Implementing DevOPS

Using DevOPS to Automate Northrop Grumman Software Development
Introduction

• Jim Brooks
• Northrop Grumman
• Software Development Analyst
• 24 years in IT
• 4 years working with Teamcenter Unified
• 2nd year using Agile methodology
• 6 kids
• Scoutmaster
• Swashbuckling Buffalo
What DevOPS is not
General Software Development

Global Product Data Interoperability Summit | 2017
DevOps: Build - Test - Release

Help your team continuously deliver software at a faster pace and with lower risk, while improving efficiency and collaboration between all teams that participate in release processes. Set up continuous integration builds for your app that run with every check in. Multi-platform build agents support C++, Java, .NET, and other applications.

In addition to continuous integration testing, you can create test plans, perform manual testing, and run unit tests. Release Management helps you automate the deployment and testing of your software in multiple environments. With it, you can either fully automate the delivery of your software all the way to production, or set up semi-automated processes with approvals and on-demand deployments.
• Typically Run from BMIDE.
  • Produces a whole bunch of files that are then zipped up for deployment.
Compiling our Software

We have created a batch file to automate compiles
- Understands that we have multiple Templates (solutions).
- Compiles all templates in the correct order.
- Sets the environment in a defined way (based on configuration stored in source control with the software)
- Calls MSBuild directly.
- Sets XML settings for the templates, sets the Template display name, and names files so that after a TEM deploy the source version (SHA) can be determined.
- Reports any errors.
- When Run from Devops(TFS) – Build results are left as an artifact that can be downloaded.
Compiling our Software

Global Product Data Interoperability Summit | 2017

25362-OneVault-AD / Build 25362-OneVault-AD_Dev180_20170905.1

.build succeeded

Ran for 9.9 minutes (TeamCenter-OneVault), completed 27.6 hours ago

Summary  Timeline  Artifacts  Tests

MDrop  Download  Explore
Part 1 – Build the Code

Developer checks new code into “Development” branch

The update is detected and the system triggers a build of that software on the build server

.zip Artifact created

Inform developers that build is staged for Release

Email failure reasons to the team

end
Approve the release

Pre-deployment approval pending
on Brooks, James D US (ES&CSO) (Reassign)
less than a minute ago

This Looks Good!

☑️ Defer this deployment to 9/7/2017 2:00 AM

Approve  Reject
Part 2 – Move to the Developer Test Server

Global Product Data Interoperability Summit | 2017

1. Pre Release Email
2. Files Copied to “Development” server.
3. Unzipped and TEM deployed
4. Testing on Development Server
5. Post Release Approval

- Migration has been cancelled by the developer/owner
- Errors are reported to the team
- Automatic or manual testing has failed
- Migration has been cancelled by the developer/owner
- end
• In order to automatically deploy we needed to use the TEM deploy in Batch mode.
• Unzipped the single compile file into separate templates.
• Set environment then call:
  • tem -update -full -templates= xxx -pass=xxx -path= xxx
Part 3 – Move to the Staging Server

Global Product Data Interoperability Summit | 2017

Pre Release Email

Files Copied to “Staging” server.

Unzipped and TEM deployed

Testing on Development Server

Post Release Approval

- Migration has been cancelled by the process owner (can be restarted later)
- Errors are reported to the team
- Automatic or manual testing has failed
- Migration has been cancelled by the developer/owner
Part 4 – Move to the Production Server

Global Product Data Interoperability Summit | 2017

- Pre Release Email
  - Files Copied to “Production” server.
    - Unzipped and TEM deployed
      - Testing on Production Server
        - Post Release Approval
          - Migration has been cancelled by the process owner (can be restarted later)
            - end
          - Errors are reported to the team
            - end
          - Automatic or manual testing has failed
            - end
          - Migration has been cancelled by the developer/owner
            - end
• Better cooperation between operations and development
• Automated builds provide us with greater stability.
• More confidence in the software we are deploying.
• Visibility/Verification that all functionality has been tested.
• Sets us in a better position to automate testing.
• Quicker “Agile” turn around.