アジャイル設計の自動化によるソフトウェア完全性の向上

シノプシス　シニア・テクニカル・マーケティング・マネージャ　Jon Jarboe

ソフトウェア・テストおよびスタティック解析ツールをアジャイル手法と組み合わせ、コードの品質とセキュリティを向上して製品発売スケジュールを「シフトレフト（前倒し）」する方法について、シノプシスのシニア・テクニカル・マーケティング・マネージャ、Jon Jarboeがご説明します。

ソフトウェアの急速な複雑化に伴い、品質とセキュリティの問題は深刻さを増しています。毎日、600万人以上のソフトウェア開発者が6,000万行を超えるコードを作成しており、その多くは安全 / セキュリティが重視されるアプリケーションをターゲットとしています。

このように複雑化が進んだ結果、ソフトウェア・アプリケーションは非常に脆弱になっており、場合によってはセキュリティ被害、安全上の問題、製品のリコール、信用の失墜、収益の損失・遅延といった実害をもたらすこともあります。

ソフトウェア開発手法はここ5年ほどで大きく進歩しましたが、多くのソフトウェア開発チームがまだ十分に堅牢な開発プロセスを導入できずにいます。

ハードウェア設計チームはかなり以前から高度なチップ設計手法を採用し、ハードウェア設計プロセスの予測性向上に努めてきました。これに対し、ソフトウェア開発コミュニティがより成熟したソフトウェア開発プロセスの必要性を認識するようになったのは、コードの品質およびセキュリティの問題が深刻化するようになった比較的最近のことです。

ソフトウェアの品質

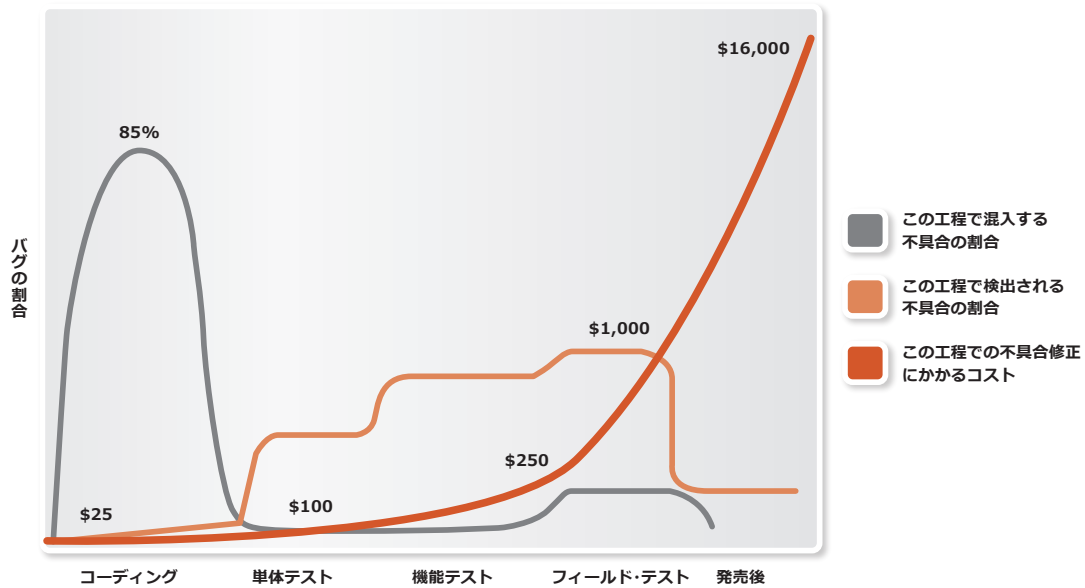
工期の圧縮が進む中、現在のソフトウェア開発プロセスは、いかに短期間で複雑かつ高機能なコードを大量に生産できるかが重視されています。従来のソフトウェア開発アプローチでは、品質保証とテストはコーディング完了後の作業と位置づけられており、テストされていない大量のソフトウェア・コードがQAおよびセキュリティ・チームにハンドオフされていました。

このように、開発サイクルの終盤になって未知の問題を含んだ大量のコードを検証するのは、開発チームにとって大きな負担となります。ソフトウェアの動作意図を記述した仕様に基づいてテストを開始することも可能ですが、コードのテストおよび見つかった問題の修正にどれだけ時間がかかるかを予測するのは非常に困難です。このように予測性が低いとテストに十分な時間を割くことができず、製品発売スケジュールを優先して既知の不具合を抱えたまま出荷に踏み切るケースも少なくありません（図1）。

