International Journal of Information Systems and Project Management

Volume 1 | Number 3 Article 3

2013

Scorecard and KPIs for monitoring software factories effectiveness in the financial sector

Vicente Montequín
University of Oviedo

César Pérez
University of Oviedo

Francisco Fernández
University of Oviedo

Joaquín Balsera
University of Oviedo

International Journal of
Information Systems and Project Management

ISSN (print):2182-7796, ISSN (online):2182-7788, ISSN (cd-rom):2182-780X

Available online at www.sciencesphere.org/ijispm

Scorecard and KPIs for monitoring software factories effectiveness in the financial sector

Vicente Rodríguez Montequín
Project Engineering Area, University of Oviedo
C/Independencia 13, 33004 Oviedo, Spain
www.shortbio.net/ montequi@api.uniovi.es

Francisco Ortega Fernández
Project Engineering Area, University of Oviedo
C/Independencia 13, 33004 Oviedo, Spain
www.shortbio.net/fran@api.uniovi.es

César Álvarez Pérez
Project Engineering Area, University of Oviedo
C/Independencia 13, 33004 Oviedo, Spain
www.shortbio.net/cesaralvarezcom@gmail.com

Joaquín Villanueva Balsera
Project Engineering Area, University of Oviedo
C/Independencia 13, 33004 Oviedo, Spain
www.shortbio.net/ balsera@api.uniovi.es

Abstract:
Financial corporations and especially banking institutions have important needs concerning to the development of software around their business core. The software, that traditionally had been developed in house by the IT departments, is now usually outsourced to IT firms. These firms work under the software factories model. An important feature within this sector is that usually the financial groups keep the ownership of these firms because the strategic value of the software for the core business. These firms work almost exclusively for the owner financial group developing their software, but they have to demonstrate that they are so profitable and competitive like any other firm. The organizational structure of these firms has some differential features. Top level tasks (software design and project management) are usually performed by the IT firm but the development is usually subcontracted to other software companies. Although financial corporations have always paid a special interest to investing in management and organizational policies to improve their efficiency, there have being always an important lack regarding to the control and monitoring of the software projects. They do not have suitable tools for monitoring actual process effectiveness. Adapting scorecards to this environment could be a useful tool for monitoring and improvement the process. Scorecard could here be used both as a tool for internal effectiveness measurement as well as externally, presenting sustainability indicators for the shareholders, the financial institutions. This paper aims to identify and define a collection of Key Performance Indicators which permit effectiveness to be improved under this context, focusing in the specific supply-chain model given by owner (financial group), software factory and software developers (subcontracted).

Keywords:
Software Factory; financial software development; Balanced Scorecard; KPIs; efficiency.

DOI: 10.12821/ijispm010302

Manuscript received: 14 May 2013
Manuscript accepted: 28 June 2013

1. Introduction

Financial corporations and especially banking institutions require a great deal of software development. The development of business applications represents more than 50% of the IT budget [1]. Traditionally, banking is the sector which not only requires the highest level of software development, but which also has a higher tendency to externalize the development of projects of that type. They had large IT departments where the business software was developed, but following the modern managerial trends, now it is outsourced to external firms. Most of these firms were created by the financial groups in the beginning of the two thousand and usually they have kept the ownership and control because the extreme importance of the software for the core business. So these firms work also exclusively for the matrix financial group, but they have to demonstrate to be more competitive than other firms. In addition, financial institutions have pushed these firms to reduce cost, which has obliged them to search for new productive models in order to remain competitive. Most of them have followed the “software factory” model. A software factory applies manufacturing techniques and principles to software development to mimic the benefits of traditional manufacturing.

Financial sector have always paid a special interest to investing in management and organizational policies to improve their efficiency. Finance and banking are one of the sectors in which more effort has been applied for measuring productivity in all of their departments. However, their weakest point is still the measurement and monitoring of the productivity of the development of the software projects that support its main core activity. One of the reasons why this productivity is not being suitable analyzed is due to the software nature of intangible. This lack has been also transferred to the software firms. The firms pay according to the estimated effort it takes to be produced and not according to the quantity and quality of the actual software produced. This model eventually involves a further increase in costs for the project.

