SYNOPSYS®

アプリケーション セキュリティ マネージャーのためのアジャイル開発

SYNOPSYS®

アプリケーション セキュリティ マネージャーのためのアジャイル開発

はじめに

競争が激しい今日のビジネス環境において、アプリケーションを適切に開発するだけでなく、迅速に開発することが、これまで以上に重要になっています。従来の「ウォーターフォール」方式は確かに有効な方法ですが、多くのステップが必要とされるため、現在のニーズには合わなくなっています。アジャイルは、開発を大幅にスピードアップする他にもさまざまな利点があるため、一般的になった開発方式です。

アプリケーションの脆弱性は、ビジネス上重要なデータに対し、かつてないほどの脅威を継続的に提起しています。各組織は、自らのWebアプリケーションに起因する持続性の脅威と戦っています。

アジャイルとアプリケーション セキュリティは共存できない、言い換えれば、アプリケーション セキュリティはアジャイル開発チームには対応不可能な要件であると多くの人が考えています。すなわち、アジャイル開発は機敏で無駄が少なく、セキュリティといった問題に煩わされることを嫌うため、アプリケーション セキュリティを導入しようとすると開発プロセスに大きな悪影響が出ると考えられています。

しかし、アジャイル開発プロセスにおいてアプリケーション セキュリティの実装に成功したと報告している組織がいくつもあります。では、どのようにして成果をあげているのでしょうか。本ホワイトペーパーは、アプリケーション セキュリティの観点からアジャイル開発を分析し、アジャイル開発方式にセキュリティを実装するために何ができるかを示唆します。

アジャイルの原則について

純粋なアジャイル開発の定義は、「アジャイル ソフトウェア開発宣言」[1] に表現されています。アジャイルの純粋な作法をそっくりそのまま採用している組織はほとんどありませんが、アジャイルの原則に基づいた方式が多くの組織で実施されています。アジャイルの原則の概要は以下の通りです。

責任は開発者が負う

開発チームは責任を負うと同時に、ミッションが無事完了するまで信頼されます。最良なコミュニケーションは、すべてのステークホルダー間の対面コミュニケーションです。

コードは常にアップロードされ更新される

臨機応変な変更が実施され、必要に応じて最終製品にも適用されます。アジャイル プロジェクトのライフサイクルは、区別可能な状態ではなくむしろ、ほぼ持続的に行われる開発およびテストで構成されます。

これは、XP（エクストリーム プログラミング）ソフトウェア エンジニアリング プラクティスと呼ばれる手法で、すべての作業用コピーが、多い場合で1日に数回、メインソースにマージされます。これにより、インテグレーションに時間がかかる問題が解決され、結果として、プロジェクトのステータスが常に変化します。

プロセスの最後に要件が決まる

アジャイル チームは、プロセスを通じて要件が変化し発展していくため、要件の文書化という初期投資は無駄になることを認識しています。初期段階では、プロジェクトのスコープを特定するために、必要最小限の要件の想定のみを行います。要件が確定するのは、プロセスがかなり進んで、チーム全体が最終的な結果を理解したときです。

顧客志向の要件

プロジェクトは顧客の協力によって完結し、顧客がプロジェクトの要件およびスコープを最終的に決定します。最終的なゴールは、動くソフトウェアをタイムリーに顧客に届けることです。

[1] http://www.agilemanifesto.org/

ユーザー ストーリー
ユーザー ストーリーとは、プロジェクトの職務の一部として、各ユーザーが何をしなければならないかを簡潔に説明した文章です。これは、要件管理を促進する1つの要素になります。

自動化による継続的なテスト
テストおよびテストケースを自動化することにより、アジャイル開発プロセスの各ステップを通じて、より頻繁なテストが可能になります。これにより、アジャイル方式の特徴である短い開発サイクルであっても、品質の高いソフトウェアが開発可能になります。

無駄のない開発
アジャイル方式は、リーン(無駄のない)生産方式の原則をソフトウェア開発プロセスに応用しています。この方式では、無駄を最小化すると共に開発チームに権限を持たせることで、完成した製品をできるだけ早く、低価格で提供します。

業務担当者の協力
開発者とステークホルダーの日常的かつ継続的な協力は、アジャイル開発の成功には不可欠です。

堅牢な動くソフトウェアの定期的なデリバリー
デリバリーの主要な尺度は、必要な機能が備わった動くソフトウェアを作ることです。デリバリーは、定期的なペースで行われます。優れたソフトウェアを作ることが第一に重視されます。

