Chapter XV
Service Evolution and Maintainability

Bijoy Majumdar
Infosys Technologies Limited, India

Krishnendu Kunti
Infosys Technologies Limited, India

Mohit Chawla
Infosys Technologies Limited, India

Terance Bernard Dias
Infosys Technologies Limited, India

Lipika Sahoo
Infosys Technologies Limited, India

ABSTRACT

Change is the only constant, and this concept holds good for services too. Service maintenance is the most tedious and longest phase of service lifecycle. The more complex the service, the more difficult it is to maintain it. Service maintenance and service evolution mandate a series of best practices and selective models to apply for better execution and administration. In this chapter we detail the challenges faced in service evolution management and the key activities involved, and their role in service-oriented architecture (SOA) quality.

INTRODUCTION

Organizations develop service components with rigor supporting different standards and sufficing the interoperability requirements to collaborate with all the stakeholders in the business transaction. But
in the face of business competition and changing requirements, the challenge is not only in developing these assets but also maintaining their quality as they evolve with the changing business requirements. Like other objects in the IT environment, services too are subject to aging and may fail to keep up with the ever-changing requirements. But services quality and flexibility mandate strict vigilance and maintainability of software services. Thus, to realize the benefits of service-oriented architecture (SOA) completely, service evolution and service maintainability become key underlying components of service quality practice.

Service evolution can be defined as the process of change that the service might undergo to meet direct or indirect change in business requirements. Service evolution starts after the development, at the deployment stage of the service, and ends at the service decommission stage. During this period, the service might be changed to include enhancements, bug fixes, and so forth. Managing this change in service is called service maintainability. In a broad perspective, service maintainability encapsulates versioning, monitoring, change management, and reporting. Services, being self-contained, independent components, need slightly different approaches in each of these activities, as explained in the following sections.

**CHALLENGES IN SERVICE EVOLUTION**

It is important that during the service development lifecycle, the phases of service evolution are also considered so that stakeholders are involved in acts beyond the focus of just developing some business functionality to adapt quickly to the unexpected change in requirement. To effectively address the challenges that one faces during service evolution, one needs to focus on the explicit support around the services and their environment for a sustainable quality in the future. The challenges faced in different aspects of service evolution are explained below. Also the approaches to address these different challenges are different.

**Service Model and Support**

The service model helps to set the uniform mindset across all stakeholders in the design (Zimmermann, Krogdahl, & Gee, 2004), development, and maintenance phase of service lifecycle. The modeling approach does help to create a framework to work with in boundaries and enhances the system. Each existing model caters to a specific problem in service development. Some examples of models are:

- Model driven architecture (MDA) streamlines the process of integration
- Web services resource framework (WSRF) defines a generic and open framework for modeling and accessing stateful resources using Web services
- Business process execution language (BPEL) defines the orchestration of services across domains
- Domain-specific languages fill the gap between general purpose languages and particular application domains by providing a language with notation and concepts geared to the domain
- Information technology infrastructure library (ITIL) is a set of concepts, techniques, and frameworks to manage software infrastructure and its operations
Related Content

A Semi-Automatic Approach to Composite Web Services Discovery, Description and Invocation  
[www.igi-global.com/article/semi-automatic-approach-composite-web/3051?camid=4v1a](www.igi-global.com/article/semi-automatic-approach-composite-web/3051?camid=4v1a)

Technical Architecture of Enabling Body of Knowledge System for Effective Learning and Information Dissemination  
[www.igi-global.com/article/technical-architecture-of-enabling-body-of-knowledge-system-for-effective-learning-and-information-dissemination/90265?camid=4v1a](www.igi-global.com/article/technical-architecture-of-enabling-body-of-knowledge-system-for-effective-learning-and-information-dissemination/90265?camid=4v1a)

Construction of Traceability System by Using Simple and Handy Type RFID Reader  
[www.igi-global.com/chapter/construction-traceability-system-using-simple/53272?camid=4v1a](www.igi-global.com/chapter/construction-traceability-system-using-simple/53272?camid=4v1a)

Authentication Techniques for UDDI Registries  
[www.igi-global.com/chapter/authentication-techniques-uddi-registries/26911?camid=4v1a](www.igi-global.com/chapter/authentication-techniques-uddi-registries/26911?camid=4v1a)