の採用によってコーディング/ユニット・テスト段階で発見可能な不具合/脆弱性の割合を 53% (誤検出を除いて 946 件) としました。

既存コード・ベースのメンテナンスに関するベネフィットに加え、新規コードのクリーニング効果によって、このベネフィットが 1 年ごとに得られます。2 年目には、8 つの既存コード・ベースに含まれるコードの 20% (不具合/脆弱性 2,040 件) について、コードの変動により再テストが必要になります。3 年目には、コード・ベース数が 10 に増え、コード・ベース全体の拡大に伴ってメンテナンスを要するコード量が多くなります (不具合/脆弱性 2,550 件)。このモデルでは、不具合/脆弱性の早期対策によって節約された人的リソースのすべてが生産的に活用されるとは限らないことを踏まえ、生産性を 75% として合計に反映させています。コード・ベースの新規開発と既存コード・ベースのメンテナンスに関して、Coverity 静的解析ツールによって節約される不具合/脆弱性対策費用の合計は、3 年間で約 304 万ドルになります。

X 社が実行できる不具合/脆弱性対策のペースや、不具合/脆弱性を優先度別 (高/中/低優先度) に分類したときの相対的な構成比率は、場合によって異なる可能性があります。また、コード再利用のベネフィットに関しては、前年にクリーニングされたコードが効果的に再利用されるものと想定しています。そうした点の不確定要素を考慮して、このベネフィットについては 5% 減のリスク調整を施して計算しました。コード・ベースの新規開発と既存コード・ベースのメンテナンスに関して、静的解析による不具合/脆弱性対策費用節約効果の合計 (リスク調整済み) は、3 年間で約 288 万ドルになります。詳細については、「リスク」の項をご覧ください。

表 2
静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
B1	1 つのコード・ベースに含まれるコード行数				1,500,000 行	1,500,000 行
B2	コードの変動によってメンテナンスが必要となるコード・ベースの数	年間で既存コード・ベースの 20% が変更されると仮定			8	10
B3	追加される新規コード・ベースの数				2	
B4	Coverity によるテストが必要な新規コードの割合	30% のコードを再利用すると仮定			70%	
B5	コードの変動によってテストが必要となる既存コードの割合				20%	20%
B6	コード 1,000 行に含まれる不具合/脆弱性の平均件数	1 は平均値。実際の値は、コード開発作業内の段階、適用分野や製品ライン、コーディング規則によって異なる			1	1
B7	新規コード・ベースに含まれる不具合/脆弱性の件数	コード・ベース 1 つに含まれる不具合の平均件数 * コード・ベース数 * 再テスト率			2,100 件	
B8	従来手法のコーディング/ユニット・テスト段階では見落とされるが、Coverity の採用によってコーディング/ユニット・テスト段階で発見可能な不具合/脆弱性の割合	資料： NIST			53%	53%
B9	従来手法のテストであれば通常は見落とされる不具合/脆弱性の総数 (誤検出を除く)	誤検出率、すなわち、発見されたエラーのうち実際には不具合/脆弱性でないものの割合 = 15%			946 件	
B10	Coverity による SDLC 早期の不具合/脆弱性対策で生じた所要時間の差分 (節約分)	上のベネフィット表を参照			20 時間	20 時間

B11	エンジニアの時間あたり平均賃金	資料： 顧客インタビュー調査			$60	$60
B12	**節約効果 — 新規コード・ベース**	B9*B10*B11			$1,135,260	
B13	**節約効果 — 既存コード・ベースのメンテナンス**	((B1*B2)/1000)*B5*0.85*B6*B8*B10*B11			$1,297,440	$1,621,800
B14	有効利用される生産性の割合	75% とする。節約される時間が生産的な作業に 100% 使われるわけではないと仮定			75%	75%
Bt	静的解析による不具合/脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス	(B12+B13)*B14	$0	$0	$1,824,525	$1,216,350
	リスク調整率	↓10%				
Btr	**静的解析による不具合/脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス (リスク調整済み)**		**$0**	**$0**	**$1,733,299**	**$1,155,533**

資料： Forrester Research, Inc.

ファジング・テストによる不具合/セキュリティ脆弱性対策費用低減効果

X 社が Defensics テストツールを導入して得られた 3 番目に大きなベネフィットは、テスト・フェーズにおける対策費用の抑制効果です。ファジング・テストによって、セキュアなソフトウェア開発ライフ・サイクルのより早い段階で重大な不具合/脆弱性を発見できるようになり、ソフトウェアの対策費用が低減されました。開発後のテストやリリース前のテストで発見されずに残った不具合/脆弱性は、すぐに多大な出費の原因となります。ラボでの脆弱性再現作業や、その後の開発、テスト、修正リリース作業などのために、様々なチームの人員による対応費用が発生します。このベネフィットに含まれる節約効果は、テスト段階で不具合/脆弱性対策を行う場合と、顧客へのソフトウェア・リリース後に不具合/脆弱性が発見されて対策を行う場合との差分に相当します。あるシステム・テスト担当スタッフ・エンジニアは次のようにコメントしています。「Defensics を使う目的は、早期発見によって、工程やリリース作業の遅い段階で深刻な問題に突き当たる事態や、納期が遅れたり、契約が破棄されたりする事態を防ぐことです」

X 社の試算によれば、新製品に対して Defensics を使用すると、初回のテストで 40 件の不具合/脆弱性が見つかります。また、エラー修正作業に必要な期間はバグの複雑さによって異なり、2 日～ 1 か月ほどの幅があります。そこで、本レポートの計算では対策に 50 時間を要するものとしました。運用開始後に不具合/脆弱性対策を行う場合の費用は、同社の試算によるとテスト・フェーズの約 2 倍になります。このモデルでは、テスト・フェーズにおける不具合/脆弱性対策によって節約された人的リソースのすべてが生産的に活用されるとは限らないことを踏まえ、生産性を 75% として合計に反映させています。8 つの製品に含まれる不具合/脆弱性の対策に当たる人員の時間あたり平均賃金を 60 ドルとすると、ファジング・テストによって節約された不具合/脆弱性対策費用の合計は、3 年間で約 180 万ドルになります。

