Tool Chain - Assessment Questions

Whiteboard Session

- Diagram the list of tools used in the DevOps life-cycle including any links between them (i.e. Jira has application link configured to BitBucket). Should include the tools used for:
  - Documentation
  - Requirements
  - Planning
  - Project Management
  - Issue Tracking
  - Compilation
  - CI/CD Orchestration
  - Static Code Analysis
  - Dynamic Code Analysis
  - Artifact Repository (compilation and deployment artifacts)
  - Test Automation Tools (including Performance)
  - Security Tools
  - Server Configuration Management
  - Monitoring and Logging
  - Production Tickets

Questionnaire

Question	**Why we ask?**	**How to score/rate?** **(1 lowest - 5 highest)**	**Weight?**	**What data can we pull to quantify?**	**Notes**
What tool is used for Source Control Management?	This will help us determine if they are using modern Source Control Management tools	1 - No Source Control System 3 - Centralized Source Control System 5 - Distributed Source Control System		We can verify by getting the URL to the source control system and list the repositories.	Next step to look at is ordering
Are integrated branches protected?	This will give us an idea on where gates are enforced in the lifecycle. A protected branch may exhibit some of the following gates: - build success - code review approval - passing quality gate - only a select few can merge	1 - No protected branches 2 - Branch is protected only by who can merge 3 - protected by who can merge and code review approval 4 - protected by who can merge, code review, and successful build 5 - protected by who can merge, code review, successful build, and passing quality gate		We can view the repository configuration to verify if the integrated branches are protected.	
What orchestration tool is used for CI/CD?	This will help us determine what additional questions to ask about the orchestration tool (i.e. Pipeline as Code)	1 - No CI/CD orchestration tool used 5 - CI/CD orchestration tool is used		We can verify by getting the URL of the CI/CD orchestration tool	

Build pipelines triggered on source control change?	This will help us understand how often and when builds are executed	1 - pipelines triggered manually 3 - pipelines triggered on SCM change for integrated branches only 5 - pipelines triggered on SCM change for integrated and feature branches		We can verify by spot checking a couple pipelines to verify that it is configured to trigger on source control change. We can then verify by the builds in the pipeline to see if the builds show that they have been triggered by SCM change.	
Is the CI/CD orchestration tool integrated with the Issue tracking tool?	This brings added visibility between the tools. Orchestration tool will have link and details to the issue associated with the change. Issue tracking tool will have build history of the changes associated with the issue.	1 - no integration 3 - one way integration 5 - two way integration		Verify by inspecting the orchestration and issue tracking tool	
How are users notified of build failures?	This will give us visibility on how user's are notified of failures and how they respond to the failure. Some notification methods produce a high level of noise and are sometimes ignored. Other methods are scoped to reduce noise and is more efficient for the user to manage.	1 - no notifications. user must log in to verify failures 2 - user notified for all build failures even if their commit is not involved 3 - user notified via email if build failure related to user's commit 4 - user notified 5 - user notified via email/chat and dashboard visible		Verify by inspecting the orchestration tool and optionally view a email notification if it is available.	
Are pipelines defined as source (Pipeline as Code)?	This will help us determine if there is a systematic way to handle drifts in pipeline configuration and change history	1 - Not defined in source our templates 3 - Defined as templates in tool but not in source 5 - Defined as code		Can be verified in source control system (i.e. existence of Jenkinsfile) and on the orchestration system to verify that the pipeline references SCM for configurations.	
Are common pipeline tasks defined as libraries that are consumed by other pipelines?	This will help us determine if common tasks are centralized instead of duplicated. Duplication can lead to drift in common tasks.	1 - common tasks not defined in libraries 3 - Orchestration tool does not support common libraries 5 - common tasks defined in libraries		Can be verified in orchestration tool.	

