DevSecOps

長谷川 颯
Hayate HASEGAWA

1　はじめに

近年，セキュリティ関連の重大事故が多く報じられている．企業や組織にとって，セキュリティに対するリスクマネジメントは重要な課題の一つである．特に，個人情報や決済情報などの重要情報を取り扱う場合には，それらを保護することは企業や組織にとっての社会的責務である．

また，ソフトウェア開発において，開発するシステムの規模が大きくなるほど，ソフトウェアの開発効率は低下すると言われている．そのため，ソフトウェア開発では，開発工程を効率化し生産性を向上することが重要である．そこで，近年様々なソフトウェア開発手法が出現している．

2　DevOps

2.1　概要

DevOps とは，ソフトウェア開発手法の一つであり，「開発 (Development)」と「運用 (Operation)」を組み合わせた言葉である．DevOps の開発サイクルを Fig. 1 に示す．開発部門と運用部門が一体となり，これまでよりも迅速かつ頻繁にアプリケーションを提供するという概念である．

開発部門は多くの新しい機能やサービスを開発し，迅速に提供することを求めるが，運用部門は安定性・信頼性を求める．DevOps では，開発部門と運用部門が密に連携することで，小規模な開発とリリースを繰り返す．これにより，システムの大幅な変更によるリスクを下げ，安定性・信頼性を保ちながら，ユーザーに新しいサービスを迅速に提供できる．

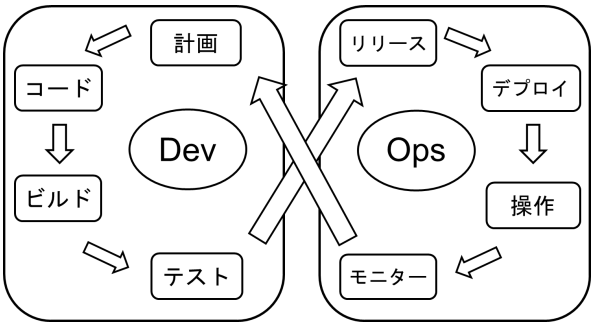

Fig.1　DevOps サイクル

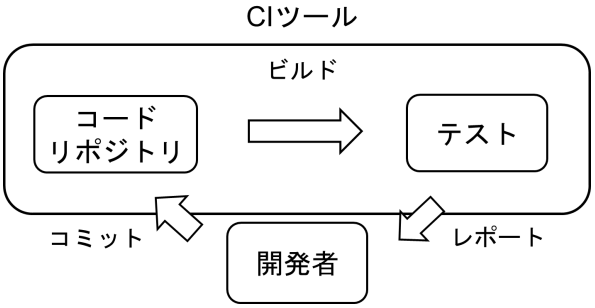

Fig.2　CI サイクル

2.2　CI（継続的インテグレーション）

DevOps という概念は，この CI によって成り立っている．CI とは，ソフトウェア開発手法の一つである．CI とは，ソフトウェア開発において「定期的・自動的」にビルドやテストを行うことで開発の効率化や省力化，納期の短周期化を図ることである．

一般に，ソフトウェア開発において「品質」,「コスト」,「納期」の三つを同時に満足することは困難とされている．その三つを満足する方法の一つが CI である．CI の開発サイクルを Fig. 2 に示す．CI では，支援ツールの導入により開発サイクルが向上するため，問題を早期に発見し，チーム開発のスピードを上げ，コードの品質を保証できる．

2.3　DevOps のメリット

DevOps のメリットは以下の三つである．

- リリースサイクルの高速化
  DevOps により，開発部門・運用部門双方が複雑な変更を行う必要がなくなる．そのため，開発のプロセス全体が高速化し，素早くサービスをリリースできる．リリースまでの時間が短くなることで，市場の需要やトレンドに合ったサービスを開発できる．
- 生産性の向上
  DevOps により，ユーザーからのフィードバックを得やすくなるため，バグや仕様のミスを早期に発見できる．修正にかかる時間や不要な開発工数が削減でき，開発効率が向上する．
- リスクの分散
  DevOps は一サイクルごとの開発規模が小さいため，変更箇所が少なく，サービスの変更によるリスクを分散できる．リスクは小さく分割されているほど対処しやすく，開発者はバグや不具合を見つけやすくなる．したがって，高い品質を維持できるため，ソフトウェア開発の信頼性が向上する．

2.4　導入事例

実際に DevOps を導入した会社に，ソフトバンク株式会社がある．ソフトバンクは自社の開発スピードを向上するため DevOps を導入した．グローバル標準の手法・技術の導入と生産効率向上の両面から，従来のウォーターフォー

ル手法では開発時間の短縮に限度があったためである．

DevOps を導入した結果，ウォーターフォール手法では 2 週間に 1 回だったデプロイ回数が，118 回と大幅に増加した．また，開発作業料やワークフロー，進捗状況を可視化できるようになり，チーム全員が確認できるようになった．自身のタスク以外の作業を手伝うなど，個人だけでなくチーム全体の効率化を考えられるようになった [1]．

