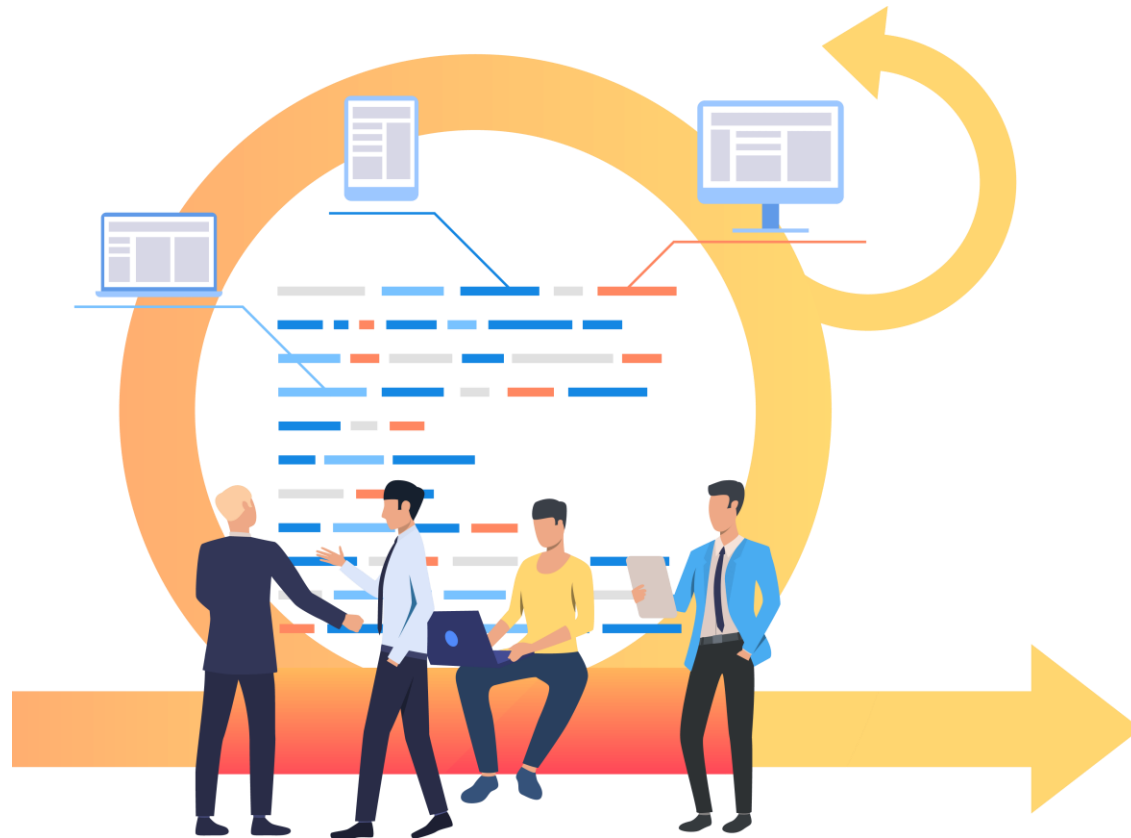


脆弱性対応勉強会Expansion

第04回

アジャイルセキュリティ



目次

- アジャイル開発のセキュリティ対策について、失敗を学ぼう
 - 失敗ケース 1 過剰なセキュリティスプリント
 - 失敗ケース 2 リリース前に脆弱性たくさん
- クラウド型脆弱性診断ツール「AeyeScan」の紹介
 - 全自動脆弱性診断のデモンストレーション
 - GitHubActionsを使ったアジャイル開発での自動診断のコツ

質問等は Twitter で「#vulnstudy」をつけてツイートしてください。
エゴサーチします。

今回の勉強会は、2021-08-12に開催したHardening Projectのイベント、/dev/hardening - Hardening Drivers Conference 2021 で公演した内容を一部アップデートした内容です。

- ・ Hardening Project

Hardening Projectは、システムを衛る技術の価値を最大化する、社会のための技術者による技術者のプロジェクトです。

<http://wasforum.jp/hardening-project/>

- ・ /dev/hardening - Hardening Drivers Conference 2021

<https://hardening.doorkeeper.jp/events/124079>

私はだれ？



関根 鉄平 CISSP

エーアイセキュリティラボ 執行役員

略歴

- セキュリティエンジニアとして、アジャイル開発チームでのセキュリティを推進
- Product Ownerとして、ソフトウェアの開発部門でアジャイル開発を推進。
- 現在は、脆弱性診断ツールAeyeScanの Product Managerや、エンドユーザでの脆弱性診断の自動化や内製化を支援。