Usually these firms perform the top level design and control of the projects, but the development is usually subcontracted to other companies specialized in software development. This structure appears as a natural reaction from the financial groups for keeping the control of their software. Subcontracting the development is the usual strategy for minimizing costs. Nowadays, the financial groups are pushing their software firms to cut cost, who are also pressing their subcontractors to reduce costs. As reaction of this context, the subcontractors have explored several strategies for cutting costs, some of them lying in a loss of quality. For instance, they have created offshore software factories. The creation of these software factories is justified as a way of increase productivity and efficiency. However, the reality could not be more different. Realistically, the truth of the matter is that there are evidences which tell us that there is a reduction in costs by employing a cheaper workforce but with a loss of quality in most cases. These facts remark the need of controlling and monitoring systems from the side of the contractor, the software factories of the financial groups.

The explored way in this work for improving this process is based on scorecard approach. Key Performance Indicators (KPIs) accommodated in scorecards is an usual tool within the strategic management, but is rarely used effectively in the field of software projects, which are more commonly evaluated by productivity assessment metrics linked to the generation of code as the "number of lines of code" or "function points" [2]. This work aims to identify and define a collection of Key Performance Indicators which permit effectiveness to be measured in these types of organizations in this supply-chain context. The different key indicators are conveniently set in a specific scorecard that allows decision-making associated with top level project portfolio management.

The context where we study was carried out is introduced in the background section. Special consideration is done regarding to the structure of these kind of Software Factories and their features. The basis of the Balanced Scorecard is also introduced. Then the suggested model is described in section 3, grouping the identified KPIs in six perspectives: financial; customer; human resources and growth; productivity; software quality-quality in use; software quality-product quality. Each perspective is summarized in a table including the KPI formulation. Finally conclusions and future work is described in the final section.

2. Background

The Software Factory is generally defined as the workplace where software is developed using techniques and principles associated with traditional industrial production. The Software Industry needs to become similar to the industrialized software manufacturing process in order that it is able to provide everything the market expects today related to efficiency, fast delivery and quality in all industrial products. Such transformation requires significant changes in the industry, but the sooner they are carried out, them the actors will gain larger competitive advantages [3]. Software engineering seems not to have made use of the latest technological advanced in software production [4].

Although the term Software factory was first introduced in 1968, it is due to the recent social, economic and technological circumstances that the term Software factory becomes notorious again among the software sector [5]. Authors like Greenfield et al. [6], from Microsoft, use the concept of the Software factory as a structured collection of related software assets that assists in producing computer software applications or software components according to specific, externally-defined end-user requirements through an assembly process. The aim of a Software factory is the improvement of productivity and quality, scale production and the maintenance of software development control [7]. The Software Industry is turning into a new business model, where the different centers work together to achieve objectives and developments.

The term software factory had already been used by 1975 when Harvey Bratman and Terry Court, from the System Development Corporation, described in one of their papers the challenge of developing an integrated set of software development tools to support a disciplined and repetitive approach to software development. This effort was a part of a large program to increase software reliability and control software production cost using standard engineering techniques. This study, which attempted in part to correlate program productivity and experience, identified the lack of a methodical and well founded body of knowledge on the software development process. The most significant problems that contributed to this shortcoming were:

- Lack of discipline and repetitiveness;
- Lack of development visibility;
- Lack of design and verification tools;
- Lack of software reusability.

Despite the introduction and use of new techniques and methodologies about software development, the fact is that for more than 30 years after the introduction of software factory concept, and even today, many of the initial issues and other problems associated with them (systematic reuse, assembly development, model-driven development and process frameworks) cannot been resolved yet [6].

Nowadays, there is a great deal of pressure on software delivery organizations to produce more software at a faster rate in the context of extreme cost pressure and growing globalization of the software delivery organization. The concept of a software factory is beginning to emerge as one way to address these challenges. The principles of software factory are necessary for enterprise software delivery to propose faster deliveries, reduce cost and increase software quality [8].

Many companies have experienced a great deal of change over the past few years due to evolution of the business environment, financial upheavals, societal changes and technical advancement. The key to addressing these changes has been analysis of the core business processes to see how they can be refined and optimized, followed by a restructure of those business processes to better meet the new context.

At the same time, IT groups have been forced to lower operating costs across the organization. The direct implication is that they must not only minimize waste and inefficiency, but increase productivity and relevance to the businesses they serve.

This combination of business process restructuring and close focus on delivery efficiency have been seen in many business domains, and has resulted in techniques such as *lean manufacturing*, *supply-chain management*, and *product line engineering*. The application of these ideas in software delivery is what we refer to here as a *software factory approach* to enterprise software delivery [9],[10].

Although there are different development standards to measure in-house development, there is little standardization in evaluating supply-chains and software factories. Standard approaches such as function point analysis and defect density can be applied, but in practice they appear inadequate. With more complex supply-chain delivery models becoming more common, we need metrics that help us address different questions:

- Which software factory is more productive and efficient?
- Which software factory experiences a delay when delivering their products?
- What is the quality of the software delivered?

