Information and Software Technology 76 (2016) 81–91

Contents lists available at ScienceDirect

Information and Software Technology

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

Empowering global software development with business intelligence

Alejandro Maté [a,*], Juan Trujillo [a], Félix García [b], Manuel Serrano [b], Mario Piattini [b]

[a] Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n - 03690 San Vicente del Raspeig - Alicante, Spain

[b] Alarcos Research Group, Department of Computer Science, University of Castilla-La Mancha, Paseo de la Universidad/4, 13071, Ciudad Real, Spain

ARTICLE INFO

Article history:
Received 7 October 2015
Revised 23 February 2016
Accepted 22 April 2016
Available online 26 April 2016

Keywords:
Global software development
Business intelligence
Kpis

ABSTRACT

Context: Global Software Development (GSD) allows companies to take advantage of talent spread across the world. Most research has been focused on the development aspect. However, little if any attention has been paid to the management of GSD projects. Studies report a lack of adequate support for management's decisions made during software development, further accentuated in GSD since information is scattered throughout multiple factories, stored in different formats and standards.

Objective: This paper aims to improve GSD management by proposing a systematic method for adapting Business Intelligence techniques to software development environments. This would enhance the visibility of the development process and enable software managers to make informed decisions regarding how to proceed with GSD projects.

Method: A combination of formal goal-modeling frameworks and data modeling techniques is used to elicitate the most relevant aspects to be measured by managers in GSD. The process is described in detail and applied to a real case study throughout the paper. A discussion regarding the generalisability of the method is presented afterwards.

Results: The application of the approach generates an adapted BI framework tailored to software development according to the requirements posed by GSD managers. The resulting framework is capable of presenting previously inaccessible data through common and specific views and enabling data navigation according to the organization of software factories and projects in GSD.

Conclusions: We can conclude that the proposed systematic approach allows us to successfully adapt Business Intelligence techniques to enhance GSD management beyond the information provided by traditional tools. The resulting framework is able to integrate and present the information in a single place, thereby enabling easy comparisons across multiple projects and factories and providing support for informed decisions in GSD management.

1. Introduction

Nowadays, the globalization of software products and the variety of skills required for their development have made necessary to implement a wide and general procedure in order to adopt new approaches in software engineering. In this context, the concept of Software Factories (SF) has arisen, in which software is produced in an industrialized fashion, with multiple delocalized development groups collaborating together across the globe and enabling what has been referred as Global Software Development (GSD) [1,2].

However, while this approach allows companies to integrate a variety of talent and skills from different places, traditional management techniques for software development were not designed for this environment. In turn, new challenges arise that threaten the success of software development in SF. On the one hand, the range of skills and cultures affects the performance of development groups, depending on how much they need to interact with others. Therefore, project estimations need to take these aspects into account in order to maintain their accuracy [3,4]. On the other hand, the lack of integrated data and missing information across development groups hinders the capability of analysis and decision making processes. Indeed, one of major failures in software factories is the lack of knowledge regarding what data has to be gathered to

* Corresponding author. Fax: +34965909326.
E-mail addresses: amate@dlsi.ua.es (A. Maté), jtrujillo@dlsi.ua.es (J. Trujillo), Felix.Garcia@uclm.es (F. García), Manuel.Serrano@uclm.es (M. Serrano), Mario.Piattini@uclm.es (M. Piattini).

http://dx.doi.org/10.1016/j.infsof.2016.04.011

support the decision making process [4,5]. In turn the lack of information leads to the inability to answer critical questions for making decisions. How are we doing with regards to the overall project development? Are we progressing faster or slower than the previous month? Has it affected the rate of defects found during this testing phase? Which groups work better with others? Which group is most affected by potential changes in requirements? Is the product ready for a release?

Finally, GSD is also subject to organizational challenges that affect the software produced. As shown in [6], the organizational structure affects the quality of the software produced. In this respect, factories involved in software development can be organized in multiple ways, ranging from distributed groups pertaining to the same company [7], to outsourcing [8] and virtual enterprises. As organizational distance increases, its impact and risks on software development also increase [9]. Furthermore, in the case of outsourcing and virtual enterprises, the different groups not only need to envision the same piece of software, but they also benefit from being aware of the business objectives pursued by the group [10,11].

While most efforts have focused on the development aspect [1,3,4,7,12,13], little if any attention has been paid to the information needs from a software development management perspective [5]. Therefore, the main research question is: How can we provide a solution with adequate support for decision making in software development management?; while the subquestions: Are existing techniques sufficient to reach this solution? and, Can we develop a general information model with which to address these needs?

In this paper we propose to tackle these challenges by taking a Business Intelligence (BI) perspective on the analysis and management of software development in SF. BI is often referred to as the use of knowledge-intensive techniques to aid in the decision making process and improve the business performance. Software production in SF can benefit from both traditional and new BI techniques [11,14–16], from obtaining more visibility of the development process thanks to data integration to modeling Key Performance Indicators (KPIs) needed for decision making and enabling powerful analytics thanks to the adoption of a multidimensional perspective of the data.

However, before BI can be successfully applied to SF, a careful analysis is required in order to (i) adapt BI techniques for software development and (ii) determine the generalizability of each component, identifying which need to be specifically tailored for each case. Therefore, the main contributions in this paper are as follows: (i) we propose a methodology that guides and integrates current BI techniques in order to create a tailored BI solution for software development management, and (ii) we propose an architecture to improve software management in SF by enriching and integrating the information available, thereby enabling BI and software analytics. Our approach takes into account the specific BI information needs required in SF, including high-level performance indicators and complementing those found in [5], and improves the managers' awareness when making decisions by creating interconnected dashboards so that managers are aware of cross-concerns.

The remainder of this paper is structured as follows. Section 2 presents the related work in the area of SF and the background on BI. Next, Section 3 shows our methodology applied to the elicitation of specific needs in SF by means of a running example. Afterwards, Section 4 presents how the BI solution for SF is implemented. Following the implementation, we present a discussion in Section 5 about the generalizability of the artifacts and the implementation. Finally, Section 6 presents the conclusions and sketches future works.

2. Related work

Global software development has been a research topic for more than a decade. Initial research was focused on analyzing what and how globalization factors affected the results of the software development process. Multiple research works [1–4,12] showed that geographical, temporal, cultural, and organizational distance among other factors affected the performance in GSD. In [2], the authors analyze two case studies on GSD to extract a set of lessons learned. From these case studies, the authors highlight the importance of management and sharing standards across development groups in order to be successful. In [12], the authors analyze potential solutions for tackling the problem of distance in GSD, discussing the advantages and disadvantages of each one. Cultural, geographical and temporal distance are discussed in [3], where authors describe each of the challenges and how software development companies try to cope with them. In order to alleviate the effect of these factors, a number of tools have been developed to aid in GSD. In [17] the authors provide a systematic mapping review to discover all the available tools in the field of global software engineering (in total 132) and what functionality these tools provide. Several of those tools discussed are widely adopted by distributed developed teams and they are used in global software projects.

These tools have contributed to alleviate the effect of these factors. As a matter of fact, it may be observed a changing trend [18] nowadays, in which these geographical, temporal and sociocultural aspects have been replaced by other factors as the most critical. In recent studies [18] is reported that GSD experts perceive as crucial several factors the team members skills, the appropriate management of GSD projects, along with process maturity play an increasingly important role. These conclusions motivate the need to properly manage all the involved data to enhance the making decision process. This is especially important for companies working in GSD contexts, given the large quantity and diversity of available information which involves greater complexity, as information from each one of the factories and its corresponding sub-projects has to be handled.

