Many attacks happen on the application layer where infrastructure security solutions are not effective.
Application Security is important, however, still rare in application development. Let's change that !

Three useful notions for proactive Application Security !

「Shift Left」

「Security by Design」

「Continuous Hardening」

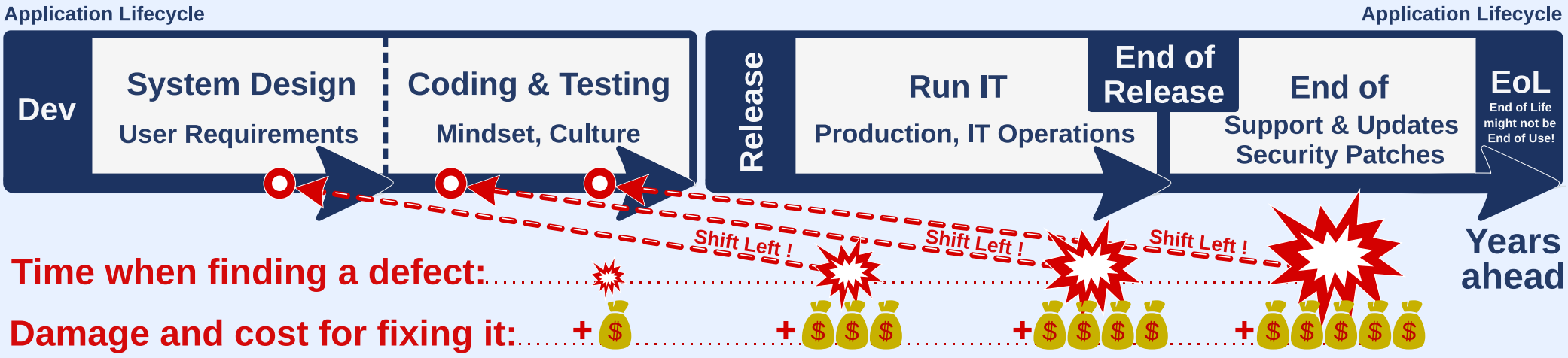
A security story so that 「bottom-up」 meets 「top-down」

Shift Left
Mindset for increasing efficiency and controlling cost during the whole application lifecycle.

「Shift Left !」

Top-Down Support
Efficient Application Security is not feasible without strong support from top management!

Find and fix defects early for reducing cost and risk
Handle it like Product Quality or Safety !



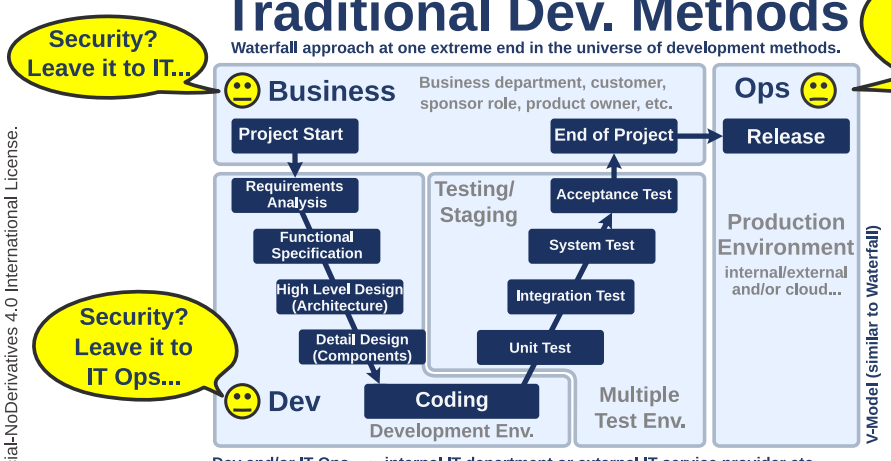
Rule of Thumb
The later it gets, the more expensive it is to change an application (getting significantly costlier!)

Top Management Involvement Required
Keep an eye on the realities in your organization regarding, 「Shift Left」, 「Security by Design」 and 「Continuous Hardening」 !