こうした開発プロセスには、テストの完全性、コスト、スケジュール予測性の点で大きな問題があります。

ソフトウェア不具合の約85%はコーディング工程で混入します。不具合はこの段階で修正するのが最も簡単でコストがかからず、後の工程になるほど修正は難しくなり、そのため開発コストも上昇します。現在はほとんどの不具合が開発プロセスの終盤に検出されていますが、これをもっと早い段階で検出・修正できるようにすれば、大幅なコスト削減が期待できます。

検証工程で発見されたすべての不具合を修正するには、コードの品質にもよりますが数ヶ月 ～ 数年かかります。これでは製品発売日の予定が立たないばかりでなく、動作不良を起こしやすく、製品の競争力も低下します。これは、検証チームが開発プロセスの終盤になって作業を開始し、そこで非常に多くの不具合が見つかってしまうという点にすべての原因があります。問題の早期発見が可能になれば、開発チームはより多くの時間を不具合の修正に当てることができます。この結果、出荷製品のバグが減少し、検証サイクルが円滑化し、スケジュールの予測性も向上します。



出典：Applied Software Measurement, Capers Jones, 1996

図1. コードの不具合を開発サイクルの早期段階で検出、修正した場合のコスト メリット

アジャイル・ソフトウェア開発

このようにQAとテストを開発サイクル終盤まで先延ばしにすると多くの問題が生じるため、アジャイル手法と呼ばれるソフトウェア開発手法を導入するチームが増えています。アジャイル手法とは、小規模なコード・モジュールを随時テストしながらライフサイクル全体で開発を進めていくアプローチで、ここ5年ほどでソフトウェア開発チームに広く浸透するようになっています。

アジャイル・アプローチの1つの利点として、ソフトウェア開発チームが開発プロセスの早期段階でバグを検出・修正できる点が挙げられます。しかし個々のコード・モジュールをすべてテストしても、最終的にはこれらモジュールを結合して全体のテストを実行する必要があります。このため、結合したモジュールの検証についてライフサイクル全体で対処しなければ、最後の検証工程に長い時間がかかる可能性があります。

あらゆるソフトウェアに要求されるセキュリティ

現在の開発 / 検証プロセスが直面しているもう1つの大きな課題として、ソフトウェアに堅牢なセキュリティ対策が必要とされている点があります。以前であれば、セキュアなコードが必要なのは機密データを扱うプロジェクトに限られていましたが、サーモスタットやホーム・オートメーション・システム、自動車などあらゆるものがインターネットに接続されるようになった現在では、ほとんどのアプリケーションでセキュリティが求められます。

アジャイルを超えて

アジャイル手法を導入すると、ソフトウェア開発チームは複雑なソフトウェア・プロジェクトにも効果的に対処し、より高品質なコードをスケジュールどおりに開発できるようになります。アジャイル・アプローチでは、これまで開発ライフサイクルの終盤でなければ見つからなかった不具合を開発工程で発見できるため、短時間での修正が可能です。しかしアジャイル・アプローチだけでは、開発スケジュールをそれほど大きくは「シフトレフト」できません。

ハードウェア設計チームが既に採用しているように、ソフトウェア開発者もプロジェクトの開発工程の早期段階で問題を発見できるよう支援する自動化ツールを利用すれば、大きな効果が得られます。ツールによる自動化を利用すると、開発者はソフトウェア・テストの作成および実行という煩雑な作業から解放され、コードの作成および検出された不具合への対処に集中できます。

ソフトウェア開発プロセスを上手に管理し、品質およびセキュリティに関するテストを自動化することによってソフトウェア開発チームの生産性が向上し、最後の検証工程も短時間で円滑に行え、スケジュールの予測性が改善します。

検証プロセスを支援する自動化ツールを利用すると、人手によるテストの作成と実行に費やす時間が削減され、チームは、より堅牢なソフトウェア・セキュリティの導入や十分なテスト・カバレッジの確保など、より戦略的な問題に取り組むことができます。