These and other many other measures need to be defined and an automatic mechanism to collect these metrics must be implemented to help compare results across external providers in real time [8].
This paper works about the presented questions, and its objective is to develop a scorecard and a set of suitable indicators to provide managers answers about the productivity and efficiency in a software factory oriented towards financial sector.

2.1 The software factory architecture, according the software factory processes

The main features of software factories for financial software development are usually the next:

- They work almost exclusively for the owner group;
- These usually have a greater demand of requests;
- The software development process is usually subcontracted to several software companies. However, due to strategic reasons, this does not apply to top level processes such as functional specifications and project management;
- Usually the payments to subcontractors are made according to the number of hours budgeted, and not by the number of hours actually performed.

The usual organizational structure for software factory orientated financial software should not differ substantially from what is presented below, which include the following processes:

- Demand Management, which aims to collect top level user requirements and establish methods for prioritizing demands;
- Functional Analysis, which transforms the identified top level user requirements into functional requirements;
- Technical Analysis, which is responsible for the technical details of the functional specifications which must be implemented;
- Development, which performs the development, construction and assembling of the requested requirements;
- Testing, which has to validate everything that has been implemented;
- Production, which performs the customer deployment;
- Quality, which assesses the global system quality.

In the proposed schema, the development process is subcontracted to several software development firms. On the other hand, the rest of the processes are under the software factory control. Fig. 1 illustrates the usual process followed in this scheme of operation, which is similar as [4].

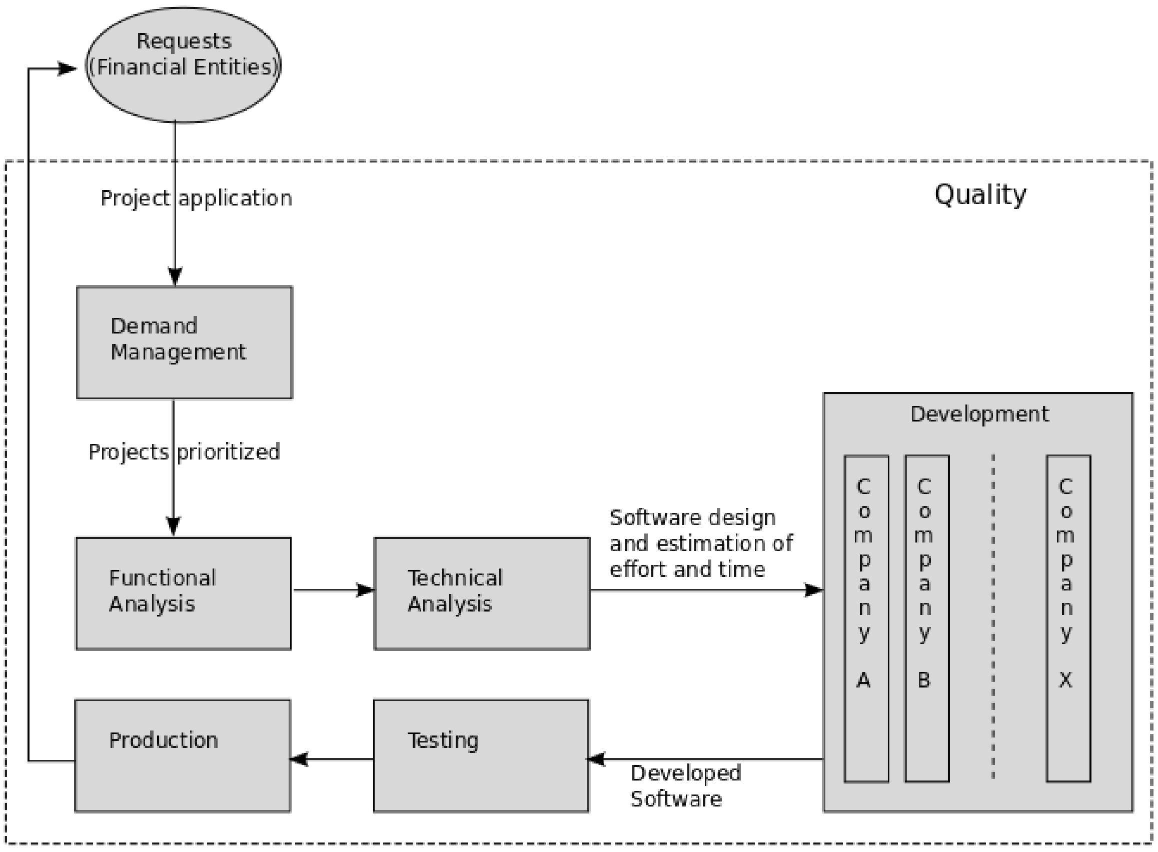

Fig. 1. Map of processes of model for software factory oriented financial sector

The system described involves several different levels of control to evaluate the efficiency. The measuring productivity of the development process is based on metrics as function points, number of defects, etc. All of these methods have been widely studied in the literature, though it is the work of Fenton [11] that most of these practices are based on. These controls are made into development software areas by the outsourced firms.