その他 社外活動など

- OWASP JAPAN Webシステム/Webアプリケーションセキュリティ要件書
- 情報セキュリティ10大脅威選考会メンバー



失敗の共有から アジャイルセキュリティを学ぼう

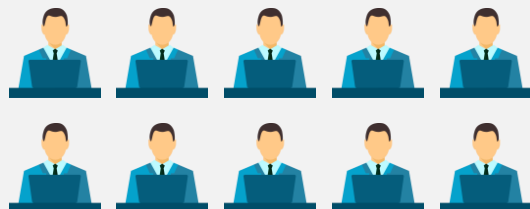
目次

- ❶ 失敗ケース 1 過剰なセキュリティスプリント
- ❷ 失敗ケース 2 リリース前に脆弱性たくさん

ケース1：遅いセキュリティスプリント

背景

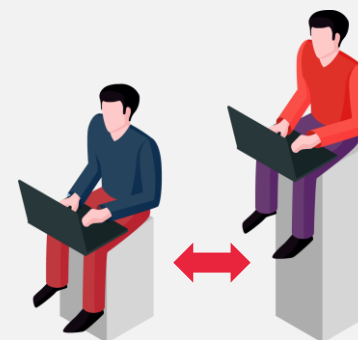
約20名の規模



スクラム開発を
やり始めた



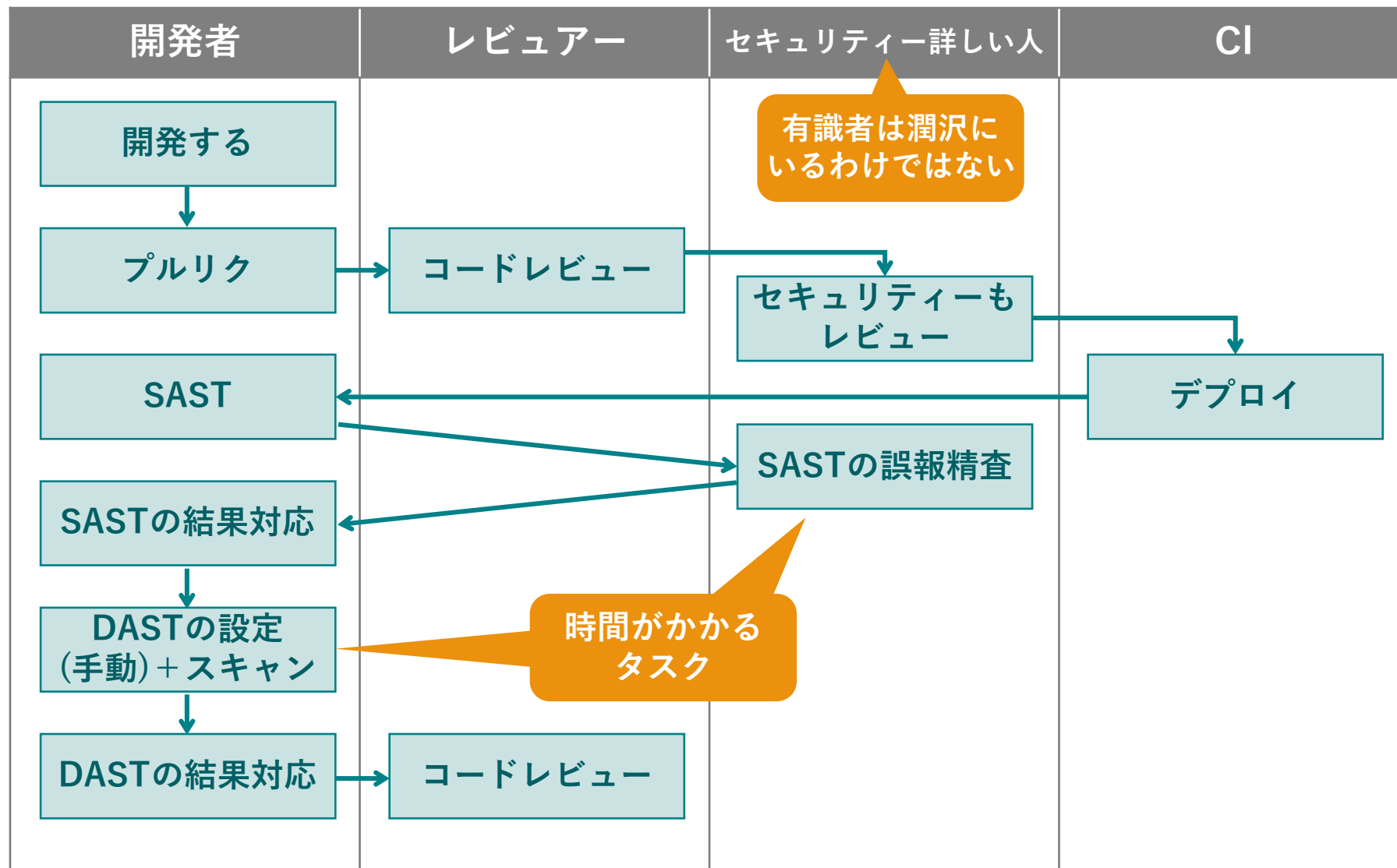
メンバーの技術力に
差が大きい



やったこと

- コードレビューでセキュリティに詳しい人がレビューを実施
- SAST(静的診断ツール)で、各スプリントでスキャンを実施
- DAST(動的診断ツール)で、各スプリントでスキャンを実施

やったことのイメージ



スプリント何もできない。結果とその後

結果

- ☑ チェックリスト難解でコードのレビューが**有識者に集まる**。
開発進まない。有識者＝沢山開発出来る人。
- ☑ **SASTは過剰検知が多い**ので全てレビューしていたら
追いつかない。
- ☑ DASTを**毎スプリントで実施**、
とても追いつかない
- ☑ バックログが**全然さばけない**

