

MOP-010S

運用管理者のための DevOps

Release に目覚めよう！

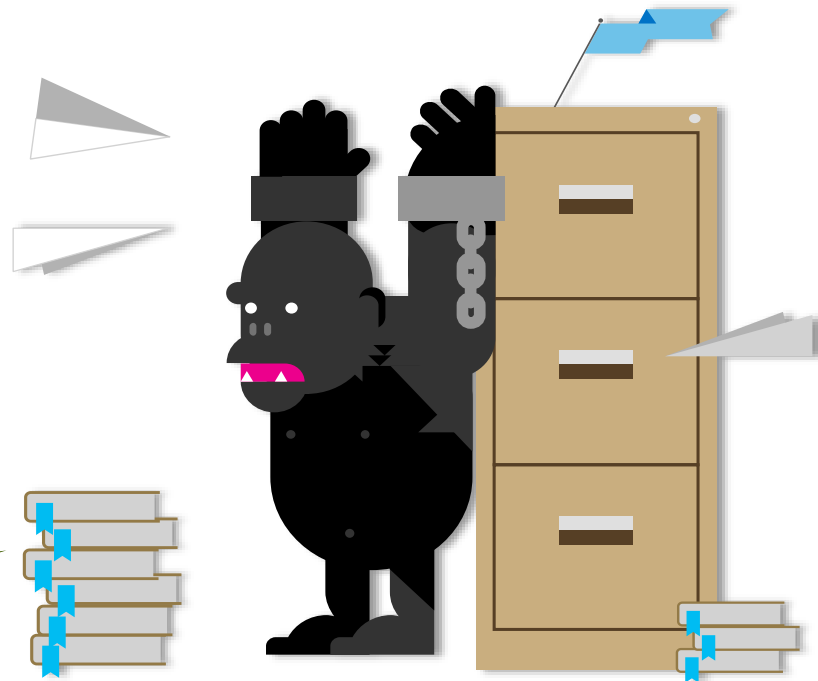
日本マイクロソフト株式会社
エバンジェリスト
高添 修

アジェンダ

DevOps はビジネスの話である・・・が、
そんな話に聞き飽きたあなたのためのセッション

- DevOps は難しい
- 現場の変化と DevOps
- まとめ

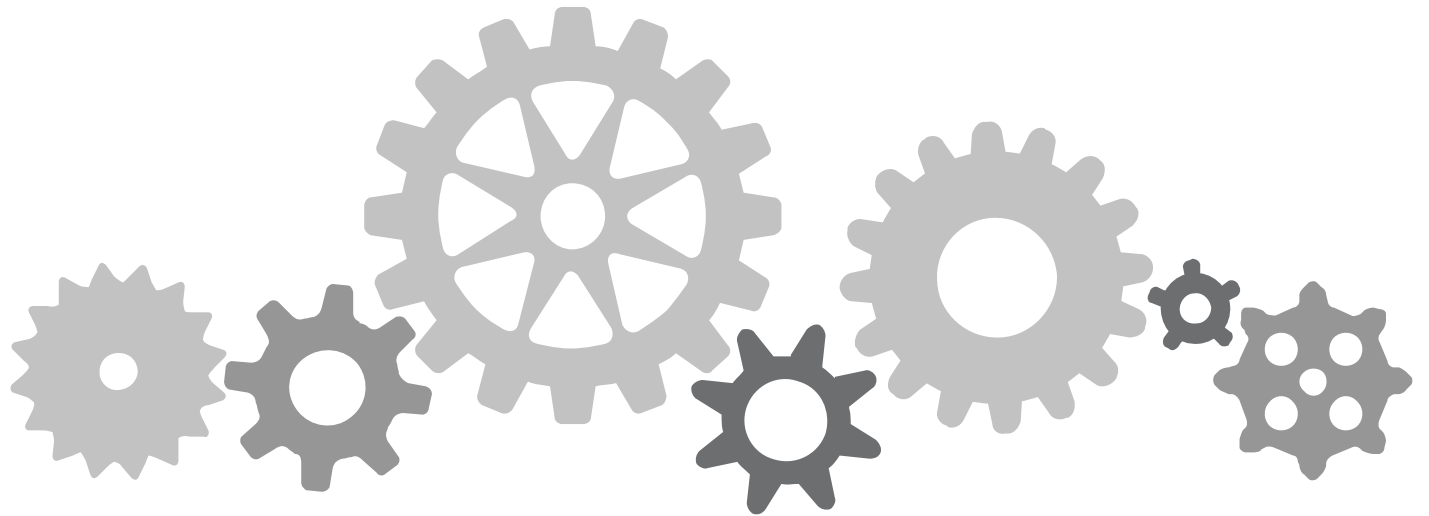
運用という名の最後の片づけだけで満足ですか？





基本

DevOps は 難しい



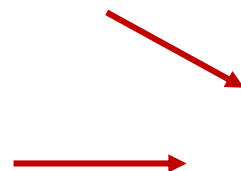
Dev と Ops は違う世界

Project Management

曖昧な要件定義

様々な開発言語の台頭

IDE
統合開発環境



ALM



Agile

Release Management
TEST、TEST、TEST・・・

モバイル開発

短期開発



Monitoring

Scripting

Backup
Recovery

IT as a Service

ITIL



仮想化



Private
Cloud



Public
Cloud

Configuration
Management

Change
Management

Runbook
Automation

SDN/SDS

Site Recovery
非所有

Dev と Ops は違う世界

使ってるツールが違う

見ている世界が違う

IDE

ALM

Agile

Release Management
TEST TEST TEST...