There are few initiatives which deal with these issues from the visualization perspective. Burkhard and Meier [19], suggests a new visualization technique to deal with portraying large-scale systems based on the underground railway map metaphor, known as Tube Map. Da Silva et al. [20] propose the use of Cockpits, as a kind of dashboard, for the visualization and coordination of distributed software development. In [21] a visualisation environment named DESGLOSA is described. The environment is composed of a set of visualisation metaphors that can hierarchically organized to show both the information related to GSD projects and subprojects and the organisational context of companies and their corresponding factories. Lastly, we should mention the improvements in visualization that are based on cities, such as those proposed by [22,23]. Some breakthroughs in this environment are the systems based on cockpits [20] and World View [24], which proposes the display of development on a world map. Another stream of research in visualization which can be useful in GSD contexts has to do with tools which provide awareness of human activities in software development [25]. Some recent contributions related to this type of tools are SecondWatch [26] and ProxiScientia [27].

However, most of the techniques to support decisions for GSD managers are based either on visualization focus, or such as pointed out in [5] on the development aspect, whereas information needs have been mostly overlooked. As a result, it is typical that the information required for software development management is scattered, making it impossible to analyze. Therefore, in [5], the authors perform a survey at Microsoft in order to elicitate what information would be required to enable software development analytics. The results report a special focus on past and present analysis in order to identify not only what happened during projects but also why and how.

In parallel, lastest research in BI has been focusing on identifying critical information for making decisions, monitoring business performance, and how this information should be visualized. Initial works, such as the Balanced Scorecard (BSC) [10] and Strategy Maps [28] have been widespread adopted by enterprises to monitor their performance. The BSC monitors the activities of the company by keeping track of multiple KPIs that are related to the main business goals, while Strategy Maps analyze the business strategy looking for potential flaws. More recent works have included additional information for business modeling and monitoring. In [29], the Business Intelligence Model (BIM) allows analysts to formally model business objectives, their relationships, and KPIs. Finally, in [30], the authors conduct a comprehensive multidisciplinary literature review with an aim to identify critical issues when implementing dashboards. They state that dashboards are likely to solve the problems of presentation format and information load, such as those present in GSD, when certain features are present (e.g. high data-ink ratio and drill down features).

Nevertheless, there are two main drawbacks for the application of existing BI methodologies and techniques in GSD. First, they typically do not cover the whole cycle of development required to implement a solution, and instead are aimed at tackling specific problems, such as alignment through KPIs [10], business strategy analysis [29], or building the data warehouse itself [31], for which an interface must be developed afterwards. This implies that their application as a complete solution relies on deep knowledge and adequate selection of the BI techniques available so that they complement each other. Such knowledge is often missing from software developers and managers for which BI is out of their scope as they are only users of the system. Second, BI techniques themselves rarely include guidelines for their application to specific domains. This entails the existence of gaps in the solutions developed due to the lack of important concepts from the domain knowledge. For example, existing techniques lack in-depth analysis of indicator relationships which can have undesired side-effects when making decisions in certain domains, and it is common to overlook important information sources during the analysis when the BI analyst is not familiar with the domain. Therefore, we require a methodology that (i) integrates specific BI techniques and (ii) guides both the BI analyst and the domain expert in their development of a tailored solution for software development management, by taking into consideration not only their specific needs but also those elicited by the industry.

Summarizing, at the moment, GSD presents a large amount of scattered information that cannot be exploited in its current state. There is a need for powerful, simple to use tools, that enable software development analytics and present information in an integrated fashion, allowing managers and developers to navigate the data from aggregated to highly detailed levels. However, before this solution can be achieved, BI techniques must be integrated in an organized fashion in order to be able to create a final solution that satisfies all the requirements in managing software development.

3. Business intelligence for global software development

Successfully applying BI to GSD requires taking into account new factors not considered in traditional software development. From a high level point of view we can differentiate two main areas of focus. The first area is related to the business objectives of the companies involved, whereas the second area is related to the software development process and the systems in place to support it. In the following, we present our methodology to enable the successful application of BI for software companies which operate in a GSD context, starting from the analysis of business objectives up to the visualization of the information gathered. Firstly an overview of this process is provided. Afterwards, we present the case study that serves as running example. This running example will be used to explain the process and its application in detail.

3.1. Process overview

The methodological approach to support decision making in GSD by applying business intelligence is depicted in Fig. 1. As it can be observed, firstly, business objectives are analyzed, in order to establish their relationships and identify the business processes responsible for achieving them and KPIs for monitoring them and making high-level decisions (1). During the business modeling steps, we make use of the Business Intelligence Model (BIM) [29], presented in the Related Work section. Afterwards, decision makers and managers are interviewed in order to analyze their individual objectives and the information they require during the decision making process (2). This step is covered by means of the i∗ for data warehouses [31] modeling language. In order to obtain a comprehensive view of the impact of decisions, the influences of individual indicators identified during this step are included in an extended business objectives model (3). After the information needs have been elicited, we proceed to inspect the available data sources in order to analyze data quality and find information that may have been overlooked during the previous steps (4). With all the data gathered, we create a repository that integrates several data sources and allows decision makers and managers to analyze aggregated and detailed information (5). The repository is modeled at the conceptual level, using the platform-independent language UML for data warehouses [32]. Finally, we elaborate a set of dashboards (6) that take into account individual and business objectives and their relationships, so that managers are aware of other indicators affected when they make a decision.

3.2. Overview of the application example

In order to apply the formerly proposed method in real contexts, the data collected from a representative company participating in the ORIGIN R&D project were used. ORIGIN project was carried out by a consortium composed of five companies and two universities in Spain, with the aim to create a framework for the management and development of software in global contexts.

The company which was chosen to validate the method is formed by more than 9000 professionals distributed across 21 factories throughout several countries in Europe, Africa, The Americas, and Asia. Nowadays the significance of GSD in its business is very remarkable, given that its big sales volume increase is due mainly to the positive evolution of the international market. In order to preserve the confidentiality of the data they were anonymized. In the following we will refer to this company as *BestApps*.

The data records contained management information about GSD projects in the company. Each record included multiple information, such as the factories involved in the project, the project managers and development groups at each factory, and the technology used for developing and managing the project among others. Despite having all this information, it was extremely difficult to compare the performance of factories and projects, since the structure of the records had been evolving across factories and years. To further complicate the assessment, indicators and criteria to make decisions were scattered across the records, thus making it difficult to track and understand the status of a project across the multiple factories involved.

In order to improve the visibility of data and enable the performance analysis across factories and projects, we applied our

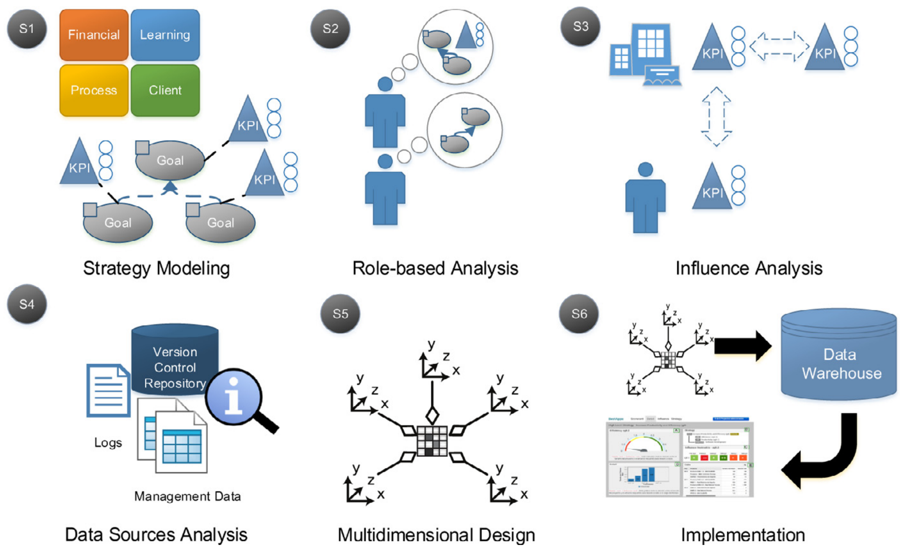

Fig. 1. Steps for applying BI to GSD.

methodology. In the following section, we present each step of the methodology with more detail by using the application example based on this real world case.