What tool is used for Deployments?	Provides a better understanding of how deployments are managed. This also provides follow on questions about best buy practices for the tool. Be sure that the tool mentioned is deploying the application and not just the underlying middleware.	1 - Deployments are manual with no documentation of steps 2 - Deployments are manual with documented run book 3 - Deployments have some automation with scripts and manual steps 4 - Deployments are fully automated with scripts or tool that follows a Imperative model 5 - Deployments are full automated with a tool that follows a Declarative model		Interviewee can provide necessary evidence. Examples can be: ▪ URL to documentation ▪ Sample scripts ▪ URL to deployment tool	
What tools are used to detect drift between deployment environments?	This will help determine if there is governance and visibility between environments and if there are differences between production. Having non-prod environment configurations similar to production will enable better detection of environment / middleware issues in non-prod before it surfaces in production	1 - No tool to detect environment drift 3 - Tool that detects environment drift 5 - Tool detects environment drift and provides a one-click mechanism to sync up environments on a selective basis		URL to tool and demonstration of how it is used	
What tool is used for Release Management Orchestration?	This will help determine what is used to manage Release Trains				
What tool or process is used to manage go/no-go release decisions?	This will give us an idea on what the process is for determining release readiness and the documentation and tracking associated around it.				
What tools/frameworks are used for Testing (functional, regression, performance)?	This will determine what tools/processes used for functional, regression, and performance testing.	1 - No tools used for test automation (all manual) 3 - tools utilized for one or more of the following but not all (functional, regression, performance) 5 - tools utilized for all testing (functional, regression, and performance)		Interviewee can demonstrate evidence of tools used.	
Are functional test gates enforced?	This will help determine if gates are in place to fail the pipeline if functional tests fail.	1 - No gates are enforced 3 - Manual enforcement 5 - Enforcement integrated in the pipeline process to automatically fail the pipeline		Can be verified by investigating the pipeline to see if tests are executed as part of the process and if the pipeline is configured to exit on failure.	

Are performance test gates enforced?	This will help determine if gates are in place to fail the pipeline if performance measurements do not meet requirements	1 - No gates are enforced 3 - Manual enforcement 5 - Enforcement integrated in the pipeline process to automatically fail the pipeline		Can be verified by investigating the pipeline to see if tests are executed as part of the process and if the pipeline is configured to exit if threshold not met.	
What tools are used for Analytics?	This will help determine if metrics are gathered during the pipeline process, aggregated in a centralized tool/area, and utilized to make decisions on the release-ability of the service.	1 - Analytics not gathered 3 - Analytics gathered but not centralized within a tool 5 - Analytics gathered and centralized within a tool.		Interviewee can provide URL of analytics tool	
What tools are used for Monitoring?	This will help determine what tools are used to proactively identify issues before end users are impacted. Monitoring applied to the following areas: ▪ infrastructure monitoring ▪ application monitoring ▪ performance monitoring ▪ log monitoring	1 - No monitoring 3 - Monitoring on some but not all ares 5 - Monitoring in all areas			
What tools are used for Security?	This will help determine if security tools are used during the release process. Types of tools include: ▪ Source Code Analysis Security Testing (white box testing) ▪ Integration Security Testing (black box testing) ▪ Penetration Testing ▪ Vulnerability Scanning	1 - No security testing tools used 3 - Tools utilized in some areas of security testing 5 - Tools utilized in all security testing areas			
Are security gates enforced?	This will help determine if security gates are embedded in the pipeline	1 - Not enforced for any application 2 - Enforced manually for some applications 3 - Enforced manually if threshold not met (any application) 4 - Enforced programmatically for some applications and manual for remaining applications 5 - Enforced programmatically if security threshold not met			
When are the security tools executed in the release process?	This will help determine if the security tools are utilized early or late in the release pipeline	1 - not executed 2 - only executed in the last stages before release 4 - executed on the integrated branch prior to cutting a release 5 - executed prior to code pushed to the integrated branch			