ゴールが違う

お互いやる事が多い

使っている言葉が違う

業界内でも育ちが違う

ITIL

仮想化

Private
Cloud

Public
Cloud

担当する会社が違う

Configuration
Management

Change
Management

Runbook
Automation

SDN/SDS

Site Recovery
非所有

唯一合意できる共通点

最後は運用でカバー

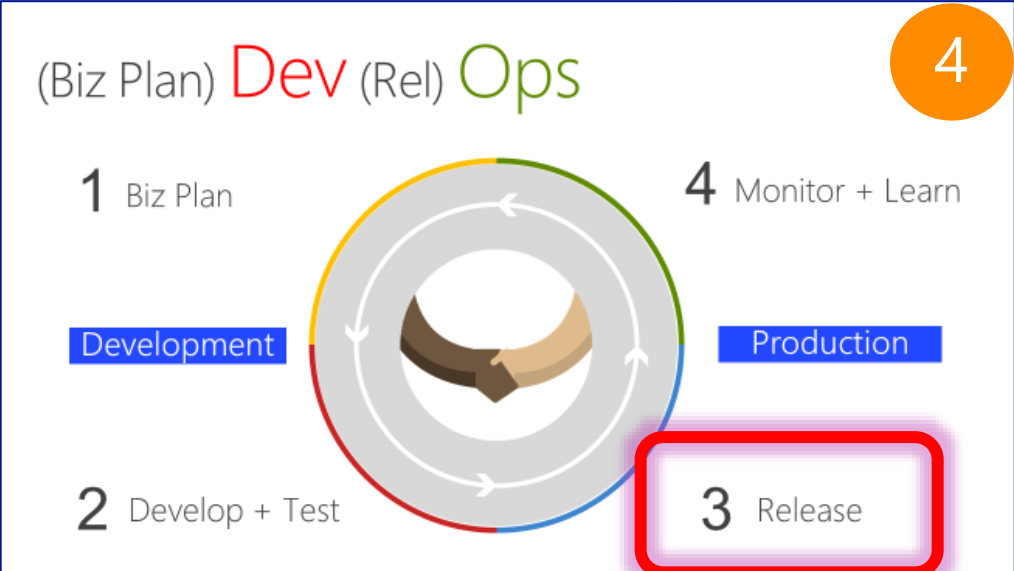
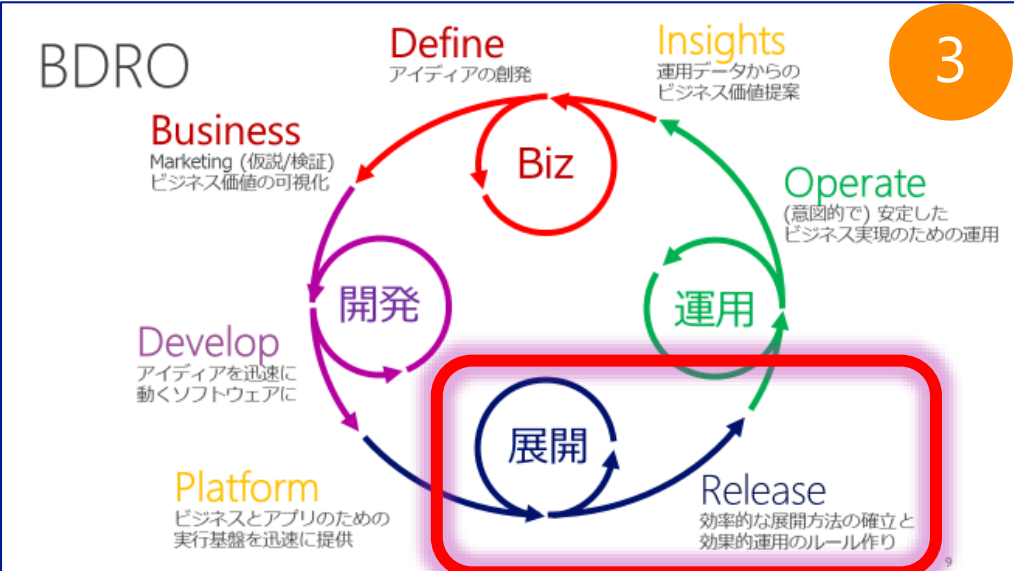
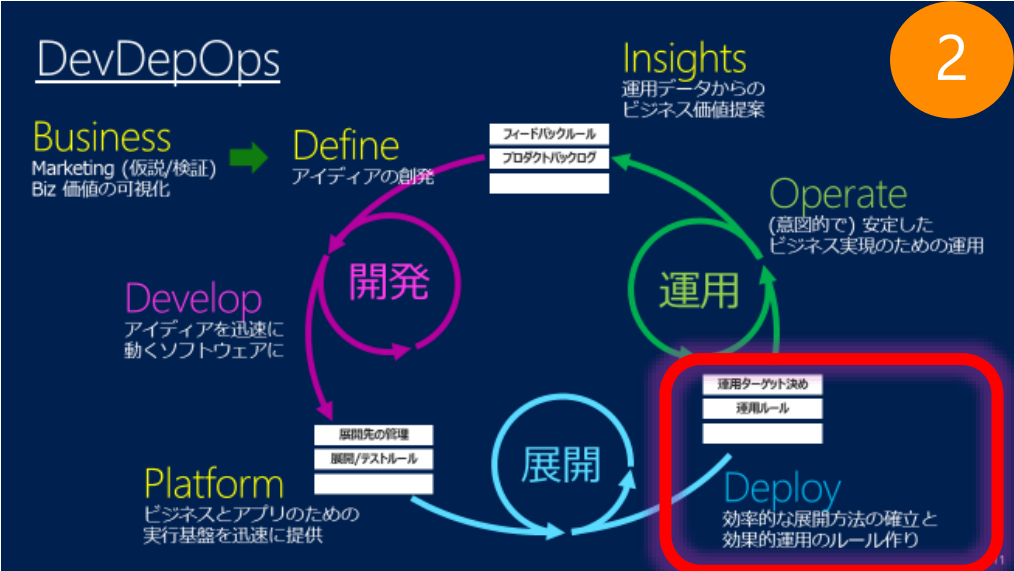
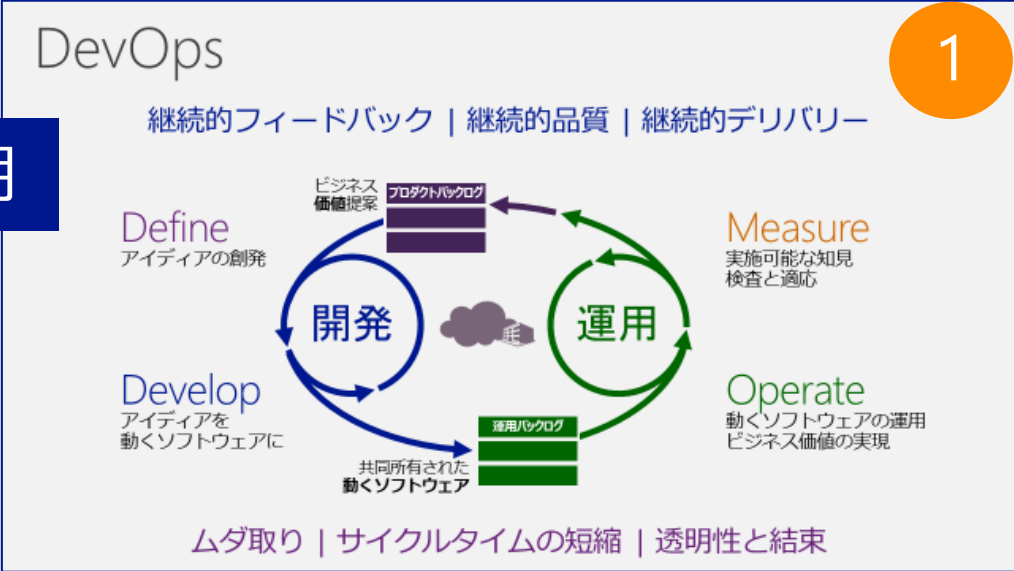
後工程へのしわ寄せは業界の常
(開発プロジェクトには期日があるが運用は・・・)



