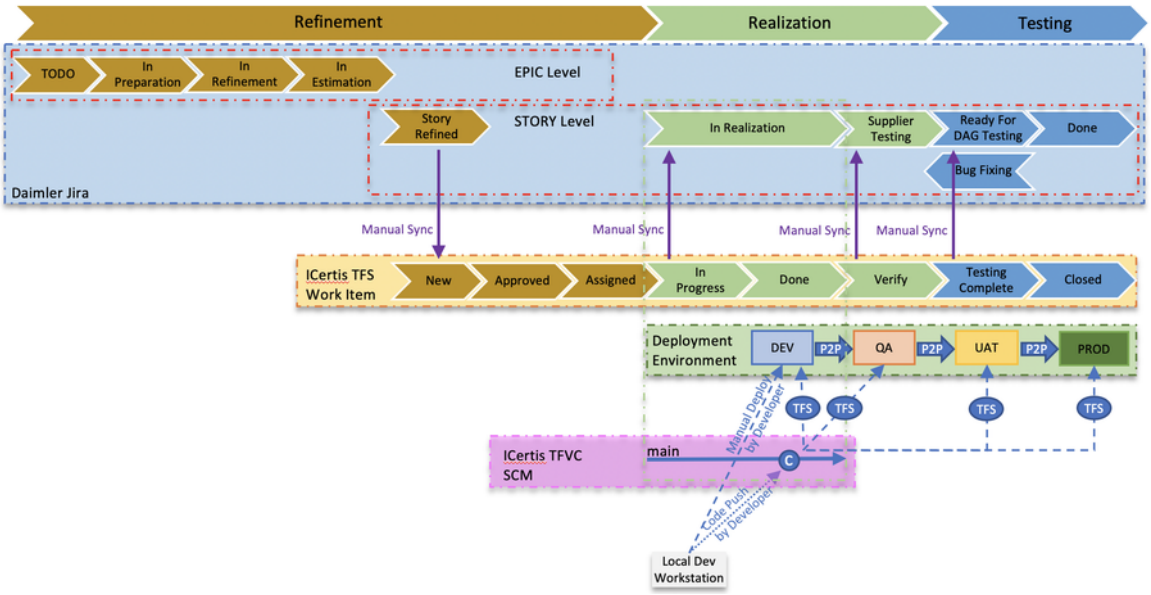
Daimler ICertis - Tool Chain Framework

Executive Summary

The Tool chain assessment focused on identifying the current tools utilized by both Daimler and ICertis for the DevOps pipeline, the integrations between each tool, and how the tools are leveraged. The target of the assessment was to identify any gaps, configuration improvements, and propose a target state tool chain that would help further automate, embed quality, and provide increased visibility and metrics in the pipeline process. Information on the current state was gathered during the onsite interviews conducted at the ICertis location.

The ICertis DevOps tool chain revolves around the Microsoft Team Foundation Server solution. This includes issue tracking, source control with TFVC, build orchestration, and test management. On the other hand, Daimler tracks issues with Atlassian Jira. Daimler's Jira instance is not linked to the ICertis TFS instance which requires manual syncing of data between the two systems. This leads to delayed status updates and potentially inaccurate data. There was also a lack of code quality engrained into the pipeline.

Current State



Issue Tracking (Jira and Team Foundation Server)

Issue tracking occurs in 2 separate systems.

- Daimler tracks Epics and Stories on the Daimler Jira instance.
- ICertis tracks stories in Team Foundation Server.

Daimler Jira Issue Types:

- **Epic**: Large unit of change that can span multiple quarterly releases.
- **Story**: Smallest unit tracked with a max story size being 20 story points (requiring a full 2 week sprint to complete 20 points).

ICertis Team Foundation Server Work Items:

- **Feature**: This is equivalent to Daimler **Epic**
- **PBI**: This is equivalent to Daimler **Story**
- **Task**: This is a break down of the **PBI**. Developers work on the task.