3 DevSecOps

3.1 概要

DevSecOps とは，DevOps にセキュリティの要素も取り込れた概念である．DevOps 自体にセキュリティの視点が全く含まれていないわけではない．しかし，言葉の意味を直接的に捉えると，開発視点に比重が置かれ，セキュリティという重要な視点が抜けやすくなるために Sec の文字が加わった．DevSecOps では，DevOps とは異なりセキュリティテストをソフトウェア開発工程全体に組み込む．DevSecOps の開発サイクルを Fig. 3 に示す．

アプリケーションを頻繁にリリースできる体制・環境を作ることができても，セキュリティ対策が不十分であればリリースできない．そこで，十分なセキュリティを担保しつつ，DevOps の目的としていた頻繁なリリースを達成するのが，DevSecOps の概念である [2]．

3.2 シフトレフト

DevSecOps を実現するにあたり，最も重要な点は「シフトレフト」対応である．シフトレフトとは，企画や設計などのソフトウェア開発の初期段階から，セキュリティや品質の検討を行うという概念である．

一般的にソフトウェア開発の工程では，初期の段階で何らかの欠陥がほぼ必ず存在すると言われている [2]．また，後の工程で対処すると，対応コストが数百倍に増幅する．シフトレフトでは従来のように，開発が進んだ後にセキュリティや品質を後付けで考えるのではなく，より早い段階でセキュリティや品質の検討を行う．これにより，テストやリリース，運用後の段階になりバグが見つかるリスクと，その対応コストを軽減できる．

3.3 DevSecOps の重要性

DevOps では，その開発サイクルの速さから，十分なセキュリティ対策を講じることが難しい．既存のセキュリティテストツールやコンプライアンス監視ツールは，DevOps で求められるような速さでコードをテストすることは想定していない．そのため，中には変化のペースについていけないものも多い．また，セキュリティ対策が十分でなければ，リリースしたアプリケーションがサイバー攻撃を受け，インシデントに繋がる可能性もある．

これまでシステム開発は，サービスに必要な機能を満たし，いち早くリリースすることに力が注がれてきた．しかし，相次ぐ情報漏えい被害などを背景に，サイバー攻撃に対応するため，セキュリティ向上の声が高まっている．また，現代は市場環境の変化が速いことにより，システム開発にはビジネスニーズへの迅速な対応が求められている．

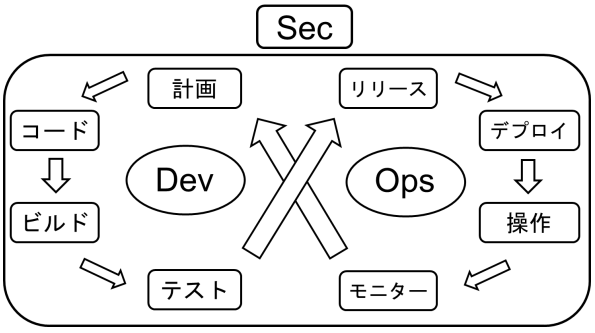

Fig.3 DevSecOps サイクル

これを受けて，システムの市場投入までの時間短縮や，迅速なサービス変更といった点で，他社との差別化を図ることが企業には求められている．したがって，「より迅速に」と「より安全に」を両立できる DevSecOps の重要性は高まっている．

3.4 DevSecOps の実現例

DevSecOps は Fig. 1 のサイクルの各段階においてセキュリティの向上を目指す．具体的には，各機能や設定をコード化することで，開発から運用までの一連のサイクルを自動化する．一連の処理のパイプライン化により，ソースコードの更新が発生すると必ずセキュリティテストが行われ，テスト漏れが発生しない．また，ソフトウェア開発全体でもセキュリティ監査を専門とする部門を設置し，プロジェクト全体としてセキュリティ向上を目指す．

4 今後の展望

DevSecOps への取り込みは黎明期にある．ソフトウェアのデプロイに既存のセキュリティテストツールやコンプライアンス監視ツールが組み込まれていることは少ない．さらに，セキュリティ対策に完璧なものはなく，DevSecOps の具体的な例が少ないのが現状である．

一つのサービスを構成するアプリケーションはより巨大化・細分化し，複雑化しているため，セキュリティの重要性はより一層高まっている．そのため，セキュリティテストツールや自動化ツールなどの環境整備がより一層進むと考えられる．また，市場のニーズをいち早く反映するためには，開発速度を向上させることも重要である．そのため，DevSecOps は今後多くの開発現場での採用が見込まれる．

参考文献

1) ソフトバンク，DevOps 導入でレッドハットのコンサルティングを採用 [事例]—IT リサーチ，https://news.mynavi.jp/itsearch/article/devsoft/3179/，参照 Apr.24， 2019
2) はじめての DevSecOps—伊藤忠テクノソリューションズ株式会社，https://www.business-on-it.com/2004-whats-devsecops/， 参照 Apr.24， 2019
3) 技術力を駆使してよりよい社会を実現—DevOpsHub，https://licensecounter.jp/devops-hub/blog/realglobe/，参照 Apr.24， 2019