

DevOps Assessment Report

DevOps Assessment for ABC

XYZ Program

TCS' BITG DevOps Center-of-Excellence (CoE)

July 2018

Confidentiality Statement**Confidentiality and Non-Disclosure Notice**

The information contained in this document is confidential and proprietary to TATA Consultancy Services. This information may not be disclosed, duplicated or used for any other purposes. The information contained in this document may not be released in whole or in part outside TCS for any purpose without the express written permission of TATA Consultancy Services.

Tata Code of Conduct

We, in our dealings, are self-regulated by a Code of Conduct as enshrined in the Tata Code of Conduct. We request your support in helping us adhere to the Code in letter and spirit. We request that any violation or potential violation of the Code by any person be promptly brought to the notice of the Local Ethics Counselor or the Principal Ethics Counselor or the CEO of TCS. All communication received in this regard will be treated and kept as confidential.

Table of Content

1.	About the Customer.....	4
2.	Assessment Scope.....	4
3.	Problem Statement and Objectives	6
4.	Participating Stakeholder Expectations	6
5.	Baselined Architectures (Process, Technology)	7
6.	Identification of Improvement Opportunities and Optimization	14
7.	Revised Technology Architecture	15
8.	Revised Process Architecture with Timelines	22
9.	DevOps Implementation Stories	25
10.	Role –Technology - Tools Mapping	28
11.	Key Metrics	29

1. About the Customer

ABC has decided to adopt a modern technology stack for building new systems. The XYZ program will be an initiative to leverage the new technology stack.

Expectation is that the XYZ initiative would not only lead to the modernization of the external facing portal but it is expected to pave the way for other future applications to adopt the new stack. Platform developed as part of XYZ will be leveraged for managing micro-apps owned by different business units within ABC. ABC team is also looking to leverage cloud, container technologies, DevOps & Agile practices.

2. Assessment Scope

The scope of the assessment is outlined below:

- ✓ Understand the current Software Engineering Landscape.
- ✓ Represent the Software engineering landscape in a quantitative fashion and create a network model of the Software Engineering process for the current state.
- ✓ The software engineering network model is then subjected to an algorithmic analysis and a critical path is identified for the current process.
- ✓ The critical path is then optimized with respect to DevOps practices, incorporating further automation, to identify a target state.
- ✓ Since the analysis considers the process and technology dimensions, the same is captured as architecture views with respect to process and technology respectively. People architecture, though is part of the assessment method used, would not be explicitly covered or represented as part of the scope for this assessment.

- ✓ Since all the aforesaid architectures are identified during baseline analysis, the corresponding target state architectures are derived from the quantitative analysis done as a part of the assessment.
- ✓ The gaps identified between the target state and current state is. Based on the same, the high level DevOps implementation stories required to achieve the target state, is included.
- ✓ The scope of this work described above is applicable for the application portfolios mentioned below:
 - APIGEE
 - PostgreSQL or Cosmos DB
 - Angular / Node JS
- ✓ The entire development will be on Azure which is ABC's preferred cloud platform.

3. Problem Statement and Objectives

Following are the challenges at a very high level that are either expressed by ABC at the inception of this work or identified by DevOps Assessment while analyzing the collected data.

1. Elapsed Time for an end to end release cycle is very high in current scenario
2. Due to all rigid manual and people dependent processes changes and enhancements are always costly and risk prone
3. Very basic level of automation results in substantially high operational costs

4. Participating Stakeholder Expectations

Business end state broad vision that are driving this study are:

1. Reduced release cycle time for updates/ content creation/ enhancements
2. Reduce the build, enhancements and operational costs
3. Reduce cost of quality for changes on an incremental basis by embracing quality engineering practices

5. Baselined Architectures (Process, Technology)

Following is the baselined process architecture in form of process network diagram. **This is baselined based on the process and technology practices adopted by existing projects and what is planned for this project. The baseline is derived from collected data, and is for reference only vis-a-vis the target state, as there is no current state defined for XYZ program.**

Notation used

Please note the following illustration for complete description of a typical network diagram (some of the details represented in the diagram below are not provided in the actual network diagram for better readability):

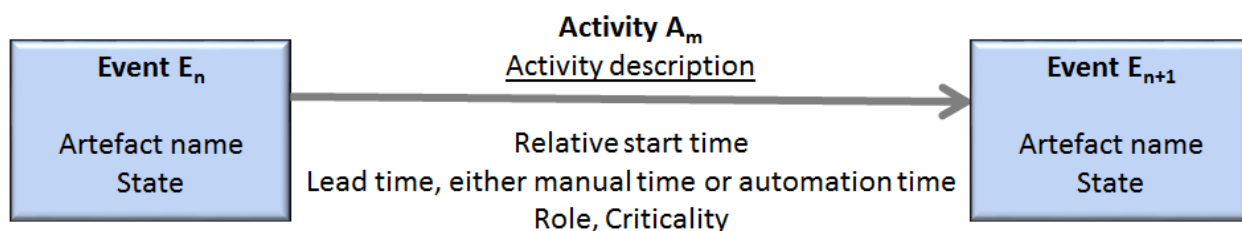


Figure 1: Process network diagram notation

- Here, **Event** (notation E_n , where n is an integer), depicts the input of the succeeding activity within the IT process. It also contains the state which can either be INIT (from which the initial activity in the process will emanate, hence will not have a preceding activity), INTERIM (which has both a preceding and a succeeding activity), and FINAL (which does not have a succeeding activity and thereby denotes an end state of the

process). INIT event for a single IT process is restricted to only one for the entire IT process, hence denotes the starting point of the IT process.

- The **Activity** is denoted by **A_n**, where n is a positive integer. An activity emanates from a state (which typically represents the input state) and finishes into another activity (which typically represents the output state for the activity). **Note** that the terms “story” and “activity” have been interchangeably used in context of this document. Further note that “Dummy” activities are used in the notation where there is no clear successor-predecessor relationship, however time or scope based dependencies across activities exist. Accompanying details covered in the As-Is network diagram are as follows:

- **Relative start time** – This is the start time of the activity relative to the preceding activity or activities; the relative start time for an activity emanating from the INIT state is taken as zero. In the diagram below, it is placed within brackets (...).
- **Role** – The role performing and/or owning the activity; valid values for this detail includes the following:
 - Business
 - Architect
 - IT Lead
 - Dev (incl. QA) (Development team which includes Quality assurance)
 - QA (Quality Assurance)
 - Ops-Rel (operations for release and deployment)
 - Ops-Infra (operations for infrastructure)
 - Dev & Ops Team (both Dev and Ops teams together)

Following is the baselined process architecture:

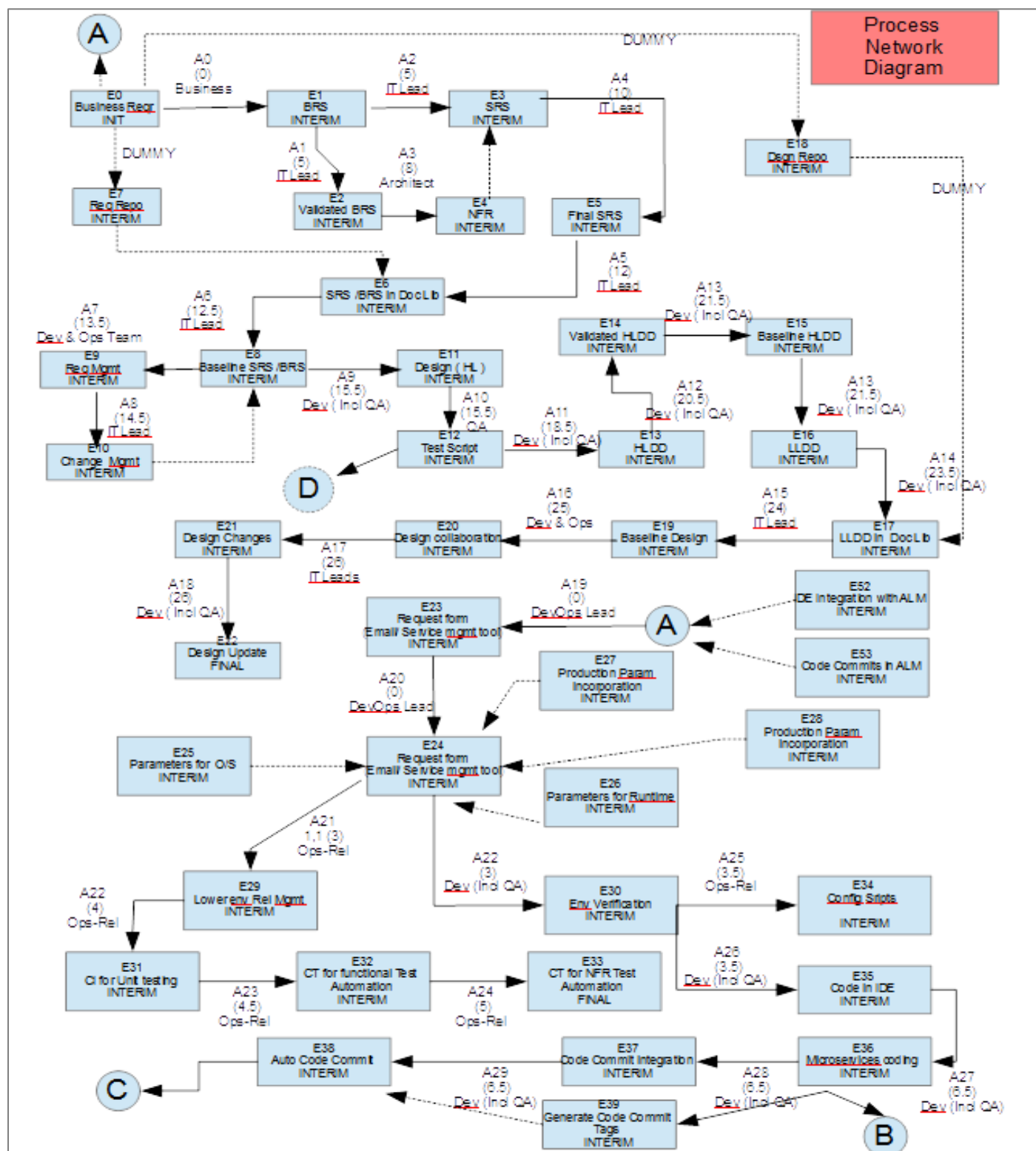


Figure 2: Baselined Process Network Diagram

(Contd.)

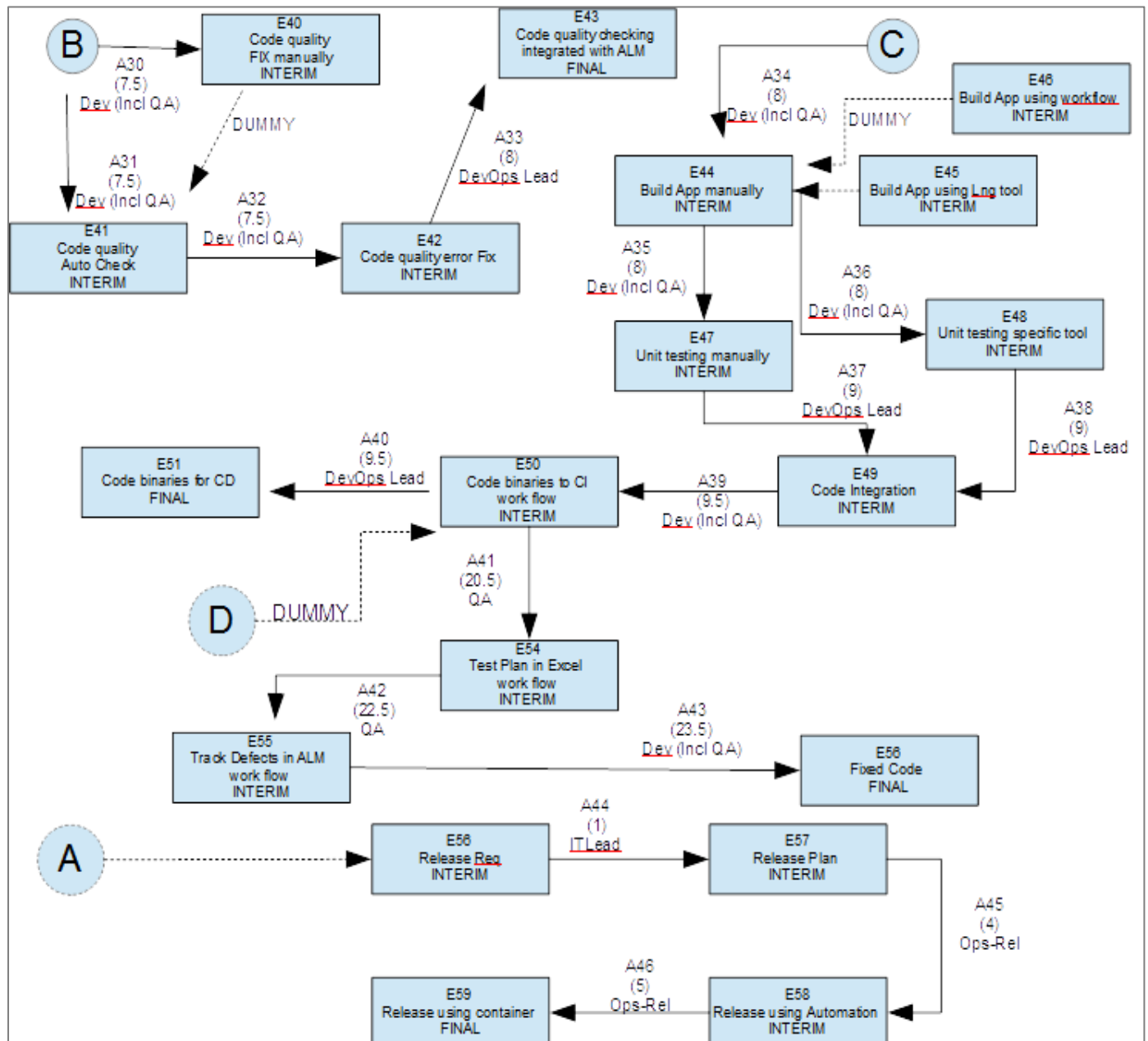


Figure 3: Baselined Process Network Diagram (Contd.)