Years ..... Months ..... ← Delivery Time — The Universe of Development Methods — Delivery Time → ..... Weeks ..... Days ..... Hours

Traditional Dev. Methods

Waterfall approach at one extreme end in the universe of development methods.



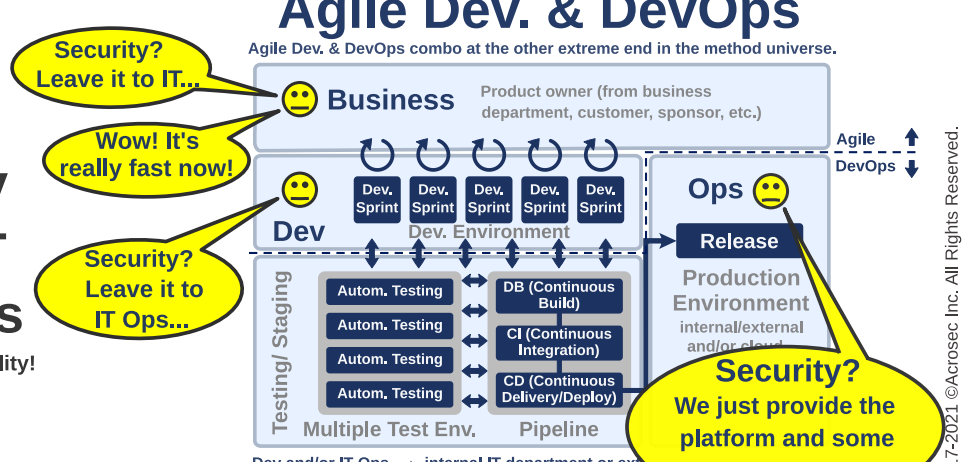
Security? We just provide the platform and some security basics...

Tragedy in the IT Trenches

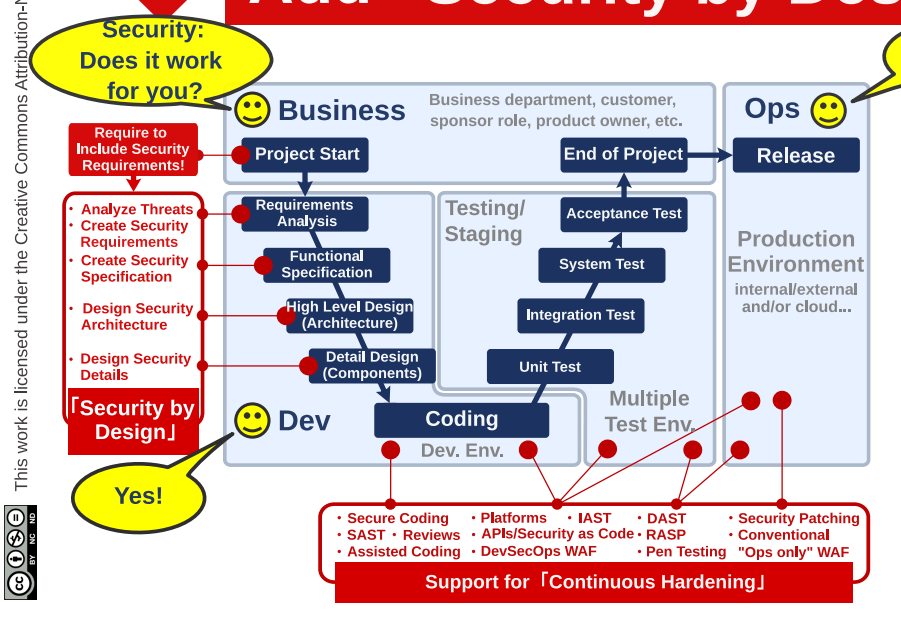
Still an often seen reality!

Agile Dev. & DevOps

Agile Dev. & DevOps combo at the other extreme end in the method universe.