3.3. Application of the methodology

In this section each step of the methodology is described in detail and its usage is illustrated by applying it to the provided case study.

S1. Business strategy modeling

Business strategy modeling can help coordinating multiple SF by establishing a clear criteria depicting the priorities for the group [33,34] and how the group intends to achieve them. Business strategy modeling is based on extracting high level goals and indicators [35] pursued by each participant and obtaining a high level view of the relationships between them.

When making decisions, managers will try to improve the indicators associated with the processes at hand and, in turn, influence the KPIs associated with strategic goals. Therefore, it is important to provide this information in order to evaluate the effects of decisions on all the relevant indicators. For this task we can use either the Business Motivation Model (BMM) [36] or BIM [29]. The former allows us to model goals and indicators (referred as objectives in the BMM specification) in a textual fashion, whereas the second one includes a graphic notation, is formalized, and provides support for reasoning techniques that can be used to analyze the strategy. For these reasons, we use BIM for tackling the Business Strategy Modeling step.

Application

BestApps company is a software development company that produces software for multiple markets and customers by means of a GSD approach. At the moment, BestApps is trying to improve its efficiency. However, since only project management files are accessible, and they are scattered across multiple semi-structured files, the company lacks visibility of the process and it is difficult to make decisions. BestApps directly owns the different factories involved in the development process, thus, for our current case study, we will exemplify the process using BestApps strategic objectives.

BestApps pursues several high level objectives that the company wishes to achieve, shown in Fig. 2. They wish to Maintain

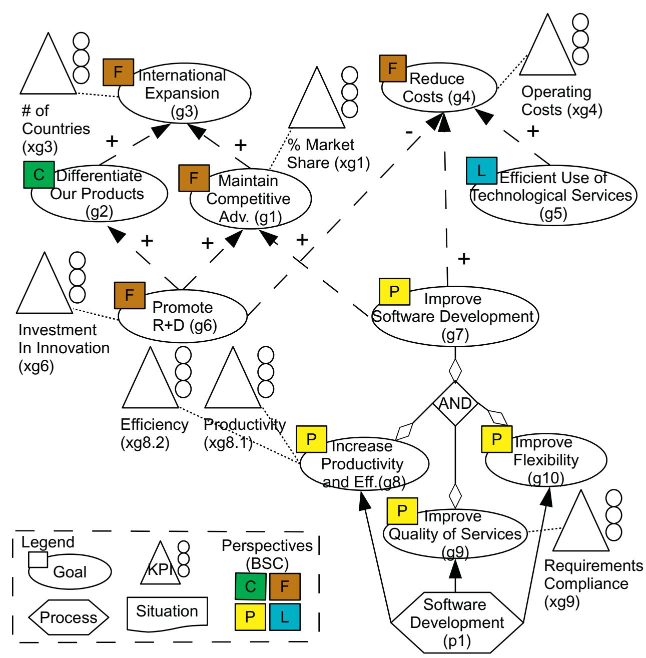

Fig. 2. BestApps strategic model.

their Competitive Advantage (g1) and Differentiate their Products (g2) in order to promote the International Expansion of the company (g3). Moreover, they also wish to Reduce Costs (g4) by making an Efficient Use of Technological Services (g5), all supported by Promoting R+D (g6) and Improving the Software Development Process (g7). Among these business goals, there are three especially relevant for us, related to the improvement of the software development process: Increase Productivity and Efficiency (g8), Improve the Quality of Services (g9) and Improve Flexibility (g10).

For each of these business goals, a KPI is required in order to objectively monitor its performance. As we are trying to improve software development, we will focus on the KPIs selected for this process: Productivity (Avg Number of Products/Time, xg8.1), Efficiency (Avg resources/products, xg8.2) and Degree of Requirements Compliance (Requirements Fulfilled/Identified, xg9). As we can see in Fig. 2, the goal Improve Flexibility does not have an associated

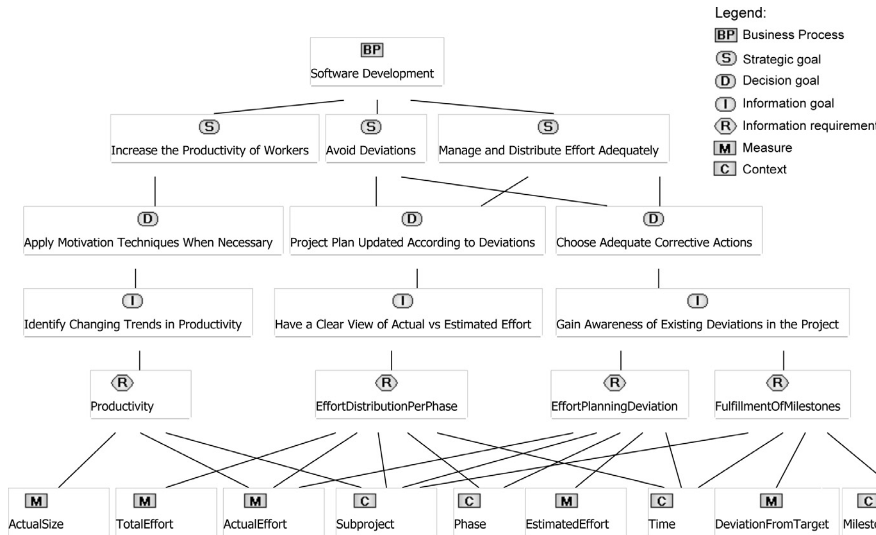

Fig. 3. Project Manager information needs.

KPI. This goal is defined as "being able to cope with changing conditions in the environment or in requirements". Therefore, we can identify influences between goals, but there is no clear criteria to determine if the company is achieving this goal or not.

Once we have identified the high level KPIs, we proceed to perform a role-based analysis of the objectives pursued by managers within the software development project. This will allow us to (i) understand how each role tries to improve the process and affects business-level KPIs, and (ii) what information they require for making their decisions.

S2. Role based decision making analysis

Role based decision making analysis is important because it ensures that we (i) build an information system that supports the different roles involved in the decision making process, and (ii) each role is provided only relevant information. In order elicit this information, we analyze their individual objectives during interviews and then refine them by asking "why" and "how" they intend to achieve them. For this task there are multiple goal modeling languages that enable us to analyze individual objectives, the most prominent ones being the i∗ family of goal modeling languages [31,37–39]. Among these languages, i∗ for data warehouses [31] and GRaND [39] allow us to establish relationships from strategic goals to the information required to achieve them, required later on for building the information repository. We choose i∗ for data warehouses for its integration within a Model Driven approach that lets us create the information repository in a semi-automatic way, reducing the time and effort required for the implementation.

Application

There are multiple roles involved in software development management, each contributing to the success of the project and, ultimately, the achievement of the high-level business objectives. The three main roles for software development management in BestApps are as follows: (i) the Project Manager, (ii) the Requirements/Scope Manager, and (iii) the Quality Manager.