For each security tool, what is the duration time for each execution with results returned?	This will help determine if the security tools take a long time to execute and this is the driving factor on why testing occurs late in the release pipeline and limits quick feedback loop.	1 - scan and results take more than a week 2 - scan and results > 1 day 3 - scan and results > 4 hrs 4 - scan and results > 1 hr 5 - scan and results < 1 hr			
What tool is used for code reviews?	This will determine what tool utilize for code reviews and what capabilities are available to gate the pipeline to ensure reviews are complete before release.	1 - no tools used for code reviews 5 - tool utilized for code reviews			
What tools are used for Collaboration / Issue Tracking?	This will help determine what tool is used for issue tracking.	1 - no tool used for issue tracking 3 - tool only used for task assignment but not associated with code changes 5 - tool used for task assignment and code changes are linked to issues within the tool			
What are the issue types?	This will help determine the categorization of issues (i.e. Theme, Initiative, Epic, Story, Task, Bug)				
What are the issue state workflows?	This will help determine the states of each issue type and how they transition thru the pipeline. This will also help us determine if there are multiple workflows for each application or if it is standardized across the organization	1 - No workflows defined 3 - Standard Open, In Progress, Closed 5 - States include standard plus In Review, Testing, etc		Tool administration can dump the configurations or provide screenshots of the workflow.	
What tool is used for Requirements Documentation?	This will help determine what tool is utilized for gathering requirements and if it is linked to the tasks in the issue tracking system	1 - Requirements are not documented 3 - Requirements documented but not linked to implementation Stories/Tasks 5 - Requirements documented and linked to implementation Stories/Tasks			
What tool is used for Knowledge Sharing and How-To documentation?	This will help determine what tool is used to share information across the organization (ex: Confluence, SharePoint)				
What tools are used for Code Quality (ie Static Code Analysis Tools)	This will let us know if the organization is scanning for code quality and add follow-on questions to determine how findings are addressed?	1 - no code quality tools utilized 5 - code quality tools utilized			

Are quality gates enforced for code quality findings?	This will let us know if there is enforcement to ensure that the quality findings are addressed	1 - only reporting, no enforcement 3 - manual enforcement by service owners 5 - enforced configured to fail the pipeline if quality gates are not met			
What tools are used for Database Automation?	This will help determine if there is a systematic approach to applying database changes	1 - No database automation 3 - database automated via scripts 5 - database automated and are idempotent (i.e. only executes tasks that have not been executed in a previous run)			
What tools are used for builds (ex: Maven, Ant, Gradle)?	This will give us an idea on whether the build tool is optimal for the language being compiled. It also gives us an idea of what follow-on questions (ex: How are dependencies defined in pom.xml)?				
What tools are used for configuration management and provisioning?	This will let us know if they are using tools that manage infrastructure as code. It also provides follow on questions for how they manage the infrastructure.	1 - Provision is manual and not documented 2 - Provision is manual but steps are documented 3 - Provision is automated via scripts 4 - Provision is automated via Infrastructure as Code tool but definitions are not idempotent 5 - Provision is automated via Infrastructure as Code tool and definitions are idempotent			
What tool is used for Container Orchestration?	This will let us know what additional questions to ask around Container Orchestration (ex: Kubernetes, Mesos, Docker Swarm/UCP)	1 - No container technologies used within organization 3 - Container technologies used but no clustering or orchestration configured 5 - Container orchestration used			
What tools are used for log analysis/aggregation?	This will let us know if there are any tools used for logging, indexing, and aggregation.	1 - No log aggregation tools 3 - Logs are aggregated and centralized, but no alerting mechanism configured 5 - Logs are aggregated and centralized with alert mechanisms configured			

What tools are used for Cloud Management / Orchestration?	This will let us know if they are using Cloud technologies and what things they have in place around Cloud provisioning, orchestration, monitoring, etc.	1 - Organization does not utilize Cloud solutions 3 - Cloud Solutions utilized but limited orchestration implemented 5 - Cloud Management tools utilized and exhibit high level of automation around provisioning and automated elastic provisioning and cleanup on load			
How are build agents provisioned and configured?	This will let us know if they have a standardized way to provision build agents to ensure that there is a consistent and elastic build environment.	1 - Agents are configured manually from a set of instructions 3 - Agents are provisioned and managed using Infrastructure as Code 5 - Agents are instantiated from Docker images based on Dockerfiles that's are e stored in source control			
What is the average, peak counts, and wait times of the build queue?	This will give us a sense of wait times for builds and let us know if there is any room to optimize resources.	1 - wait times > 5 mins 2 - wait time 2-5 mins 3 - wait times > 2 mins 4 - wait times 1-2 mins 5 - wait times < 1 min			
Are the performance and the health of the DevOps tools monitored?	This will ensure that the tools are available and do not impact release pipelines	1 - no monitoring in place. 2 - health monitoring in place for some tools. 3 - health monitoring in place for all tools. No performance monitoring. 4 - health monitoring in place for all tools. performance monitoring in place for some tools. 5 - health and performance monitoring in place for all tools			

