TEST ENVIRONMENT MANAGEMENT

A QUICK GUIDE TO DIRECT ENVIRONMENT SUCCESS

ANURAG SHARMA

ITSM PRESS

Buy Online At

Test Environment Management: A Quick Guide to Direct Environment Success

Test Environment Management: A Quick Guide to Direct Environment Success

ISBN Paperback: 9781912651269
ISBN eBook: 9781912651276

First Published in Great Britain by ITSM Shop Ltd

ITSM Press is the wholly owned publishing imprint of ITSM Shop Ltd.

Whilst all care and attention has been taken to compose this publication, neither the publisher, nor the author, nor the editor can accept any liability for any loss or damage caused by any possible errors and/or incompleteness in this publication.

Other ITSM Press Publications:

A Concise Introduction to ISO 14001:2015
A Concise Introduction to ISO 9001:2015
A Concise Introduction to ISO 45001:2018
Service Management: It’s All About the People

ITSM Shop Ltd
95 Duxford Road, Whittlesford, Cambs, CB22 4NJ, UK
+44 (0) 3333 445 286
customerservices@itsmshop.co.uk
www.itsmshop.co.uk/

Trademark Notice:

ITIL® is a registered trademark of AXELOS Limited

ISO® and IEC® are registered trademarks of ISO and IEC

COBIT® is a registered trademark of ISACA

Table of Contents

ABOUT THIS BOOK

In 2013, I was assigned a task to implement ‘Environment Management as a Service’ from scratch for a big UK Retailer and that time, except my service management and technical experience, I had nothing to deliver this part of the requirement. One late evening, one of my colleagues asked if I could recommend any books on Test Environment Management. At the time, we hadn’t found any book which even helps on fundamentals on Test Environment Management. From that evening I decided to document all parts of the journey and support the IT community on Test Environment Management.

What this book is about

This book is specifically about understanding Test Environment, implementation and then running Environment Management as a service. This book provides fundamentals of Test Environments including definitions, types and terminologies and framework to implement Environment as a service. This book is based on real-world experience in implementation Test Environment Management as a service.

It also addresses organisation issues of running Test Environment Management as service, such as stakeholder management, to realise real value for the organisation. This book uses ITIL®, LEAN and COBIT® as fundamental blocks. Having said that, if you are unfamiliar with these management practices, don’t be concerned, because terminologies are well covered in this book.

What this book is not about

This book aims to provide a very basic aspect of Test Environment Management. It’s a set of good practices. It’s not a set of rules for Test Environment management; it’s non-prescriptive and leaving the flexibility and freedom to organisations going to implement Test Environment Management as function.

Why read this book

The principal audience for this book is made up of Test Environment Management practitioners and IT professionals. The ultimate objective of this book is to provide processes and framework to Environments Practitioners, which can help them to deliver Best in Class Environment Experience to their Customers.

This book will help them to

- Understand how to provision, standardise and manage Test Environments.
- Understand how to create, implement and operate Test Environments as a service, and also to realise benefits of managing Test Environments.
- Provide measurement metrics, key performance indicator and reporting.

INTRODUCTION

Today’s world and customer experience both are changing at the speed of thoughts. The key focus area of every single organisation is to provide best in class customer experience, which is delivered using technology. To remain competitive in the market, every organisation transforming digitally, delivering innovation. It’s a fact that the world that consumers and companies operate in is changing in order to make sense of a fast-changing context. But this time Digital transformation is more of a mindset.

Its radical systematic change, which is going to influence every sphere of our lives. It’s not only a technical change where organizations are changing process and infrastructure but this also a cultural and behavioural change.

Faster time to market, the speed of implementation, quality, service availability, lower cost of ownership is becoming critical elements of transformation. In an ideal world, Digital transformation is the thumb rule followed by each industry, no matter, small scale or large. IT or non-IT, newbie, experienced or legends.

But, at ground level, there are industries which started thinking and down to their traditional waterfall practices. Old ways of working, system complexity and pressure to survive in new era creating absolute chaos in their world. It’s like an IT disease which can be cure by management practices like DevOps and agile. These management practices require some time to get mature. It is not an overnight revelation. It requires a change of culture, technologies and way of working. Until the cure is applied, there is need of pain reliever so that transformation journey can start walking. Test Environment Management is the pain relief that alleviates the symptoms.

In the current marketplace, there is frequent ask to rollout Test Environment Management as a service driven by IT Service Management to bring agility and flexibility. Test Environment Management services actually help in walking before an organization start running on management practices like agile and DevOps.

Test Environment Management is becoming the key function in IT. Test Environment Management is a critical factor in success for any transformational journey. It is an element which not only helps to deliver the right product at the right time, but drives cultural change, too. The target objective of Test Environment Management function is to create a foundation for continuous delivery and continuous integration and implement agility and flexibility throughout the journey.