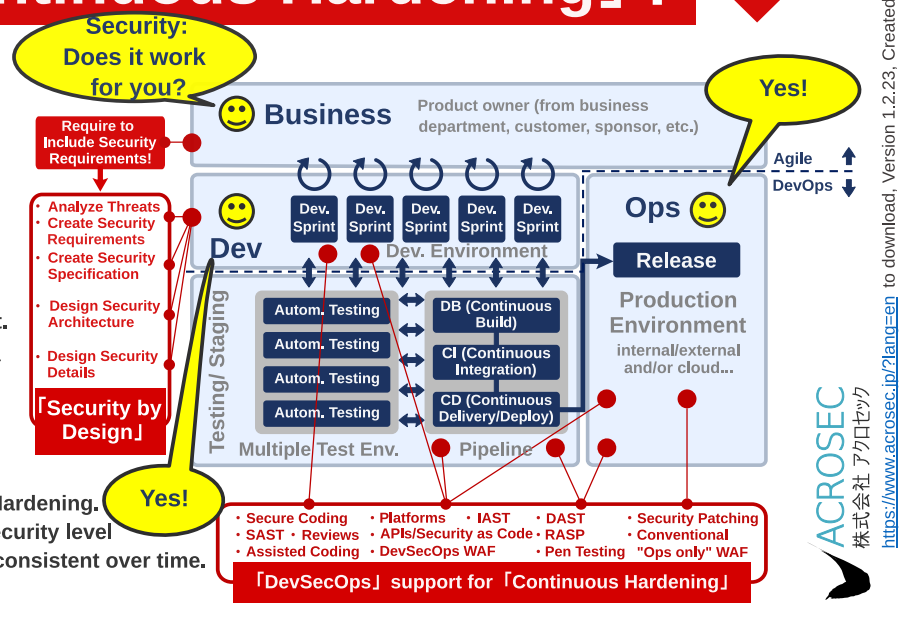


Add 「Security by Design」 & 「Continuous Hardening」 !



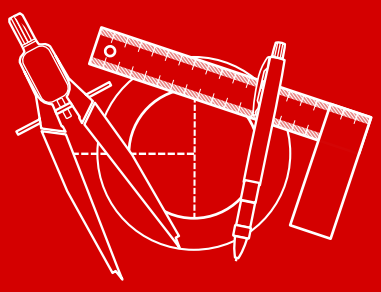
Better Approach

1. Business requires a proactive approach and provides budget.
2. Development project creates a security design. Suggested starting point: Zero Trust.
3. Development and other teams work together for efficiently
   a) implementing Continuous Hardening.
   b) maintaining the intended security level afterwards in order to stay consistent over time.

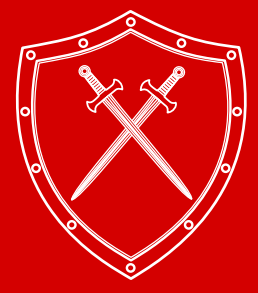


数多くの攻撃はインフラストラクチャーセキュリティが有効ではないアプリケーションレイヤーで行われる。しかし
アプリケーション開発の段階で「Security by Design」とハードニングは　まだ一般的ではない。それを変えよう！