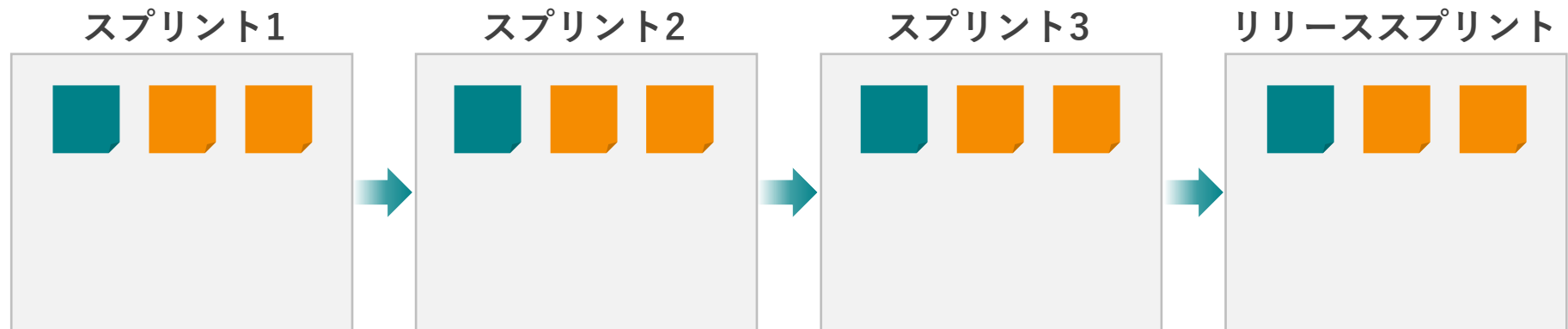


今回のケース

■ 機能のタスク ■ セキュリティのタスク

プロダクトオーナー

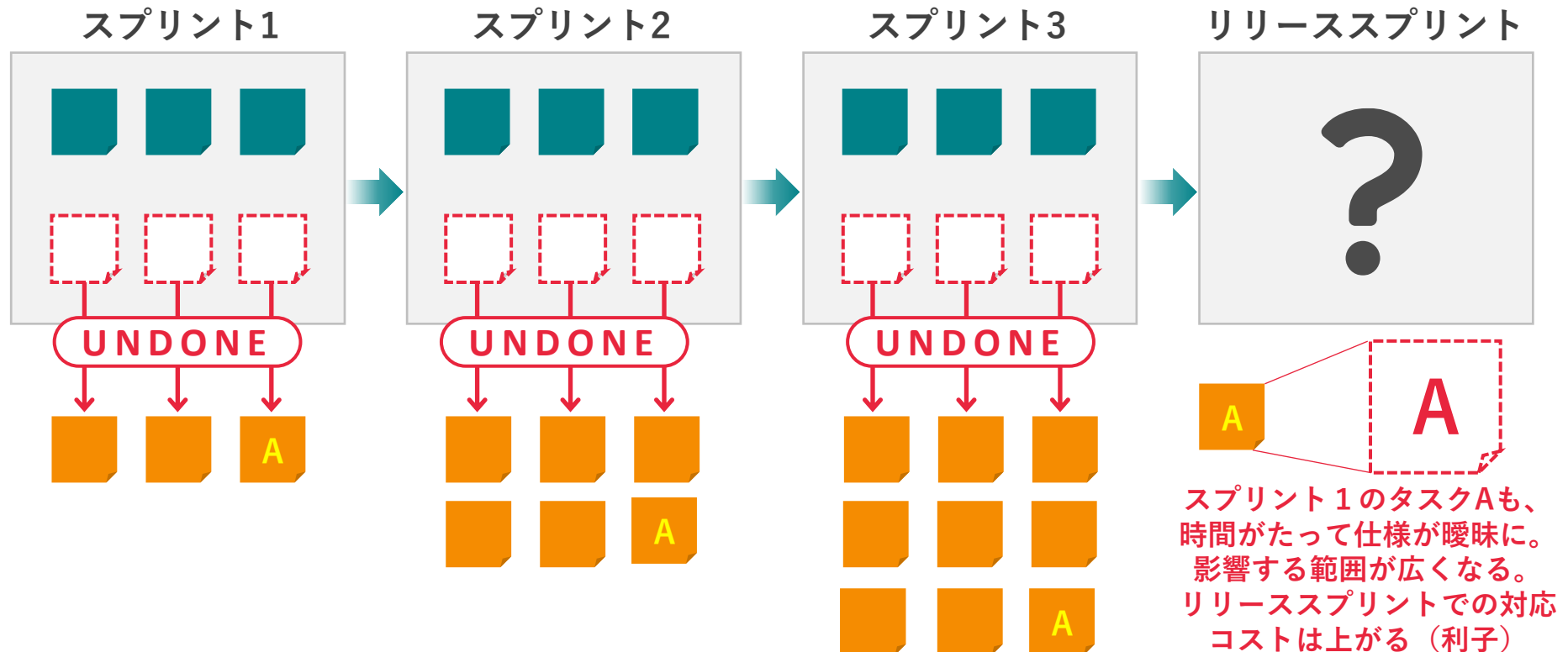
全然リリース
少ない。。



よくあるケース(技術的負債)