Observations on the above process architecture

- The total deployment time to production for the given application is **5.2 weeks** based on the critical path analysis; note that this timeline is subject to variation based on factors including but not limited to actual operational issues, business decision on production releases, and IT personnel availability
- Release ready container is provisioned in **5 days** post receiving the requirement for release, hence such environment has a substantial waiting time before actual deployment
- Development for APIGEE stack has been primarily considered as such development would typically be subject to an SDLC (Software Development Life Cycle) cycle requiring build, test and release

Following is the baselined technology architecture mapped to the above process:

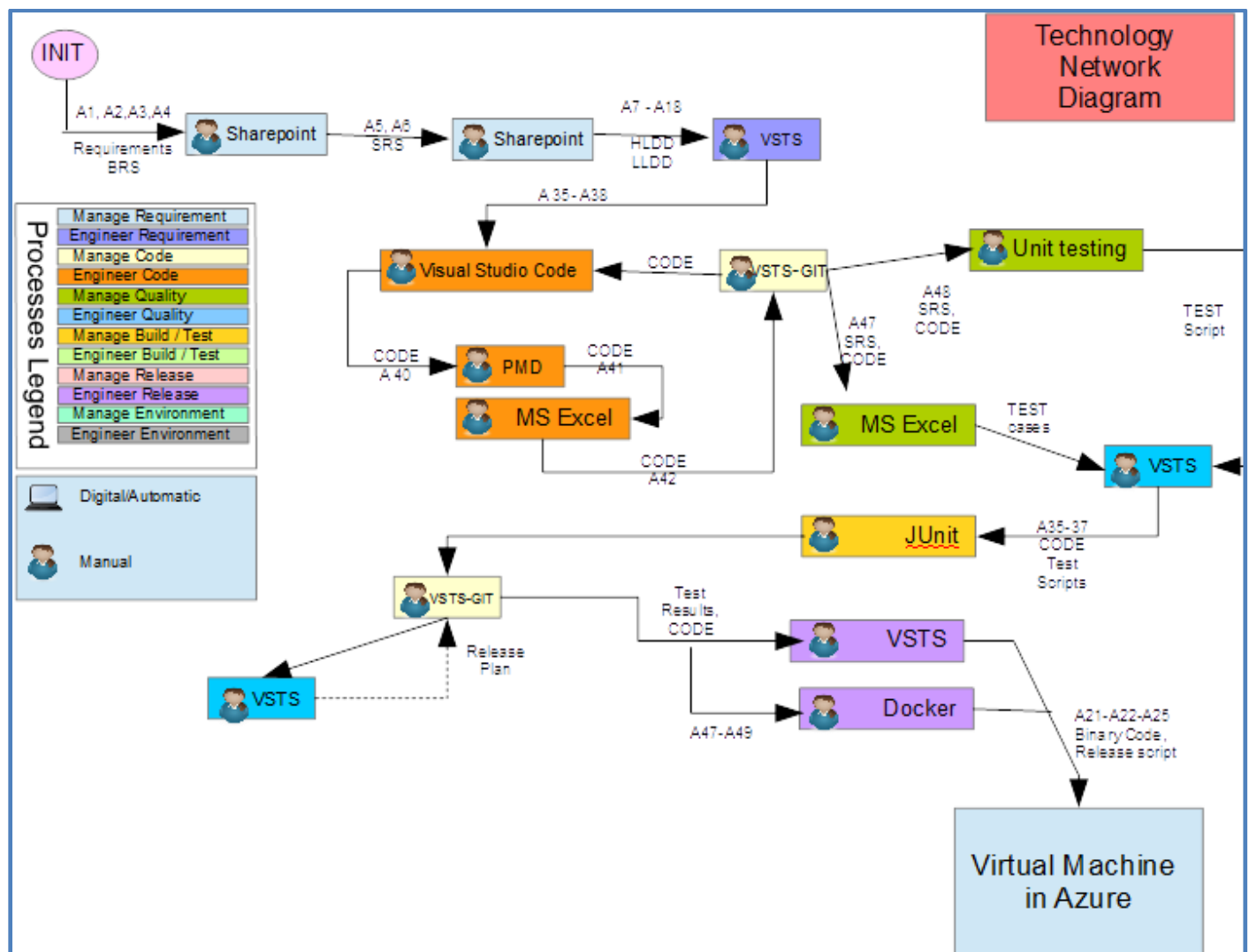


Figure 4: Baselined Technology Architecture

Observations on the above technology architecture

- The virtual machine (VM) for servers including accompanying environment on Azure platform is provisioned by the CoreInfra (Core-I) group.

- The developer themselves configure their virtual workstation after logging in to their VDI; there is no template based automated system for this.
- Request for server machine creation is placed through CoreInfra Cloud portal.
- For change and release management CA Service Desk manager is used. Release and deployment process is workflow based. However, present release and deployment process is manual without an integrated automation for the CI-CD (Continuous Integration – Continuous Deployment) pipeline.
- These technology and process mapping is done analyzing the current scenario in ABC's ecosystem and what is planned for XYZ

6. Identification of Improvement Opportunities and Optimization

Process

1. Currently there is very little automation in testing phase in overall ABC's ecosystem
2. Test results and defect tracking is not done through ALM tool
3. Final SIT (System Integration Testing) and Performance testing does not have a separate test environment
4. After the Dev team generates binary deployable artifact the release process starts with a manual entry in a tool and does not follow any automated process henceforth; also, there is no binary code repository
5. Test plan is maintained through Excel currently
6. Developer environment setup is not templated and is done manually after getting the individual VDI

Technology

1. There is little or no automatic flow between the different tools used in different phase
2. All the different engineering and manage activities are done through manual process leading to people dependency and time
3. One release cycle based on critical path is taking minimum 5.2 weeks duration with happy path considerations
4. Containerization is planned for one part of the deliverables but not synchronized
5. Release and movement of deployable across environments are happening primarily manually
6. Test automation is partially present with Jasmine and JUnit, but it is not deployed to cover 100% unit test automation

7. Revised Technology Architecture

Following is revised technology architecture mapped to the revised process architecture.
Revised state only indicates the target state.

The following diagram depicts a work stream progress with respect to a complete CI-CD cycle. The tools / technology are represented as mapped to the process flow. To indicate which tool is planned for which phase of the CI-CD cycle a color legend is used and the same is given in the box at the right.