All of the Refinement activities are tracked on the Daimler Jira instance. ICertis manually creates a TFS **Feature** for each Daimler Jira **Epic**. On refined into Jira **Stories**, ICertis manually creates an associated TFS **PBI** to track internal development and further break down the **PBI** into TFS transitions thru the stages of the workflow, status is manually sync'ed back to the Daimler Jira instance. The manual syncs are time consuming, get instant status feedback to Daimler.

Source Code Management (Team Foundation Version Control)

ICertis leverages the Team Foundation Version Control (provided by TFS) for source code management.

Branching Model

Trunk based development is practiced with all developers checking in code directly to the **main** branch. The main branch is not gated and an hor expected to ensure their code compiles, is reviewed by peers, and tested. No enforceable gates can lead to an unreliable integrated **main** branc

The **main** branch is locked when release activities occur to ensure that no additional changes affect the main branch while the releasable code is stabilize the release are allowed on the **main** branch during this phase. All development changes for the next release will be put on hold until the hardened.

Continuous Integration / Continuous Delivery (Team Foundation Server)

Team Foundation Server (TFS) is used for build and deployment orchestration.

Build

Build definitions are configured in TFS. The builds are initiated manually and are usually executed prior to deploying to an environment:

- Build and Deploy from TFS to DEV environment at the end of the day on Mondays and Wednesdays
- Build and Deploy from TFS to QA environment at the end of the day on Tuesdays and Thursdays

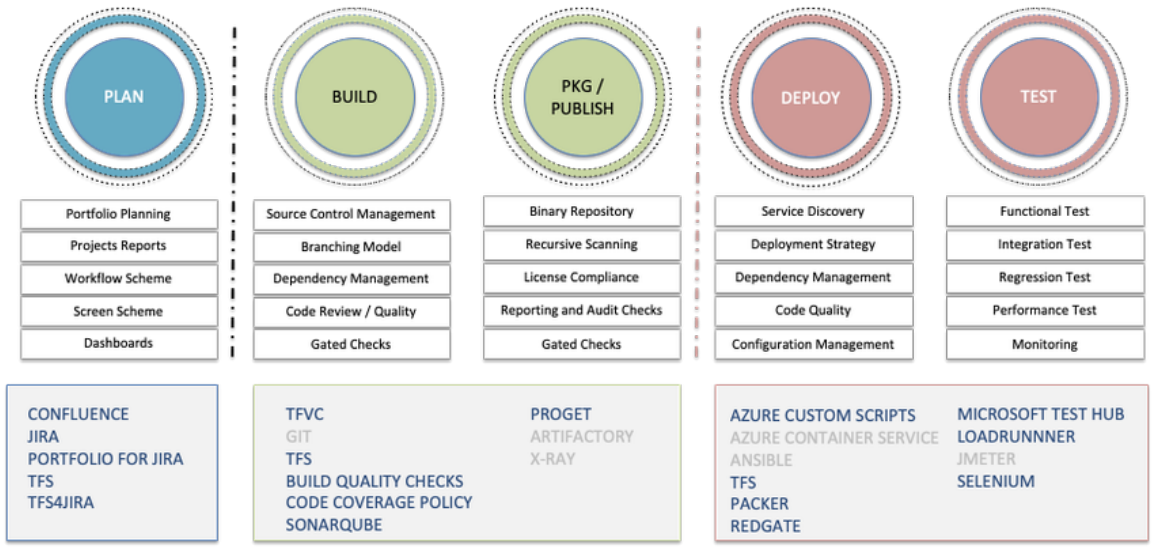
Application Deploy

Application deployment logic is scripted and configured in TFS. Developers are free to compile locally on their workstations and push binaries dir can lead to instability on the DEV environment since deployments are not done in a controlled manner by always utilizing the TFS build definitions artifacts. At any point in time, an understanding of the artifacts installed on the DEV environment can not be reliably determined since any develc at any point in time to the environment.

