

VIEW POINT

Improve Software Quality and Accelerate Delivery by
Adopting Service Virtualization Wisely



Quality

- Manoj Aggarwal, Delivery Manager

Context

As business applications moved from tightly coupled solutions to service-based distributed architecture, it opened a new range of business and technical possibilities. Services-based architecture enables IT solution providers to expose their business solutions to consumers in a loosely coupled on-call consumer / provider model. Within the organization landscape, it provides enterprise application teams a mechanism to deploy reusable business functions as services for internal consumption.

Though services have improved the software delivery process, there are some key lessons to be learnt. Business applications depend on SOA based services for critical business functions. This makes developers and testers realize that services behavior knowledge and service availability are the key factors that will define the success and quality of the solutions developed by leveraging these services. To address these dependencies, developers started 'mocking and stubbing' of the service for development and unit testing. Testing teams depended heavily

on the proactive services environment blocking to ensure availability of services and data for testing. Both these approaches lead to increased effort and timelines for development and testing. However, with software development moving to shorter cycles in Agile and DevOps world and services becoming increasingly complex, these service dependencies became a key obstacle in faster delivery, especially in situations involving parallel development.

Drawing from the hardware virtualization experience, virtualization of services evolved into an industry standard practice that allowed both developers and testers with a realistic approach to handle services dependencies. Developers managed to integrate code with the virtualized services for 'live like' code integration in development environment, enabling these to test and capture potential code issues early. Application testers benefited from the reduced environment and test data dependency result in faster execution and effective leveraging of the automated testing scripts.

When To Go for Service Virtualization

As awareness of the service virtualization and its associated benefits percolated to IT teams, both providers and consumers of services started to deploy virtualization solutions at a brisk pace. A fair number of implementations delivered significant business benefits ranging from 20-30% infra cost reduction, 10-15% reduction in delivery timelines, and 15-20% reduction in defect slippage. A significant majority of virtualization solutions continued to struggle with justification of investments in the development and maintenance of these solutions.

At Infosys, we believe a holistic analysis of the service consumption pattern such as a service user base, data requirement, and cost of consumption are some parameters that help decide the best fit for service virtualization solution deployment. One more factor that will increasingly define virtualization deployment is the software development methodology followed by IT teams.





Service virtualization works best for service providers in a number of instances. This includes the following:

- Services are yet to be developed and consumers need services for their development / testing to meet rollout timelines
- Services consumer base is very high and availability of test data / environment is a challenge
- Maintaining live services environment is expensive
- Services behavior is fairly stable and test data / scenarios can be covered with the limited data set

Service virtualization works best for service consumers in a few instances. This includes the following:

- Services provider cannot provide services for the development / testing team
- Provider does not provide a live / virtualized service economically
- Availability of services and data in test environment is a challenge
- Service that is frequently needed by development teams due to the volatile nature of the consuming application

It is not recommended for both consumer and provider in the following instances:



Services are readily available and inexpensive to maintain



Services need live data for testing and development






Services behavior changes very frequently



Services have complex business logic with volatile test data

Service Virtualization Approach

Infosys recommend a three-step process for service virtualization as depicted below:

		
Services Identification & Solution Definition	Build & Deploy	Maintain Services (Heal Services)
Analyze architecture Landscape	Virtual Service Design Documentation	Continuous improvements and enhancements to support new requirements
Identify constrained systems	Build Virtual Services	Republish new versions and refresh test data
Formulate high level Strategy	Test Virtual Services to ensure completeness & correctness	Decommission end of life virtualized services
Perform tool fitment analysis	Deploy & Publish Virtual Endpoints	
Feasibility analysis		

Conclusion

Service Virtualization has evolved into a key component of software development lifecycle. Effective deployment of service virtualization can bring immense benefits with improved quality of the code and faster delivery. However, all services need not to be virtualized and project teams should plan switch to live services at an appropriate juncture to ensure applications are tested in the 'LIVE' environment and a reasonable balance is maintained between the real and virtual world of services.

For more information, contact askus@infosys.com



© 2016 Infosys Limited, Bangalore, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.