セキュリティ開発ライフサイクル（SDL）におけるQAの役割

セキュリティ対策をソフトウェア開発プロセスに組み込む方法とQAの役割について考察しました。実際の適用は未だですが…

OWASP ASVS／Testing Guide／
MS Threat Modeling Tool 2016
を使用。

SWセキュリティ対策グループ
元キヤノン㈱　永田哲（リーダー）
㈱リンクレア　早崎伸二
キヤノン㈱　日下宏
㈱モバイルインターネットテクノロジー　二川勇樹

謝辞

以下の方々にご協力をいただきました。
この場を借りてお礼申し上げます。

・アンケートに協力頂いた「部長の会の皆様」
・「セキュリティ品質の確保」についてご講演いただいた「株式会社 神戸デジタルラボ」の長山氏
・「セキュリティ品質活動」についてご講演いただいた（株）日立ソリューションズの隅垣氏

2/7放送のNHK「サイバーショック」からの教訓

- 北海道新幹線の安全機構を含めて日本の技術は盗まれている（約1000社から２万件）

  → 年金機構を仕組まれたのと同じ遠隔操作マルウェア

- 脆弱なサイトが隠れ蓑になっている

  →「小さいから狙われない」ということはない。踏み台にされる。

  → 鎖は最も弱い環から切れる（攻撃しやすい）

  → 脆弱なサーバー、アプリが攻撃に加担していることになる(攻撃インフラの一部になっている)

  ➡ 製造者のセキュリティ意識の希薄さと無責任さが問題

南アの銀行にハッキング　ＡＴＭ不正直前、引き出し承認消す？

2016/6/28 13:39

全国17都府県のコンビニのＡＴＭで現金約18億円が一斉に引き出された事件で、顧客カードの情報が流出した南アフリカの銀行のシステムが不正アクセスに遭い、**引き出しの直前にプログラムが誤作動を起こしていた**ことが28日、捜査関係者への取材で分かった。

捜査関係者によると、引き出しには南アのスタンダード銀行発行の顧客カード情報が入った偽造カードが使われた。引き出しが行われた５月15日早朝の約３時間、同銀行のシステムには出金を承認した形跡がなかった。**ハッキングで誤作動を引き起こして承認させた後、形跡を消去した可能性があるという**。

偽造カードの作成では、**同銀行から流出した約３千件の個人情報が悪用された**ことが既に判明。警察当局は犯行グループが事前に偽造カードを用意するため、個人情報もハッキングで不正に入手した可能性が高いとみている。

ある金融情報技術（IT）コンサルタントは、犯人が日本を選択した理由について「リスクが低いほか、ATMのネットワーク管理が甘く、不正分析ソフトによって見破られないと考えたからだろう」と指摘。

出典元：日本経済新聞（電子版）

脆弱性の90%以上がSQLインジェクション

- 2016年05月14日 15時03分

中国のハッカーサイトが今年２月以降、セキュリティーに穴（脆弱性）のある日本の公的機関や企業のサイト名１００件以上を公表していることがわかった。放置すればサイバー攻撃の入り口にもなる危険な「穴」だが、日本では非公開が原則だ。「公表されるとかえって攻撃者に狙われる」と関係者は困惑するが、危険を表面化させない日本の対応が大量の脆弱性放置につながっているとの指摘もある。　公表しているのは、中国のセキュリティー専門家らが運営する脆弱性報告サイト「WooYun」。ハッカーらが脆弱性を見つけて投稿し、一定の猶予期限が過ぎると、脆弱性が修正されてもされなくても、サイトに詳細が掲載される仕組みだ。発見者に報酬はないが、発見内容が評価され一流企業に迎え入れられるなど、ハッカーの登竜門になっている。これまで中国のサイトなどを中心に１０万件以上の脆弱性が報告されたが、今年２月からは日本のサイトについての投稿も目立つようになった。 出典元：読売新聞