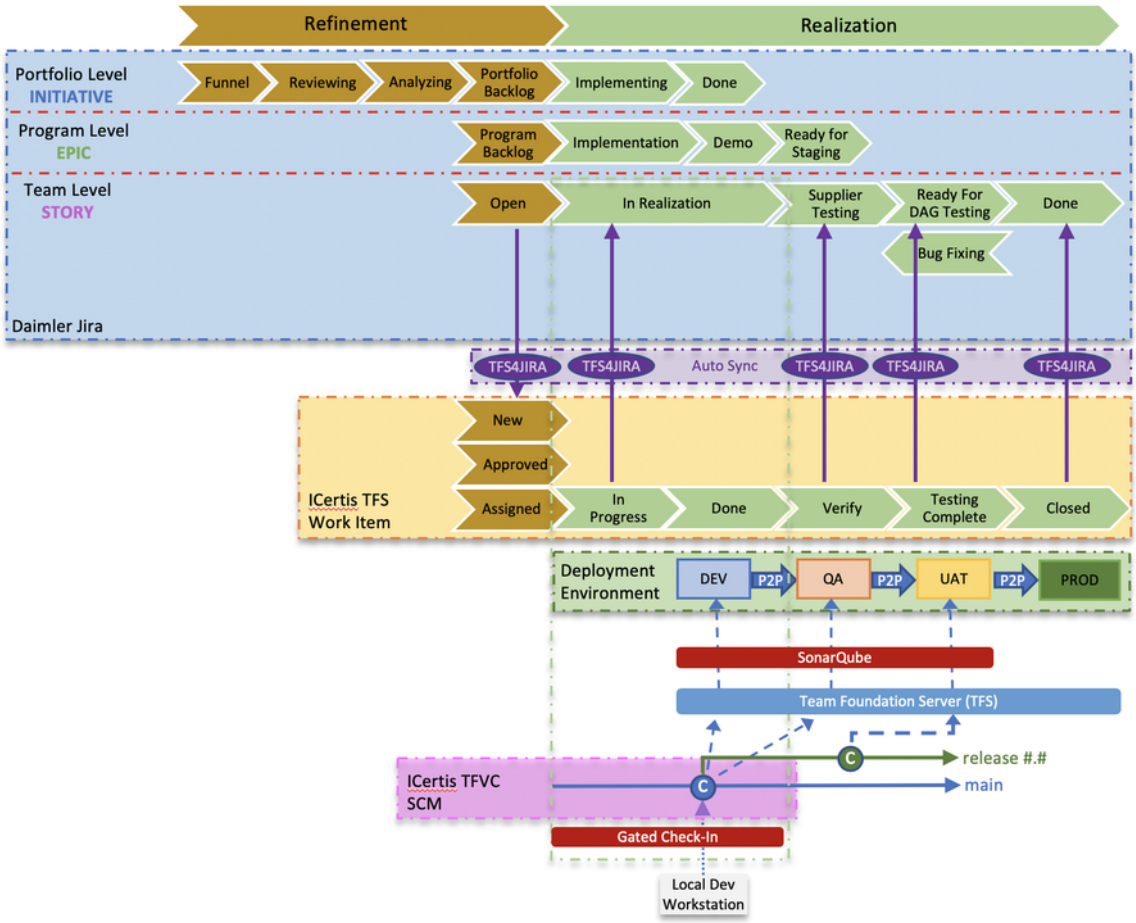
Database Migration

Database schema and configuration data changes are migrated from lower to upper environments using the P2P tool. The P2P tool is a homegr database differences between environments and to selectively migrate data from one environment to the next. All changes start in the DEV envir environments in sequence (DEV > QA > UAT > PROD).

Target State



Tool chain selection (highlighted tools in blue color at bottom of chart) are the preferred tools based on a phased implementation approach, availa and organization requirements. Items greyed-out are to be implemented in 2nd phase – or if there is a change in tool strategy.



