Organizational Impact on Software Development of eServices Techniques

Maurizio Cavallari[1,2](✉), Francesco Tornieri[1], and Marco De Marco[2]

[1] Università Cattolica del Sacro Cuore, Largo Gemelli 1, 20123 Milan, Italy
{maurizio.cavallari,francesco.tornieri}@unicatt.it
[2] Università Telematica Internazionale UniNettuno,
Corso Vittorio Emanuele, 39, Rome, Italy
marco.demarco@uninettunouniversity.net

Abstract. New Software Development techniques with respect to cloud computing and eServices had modified IS architectures which were well established and consolidated in the past. The new methodologies of (software, micro/e) "Services" has pushed towards the adoption of software development organization independent from traditional tiered-architecture with the result of reducing both scale-up and down times as well as interruption times due to migration to different platforms. The eServices development organization relays on MicroServices architecture so that it is decomposing the legacy architecture in micro-components, each one with an independent life-cycle but interconnected and correlated, i.e. eServices. Each eService is hosted within a single container which has a proper software lifecycle and with minimal set of executable operating system libraries. The analysis goes into details about the structure and the development of eServices with MicroServices architecture. The paper discusses the new technological tendencies under the lens of an Organizational approach.

Keywords: Organization · Software development · eServices · Information system · Architectures · Software life-cycle · MicroServices · Scalability

1 Introduction

In the IT world in recent years, there has been a strong push in terms of innovation aimed at creating technology that would simplify, where possible, and speed up the deployment of new, or the maintenance of existing, software.

Two trends have emerged:

- Micro Servicing
- DevOps.

These trends enable the development of the application in a distributed way and they also make parts of the business functionality services available to developers [1].

They help to overcome many of the development limits making it easier to work collaboratively and to write/correct code independently [2].

S. Za et al. (Eds.): IESS 2017, LNBIP 279, pp. 64–75, 2017.
DOI: 10.1007/978-3-319-56925-3_6

Emerging field of DevOps borrows practices from software engineering to tackle complexity. The need for automation to improve scalability and testability while simultaneously reducing the operators' work, has always present in the software development research field.

The knowledge/skills/abilities needed for both software development and operator, encounter perfectly the four perspectives of DevOps, i.e. collaboration culture, automation, measurement, and sharing.

Consequently, when we think about software developmental we can say that decomposing the applications business logic into independent functional components has many advantages [3]:

- allows a development team to be small and highly focused
- can choose whichever technology best suits their specific use
- deliveries are composed by a set of small individual services modules
- could then be deployable on different servers and infrastructure.

2 Ops, Dev and DevOps