セキュリティ対策漏れの責任を開発会社に問う判決(2014)「東京地裁平成23年(ワ)32060号」

- X社が運営するECサイトに対して、外部からの不正アクセスにより、最大7316件のクレジットカード情報が漏洩した。
- X社は謝罪、対応、調査等の費用、売上減少による損害等に関して、Y社（開発会社）に対して、委託契約の債務不履行にもとづき1億円余りの損害賠償を請求、東京地裁に起訴した。結果、**原告が勝訴**。

東京地裁の判決：

- クレジットカード情報が漏洩した原因は複数考えられるが、脆弱性やアクセスログ、不正利用の状況から**SQLインジェクション攻撃によるものと断定**
- **セキュリティ対策についてX社からの要求提示はなかったが、Y社は必要なセキュリティ対策を講じる義務（債務）があり、それを怠った債務不履行がある**
- **要求仕様書に記載はないがSQLインジェクション対策を怠ったことは重過失である**

（→経済産業省はIPAが紹介するSQLインジェクション対策の措置を重点的に実施することを求める旨の注意喚起をしていた、など。）

出典元：http://blog.tokumaru.org/2015/01/sql.html

21世紀の品質はセキュリティが必須

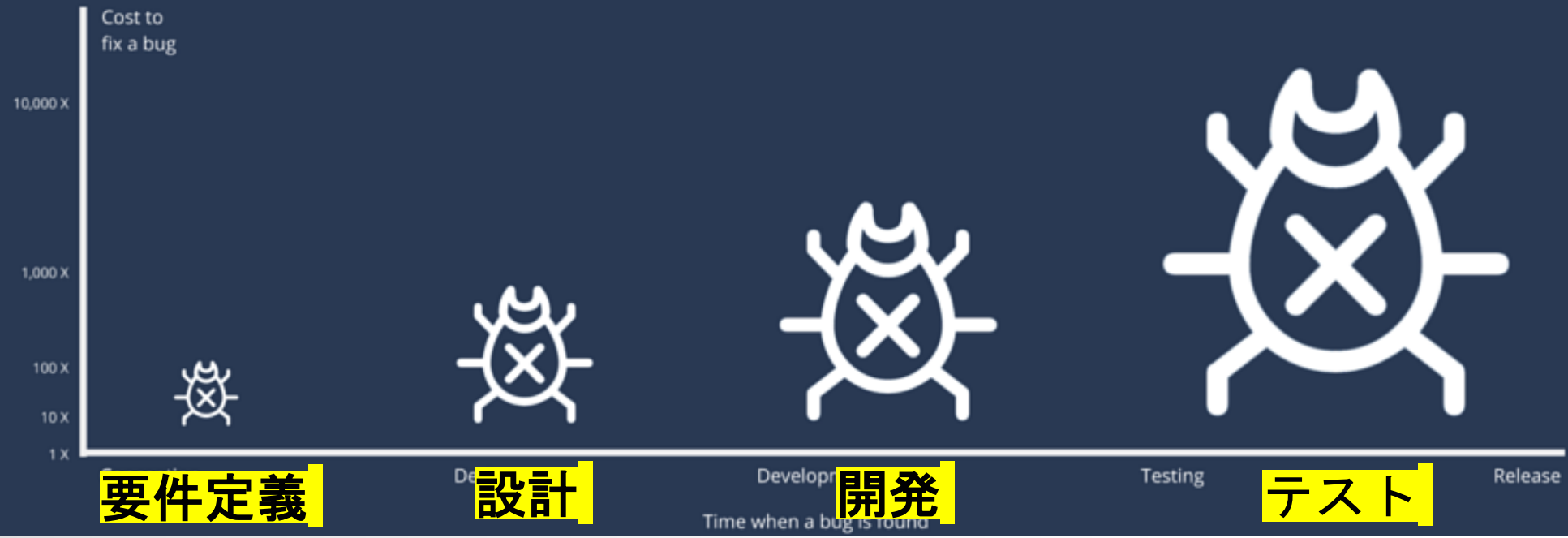
- セキュリティ品質の確保は製造者責任。従って開発者／QA担当者へのトレーニングと開発工程への新たな投資が必要。
- セキュリティに関する意識とスキルは普通の開発者やQA担当者が身につけるべきもの（セキュリティの専門家だけで防御しきれるものではない）
- セキュリティ品質はどこまでやればいいのか？　MITREのCAPEC（約500の攻撃パターン）やCWE（約700の脆弱性）を基準にするのか？　まずはOWASP ASVSを参考にする。
- **QA部門はOWASP ASVS V3のLevel2 or 3の検証要件をプロダクトが満足することを確認し、証跡を残す**
- **QA責任者は経営者に代わってプロダクトのセキュリティリスクを判断する**
- 守る側のスキルとモチベーションを教育と意識改革によって上げる必要がある。（攻撃者はスキルもモチベーションも高い）
- **特にテスト担当者にはスキルアップのチャンスと考えるべき（外部ペンテスタを100%信頼しますか？）。攻撃者目線でアビューズケースを考えるのは楽しい！？**
- セキュリティの専門家は大幅に不足しており、かつ製品の内部構造を良く理解しているわけでは無い。➡ **自分達の製品は自分達で守る意識／スキルとプロセスが必要**