The proposed scorecard presented here does not incorporate this level, since within the presented context it is the measurement in the top level that is much more interesting. At this level we need to focus on the overall project portfolio and productivity ratios of the whole system. Some of the questions that the scorecard must find and provide answers for are as follows:

- What is the software factory Performance?
- What are the cost and time deviations compared to the estimations?
- What is the employees’ productivity?
- What is the level of customer satisfaction?
- What is the value provided by developers and the rest of the human capital?

2.2 Using scorecard for the efficiency measurement

Organizations have used systems consisting of a mix of financial and non-financial measures to track progress for quite some time. In the mid-1990s, a new method emerged. In the new method, measures are selected based on a set of “strategic objectives” plotted on a “strategic linkage model” or “strategy map”. With this new approach, the strategic objectives are distributed across the four measurement perspectives to form a visual presentation of strategy and measures. This method was named “Balanced Scorecard”. The Balanced Scorecard is a strategy performance management tool - a semi-standard structured report, supported by design methods and automation tools, which can be used by managers to keep track of the execution of activities by the staff within their control and to monitor the consequences arising from these actions. The Harvard Business Review, in its 75th Anniversary issue, cites the Balanced Scorecard as being one of the 15 most important management concepts to have been introduced via articles in the magazine. Since its introduction in 1992, the Balanced Scorecard has featured in a wealth of academic and practitioner papers, and has been the subject of several bestselling books.

The Balanced Scorecard was originally proposed as an approach to performance measurement that combined traditional financial measures with non-financial measures to provide managers with richer and more relevant information about organizational performance, particularly with regard to key strategic goals [12].

By encouraging managers to focus on a limited number of measures drawn from four “perspectives”, the original Balanced Scorecard aimed to encourage clarity and utility. Over time Balanced Scorecard has developed to form the center-piece of a strategic communication and performance measurement framework that helps management teams articulate, communicate and monitor the implementation of strategy using a system interlinked with the long-term destination of the organization. More recent insights suggest that a successful Balanced Scorecard implementation will require adjustments to be made to other management processes used by the enterprise. Only in so doing will the Balanced Scorecard be able to become a central part of a “strategic management framework” [13].

Some of the benefits of using the Balanced Scorecard are:

- It improves the bottom line by reducing process cost and improving productivity and mission effectiveness;
- Measurement of process efficiency provides a rational basis for selecting what business process improvements to make first;
- It allows managers to identify best practices in an organization and expand their usage elsewhere;
- The visibility provided by a measurement system supports better and faster budget decisions and control of processes in the organization. This means it can reduce risk;
- Visibility provides accountability and incentives based on real data, not anecdotes and subjective judgments. This serves for reinforcement and the motivation that comes from competition.

The Balanced Scorecard allows manager to look at the business from four important perspectives: Financial; Customer; Internal Business; and Innovation and Learning. It provides answers to four basic questions:

- How do customers see us?
- What must we excel at?
- Can we continue to improve and create value?
- How do we look to shareholders?

Each question requires a set of indicators to be measured and analyzed to answer these questions.

Since the Balanced Scorecard was popularized in the early 1990s, a large number of alternatives to the original “four box” balanced scorecard promoted by Kaplan and Norton in their various articles and books have emerged and new variations appeared for specific sectors, like pharmaceutical, technology, engineering, and software companies. But, in general, a specific set of indicators have to be chosen for each specific organization.

2.3 KPIs, a metric of performance measurement

Key Performance Indicators (KPIs) help organizations understand how well they are performing in relation to their strategic goals and objectives. In the broadest sense, a KPI provides the most important performance information that enables organizations or their stakeholders to understand whether the organization is on track or not. KPIs serve to reduce the complex nature of organizational performance to a small number of key indicators in order to make it more manageable [14].

KPIs or Key Performance Indicators are the selected measures that provide visibility into the performance of a business and enable decision makers to take action in achieving the desired outcomes. Organizations that measure performance identity the handful of critical success factors that comprise every strategic objective [15].

