

Chapter 32

Experiences with Cloud Technology to Realize Software Testing Factories

Alan W. Brown

IBM Rational and University of Surrey, UK

ABSTRACT

In enterprise software delivery, the pursuit of software quality takes place in the context of a fundamental paradox: balancing the flexibility that drives speed of delivery with the rigor required to verify that what is being delivered is complete, correct, and appropriate for its intended use. One common approach to address this concern is to create “software testing factories” with the aim of increasing testing efficiency by standardizing and speeding up delivery of testing services. To achieve this balance, software testing factories are turning to cloud-based infrastructures as an essential delivery approach. Cloud technology exhibits characteristics that make adoption of software testing factories particularly attractive: elasticity of resources, ease of deployment, and flexible pricing. In this chapter, the author examines the role and structure of software testing factories and their realization using cloud technology, illustrates those concepts using real world examples, and concludes with some observations and a discussion on future directions.

INTRODUCTION

Enterprise organizations are looking to enterprise software delivery as a core capability to bring efficiency and stability in delivery of services to the business, and the driving force for innovation and differentiation for new services or new products to the market. To achieve this, they are enhancing their own enterprise software delivery organization with systems integrators and technol-

ogy partners to create centers of excellence and capability centers specialized in delivering value to the business. We call this a “software factory” approach.

At the same time, enterprise organizations are demanding rapid innovation and evolution of existing enterprise software solutions to support deployment to new platforms, to address new market needs, and to overcome competitive threats. While standardization using software

DOI: 10.4018/978-1-4666-6539-2.ch032

factories helps reduce cost and improve predictability of enterprise software delivery, agile ways to approach innovation are essential for solution differentiation and to enhance the value delivered to the enterprise's clients. Significant invention, adaptation, and flexibility is essential in research and development activities to drive forward the solutions brought to market, and to improve the services made available. Yet often these aspects are viewed as being in conflict with the needs for efficiency and control.

This clash between rigor and flexibility is often most clearly seen in the area of software testing. It is in the pursuit of software quality that we frequently see challenges in the balance between speed of delivery and rigor in verifying that what is being delivered is complete, correct, and appropriate for its intended use. In creating "software testing factories," many organizations are looking to increase testing efficiency at the same time that they improve software quality and accelerate their time to market.

To achieve this balance they are not only adopting new processes and techniques, they are also deploying more flexible technology infrastructures, notably based on cloud technologies. Cloud technology exhibits characteristics that make adoption of software testing factories particularly attractive: elasticity of resources, ease of deployment, and flexible pricing. As a result, many organizations are making investments in cloud technology to support their implementation of software testing factories.

In this chapter we examine the role and structure of software testing factories and their realization using cloud technology. We first explore the idea of the "software factory" and the characteristics of the software factory approach as it applies to software testing. We then discuss cloud technologies and examine how they help realize a software testing factory. We illustrate those concepts using real world examples, and conclude with some observations and directions.

BACKGROUND

Over 40 years ago, the original NATO reports (Naur & Randell, 1968; Randell & Buxton, 1969) focused attention on some of the core elements of an industrialized approach to enterprise software delivery; increasing productivity and quality of software delivery in the face of severe skills shortages, the importance of standardized processes to improve predictability, and the role of measurement and metrics in gaining insight into project progress and for optimizing development and delivery activities.¹ In the succeeding years there was a great deal of attention turned toward these themes, particularly in understanding how different forms of software process improvement could raise the quality and consistency of software delivery (Hunter & Thayer, 2001; Humphrey, 1991). This resulted in "spiral" and "iterative" models of software development (Boehm, 1988; Krutchen, 2002), and measured improvement schemes such as the Capability Maturity Model (CMM) (CMU, 2010).

More recent work on the industrialization of software has focused attention towards automation and verification aspects of software production (Clements & Northrop, 2001). From one perspective, component-based design techniques and reuse libraries were seen to be the central elements to create catalogs of parts for assembly of systems from pre-developed pieces (Brown, 2000). While from another perspective the key to automation was the role of more formal modeling languages amenable to improved analysis techniques from which working systems could be generated (Greenfield et al., 2004).

In fact, many existing texts have discussed the scope of enterprise software, and the many challenges faced in their delivery into production, and their on-going maintenance. Broadly speaking, we see that delivering quality enterprise software is challenging as a result of many factors. But it is useful to highlight three of these for particular attention:

25 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/experiences-with-cloud-technology-to-realize-software-testing-factories/119878

Related Content

BIG Data: An Enabler in Developing Business Models in Cloud Computing Environments

K. Hariharanath (2019). *Global Virtual Enterprises in Cloud Computing Environments* (pp. 171-200).

www.irma-international.org/chapter/big-data/219555

Examining Software-Defined Networking for Cloud-Based IoT Systems

Garima Singh (2018). *Examining Cloud Computing Technologies Through the Internet of Things* (pp. 192-215).

www.irma-international.org/chapter/examining-software-defined-networking-for-cloud-based-iot-systems/191839

Advanced Brain Tumor Detection System

Monica S. Kumar, Swathi K. Bhatand Vaishali R. Thakare (2020). *International Journal of Fog Computing* (pp. 31-45).

www.irma-international.org/article/advanced-brain-tumor-detection-system/266475

Co-Design Flow for Embedded Systems (MPSoC): Hybrid Model and Performance Estimation

Kamel Smiriand Nourhen Fourati (2018). *Soft-Computing-Based Nonlinear Control Systems Design* (pp. 44-62).

www.irma-international.org/chapter/co-design-flow-for-embedded-systems-mpsoc/197485

Feedback-Based Resource Utilization for Smart Home Automation in Fog Assistance IoT-Based Cloud

Basetty Mallikarjuna (2020). *International Journal of Fog Computing* (pp. 41-63).

www.irma-international.org/article/feedback-based-resource-utilization-for-smart-home-automation-in-fog-assistance-iot-based-cloud/245709