ソフトウェア開発の鉄則

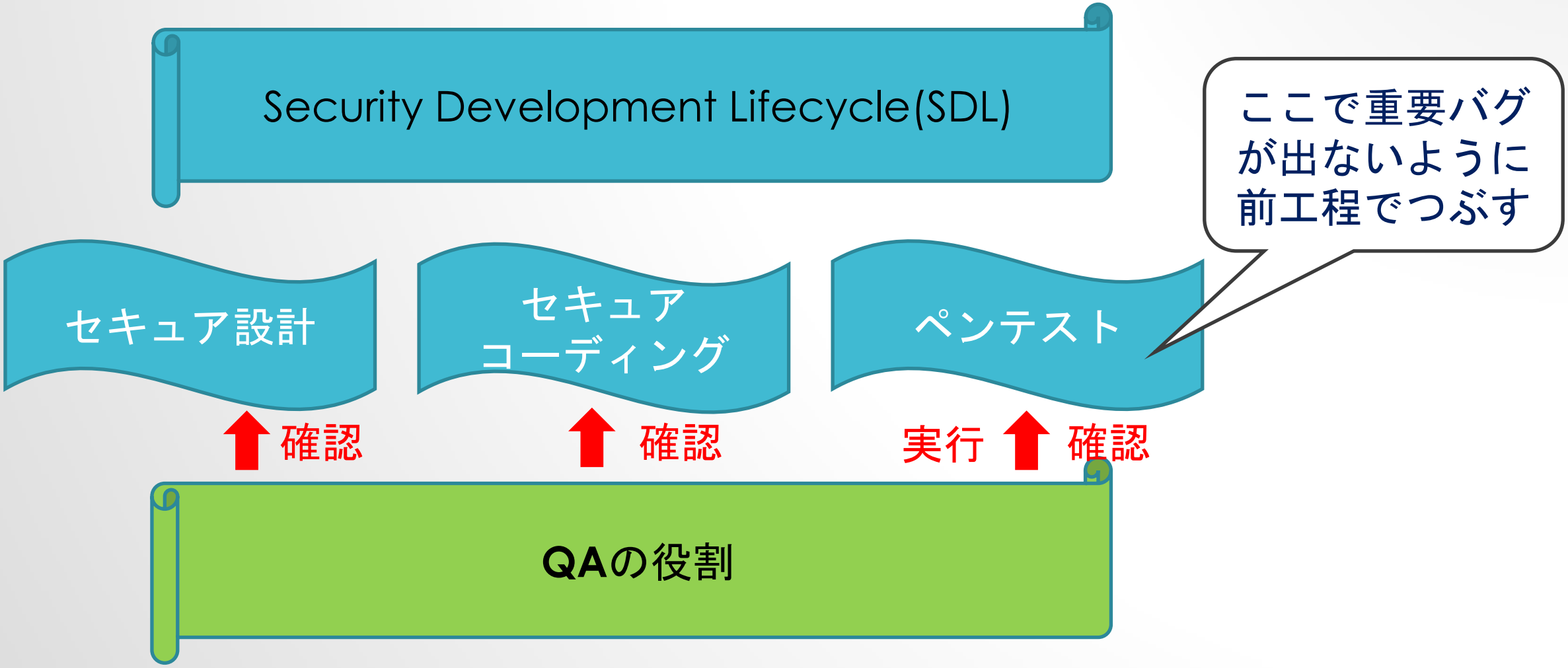
配布せず

コストと日程遅延を最小化するには

バグを早く見つけて修正する

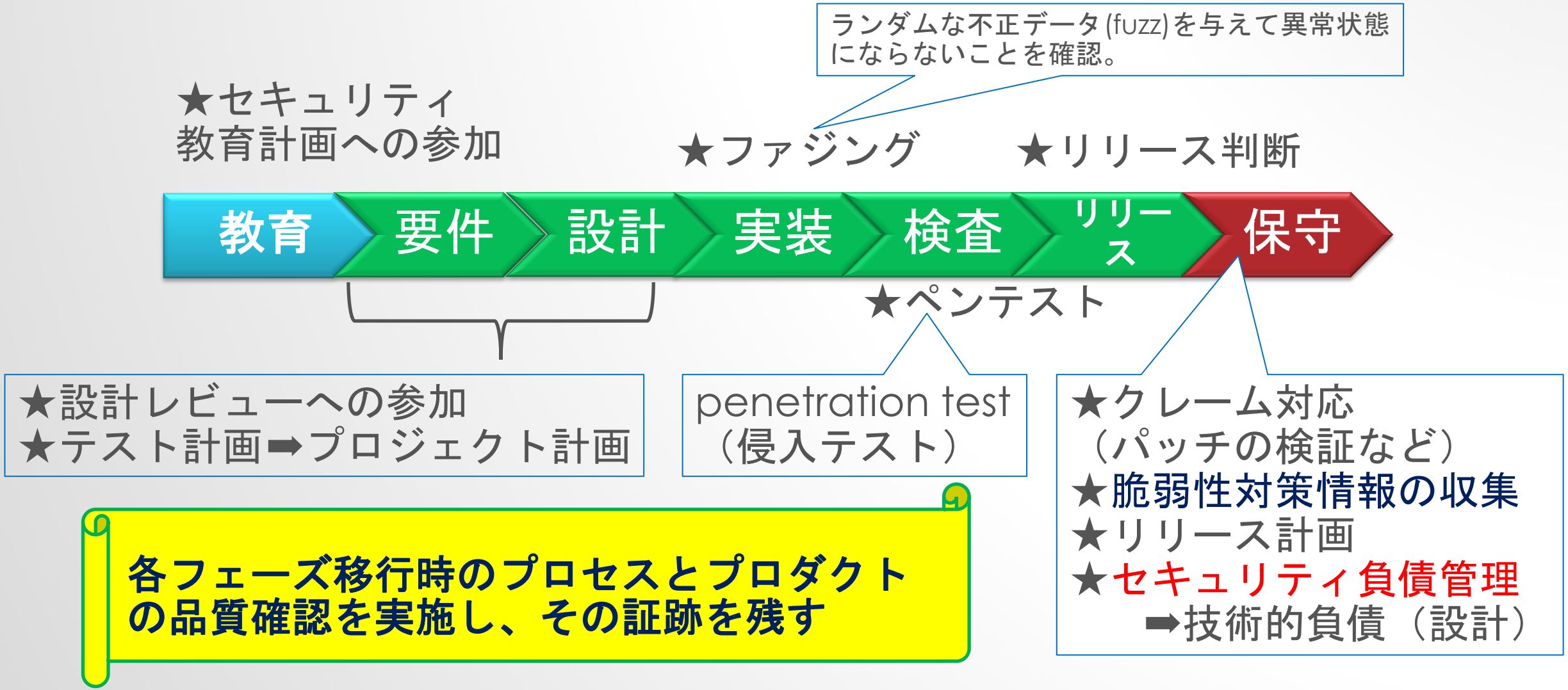


ソフトウェア・セキュリティが主対象



開発者もQA担当者も意識とスキルが必要

セキュリティ開発ライフサイクル(SDL)におけるQAの役割



各フェーズ移行時のプロセスとプロダクトの品質確認を実施し、その証跡を残す

セキュリティ対策の実態

