

A Framework for Managing large performance QA teams in a dynamic Business and Technology environment

Rajesh S

Independent Validation Solutions

Infosys Technology Limited, Bangalore (NASDAQ: INFY)

Introduction

With the ever increasing adoption of the web, Businesses are constantly being challenged to ensure that their online applications are able to provide a good quality experience to their customers. This creates a demand on various internal teams like Application architects, developers, Capacity Management and not least, the Performance testing groups to work together in achieving the application performance requirements. Another constant change is Technology which is a great enabler, but forces Performance teams to constantly keep up with the knowledge, features and problems of new technologies leading to quite a few challenges.

This article defines a framework for effective Performance QA teams in a dynamic Business and Technological environment, identifying and categorizing the key components which add value to the Business. It covers the typical challenges encountered and lists learnings and best practices from experiences in working for a leading International Financial services organization. It can be applied to the Performance QA groups of any large organization supporting web applications.

1. Business and Technology Environment

Business Environment: In an intensely competitive environment, Business organizations are constantly designing new, innovative and better solutions to attract customers. This results in

- A constantly increasing number of applications and components being hosted on the application servers
- Increasing usage of these applications resulting in a greater demand on computing resources
- The constant effort to provide better, faster and more reliable web applications

Technology Environment: Technology is constantly changing all the time.

Applications need to constantly upgrade technology, migrate to different platforms,

and adapt new technology. This encompasses both, Hardware and Software. This creates it's own Performance testing challenges in terms of

- Experiencing, identifying and resolving performance issues unique to new technologies
- Large scale testing to support movement to new technologies

The impact of these external drivers is all the more significant on Performance QA groups which support multiple large applications using a large team.

The below framework provides a structured way to develop effective performance testing practices to address the challenges of large teams. The Framework covers 3 pillars of Performance testing- Process, Infrastructure and People, and looks at how key components under each help to address challenges and add value to Performance testing.

2. Process Infrastructure People Framework

Process, Infrastructure and People are the pillars which support the structure of Performance testing. Under this framework, we will go into the details of each of these and identify the factors to be considered and Best practices that bring value to Performance testing

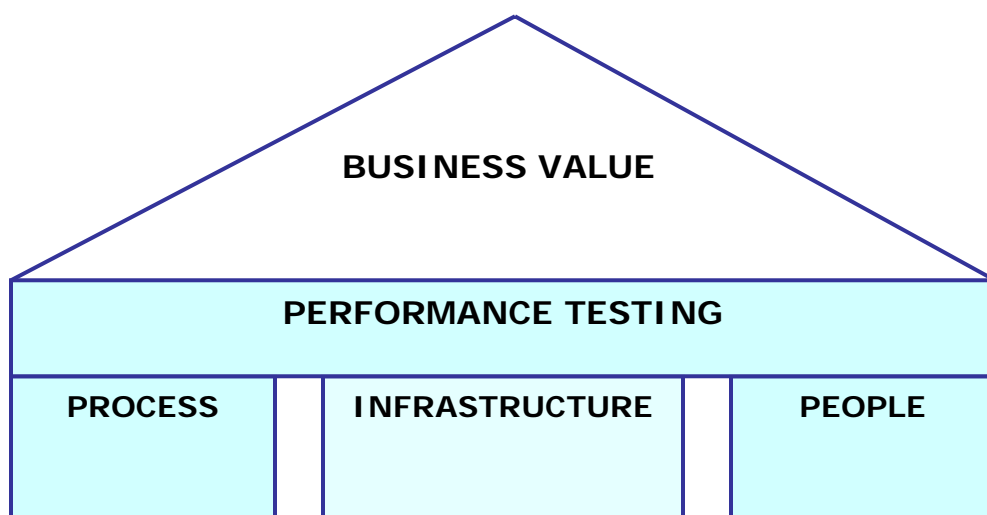


Fig 1 The Process Infrastructure People framework

3. Process

Process here is meant to include the set of activities defined and executed by Performance QA teams to achieve their Goals. This includes Performance test methodology, Test management and Quality activities.