Typically, KPIs are monitored and distributed in dashboards or scorecards to provide everyone in the organization with an understanding of the strategy implementation progress. KPI utilization enables learning and improvement on critical operations, capabilities and processes across business areas [16].

In order to be evaluated, KPIs are linked to target values, so that the value of the measure can be assessed as meeting expectations or not.

3. Model description

This work shows a set of KPIs in order to establish into the Balanced Scorecard here proposed.

KPIs are distributed according the four perspectives of the Balanced Scorecard. KPIs about Financial, Customers and Innovation and Learning perspective are usual and commonly used. KPIs about Internal Business are distributed in Productivity and Software Quality sub-perspectives.

3.1 Perspectives

3.1.1 Financial perspective KPIs

In the private sector, these measures have typically focused on profit and market share. Managers must answer the question: How do satisfy the financial expectations of our stakeholders?

The set of proposed KPIs are:

- ROI;
- Added Value;
- Efficiency;
- % Development Software Cost;
- % External / Internal Client Sales Revenue.

Table 1 shows the financial KPIs into the proposed balanced scorecard.

3.1.2 Customer perspective KPIs

Managers must know if their organization is satisfying customer needs. They must determine the answer to the question: How do customers perceive us?

The set of proposed KPIs are:

- Customer Satisfaction Index;
- Service Level Agreements.

Table 2 shows the customer KPIs into the proposed balanced scorecard.

3.1.3 Innovation and learning perspective KPIs

An organization's ability to innovate, improve and learn ties directly to its value as an organization. Managers must answer the question: Can we continue to improve and create value for our services?

The set of proposed KPIs are:

- Staff turnover;
- Outsourcing firms turnover;
- Human Capital.

Table 3 shows the innovation and learning KPIs into the proposed balanced scorecard.

3.1.4 Internal business perspective KPIs

Managers need to focus on those critical internal operations that enable them to deliver their work program. They must answer the question: What must we excel at? This Perspective is the largest and more important in this work, and it is subdivided in other two perspectives: Productivity and Software Quality perspectives.

3.1.4.1 Productivity sub-perspective KPIs

Productivity is the ratio of output to inputs in production; it is a measure of the efficiency of production. Productivity growth is important to all the organizations because more real income means that the organization can meet its (perhaps growing) obligations to customers, suppliers, workers, shareholders, taxes and still remain competitive or even improve its competitiveness in the market place.

Furthermore, it is necessary to measure cost and time deviation. Earned Value is a technique for measuring project performance and progress. It has the ability to combine measurements of scope, schedule and cost.

The set of proposed KPIs are:

- Performance;
- In/Out Request Rate;
- Time Deviation (scheduled error);
- Cost Deviation;
- Employee Productivity;
- % Calls to Reuse Software Components;
- % Error.

Table 4 shows the productivity KPIs into the proposed balanced scorecard.

3.1.4.2 Software quality sub-perspective KPIs

The ISO/IEC 25000:2005 provides guidance for the use of the new series of International Standards named Software product Quality Requirements and Evaluation (SQuaRE). The purpose of this guide is to provide a general overview of SQuaRE contents, common reference models and definitions, as well as the relationship among the documents, allowing users of this guide a good understanding of those series of International Standards, according to their purpose of use.

The international standard SQuaRE provides [17]:

- Terms and definitions;
- Reference models;
- General guide;
- Individual division guides;
- Standards for requirements specification, planning and management, measurement and evaluation purposes;
- The cost and time deviations compared to the estimations.

Furthermore, the ISO/IEC 25010 defines characteristics and sub characteristics that provide a consistent terminology for specifying, measuring and evaluating system and software product quality. Also a set of quality characteristics against which stated quality requirements can be compared for completeness. In combination with other standards the ISO 25010 can be used as framework to support different processes, e.g. requirements definition or software quality evaluation[18].

The standard ISO/IEC 25010 defines two different models:

- A quality in use model composed of five characteristics (some of which are further subdivided into sub characteristics) that relate to the outcome of interaction when a product is used in a particular context of use. This system model is applicable to the complete human-computer system, including both computer systems in use and software products in use. Table 5 shows the quality in use KPIs into the proposed balanced scorecard;
- A product quality model composed of eight characteristics (which are further subdivided into sub characteristics) that relate to static properties of software and dynamic properties of the computer system. The model is applicable to both computer systems and software products. Table 6 shows the product quality model KPIs into the proposed balanced scorecard.

The characteristics defined by both models are relevant to all software products and computer systems. The characteristics and sub characteristics provide consistent terminology for specifying, measuring and evaluating system and software product quality. They also provide a set of quality characteristics against which stated quality requirements can be compared for completeness.