- The **Project Manager** has three strategic goals to meet, shown in Fig. 3. He wishes to Manage and Distribute Effort put into the different tasks throughout the project, Avoid Deviating from the plan specified, and Increasing the Productivity of the workers ("S"). In order to achieve these goals he needs to make several decisions ("D") using information from four individual indicators. These indicators are represented by the information requirements in the diagram ("R"). The first indicator, Productivity, establishes the ratio between two measures ("M") Actual Effort to Actual Size. The second indicator, Effort Distributed per Phase, calculates the ratio between the Actual Effort for a certain Phase compared to the Total Effort. The third indicator, Effort Planning Deviation, provides information about how much the project is deviating from the plan. This indicator is calculated by subtracting the Actual Effort to the Estimated Effort and dividing the result by the total effort. Finally, the fourth indicator is the Fulfillment of Milestones, which keeps track of the deviation from the agreed dates for each Milestone with the Client. Each of these indicators is related to several contextual elements ("C") that will be transformed into dimensions in our BI system.
- The **Scope Manager** is focused on maintaining the stability of the project by managing changes in requirements. His two main strategic goals are Ensuring the Stability of the project and Obtain Client's Approval. For achieving these strategic goals, he has to decide whether to accept or reject changes in requirements, and reach an agreement with the client. On the one hand, the relevant indicators for helping the Requirements Manager gather information on the Stability of the Project are the ratio of Requirements Added, Modified, and the Requirements Stability, obtained by how many requirements are altered compared to the Total Requirements. On the other hand, the ratio of Requirements Cancelled and Approved help him understand how much the Client is satisfied with the current product.
- Finally, the **Quality Manager** ensures that the software developed meets the quality standards of the company. This way, his main strategic goals are Increasing Product Quality, Improving Test Effectiveness and Improving the Quality of the Maintenance phase. In order to achieve these strategic goals, he focuses on nine different indicators. On the one hand, Test Coverage and Test Effectiveness help him decide if testing tasks of the project require an improvement. On the other hand, Ratio of Defects, Quality Level, and Technical Quality of the Source Code help him decide if the code needs refactoring. Finally, the Issue Resolution Degree and Time, Software Evolution, and System Stability provide him information to evaluate the Quality of the Maintenance Phase and adjust the effort dedicated to it.

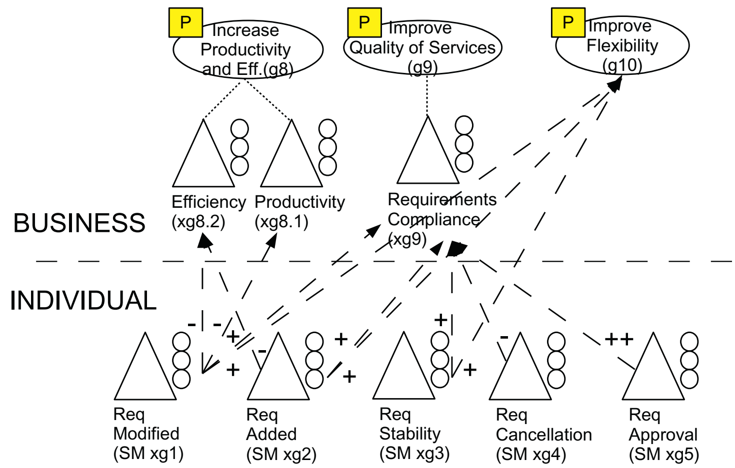

Fig. 4. Influences of Scope Manager's indicators on business KPIs.

Aside from these roles, given the GSD nature at BestApps, a **Global Manager** is in charge of supervising the progress made across SF and projects. This manager is focused on deciding if new factories are incorporated to the projects and requesting corrective actions if deviations appear at project level. Therefore, the Global Manager uses a high level view of the data, characterized by the introduction of Projects as aggregations of Subprojects and Factories, allowing him to compare how much effort is put on the project by each factory and what deviations exists.

As we can see, each of the roles focuses on its own set of indicators and requires different information in order to make decisions. However, decisions often affect more than one indicator, both at individual as well as at company level. Whether directly or indirectly, these collateral effects are not always desired, but it is rarely the case that decision makers are provided with the corresponding information. In order to avoid this pitfall, our methodology models the effect of individual indicators both on the global KPIs as well as on other individual indicators. This will serve us later on to implement a visualization of the information model that considers these relationships and provides managers comprehensive information to make their decisions.

S3. Modeling indicator influence relationships

Individual indicators are important for managers because they help them prioritize and make objective decisions based on available data. However, it is often overlooked that maximizing individual indicators may not be positive for all business goals, or even for other indicators that are also being monitored. Influence relationships sometimes lead to the unfortunate situation where two managers try to adjust the value of their own indicators and constantly disrupt each other as a result.

By modeling the relationships between indicators we provide awareness about the impact of each decision made. This step should be performed in collaboration with experts, or using mining techniques [40,41], as the relationships between indicators may need to be estimated.

Application

In the case of BestApps, our collaboration obtained the results shown in Fig. 4, where Scope Manager's (SM) indicators are related to business KPIs and goals. A plus (+) sign shows a positive influence on another indicator, whereas a minus (-) represents a negative influence.

There are three KPIs that may be affected by individual indicators: Productivity (xg8.1), Efficiency (xg8.2), and Requirements Compliance (xg9). Furthermore, we can estimate the impact on

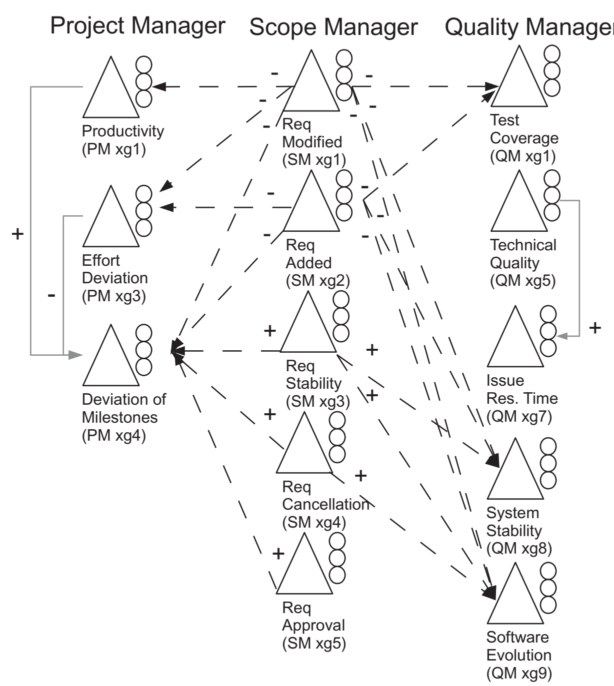

Fig. 5. Influences across individual indicators.

the business goal Improve Flexibility by means of its definition. An increase in Requirements Modified (SM xg1) leads to a decrement in Efficiency (xg8.2), as time and resources required to obtain the product increases. Productivity (xg8.1) is also likely to decrease, since modifications do not alter significantly the size of the product but require effort to be implemented. Finally, dealing with an increasing number of requirements modified increases Requirements Compliance (xg9) and Flexibility (g10), as the updated needs of the customers are taken into account. A similar effect is produced by the addition of requirements (SM xg2). However, in this case, new requirements are likely to increase the code base, thus it is not expected to have an effect on Productivity. Furthermore, Requirements Stability affects positively Requirements Compliance and Improve Flexibility, since it indicates that the user is satisfied with current requirements. Finally, Requirements Cancellation and Requirements Approval are mainly related to Requirements Compliance.

Fig. 5 shows the analysis of the impact of individual indicators on other individual indicators. For the sake of brevity, indicators that do not have any relationship are not shown. Productivity (PM xg1) has a positive effect against the Deviation of Milestones (PM xg4), whereas Effort Deviation (PM xg4) is likely to harm it. Furthermore, Scope Manager's indicators are the ones with the biggest range of influences since they affect the requirements of the product being developed. Adding or modifying requirements (SM xg1, xg2) has a negative impact on most indicators of other managers, whereas maintaining requirements stable or cancelling them helps to correct Deviation of Milestones and minimize Software Evolution. Finally, in the case of the Quality Manager, the Technical Quality of the code (QM xg5) is likely to help improve Issue Resolution Time (QM xg7).