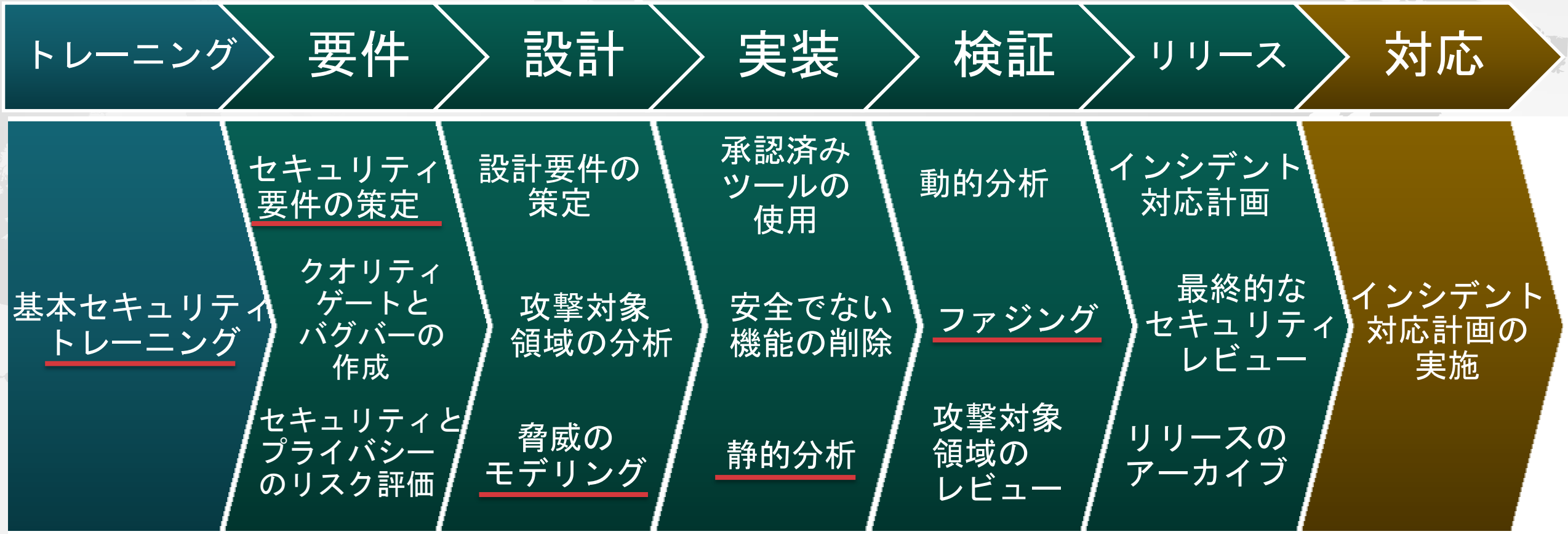
1. 不要なポートを閉じる、あまり使われない機能はデフォルトでオフ（攻撃表面（アタックサーフィス）の最小化）
2. 静的解析ツールによるソースコード上の脆弱性の排除（当然、セキュリティ対策機能抜けはチェックできない）とペンテスト

---------- ここまではやられているケースが多い ----------

3. 攻撃者目線でアタックを想定し、リスクベースでセキュリティ対策機能を設計段階から組み込む

➡ セキュリティ対策を組み込みながら開発者をスキルアップする方法
MSが約15年前にこの問題を解決したいと考えて、
2006年に"The Security Development Lifecycle"として公表。
SDLプロセスと脅威分析パターンを載せている。

SDLのベストプラクティス（MSのSDL）



Japanese_Simplified Implementation of the SDL.docxより
以下も参照
https://www.microsoft.com/en-us/sdl/default.aspx