Defensics によって発見される不具合/脆弱性の件数や、それらの不具合/脆弱性の対策に要した時間については、X 社では場合によって大きな違いが見られます。そうした違いを考慮し、このベネフィットについては 5% 減のリスク調整を施して計算しました。ファジング・テストによる不具合/脆弱性対策費用の節約効果について、3 年間のベネフィットの合計を計算すると、170 万ドルという数値が得られます (リスク調整済み)。詳細については、「リスク」の項をご覧ください。

表 3

ファジング・テストによる不具合/セキュリティ脆弱性対策費用低減効果

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
C1	テストの初回実行で発見される不具合/脆弱性の件数	コードの再利用により、発見される不具合/脆弱性の件数は年を経るごとに減少する		40 件	30 件	30 件
C2	テスト段階の不具合/脆弱性対策に必要な人時			25 時間	25 時間	25 時間
C3	リリース/メンテナンス段階の不具合/脆弱性対策に必要な人時	コーディング/開発段階と比べて 10 倍必要になると仮定		50 時間	50 時間	50 時間
C4	エンジニアの時間あたり平均賃金	資料： 顧客インタビュー調査		$60	$60	$60
C5	1 年間にテストが実施される対象製品の数	1 年のうちに 8 か所で Defensics を使用		8 製品	8 製品	8 製品
C6	有効利用される生産性の割合	75% とする。節約される時間が生産的な作業に 100% 使われるわけではないと仮定		75%	75%	75%
Ct	ファジング・テストによる不具合/脆弱性対策費用低減効果	((C1*C3*C4*C5)-(C1*C2*C4*C5))*C6	$0	$728,970	$548,978	$548,978
	リスク調整率	↓5%				
Ctr	**ファジング・テストによる不具合/脆弱性対策費用低減効果 (リスク調整済み)**		**$0**	**$692,522**	**$521,529**	**$521,529**

資料： Forrester Research, Inc.

新製品の市場投入に要する期間の短縮効果

X 社は、ファジング・テストツールによって得られる主要なベネフィットの 1 つとして、製品の市場投入を迅速化できる点を挙げています。Defensics の採用以前、新製品 1 本をリリースするまでには約 1 年の期間が必要でした。ところが、Defensics の自動化テストツールを導入するとソフトウェアのテスト作業が迅速化され、試算では製品開発タイムラインを 4 か月短縮できるようになりました。同社では、1 年間に約 8 本の製品を Defensics でテストし、市場投入までの所要期間を 1 年から 8 か月に 25% 短縮させています。販売する製品は、ルーター、セットトップボックス、DSL 機材などの通信およびコンシューマー機器であり、販売台数は 1 機種につき 100 万台程度です。販売価格の平均は 1 台 15 ドル程度で、利益率は 20% 程度と試算されます。製品の発売を通常よりも 4 か月早めることが可能になったため、製品開発に充てる投資金額が抑えられ、販売の対価を手にする時期が早まりました。本レポートの計算では、X 社の WACC (加重平均資本費用) に基づき、資本費用を 8% としています。その結果、節約の額は年間で約 64 万ドル、3 年間の合計では 192 万ドルとなります。製品の数、製品投入までの平均的所要期間、製品の平均価格は社内でも部門によって異なります。そうした違いを考慮し、このベネフィットについては 5% 減のリスク調整を施して計算しました。新製品投入の迅速化によって得られるベネフィットの合計 (リスク調整済み) は、3 年間で約 182 万ドルになります。詳細については、「リスク」の項をご覧ください。

表 4

新製品の市場投入に要する期間の短縮効果

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
D1	1 年間にファジング・テストが実施される対象製品の数			8 製品	8 製品	8 製品
D2	標準的な製品リリース・サイクル	月数		12 ヶ月	12 ヶ月	12 ヶ月
D3	Defensics を使用する場合の製品リリース・サイクル	月数		8 ヶ月	8 ヶ月	8 ヶ月
D4	1 製品あたりの販売台数			1,000,000 台	1,000,000 台	1,000,000 台
D5	製品の平均販売価格			$15	$15	$15
D6	利益幅			20%	20%	20%
D7	資本費用の割合	調査対象企業の WACC		8%	8%	8%
Dt	新製品の市場投入に要する期間の短縮効果	D1*((D4*D5/12)*D6*(D2-D3)*D7	$0	$640,000	$640,000	$640,000
	リスク調整率	↓5%				
Dtr	**新製品の市場投入に要する期間の短縮効果 (リスク調整済み)**		**$0**	**$608,000**	**$608,000**	**$608,000**

資料： Forrester Research, Inc.