By identifying all the existing influences, we can build an information system that improves awareness and avoids undesired collateral effects. Once the strategic analysis has been completed, the next step is to analyze the data sources in order to locate relevant data for making decisions.

S4. Data source analysis

Data sources provide the necessary data for making decisions. As we will see, most of the data provided by data sources are low-level data. While low-level data is interesting for finding the root of a problem, it is not adequate to support the decision-making process, as (i) managers cannot feasibly understand the implications of such a large and varied amount of data and (ii) isolated data often cannot explain the rationale of problems at hand because part of the information is missing. Therefore, these data will require to be integrated and aggregated in order to ease its use [5].

Application

Within GSD environments there are multiple data sources from where information can be extracted, from repositories to messages, conversations, and professional social networks [42]. Since our intention is to provide accurate and reliable information for decision makers and managers, in this paper we will focus on the most common and reliable set of data sources: project repositories, ticketing systems, and project metadata.

First, project repositories store the source code of the project being worked on. They come in different fashions depending on the technology used, such as Git or Subversion (SVN), and can be standalone or integrated with management platforms, such as Atlassian or Hackystat [43]. They can provide us technical information about the project such as the number of commits, the author of each commit, the timestamp when the commit was made, the files that were affected, and the size of files. Additionally, depending on the system we can obtain the number of lines of code, or the ratio of coverage of the current test suite among others.

Second, ticketing systems track pending tasks, requests for changes, and reported bugs. These systems include information about who is working on each task, how much time was planned and how much time was actually needed for a task, the priority level of the ticket, the status, and what other tasks and files are related to the task at hand. Furthermore in the case of bugs, ticketing systems also include the version of the project affected, the relevant files, the person who reported the bug.

Third, project metadata and project management files provide crucial information regarding how projects are developing. As in the case of repositories, project metadata and management files can be supported by a number of technologies, typically Microsoft Project files. Among the metadata included, we can highlight the project name, the target market, the customer, the project manager, and the type of project lifecycle. Finally, among project management data we can find estimated and current effort, tests data, the project phase, the resources dedicated to the project, the list of requirements and maintenance information.

Once we have identified all the relevant data, the next step is to structure it into a multidimensional view to enable data navigation.

S5. Multidimensional design

Multidimensional modeling is a technique proposed [14] for enabling Online Analytical Processing. One of the major challenges of multidimensional modeling is creating a model that is able to represent the information while being understandable by decision makers and satisfying their needs [31,32]. First, we will present the basic concepts involved in multidimensional modeling. Then, given the large amount of information available, we will present a summary of our multidimensional model.

Multidimensional modeling is based on structuring the information in terms of Facts (center of analysis) and Dimensions (context of analysis). On the one hand, Facts contain FactAttributes, that provide information about the performance of the process being analyzed. For example, the NumberOfDefects provides information about the quality of the software. On the other hand, Dimensions provide context information for the analysis, allowing the user to analyze performance from multiple perspectives. For example, the Project dimension allows us to analyze how much we have been deviating from our estimations across multiple projects. In addition, dimensions also enable managers to navigate data thanks to their hierarchical structures. For example, a Project with 5 defects can be divided into several Subprojects, one with 1 defect and another with four defects.

The representation of multidimensional models can be done through several modeling languages which share these common concepts, including UML for data warehouses [32], dimensional fact model [44], or the extended ER model [45]. In our case, we choose UML for data warehouses due to its enhanced expressivity, allowing us to represent advanced multidimensional details such as cardinality of dimensions and hierarchies, and due to its integration within a Model Driven approach [31].

Application

We will start by describing the dimensions involved in software development analytics, part of which can be seen in Fig. 6. We have three dimensions related to the customer: Market, Client, and Milestone. These dimensions are especially relevant for managers since they help to compare how different clients are satisfied by the products and services offered [5]. Market provides information about the type of software being produced, allowing BestApps to compare how products that target different types of markets perform. For example, we can compare how well ERPs perform compared to Analytic solutions. Next, Client describes the information known about the specific client that is using a product of the company. Clients can be further aggregated into areas of activity. For example, one of BestApps clients is SteelWheels (fictitious), which belongs to the area of Motorvehicles. The last dimension in the group, Milestone, stores information about the set of Milestones agreed with a Client.

Furthermore, we have three dimensions that are related to all kinds of software development: Project, Phase and Issue (omitted). First, a Project represents one of the multiple developments that the company is undertaking. Projects are composed of one or more Subprojects, each assigned to one or more Factories. It is worth noting that, particularly to this company, Projects also represent versions of the same software. Therefore, if managers wish to analyze Versions and Releases as a separate entity, an independent dimension should be added to the multidimensional model containing the corresponding information. Second, Phase dimension represents the multiple phases that a project can pass through. This dimension allows managers to evaluate how each Project evolves through multiple Phases. Third, an Issue represents tasks requiring

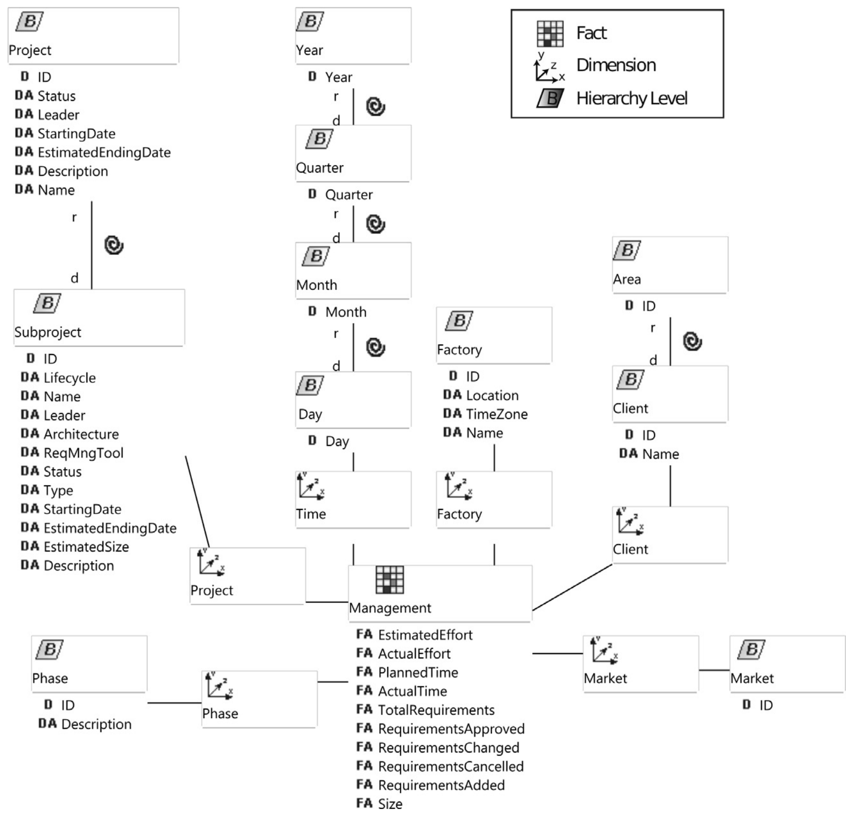

Fig. 6. Management fact and related dimensions in our information schema.

attention from a developer, and can be grouped according to their Priority.