スタティック解析とダイナミック・テスト

スタティック解析は、ソフトウェア品質を改善するための基本的な技術です。スタティック解析ツールは、コードを実行するのではなくソフトウェアのソースコードに高度なアルゴリズムを適用して問題を検出します。このため、開発者はあるルーチンを作成したらすぐにコードの品質改善に取り組むことができます。コードは完全である必要はなく、機能の一部が記述されていなくても解析を実行して品質やセキュリティの問題を特定できます。

スタティック解析技術を使用すると、コード可読性や変数名の一貫性を改善できるだけでなく、NULLポインタやバッファ・オーバーフローなどコードの品質とセキュリティに影響する多くの問題を特定して修正できます。

ソフトウェアのスタティック検証は、ハードウェア設計向けのスタティック解析ツールからいくつかの手法を取り入れています。このため、ハードウェア設計用のスタティック・ツール同様、パターンに基づいて本来違反でないものを違反と報告してしまう「誤検出」の問題がつきまといます。ツールによっては誤検出の発生件数が非常に多く、どのエラーを重点的に修正すればコード品質を改善できるのかが分かりにくいものもあります。

スタティック検証は、開発の早期段階で非常に多くのバグを取り除くのに役立ちますが、それだけではアジャイル開発プロセスを加速することはできず、いずれかの時点でソフトウェア開発チームがコードを実行してその結果を解析する必要があります。

ダイナミック・テストは、アプリケーションを実際に動作させて仕様への適合性を検証するもので、これにはUX(ユーザー・エクスペリエンス)テストなどがあります。ダイナミック・テストは、スタティック解析テストで報告された不具合が誤検出なのか実際の不具合なのかを確認するのにも役立ちます。機能の正しさを検証するには、テスト担当者が自動および手動のテストケースを作成する必要があります。

製品を適切にテストするには、テストで不具合を発見したらコードを修正し、同じテストを再度実行するという手順を繰り返す必要があり、テストケースのスケジューリングと実行には多くの時間が費やされます。しかしほとんどの場合、テスト担当者はコードのどの部分が変更されたかを知りません。このため、不具合を見逃すリスクを避けるためにすべてのテストを何度も繰り返し実行しているのが現状です。

手動テストの場合、テスト環境を準備し、テストを実行して結果を文書化し、テスト環境をクリーンな状態に戻すという一連の作業をすべて人手で行う必要があるため、特に時間と労力がかかります。その上、本来不要なテストを繰り返し実行していては、時間のロスはさらに大きくなります。

シノプシスのCoverity Software Testing Platform

シノプシスのCoverity Software Testing Platformを利用すると、ソフトウェア・チームは、開発プロセスの早期段階で品質 / セキュリティ・テストを実施できるため、アジリティ、レジリエンス、予測性を備えたソフトウェア開発が実現します。

ダイナミック・テストの時間を短縮

シノプシスのCoverity Test Advisor QA Editionには、個々のテストケースでテスト実施済みの部分、およびコードの変更箇所を認識する機能があ

次ページに続く▶

前ページより続く ▶

り、インクリメンタルなダイナミック・テストが可能です。このため、同じテストを繰り返す無駄が省け、テスト時間が大幅に効率化します。

処理速度とスケーラビリティ

スタティック解析ツールは無償やオープンソースのものを含め多くの製品が出回っていますが、ソフトウェア・チームのニーズに最適なプラットフォームを選ぶのは容易ではありません。小規模なデザインなら問題なく動作しても、解析するコード量が増えると膨大な時間がかかる製品も多く存在します。シノプシスのスタティック解析ツールは、カーネルを含め数百万行のコードを含むプロジェクトも短時間で効率よく解析できます。短時間でより多くのバグを検出・修正できるようになるため、リスクを負うことなく開発期間を短縮できます。

精度とユーザビリティ

多くの解析ツールは、波括弧の位置や変数名が正しいかなど、単純な問題を指摘します。これは小規模なコード・フラグメントには役立つかもしれませんが、大規模なプロジェクトでは短時間で大量の違反アラートが発生し、開発チームはどの「問題」から先に対処すればいいのか迷ってしまいます。Coverity Software Testing Platformは、開発者が優先的に対処すべき問題に集中できるように設計されており、ツールとしての価値を高めています。