「出番が後工程ではない」とするとどうなるか？

試行錯誤して気づいた“中継ぎ”の存在

初期



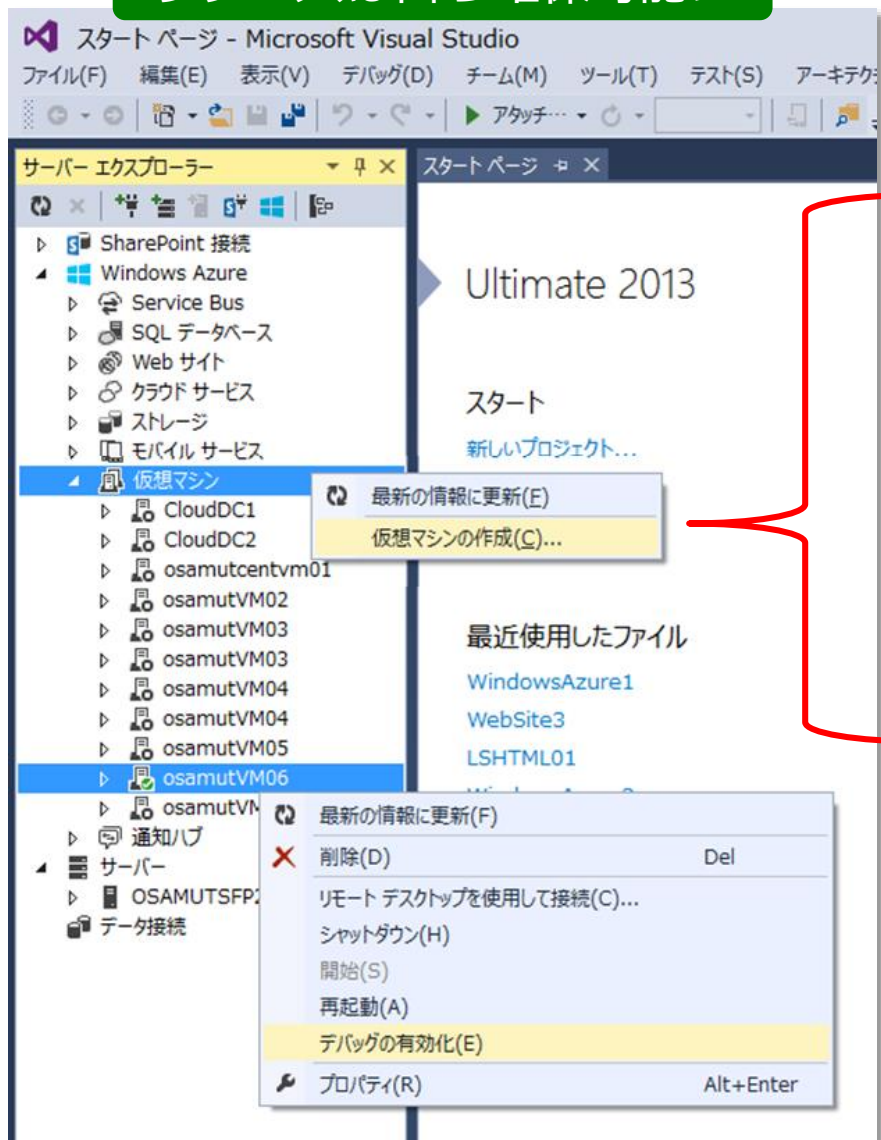


Release の現場から見た
3 つの大きな変化

現場の変化と DevOps

① Cloud の台頭 ～開発チームにとってのクラウド～

リソースは自ら確保可能に



ミドルウェアは
インストール済み

Microsoft SQL Server
