

IT Test Environment Management

Neha Bagmar

Abstract— Aim of this study is to assess the current state of Test Environment, Customer needs & requirements and what solution we have to manage and maintain test environments as an next generation test environment management, which can help customers overcome business risks and test in a production-like environment, thus resulting in releasing a superior software product.

1 INTRODUCTION

Over the past several years, the tools and techniques available to software testing have become plenty and more advanced and cultured. Also, the scope of software testing in today's test environments is increasing rapidly, greatly due to the growth and distribution of the ways people can access core systems, applications and architectures across multiple new interfaces, endpoints and access points.

IT test environment is a complex foundation of infra, software, application layers upon which all of software testing is based. It must have the specified hardware, software components properly installed with the right versions, configuration and settings and be stable and usable as expected, in a timely manner.

Despite the careful control under which testing takes place, setting up the test environment and configuration is chaotic as test environments are not solely managed by a test environment team. Also not having a dedicated test environment or a test environment management team — can cause significant risk in quality and productivity.

In addition to the above, we need to consider if each environment is a better fit for any platform including the cloud. Virtual environments, with their scalability benefits and connections to new technology and methodologies, still have complexity due to infrastructure, network, virtualization tools, platform integration and backup/recovery.

In this study, we discussed the current state of Test Environment, Next Generation Test Environment, Customer needs & requirements and what solution we have to manage and maintain test environments, which can help customers overcome business risks and test in a production-like environment, thus resulting in releasing a superior software product.

-
- Neha Bagmar is currently working as Senior Consultant in Tech Mahindra Ltd, Pune, India PH-00917888042524. E-mail: nb00471328@techmahindra.com.

2 CURRENT STATE OF IT TEST ENVIRONMENTS

As per the EMA (Enterprise Management Associate) Test Environment Management Survey results, only 4% of large enterprises have fully integrated Test Environments within their application development process.

Test Environments currently exist in a rather unstructured, partially integrated and loosely coupled manner. As software releases move through various phases such as system integration testing, non-functional testing, user acceptance testing and into pre-production, the complexity of the environments and associated costs with supporting these environments grow exponentially.

Test Environments needs to be setup involving all the practices or functions involved in the SLDC with different models and methodologies (Waterfall, Agile etc.) providing various handoffs to make the environment available in a consistent manner and with higher degree of accuracy. Even missing one piece will make the delivery, development and testing team wait for hours to days to deliver and so often the time pressure results in poor quality.

It is therefore important that test environments and its resources are managed appropriately to not only reduce errors caused by misconfiguration but also to improve the shared use and utilization of these environments. Conflict occurs when multiple teams have to share one testing environment, and there are constant changes to the environment and disagreement over who gets to use it, creating inefficiencies. In many cases, test environments are created by utilizing the organization's multiple resources and infrastructure. Once the testing is finished, it may be unrequired again right away but might be required soon and hence it is just kept up and running, therefore driving up costs. There are many further examples, but here's the gist: Currently the use of test environments is not efficient, collaborative, effective and cost-effectiveness is not taken into account.

3 CUSTOMER NEEDS/REQUIREMENTS

With the changing scenarios and evolving test environments, test environment management strategy has become an indispensable need for the organization's today. Leveraging test environment management approach in a software development process can be significant in improving productivity, reducing costs, and accelerating the releases. Organizations

are adopting the shift-left approach and CI-CD-CT (Continuous Integration, Deployment & Testing) strategy to enhance an application's quality, decrease the possibility of failure after deployment and reduce the number of defects stemming out due to inconsistencies.

Latest Tools, Technology, Methodology along with Automation are the key factors that enable organizations to address their customer's needs.

- Zero Tolerance with respect to conflicts in Test Environment while Testing
- Risk minimization through better compliance with regulatory requirement
- Consistent and accurate environments through automated environment provisioning
- Self-Healing and proactive resolution of quality related issues
- Optimum utilization of infrastructure assets
- Continuous availability of test environments during testing
- Elimination of errors due to manual interventions and delays stemming from dependencies
- Bringing all the asset information together for effective analysis and decision-making.

4 BENEFITS OF A IT TEST ENVIRONMENT MANAGEMENT APPROACH

Mature Test environment management strategy and approach yields benefits that includes:

- 4.1 Speed up the time to market:** The emergence of more sophisticated application solutions, combined with cheaper storage and alternative infrastructure models such as virtualization and cloud technologies, have led IT landscapes becoming increasingly integrated and complex. More than ever, there is a need to maintain control of these application ecosystems whilst maximizing the availability and utilization of these investments, in order to speed up their overall product time-to-market.
- 4.2 Faster test environment provisioning and de-provisioning support services:** On demand provisioning facilitates continuous deployment and provides easier access to production like environments during development and testing stages of the delivery pipeline. It is important to assess the current environments used for running in BAU across the software delivery lifecycle. Getting the inventory of current state and decommissioning of environments, which is not in use, can reduce cost & enable visualization of the current state at any given point in time.
- 4.3 Increased Environment availability:** Even unavailability of one component will make the delivery, development

and testing team wait for hours to days to deliver and often the time pressure results in poor quality. Test Environment management strategy needs to be setup involving all the practices or functions involved providing various handoffs, continuous monitoring and health checks to make the environment available in a consistent manner and with higher degree of accuracy.