ワークフローと自動化

Coverity Software Testing Platformは、複数のソースおよび解析ツールからの情報を1ヶ所に集約でき、あらゆる開発作業のハブとして利用できます。この一元化されたフレームワークにすべてのツールの出力を統合し、ワークフローの自動化およびレポート機能をカスタマイズできます。このため、チーム内で問題の優先付けと修正が効率よく行え、開発およびコンプライアンスの進捗状況をリアルタイムに可視化できます。

製品ラインナップの拡充

Codenomicon社の買収

シノプシスは先ごろ、半導体および電子デバイスの組込みソフトウェアを強化するため、ソフトウェア・セキュリティの世界的企業であるCodenomicon社を買収しました。Codenomicon社はハートブリード・バグを独自に発見・報告したことでも知られています。同社の人材・技術・製品が加わることにより、ソフトウェア資産をセキュリティ脆弱性から効果的に保護する上で必要な可視性とリアルタイム・インテリジェンスがCoverity Software Testing Platformに追加されるなど、ソフトウェア・セキュリティ確保に向けたシノプシス・ソリューションが強化されます。

データ保護に対する関心が高まる中、企業は最も重要なソフトウェア・アプリケーションの完全性・秘匿性・安全性を確保する必要に迫られています。ファイルおよびプロトコル・レベルのテスト / ファジング・ツールのCodenomicon Defensics、およびソフトウェアに使用されているサードパーティ・ライブラリの検出および脆弱性評価ツールのCodenomicon AppCheckがCoverity Software Testing Platformに加わることにより、シノプシスが提供するセキュリティ・ソリューションはソフトウェア開発ライフサイクル全体をより包括的にサポートできるようになります。

Quotium社製品のSeekerを買収

シノプシスはQuotium社から、同社資産の一部である同社製品SeekerとそのR&Dチームを買収することで最終合意に達しました。これにより、Coverity Software Testing Platformに対話型アプリケーション・セキュリティ・テスト（IAST）の機能が加わり、アプリケーション・セキュリティ確保に向けたシノプシス・ソリューションが強化されることになります。実行中のアプリケーションに対するテスト / 解析機能を持つSeekerは、Coverity Software Testing Platformのスタティック解析技術を補完する役割を果たし、ユーザーはより包括的にセキュリティ脆弱性を検出・修正し、ビジネス・データをアプリケーション攻撃から効果的に保護できるようになります。ソフトウェア開発ライフサイクルへのセキュリティ・テスト技術の導入を加速することにより、金融、健康、エネルギー、小売などの産業向け製品の安定性と確実性が向上し、製品の早期市場投入が可能となります。

まとめ

QAやセキュリティなどの検証作業は、開発工程の終盤で実施すると多大な労力とコストがかかります。これを開発工程初期段階へ前倒しする「シフトレフト」により、少ない労力で不具合の修正が可能となり、セキュリティ・品質・スケジュール面でのリスクを軽減し、製品リリース・サイクルの効率と予測性を改善できます。

現在、アジャイル手法を採用するソフトウェア開発チームが増えています。適切なツールを選べば品質 / セキュリティ・テストの一部を初期段階へと前倒しできるようになり、アジャイル・プロセスの一環として検証作業を加速してリリース・プロセスを合理化できます。こうすれば開発者はバグの追跡から解放され、その時間をコーディングに費やすことができます。

Coverity Software Testing Platformとして幅広いアプリケーション・スイートを提供してきたシノプシスは、Codenomicon Defensics / AppCheckおよびQuotium Seekerの高度なセキュリティ・テスト技術をラインナップに加えるなど、ソフトウェア開発プロセスをサポートする先進のソフトウェア開発ツールの拡充に努めています。

詳細情報

- ウェブサイト：Coverity　http://www.coverity.com/html_ja/

著者紹介

Jon Jarboe: これまで20年以上にわたり、組込みシステムからエンタープライズ・アプリケーションまで幅広い分野でソフトウェア開発ツールの改良に従事。ソフトウェア開発ライフサイクル（SDLC）全体で統制されたソフトウェア・テストを実施することによる品質・セキュリティ・生産性の利点を強く提唱。現在所属するシノプシスのソフトウェア・インテグリティ・グループでは、開発チームが直面している課題の把握に努め、これら課題を解決するツールの開発に助言を行うとともに、統制されたテストによって開発のスピードと機敏性が向上し、品質・セキュリティ・効率が改善することへの理解を広める取り組みにも尽力。