セキュリティ・テスト体制の改善によるビジネス中断回避効果

X 社によれば、ファジング・テストツールの追加導入によって、同社はコンシューマーの家庭に設置したインターネット対応デバイスに関連するビジネス中断や総合的なリスクを最小限に抑える力を向上させることができました。Defensics の採用以前には、2 大顧客のうち 1 社のネットワークで予期しないルーター再起動が発生し、日常的な運用に深刻な中断が発生する事態を経験しています。当時はまだ、市場リリース済みの製品に不具合/脆弱性が含まれているかどうかのテスト手段を持っていなかったため、とにかく最高のリソース (賃金も非常に高い) を総動員してトラブルシューティングに当たるほかありませんでした。結局、再起動の発生原因は悪意のない通信エラーだったことが判明しましたが、いずれにしても対応費用は発生し、ビジネスの評判は大きく傷つきました。膨大な数のインターネット対応デバイスが市場で稼働している現在、大きな話題になるセキュリティ事案が年に 1 ～ 2 回程度は発生し得るものと試算されます。それが悪意によって引き起こされたものであれ、単なる予想外のエラーであれ、会社に費用が発生することに変わりはありません。インシデント 1 件につき、調査、優先順位付け、対策作業には平均 15 人のリソースが必要です。最重要顧客から見たブランド・イメージ (および契約) が脅かされる事態ですから、主要な関係人員には、技術/アカウント担当/マーケティングの最上級スタッフが含まれます。そうした上級リソースの時間あたり賃金は 95 ドルと試算され、インシデント対応には 80 時間を要すると試算されます (会議、顧客との連絡、技術的介入、広報、マーケティング活動を含む)。ビジネスの中断回避による年間の費用節減効果は 20 万ドル前後、3 年間では 45 万 6,000 ドルにのぼります。セキュリティ・インシデントには、悪意によるものもあれば、まったく無害なものまであり、節約効果を予想することは困難です。そうした違いを考慮し、このベネフィットについては 5% 減のリスク調整を施して計算しました。セキュリティ・テストの改善によるビジネスの中断回避効果について、3 年間のベネフィットの合計を計算すると、約 43 万 3,000 ドルという数値が得られます (リスク調整済み)。詳細については、「リスク」の項をご覧ください。

表 5

セキュリティ・テスト体制の改善によるビジネス中断回避効果

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
E1	社会の注目を集める、大口顧客でのセキュリティ・インシデント発生件数			2 件	1 件	1 件
E2	インシデントの調査、優先順位付け、対策作業に必要なリソースの人数			15 人	15 人	15 人
E3	インシデントの調査、優先順位付け、対策作業に充てられた時間数			80 時間	80 時間	80 時間
E4	両チームの主要な関係者の時間あたり賃金	アカウント担当/技術/マーケティングの最上級スタッフと仮定		$95	$95	$95
Et	セキュリティ・テスト体制の改善によるビジネス中断回避効果	E1*E2*E3*E4	$0	$228,000	$114,000	$114,000
	リスク調整率	↓5%				
Etr	**セキュリティ・テスト体制の改善によるビジネス中断回避効果 (リスク調整済み)**		**$0**	**$216,600**	**$108,300**	**$108,300**

資料： Forrester Research, Inc.

総ベネフィット

表 6 に、上記 4 つのベネフィットの総額と、10% のリスク調整を施した現在価値 (PV) を示します。調査対象企業に 3 年間でもたらされるベネフィットの合計 (リスク調整済み) は、現在価値で 1,030 万ドルになります。

表 6
総ベネフィット (リスク調整済み)

参照番号	項目	導入時	1 年目	2 年目	3 年目	合計	現在価値
Atr	静的解析による不具合/脆弱性対策費用低減効果 — 既存コード・ベース	$0	$1,386,639	$2,311,065	$924,426	$4,622,130	$3,865,087
Btr	静的解析による不具合/脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス	$0	$0	$1,733,299	$1,155,533	$2,888,831	$2,300,647
Ctr	ファジング・テストによる不具合/脆弱性対策費用低減効果	$0	$692,522	$521,529	$521,529	$1,735,579	$1,452,413
Dtr	新製品の市場投入に要する期間の短縮効果	$0	$608,000	$608,000	$608,000	$1,824,000	$1,512,006
Etr	セキュリティ・テスト体制の改善によるビジネス中断回避効果	$0	$216,600	$108,300	$108,300	$433,200	$367,781
	総ベネフィット (リスク調整済み)	**$0**	**$2,903,761**	**$5,282,192**	**$3,317,787**	**$11,503,740**	**$9,497,933**

資料： Forrester Research, Inc.

費用

調査対象の X 社には、Defensics および Coverity ソリューションに関する様々な費用が発生しました。

- Coverity のソフトウェア・ライセンス料。
- Defensics のソフトウェア・ライセンス料。
- オンサイト・コンサルティング・サービス料。これには、Coverity および Defensics の実装とインテグレーションに関する支援費用が含まれます。

以下の費用は、ソリューションに関連する初期計画、実装、継続的なメンテナンスのために調査対象企業が実際に支出した対内費用および対外費用を合わせたものです。

Coverity のソフトウェア・ライセンス料

Coverity テストツールのソフトウェア・ライセンス料は継続的に発生し、開発チームが作成したコード行数に基づいて、1 年ごとに Synopsys に支払われます。X 社は 1,200 万行のコードを抱え、使用プログラミング言語は 8 種類にわたっています。2 年目には 300 万行のコードを新たに追加しました。コード 1 行につき 0.09 ドルの料金がかかることから、年間 100 万～ 130 万ドル程度の支払いが継続的に発生します。同社がソフトウェア・ライセンス料として 3 年間に支払った金額は、378 万ドルです。

表 7
Coverity のソフトウェア・ライセンス料

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
F1	コード行数			12,000,000 行	15,000,000 行	15,000,000 行
F2	コード 1 行あたり料金			$0.09	$0.09	$0.09
Ft	Coverity のソフトウェア・ライセンス料	F1*F2		$1,080,000	$1,350,000	$1,350,000
	リスク調整率	0%				
Ftr	**Coverity のソフトウェア・ライセンス料 (リスク調整済み)**			**$1,080,000**	**$1,350,000**	**$1,350,000**

資料： Forrester Research, Inc.