Now the question is why Test Environment, Not Production?

From the dawn of information age, Production Environment Management is getting mature, but Non-Production or Test Environment is always underestimated. The key reason to underestimate Test Environment because business doesn’t want to spend on this and mindsets which consider Test Environment Management an overhead. But it’s a fact; a poorly manage test environments is the common denominator when there is a delay in testing or compromising on test coverage.

In most of the cases, around 30 to 40% of Testers time occupied with test environments issues. In typical IT Environment, the test environment is underutilized, and there is no governance around it. Considering the need for quality, agility and continuous delivery, It’s essential for an organization to define the test environment strategy and establish a centralised function to maintain Test Environments.

Since Test Environment function is underestimated from the beginning, so there is lack of standards, process and practices available to maintain Test Environment. Despite most companies spending between 30%-45% of their IT spend on Non-Production (Training, Development & Testing) the art of “**Test Environment Management**” (or TEM) still appears to be an area of significant confusion and neglect. In the hope to educate and raise the bar on TEM, here in this book our focus is on Test Environment Management. In coming sections, we will understand fundamentals of Test Environments and Test Environment Management Services, and how an organization can implement test environment management as a function.

SECTION I:
FUNDAMENTALS OF TEST ENVIRONMENTS

What is Test Environment?

An IT Environment is a collection of Hardware, Software and Data working together to support business activities in Production, Testing or Development. An Environment can be single machine supporting either a single system or a collection of an integrated system.

Test Environment is the set-up of software, hardware, required data and network configured for users to execute test scenarios. An Environment user can be a testing team, development team, business team or operations team.

Test Environment is a close-enough replica of an actual production environment where testing would happen. Its purpose is to facilitated production like behaviour to test software. Test Environments are typically built with specific functionalities suitable for performing validation on a developed solution. They act as a test bed in verifying software or application design. These functionalities are included as part of testable requirements in order to validate the new solution does not impact the organisation production environment.

Type of Environments

Every organisation have their test environment, and the need for various environment depends on cost, the number of projects or programme going to use environment and their demand capacity management portfolio. Typically within environment management, you will find environments can be categorised into two different ways

- **Service point of view**

This categorisation is based on physical location of the environment

  - **On-premise environments**

    On-premise environments are the collection of Hardware, Software and Data working together to support business activities, created within the organization. These environments require a greater degree of control and maintenance and consume a good amount of IT budget of an organization.

  - **Cloud-based environments**

    It’s Environment where you just need to provision your application and run through an automated test suite and then tear down the environment. It works on an on-demand model with the ultimate aim to reduce cost and improve agility. The nature of test environments is inherently temporary, and you only need to pay for service until you use environment. Nowadays, cloud-based environments are the first preference for any organisation because there is no way that’s more effective at reducing your overall test environment spend than migrating to cloud test environments and embracing testing and infrastructure automation.

  - **Hybrid Environments**

    Its combination of on-premise and cloud-based environments. Sometimes organisation have to provision environments in hybrid due to the limitation of available cloud and old legacy applications which cannot be installed on the cloud.

- **User point of view**

This categorisation is based on environment provisioned and allocated for a purpose

  - **Development Environment**

    This is the environment where design and build activities are carried out, unit tested before been promoted to an actual test environment. This environment is basically provisioned with all sets of required development tools.

  - **Sandbox Environment**

    It’s a playground to try various development techniques in order to gain confidence in the build. Authorised users can develop their application knowledge, and self-test their codes or various design to ensure it works before moving/building the specific component in the development environment.

  - **QA/Testing Environment**

    This is the initial environment in which system or application under test should be promoted to after development work is complete. It is provisioned based on testable requirement it may be integrated or standalone depend on testing phases. This Environment can be further categorised based on testing phase, e.g. System Integration Test Environment(SIT), System Test Environment(ST), Performance test environment(PT) etc.

- **Pre Production Environment**

  An environment which provisioned as almost identical to a production environment with the same infrastructure, same application, same configuration, fully integrated systems and application to ensure new solution are regression tested to validate co-existence with existing system landscape before promoting the solution to a production environment. The only difference between pre-production environment and production is that it should not transmit real transactions or information to end users.

- **Staging Environment**

  A stage or staging environment is an environment for testing that exactly resembles the production environment. In other words, it’s a complete but independent copy of the production environment, including the database. Staging provides a true basis for QA testing because it precisely reproduces what is in production.

- **Production Environment**

  This is the environment where software and other products/service are actually put into operation for their intended uses by end users.

- **Disaster Recovery Environment**

  An environment dedicated to run a disaster recovery test in ensuring that the system/ecosystem landscape does have the ability to sustain potential or system failures in an event where there is possible fallout.