Plan

The planning phase entails a collaboration tool and requirement lifecycle management. Collaboration tool for managing documentation, process
Requirement gathering tool to support development, test, and release processes.

Tool	Description
Confluence / Wiki	Primary tool for collaboration and documentation related to product innovation, test plans, and organizational process and practice
Jira	• Product requirements structured into Epics and Features • Product owner / Scrum team creation of Stories and Tasks • Release Management process flows related to release trains (Production Change Request, Deploy Request, Infrastructure Cl • Infrastructure Change Control Management
Portfolio for Jira	Jira plugin that provides portfolio management capabilities to Jira. Key features: • Allows adding additional hierarchy levels. Recommend adding an **Initiative** issue type that is one level above Epic (**Initiative** Epics and provides better visibility on status for the group of Epics. This will also allow Daimler to adopt SAFe and start mana Increment planning. • Better management of a Release Train • Enhanced schedule management with the capability to configure different scenarios with options like velocity and teams to se Further information available in the Scaled Agile with Atlassian and SAFe document.
TFS	ICertis utilizes TFS internally for work item tracking and setting up sprints. Recommendation is to continue utilizing TFS for issue t tight integration with code changes in TFVC.
TFS4JIRA	TFS4JIRA is a Jira plugin that integrates Microsoft TFS with Atlassian Jira. It allows you to see TFS checkins within Jira and sync will alleviate the need to manually update Daimler's Jira instance when TFS Work Item status changes and will provide a more acc status. Will require configurations and mappings between the two systems.

Build

The Build phase consist of developer IDE, source code management, branching for parallel development, build lifecycle management, code quali This phase requires standardization of tools and processes for consistent development practices. Requires fully automated and integrated solutic component / team requirements and maturity level.

Tool	Description
TFVC	ICertis currently leverages TFVC as their source control system. This is a centralized version control system that is provided by th current development practices at ICertis, but we recommend migrating to Git. Details are available in the Git section below.
Git	Git is a distributed version control system and is by far the most widely used modern source control system in the world. Microsof control unless you have a specific need for centralized version control features in TFVC. The 4 Not-So-Obvious Reasons Why You details some key differences between the 2 source control systems and the benefits of Git. The three big players in the Git Repository Management space - Atlassian Bitbucket, GitLab, and GitHub. Atlassian Bitbucket pro tools like Jira. GitLab provides a great open-source option. GitHub is a the most popular and widely used repository manager. Th the repository management system that best fits the organization.
TFS	ICertis utilizes Team Foundation Server for build orchestration. There are other orchestration tools on the market, but there is no time.
Build Quality Checks	*Build Quality Checks* is a plugin that can be used to fail TFS builds if a specific quality gate is not met. This can be used in conjun detailed below to ensure that code coverage policies are met.
Code Coverage Policy	The *Code Coverage Policy* plugin allows breaking the build if code coverage falls below a certain value or decreases between bui
SonarQube	Open source code quality and static analysis tool that provide instant feedback on poor development patterns, code duplication, v Organizational profiles and rules should be defined with built-in quality gates and leaks allowance (to be integrated with Continuou dashboard for real time updates and custom rules can be incorporated on out-of-box rule service. Need to provide database to m JIRA to create bug ticket types. ICertis attempted to get SonarQube configured in C7-SP1 but had issues configuring it with the solution file.
Checkmarx	Source code analysis tool for identifying, tracking, and repairing flaws in source code related to security vulnerabilities, compliance leveraged without having compiled source code (IDE), or integrated into continuous integration pipelines to ensure security quality

Package / Publish

Package and Publish phase centers around artifact lifecycle management. Application packaging is a process of binding the relevant files, reso software collection that provides functionality as part of a larger system. Artifacts are then published to an artifact repository system, ensuring reu