Next, we have the Factory and Time dimensions. On the one hand, Factory is a specific dimension of GSD, relating the information stored in the facts to the factory or factories responsible for it. Time, on the other hand, is a standard dimension on most data warehouses, as it allows us to analyze the information in chronological order or slice the data pertaining to a specific period. Our schema presents a standard Time hierarchy, that can be further extended depending on the needs of each specific case.

Finally, the remaining dimensions omitted are Commit, Developer, and Artifact dimensions, which store the information regarding changes performed on the software during the development

These dimensions are distributed across three facts: (i) Management, seen in Fig. 6, containing measures related to the management aspects of software development, (ii) Milestones, containing specific information related to the deviation from the milestones agreed with the Client, and (iii) QualityAndMaintenance, containing measures related to quality aspects of the process and the software product. Each of these facts is linked with several dimensions that enable the comparison of performance across groups and data navigation. In order to verify that the schema proposed meets BestApps information needs, as well as those specified in literature, we mapped them to the data stored in the schemata proposed. All information needs were satisfied except for those that require an inspection or analysis of the code. We recommend to be integrate this information directly into the end user layer, as historical values are rarely stored.

The multidimensional model proposed allows decision makers to navigate through the data by means of dimensions and hierarchies. In the following section, we will show how an interface for analysis by means of dashboards that is easy to use and contains all the critical elements identified, making the information accessible to software development managers [5].

4. Implementation and visualization

The implementation of the dashboards was performed taking into account the information described in previous sections.

In Fig. 7 we can see one of the dashboards proposed for BestApps, more concretely the detailed view of the KPI Efficiency (xg8.2). This initial set of dashboards is tailored according to the business strategy and provides different perspectives of the data to support the decision making process. In addition to this set of dashboards, each role has available its corresponding view containing the information identified as relevant thanks our methodology introduced in previous sections. Our solution includes a Scorecard view with all the KPIs defined in the strategy. For each of them, we provide a description including how it is calculated (A), the elements that are affected by the KPI (B), its influence on other KPIs (C), and a detailed view of the evolution of its value across time (D). The influence map is represented by means of a new chart called Influence heat-matrix, let managers acknowledge the existing influences and navigate between the KPI details view. For showing the relationship in a chart we join a co-occurrence chart and a heatmap providing a custom visualization, similar to [41]. This view helps managers to assess the impact of their decisions on other KPIs.

The personalized set of dashboards presented in Figs. 8 and 9 present information tailored to the specific roles involved in GSD. Fig. 8 shows the Map view available exclusively to the global manager, who requires an overview of the different factories involved in the projects selected and their current performance. This view allows the global manager to quickly pinpoint problematic projects and factories and drill into the details in order to locate the source of the problem. Meanwhile, Fig. 9 shows the personalized view of the Scope Manager, focused on giving an overview of his most relevant KPIs across projects.

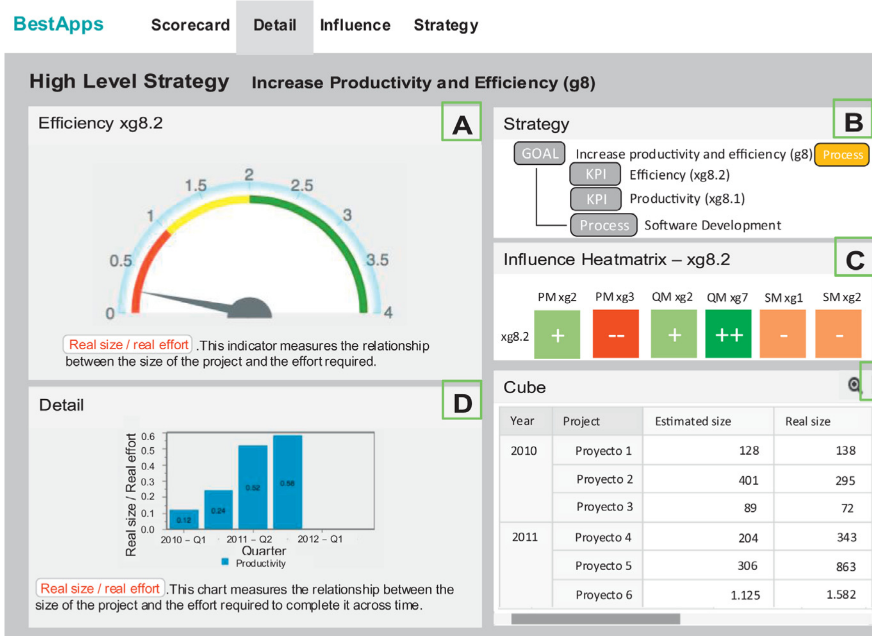

Fig. 7. Detailed view of KPI Efficiency (xg8.2).

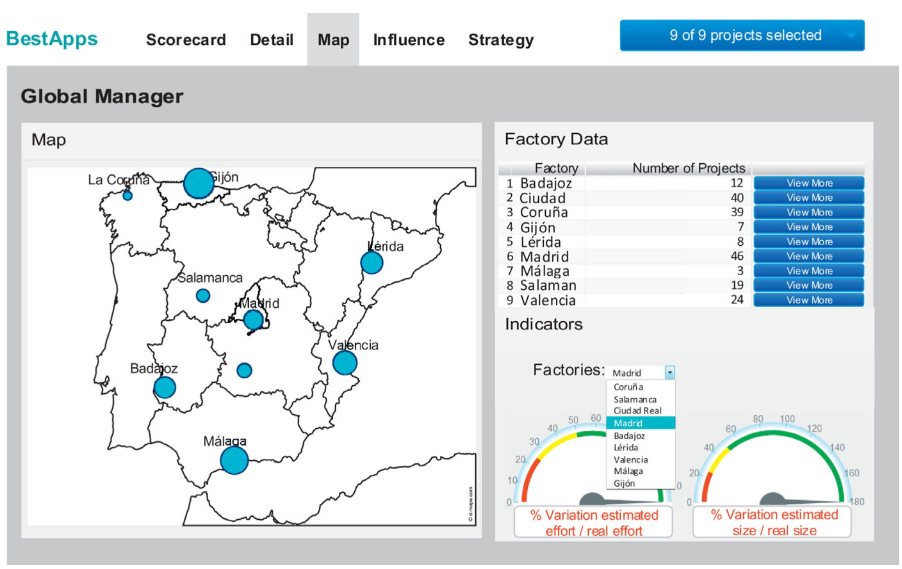

Fig. 8. Personalized dashboard for the global manager.

As shown, our visual approach allows every management role to be aware if the business objectives are being achieved, what is the status of relevant indicators for other management roles, and make a decision being aware of the impact that it will have on the rest of the elements involved.

5. Discussion and limitations

We have shown that, by following our methodology, we are able to cover the information needs of managers in software development. While the methodology can be applied to other case studies "as-is", the results of certain steps are expected to vary with the case study at hand. In this section we shall first highlight the reasons behind these variations in order to provide a more comprehensive view of the results obtained. We shall then provide a discussion on the limitations of the approach.

First, regarding the artifacts produced, business strategy modeling and alignment depends on the companies involved. In our case, BestApps was the single owner of all the factories. In other cases, the strategy will be a combination of the goals of all the participants. The benefit of applying this step is that all companies will share a common vision of the objectives pursued, thus helping avoid potential conflicts between them. Furthermore, the set of personal indicators obtained depends on the objectives prioritized by each particular software management role. In our experience, the set of indicators presented is expected to be relatively stable across different case studies.