Environments Lifecycle

Environment lifecycle defines the various stages/phases in the life of environment. Test Environment life cycle describes the period of time over which an environment is developed based on demand, brought to the organisation and eventually removed from the organization. The cycle is broken into six phases:

- Demand

- Design
- Build
- Support
- Reuse/Repurpose
- Retire/Decommission

The below diagram shows typical phases/stages performed in the life of environments

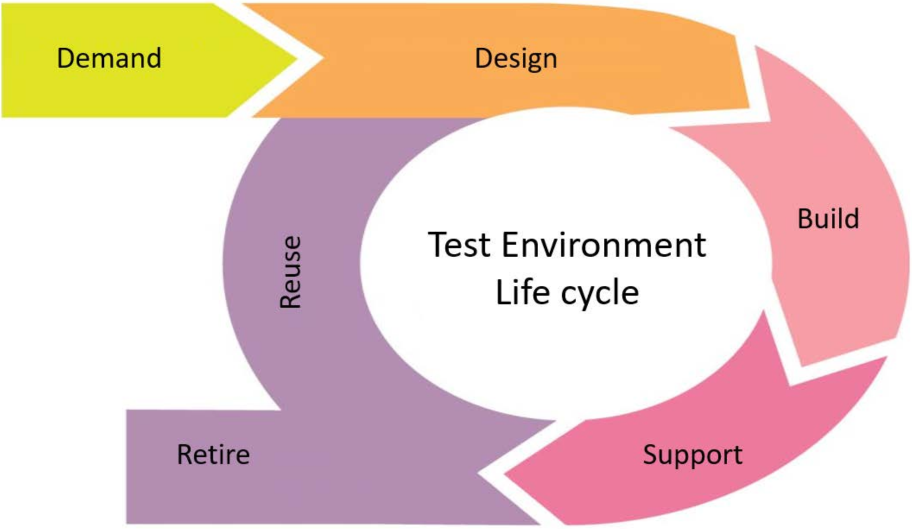

Figure 1.1 – Test Environment Life cycle

1. **Demand**

The Demand is the first phase which helps in gaining an understanding of the demand for test environments in the particular project or programme situation then determines and details the test-specific requirements, such as the number of test environments, scope of the data sets, configuration, integration, etc. Demand initially come from the overarching Projects, and early insight will ensure demand helps in proactively planning, coordinating and providing fit for purpose environments in a timely fashion.

2. **Design**

The design is phase about turning demand into a design of the test environment; This helps to determine the solutions for provisioning test environments based on what is in scope. This may include building new environments, either from the ground up or from templates, and/or sharing existing environments within the client landscape. This phase consists of determining tool requirements and set-up, to manage artefacts (code, packages, configuration) for re-use, compliance and audit purposes.

3. **Build**

The build phase is about actually building environment based on design. It's about provisioning infrastructure, application configuration and integrated into the right environment at the right time. This also includes managing the provision of test data within application environments, keeping code base up-to-date (aligned with production) etc.

4. **Support**

Its operational phase; once the environments have been provisioned, the test teams will require support on many levels from the environment team, including Incident/request/problem/access management, change delivery (configuration, builds, firewall rules), job execution (batch, file transfers), release and deployment support etc.

5. **Reuse/Repurpose**

This is the phase driven by demand, once environment release by project it can be allocated to other project based on the requirement. The environment can be repurposed subject to the requirement. For instance, if initially environment build for system testing and now same project or another project required it for Performance testing than Environment can be reuse and repurpose based on the requirement, in this case, increasing capacity will help in repurposing environment.

6. **Retire/Decommissioning**

An Environment can be decommissioned if future demand does not warrant continued investment. This helps in optimizing resources and saving cost.

Understanding the Business need of Test Environment

Test Environments increases the overall software delivery efficiency and ensures robust software quality. Using Test Environment QA can perform all required testing which can provide enough confidence to Production support and Business team. Below are key benefits

1. **Faster Time-to-Market:**

Test Environment greatly helps to reduce the time-to-market of an application by driving agility and flexibility. This also reduces overall approval cycle in taking new software into BAU Environment for Production support and business approval. It is of no surprise that, while the initial investment may be on the higher side, test environment can save the organisation money.

2. **Testing Efficiency Improvement and Higher test coverage**:

Testing takes up a significant portion of the overall application development lifecycle. This goes to show that even the slightest improvement of the overall efficiency can make an enormous difference to the overall timeframe of the project. Through the implementation of production like test environments, more number of tests to the production like behaviour and a sharp drop in the amount of time required to run tests. It contributes to a higher quality of work, thereby decreasing the necessity for fixing glitches after release and reduces project costs.

3. **Earlier Detection of Defects**:

Since Test Environment provides a production like an interface to the tester, so it’s easy to detect any defect earlier. This helps increase the overall development speed while ensuring correct functionality across areas. The earlier a defect is identified, the more cost-effective it is to fix the glitch.