設計バグと実装バグは半々（Cigital社）

Cross Site Scripting
Buffer Overflow

セキュリティ対策が無い／間違っている／弱い

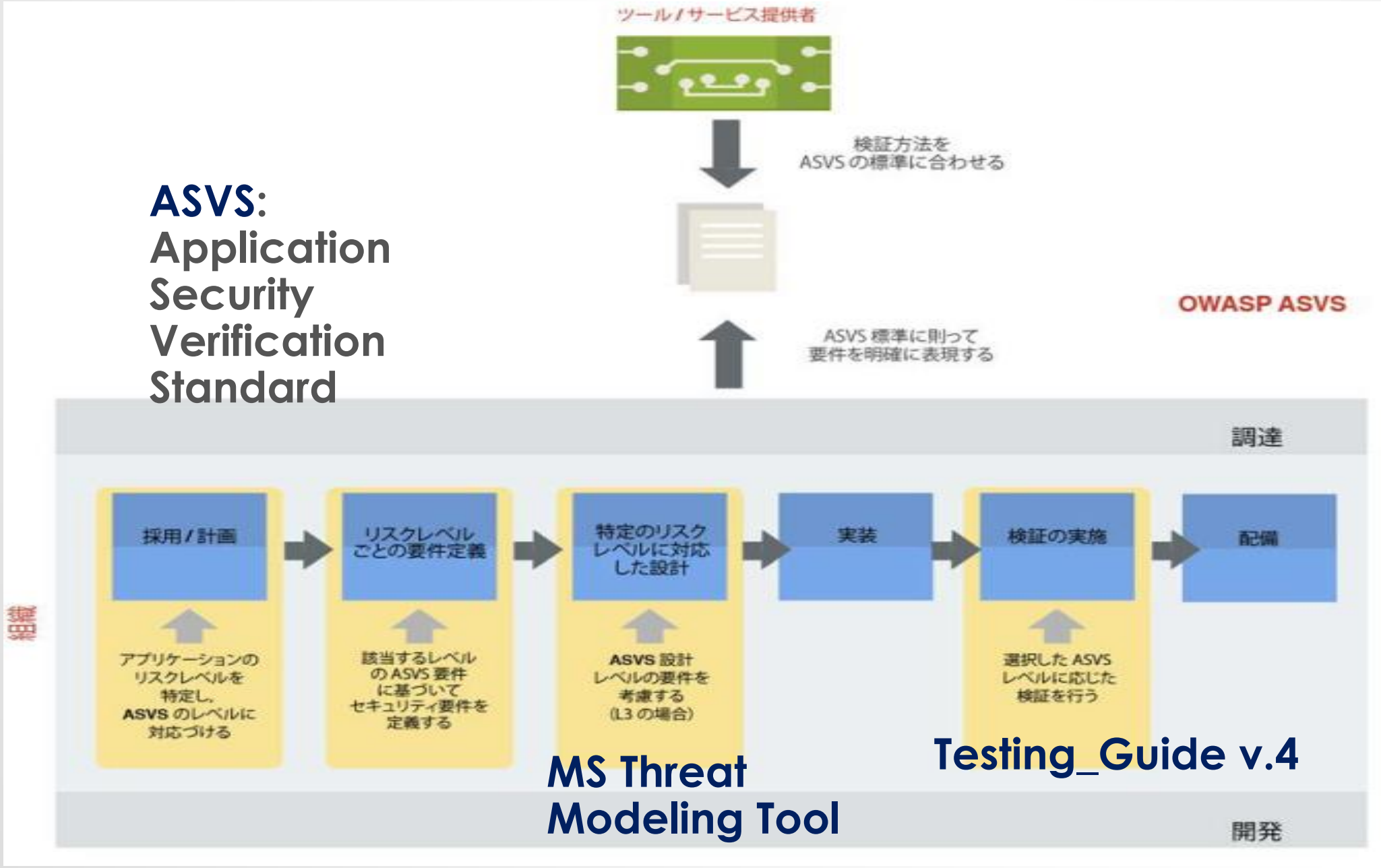
実装バグ　　設計バグ

Code Review　静的解析ツールを含む

Penetration Testing

Architecture Analysis

ASVS:
Application Security Verification Standard



MS Threat Modeling Tool

Testing_Guide v.4

要件フェーズにおけるQAの役割

- 要件定義書でセキュリティ要件（守るべきものとその防御策）が明確か確認する。
  更にOWASP ASVS（The Open Web Application Security Project Application Security Verification Standard アプリケーションセキュリティ検証標準 3.0.1 https://www.jpcert.or.jp/securecoding/materials-owaspasvs.html ）のLevel２ないし３の検証要件がセキュリティ要件として含まれているかを確認する。
（ステークホルダ、規制、技術、ビジネスからのセキュリティ要件が明確であればレビュー可能であり、また実装、テストの結果がセキュリティ要件までトレース可能（証拠証跡）になる）
- ユースケースに対応したテスト可能なアビューズケース(abuse case)が記述されているか？確認し、なければ提案する。
- セキュリティとトレードオフになるユーザビリティやパフォーマンスの要求が明記されていることを確認する。
- 上記を考慮して単体／結合／システムテスト時のセキュリティテストを設計する。

MS 脅威モデルの概要

Microsoft SDL 脅威モデリング: システムに対するセキュリティ脅威を理解し、その脅威からリスクを判定し、適切な軽減策を確立するプロセス

- Microsoft SDLに従って開発するアプリケーションに対する重要な要求事項
- 脅威モデリング
  - DFD＋信頼性の境界を描く
  - 脅威の一覧を作成（ツールが自動生成）
  - 軽減策の立案／作成（ほとんど軽減策も提案されるが抽象度が高い）
  - 確認する
- セキュリティの専門家と非専門家の両方が実施できる

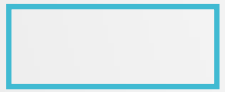
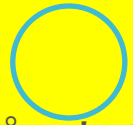
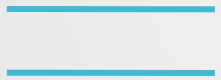
* https://www.microsoft.com/en-us/SDL/adopt/starterkit.aspx のMicrosoft Threat_Modeling_Principles.ppt からの引用

STRIDE 脅威のタイプ

脅威	定義
