Science of Computer Programming 121 (2016) 55–74

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Knowledge discovery in software teams by means of evolutionary visual software analytics

Antonio González-Torres [a], Francisco J. García-Peñalvo [b], Roberto Therón-Sánchez [b], Ricardo Colomo-Palacios [c,*]

[a] School of Engineering, ULACIT, Costa Rica
[b] Department of Computer Sciences, University of Salamanca, Spain
[c] Faculty of Computer Sciences, Østfold University College, Norway

A R T I C L E I N F O

Article history:
Received 12 March 2015
Received in revised form 21 September 2015
Accepted 30 September 2015
Available online 21 October 2015

Keywords:
Visual Analytics
Software Evolution
Software Evolution Visualization
Visual Software Analytics
Evolutionary Visual Software Analytics

A B S T R A C T

The day-to-day management of human resources that occurs during the development and maintenance process of software systems is a responsibility of project leads and managers, who usually perform such a task empirically. Moreover, rotation and distributed software development affect the establishment of long-term relationships between project managers and software projects, as well as between project managers and companies. It is also common for project leads and managers to face decision-making on human resources without the necessary prior knowledge. In this context, the application of visual analytics to software evolution supports software project leads and managers using analysis methods and a shared knowledge space for decision-making by means of visualization and interaction techniques. This approach offers the possibility of determining which programmer has led a project or contributed more to the development and maintenance of a software system in terms of revisions. Moreover, this approach helps to elucidate both the software items[1] that have been changed in common by a group of programmers and who has changed what software items. With this information, software project leads and managers can make decisions regarding task assignment to developers and staff substitutions due to unexpected situations or staff turnover. Consequently, this research is aimed at supporting software practitioners in tasks related to human resources management through the application of Visual Analytics to Software Evolution.

1. Introduction

The current tendency to perform the development of software systems in a distributed form by means of the so called distributed software development (DSD) must be considered when evaluating software processes [1–3]. DSD scenarios normally impede fluid communication and the construction of a shared knowledge space, which facilitates the understanding of the state of the project [4,5]. It must be said that, without the construction of the aforementioned shared knowledge space, it is not possible to efficiently manage resources nor assign tasks appropriately, and neither is it possible to ade-

* Corresponding author.
E-mail address: rcolomo@gmail.com (R. Colomo-Palacios).

[1] Source code files, classes and interfaces are usually referred as software items in this research paper. Therefore, Gridmaster represents software items at the file level as well as the class and interface level, whereas the Socio-Technical Graph represents contributions at the file level.

http://dx.doi.org/10.1016/j.scico.2015.09.005

quately manage collaboration between team members, manage the evolution of quality control and remain informed of all the activities carried out by members of the group [6].

The construction of a shared knowledge space and the fluidity of communication may be affected not only by geographical distance but also by cultural factors, which can result in less efficient communication between DSD work teams [7,8]. Additionally, a high level of activity and the technical complexity of much of these activities further limit communications, as it is not feasible, usually for reasons of time, to communicate either orally or in writing all the details of the work that is carried out. Thus, to construct a shared knowledge space, communications between work teams are not sufficient; neither is the use of a computer system in which details of the activities carried out could be reported by means of completing forms [9]. The result is that automated mechanisms that report the changes made and the tasks carried out are frequently put in place by means of Integrated Development Environments (*IDEs*) or the metadata and logs stored by Software Configuration Management (*SCM*) tools or bug tracker tools, which are supported by ontologies in the most advanced systems [10–12].

To keep all the members of a team and, in particular, the project administrators informed about what is happening in the development and maintenance of a software system, it is not sufficient merely to rely on the details of the activities that have been carried out recently. It is necessary to perform a detailed analysis, usually for a period of development time and exceptionally for the entire period of development, and provide methods that facilitate comprehension [9].

The aforementioned analysis is known as software evolution analysis, and its principal objectives are to provide information that contributes to the understanding of the Software Evolution (*SE*) process, thus supporting the improvement of system development and maintenance processes, including project management and, as a result, improving human resource management. However, *SE* analysis produces large and complex datasets due to the number of variables involved in the process of change and the complexity of their relationships, which makes an adequate analysis difficult. Although the result of analysing the evolution of software elements provides useful information, it does not provide sufficient information to carry out the tasks of understanding changes and project evolution in a satisfactory fashion to provide adequate support for decision-making. This has led to the following research question being formulated:

> *How can software project leads and managers be supported in the decision-making process with regard to human resource management and the tasks carried out by programmers, by using the analysis results of the evolution of software projects and the socio-technical relations that are established during its evolution?*

Research efforts have focused on the use of visual representations combined with interaction techniques to gain insight when using large and complex datasets. In the context of software systems, these research efforts have concentrated on Software Visualization (*SV*) [13] and Software Evolution Visualization (*SEV*) [14]; although, more recently, some research has been carried out on the application of Visual Analytics (*VA*) to software systems [15] and SE [16] with the aim of providing better results. The aim of such research is to support the process of understanding *SE* and improve the design and implementation strategies of tools directed to satisfy the analysis needs of both programmers and managers.

Therefore, to obtain the results that will answer the formulated research question, it is necessary to design and implement an architecture that demonstrates the usefulness of the application of *VA* to *SE* by using, as data sources, the metadata and source code stored by *SCM* tools for each revision of a software project. However, the application of *VA* to *SE* is relatively new and a definition of this process is lacking. Thus, this makes it necessary to produce a definition that can be used as a base for the design and implementation of the aforementioned architecture.

Consequently, to define the process of applying *VA* to *SE*, it is necessary, on the one hand, to describe the process of applying *VA* to *SE* and, on the other hand, the identification of the roles, borders, interactions and relationships between research areas, methods and techniques involved in such a process.

In accordance with what has already been stated, a secondary objective of this research is to define the process of applying *VA* to the evolution of software systems and answer the following research question:

> *How can the process of applying Visual Analytics to the evolution of software systems be defined and explained, taking into account the components, methods and techniques involved in this process?*

The remainder of this research paper addresses the following topics: Section 2 discusses the background of the application of *VA* to *SE*, Section 3 describes the process of applying *VA* to *SE*; Section 4 demonstrates the utility of the process by the definition of Maleku,[2] an architecture that was implemented based on the aforementioned process; Section 5 explains the visualizations and interaction design; Section 6 discusses the use of collaboration patterns and socio-technical

[2] The architecture was named after Maleku to honour a small native group from the region of Guatuso, Costa Rica, with an ancient tradition in designing and decorating colourful masks and drums.