自主的に改善されるプロセス
チームは定期的に、現状のプロセスの問題点について内部で検討し、その解決策を自らのプロセスに反映させます。

アジャイルにおけるアプリケーション セキュリティの実現
一言で言えば、アジャイル開発は、迅速な開発を可能にし、検討や要件定義の必要性を低減するために用いられます。すなわち、この手法は、顧客のニーズを短期間で満足することを目的としています。顧客のフィードバックおよびテストに基づいて、製品に対するイテレーション サイクルまたはスプリントが繰り返されます。このプロセスは、アプリケーションが顧客要件およびフィードバックによって決定されるゴールに到達するまで続けられます。

どうすればこのプロセスにセキュア開発やアプリケーション セキュリティ テストを当てはめることができるのか疑問に思う方もいるでしょう。もしプロセスが終了するまで待っていたら、従来型のプロジェクトに後戻りすることになり、問題となるでしょう。従って、プロセスの各ステップにセキュリティを組み込むことが重要なのです。アジャイル方式を用いた速いペースで進む開発において、どうやってアプリケーション セキュリティを達成するのかを示す原則を以下に紹介します。

アジャイル開発はチームの協力の上に成り立っていますので、アプリケーション セキュリティもチームとして取り組む必要があります。また、アジャイルは無駄がないことおよびスピードが速いことで知られていますので、アプリケーション セキュリティをそのプロセス内で実施するには、同じ原則に従う必要があります。

明確な要件を定義する
開発チームおよび品質保証チームは、セキュリティが提示する要件に困惑してしまうことがよくあります。多くの場合、開発者は、セキュリティ手順は冗長で、単に開発チームの作業を増やしているに過ぎないと感じています。これは、セキュリティ テストがプロジェクトの最終段階でしか行われない場合に特に顕著になります。

このため、明確な期待値を、理想的には最初から、定義しておくことが重要です。開発チームおよび検証チームは、求められるセキュリティのレベルが何であるかを知り、このレベルに到達することの意味を理解する必要があります。さらに、どのようなセキュリティテストが実施され、どのような結果が求められるのかについても理解しておく必要があります。セキュリティに関してチームは何を重視すべきかを詳しく説明することにより、開発チームは、テストプロセスそのものおよび背後にあるロジックに関する情報に基づいて、セキュリティをプロジェクト全体に組み込むことが可能になります。開発チームのために以下の質問に答えることは、要件定義の第一歩として役立つでしょう。

- セキュア開発およびセキュリティのテストを行うために特に重視すべき分野は何か。
- これらのテストによって定期的なペネトレーション テストやセキュリティ監査が代替されるのか、あるいはこれらのテストと並行して実施されるのか。
- 開発者はどのくらいの頻度でセキュリティのテストを行わなければならないか、およびこれらのテストの責任者は誰か。
- 開発チームが取り組まなければならないのはどのセキュリティ標準か。(これには、OWASPやPCI-DSSなどの業界スタンダード、社内要件やその他の基準が使用可能です。)

このような質問に答えることで、開発チームは、疑念や曖昧さといった出発点からではなく、理解と協調に基づいて、アプリケーション セキュリティを開発プロセスにうまく統合することが可能になるのです。

要件を与えるのみではなく、トレーニングやセキュアなフレームワークの提供などを通じて、要件を達成するというゴールに向かって

支援することが重要です。

プロセスと対立するのではなく、プロセスに協力する

アジャイル開発チームは、プロジェクト ライフサイクルで用いられる確かなプロセスを持っています。アジャイルに関係するほとんどすべてのものと同様、これらのプロセスは軽量かつ無駄がないことを目的としています。アプリケーション セキュリティをアジャイル開発にうまく統合するための最良の方法は、既存のインタラクションと一緒に取り組むことです。この方法により、作成されたコードがプロジェクトのすべての要件に適合するばかりか、セキュアで質の高いものになるのです。アジャイル ソフトウェア開発宣言には、「プロセスやツールよりも個人と対話(インタラクション)を」重視することが述べられています。セキュリティ関して言えば、ツールを完全に忘れることはできません。従って、使用するツールは、全員が理解可能な共通言語を使ってチームメンバー間のインタラクションを促進し、プロセスの遂行を妨げないものでなければなりません。

すなわち、もしチームが特定のバグトラッキング ソフトウェアを使用しているとしたら、セキュリティの問題は同じインターフェースでバグとして提示される必要があります。もしテストの自動化およびリグレッション テストがすでに行われているとしたら、既存のプロセスに統合可能なアプリケーション セキュリティ ツールを選定してください。また、開発者が理解できる言葉で結果を返すツールが優先されるべきでしょう。さらに、脆弱な箇所がコード内で特定され、修正の優先度に関する議論を避けるために、脆弱性のリスクレベルに関する分かりやすい解説や具体的な説明が提供されることが望ましいでしょう。