Defensics のソフトウェア・ライセンス料

Defensics テストツールのソフトウェア・ライセンス料は継続的に発生し、Defensics の使用ライセンス本数に基づいて、1 年ごとに Synopsys に支払われます。X 社は Defensics のライセンスを 32 本保有し、複数のチームで共用しています。ライセンス 1 本あたりの料金は約 1 万 1,000 ドルであることから、年間 35 万 6,832 ドルの支払いが継続的に発生します。同社がソフトウェア・ライセンス料として 3 年間に支払った金額は、107 万 496 ドルです。

表 8
Defensics のソフトウェア・ライセンス料

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
G1	Defensics の年間ライセンス・サブスクリプション料金			$11,151	$11,151	$11,151
G2	ライセンス本数			32 本	32 本	32 本
Gt	Defensics のソフトウェア・ライセンス料	G1*G2		$356,832	$356,832	$356,832
	リスク調整率	0%				
Gtr	**Defensics のソフトウェア・ライセンス料 (リスク調整済み)**		**$0**	**$356,832**	**$356,832**	**$356,832**

資料： Forrester Research, Inc.

オンサイト・コンサルティング・サービス料

X 社では、Defensics の導入 1 年目に、非常に急ピッチで進む導入プロセスを支援する目的でオンサイト・コンサルティング・サービス料を支出しています。ルーターのセキュリティに対する潜在的脅威に対処する目的で使用を開始したことから、迅速な立ち上げが必要であり、そのために Synopsys のサポートを利用しました。X 社によると、有料コンサルティングに加え、Synopsys の継続的サポートとパートナーシップも追加料金なしで提供されました。オンサイト・コンサルティング・サービスのために最初に支払った金額は 2 万 5,000 ドルです。その後、Defensics を使用する部署が増えたことを受け、2 年目には 1 万ドルを支払って追加のコンサルティング・サービスを利用しています。3 年間でオンサイト・コンサルティング・サービスのために発生した費用の合計は 3 万 5,000 ドルでした。

表 9
オンサイト・コンサルティング・サービス料

参照番号	評価項目	計算式	導入時	1 年目	2 年目	3 年目
H1	オンサイト・コンサルティング・サービス		$25,000		$10,000	
Ht	オンサイト・コンサルティング・サービス		$25,000	$0	$10,000	$0
	リスク調整率	0%				
Htr	**オンサイト・コンサルティング・サービス (リスク調整済み)**		**$25,000**	**$0**	**$10,000**	**$0**

資料： Forrester Research, Inc.

総費用

表 10 に、総費用と現在価値 (PV、割引率 10%) を示しています。X 社が 3 年間で支出する費用の合計は、現在価値で 403 万ドルになります。

表 10
総費用 (リスク調整済み)

参照番号	費用項目	導入時	1 年目	2 年目	3 年目	合計	現在価値
Ftr	Coverity のソフトウェア・ライセンス料	$0	$1,080,000	$1,350,000	$1,350,000	$3,780,000	$3,111,796
Gtr	Defensics のソフトウェア・ライセンス料	$0	$356,832	$356,832	$356,832	$1,070,496	$887,388
Htr	オンサイト・コンサルティング・サービス	$25,000	$0	$10,000	$0	$35,000	$33,264
	総費用 (リスク調整済み)	**$25,000**	**$1,436,832**	**$1,716,832**	**$1,706,832**	**$4,885,496**	**$4,032,448**

資料： Forrester Research, Inc.

柔軟性

TEI 法における柔軟性とは、今後、特定の機能やツールが追加されると、これがさらなる業務上のベネフィットにつながる可能性があるため、先行投資を検討する余地があるということです。組織は将来の取り組みに先行投資できる、または先行投資の「妥当性」が確保できるものの、必ずしも先行投資する必要はありませんので、柔軟に対応できます。顧客が Coverity の導入を決めてから、後になって他の用途やビジネス・チャンスに気づくといった状況は何とおりか考えられます。柔軟性は特定のプロジェクトの一部として定量的に評価することも可能です。

Coverity と Defensics には、次のようにしてリスク・エクスポージャーを最小限に抑える効果があります。

- **コードの不具合密度を下げ、再利用コードの高品質化を促す。**X 社では、各種製品ライン間でかなりの量のコードが再利用されています。もし、不具合密度 (コード行数あたりのエラー含有率) の高いコードを使ってライブラリが構築され、そのエラーが修正されない場合、不具合を含んだコードがライブラリから何度もコピー & ペーストされて様々な目的に使われ、不具合密度が再び高まることになります。Coverity を採用した結果、同社は不具合密度を下げ、エラーを含んだコードが共有ライブラリ内に混入することを防止できるようになりました。これには、コード・ベース全体に含まれる不具合の件数を総合的に減らす効果があります。この点は本レポートにおける財務モデルの計算に盛り込まれていませんが、不具合密度を下げた上でコード再利用を進めることは、全体的な開発費用の低減につながります。ソフトウェア開発ライフ・サイクルの早い段階でバグを発見すると、テスト段階や運用段階でバグ対策を行う場合に比べて影響をはるかに小さく抑えることができます。

- **コンシューマーのリスク・エクスポージャーを下げ、潜在的被害の発生を防ぐ。**X 社に対して実際にセキュリティ攻撃が行われた形跡はないものの、セキュリティ脆弱性によって防御に隙が生まれる恐れは常にあります。同社は、このリスクを強く意識して対応体制を整えています。当局に提出した 10K レポートでも、同社はコンシューマーの情報やデータが悪用されるリスクに常にさらされていることを強調し、次のように述べています。「ハードウェアまたはソフトウェアの不具合は、不正なユーザーに、コンシューマーのネットワークやコンシューマーのホーム・ネットワークに対するアクセス手段を与える原因にもなりかねない。そうした不具合によって、顧客の当社に対する評価が悪化する恐れがあるだけでなく、当社が顧客と交わした契約のもとで損害賠償請求を受ける恐れや、規制当局から罰金を課せられる恐れもある」