4.4 Operational Uplift across DevOps & Test: Automated tests that run continuously and automated deployment that deploy code continuously every time a new commit is published surely enhance quality and make your DevOps more efficient. These tests are performed on increasingly complex environments as the application moves through the pipeline. By reducing dependency on the manual testers, automation in test environment management can be effectuated by TEM tools for CI/CD, workload automation, and configuration management. With continuous testing as an integral part of your DevOps, you get quality code your team is proud of and your customers are happy with.

4.5 End-to-end environment management - Single ownership: A mature TEM team produces cost savings. Allowing development teams to create their own testing environments results in waste. Developers may not understand the cost of environments and may create active environments that remain unused for long periods of time. This racks up costs, especially when these environments are hosted in the cloud. A centralized team have a much better understanding of these costs and will collect metrics in order to determine which environments cost more than they're worth.

5 NEXT GENERATION TEST ENVIRONMENT MANAGEMENT

Key factors emerging in test environment management are creating opportunities to manage and maintain the test environments in a more holistic and streamlined manner in future.

5.1 Cloud/VMs/Container (Technology): It takes several days or even weeks to setup OS and configure all the software required to deploy the application on the server on premise manually but with the advent of VM's(virtual machines) and Cloud the problem is solved to a large extent which has let to tremendous growth in the scale by which the environments are provisioned on which applications are deployed for QA/Testing to test before it is released to production. In addition to that Container technology tools have the ability to quickly spin up an environment(Container) for a Developer or tester which makes it easy for them to come back with quick analysis, code fix or testing the same before pushing into the testing pipeline(SIT/UAT/Pre-Prod). There are tools available,

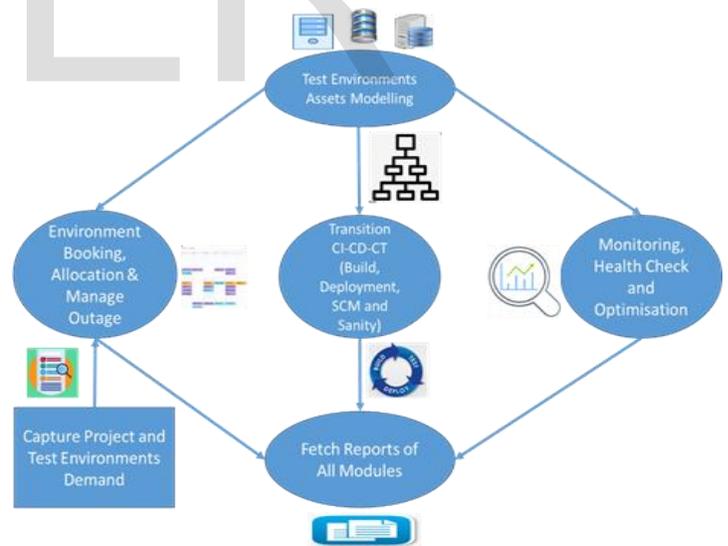
which can help in dynamic creation and configuration of virtual dev or test environments and suits well with virtualization software/ tools.

- 5.2 **Assess before you Create or Virtualize:** Before initiating the environment creation or virtualization process, it is imperative to understand the requirements thoroughly. Assess the available baselines, footprints used in the past, outcome and the lesson learned. Visualize the plan, create the breakdown structure, check if any available resources can be re utilized and analyze the process that can be automated. When the test environment management strategy is devised keeping the consumer needs in mind, the test environments can be provisioned in speedy manner with right configuration with the help of available tools and automation. Thorough documentation of test environment resource allocation, configurations and incidents helps in effective management of test environments. Such documentation should be complemented by the four W's namely Who (refers to the person or team requesting or initiating the alteration), What (description of the alteration, request or incident), When (time or period of the alteration, request or incident) and Why (the reason of alteration, request or incident).
- 5.3 **Single End to End Ownership:** Assigning complete ownership of the test environment to a dedicated team makes it the single point of contact for all test related activities and makes it accountable for test environment management, thus removing problems associated with accountability or poor management and irregular maintenance, which generally happens when ownership is shared across different teams.
- 5.4 **Regular Evaluation of Test Environments:** This helps to keep the test environment as realistic and precise as possible. After each major test cycle, evaluate what is working, what is not, the resources that are under- or over-utilized etc. and make required changes accordingly to imitate end-user environments better. It is equally important to consider the requirements of the testing teams and alter environment resource allocation accordingly.
- 5.5 **Standardize and Automate:** Most of the activities across test environment lifecycle can be automated by using a single tool or through multiple tools. Nowadays enormous number of tools are available for provisioning, deployments, configuration, build, testing etc. To improve the turnaround time for test environment provisioning, deployments, organizations must spend money on automation strategies & platforms like DevOps, Cloud technology, Containerization and Service Virtualization. For example by automating the CI-CD process on cloud, the organization can fulfill ad-hoc requests and ensure 'on -time' deployment and testing with lesser cost and effort.