Spoofing なりすまし	他のものか他人になりすます
Tampering 改ざん	権限無しにコードやデータを変更する
Repudiation 否認	アプリのある機能を実行していないと主張する能力
Information Disclosure 情報漏えい	権限のないユーザーに情報を見せる
Denial of Service サービス妨害	正当なユーザーへのサービスを止めたり、低下する能力
Elevation of Privilege 権限の昇格	ユーザーが権限無しにアプリに対する権限を昇格できる能力

*https://www.microsoft.com/enus/SDL/adopt/starterkit.aspxの
Microsoft Threat_Modeling_Principles.ppt　からの引用

DFD要素のタイプで決まるSTRIDE脅威

要素	S	T	R	I	D	E
 外部エンティティ	✓		✓			
 プロセス	✓	✓	✓	✓	✓	✓
 データストア		✓	✓	✓	✓	
 データフロー		✓		✓	✓	

* https://www.microsoft.com/en-us/SDL/adopt/starterkit.aspx のMicrosoft Threat_Modeling_Principles.ppt からの引用

設計フェーズにおけるQAの役割

- 脅威モデリングに参加する（例えばMicrosoft Thread Modeling Toolを使用する）。
- 脅威モデル（DFD図のようなシステム全体のモジュール連携図が含まれていること）から攻撃経路（attack vector)、特定された脅威及びその防御策（orリスク軽減策）を確認する。
- 再度ASVSから導出したセキュリティ要件が防御策に漏れなく含まれていることを確認する。
- 防御策の機能確認ためにOWASP Testing_Guide v4（ https://www.owasp.org/index.php/OWASP_Testing_Project)などを参考にテストケースを作成する。
- 防御策を破るつもりでアビューズテストケースを作成する。
- 特にデータをパースする箇所には適切なファズテストを計画する（ファズの準備）

実装フェーズにおけるQAの役割

- 静的解析ツールの結果がセキュリティ品質基準を満足していることを確認する
- SEI CERT Coding Standards （C/C++/Java/Perl版がある）(https://www.securecoding.cert.org/confluence/display/seccode/SEI+CERT+Coding+Standards あるいは https://www.jpcert.or.jp/securecoding/index.html)にあるようなプラクティス（セキュリティ品質基準に記載）が実施されていることを確認する
- ファズテスト（IPA発行「ファジング活用の手引き」などを参照）を実施する
- アビューズケースを実施する（スタブやドライバーを用意し早めに実施する）