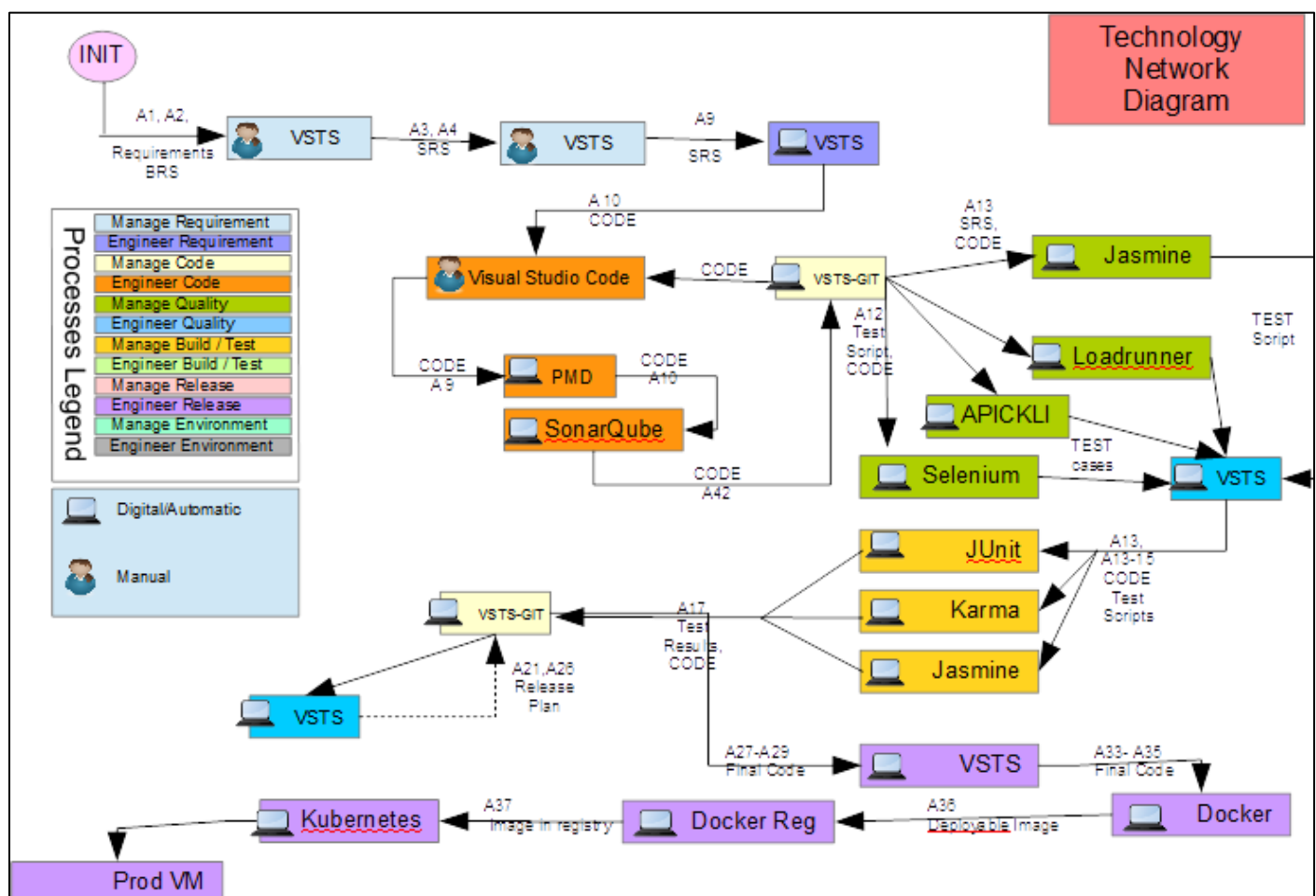


Figure6: Revised technology Architecture

Following is the technology state diagram conforming to the above architecture:

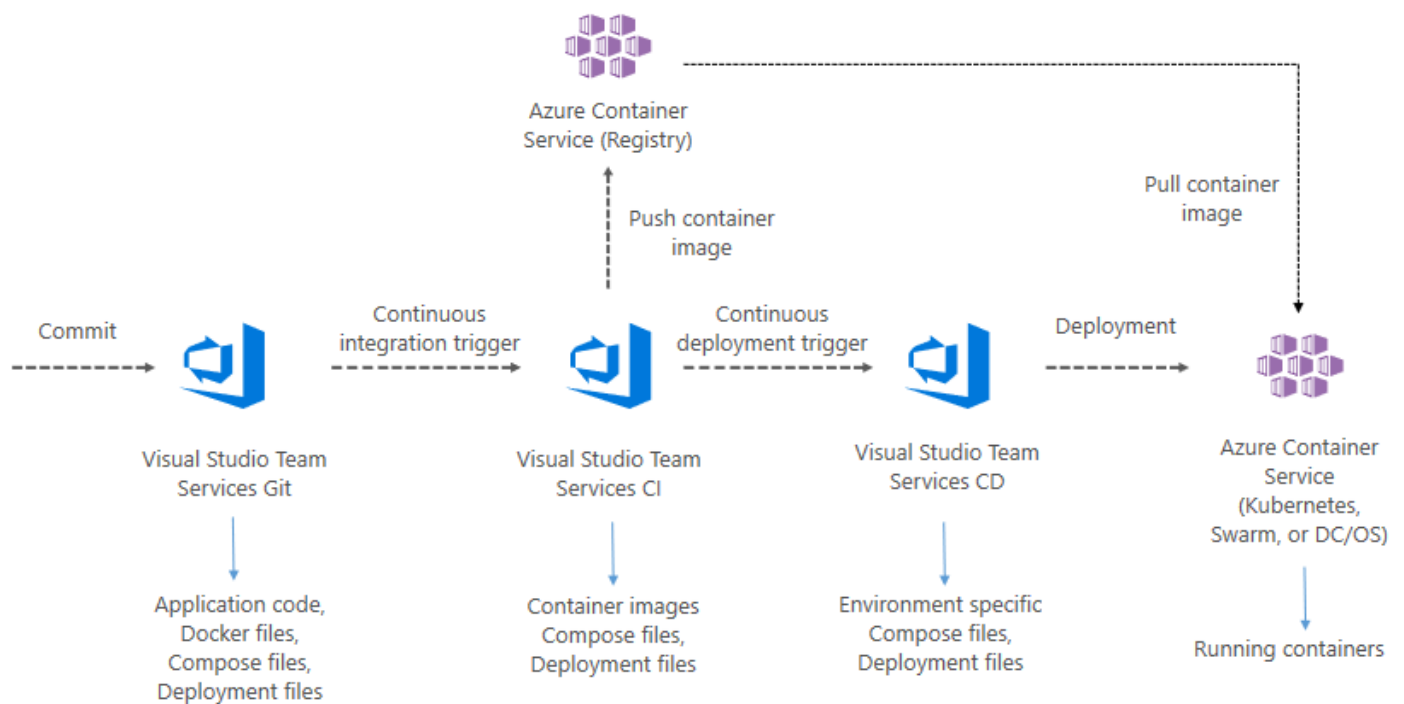
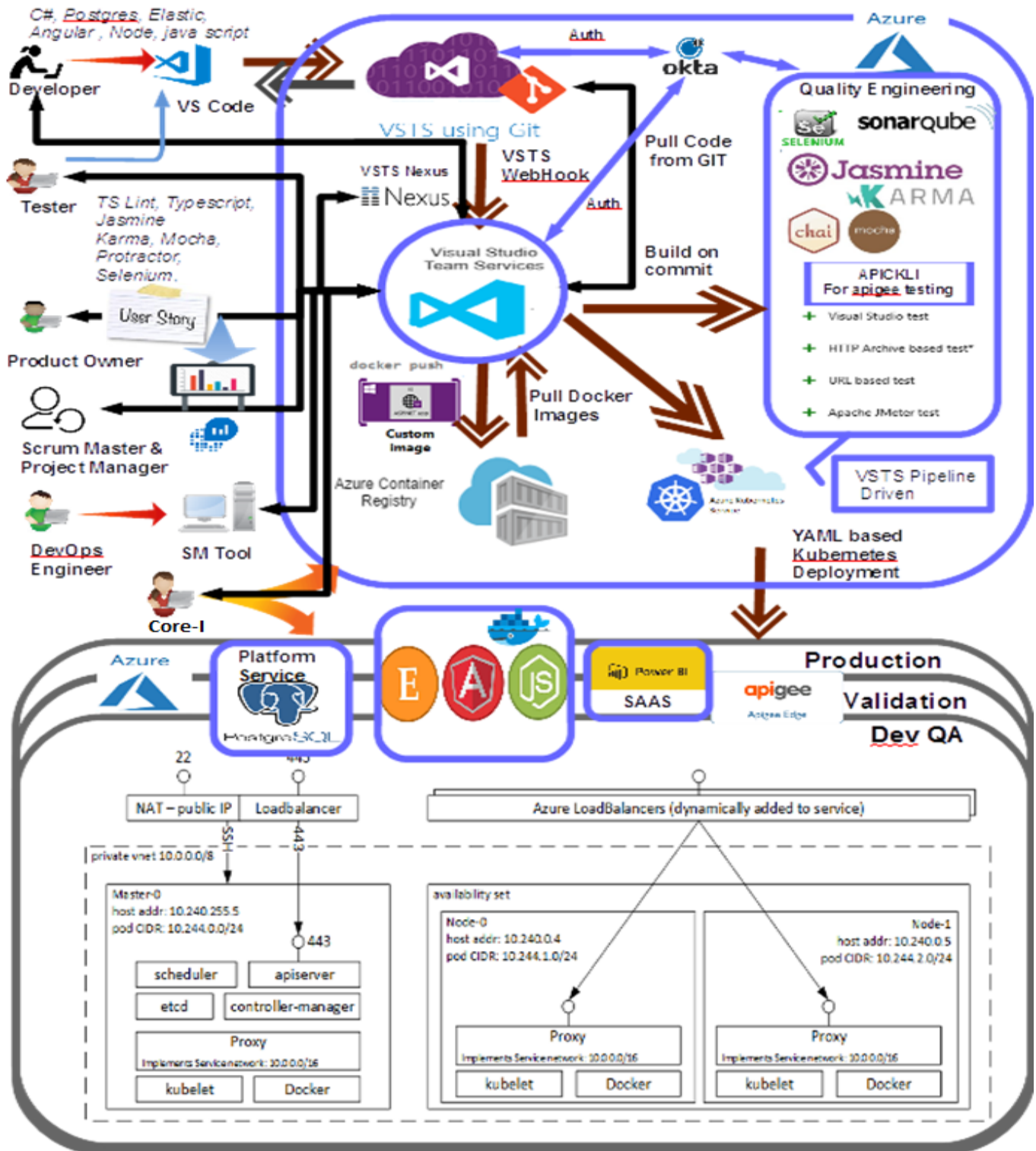


Figure7: Technology State Architecture- Reference Diagram by Microsoft



** IP Addresses are indicative and for reference only

Figure 8: Revised Technology State Architecture Network

Observations on the above technology architecture

1. This architecture provides a unified approach to DevOps for all the three application architecture styles, using containerized CI/CD
2. It uses VSTS as the container orchestration platform, with separate production and non-production Kubernetes clusters on Azure
3. Dev and QA will be contained in the non-production cluster
4. Visual Studio Code is used as the IDE for Node and Angular Stack
5. Common testing approach for all the three architecture styles with underlying variations to be considered during actual execution
6. The XYZ is proposed to implement the aforementioned modern state of art automation solution for faster delivery
7. The architecture encompasses the DevOps paradigm
8. The architecture uses Docker container as a mechanism to bundle code and environment together as IAAC (Infrastructure As A Code)
9. The architecture assumes that Microsoft Azure will be used as the cloud development platform
10. The choice of Microsoft Azure as well as VSTS is mandated by ABC
11. Visual studio code will be used as the IDE along with integration with VSTS and Azure plugins
12. All the automation and continuous integration to be used in the XYZ Program needs to be an Azure service; This is also one of the stated mandate by ABC
13. VSTS will be used for the following purpose
 - Source code Repository (VSTS-GIT)
 - Continuous Integration
 - ALM tool
 - Continuous Delivery tool
14. Azure Kubernetes Service [AKS] will be used for container (Docker) orchestration
15. Azure container registry [ACR] will be used to store the binary images
16. Application binary will be managed by VSTS and stored in NEXUS

17. ELK will be the analytics tool for visual representation of DevOps metrics
18. The entire Kubernetes cluster management and associated VM provisioning will be done by CoreInfra
19. YAML based pipeline as code will be used both for VSTS as well as for Kubernetes
20. The test automation tools will be a part of the VSTS pipeline, the automation tools considered as
 - SONARQUBE for code quality
 - Jasmine, Karma, Chai and Mocha - for web testing automation
 - Apickli for API testing along with Cucumber.js [for BDD], Gulp (alternatively, Grant) integration [for test execution workflow]
 - Selenium for functional testing
 - VSTS performance testing facility
21. The Azure Kubernetes service will provide the following
 - Deployment Automation
 - Identity and security integration
 - Support for logging and monitoring
 - Cluster node scaling
 - Cluster node upgrades
 - Http Application routing
22. VSTS will be used to provide the following
 - It will be used as an ALM tool
 - It will be used as a tool for the analyst to document user stories
 - it will be used by programmers for
 - integrated unit testing
 - source code control
 - automated builds
 - It will be used by testers to
 - Manage tests and create VSTS based test scripts
 - Advanced reporting for test analytics
 - It will be used by project managers, scrum masters to manage and monitor teams using various scrum tools.

23.The in-house Service management tool will be used for raising IT service requests. This request will be routed to VSTS and the relevant automation will be executed as a YAML based pipeline.

24.VSTS workflow will be used for any workflow setup needs

25.For authentication VSTS will integrate with OKTA / AD

APIGEE: Recommended CI-CD tools and workflow

Since the main objective of CI is to reduce the risk of breaking changes reaching and deployed to production environment, by relying on an automated process that validates and deploys changes as they are happening; increases the trust and improves the speed of new features and improvements delivered to the target. The CI-CD strategy for APIGEE for XYZ is recommended with the aid of the following diagram.