It is important that the quality characteristics are specified, measured, and evaluated whenever possible using validated or widely accepted measures and measurement methods. The quality models in this International Standard can be used to identify relevant quality characteristics that can be further used to establish requirements, their criteria for satisfaction and the corresponding measures.

Quality in use is the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use. The properties of quality in use are categorized into five characteristics: effectiveness; efficiency; satisfaction; freedom from risk; and context coverage.

The product quality model categorizes product quality properties into eight characteristics (functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability and portability). Each characteristic is composed of a set of related sub characteristics.

3.2 Proposed Scorecard and KPIs

The following is the proposed scorecard objective of this work, and it includes a specific set of KPIs. The KPIs are categorized according to four different approaches or perspectives: financial; customer; human resources and growth; and internal processes (productivity and quality). The proposed scorecard is linked to the new work processes scheme in this software factory, and it includes specific and nonspecific KPIs for its use. The model here presented is aimed to assess the project management portfolios and it is only applied to software factories for financial institutions.

Table 1. Financial perspective for the proposed scorecard

Financial Perspective		
ROI (Return of Investment)	What is the return of investment?	$\text{ROI} := \left(\frac{\text{Net Profit}}{\text{Shareholders Investment}}\right) \text{x } 100$
Added Value	What is the added value provided by the human capital?	$\text{A.V.} := \frac{\text{Sales Revenue} - (\text{Total Cost} - \text{Staff Cost})}{\text{Number of Staff}}$
Efficiency	What is the efficiency of the activity?	$\text{Efficiency} := \left(\frac{\text{Structure Cost}}{\text{Income}}\right) \text{x } 100$
% SW Development Cost	What is the ratio between SW development cost and total cost?	$\%\ \text{SW.D.C.} := \left(\frac{\text{Software Development Cost}}{\text{Total Cost}}\right) \text{x } 100$
% External / Internal Client Sales Revenue	What is the ratio between external and internal sales revenue?	$\%\ \frac{\text{E}}{\text{I}}\text{S.R.} := \left(\frac{\text{Software Development Cost}}{\text{Total Cost}}\right) \text{x } 100$

Table 2. Customer perspective for the proposed scorecard

Customer Perspective		
Customer Satisfaction Index	What is the level of customer satisfaction?	$\text{CSI} := \sum \left(\frac{\text{Indicator Value}}{\text{Maximum Value Indicator}} \text{ x Weight}\right)$
Service Level Agreements	What is the level of compliance with the SLA?	$\text{SLA} := \sum \left[\left(\frac{\text{SLA Value}}{\text{Operative Level Agreed for SLA}}\right) \text{x Weight}\right]$

Table 3. Human resources and growth perspective for the proposed scorecard

Human Resources and Growth Perspective		
Staff Turnover	What is the level of stability?	$\text{Staff Turnover} := \frac{\left(\frac{\text{Recruitments} + \text{Layoffs}}{2}\right)}{\text{Average Employee}} \text{ x } 100$
Outsourcing firms Turnover	What is the level of outsourcing firms' stability?	$\text{O.F.Turnover} := \frac{\left(\frac{\text{Recruitments outsourcing firms} + \text{Layoffs outsourcing firms}}{2}\right)}{\text{Average Outsourced Firms}} \text{ x } 100$
Human Capital	What is the most optimal task for the employees?	$\text{H.C.} := \text{MAX}\left(\sum_{\text{Employee}=1}^{\text{Employee}=\text{N}} \text{Training x Position in SF x Personal Factors}\right)$

Table 4. Internal business perspective - Productivity for the proposed scorecard

Internal Business Perspective – Productivity		
Performance	Performance	$\text{Performance} := \left(\frac{\sum \text{Working Hours Budgeted by Request}}{\sum \text{Working Hours Performed by Request}}\right) \text{x } 100$
In/Out Requests Rate	Increase in hours of Requests over the previous year	$\frac{\text{In}}{\text{Out}} \text{Requests} := \left(\frac{\sum \text{Working Hours Completed by Request}}{\sum \text{Working Hours Required by Request}}\right) \text{x } 100$
Time Deviation (Scheduled Error)	What is the time deviation error?	$\text{T. D.} := \sum (\text{Working Hours Performed} - \text{Working Hours Budgeted})$
Cost Deviation	What would be the minimum cost for the work performed?	$\text{C. D.} := \sum_{\text{Si}>0} (\text{Working Hours Budgeted} - \text{Working Hours Performed})$
Employee Productivity	What is the average number of hours allocated to each employee in a year?	$\text{E. P.} := \left(\frac{\sum \text{Working Hours Budgeted by Request}}{\text{Number of Employee}}\right)$
Calls to Reuse Software Components	What is the level of Software Components Reused?	$\text{SW Reuse} := \left(\frac{\sum \text{Calls to Reuse SW Components by Request}}{\sum \text{Calls to SW Components by Request}}\right) \text{x } 100$
% Error	Errors detected in testing process	$\text{\% Error} := \left(\frac{\sum \text{Increased Working Hours for Errors by Request}}{\sum \text{Working Hours Charged by Request}}\right) \text{x } 100$