- **自社や供給先大規模企業の評判が傷つくリスクを下げる。**X 社の年間売上高の内訳を見ると、最大顧客 2 社からの売上が 35% を占めます。同社にとって、あらゆる種類のセキュリティ脆弱性は (悪意にさらされる性質の有無にかかわらず) 収益や販売契約をおびやかす脅威になります。同社の 10K レポートには、「当社のビジネスは少数の顧客に集中している。いずれかの顧客との契約を失う事態や、いずれかの顧客に対する販売が大幅に落ち込む事態が発生した場合、当社のビジネスに重大な悪影響が及ぶ恐れがある」との記述があります。ただし、Defensics と Coverity を併用するようになった現在は、案件を失いかねないほどの評判リスクから自社を守れる可能性が高まったといえます。

› **テスト・スクリプトの開発と実行に関する費用を抑制する。**X 社の場合は独特の状況下でツール・セットが導入されているため、本レポートではこの点をモデルに盛り込んでいませんが、テスト・スクリプトの開発と実行に関する費用抑制効果が社内の他部署にベネフィットをもたらす可能性もあります。もし、テスト・スクリプトを独自に作成して手動実行している部署がある場合、その体制を静的解析ツールとファジング・テストツールで置き換えることができれば、さらなる費用低減効果が見込まれます。X 社は数十種類の製品を擁し、開発チームも多数あるため、両ツール製品が社内に普及して使用が広がるにつれて、このベネフィットを享受するチームが現れる可能性は高いと考えられます。

リスク

Forrester は、この費用ベネフィット解析について 2 種類のリスクを想定しています。それは、「導入に関するリスク」と「導入効果に関するリスク」です。「導入に関するリスク」とは、Coverity および Defensics への投資に伴うリスクであり、当初予想された投資の枠組みから逸脱し、結果的に予想以上に費用がかかるリスクのことです。「影響度リスク」とは、業務や技術に関する社内ニーズが Coverity および Defensics への投資によって満たされず、全体的な利用価値が低下するリスクのことです。このような不明確さが高いほど、費用とベネフィットの見積りに対する影響範囲が大きくなります。

表 11
ベネフィットと費用のリスク調整

ベネフィット	リスク調整率
静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — 既存コード・ベース	↓ 5%
静的解析による不具合/セキュリティ脆弱性対策費用低減効果 — コード・ベースの新規開発およびメンテナンス	↓ 5%
ファジング・テストによるセキュリティ脆弱性対策費用低減効果	↓ 5%
新製品の市場投入に要する期間の短縮効果	↓ 5%
セキュリティ・テスト体制の改善によるビジネス中断回避効果	↓ 5%

資料： Forrester Research, Inc.

費用やベネフィットの見積もりを直接調整することで「導入に関するリスク」と「導入効果に関するリスク」を定量的に評価すると、より正確で実用的な見積もりになり、この結果、ROI をより正確に把握できます。一般に、リスクを当初より高めに見積もると費用に影響し、低めに見積もるとベネフィットに影響します。リスク調整済みの値はリスクを考慮した、現実的な予測値と考えてください。

今回の分析でベネフィットに影響する「影響度リスク」は次のとおりです。

› 発見される不具合/脆弱性の種類は、対策が簡単なものから難しいものまで様々です。また、後の段階ではなくコーディング段階において作り出される不具合/脆弱性の割合も、場合によって異なります。諸事情を考慮し、既存コード・ベースに関する不具合/セキュリティ脆弱性対策費用低減効果については、ベネフィットに 5% 減のリスク調整を施しています。

› X 社が実行できる不具合/脆弱性対策のペースや、不具合/脆弱性を優先度別 (高/中/低優先度) に分類したときの相対的な構成比率は、場合によって異なる可能性があります。また、コード再利用のベネフィットに関しては、前年にクリーニングされたコードが効果的に再利用されるものと想定しています。諸事情を考慮し、新規および既存コード・ベースに関する不具合/セキュリティ脆弱性対策費用低減のベネフィットについては 5% 減のリスク調整を施しています。

- Defensics によって発見される不具合/脆弱性の件数や、それらの不具合/脆弱性の対策に要した時間については、X 社では場合によって大きな違いが見られます。諸事情を考慮し、ファジング・テストによる不具合/脆弱性対策費用低減効果については、ベネフィットに 5% 減のリスク調整を施しています。
- 製品の数、製品投入までの平均的所要期間、製品の平均価格は社内でも部門によって異なります。諸事情を考慮して、新製品市場投入までの期間短縮によるベネフィットについては 5% 減のリスク調整を施しています。
- セキュリティ・インシデントには、悪意によるものもあれば、まったく無害なものまであり、節約効果を予想することは困難です。諸事情を考慮し、セキュリティ・テスト改善によるビジネス中断回避のベネフィットについては 5% 減のリスク調整を施しています。