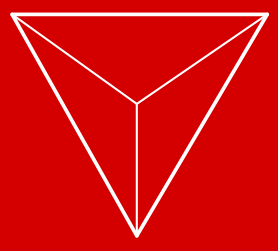
アプリケーションセキュリティ維新の三傑



「Shift Left」



「Security by Design」



「Continuous Hardening」

継続的なハードニング

「トップダウン」と「ボトムアップ」を結ぶセキュリティストーリー

経営トップ　　ITの現場

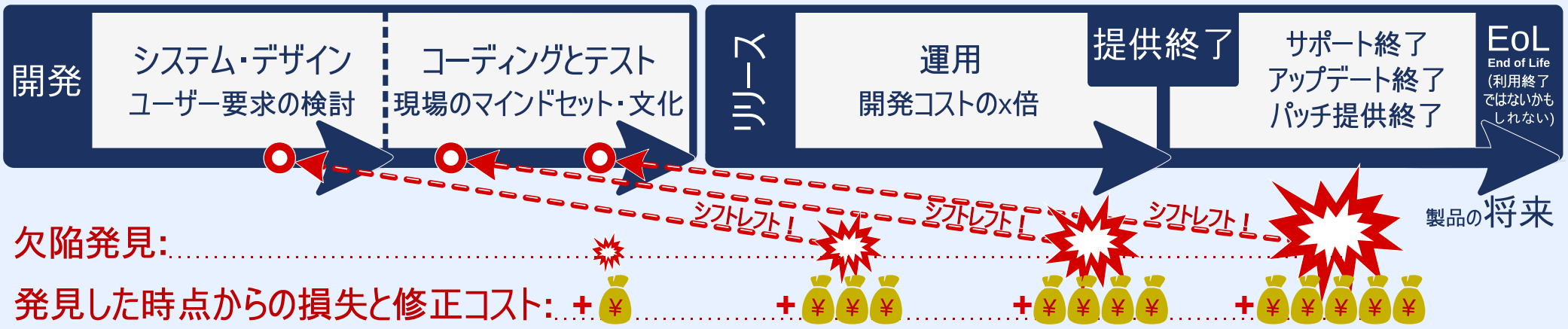
シフトレフト
アプリケーション・ライフサイクル全体においてテストの強化で運用効率をアップさせる発想(リスクとコストのコントロール)

「シフトレフト（Shift Left）！」

欠陥の早期発見と修正は　コストとリスクを抑える作戦
品質改善や安全管理のように！

トップダウンからのサポートが必須
経営トップの強力なサポートがなければ、効率的なアプリケーションセキュリティは不可能！

アプリケーション・ライフサイクルの全体　　アプリケーション・ライフサイクルの全体



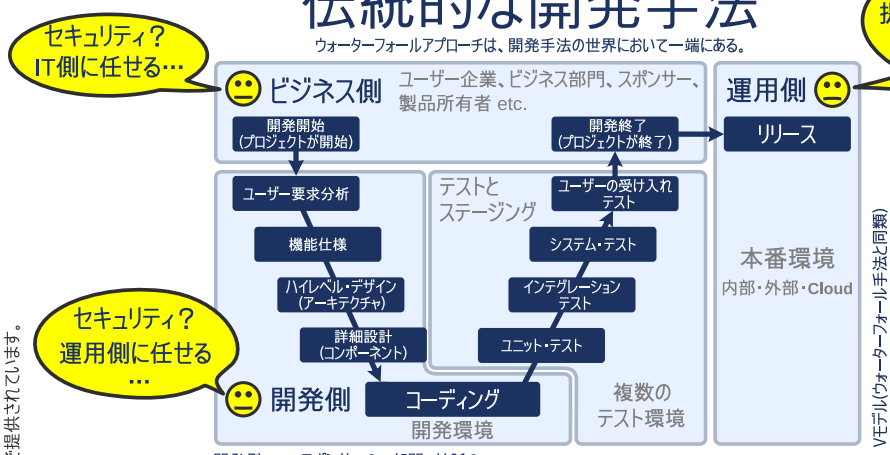
リスクとコスト
時間の経過によりセキュリティリスクと欠陥の修正コストが高くなる

経営者サイドの役割
組織内の経営者は「Shift Left」、「Security by Design」と「継続的なハードニング」に関する現場の状況を理解する必要がある！

数年間 ..... 数ヶ月間 ..... ← リリースまでの時間　　開発手法の世界　　リリースまでの時間 → ..... 週間 ..... 数日間 ..... 数時間

