Data & Knowledge Engineering 85 (2013) 2–14

Contents lists available at SciVerse ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Approximation of COSMIC functional size to support early effort estimation in Agile

Ishrar Hussain *, Leila Kosseim, Olga Ormandjieva

Department of Computer Science and Software Engineering, Concordia University, 1400 de Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8

ARTICLE INFO

Article history:
Received 22 December 2010
Received in revised form 4 July 2011
Accepted 25 June 2012
Available online 8 July 2012

Keywords:
Software requirements
Functional size measurement
Text mining
Natural language processing
Agile development processes

ABSTRACT

The demands in the software industry of estimating development effort in the early phases of development are met by measuring software size from user requirements. A large number of companies have adapted themselves with Agile processes, which, although, promise rapid software development, pose a huge burden on the development teams for continual decision making and expert judgement, when estimating the size of the software components to be developed at each iteration. COSMIC, on the other hand, is an ISO/IEC international standard that presents an objective method of measuring the functional size of the software from user requirements. However, its measurement process is not compatible with Agile processes, as COSMIC requires user requirements to be formalised and decomposed at a level of granularity where external interactions with the system are visible to the human measurer. This time-consuming task is avoided by agile processes, leaving it with the only option of quick subjective judgement by human measurers for size measurement that often tends to be erroneous. In this article, we address these issues by presenting an approach to approximate COSMIC functional size from informally written textual requirements demonstrating its applicability in popular agile processes. We also discuss the results of a preliminary experiment studying the feasibility of automating our approach using supervised text mining.

1. Introduction

The agile development process breaks down the software development lifecycle into a number of consecutive iterations that increases communication and collaboration among stakeholders. This type of process focuses on the rapid production of functioning software components along with providing the flexibility to adapt to emerging business realities [1]. In practice, agile processes have been extended to offer more techniques, e.g. describing the requirements with user stories [2]. Instead of a manager estimating developmental time and effort and assigning tasks based on conjecture, team members in agile processes use effort and degree of difficulty in terms of points to estimate the size of their own work, often with biased judgment [3]. Hence, an objective measurement of software size is crucial in the planning and management of agile projects.

We know that effort is a function of size [4], and a precise estimation of software size right from the start of a project life cycle gives the project manager confidence about future courses of action, since many of the decisions made during development depend on the initial estimations. Better estimation of size and effort allows managers to determine the comparative cost of a project, improve process monitoring, and negotiate contracts from a position of knowledge.

* Corresponding author.
E-mail addresses: h_hussa@cse.concordia.ca (I. Hussain), kosseim@cse.concordia.ca (L. Kosseim), ormandj@cse.concordia.ca (O. Ormandjieva).
URLs: http://www.linkedin.com/in/ishrar (I. Hussain), http://users.encs.concordia.ca/~kosseim (L. Kosseim), http://users.encs.concordia.ca/~ormandj (O. Ormandjieva).

0169-023X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2012.06.005

The above has led the industry to formulate several methods for functional size measurement (FSM) of software. In 1979, Allan Albrecht first proposed FSM in his work on function point analysis (FPA) [5], where he named the unit of functional size as "Function Point (FP)". His idea of effort estimation was then validated by many studies, like [6,7], and, thus, measuring the functional size of the software became an integral part of effort estimation. Over the years, many standards have been developed by different organisations on FSM methods, following the concepts presented in Albrecht's FPA method. Four of these standards have been accepted as ISO standards: they are IFPUG [8], Mark II [9], NESMA [10] and COSMIC [11].

In recent years, many studies (e.g. [12–14]) have attempted to automate the process of different functional size measurement methods, but, to our knowledge, none has addressed this problem by taking the textual requirements as input to start the automatic measurement process. In addition, all these work depended on extracting manually the conceptual modeling artifacts first from the textual requirements, so that a precise functional size measurement can be performed. On the other hand, the work documented in this paper aims to develop a tool that would automatically perform a quicker approximation of COSMIC size without requiring the formalisation of the requirements. This is in response to the high industrial demands of performing size estimation during agile development processes, where formalisation of requirements are regarded as costly manipulation, and, thus, ignored during size estimation. Our methodology extends the idea presented in the Estimation by Analogy approach [15] and the Easy and Quick (E&Q) measurement approach, that was originated in the IFPUG standard [16]. The applicability of this approach in COSMIC was manually demonstrated by [17].

2. Background

2.1. COSMIC

For the purpose of this research, we have chosen to use the COSMIC FSM method developed by the Common Software Measurement International Consortium (COSMIC) and now adopted as an international standard [11]. We chose this method in particular, because it conforms to all ISO requirements [18] for FSM, focuses on the "user view" of functional requirements, and is applicable throughout the agile development life cycle. Its potential for being applied accurately in the requirements specification phase compared to the other FSM methods is demonstrated by the study of [19]. Also, COSMIC does not rely on subjective decisions by the functional size measurer during the measurement process [11]. Thus, its measurements, taken from well-specified requirements, tend to be same among multiple measurers. This is particularly important for validating the performance of our automatic size measurements.

In COSMIC, size is measured in terms of the number of *Data-movements*, which accounts for the movement of one or more data-attributes belonging to a single *Data-group*. A data-group is an aggregated set of data-attributes. A *Functional Process*, in COSMIC, is an independently executable set of data-movements that is triggered by one or more *triggering events*. A triggering event is initiated by an *actor* (a functional user or an external component) that occurs outside the boundary of the software to be measured. Thus, a functional process holds the similar scope of a use case scenario, starting with the triggering event of a user-request and ending with the completion of the scenario. Fig. 1 illustrates the generic flow of data-groups from a functional perspective, presented in the COSMIC standard [11].

As shown in Fig. 1, the data-movements can be of four types: Entry, Exit, Read and Write. An Entry moves a data-group from a user across the boundary into the functional process, while an Exit moves a data group from a functional process across the

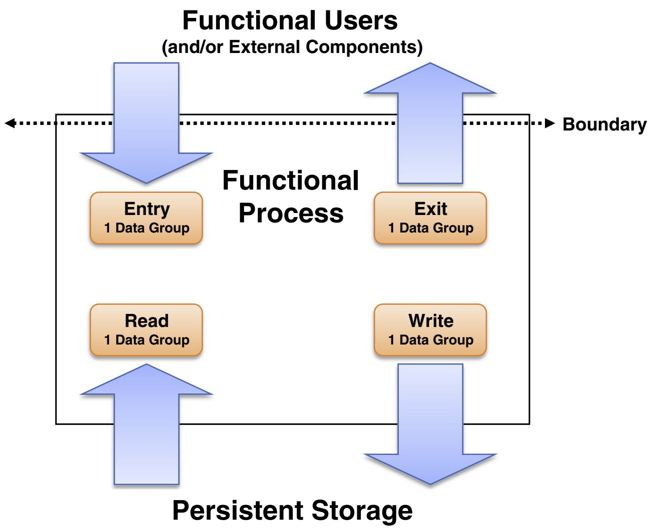

Fig. 1. Generic flow of data-groups in COSMIC [11].