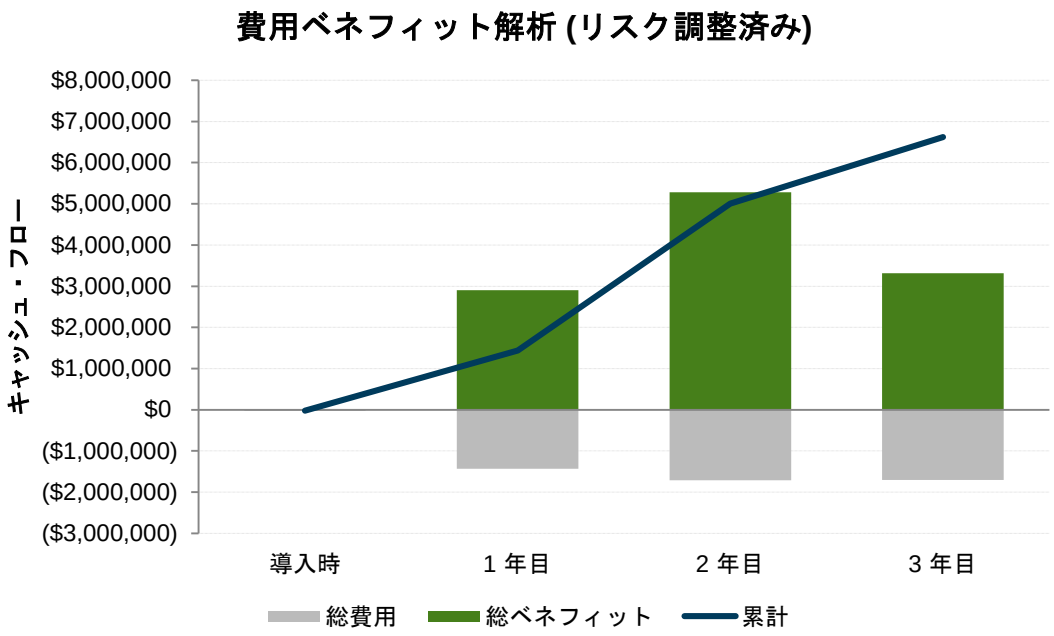
リスクや不確実性を考慮して調査対象企業に関する費用およびベネフィットの見積もりに適用した調整値の一覧を表 11 に示します。実際にリスク調整を加える場合は、費用とベネフィットの見積もりがどの程度信頼できるかを判断したうえで、リスク範囲を設定してください。

財務データのまとめ

「ベネフィット」と「費用」の各項で計算した数値に基づき、調査対象の X 社が行った Defensics と Coverity への投資に関する ROI および NPV を算出しました。

表 12 に、ROI と NPV のリスク調整済み値を示します。これらの値は、「費用」と「ベネフィット」の各項に示すリスク未調整の値に、「リスク」の項の表 11 に示すリスク調整の割合を適用したものです。

図 4
財務分析 (リスク調整済み)



資料： Forrester Research, Inc.

表 12
キャッシュフロー (リスク調整済み)

	導入時	1 年目	2 年目	3 年目	合計	現在価値
費用	($25,000)	($1,436,832)	($1,716,832)	($1,706,832)	($4,885,496)	($4,032,448)
ベネフィット	$0	$2,903,761	$5,282,192	$3,317,787	$11,503,740	$9,497,933
純ベネフィット	**($25,000)**	**$1,466,929**	**$3,565,360**	**$1,610,955**	**$6,618,244**	**$5,465,485**
ROI（投資利益率）						**136%**

資料： Forrester Research, Inc.

製品の概要：Coverity

以下の情報は Synopsys から提供されたものです。Forrester は、主張の妥当性をいっさい検証しておらず、また、Synopsys や同社製品・サービスを支持するものでもありません。

Coverity® は、正確で包括的な静的解析および静的アプリケーション・セキュリティ・テスト (SAST) プラットフォームです。コードに含まれる重大な不具合やセキュリティ上の弱点によって脆弱性、クラッシュ、メンテナンス上の不都合が発生する前に、コードが記述される段階でそうした不具合を発見します。

製品の概要

Coverity は、重大な品質上の不具合や潜在的なセキュリティ脆弱性を開発段階で特定することにより、リスクの低減と全体的なプロジェクト費用の抑制に役立ちます。Coverity は、特許認定された技法、10 年にわたる研究開発の成果、100 億行を超える非公開および公開ソースコードの解析結果を基にして、対応行動可能かつ正確な対策ガイダンスを提供します。

主な特長

解析の深度と正確さ

- Coverity には、あらゆるビルド・システムとのインテグレーションを実現し、ソースコードの挙動を深く理解できる精細な表現を生成する機能があります。
- Coverity では、パスを全面的に網羅し、すべてのコード行とすべての潜在的な実行パスを確実にテストできます。複数の特許認定済み技術により、解析の深度と正確さが確保されています。
- ソースコードとその基礎のフレームワークに対する深い理解に基づいて、Coverity プラットフォームは、非常に正確な解析結果を導き出します。開発者は大量の誤検出を処理するために無用の時間を費やす必要がなく、開発ライフ・サイクル内にセキュリティを効果的に組み込むことができます。

解析の速度とスケール

Coverity は、既存ワークフローとの親和性を考えてゼロから設計されており、以下の機能を備えています。

- 並列解析：最大 16 コアを同時に使って動作し、逐次解析と比べて最大 10 倍のパフォーマンスを発揮します。
- インクリメンタル解析：コード全体を毎回解析するのではなく、変更箇所や変更の影響を受ける箇所だけを再解析することにより、解析処理を高速化します。
- スケーラビリティ：開発者の人数が非常に多く地理的に分散している環境に対応し、コード量が 1 億行を超える巨大なプロジェクトも容易に解析できます。

修正案件管理と対策作業の効率性

- このプラットフォームには、共同作業に適した修正案件管理インターフェイス Coverity Connect が用意されています。セキュリティ分野の高度なノウハウを持たない開発者でも、対応行動を示す情報や的確な対策ガイダンスにアクセスし、適切な不具合修正方法や最適なコード修正箇所を知ることができます。
- Coverity Connect のソースコード・ナビゲーション機能を使用すると、不具合にたどり着くためのパスを正確に識別し、共有コード内の不具合出現箇所すべてを自動的に特定できます。