How are tool issues identified, tracked, and resolved?	This will help determine if DevOps tool issues are tracked and addressed. I will also give us an insight on the reliability of the DevOps tool chain.	1 - no tracking of issues related to DevOps tools. 3 - Issues tracked. Open issue counts greater than 10 4 - Issues tracked. Open issues less than 10. one or more issues open longer than 1 day 5 -Issues tracked. Open issue counts less than 10. All open issues are less than 24 hours old.			
Are organization SLAs associated with DevOps Tools?	This will help us understand the organization's expectancy of DevOps tool availability.	1 - No SLAs defined 3 - SLAs defined but are defined in days 5 - SLAs defined as 4 hours or less			
Is there multiple instances of the tools to support Disaster Recovery and Upgrade testing?	This will help us understand how tools are managed when there is wide spread outages. I will also help us understand if there is a test environment for Tool upgrades before implementing in production.	1 - Only one instance of each tool. 2 - Some tools have multiple instances but not all. 3 - All tools have at least 2 instances to support failure 4 - All tools have at least 2 instances to support failure. Some tools hav instances pread across multiple data centers for DR. 5 - All tools have at least 2 instances and are spread across multiple data centers to support disaster recovery.			
Are unit test code coverage enforced?	This will help us understand if there is standardization across the organization that enforces unit test code coverage.	1 - No code coverage enforcement 2 - Some applications require code coverage percentage to be met. 3 - All applications require code coverage to be met. 4 - All applications require code coverage to be met with the lowest threshold greater than 50% 5 - All applications require code coverga to be met with the lowest threshold greater that 80%			

Does the build pipeline fail if unit tests do not pass?	This will help us understand if there is standardization across the organization that enforces successful unit test runs.	1 - Unit tests are not executed for all applications 3 - Unit tests executed for all applications but do not enforce that the tests pass 4 - Unit tests executed for all applications but only enforce that they pass for some applications (>50% of apps) 5 - Unit tests executed for all applications and pipeline fails if any tests fail.		▪ Can check the pom.xml to verify that application is not configured to skip tests ▪ Can check orchestration tool to ensure that tests are executed and configured to fail pipeline on test failure	
Are automated functional tests integrated to execute after deployment?	This will help us understand the level of test automation and if it is configured to always run - ensuring code is tested and provide quick feedback.	1 - automated tests are not configured to automatically execute after deployments 3 - automated tests are configured to automatically run after deployment for some applications 5 - automated tests are configured to automatically run after deployment for all applications.			
Issue Tracker integrated with Source Control	This will give us an idea of the level of integration between tools that provides higher visibility and enables gates throughout the pipeline.	1 - No integration 3 - Integrated for reporting capabilities 5 - Integrated for reporting capabilities and automated actions configured when action performed in other tool.			
Issue Tracker integrated with Code Review Tool	This will give us an idea of the level of integration between tools that provides higher visibility and enables gates throughout the pipeline.	1 - No integration 3 - Integrated for reporting capabilities 5 - Integrated for reporting capabilities and auto-transitions on different states of the review.			
Issue Tracker integrated with CI/CD Orchestration	This will give us an idea of the level of integration between tools that provides higher visibility and enables gates throughout the pipeline.	1 - No integration 3 - Integration for reporting capabilities 5 - Integration for reporting capabilities and triggers enable to perform actions between tools.			