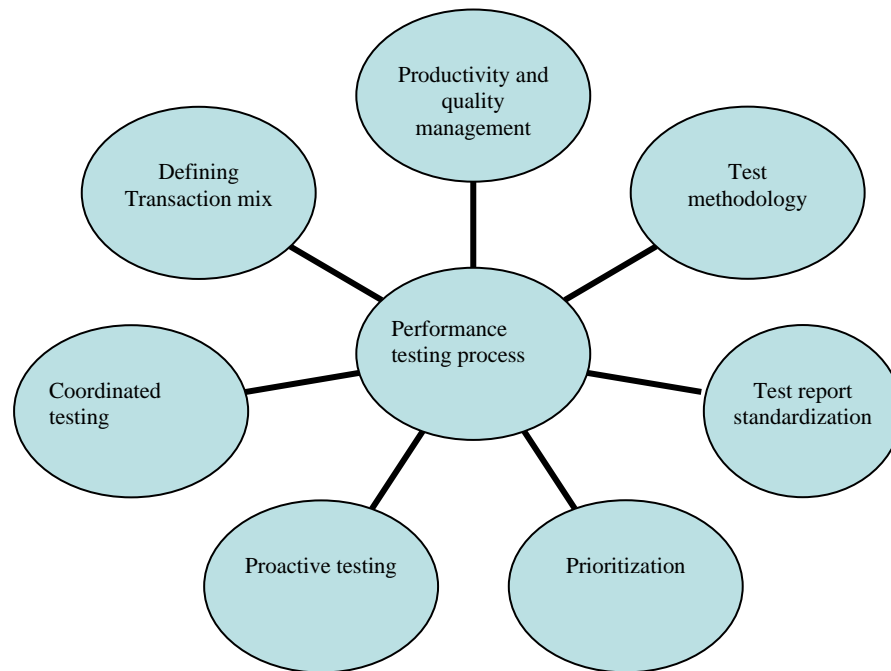


Fig 2 Process factors

3.1. ***Test methodology standardization***

Performance QA teams in large organizations, by virtue of having to support multiple applications, have to maintain a balance between meeting specific needs of various application groups and the need to have a standard test methodology and test reporting. What usually works is a modular approach. In case of test methodology,

Define and strictly adhere to a minimum set of tests required for performance QA signoff guided by the Technical objectives of the group. Eg.

- 1 hr Capacity test at a steady 3 times the average (1 hr period) peak load
- 24 hr Longevity test at 2 times the average daily load
- Stress test to find the maximum TPS

Additional tests can be executed for specific applications only. Eg.

- Failover test
- Overload test
- Volume test

3.2. Test Reporting

Define a standard format which will be applicable to most applications. Add on reports wherever required for the specific application

Defining a standard methodology and reporting format (apart from the few exceptions) helps in ensuring consistency of deliverables, better quality and makes it easy to cross-train resources across projects. A standard report format could include the below data and it's graphs for eg.

Load Runner reports	Server metrics
Total transactions (Pass/Fail)	CPU % vs time/load
TPS vs time/load	Total, Used, Free Memory vs time
Total TPS vs time/load	Context switches vs time
RT per transaction vs Time/load	Garbage Collection vs time

3.3. Prioritize where possible

Shorter QA cycles leave the Performance Engineer with very less time to complete the cycle of tests. This leaves one needing to prioritize tests and execute whatever is possible in the order of priority for each build. A typical plan could be-

	3X Capacity	Longevity	Stress	Failover	Volume
Priority	1	1	2	3	3
Build 1	√	√	√	√	√
Build 2	√	Shorter duration test	√	X	X
Build 3	√	Shorter duration test	√	X	X
Prod. Candidate	√	√	√	√	√

3.4. Proactive Performance Testing

Performance testing is done a few weeks prior to production install and leaves very little time to resolve issues. Some of these bugs could be very fundamental in nature

which requires an OS, Application Server, Database patch from the vendor and cannot be addressed in a short time leading to a Business impact.

The below techniques provide advance insight into the application performance prior to full fledged testing

Profiling: Application profiling can be done during the Development phase itself in the development environment. Profiling an application allows a developer to glean valuable metrics

- Memory usage of given objects
- Execution times of specific methods
- VM Heap usage and potential Memory leaks

Predictive Performance testing techniques: Predictive Performance techniques offer a way of determining application performance much before full fledged tests are run on the Performance test environment. By running specific tests even in the development environment, application performance and bottlenecks can be predicted with the help of Profiler output, Development server statistics, test results and production hardware data.