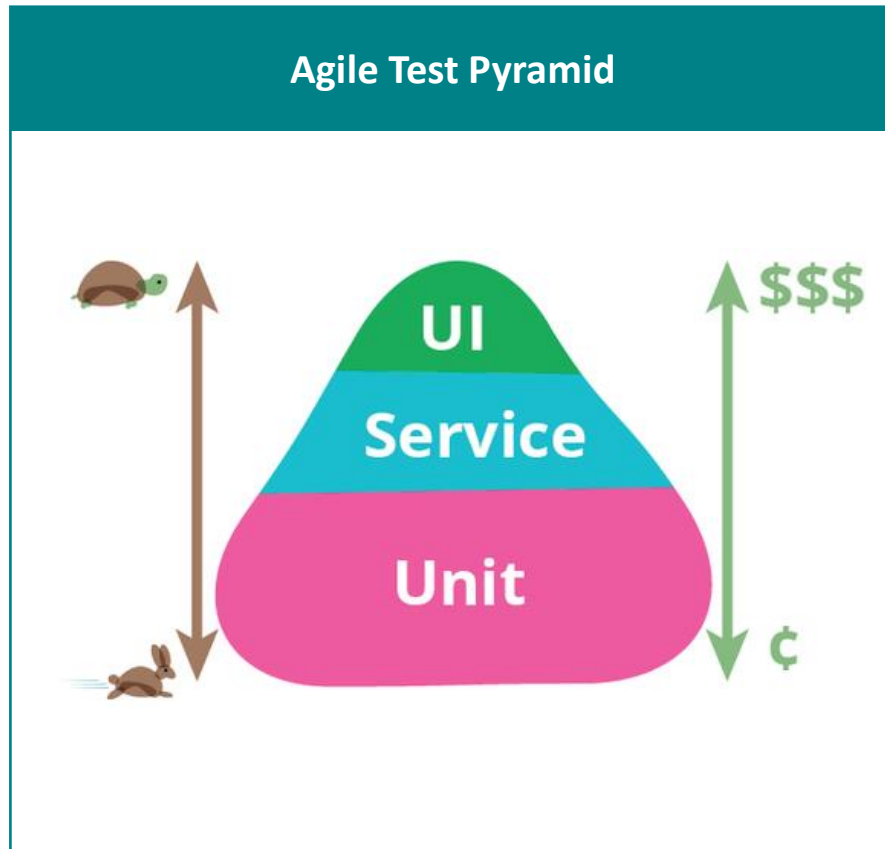
■ 機能のタスク ■ セキュリティのタスク

タスクボード



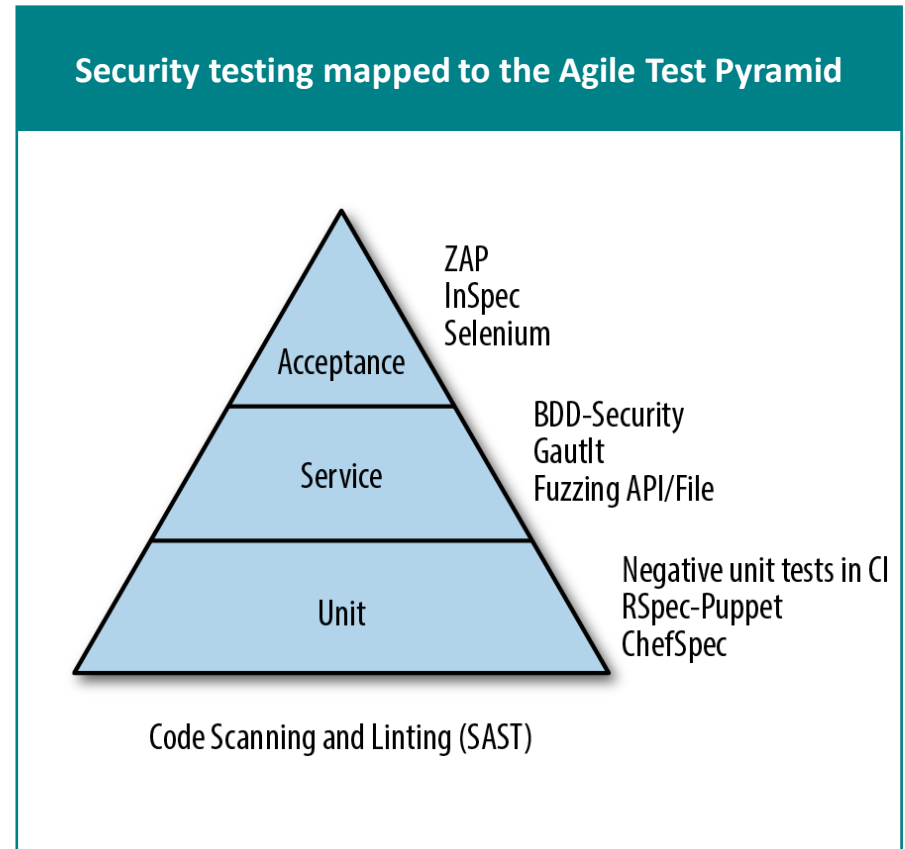
アジャイル開発のセキュリティテスト

アジャイル開発のテストピラミッドを、
セキュリティテストにあてはめる。
UIに近いテストほど高コスト！



出展：Martin Fowler

<https://martinfowler.com/bliki/TestPyramid.html>



出展：O'Reilly Media, Inc. 「Agile Application Security」

Figure 11-1. Security testing mapped to the Agile Test Pyramid

その後

OWASP ASVSを使った
セキュリティの
チェック担当者で実施。

▶ レビューアーの負荷 ↓ 開発速度 ↑

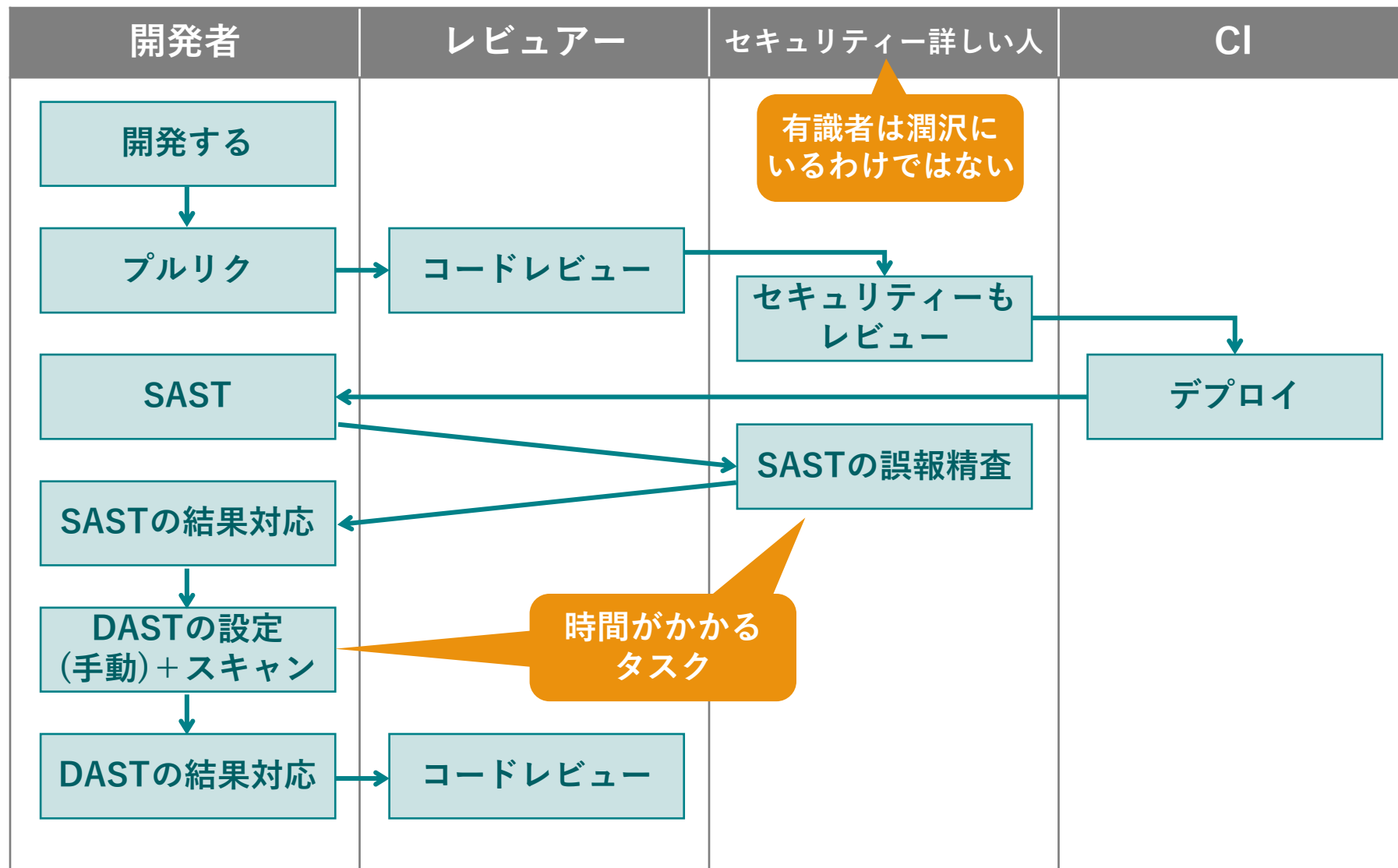
SASTのタイミングは
プルリクエスト時。
誤検知のレビューはしない。
SASTのFB(検知)は全て直す。
誤検知も直す (ツールに合わせる)