SQL Server 2014 RTM Web
Windows Server 2012 R2

Microsoft SQL Server
SQL Server 2012 SP1 Enterprise
Windows Server 2012

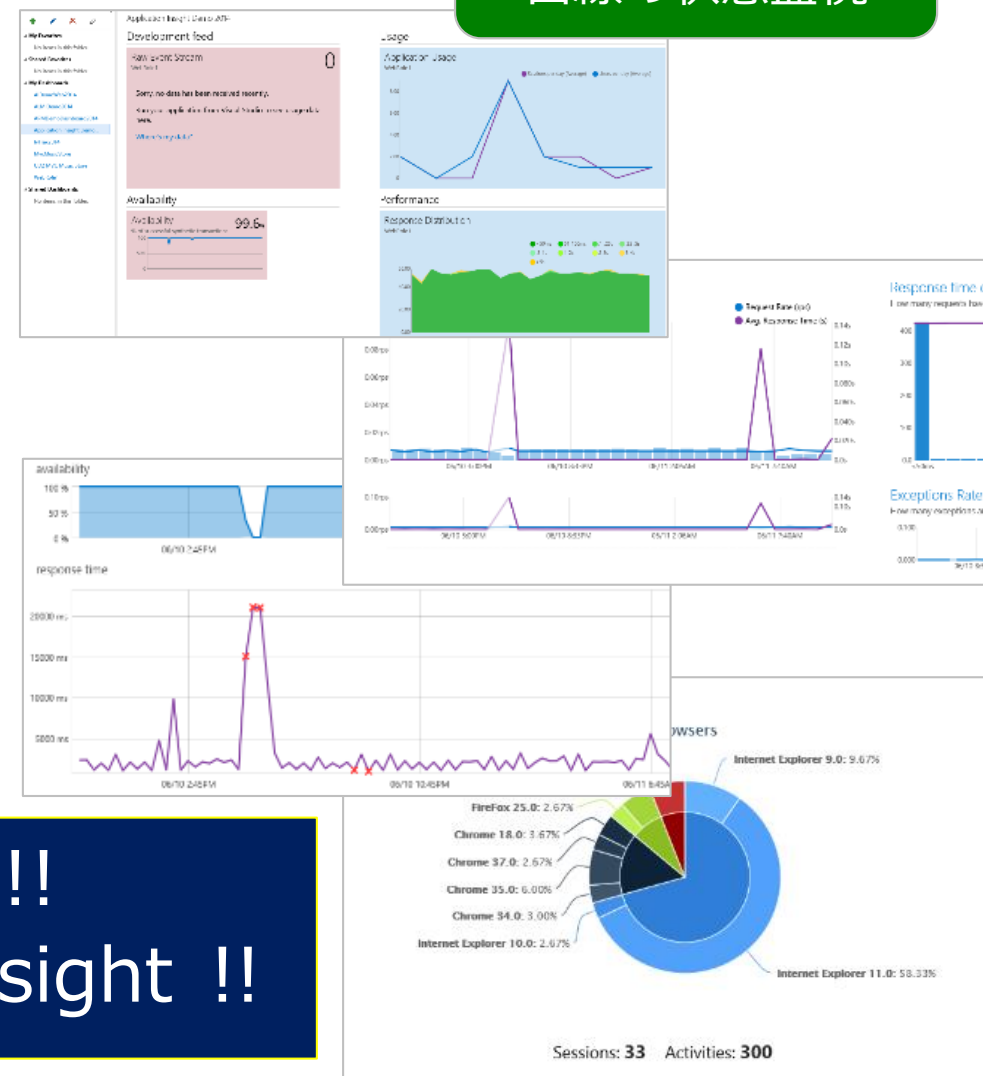
Oracle Database 11g R2 Enterprise Edition
Windows Server 2008 R2