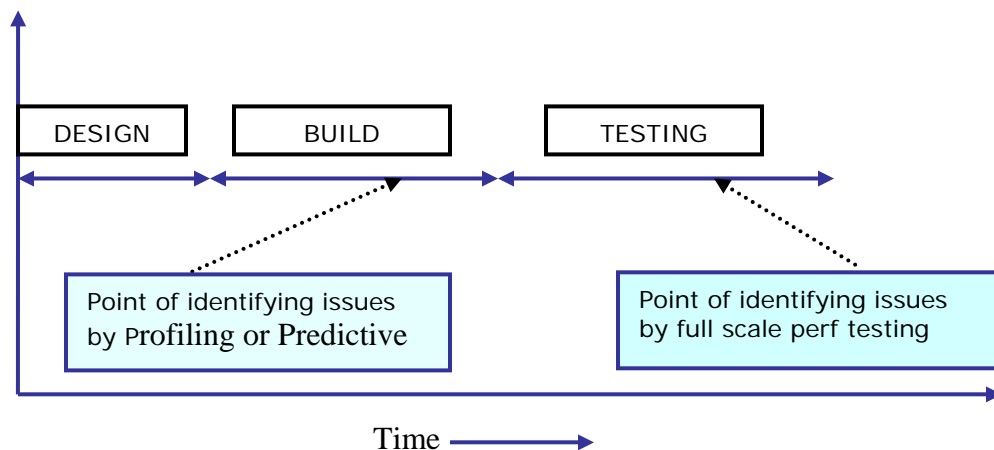



Fig 3 Proactive techniques help to identify Performance issues earlier

3.5. Coordinated testing of applications

All organizations test their web applications in isolation to identify, analyze and resolve performance problems. But in a large organization with ever increasing number of applications, coordinated simultaneous testing of these applications is ignored at times. This is more so where multiple applications/groups share common

infrastructure. Coordinated testing helps to identify problems which otherwise remain hidden. Resource Contention issues, poor Database performance, Application Server Capacity issues are typical problems uncovered by coordinated testing.


 Run coordinated tests involving all applications sharing common hardware or components before every major release.

3.6. Defining Transaction Mix

_Defining the right Transaction mix is a key element of Performance testing because it is effectively the Performance requirements. There are 2 broad scenarios

Maintenance releases: Application URLs are mostly known in advance. Analyze the Production server access logs and also take the help of Analysts to define the mix in terms of

- TPS
- User Arrival Rate
- Navigation patterns
- Wait time
- Functional(and Data) breakdown

 The Transaction mix and production load should also be obtained as frequently as possible- typically once for each release- rather than continue to use outdated data.

New Applications: Assume a new application with the below projections from Business


Peak # of trades/day – 100,000

Business hours- 10:00 AM to 4 :00 PM – 6 hrs

Each trade involves 5 calls to the application

TPS target for testing= $100,000 * 5 / (6 * 3600) = 24$ TPS

- Obtain application URLs from Developers and Analysts
- Coordinate with Capacity planning to define the required Infrastructure
- Coordinate with Server owners to plan the deployment of applications
- Apply suitable headroom over the TPS target while testing applications

 Consider Business Estimates of future application usage while planning

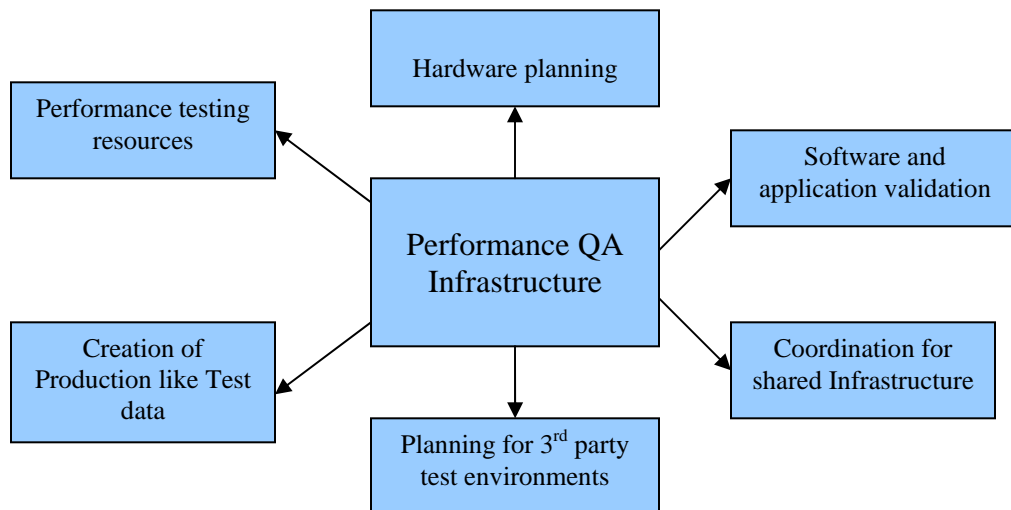
3.7. Productivity/Quality Management

Like any other function, the Performance QA team also needs to have in place processes and methods to demonstrate continuous Quality and Productivity improvement

- Standardize the test process and reporting. This has already been described in an earlier section
- Automate wherever possible- Environment validation, Server monitoring, Report generation, Parsing of log files etc. Choose between Market software and in-house developed tools
- Storage of results: Test results, at-least the important ones need to be stored for later retrieval and analysis.
 - Options of- storing on a LAN, use of Document Management platforms like EDMS, Share point etc.
 - Easy way of sharing the test results with other stakeholders like Business partners, Project Manager, Developers, Architect, Analyst etc.