セキュリティは原則的に、プロジェクト最終段階で手戻りが発生する恐れがある別プロセスにするのではなく、アジャイル プロセスの各ステップに統合されなければなりません。

頻繁なコード変更に対応する

アジャイルでは頻繁にコードの変更が行われ、実際それが推奨されています。だからこそ、セキュリティ テストも同じ基準に応えることが非常に重要です。アジャイル方式におけるアプリケーション セキュリティも、迅速な変更が可能でなければならないのです。アプリケーション セキュリティがこの継続的なセキュリティ対応のニーズに応えることができれば、解決困難なハードルや、開発者にアジャイルの原則から外れた作業を強いる可能性がある問題などで設計者を悩ませることなく、セキュリティを保護することができます。

これは、実行に長時間を要するツールや、結果を手作業で解釈しなければならないツールは、このような環境では効果を発揮できないということを意味しています。すなわちこれは、ツールによるテストが完了し、セキュリティ チームのレビュアーがレビューを完了する頃には、新たなコードの製造を始めてから数日あるいは数週もたってしまっているという事態があり得るためです。

セキュリティ ストーリーを作成する

ほぼすべてのチームにおいて、要件の定義の少なくとも一定レベルに、ユーザー ストーリーが使われています。アプリケーション セキュリティの要件をユーザーストーリーの形式で提示することによって、慣れた手順が維持され、アジャイル プロジェクトの標準に当てはめて処理することが可能になります。さらに、非常に重要かつ忘れてはならないことは、セキュリティは全員で取り組まなければならない仕事であり、他の人やチームに任せておけばよいというものではないということです。セキュリティをユーザーストーリーとして説明することにより、開発者にそれが自分のタスクの重要な一部であることを認識させると共に、セキュリティを開発およびテストサイクルを構成する一つの要素にすることができるのです。

アジャイル アプリケーション セキュリティ ワークフローの構築に協力する

セキュリティに関して何が期待されているのかをアジャイル開発者に説明しましょう。それから、彼らと直接協力して、現行の習慣、イテレーションおよびデッドラインに適合するワークフローを作成してください。開発チームからよく聞かれる質問に、次のようなものがあります。

- セキュリティ テストは誰が実行するのか。各開発者が自分のコード対して実行するのか。あるいは、セキュリティテスト専用にQAメンバー1名を配置するのか。
- セキュリティ テストの実施頻度はどの程度か。テストは各コードに対して実施されるのか、またはインテグレーション後に実施されるのか。
- テスト結果は誰に送られるのか。開発チームか、それともセキュリティ チームか。
- 誰がサインオフ(承認)の責任を負うのか。

アジャイル アプリケーション セキュリティ ワークフローの意義は、無駄がなく迅速であるというアジャイルの原則を維持しつつ、アプリケーション セキュリティがすべてのフェーズに組み込まれた開発プロセスを構築することです。

トレーニングプログラムを提供する

多くの開発者は、アプリケーション セキュリティを適切に理解しテストを実施するために必要なトレーニングを受けていません。また、もしトレーニング プログラムを整備したとしても、開発者の入れ替わりにより、全員に最新の知識を付与することは非常に困難

です。開発チームに責任を移譲する前に、プロセスを無理なく行えるよう、十分な情報を付与する必要があります。

このため、プロセスにトレーニング機能を備えたツールの使用が推奨されます。このようなトレーニングは、現行のプロジェクトにとって有益であるばかりでなく、同様の手法で開発される将来のプロジェクトにも役立ちます。これは、アジャイル開発の利点の1つで、将来のプロジェクトはずっと簡単に行えるようになります。

ここで言うトレーニングとは、必ずしも膨大な項目をカバーし最後にテストが行われる2週間もかかるようなコースではないことを覚えておいてください。しかも、そのようなトレーニングで学習した内容は、ほとんどすぐに忘れられてしまうのが常です。有効なトレーニングとは、例えば、ある開発者のコードに特定の脆弱性がある場合、当該開発者はその脆弱性に関する短いトレーニングを受けることができたり、また、現在作業しているアプリケーションに関連する脆弱性に特化したトレーニングを受けることができたりすることを意味します。