Oracle Database 11g R2 Standard Edition
Windows Server 2008 R2

JDK 7
Windows Server 2012

Oracle WebLogic Server 12c Enterprise Edition
Windows Server 2012

アプリケーション
目線の状態監視

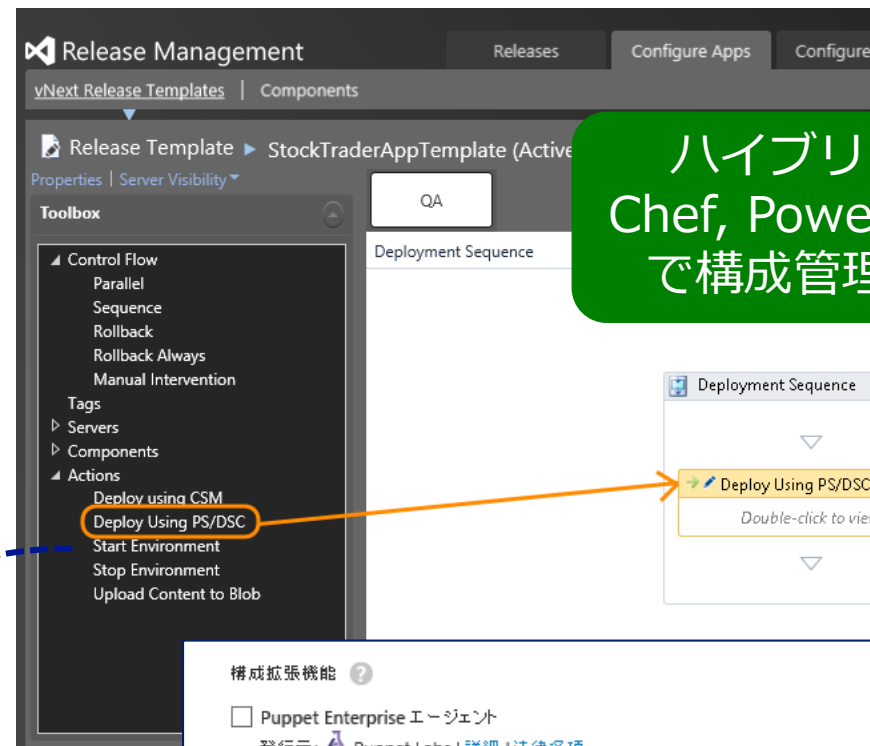


NoOps !!
運用からの Insight !!

Release ～効率化のためのテクニックが重要～

- OSS系 Build & Deploy Tool
- ファイルコピー
- 自作スクリプト
- Visual Studio Release Management
- . . .

- クラウドをスクリプトで管理
 - API 活用、PowerShell Script
- サーバー環境の自動構成管理基盤
 - Chef
 - PowerShell DSC
 - Puppet