Table 5. Internal business perspective - Software quality – Quality in use

Internal Business Perspective – Software Quality - Quality in Use (ISO/IEC 25010:2011)		
Effectiveness	Accuracy and completeness with which users achieve specified goals	Effectiveness
Efficiency	Resources expended in relation to the accuracy and completeness with which users achieve goals	Efficiency
Satisfaction	Degree to which user needs are satisfied when a product or system is used in a specific context of use	Usefulness; Trust; Pleasure; Comfort
Freedom from risk	Degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment	Economic risk mitigation; Health and safety risk mitigation; Environmental risk mitigation
Context Coverage	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified context of use and in contexts beyond those initially explicitly identified	Context completeness; Flexibility

Table 6. Internal business perspective - Software quality – Product quality model

Internal Business Perspective – Software Quality – Product Quality Model (ISO/IEC 25010:2011)		
Functional Suitability	Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions	Functional completeness; Functional correctness; Functional appropriateness
Performance efficiency	Performance relative to the amount of resources used under stated conditions	Time-behavior; Resource utilization; Capacity
Compatibility	Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment	Co-existence, Interoperability
Usability	Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use	Appropriateness; recognizability; Learnability; Operability; User error protection; User interface aesthetics; Accessibility
Reliability	Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time	Maturity; Availability; Fault tolerance; Recoverability
Security	Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization	Confidentiality; Integrity; Non-repudiation; Accountability; Authenticity
Maintainability	Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers	Modularity; Reusability; Analyzability; Modifiability; Testability
Portability	Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another	Adaptability; Installability; Replaceability

3.3 Application of the scorecard to a specific case study

The presented model is being currently implemented in a software factory associated to an important Spanish Banking Group. Due to terms of confidentiality, its name cannot be published here. The organization follows the scheme described in Figure 1, outsourcing to several suppliers for the software development process. They have a maturity level 3 according to the standard CMMI-DEV. This is an organization that has suffered a major transformation in the past two years, due to former traditional projects the company functions by adopting a closed process management approach. The proposed scorecard serves as a comparison of the performance of the new model and the former one. Due to its recent application, there is only data available for a few indicators, but an evaluation of some internal processes KPIs showed that performance increases when there are several different companies working in competition and when the cost of the projects is linked to budgeted time and it is not linked to performed time. The software factory productivity increased from 92% to 107%, decreasing the project duration by 7% on average in the last year and a half.

4. Conclusion

This paper presents a scorecard for a software factory suitable for the financial sector. This sector has externalized the development process, but they have kept the top level processes due to strategic reasons. This kind of organizations has usually two different wok lines: providing services and developing projects. This study is coped only to the project side, considering the project portfolio of the organization. The scorecard is intended for the analysis by management levels, allowing the top managers to take the appropriate decisions. The proposed scorecard does not incorporate the traditional software engineering metrics, since within the presented context the measurement in the top level is much more interesting. Considering that the proposed model is aimed at the Software Factory top level managers, global view indicators have been prioritized rather than low level traditional software engineering quality indicators. The use of this Scorecard does not exclude the use of additional Scorecards designed for other specific areas, like developing or testing. According to this, four different approaches were identified: financial; customers; innovation and learning; and internal processes. The model is being tested in a real context in a Spanish Software Factory. Although the gathered experience is being very positive, it is anticipated that an evolution of the model will take place according to the feedback and the results provided. The adoption of the model is still very recent so no definitive conclusions can be drawn. It is possible that in a short period of time the management of this factory will make changes in other departments as it has done in the development processes. It should be also considered the limitation that the model is being tested only in one organization, so the proposed model could be biased to the internal operative of this organization. In addition, the model could be also affected by the local business culture. After this initial experience, the model will be extended to other Software Factories as further work.

The results provided by this work will constitute the bases for a further study detecting process weakness in this kind of organizations and looking for a new model that could improve their efficiency. The new model could establish a new process standard that brings them closer to the Software Factory paradigm.