▶ レビューアーの負荷 ↓ 開発速度 ↑

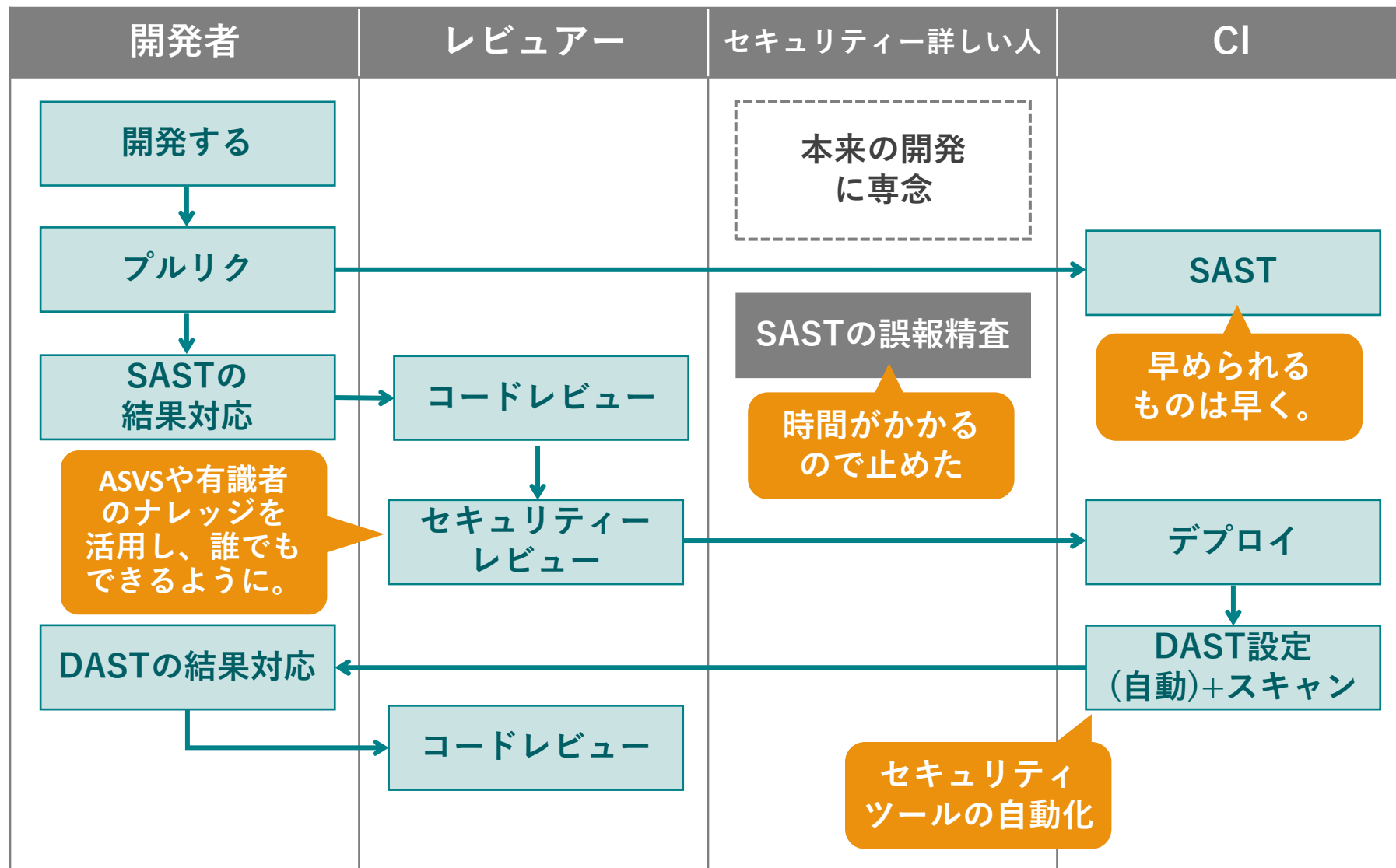
DASTはCIに組み込んで
全自動化

▶ 開発者の負荷 ↓ 開発速度 ↑

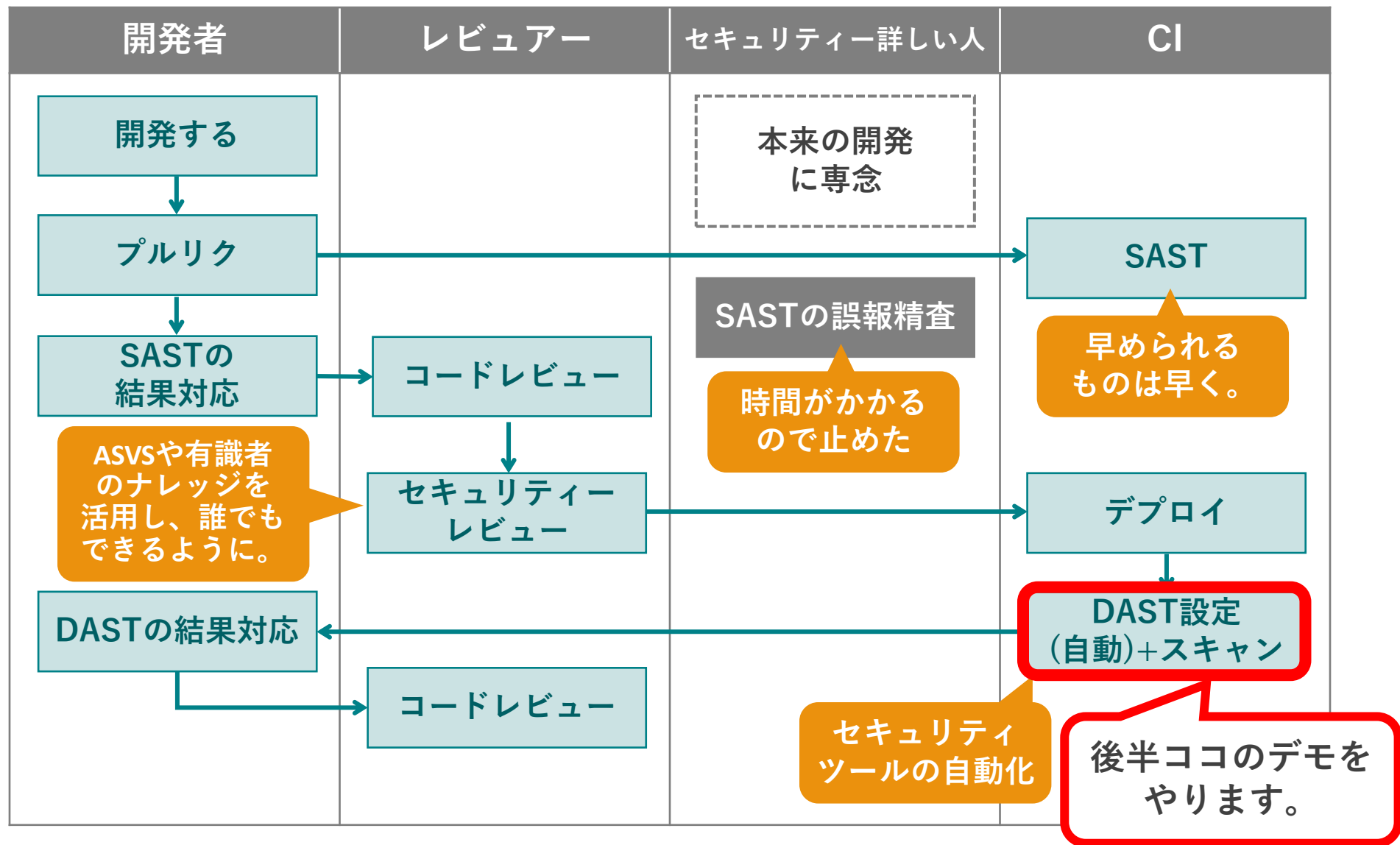
やったことのイメージ



やったこと



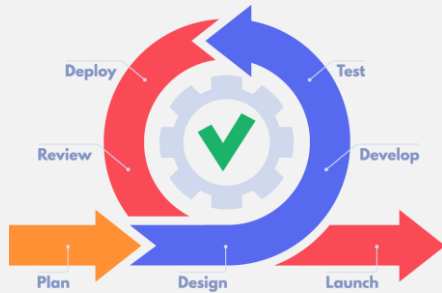
やったこと



失敗ケース 2 リリース前に脆弱性たくさん

背景

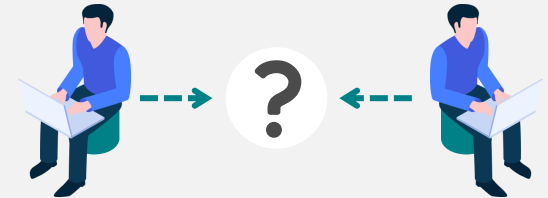
アジャイル開発に
なれている。



セキュリティ専門のメンバーは
いないが、スキルレベルの
高いメンバで構成。



個人事業主や複数の会社の
チーム構成でコミュニケーションは
取りにくい状況であった。



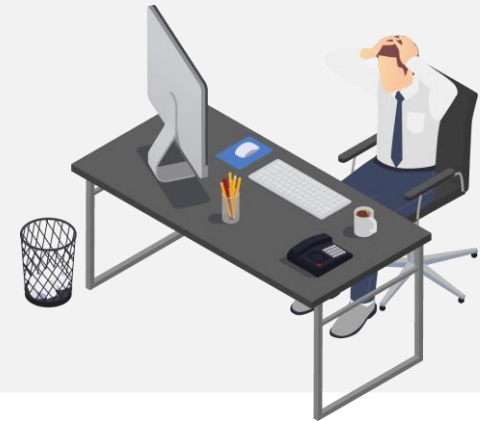
やったこと

- プルリク時のソースコードチェックや、テスト時のテストパターンにセキュリティ要素を入れた。
- 脆弱性診断をリリーススプリントで外部に依頼

失敗ケース 2 リリース前に脆弱性たくさん

結果

- ☑ XSSやSQLInjectionのような問題は、ほとんどなかった。
- ☑ 設計書が無い？ので、設計に関するセキュリティレビューがなされなかった。
- ☑ 認証認可系で、致命的な脆弱性多数。
- ☑ リリーススプリントで判明したため対応コストが大きくなった。



その後

- ASVSを使ったタスクの設計
- セキュリティメンバがチームに加わりアタッカーストーリー作成
- 探索的テスト (Exploratory Testing) をセキュリティテストで実施。

アタッカーストーリー作成

ユーザーストーリー

ユーザーストーリーとは、以下の簡易的なフォーマットで記述されるセンテンスです。
As a [role], I want [goal/desire] so that [benefit].
(出展 : User story - Wikipedia)
https://en.wikipedia.org/wiki/User_story