ハイブリッド対応
Chef, PowerShell DSC
で構成管理も自動化



Azure VM
構成管理機能

② Container (Docker) の台頭

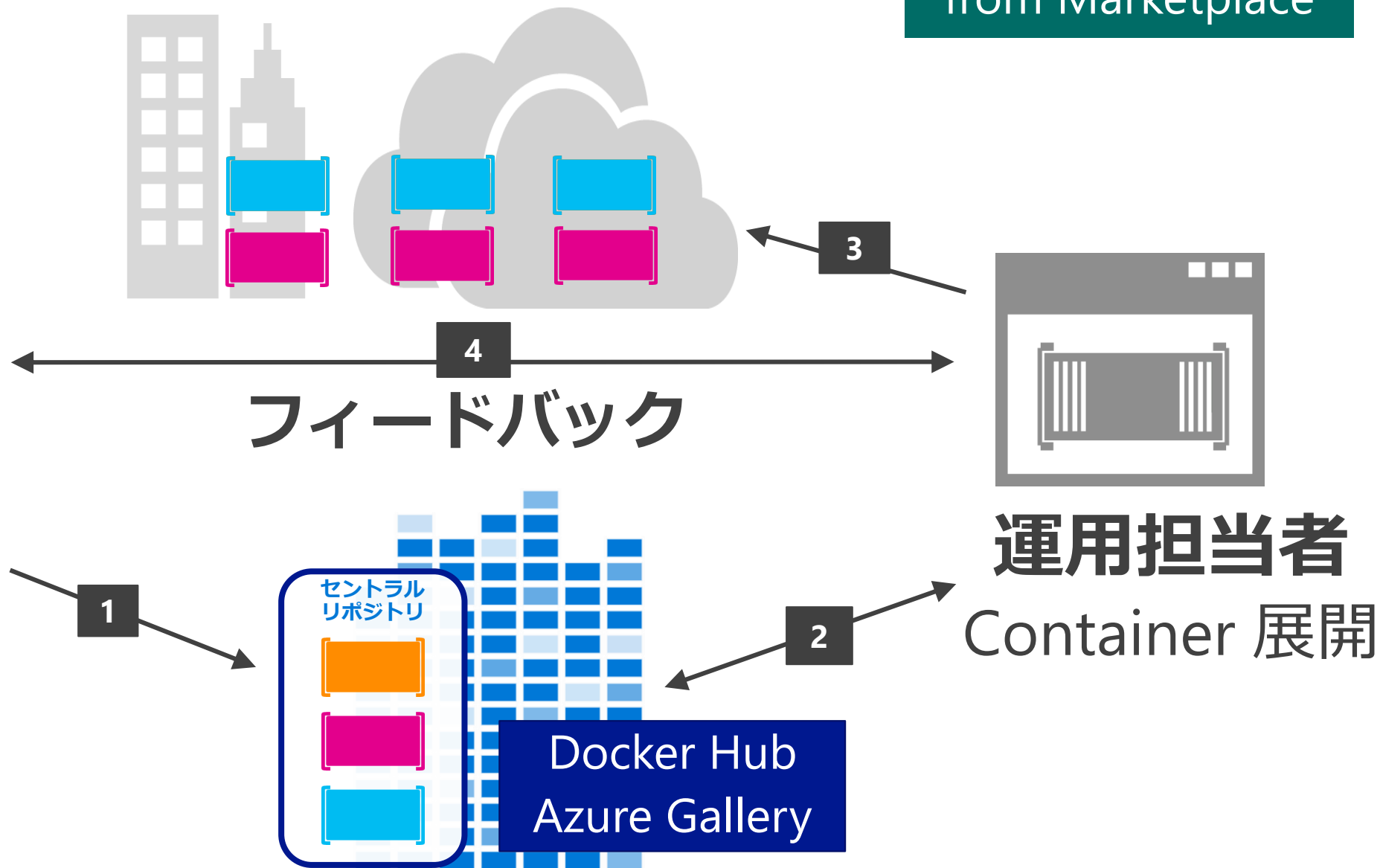
Release
from Store
from Marketplace

Container を
アップデート



開発者

Apps &
Container 作成



運用担当者

Container 展開

```
using System.Runtime.InteropServices;  
using System.Text;
```

アプリケーション v1 のコード

```
namespace DemoApp  
{
```

```
    class Program  
    {
```

Immutable Infrastructure が現実のものへ . . .

```
        public static void Main(string[] args)  
        {
```

```
            Console.WriteLine("This is a pretty cool app...");
```

```
            StringBuilder str = new StringBuilder(32);
```

```
            //_snwprintf(str, (IntPtr)str.Capacity, "Now it's really cool!");
```

```
            //Console.WriteLine(str);
```

```
        }
```

```
    }
```

```
}
```

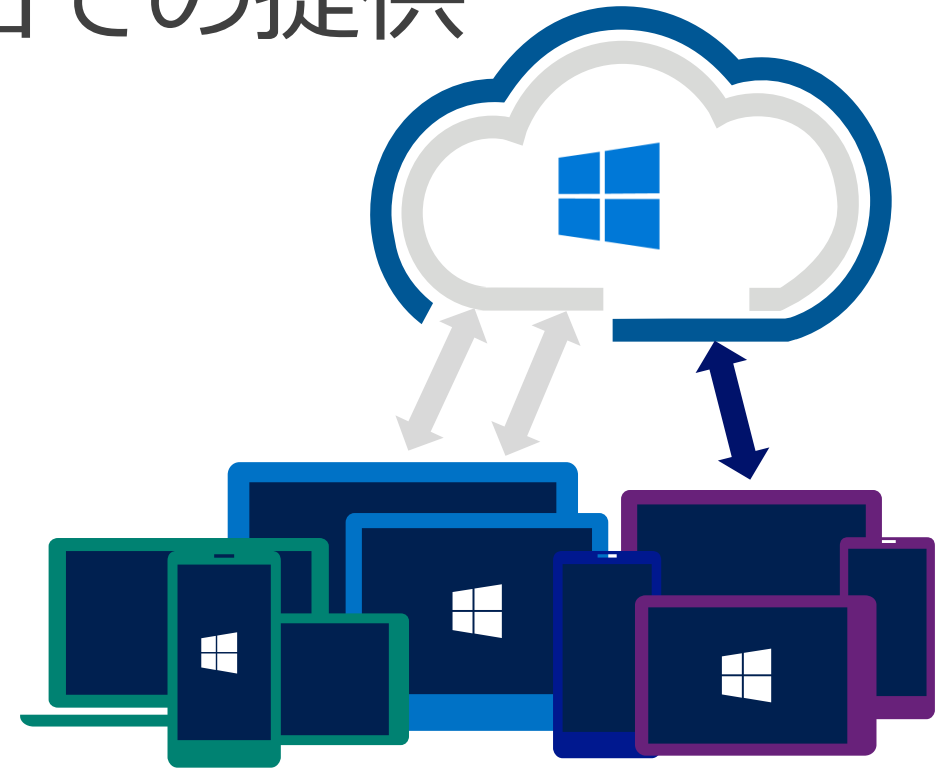
③ Windows as a Service の始まり

- 機能改善、新機能の追加を無料サービス化
- Windows Update (クラウド)経由での提供

さらに Windows 10 リリースから 1 年間
(同じ地点からスタートできるように)