ミスを恐れず、経験して向上する

アジャイル開発とは、プロジェクトの進行に合わせて学ぶことです。すべてのイテレーションは改善と変更を含んでおり、最終的な結果に向かって徐々に進みます。アプリケーシ ョン セキュリティ プロセスも、途中で変更と改善を行いながら、同じ方法で完成させればよいのです。

まとめ

セキュアなソフトウェアは、いかなる方法で開発されようとも、適切にテストが実施され、適切なセキュリティ対策に従っています。アジャイル開発方式はセキュアなコーディングプラクティスおよびアプリケーション セキュリティ テストと相いれないという共通の認識は、ほとんどが間違いです。また、柔軟性を維持しながら、アプリケーション セキュリティをアジャイル開発システムに統合することも可能です。

開発者は、アジャイル開発にセキュリティを取り入れないことによるリスクに目を向けるべきです。ソフトウェアがセキュリティの観点から十分にテストされない場合、データ喪失の危険を持ったソフトウェアや、ハッカーの攻撃を受けやすいプログラムを開発するリスクが高まります。セキュリティ テストにはコストが必要なことは確かですが、一般的に、テストが不十分なことによって発生するコストより低いと言えます。

SDLCとの親和性が高いアプリケーションセキュリティ — Seeker

シノプシスのSeekerは、ソフトウェア開発ライフサイクル(SDLC)を通じて、ランタイムコードおよびデータを解析するアプリケーション セキュリティ テスト ソリューションです。Seekerは、模擬攻撃に対するアプリケーションの振る舞いを解析し、真の脅威を提起するコードの脆弱性を検出します。Seekerはまた、ビジネス上クリティカルなデータに対するリスクを実証するエクスプロイトを生成して、脆弱性の管理を支援します。Seekerは、SDLCを通じたアプリケーション セキュリティ テストに最適なソリューションで、完全な自動化、アジャイルとのシームレスな連携、および統合環境の維持が可能です。Seekerには以下のような利点があります。

- 開発プロセスとアプリケーション セキュリティ テストのシームレスな統合、ならびに機能テスト、ビルドされたサーバーおよび他のあらゆる自動化への統合を実現する強力かつ革新的なインターフェース。必要なことは、すべての自動テストが置いてある場所にSeekerを組み込むことだけです。
- 脆弱なコードがハイライトされ、修正方法が提供されるため、開発者およびテスターの作業が最小化されます。
- リスクは簡単な言葉で説明されていますので、各ステークホルダーは簡単に優先順位を設定し、脆弱性管理計画を立案することが可能です。
- 開発者には原因および対策の両方が示されますので、シナリオの再作成、問題があるコードの参照、および簡単な修正ソリューションの入手が1カ所から行えます。
- 脆弱性はバグと同様に管理可能です。
- 高速なテストプロセスにより、新たなコードのテストやリグレッションテストの一部として、アプリケーションのフルテストを短時間で実行可能です。

Seekerを使うことによって、アジャイル開発チームの開発プロセス中にセキュリティが実装されるようになります。これにより、アジャイルフレームワークの制限の範囲内で、プロジェクトを計画通り完了させることが可能ですので、大きな投資対効果に加えて顧客満足の向上が期待できます。Seekerはシンプルで、導入時の負担や開発プロセスの変更をほとんど必要としないため、すぐに結果を出すことができます。

シノプシスについて

Synopsys, Inc.(Nasdaq上場コード:SNPS)は、我々が日々使用しているエレクトロニクス機器やソフトウェア製品を開発する先進企業のパートナーとして、半導体設計からソフトウェア開発に至る領域(Silicon to Software)をカバーするソリューションを提供している。電子設計自動化(EDA)ソリューションならびに半導体設計資産(IP)のグローバル・リーディング・カンパニーとして長年にわたる実績を持ち、ソフトウェア品質/セキュリティテストの分野でもCoverityソリューションで業界をリードしており、世界第15位のソフトウェア・カンパニーとなっている。シノプシスは、最

先端の半導体を開発しているSoC（system-on-chip）設計者、最高レベルの品質とセキュリティが要求されるアプリケーション・ソフトウェアの開発者に、高品質で信頼性の高い革新的製品の開発に欠かせないソリューションを提供している。カリフォルニア州マウンテンビューに本社を置くシノプシスは、北米、南米、ヨーロッパ、日本、アジアおよびインドに約113箇所の拠点を有している。

SYNOPSYS®

日本シノプシス合同会社 営業本部 ソフトウェア インテグリティ グループ
〒158-0094 東京都世田谷区玉川2-21-1二子玉川ライズオフィス
TEL: 03-6746-3600 Email: sig-japan-sales@synopsys.com

http://www.synopsys.com/JP2/Software