References

[1] R. f. Asprón, “Medir la productividad del desarrollo de software en Banca,” *Financial Tech Magazine*, vol. *217*, July, 2010.

[2] J. Garzás and D. Cabrero, *El valor y el retorno de la inversión en TSI. En El Gobierno de las TSI,* Ra-ma, 2007.

[3] E. A. Mikel, “The Software Factories,” *Dyna (Bilbao)*, vol. *82*, no. *6*, pp. *330–333*, 2007.

[4] R. P. Valderrama, A. C. Cruz, and I. P. Valderrama, “An Approach toward a Software Factory for the Development of Educational Materials under the Paradigm of WBE,” *Interdiciplinary Journal of E-Learning and Learning Objects*, vol. *7*, 2011.

[5] J. Garzás and M. Piattini, *Factorías de Software: Experiencias, tecnologías y organización*, Ra-ma, 2007.

[6] J. Greenfield and K. Short, “Software Factories Assembling Applications with Patterns, Models, Frameworks and Tools,” *Microsoft Corporation*, 2004.

[7] F. Siqueira, G. Barbaran, and J. Becerra, “A software factory for education in software engineering,” in *IEEE 21st Conference on Software Engineering Education and Training*, South Carolina, USA, 2008.

[8] A. W. Brown, A. L. Mancisidor, and L. Reyes Oliva, *Practical Experiences with Software Factory Approaches In Enterprise Software Delivery*, IARIA, 2011.

[9] M. Poppendieck and T. Poppendieck, *Lean software Sevelopment: An agile toolkit*, Addison Wesley, 2003.

[10] M. Hotle and S. Landry, *Application Delivery and Support Organizational Archetypes: the Software Factory*, *Gartnet Research Report G00167531*, May, 2009.

[11] N. E. Fenton and S. L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*, PWS Publishing Co, 1998.

[12] R. S. Kaplan and D. P. Norton, “The Balanced Scorecard - Measures that Drive Performance,” *Harvard Business Review*, January-February, 1992.

[13] R. S. Kaplan and D. Norton, “Using the Balanced Scorecard as a strategic management system,” *Harvard Business Review*, 1996.

[14] B. Marr, *How to design Key Performance Indicators, Management Case Study*, The Advanced Performance Institute, 2010.

[15] R. S. Kaplan, *Measuring Performance*, Harvard Business School Publishing, 2009.

[16] The KPI Institute. (2013). *Key Performance Indicators* [Online]. Available: http://www.smartkpis.com/.

[17] Joint Technical Committee ISO/IEC JTC 1/SC7, *ISO/IEC 25000:2005 Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*.

[18] Joint Technical Committee ISO/IEC JTC 1/SC7, *ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*.

Biographical notes

Vicente Rodríguez Montequín
Vicente Rodríguez Montequín, B.S., M.S., Ph.D, is a Project Management professor at the University of Oviedo, Spain. He is also professor-tutor at UNED. He is a member of the Asociación Española de Ingeniería de Proyectos (AEIPRO) and Certified Project Management Associate by IPMA (International Project Management Association). He has participated actively in several international projects since 1997 and he has supervised several Masters and Doctoral dissertations in the Project Management field. His main research is aimed at the Project Management and process improvement field.

www.shortbio.net/montequi@api.uniovi.es

César Álvarez Pérez
Postgraduate in Project Management and Financial Entities Management, he has a degree in Software Engineering. Nowadays he studies Ph.D in Project Management and he usually collaborates with the Project Engineering Area of the Oviedo University about new ways to measure software development efficiency. He works as a director of strategic planning in a Spanish medium size financial entity and he is part of the organization workgroup of a related enterprise software factory.

www.shortbio.net/cesaralvarezcom@gmail.com

Francisco Ortega Fernández
Francisco Ortega Fernández, B.S., M.S., Ph.D, is a Project Management Full Professor at the University of Oviedo, Spain, as well as the group research coordinator. He is a member of the Asociación Española de Ingeniería de Proyectos (AEIPRO) and Certified Project Management Associate by IPMA (International Project Management Association). He has participated actively in several international projects since 1992 and he has supervised several Masters and Doctoral dissertations in the Project Management field.

www.shortbio.net/fran@api.uniovi.es

Joaquín Villanueva Balsera
Joaquín Villanueva Balsera, B.S., M.S., Ph.D, is a Project Management lecturer at the University of Oviedo, Spain. He is specialized in software cost estimation and Data Mining.

www.shortbio.net/balsera@api.uniovi.es