例)
買い物客である私は、
商品を買いたい。
これによって商品を購入できる。

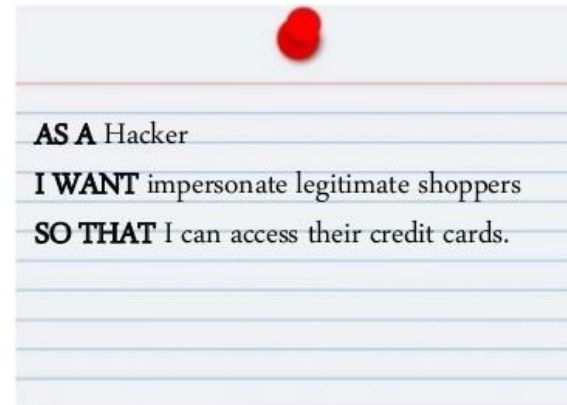


projectconevent.com

アタッカーストーリー

ユーザーストーリーに対して、攻撃者視点でのストーリー（アタッカーストーリー、セキュリティストーリー）を考える。

例)
ハッカーである私は、
ユーザになりすましたい。
これによってクレジットカード情報を盗む。

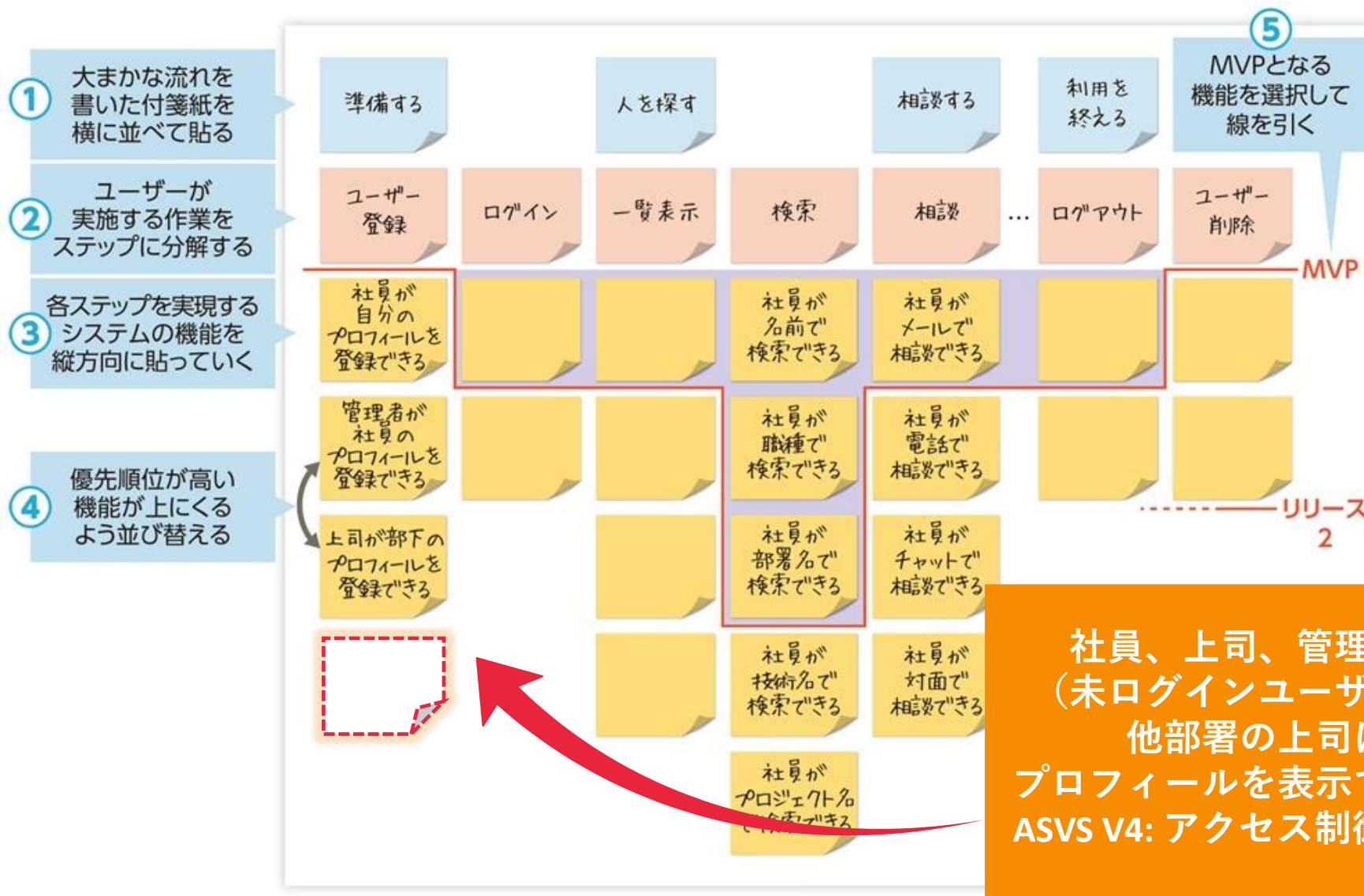


projectconevent.com

出典 : Abuser Stories: Thinking Like the Bad Guy to Reduce Software Vulnerabilities

<https://www.slideshare.net/projectcon/abuser-stories-thinking-like-the-bad-guy-to-reduce-software-vulnerabilities>

ユーザストーリーマッピングにセキュリティタスクを加えてみる。



出展：日経BP デザイン思考を生かしてシステムを素早く開発する方法

<https://xtech.nikkei.com/atcl/nxt/mag/sys/18/111900047/111900003/>

探索的セキュリティテスト（Exploratory Security Testing）のススメ

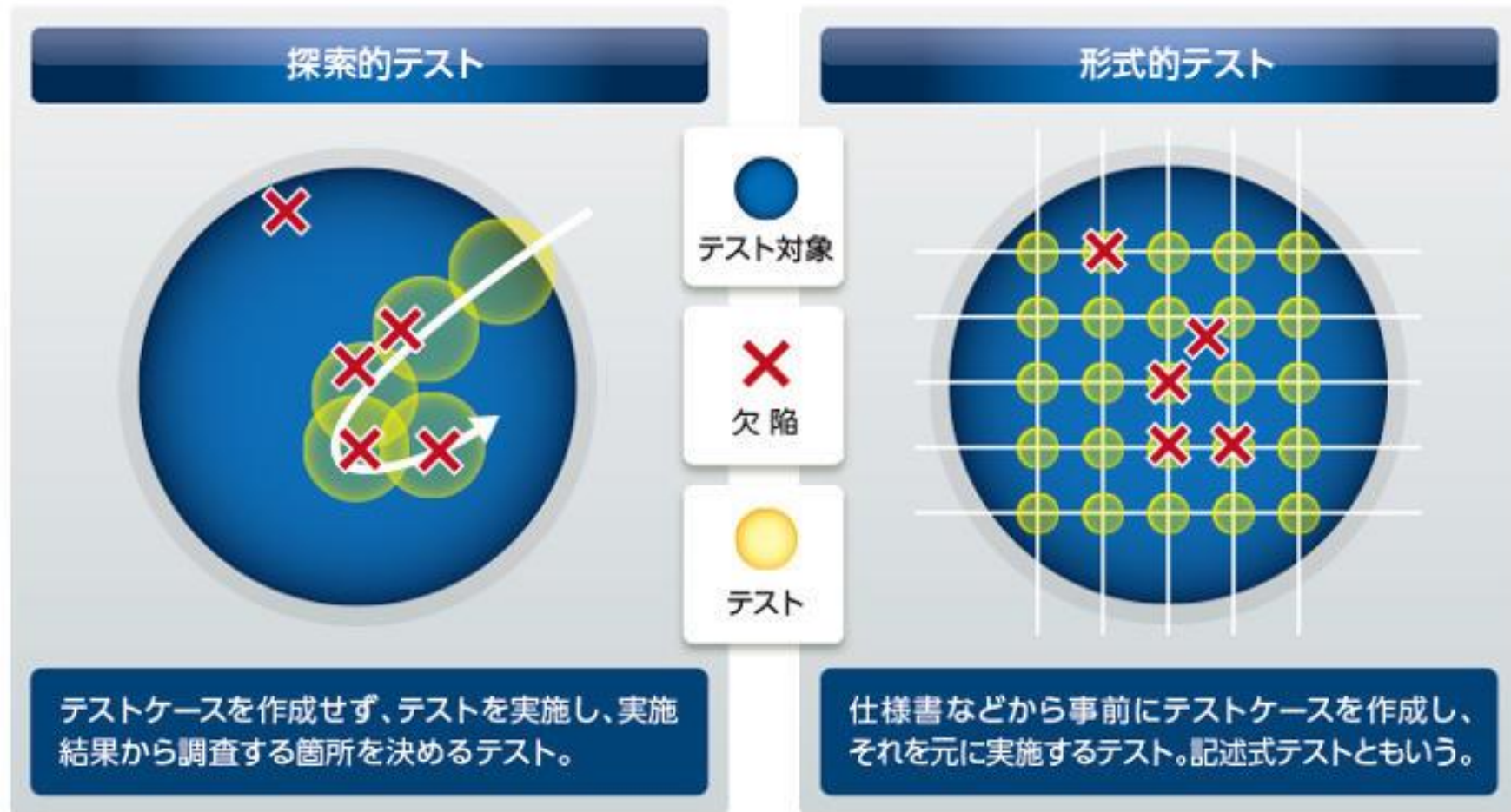
- ☑ テスト手法の一つ。
- ☑ テスト設計と実施を同時に行う。
- ☑ 短期間でテストが実行可能なので、アジャイル開発にマッチする。