検査フェーズにおけるQAの役割

- セキュリティ要件をASVSに則って検証する。詳細なテスト方法についてはOWASP Testing Guide https://www.owasp.org/index.php/OWASP_Testing_Project)を随時参照する。
- 機能、セキュリティを除く非機能テスト中に目標のセキュリティ品質を侵害していないことを確認する
- アビューズケースを含むペンテストをエンドツーエンドで実行し、目標のセキュリティ品質を満足することを確認する
- BTS/ITSにおいて、セキュリティバグはセキュリティを担当する開発者とQA担当者のみに読み・書き権限を与える。同様にステークホルダ毎に残存バグ情報や軽減策の表現を調整する

保守フェーズにおけるQAの役割

- この期間中に発見された新しい脆弱性に対する修正確認テストを計画する際は、修正の影響範囲を考慮して追加のテスト、既存のセキュリティテスト、ユーザビリティ及びパフォーマンステストも計画する。
- 機能追加やユーザビリティ／パフォーマンス向上の対策が計画された場合、セキュリティへの影響を明文化し、必要なレグレッションテストを計画する。
- 自製ソフト、購入ソフト、OSS、フレームワーク、OS、ハードウェアなどの変更及びFW,IDS/IPS, WAFの設定変更のセキュリティへの影響を明文化し、必要なレグレッションテストを計画する。

まとめ

- システム／ソフトウェアのセキュリティ品質の作り込みは製造者責任。そのために開発人員のトレーニングと開発工程への（追加）投資が必要（社会的責任であり、ROI云々ではない）。
- 一般開発者やQA担当者もセキュリティへの意識とスキルを習得する必要がある（OWASP ASVSのレベル２ないし３）。
- セキュリティ専門部隊はJVN iPedia(http://jvndb.jvn.jp/)やMITRE CAPECやCWEなどを常にウオッチし、影響する脆弱性への対処策をセキュリティ品質要件に適時加える。
- QA部門は開発部門と協働で上流からセキュリティ品質を作り込むプロセスを確立、維持する。

最後に

スティーブ ジョブスの言葉

顧客はより幸せでよりよい人生を夢見ている。
製品を売ろうとするのではなく、彼らの人生を豊かにするのだ。

Your customers dream of a happier and better life. Don't move products. Enrich lives.

自身がクオリティの判断基準となれ。
中には高い質を求められる環境に慣れていないものもいる。

Be a yardstick of quality. Some people aren't used to an environment where excellence is expected.

ご清聴ありがとうございました。

（ご参考）引用元・教育用資料とツール

1. スライド12の引用元：https://www.cigital.com/resources/videos/on-demand-webinar-threat-modeling/thank-you/?submissionGuid=d56b4342-b1f3-4de8-aad6-fda686b06dcd
2. スライド13の引用元：OWASPアプリケーションセキュリティ検証標準 3.0.1のP.11
3. 安全なWebアプリケーションの作り方　徳丸浩著(SB Creative)
4. 安全なウェブサイトの作り方／安全なSQLの呼び出し方 (IPA)
5. つながる世界のセーフティ&セキュリティ設計入門 (IPA)
6. 『Androidアプリのセキュア設計・セキュアコーディングガイド』と Secure Coding Checker (JSSEC)
7. OWASP Top 10 – 2013 日本語版 (Open Web Application Security Project)
8. Web システム／Web アプリケーションセキュリティ要件書(OWASP)
9. OWASP Top 10 Proactive Controls 2016 Japanese （Webアプリケーション開発者が気を付けるべき10のセキュリティ技術)
10. OWASP Testing Guide 4.0 (2014)
11. “The Security Development Lifecycle” by Michael Howard & Steve Lipner (2006 Microsoft)
12. “Threat Modeling: Designing for Security” by Adam Shostack (2014 Wiley)
13. AppGoat (IPA)