4. Infrastructure

Performance testing is an effort to unearth issues by simulating production like conditions to the extent possible. This means that the performance environment too has to be a replica of the production environment. In the below section, we will see how Infrastructure plays a pivotal role in the success of Performance QA groups



4.1. Hardware-Infrastructure

Hardware Infrastructure in the Performance lab should ideally replicate the Production environment to be able to conclusively identify performance issues. Large organizations usually ensure this, but for various reasons- usually cost related, the performance environment hardware could be a scaled down version of the production environment. In this case, the performance test results will be different from what can be expected in production. This has to be factored while interpreting the results.

 Plan sufficient lead time to acquire new hardware. Do not wait until the application performance starts degrading.

4.2. Software Infrastructure

Define a process to ensure that the application software versions and settings are similar to that in production. Wrong results are often obtained because of discrepancies between the 2 environments.

Eg.

Production Settings	Performance Environment Settings
app1.maxconnections=15 app1.minconnections=2 app1.maxQueueLen=100 app1.poolSize=10 app1.poolMax=50	<i>app1.maxconnections=5</i> app1.minconnections=2 app1.maxQueueLen=100 app1.poolSize=10 <i>app1.poolMax=20</i>

Some best practices to ensure software/build accuracy

- Define a process for installing new builds in the performance environment with sufficient controls and limit write/install access to a select group
- Develop simple automated tools which verify the application configurations and other settings and compare against master settings.
- Have checkpoints in the testing process where the tester needs to verify the builds installed using the validation tool before beginning the tests

4.3. Common Infrastructure:

Every large application uses common infrastructure like Databases, services, authentication servers etc. Some of them could be owned by external groups which have their own performance teams.

Coordinate testing and application monitoring with these external groups so that

- The components are tested under realistic production scenarios rather than in isolation.
- The stress tests of each do not affect other application tests
- The individual applications down-time or maintenance cycles do not affect performance tests

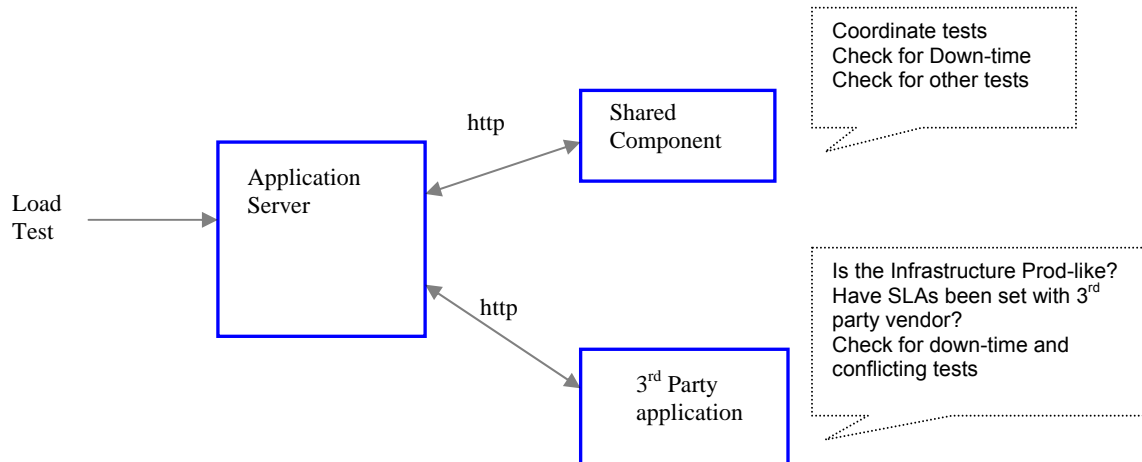


Fig 5 Application view including shared services and 3rd party servers

4.4. 3rd party applications/software

A lot of applications use 3rd party software or feeds from 3rd party applications. In such cases, the key point is to not just go by the SLA with the provider, but ensure that the application is tested in an integrated manner. Some of the requirements of the 3rd party provider which need to be planned well in advance are:

- Production like test environment of the 3rd party application
- Availability and access to the 3rd party test environment during the testing cycle
- Support and Issue resolution of 3rd party application

4.5. Test data

Apart from having realistic test data to carry out all the scenarios, Performance testing also requires that the data volumes in the test environment are similar to what is expected in production. The following table illustrates the impact of having

different data in the Production and Performance environments using hypothetical examples