テスト例

アタッカーストーリーを元に、テスト方針（テストチャーター）を作成。
何をテストするか、一般ユーザのクレジットカード番号を扱う箇所。
どうやってテストするか、パラメータの改ざん、セッション管理の強度を行う。
どんなセキュリティリスク、クレジットカードの漏洩しないか？

形式的テストと探索的テストの違い



出展:バルテス株式会社Qbookより引用
https://www.qbook.jp/column/20210323_1126.html

-最後に- セキュリティ屋と開発屋は仲良くしよう。

セキュリティ屋

- チェックリスト魔
- 最後にNOと言うエンジニア



開発屋

- ベロシティとか、アジャイルは文化とか難しい単語を喋りだす
- 会社でトランプをやりだす

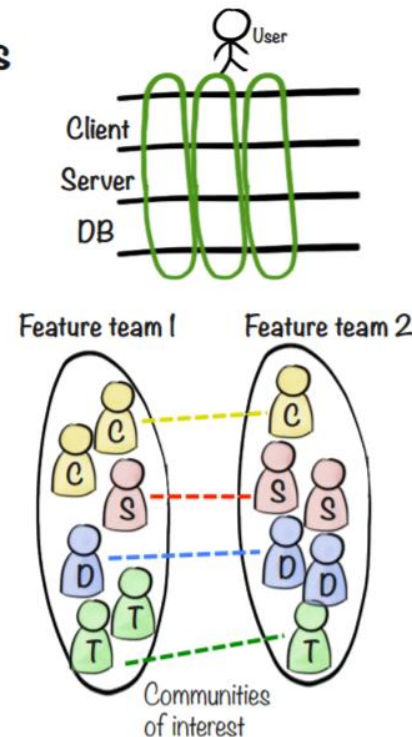
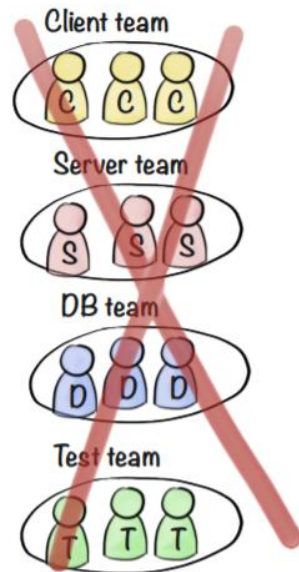


アジャイルでは異なる機能を持ったメンバーが一つのチームで開発する。

セキュリティ屋

- チェックリストではなく、セキュリティと一緒に考えて、具体的な実行可能なタスクを考えてくれる。
- セキュリティーテストを自動化してくれるエンジニア

Cross-functional teams
are vertical

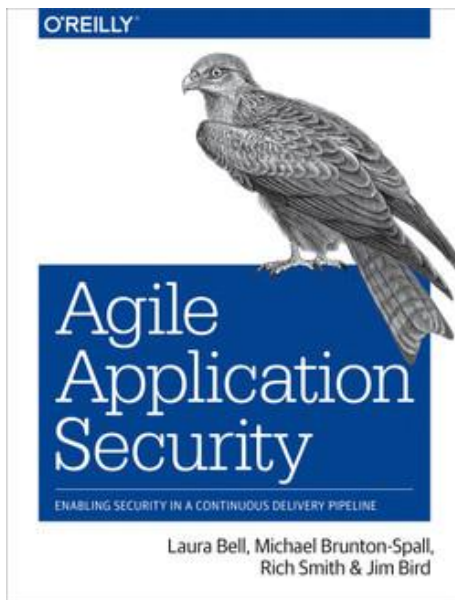


開発屋

- 誰にでもわかる言葉で、セキュリティ屋もチームに入れて。
- セキュリティを丸投げしない。

出展:crisp.se What-is-Agile.P68 より引用
<https://blog.crisp.se/wp-content/uploads/2013/08/20130820-What-is-Agile.pdf>

質問等は Twitter で「#vulnstudy」をつけてツイートしてください。
エゴサーチします。



参考文献：O'Reilly Agile Application Security

脆弱性診断の内製化を 成功に導く AeyeScan

～いつでもだれにでも診断できる環境をご提供～

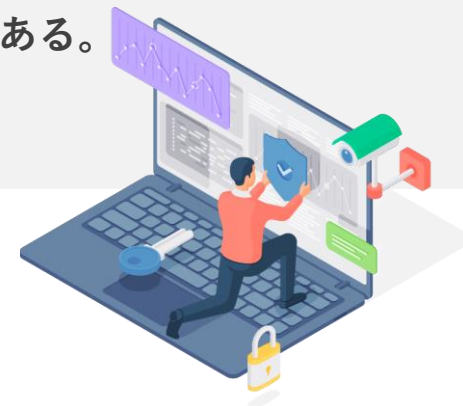


脆弱性診断の内製化のメリット

開発・CSIRT・リスクマネジメントの現場において、
このような課題・要望がありませんか？



- セキュリティコードレビュー、テストに時間がかかり生産性が低下している。
- サイトの公開、リリースの事業状況にあわせて柔軟に脆弱性診断したい。
- コストが理由で脆弱性診断の実施を断念しているサイトがある。
- 脆弱性診断費用を抑えたい。



● ● ●
内製化で解決します。

内製化のパターン

脆弱性診断の内製化には大きく 2 つのパターンがあります。

セキュリティエンジニア

高度なセキュリティ技術をもったエンジニアによる診断が可能な状態

担当者 診断の担当者を選任または、セキュリティベンダからの中途採用

ツール 様々な設定や制御が可能なツール。使いこなすには、ベンダートレーニングの受講が前提。

プロセス 関係者からの診断の依頼をもとに実施。

目指すゴール セキュリティの専門知識をもったメンバーが、システムのセキュリティを評価するチーム



プロ向けツールで実現する内製化

セキュリティエンジニアではない人

だれでもいつでも診断が可能な状態

担当者 立場や利用頻度を問わず関係者全員

ツール 簡単な設定で自動化されたツール。教育不要

プロセス システムのライフサイクルに自動的に組み込む。

目指すゴール システムのライフサイクルに携わる全員が、セキュリティの役割を担う



だれでも使えるツールで実現する内製化



内製化の課題は？

AeyeScanとは

SaaS型のWebアプリケーション脆弱性診断プラットフォームです。

診断内製化に必要な、簡単・高精度な脆弱性診断を、
AI + RPA (Robotic Process Automation)を活用し実現させます。



お手軽に診断を開始



高精度・レポートが
分かりやすい



SaaS
プラットフォーム共有

脆弱性診断デモンストレーション GitHub Actionsを使った全自動診断

質問等は Twitter で「#vulnstudy」をつけてツイートしてください。
エゴサーチします。