› 適切な開発者への不具合対策作業の割り当てを自動的に行うことができます。ユーザーは、すべての未解決セキュリティ修正案件のリストや、セキュリティ専門コミュニティ OWASP のトップ 10 案件リスト、CWE および PCI 関連の問題を手早く確認できます。

ソフトウェア開発ライフ・サイクルのインテグレーション

› ソースコード管理、ビルドおよび連続的インテグレーション、バグ・トラッキング、統合開発環境 (IDE)、アプリケーション・ライフ・サイクル管理 (ALM) ソリューションなどのツールやシステムは、開発プロセスをサポートする非常に重要なものです。Coverity プラットフォームは、こうしたツールやシステムとのインテグレーションを短時間で実現できます。

› Coverity は開かれたプラットフォームです。開発者がサード・パーティーの解析結果をワークフローに読み込んで、あらゆる種類の不具合を同じように参照・管理し、一元化されたビューでソフトウェア不具合やリスクを扱うことができます。

利用拡大の促進とリスクの緩和

Coverity Policy Manager を使用すると、コード・セキュリティ、品質、テストに関する一貫した基準を定義し、開発チームの違いを越えて横断的に適用できます。どのチーム、プロジェクト、コンポーネントが標準に準拠しているかを把握できるほか、計測可能な段階的関門を設定して、不具合とテストに関する所定の条件を遵守させることができます。Coverity Policy Manager のカスタマイズ可能なビューでは、具体的な目的に応じた開発のメトリクスやしきい値の選択が可能です。

製品の概要：Defensics

以下の情報は Synopsys から提供されたものです。Forrester は、主張の妥当性をいっさい検証しておらず、また、Synopsys や同社製品・サービスを支持するものでもありません。

Defensics® は強力なテスト用プラットフォームです。開発者およびアセット・オーナーが、ソフトウェアとデバイスに含まれる未知の脆弱性を事前に発見して対策を講じることができます。

製品の概要

Defensics のコア・テクノロジーは、テスト対象システムに対して計画的に異常なデータを送り込むことにより未知の脆弱性を発見する、ファジング・テストという自動化された手法です。ファジング・テストでは、出回っている他のどのようなソリューションよりも効果的に脆弱性を発見できます。2014 年 4 月には、Synopsys とは無関係に運用されていた Defensics が、50 万箇所以上の Web サイトに被害を与えた OpenSSL の脆弱性「Heartbleed」を発見しました。

主な特長

Defensics の主な特長は次のとおりです。

- **導入するだけで機能を発揮。**構築済みのテスト・スイートが付属している完全自動化テスト・プラットフォームであるため、手作業でテストを作り込むための作業担当者や手間が必要ありません。
- **広範なプロトコルに対応。**290 種類あまりのネットワーク・プロトコル、ファイル形式、その他各種インターフェイスに対応した高度なテスト・スイートが用意されています。テスト開発専任のチームによって、テスト・スイートの追加、改良、サポートが継続的に行われています。
- **高度なテストケース生成機能。**「テンプレート」「世代的」「進化的」といった異なるタイプのテスト・エンジンを活用し、様々な手法を盛り込んだ効果的なテストケースを生成します。
- **障害検知。**有効ケースまたは機能応答、リソース監視、動的バイナリ解析、ソースコード・インストルメンテーションなど、障害や異常動作を検知するための高度な技法をサポートしています。
- **対策作業のパスを明示。**ドキュメンテーションおよびレポート機能を完備しているため、重大な障害の根本原因について、対策プロセス関係者間で共有できる反復可能かつ理解しやすい形式で情報を伝えることができます。
- **優れた使い勝手。**直感的にわかりやすいユーザー・インターフェイスにより、セキュリティの専門家でなくても高度なテストと対策作業を簡単に実行できます。

付録 A: Total Economic Impact™ の概要

Total Economic Impact (TEI) 法は、Forrester Research が開発した評価方法です。企業の意思決定プロセスの構築に役立ち、ベンダーは、評価結果に基づいて自社の製品やサービスの価値を顧客に紹介することができます。また、TEI 手法を使用することで、経営幹部や重要な利害関係者に対して IT プロジェクトの実質的な価値を実証、正当化、および実現することができます。TEI 法は、IT ベンダーによる顧客の獲得と維持、および顧客サービスの向上に役立ちます。

TEI 法では、「ベネフィット」「費用」「柔軟性」「リスク」の 4 項目により投資対効果を評価します。

ベネフィット

ベネフィットとは、提案された製品やプロジェクトによってユーザー企業 (IT 部門や業務部門) にもたらされる価値のことです。通常、製品やプロジェクトの価値を評価するときには、IT 関連の経費や経費削減のみが強調されがちで、IT 投資が組織全体に与える影響の分析は後回しにされてしまいます。TEI 法および調査結果に基づいて作成される財務モデルでは、ベネフィットの計測と費用の計測に同じ重みを与えることで、IT 投資が組織全体に与える影響を徹底的に評価することができます。また、ベネフィット分析では、ユーザー組織と率直な話し合いを行い、実際にどのような価値創出につながるのかを理解することが重要になります。さらに、Forrester では、プロジェクト完了後のベネフィットの見積もりに関する説明責任と見積もりの根拠に関する説明責任を明確に区別しています。これにより、ベネフィットの見積もりを直接最終的な収益へとつなげることができるのです。

費用

費用とは、提案されたプロジェクトから価値 (ベネフィット) を得るために必要な投資のことです。IT 部門や業務部門は、人件費、材料費、外注費等の経費を「費用」として実質的に負担することになります。提案された価値を実現するために必要な投資と経費すべてが「費用」として考慮されます。さらに、TEI 法の費用区分では、ソリューションに関連して既存の環境で継続的に発生する追加費用もすべて考慮されます。費用はすべて、プロジェクトから得られるベネフィットに結びつくはずです。