- Windows 7/8.1 は 無料アップグレード ※

※ デバイスによって利用できる機能が異なる場合がありますので、
ハードウェアおよびソフトウェアの要件をご確認ください。
また、一部のエディションは対象外です。



Windows 10 時代の アプリケーションへの最新アプローチ

ユニバーサル Windows プラットフォーム モデル

アプリケーション仮想化 (Office 365 ProPlus) モデル

Microsoft Edge モデル

Dev
(Rel)
Ops

ユニバーサル Windows プラットフォーム モデル

小さなアップデートが前提のアプリケーション
展開(配信)とアップデートをストア(クラウド)型に変更



ユニバーサル Windows プラットフォーム
共通で一貫した API と マルチデバイス 1 アプリの実現

Windows プラットフォーム

Universal Windows 8 Apps (C++/C#/JS)

Universal Windows 10 Apps (C++/C#/JS)

Project Centennial (Classic Apps)

Web プラットフォーム

Microsoft Edge HTML エンジン

Cordova Tooling (HTML/JS)

Hosted Web Apps (HTML/JS)

他のモバイルプラットフォーム

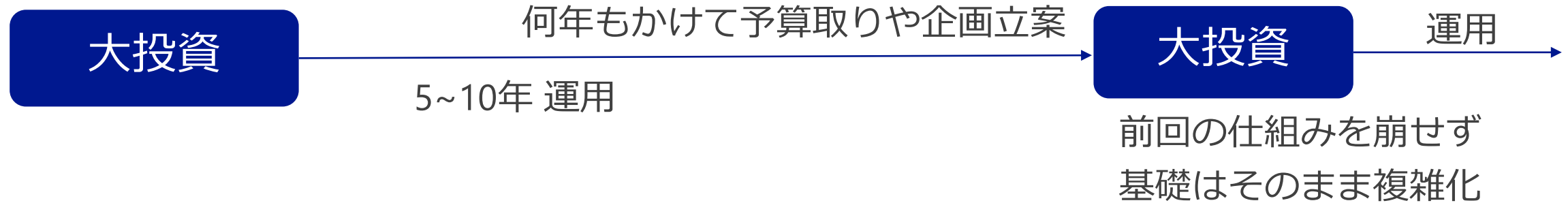
Project Astoria (Java/C++)

Project Islandwood (Objective C/C++)

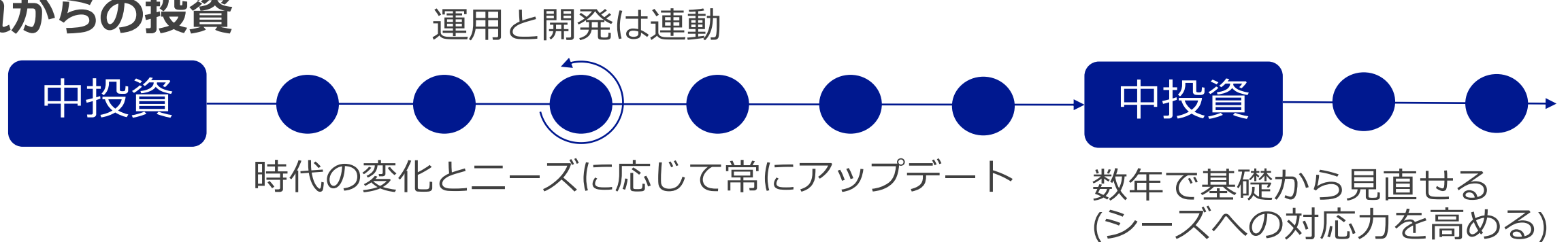
アプリケーション開発サイクルと投資の見直しを

プロジェクトのあり方そのものに見直しが必要
(解散しないプロジェクト)

これまでの投資



これからの投資





まとめ

運用は最後の工程にあらず

- Release が変われば運用が変わる
- 効率的 Release にはインフラ管理も欠かせない
- 開発 - 展開 - 運用 というプロセスが重要
 - ※ 自然な形で DevOps を実現

今こそ腕の見せ所
一緒に運用から IT を
変えていきましょう！