Second, the information repository has been designed by considering all the information needs elicited from both literature and the case study, and it is therefore expected to change only depending on the (lack of) information available. We must, however, highlight that some managers may wish to take a different perspective on the information. For example, as mentioned previously in *S5*, it may be useful for them to separate the information of Projects from its multiple versions and releases. In these cases, the adapted schema can be obtained by repeating the role-based decision

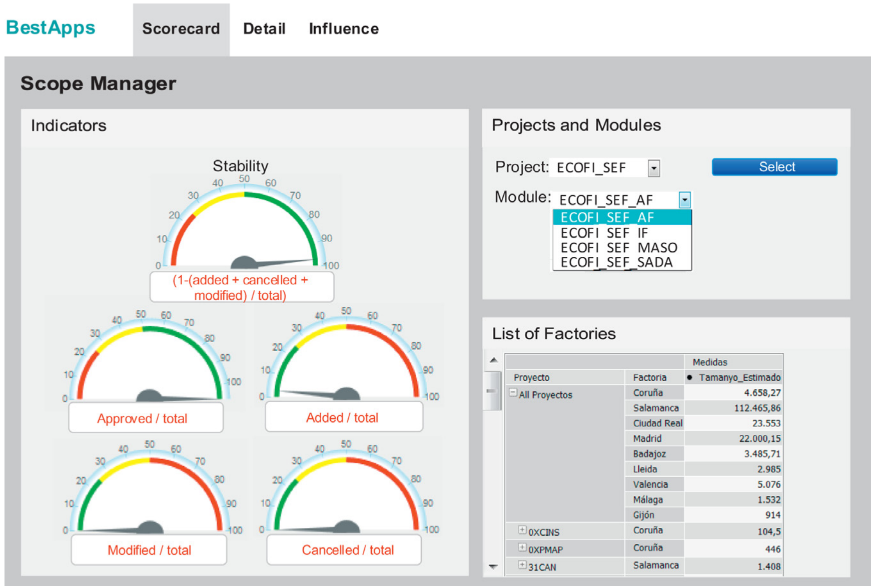

Fig. 9. Personalized dashboard for the scope manager.

making analysis described in *S2*, and deriving the new multidimensional model according to their preferences.

Third, in terms of implementation, the processes used to extract the information are to be implemented on a case by case basis depending on the data sources at hand. Examples of these sources are the Git/SVN repositories, Microsoft Project files, PROM [46] or Hackystat [43]. Furthermore, dashboards and analysis cubes created will depend on the BI tool preferred by each company. We chose Pentaho [47] since it has been widely adopted and is open source, thus it can be reused across other case studies. However, other BI solutions (SASS, etc.) could be chosen for the implementation.

Regarding the limitations, while business strategy modeling performed with BIM in S1 requires little training (1 session), business alignment and conflict resolution may require additional effort. BIM includes formal reasoning techniques to help identify conflicts and potential alternatives. However, they require more in-depth knowledge of the framework in order to interpret the results and propose alternatives if none exist at the moment. Since BestApps was the sole owner of all factories, this issue did not arise in our study.

In *S2* and *S3*, depending on the number of performance indicators that the company uses, the number of relationships can increase dramatically. In this respect, we recommend the definition of few high-level aggregated indicators associated with the software development process. The reason for this is twofold. First, these indicators provide the managers with a concise idea of how decisions made regarding their individual indicators align with the objectives of the company. Otherwise, even if all the relationships can be identified, managers may have difficulties in making informed decisions owing to the increased visual and cognitive complexity. Second, in our experience data quality can be a problem at the moment at which it is extracted from software development repositories, and the application of mining techniques in the early stages of the methodology so as to simplify the analysis of large groups of indicators may therefore require additional data cleaning efforts. Despite these limitations, indicator relationships are still useful because of the awareness that they provide about the impact of individual decisions. While this may not prevent a manager from prioritizing her own indicators, it improves communication and enables other managers to understand why their own indicators are being affected.

Finally, in *S6* the visualization has to be adapted to the preferences of each particular user. In our case, the global manager explicitly stated his requirement to visualize the information in a geographic fashion. Depending on the availability of the information, these visualizations would need to be updated manually when new factories appear (lack of detailed geographical information) or could be integrated automatically by drawing them on the map according to their associated latitude and longitude values. The remaining charts and dashboards do not pose a problem since they have all the information required to show new data when new information is integrated into the system.

6. Conclusions and future work

In this paper we have presented a methodology with which to creating the support needed for decision making in global software development environments. We have covered all the steps involved in the process and shown how to apply them by means of our case study. During this process, we have identified the information needs of managers in both traditional software development and GSD. As a result, the application of our approach has allowed us to elaborate an information schema that provides visibility of the development process and supports most information needs requested by managers. Furthermore, as highlighted by both managers and developers, one of the most important aspects as regards enabling software development analytics is that they are easy to use [5]. In this respect, we have shown that by adapting BI for software development, it is possible only to integrate and present information that would otherwise be unrelated and scattered, but we can also enable decision makers to navigate and analyze the data easily by means of roll-up and drill down operations, dashboards, and other tools provided by BI platforms. Moreover, thanks to our approach, the dashboards obtained make managers aware of other indicators that may be affected by their decisions, thus preventing collateral effects from going unnoticed. Finally, we have discussed the generalizability and limitations of the approach and how it could be adapted to other specific cases. In the future, we intend to apply our approach to other companies, comparing the results obtained in order to obtain a more comprehensive set of results.

In addition, the work presented in this paper opens up new possibilities for future works. Firstly, implementing a platform that automates as much of the work presented as possible will enable software factories to quickly start analyzing their performance. This will in turn also enable a quantitative evaluation of the benefits of the solution, as it will enable us to define metrics with which to compare the performance of similar projects with and without support from our solution. Second, analysis techniques can be adapted in order to provide additional information by exploiting the relationships established in the schema. Finally, this work can be extended by moving from the management perspective to the developer perspective, and attempting to augment the information provided to developers while they are coding software. While managers often focus on increasing customer satisfaction, developers focus on improving the effectiveness and efficiency of the product, and a tool that helps them direct their focus on the parts requiring most attention could therefore be very useful for them.

Acknowledgments

This work has partially funded by the GEODAS-BI (TIN2012-37493-C03-03) and GEODAS-BC (TIN2012-37493-C03-01) projects from the Ministry of Economy and Competitiveness (MINECO) and the Fondo Europeo de Desarrollo Regional FEDER, SDGear (TSI-100104-2014-4, ITEA 2-Call 7, co-funded by the Ministerio de Industria, Energ y Turismo dentro del Plan Nacional de Investigacin Cientfica, Desarrollo e Innovacin Tecnolgica 2013-2016, and the APOSTD grant (APOSTD/2014/064) from the Generalitat Valenciana.