6 ONE STOP SOLUTION

Tech Mahindra's ELMS (Environment Lifecycle Management System) is a platform, which provides an enterprise-scale Test Environment Lifecycle Management that allows large organizations to achieve positive benefits in terms maintaining high-availability, eliminating redundancies, lowering costs, and reducing time-to-market. Tech Mahindra has strong expertise and provides solutions, services, processes, methodologies, and tools to manage test environment efficiently, effectively, and consistently to allow QA and organization to achieve their objective.

This SaaS-based solution allows test environment managers to rapidly audit pre-existing test environments and provides a centralized platform for managers to view requests for environment allocation for testing. Managers will be able to track their assets while they're in use, record changes, and estimate their impact in real time. In this scenario, no unassisted human intelligence will be able to efficiently provision, update, create, or delete testing environments. In order to prevent chaos, automation software is required. ELMS replaces an ad-hoc collection of spreadsheets and calendars by controlling the assignment of test environments to test teams. Developers, testers and other stakeholders have a highly visible and understandable portal to view the allocation of test environments, as well as release calendars.



ELMS's ecosystem is a fully configurable and easily integratable solution that comes with out of the box "enterprise management" functions that support IT & Test Environment Management, Release Management, IT Operations Management, Configuration Management & Service Management. ELMS integrates effortlessly with an organization's other test and DevOps tools and assets. Toolsets such as Rally, Cucumber, JUnit, Apache Maven, and Jenkins are easily integrated to enable application delivery teams to strengthen continuous inte-

gration/continuous delivery (CI/CD) pipelines using DevOps methodologies.

The benefits of using ELMS tool can greatly reduce the challenges that organizations are currently facing and fulfill the objectives of development and testing teams. For those already using a TEM tool, the survey found that faster provisioning was the most common benefit (89%), which in turn led to a corresponding reduction in the number of test environments that were needed, and an improvement in the quality of applications being developed and tested.

There are also financial benefits for businesses as well. Regardless of the stage of maturity of the TEM tool in use, they begin paying for themselves straightaway. For the average enterprise in the survey, with around 76 production releases per year, the predicted TEM cost savings are about \$1.4 million, primarily across the following TEM categories:

1. Enabling faster provisioning of test environments
2. Reducing the number of test environments needed
3. Reducing development and test time for each release
4. Reducing Test Environment Management costs
5. Improving application quality

No matter what the volume of overall production releases is, the payback on TEM investment will occur almost immediately, and if these savings are viewed as cost avoidance, ROI will be realized within a couple of months. Though the exact numbers in savings and ROI will vary from business to business, it is clear that by implementing the tool, businesses will start to see the positive financial effects almost instantly, alongside the improvements they make to the testing process itself.

With the number of test environments increasing – as well as customers' expectations – businesses need to invest in the best management tools to make release deliveries faster and of higher quality. The EMA (Enterprise Management Association) survey found that the respondents' top DevOps priority for 2019 is faster software delivery, and while TEM tools will not drive this exclusively on their own, they can certainly make a world of difference to not just this but testing as a whole.

CONCLUSION

It is important to setup required and supporting functions to make continuous infrastructure a reality to provision environments dynamically for continuous testing. While we cannot sprint to get to the end point it is good to start assessing what we have & setup a clear road map on what we want to achieve to be in the journey of DevOps with respect to continuous infrastructure to enable continuous testing.

ACKNOWLEDGMENT

The authors wish to thank Shubhadeep Chakraborty. This

work was done under his guidance.

REFERENCES

- [1] <https://www.softwaretestingnews.co.uk/test-environment-management-how-save-time-effort-and-money-with-tem-tools/>
- [2] <http://go.plutora.com/ema-test-environment-survey-report>
- [3] <https://www.enov8.com/blog/test-environment-management-use-case/>
- [4] <https://www.cigniti.com/blog/test-environment-management-strategy/>