伝統的な開発手法

ウォーターフォールアプローチは、開発手法の世界において一端にある。

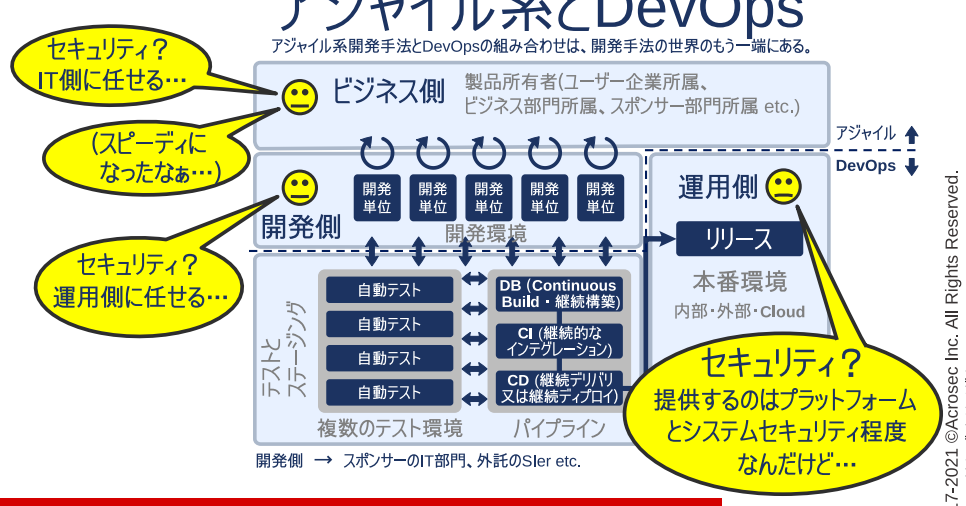


IT現場の悲劇

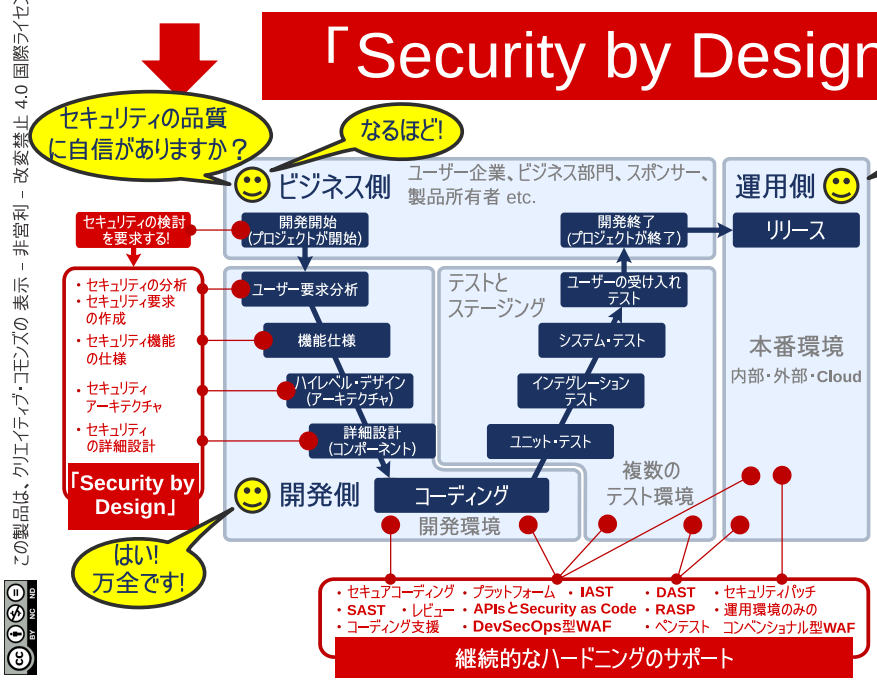
よく見られる現場の状況

アジャイル系とDevOps

アジャイル系開発手法とDevOpsの組み合わせは、開発手法の世界のもう一端にある。



「Security by Design」＆「継続的なハードニング」の導入！



理想的なアプローチ

1. ビジネス側は明確にセキュリティを要求し、適切な予算を提示する。
2. 開発側は適切なセキュリティ設計を行い、製品を開発する。
   適切な出発点: Zero Trust アプローチ
3. 運用側と開発側はともに継続的に:
   a) セキュリティ・ハードニングを実効する。
   b) セキュリティ品質維持のために協力する。