柔軟性

TEI 法では、直接的なベネフィットは投資価値の一部にすぎません。一般に、プロジェクトを正当化するためには直接的なベネフィットが非常に重要になりますが、Forrester は、組織は投資の戦略的価値も測定する必要があると考えています。柔軟性とは、既に行われた初期投資に加えて、今後予定している投資を前倒しで行ったときに得られる価値のことです。たとえば、オフィス製品一式を企業全体でアップグレードするための投資を行った場合、標準化の推進 (効率性の向上) とライセンス料の削減につながる可能性があります。一方、組み込みのコラボレーション機能を使用すると、従業員の生産性向上につながる可能性がありますが、この機能を実際に使用できるようになるのは、将来のある時点で社員研修に追加投資をした後のことです。しかし、このようなベネフィットが得られるという可能性自体に、見積もり可能な現在価値があるのです。TEI 法の柔軟性とは、このような価値を把握することです。

リスク

リスクとは、投資におけるベネフィットと費用の見積もりの不確かさを測定したものです。不確かさを測定するときには、1) ベネフィットと費用の見積もりが当初の期待値に一致する可能性、および 2) 見積もりを長期間にわたって確認し、追跡できる可能性を検討します。TEI 法では入力値に「三角分布」の確率密度関数を適用します。各費用およびベネフィットのリスクを見積もるときには、少なくとも 3 つの値を計算します。

付録 B: Forrester と顧客の時代

貴社のテクノロジーを業務に活用している顧客は、今や、貴社の製品やサービス、価格、評判を貴社以上に把握しています。競合企業は、競争のため、貴社の活動を模倣したり、巧みに妨害することがあります。顧客を獲得し、顧客サービスを向上させて顧客維持につなげる唯一の方法は、Customer Obsession (顧客満足への執着) です。

顧客満足に執着する企業は、競争優位の長期的な維持以上に、顧客に関する情報の収集と顧客エンゲージメントの向上を最優先し、これを視野に入れて戦略を立て、エネルギーと予算をつぎ込んでいます。

CMO と CIO が協力して全社的な変革を成し遂げる

Forrester では、この顧客の時代における戦略を策定するために、新しい競争優位の確立に有効な以下の 4 原則から成るブループリントを用意しています。

カスタマ・エクスペリエンスを改善し、持続可能な競争力を獲得する。

企業の成長を促す新しい IT 戦略により、ビジネスを加速させる。

モバイル・マインドシフトを推進し、顧客が必要なものを必要なときに提供する。

革新的な解析法により、(膨大な量の) データを実用的なビジネス・インサイト (洞察) に変える。

付録 C: 用語解説

割引率: キャッシュフロー分析で、貨幣の時間的価値を考慮するために使用する利率。各企業は、通常、自社の事業環境や投資環境に基づいて独自の割引率を設定します。Forrester はこの分析において、年間割引率を 10% に設定しています。組織は通常、現時点の事業環境に基づき、割引率を 8 ～ 16% の範囲で設定しています。これを適用する場合は、各組織の経理部と相談の上、組織内で使用する適正な割引率を設定することをお勧めします。

正味現在価値 (NPV): 利率 (割引率) が設定されている場合の (割引後の) 将来の正味キャッシュフローの現時点での価値。あるプロジェクトの正味現在価値が正であれば、通常は、投資すべきであることを意味します。ただし、他のプロジェクトの正味現在価値の方が高い場合は別です。

現在価値 (PV): 利率 (割引率) が設定されている場合の (割引後の) 見積もり費用およびベネフィットの現時点での価値。費用およびベネフィットの現在価値からキャッシュフローの正味現在価値の合計を計算します。

回収期間: 投資金額が回収され、損益分岐点に到達するまでの期間。損益分岐点とは、純ベネフィット (ベネフィットから費用を差し引いた値) が初期投資金額に等しくなる時点のことです。

投資利益率 (ROI): プロジェクトに投資した金額に対する、期待される利益の割合。ROI は、正味利益 (利用価値から費用を引いた値) を費用で割ることによって求められます。

キャッシュフロー表に関する注意事項

本調査で使用したキャッシュフロー表に関する注意事項を以下に示します (以下のサンプル表を参照)。初期投資の欄には、「時間 0 (導入時)」または 1 年目の開始時に発生した費用が記載されます。この費用には割引率が適用されません。1 年目から 3 年目まで、すべてのキャッシュ・フローには、各年の末日付で割引率 10% が適用されます。見積もりの総費用及び総ベネフィットの各値について現在価値 (PV) を計算しています。正味現在価値 (NPV) は、初期投資と各年の割引後キャッシュフローの現在価値の合計であり、要約表にのみ記載されます。

「総ベネフィット」、「総費用」、「キャッシュフロー」の各表の合計金額及び現在価値については、四捨五入のため合計値が合わないことがあります。

表 (サンプル)

サンプル表

参照番号	評価項目	計算式	1 年目	2 年目	3 年目

資料： Forrester Research, Inc.

付録 D: 巻末の注

[1] 本レポートでは、費用およびベネフィットの見積もりの不確かさを考慮し、費用ベネフィット解析の各評価項目についてリスク調整済みの値を示しています。詳細については、「リスク」の項をご覧ください。

[2] 資料： “Know Your Code: How Static Analysis Tools Make Applications More Secure,” Forrester Research, Inc., November 20, 2009.

[3] 不具合対策の相対費用に関する各種の仮定は、2012 年 7 月発行の調査レポート『Codenomicon の Defensics セキュリティ・テスト・スイートに関する Total Economic Impact』に基づいています。現行の調査で使われている各種の比率には、本シリーズの前回調査で使われたものと一貫性があります。