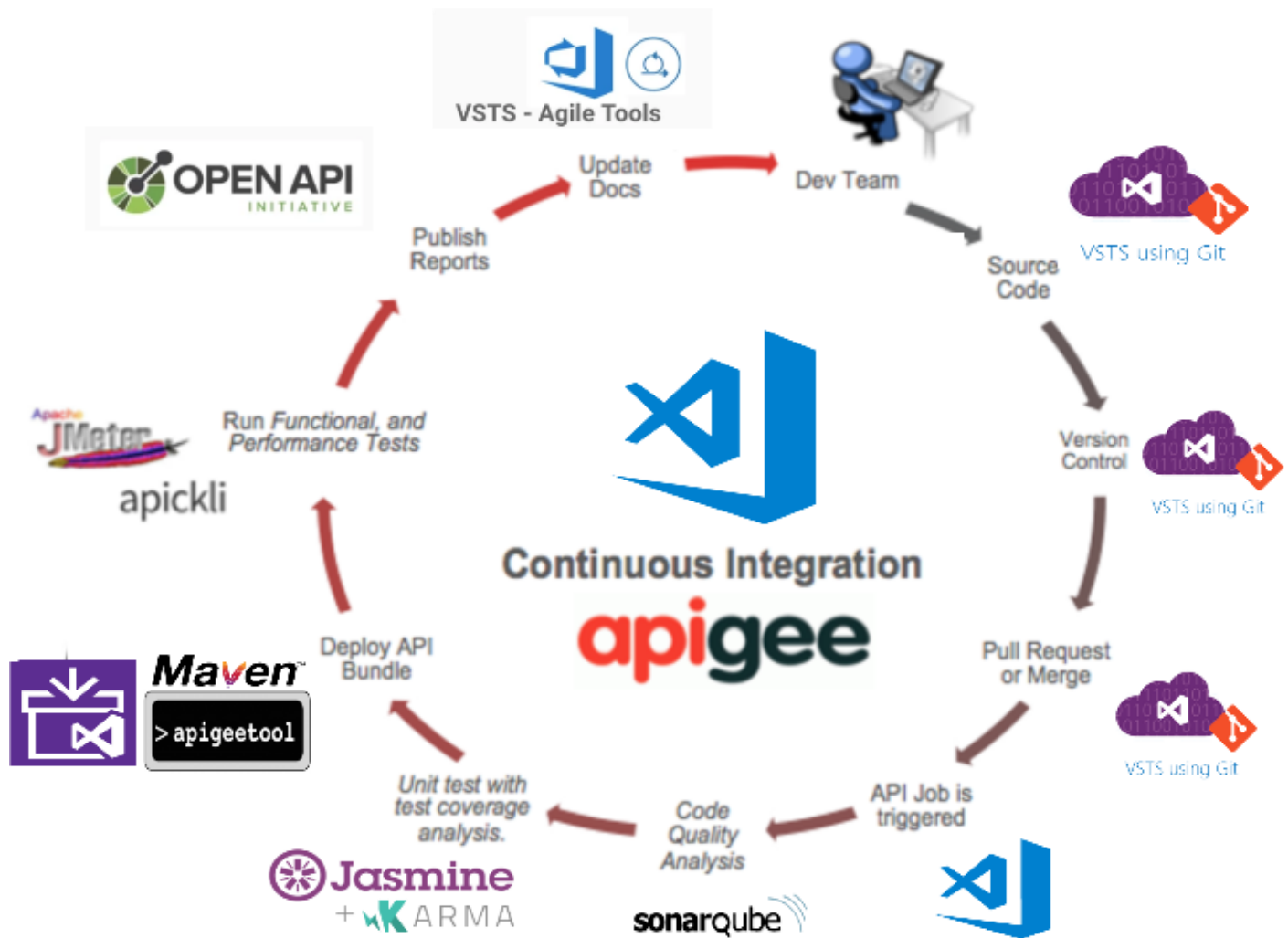







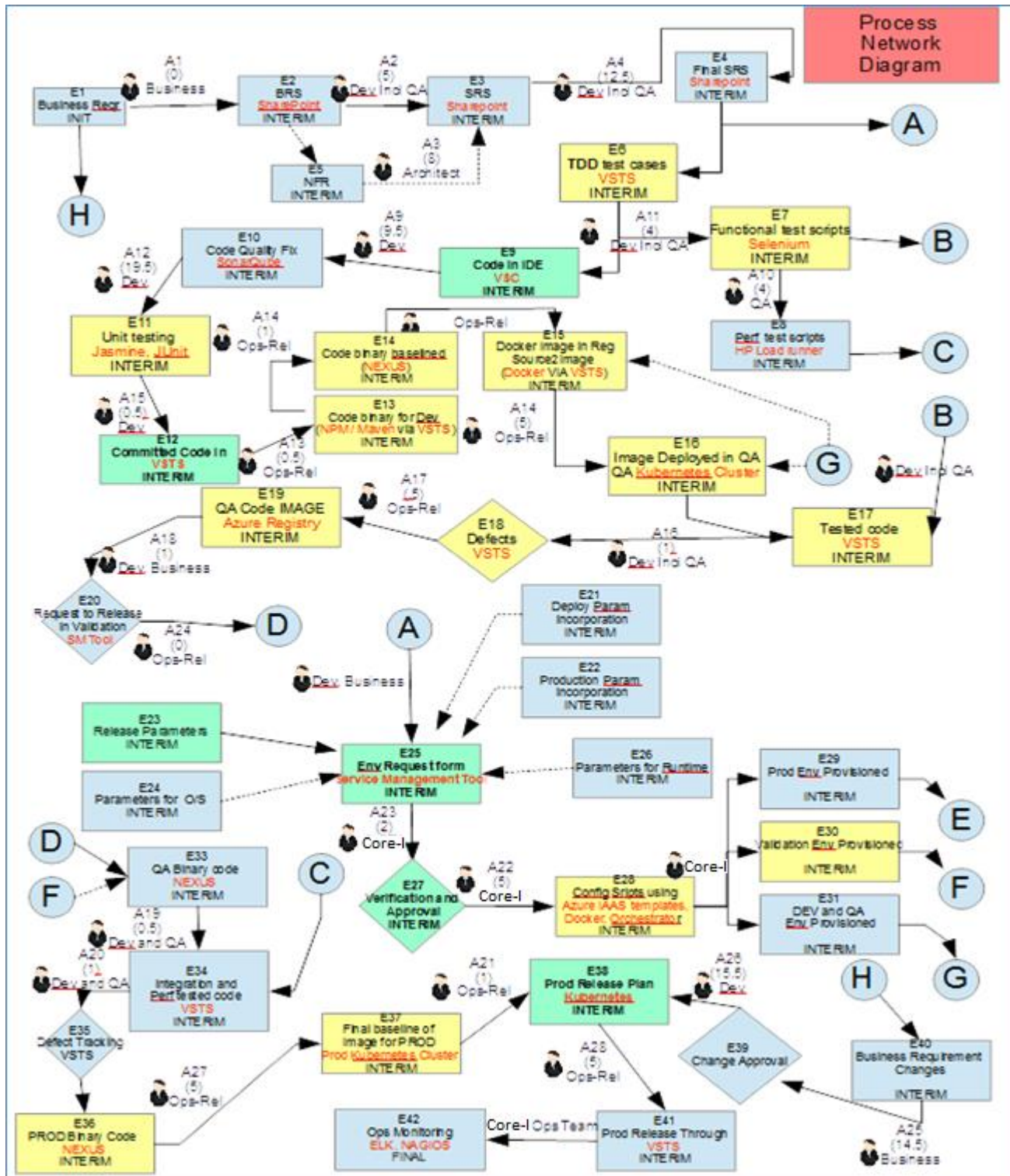
Figure 9: CI-CD for APIGEE

The developer will check in the source code in VSTS as the source code repository on Azure. On code commit a job will be triggered from VSTS, here used as an orchestrator, for code quality analysis using SonarQube. Jasmin and Karma will be used for the unit test automation, which will be automatically invoked through orchestrator. Once the unit testing is done the code will be build using Maven based Apigeetool. Apickli and JMeter to be used for run as jobs for functional and performance testing.

8. Revised Process Architecture with Timelines

Keeping the improvement opportunities identified in section no. 6 the revised process architecture is given as in the diagram below. The approach taken to come up with this suggestion is to reformulate the process with proper workflow tollgates, automation and tool support. In this way the critical path was further optimized. For the diagram below, following is the legend for elements used to categorize process flows, activity types etc.

Legend	Meaning
	DevOps Stories / Improvement over current Scenario
	Workflow / Decission making process
	Critical Path Activity
	Other activities over and above critical path
	Role who will be responsible for the action



Observations

1. Based on the revised critical path analysis the end to end deployment duration is optimized to **2.1 weeks****
2. In the picture above the boxes marked in Amber are the DevOps stories
3. The DevOps stories were identified vis-à-vis the gaps in baselined process
4. Each process is associates with a tool as per technology architecture; the tool name is given in Red
5. Role who will own responsibility is mentioned along with the activity
6. Developer PC/VDI will have IDE (Visual Studio Code; VSC) integrated with code quality checking tool (SonarQube), unit testing automation tool and ALM tool (VSTS) for seamless integration
7. Code versioning will be maintained along with branching in VSTS-GIT
8. On each code commit SonarQube and unit testing jobs will fire automatically from IDE integration
9. Apart from IDE the tools will be integrated in a CI/CD pipeline using VSTS
10. Once the code is ready for deployment the binary deployment file is stored in Nexus
11. VSTS-Docker will create an image of the binary in Nexus and register that in Docker Registry
12. This image will then be deployed through Kubernetes; the associate paradigm is termed as “Source to image”
13. Once the code is in production the Service/ Support Ops team takes over; such Ops transition (or, Ops-to-Dev) is not considered as part of this assessment

Note: The actual DevOps implementation success (including benefits envisaged in the target state), and validity and completeness of the target process and technology architectures mentioned herewith are subject to activities in the foundation sprint (as given in the below plan); based on this, there may be associated variation during the actual execution of the program.

9. DevOps Implementation Stories

Based on the analysis we have done as part of the QODE assessment following are the DevOps stories. For each story, the role responsible to perform the associated activity and the technology used, are given below in the table.

Devops Stories	Tool / Technology	Role
As a / an DevOps Architect I want to Define tools and technology stack across DevOps spectrum so that it provides clarity in terms of tools, products and frameworks that need to be considered from CI/CD perspective	MS-VISIO	DevOps Architect
As a / an DevOps Architect I want to Define code merging, versioning strategy so that necessary conventions are followed and code is managed properly	VSTS-GIT	DevOps Architect / Developer
As a / an Data Architect I want to Define development process so that it aligns to the DevOps process with respect to continuous integration/deployment for Data	VSTS	Developer / DevOps Architect