Tool	Description
ProGet	ProGet is a binary code repository that is used by ICertis primarily for managing Nuget packages. ProGet stems from the .NET wor seems to fit in the ICertis software development environment.
Artifactory	Open source binary repository manager allows hosting different package types in one platform. Artifactory can be used through a p where individual application components are stored and versioned. These versions can be assembled into a full product – allowing chunks, making more efficient use of resources, reducing build times, and better tracking of binary vulnerability management. If ICertis is looking to expand their binary code repository capabilities, Artifactory would be a great because it supports the most rep has the largest ecosystem of tools and integrations.
JFrog X-Ray	JFrog product provides multi-layer approach to analyzing software artifacts, metadata and containers for vulnerabilities, compliance Capability can be integrated to CI/CD workflow where Xray index artifacts in Artifactory repositories based on quality gates, on-dem

Deploy

Deployment strategy phase consists of deployment, activation, deactivation, updates, monitoring of software, database, configuration and infrastr strategy we can deploy our applications utilizing Blue/Green and Linear models to mitigate risk of application availability. Target state deployment virtualization and containerization platforms.

Tool	Description
Azure Custom Scripts	Azure Custom Scripts Extension downloads and executes scripts on Azure VMs. This can be used when creating a VM or anytime used to install applications or configure the VM as desired.
Azure Container Service	Azure Container Service (ACS) provides a way to simplify the creation, configuration, and management of a cluster of virtual machi containerized applications. ACS leverages Docker images to ensure that application containers are fully portable. It also supports your choice of Kubernetes, D Docker Swarm for orchestration to ensure that these applications can be scaled to thousands, even tens of thousands of containers
Ansible	Ansible is a "zero footprint" automation platform that builds on native management technologies that ship with the target system (Po case). It is used to automate the configurations of VMs. Ansible resources are declarative, meaning you define what the final state needed to achieve the final state.
TFS	ICertis leverages Team Foundation Server to orchestrate/initiate application deployments. Need to verify this with ICertis
Packer	Packer is used to automate custom VM image builds. ICertis currently leverages Packer to create the base images used in the Azu
RedGate	

Test

The test phase focuses on automation tools for providing comprehensive testing capabilities related to functional, integration, performance, syste testing activities. One key objective of Test phase is to enable shift-left concept, focusing on behavioral driven development incorporating test dri development phases enabling development teams access to implement and measure unit test, code quality, static analysis, static application sec integration testing, and regression testing. These capabilities should be enabled in the development workspace via dedicated development envir environment.

Tool	Description
Microsoft Test Hub	Microsoft Test Hub in TFS is a fully featured Test management solution to create and house Test Plans and Test Suites. ICertis windows strongly recommends using Test Hub because there will be no more Microsoft Test Manager versions after 2017. https://docs.microsoft.com/en-us/azure/devops/test/mtm/guidance-mtm-usage?view=vsts
LoadRunner	LoadRunner is a performance tool used for load testing and is currently being leveraged by ICertis. LoadRunner has been the le comes with steep licensing fees. Besides the cost of the software itself, running large scale performance tests with LoadRunner create your own performance testing lab, including an adequate number of virtual user license purchases. Aside from proper cap approach to scheduling performance tests, requiring advance coordination to secure performance testing time slots. There are other alternatives (like Jmeter) that are more cost effective and differ in the way they define load conditions.
Jmeter	Jmeter is an open source project that is very similar to LoadRunner as far as technical capabilities go. It essentially runs simulate that a real browser would make. In most cases, it is easier to use and has a more user friendly GUI. It also provides the capabili synchronized users as opposed to LoadRunner which defines load in terms requests per second. With the high costs of LoadRu
Selenium	Open source functional automation tool for web based applications. Selenium IDE provides quick and frequently recording and e for multiple test scripts to be executed at same time on multiple machines. Selenium WebDriver provides programming interface

Transformational Roadmap

Intermediate Phase

- SonarQube Integration
- TFS4JIRA Integration

Long Term Phase

- Azure Container Services implementation for DEV environment to allow for flexible environment for feature testing.