Production Environment	Performance Environment
<p><i>2000 unique Funds</i></p> <ul style="list-style-type: none"> • Response time 10 sec • Memory allocation/request 50 MB • Observed Out of Memory errors under load 	<p><i>50 unique Funds</i></p> <ul style="list-style-type: none"> • Response time < 1 sec • Memory allocation/request 1 MB • No severe Memory errors observed
<p><i>10 different account types</i></p> <ul style="list-style-type: none"> • Few accounts creation involve different components • Response time 5 sec • CPU utilization 60% 	<p><i>All accounts are of the same type</i></p> <ul style="list-style-type: none"> • Simple account type • Response time < 1 sec • CPU utilization 10%

4.6. Performance Testing Resources

Plan and equip the team with

- Required Performance Testing, Monitoring and Analysis tools
- Premium level of support for these tools wherever possible

5. People

People are invariably the prime asset of any knowledge or skill based organization. Performance being a specialized testing group, the people factor is especially important. Some of the key aspects of the People factor are covered below

5.1. Technical Skills

There are 2 aspects to this, Performance testing skills and Technical skills. To stay ahead of the technology changes, requires the Engineer to have a good understanding of technology- Hardware, networks, Operating System, Application servers; trends and developments in the technology field. Management needs to focus on

- Effective and regular in-house training on latest technologies
- Have Performance Engineers participate in trainings along with Developers
- Strong relationships with Development teams

- Regular External training on latest trends and developments in Performance testing.

5.2. Business Domain knowledge

Even though Performance QA is a specialized field, a good understanding of the Business domain is very helpful to the Performance engineer in-

- Understanding and building scenarios effectively and designing realistic performance tests
- Communicating with other groups like Project Managers, Business partners and Analysts.
- Develop a specialization in a particular Business domain.

5.3. Coordination and Communication

Communication and Coordination is very important in any team, but some unique features in Performance QA teams are

Notification of tests: Performance Engineers should send out a notification when they run load tests so that their tests are not inadvertently affected by installs, application restarts etc. A sample format could be

Test Notification
 Test Name: test 1
 Applications impacted: App1, App2
 Servers impacted: Websvr1, AppSvr1,
 Time: Start: 12 Jan 6:00 PM- 13 Jan 6:00 AM
 Load: 10 TPS
 Contact details: abc@xyz.com

Reservation System: Have a (preferably Online) resource reservation system which Engineers can use to book resources for Performance testing

Maintenance Calendar: The server owners should clearly communicate and adhere to a maintenance calendar. Sample calendar

Database Maintenance Calendar
 Data synch- Daily between 6:00-6:30 PM
 Weekly maintenance- Friday 8:00 PM- 10:00 PM

Monthly maintenance- 1st Sunday 8:00 AM- 5:00 PM

5.4. Knowledge Management

The people intensive nature and high degree of specialization make KM very important for Performance testing teams

KM best practices include

- Efficient documentation – Application documents, Test plans, known issues, Common problems
- Regular Knowledge sharing sessions within the team
- Cross training of people on different technologies and applications

6. Learnings

It is a given that we need to “Run in order to stay in the same place” as far as Performance testing is concerned. To sum up, we can make the following observations.

- Without proactive planning and Management , Performance QA teams will not continue to deliver Business value for long
- Organizations need to invest in required Infrastructure, ensure proactive capacity planning and efficiently manage their resources to generate optimum value
- They will need to define clear processes and have suitable mechanisms to ensure their implementation
- Lastly, and possibly the most important of all- they need to aggressively invest in people with Business, Technical and Performance skills and support them to continuously enhance their capabilities

References:

1. Infosys Technologies Limited Sources, www.infosys.com

Speaker profile(s)

Name:

Education: 1994: B.E. Mechanical (Bangalore University) ; 1998 PGDM (IIM, Lucknow)

Total Experience: 10 years

Software Testing Experience: 4 Years

Experience:

I have been working with Infosys Technologies Ltd. For the past 10 years. I joined Infosys after completing my PGDM course.

I worked on Software development, maintenance and platform migration projects using Microsoft Technologies like .Net framework, Com +, Visual Basic and ASP during the next 6 years.

Since the past 4 years, I have been working on testing projects covering Functional testing, Test Automation and Performance testing for different clients in Infosys.

For the last 2 years, I have been working on and managing large Performance testing projects for a large US based Financial services organization.

I am part of the Independent Validation Solutions group at Infosys

Interests: (Optional) Software quality assurance, software development process, Performance Testing and Analysis, Project Management