References

[1] D. Damian, D. Moitra, Guest editors' introduction: Global software development: How far have we come? IEEE Softw. 23 (5) (2006) 17–19.
[2] R. Prikladnicki, J. Nicolas Audy, R. Evaristo, Global software development in practice lessons learned, Softw. Process Improv. Prac. 8 (4) (2003) 267–281.
[3] H. Holmstrom, E. Conchúir, P. Agerfalk, B. Fitzgerald, Global software development challenges: A case study on temporal, geographical and socio-cultural distance, in: International Conference on Global Software Engineering, 2006. ICGSE'06, IEEE, 2006, pp. 3–11.
[4] M. Jiménez, M. Piattini, A. Vizcaíno, Challenges and improvements in distributed software development: a systematic review, Adv. Softw. Eng. 2009 (2009) 3.
[5] R. Buse, T. Zimmermann, Information needs for software development analytics, in: Proceedings of the 2012 International Conference on Software Engineering, IEEE Press, 2012, pp. 987–996.
[6] N. Nagappan, B. Murphy, V. Basili, The influence of organizational structure on software quality: an empirical case study, in: Proceedings of the 30th international conference on Software engineering, ACM, 2008, pp. 521–530.
[7] C. Bird, N. Nagappan, P. Devanbu, H. Gall, B. Murphy, Does distributed development affect software quality?: an empirical case study of windows vista, Commun. ACM 52 (8) (2009) 85–93.
[8] S. Krishna, S. Sahay, G. Walsham, Managing cross-cultural issues in global software outsourcing, Commun. ACM 47 (4) (2004) 62–66.
[9] D.-C. Gumm, Distribution dimensions in software development projects: a taxonomy, IEEE Softw. 23 (5) (2006) 45–51.
[10] R. Kaplan, D. Norton, B. Rugelsjoen, Managing alliances with the balanced scorecard, Harvard Business Review 88 (1--2) (2010) 114–120.
[11] D. Parmenter, Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs, John Wiley & Sons, Hoboken, New Jersey, 2010.
[12] E. Carmel, R. Agarwal, Tactical approaches for alleviating distance in global software development, IEEE Softw. 18 (2) (2001) 22–29.
[13] I. Richardson, V. Casey, J. Burton, F. McCaffery, Global software engineering: A software process approach, in: Collaborative Software Engineering, Springer, 2010, pp. 35–56.
[14] R. Kimball, M. Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, John Wiley & Sons, Hoboken, New Jersey, 2011.
[15] P. Vassiliadis, A. Simitsis, S. Skiadopoulos, Conceptual modeling for etl processes, in: Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP, ACM, 2002, pp. 14–21.
[16] W. Eckerson, Performance Dashboards: Measuring, Monitoring, and Managing your Business, John Wiley & Sons, New York, 2010.
[17] J. Portillo-Rodríguez, A. Vizcaíno, M. Piattini, S. Beecham, Tools used in global software engineering: A systematic mapping review, Inf. Softw. Technol. 54 (7) (2012) 663–685.
[18] A. Vizcaíno, F. García, J.C. Villar, M. Piattini, J. Portillo, Applying q-methodology to analyse the success factors in gsd, Inf. Softw. Technol. 55 (7) (2013) 1200–1211.
[19] R. Burkhard, M. Meier, Tube map: Evaluation of a visual metaphor for interfunctional communication of complex projects, in: Proceedings of I-Know, 4, 2004, pp. 449–456.
[20] I. Da Silva, M. Alvim, R. Ripley, A. Sarma, C. Werner, A. Van Der Hoek, Designing software cockpits for coordinating distributed software development, in: 1st Workshop on Measurement-based Cockpits for Distributed Software and Systems Engineering Projects, 2007, pp. 14–19.
[21] F. García, M.Á. Moraga, M. Serrano, M. Piattini, Visualisation environment for global software development management, IET Softw. 9 (2) (2015) 51–64.
[22] R. Wettel, M. Lanza, Visualizing software systems as cities, in: 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, 2007. VISSOFT 2007, IEEE, 2007, pp. 92–99.
[23] F. García, M. Moraga, M. Serrano, M. Piattini, Visualizing software systems as cities, 2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, 2007.In Press (In: IET Software).
[24] A. Sarma, A. Van Der Hoek, Towards awareness in the large, in: International Conference on Global Software Engineering, 2006. ICGSE'06, IEEE, 2006, pp. 127–131.
[25] M.-A. Storey, D. Čubranić, D. German, On the use of visualization to support awareness of human activities in software development: a survey and a framework, in: Proceedings of the 2005 ACM Symposium on Software Visualization, ACM, 2005, pp. 193–202.
[26] E. Ye, L.A. Neiman, H. Dinh, C. Liu, Secondwatch: A workspace awareness tool based on a 3-d virtual world, in: 31st International Conference on Software Engineering-Companion Volume, 2009. ICSE-Companion 2009, IEEE, 2009, pp. 291–294.
[27] A. Borici, K. Blincoe, A. Schroter, G. Valetto, D. Damian, Proxiscientia: Toward real-time visualization of task and developer dependencies in collaborating software development teams, in: 5th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, 2012, pp. 5–11.
[28] R. Kaplan, D. Norton, Strategy Maps: Converting Intangible Assets into Tangible Outcomes, Harvard Business Press, Boston, 2004.
[29] J. Horkoff, D. Barone, L. Jiang, E. Yu, D. Amyot, A. Borgida, J. Mylopoulos, Strategic business modeling: representation and reasoning, Softw. Syst. Model. (2012) 1–27.
[30] O. Yigitbasioglu, O. Velcu, A review of dashboards in performance management: Implications for design and research, Int. J. Account. Inf. Syst. 13 (1) (2012) 41–59.
[31] J.-N. Mazón, J. Pardillo, J. Trujillo, A model-driven goal-oriented requirement engineering approach for data warehouses, in: Advances in Conceptual Modeling–Foundations and Applications, Springer, 2007, pp. 255–264.
[32] S. Luján-Mora, J. Trujillo, I.-Y. Song, A uml profile for multidimensional modeling in data warehouses, Data Knowl. Eng. 59 (3) (2006) 725–769.
[33] B. Livieri, M. Bochicchio, Information systems and performance management for collaborative enterprises: a proposal, in: 26th CAiSE FORUM, 2014, pp. 1–8.
[34] U. Bititci, K. Mendibil, V. Martinez, P. Albores, Measuring and managing performance in extended enterprises, Int. J. Oper. Prod. Manage. 25 (4) (2005) 333–353.
[35] A. Maté, J. Trujillo, J. Mylopoulos, Conceptualizing and specifying key performance indicators in business strategy models, in: Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research, IBM Corp, 2012, pp. 102–115.
[36] Object Management Group, Business Motivation Model (BMM), 2014, (http://www.omg.org/spec/BMM/1.2/).
[37] E. Yu, Modelling strategic relationships for process reengineering, Soc. Model. Require. Eng. 11 (2011) 2011.
[38] D. Amyot, Introduction to the user requirements notation: learning by example, Comput. Netw. 42 (3) (2003) 285–301.
[39] P. Giorgini, S. Rizzi, M. Garzetti, Grand: A goal-oriented approach to requirement analysis in data warehouses, Decis. Support Syst. 45 (1) (2008) 4–21.
[40] R. Rodriguez, J.A. Saiz, A. Bas, Quantitative relationships between key performance indicators for supporting decision-making processes, Comput. Ind. 60 (2) (2009) 104–113.
[41] G. Dasgupta, Y. Shrinivasan, T. Nayak, J. Nallacherry, Optimal strategy for proactive service delivery management using inter-kpi influence relationships, in: Service-Oriented Computing, Springer, 2013, pp. 131–145.
[42] C.-Y. Lin, N. Cao, S. Liu, S. Papadimitriou, J. Sun, X. Yan, Smallblue: Social network analysis for expertise search and collective intelligence, in: IEEE 25th International Conference on Data Engineering, 2009. ICDE'09, IEEE, 2009, pp. 1483–1486.
[43] P. Johnson, H. Kou, M. Paulding, Q. Zhang, A. Kagawa, T. Yamashita, Improving software development management through software project telemetry, IEEE Softw. 22 (4) (2005) 76–85.
[44] M. Golfarelli, D. Maio, S. Rizzi, The dimensional fact model: a conceptual model for data warehouses, Int. J. Cooperative Inf. Syst. 7 (02n03) (1998) 215–247.
[45] C. Sapia, M. Blaschka, G. Höfling, B. Dinter, Extending the e/r model for the multidimensional paradigm, in: Advances in Database Technologies, Springer, 1999, pp. 105–116.
[46] A. Sillitti, A. Janes, G. Succi, T. Vernazza, Collecting, integrating and analyzing software metrics and personal software process data, in: Proceedings of 29th Euromicro Conference, 2003, IEEE, 2003, pp. 336–342.
[47] Pentaho Corporation, Pentaho business analytics, 2015, (http://www.pentaho.com).