As a / an architect I want to Automate developer workstation build across different applications including DevOps so that developers do not have to get the necessary tools, APIs, plugins required for development	Docker, Kubernetes VSTS and Azure	DevOps Architect/CoreInfra
As a / an DevOps Architect I want to define monitoring KPIs for applications , infrastructure, networks along with other architects and change teams so that Appropriate monitoring can be established along with thresholds and incident management	Azure Log Analytics and VSTS	CoreInfra/DevOps Architect
As a / an DevOps Engineer I want to Setup and configure VSTS along with projects / boards so that issues and incidents can be tracked and resolved.	Visual Studio Code/VSTS	DevOps Engineer and Developer
As a / an DevOps Engineer I want to Setup & Configure VSTS (including Jobs, Pipelines, Plugins , Governance etc) so that continuous integration and deployment jobs can be orchestrated across different applications in scope and environments	Visual Studio/VSTS	DevOps Engineer/DevOps Architect
As a / an DevOps Engineer I want to Setup & Configure VSTS Jobs using Pipeline Scripts so that continuous integration and deployments can be run with desired results	Visual Studio/VSTS	DevOps Engineer
As a / an DevOps Engineer I want to Setup & Configure NEXUS so that all the binaries including are managed in one repository	NEXUS	DevOps Engineer / CoreInfra

As a / an DevOps Engineer I want to Setup & configure Kubernetes so that automated deployments and/or configuration changes can be pushed for respective applications across environments. As a / an DevOps Engineer I want to Integrate VSTS for automated CI/CD builds based on events and/or status changes on workflows so that automated builds outcomes are updated in VSTS ALM tool	Azure Kubernetes Service VSTS	DevOps Engineer/CoreInfra DevOps Engineer
As a / an DevOps Engineer I want to Setup and Configure VSTS-Git so that source code management can be established and align with development process across different teams	VSTS-GIT	DevOps Engineer
As a / an DevOps Engineer I want to Trigger CI/CD build when code is checked into VSTS-GIT (master/develop branch only) so that builds are run	VSTS-GIT, VSTS	DevOps Engineer
As a / an DevOps Engineer I want to Setup & Configure SonarQube so that development team can manage code quality, code coverage and any vulnerabilities which would have been introduced in respective applications	SonarQube	DevOps Engineer
As a / an DevOps Engineer I want to setup analysis plugins for Node, Angular, .Net, APIGEE so that projects teams can view various metrics provided by SonarQube	VSTS	DevOps Engineer
As a / an DevOps Engineer I want to Setup and configure defect management tool (e.g. TRAC etc) so that development team can efficiently manage code defects	VSTS	DevOps Engineer
As a / an DevOps Engineer I want to Setup and configure testing tools required to cover unit testing so that different metrics and thresholds can be defined and used by assurance and/or change teams to build fault tolerant applications	Jasmine, Karma, Junit, Mocha, Chai	DevOps Engineer / Developer
As a / an DevOps Engineer I want to Setup and configure various testing tools required to cover functional testing so that different metrics and thresholds can be defined and used by assurance and/or change teams to build fault tolerant applications	Selenium, Apickli	DevOps Engineer / QA team
As a / an DevOps Engineer I want to Setup and configure various testing tools required to cover integration testing so that different metrics and thresholds can be defined and used by assurance and/or change teams to build fault tolerant applications	Selenium	DevOps Engineer/ QA team
As a / an DevOps Engineer I want to Setup and configure various testing tools required to cover performance testing so that different metrics and thresholds can be defined and used by assurance and/or change teams to build fault tolerant applications	HP LoadRunner, VSTS performance testing suit	DevOps Engineer/ Performance Engineering team
As a / an DevOps Architect I want to define testing approach for various testing techniques and or tools so that different types of testing can be performed	VISIO	DevOps Architect

