

Build, Configuration, Integration and Testing Tools for Large Software Projects: ETICS

Marc-Eliañ Bégin², Guillermo Diez-Andino Sancho², Alberto Di Meglio²,
Enrico Ferro¹, Elisabetta Ronchieri^{1,*}, Matteo Selmi¹, and Marian Żurek²

¹ INFN, Italy

² CERN, Switzerland

`elisabetta.ronchieri@cnafe.infn.it`

Abstract. Software development within geographically dispersed and multi-institutional projects faces challenges in the domain of validation and quality assurance of software products. Experience in such projects, especially in the area of Grid computing, has shown that the lack of appropriate tools and procedures may cause high overall development costs and delays in the deployment, development and maintenance of the software. In this paper, we introduce ETICS, an integrated infrastructure for the automated configuration, build and testing of Grid and distributed software. The goal of the infrastructure is to provide a service for software projects by integrating well-established procedures, tools and resources in a coherent framework and adapting them to the special needs of distributed projects. A set of versatile tools and best-practice guidelines for quality assurance implementation are also provided to maximize the project's chances of delivering reliable and interoperable software.

1 Introduction

Several large-scale open-source software projects have to deal with the need to organize complex software life cycle management infrastructures and processes in order to guarantee required levels of quality, interoperability and maintainability. Often these projects have to face resource, skill, time and budget constraints that may lead to the risk of releasing software difficult to deploy, maintain, understand and integrate with other applications. Fixed-term research projects such as DILIGENT [1], [2] and EGEE [3], [4] have to focus on developing software of increasing functionality through their lifetime, but cannot always guarantee that the software will still be accessible, maintainable and documented after the conclusion of their mandate. In such distributed development environments, ensuring that components developed by different developers, in different languages, on different platforms and with non homogeneous tools and processes is often a daunting challenge that may lead to software difficult to manage. Furthermore permanent projects such as QUATTOR [5] suffer from the lack of well-defined build procedures and this makes it difficult for other institutes to

* Corresponding author.

adopt them. Under the pressure of short deadlines and large requirement sets, project managers may have to face the decision of cutting testing and quality assurance verifications, which can be a cause of delaying the release or impairing the usability of the software because of the excessive number of undetected problems. Even when functional tests are performed, the nature itself of complex middleware, such as that developed for the computational Grid¹, render costly the provision of adequate hardware and network resources. When middleware and applications are deployed on tests or certification testbeds a lot of time is usually spent trying to make middleware suites and applications to interoperate due to the different configuration assumptions and different versions of common libraries.

In this work, we introduce ETICS, an integrated infrastructure for the automated build, configuration, integration and testing (BCIT) of software [6], specifying its requirements and architecture. ETICS aims to support such research and development initiatives by integrating existing procedures, tools and resources in a coherent infrastructure, additionally providing an intuitive access point through a Web portal and a professionally-managed, multi-platform capability based on Grid technologies [7]. Consequently, developers and software managers will be able to integrate their code, libraries and application, validate the code against standard guidelines, run extensive automated tests and benchmarks, produce reports and improve the overall quality of the software. ETICS goal is not to develop new software but to adapt and integrate already existing capabilities, chiefly open source, providing other research projects with the possibility of focusing their efforts in their specific research field and to avoid wasting time and resources in such required, but expensive, activity. Nevertheless, ETICS also adds any missing features, such as a consistent schema in order to configure, build and test software projects with different characteristics (e.g., platforms, development languages).

This paper is organized as follows. Section 2 describes the requirements of the system, whilst Section 3 documents the related work, explaining what ETICS can add to the state of the art. Section 4 details the certification process. Section 5 describes the architecture. Section 6 presents a rigorous definition of the basic concepts, whilst Section 7 reports useful user's operations supported in the ETICS infrastructure. Few use cases are provided in Section 8. Section 9 reports the conclusion and future activities.

2 Requirements for the Design of the BCIT Framework

In this section, we describe the requirements for the design of the build, configuration, integration and testing framework for distributed software. First of all, it is fundamental to establish an international and well-managed capability for software configuration, integration, testing and benchmarking for the scientific community (for what concerns the software configuration in a complex testbed,

¹ Computational Grid provides a set of services that allow a widely distributed collection of resources to be tied together into a computing framework.