As a / an DevOps Engineer I want to setup & gather regression functional test results on a dashboard so that we verify that software which was previously developed and tested still performs the same way after it was changed or interfaced with other software	SonarQube, Azure Log Analytics	DevOps Architect
As a / an DevOps Engineer I want to setup and integrate test cases created which could be run from VSTS so that we get the necessary metrics around test cases from development etc	VSTS Orchestrator	DevOps Engineer, Developer
As a / an Developer I want to be able to pull code from source code management in my development IDE so that I am able to code	VSC , VSTS-GIT	Developer, DevOps Engineer
As a / an Developer I want to to be able to check in code in the source code repository so that the code can be built	VSTS-GIT	Developer, DevOps Engineer
As a / an Developer I want to to be able to create the build file so that the code can be built	NPM	Developer, DevOps Engineer
As a / an Developer I want to to run code quality tools on my code so that I am able to produce better quality code	SonarQube	Developer
As a / an Developer I want to to integrate my IDE with VSTS so that I can pull the stories in my IDE so that I am able to code faster with respect to the requirements	VSTS as ALM	Developer

10. Role –Technology - Tools Mapping

Following table maps the different activity to the different role relevant to the program execution, as well as the technology/ tool applicable. Activity numbers are given as cross-reference to the revised process architecture diagram above.

Activity No	Description	Role	Tool /Technology
A1,A2,A3	Preparation and baselining of SRS along with identification of NFR	Business Dev team	VSTS
A4	Coding using IDE	Dev Team	VSC(Visual Studio Code)
A9	Automatic code quality check and fix	Dev Team	SonarQube
A12	Unit testing automation	Dev Team	JUnit, Jasmine, Chai and Mocha
A10	Performance test automation	QA team	Loadrunner, Performance tester
A11	Functional and integration testing automation	QA team	Selenium, APICKLI, Cucumber.js, Grunt and Gulp
A12	Code commit to repository	Dev Team	VSTS-GIT
A13, A14	Build into Binary	Dev Team	NPM
A15, A16, A17, A18, A19	Source to Image	Dev team, Release-Ops Team	Docker, Kubernetes
A 20, A27	Defect management	VSTS	Dev and QA Team
A25	Change Management	VSTS	Business, Dev and QA team
A28	Metrics and Monitoring	Azure Log Analytics, Nagios	CoreInfra, OPS team

11. Key Metrics

Given below are the key DevOps metrics that may be established and tracked towards the success of the program execution:

Lead Time to Deployment (LTDD)	
KPI	Time To Market
Description / Significance / Need of the Metrics	<ul style="list-style-type: none"> • The time elapsed between development writing one new line of code and this new code being used by live users, in production
Data Required	<ul style="list-style-type: none"> • Time taken by the Development Team to Code • Time Taken to deploy the new code/ change in Production
Formula To Calculate	LTDD = Time taken by Dev team + Time taken by Ops team to deploy a change
Goal	The metric value should reduce
Frequency	Release End

Increase in No. Of Releases	
KPI	Time To Market
Description / Significance / Need of the Metrics	<p>This metric measure the increase in releases in a Month/Quarter.</p> <p>For Example – For an Agile project with 1 week Sprint cycle and 1 Month Release without DevOps, 1 Release a month will happen. But with DevOps it will be 4 Releases (hence, metric is 4x)</p>
Data Required	Total No. of releases in last Month/Quarter and Total No. of release in current Quarter
Formula To Calculate	<p>Increase in No. of Release = ((Total No. of Release in current Month/Quarter – Total No. of Release in previous Quarter) / Total No. of Release in previous Month/Quarter) * 100</p>

Goal	The metric value should increase
Frequency	Month End

Cross Skilled Employees	
KPI	Quality
Description / Significance / Need of the Metrics	This metric measures the No. of resources cross skilled in a Month/ Quarter. In DevOps it is important to measure the No. of resources cross skilled - on Development, QA and Ops skills, to enable self-dynamic teams
Data Required	Headcounts, Trainings list
Formula To Calculate	Total No. of resources trained on multiple areas of work
Goal	The metric value should increase
Frequency	Month End

No. of Outages Per Release	
KPI	Quality
Description / Significance / Need of the Metrics	This metric defines the total No. of outages per release in a given time period. Or the total no. of times the environment was not available per Release
Data Required	Track the total No. of outages / breakdowns
Formula To Calculate	No. of Outages Per Release
Goal	The metric value should decrease
Frequency	Release End

Defect Escape Rate	
KPI	Quality
Description / Significance / Need of the Metrics	This metric defines the total No. of outages per release in a given time period. Or the total no. of times the environment was not available per Release
Data Required	Track the total No. of outages / breakdowns

Formula To Calculate	No. of Outages Per Release
Goal	The metric value should decrease
Frequency	Release End

Defect Rejection	
KPI	Quality
Description / Significance / Need of the Metrics	Measures the percentage of all the invalid defects (ones which are rejected by the Dev Team)
Data Required	Total of defects raised and total No. of invalid / rejected defects
Formula To Calculate	Defect Rejection = (No. of defects rejected / Total No. of defects raised)*100
Goal	Red: 100%to 10% Amber: 9.9%to 5% Green: 4.9%to 0% The metric value should reduce
Frequency	Release End

Person Hours Spend Per Release	
KPI	Cost
Description / Significance / Need of the Metrics	To determine the average Person Hours spend on each release.
Data Required	Time / Effort spent on a Release
Formula To Calculate	Average Person Hours Spent = Time or Effort spent on a Release by team / Team Size
Goal	The metric value should reduce
Frequency	Release End

Thank You

Contact

For more information on this document, contact raghubir.bose@tcs.com and manas.shome@tcs.com

About Tata Consultancy Services (TCS)

Tata Consultancy Services is an IT services, consulting and business solutions organization that delivers real results to global business, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled infrastructure, engineering and assurance services. This is delivered through its unique Global Network Delivery Model™, recognized as the benchmark of excellence in software development. A part of the Tata Group, India's largest industrial conglomerate, TCS has a global footprint and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

For more information, visit us at www.tcs.com.

IT Services

Business Solutions

Outsourcing

All content / information present here is the exclusive property of Tata Consultancy Services Limited (TCS). The content / information contained here is correct at the time of publishing. No material from here may be copied, modified, reproduced, republished, uploaded, transmitted, posted or distributed in any form without prior written permission from TCS. Unauthorized use of the content / information appearing here may violate copyright, trademark and other applicable laws, and could result in criminal or civil penalties. **Copyright © 2018-19 Tata Consultancy Services Limited**