Revista
Española de

Innovación, Calidad e Ingeniería del Software

REICIS

Volumen 4, Número 2 (especial X JICS), septiembre, 2008

Web de la editorial: www.ati.es/reicis
E-mail: editor-reicis@ati.es
ISSN: 1885-4486

Publicado por la Asociación de Técnicos de Informática
www.ati.es

Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS)

Contenidos

REICIS

Editorial

REICIS

El grupo de Calidad del Software de ATI ha consolidado su posición como principal promotor de la disciplina de ingeniería y calidad del software con la décima edición de las Jornadas sobre Innovación y Calidad del Software (las tradicionales JICS). Estas X JICS pretenden además potenciar la presencia iberoamericana en este foro de promoción de la cultura de la calidad del software y de la innovación en el desarrollo de sistemas y aplicaciones por lo que constituyen la promoción de una I Conferencia Iberoamericana de Calidad del Software (CICS). Por otra parte, las X JICS incorporan la presencia de la ponencia de un destacado experto europeo en la disciplina de ingeniería de software como es Darren Dalcher, Director del UK National Centre for Project Management en la Middlesex University y editor de la revista Software Process Improvement and Practice.

Por otra parte, queremos resaltar la línea de calidad de los trabajos, eminentemente prácticos pero rigurosos, aceptados entre los remitidos en la convocatoria de contribuciones: las ponencias aceptadas (con una tasa de rechazo del 40%) han sido sometidos a un completo proceso de revisión por el comité de programa así como a una cuidadosa labor de revisión de estilo, de terminología y de ortotipografía para garantizar el mejor resultado para nuestros lectores. Por supuesto, no cabe olvidar el apoyo de los patrocinadores (Telelogic, Steria, Deiser, GESEIN y SOGETI) no sólo aportando recursos sino también interesantes presentaciones de experiencias prácticas de sus expertos. Los debates promovidos en las mesas redondas así como la promoción de las actividades de comunicación y *networking* entre los participantes, tanto a nivel presencial como a través de la lista de distribución, los medios electrónicos y la nueva oferta formativa con plataforma *e-learning*. En definitiva, el evento más completo con toda la información disponible en la página del grupo de Calidad del Software (www.ati.es/gtcalidadsoft) acorde a la trayectoria pionera en España que, desde 1997, está proporcionando, a través de la Asociación de Técnicos de Informática, el apoyo para la productividad y la calidad en los proyectos de software. Este perfil ha sido reconocido por el apoyo del Ministerio de Industria, Turismos y Comercio con su apoyo institucional dentro de la convocatoria de la orden ITC/390/2007. Por último, debemos resaltar la aportación de datos de gran importancia no sólo mediante los eventos organizados sino también a través de la realización de estudios específicos (por ejemplo, sobre las prácticas de pruebas, el diseño de casos y los factores que dificultan su implantación eficiente y eficaz en las organizaciones) que permiten un mejor conocimiento de la práctica real de esta disciplina en España.

Luis Fernández Sanz
Juan J. Cuadrado-Gallego
Editores

Presentación

REICIS

En este número especial de septiembre de 2008 de REICIS, por primera vez en la historia de nuestra revista, esta publicación se convierte en el vehículo de difusión del evento decano en España en el ámbito de la ingeniería y la calidad del software: las Jornadas de Innovación y Calidad del Software (JICS) que alcanzan así su décima edición desde su inicio en 1998. En esta ocasión, el Grupo de Calidad del Software de ATI (www.ati.es/gtcalidadsoft) no sólo ha querido cumplir con esta decena de ediciones sino que ha apostado por una apertura a nuevos retos como la presencia de eminentes ponentes invitados de gran presencia internacional y la potenciación de los vínculos iberoamericanos para convertir a este evento en la referencia sobre calidad del software en la amplia comunidad latina. Los trabajos aceptados han sido sometidos a un completo proceso de revisión por el comité de programa así como a una cuidadosa labor de revisión de estilo, terminología y ortotipografía para garantizar la mejor calidad para nuestros lectores. Este número especial constituye en definitiva la publicación de las actas de las X JICS y, por ello, cuenta con un tamaño mayor del habitual. Esperamos repetir este número especial el próximo año con la undécima edición de las Jornadas de Innovación y Calidad del Software. Agradecemos la labor del comité de programa coordinado por la Dr. M.Idoia Alarcón (Universidad Autónoma de Madrid) y compuesto por la siguiente lista de expertos:

- Antonia Mas (Universitat de les Illes Balears)
- Luis de Salvador (AGPD)
- Ricardo Vargas (Universidad del Valle de Méjico)
- Javier Tuya (Universidad de Oviedo)
- Antonio de Amescua (Universidad Carlos III de Madrid)
- María Moreno (Universidad de Salamanca)
- José Antonio Calvo-Manzano (Universidad Politécnica de Madrid)
- José Antonio Gutiérrez de Mesa (Universidad de Alcalá)
- Isabel Ramos (Universidad de Sevilla)
- Esperança Amengual (Universitat de les Illes Balears)
- José Ramón Hilera (Universidad de Alcalá)
- Mercedes Ruiz (Universidad de Cádiz)
- María Teresa Villalba (Universidad Europea de Madrid)
- Adolfo Vázquez (INSA)
- María José Escalona (Universidad de Sevilla)
- Ana Araújo (Ministerio de Medio Ambiente)
- Antonio Rodríguez (Telelogic)
- Gurutze Miguel (TQS)
- Beatriz Pérez (Centro de Ensayos de Software, Uruguay)
- José Javier Martínez (Universidad de Alcalá)
- José Díaz (SSQTB)

Luis Fernández Sanz

Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 2, 2008

Making Software Process Management Agile

José Manuel García, José Javier Berrocal, Juan Manuel Murillo
Universidad de Extremadura
{jgaralo,jberolm,juanmamu}@unex.es

Abstract

The software development process is becoming more complex everyday and thus requires specific tools that facilitate its management. In this article, Zentipede, a tool for managing the software development process in software factories, is presented. This tool simplifies the software process management and provides a lot of statistical data to help measure and improve the quality of developments carried out. Real data obtained in a software factory and the results after using the tool are also presented.

Key words: software process, development process, software factory.

Resumen

El proceso de desarrollo software es cada día más complejo y por lo tanto requiere de herramientas cada vez más especificas que faciliten su gestión. En este articulo se presenta Zentipede, una herramienta para la gestión del proceso de desarrollo software en entornos de fábricas de software. La herramienta simplifica la gestión del proceso software y proporciona una gran cantidad de datos estadísticos que ayudan a medir y mejorar la calidad de los desarrollos llevados a cabo. Se presentan también datos reales obtenidos en una fábrica de software y los resultados conseguidos tras la utilización de la herramienta.

Palabras clave: proceso software, proceso de desarrollo, fábrica de software

1. Introduction

In the last few years, software development has been directed towards obtaining higher quality products in the least possible time and at the lowest cost. One of the main steps applied to achieve these objectives has consisted in discarding obsolete waterfall development processes and replacing them with more agile and iterative processes [1].

The adoption of this type of development process along with the constantly expanding tendency to centralize development in software factories, with its benefits and disadvantages [2], has increased the need for tools that improve the control and exploitation of the resources. On one hand, tools are needed to measure the use of the development processes and the quality they provide (http://kpilibrary.com/), [3]. Tools are also needed to facilitate the management of the software process in this kind of environment [4].

Although there are tools that cover some of these characteristics, many of them are not adapted to agile and iterative development processes. Moreover, the integration of many tools is clearly needed in order to avoid the duplication of effort [5], [6].

In this paper, we present Zentipede[4] [5] (http://www.zentipede.org/), a tool designed to fulfill these needs. This system facilitates the management of the software development process in distributed environments supporting collaborative work in software development. This tool automates some tasks for quality management, provides statistical data about the quality of developments and integrates the functionality of multiple applications to reduce the duplication of effort.

The system originated from a real need present in a software factory located in the University of Extremadura. In this environment, we detected some deficiencies in software process management in existing applications, mainly because they did not focus on agile and iterative processes.

The second section of this paper describes what motivated the creation of the tool. In the third section its main features are presented, giving more detail about every one of the modules that compose it in sections four, five and six. Its advantages over other related work are detailed in section seven. The results obtained after its use in a software factory are discussed in section eight. The last section deals with conclusions and future work.

2. Motivation

The tool Zentipede arose, as mentioned in the previous section, to fulfill a need present mainly in software factories. The specific case at hand is that of the Teseo and InodUex factories created at the University of Extremadura with the collaboration of SDAE and INDRA, respectively. In these factories, the projects developed were mainly sent by the founding companies from some of their different locations.

Due to this relocation of development and developers, there was an increase in the need for exhaustive controls that allowed for the best possible use of available resources at all times. Typically, these management tasks are done by hand.

[4] The name of the system comes from the English word centipede. Zentipede has been chosen as a hallmark and due to the similarity of the Z with a moving centipede.

[5] The project has been financed with funds from CICYT project TIN2005-09405-C02-02 and tested in the software factories Teseo and InodUEx.

As the factory grew and the number of projects and resources involved increased, the time necessary for the management tasks also increased. This produced an increased workload for managers, which led to less time allocated to the development of projects in order to increase the time spent managing them.

Zentipede was conceived as a tool to fulfill these needs and facilitate the management of software development processes carried out by relocated teams. To do so a web application was developed first to collect information from everyday tasks carried out in a software factory and generate reports from them.

3. General characteristics

The main aim of the Zentipede tool is to facilitate the management of the development process in a software factory. To fulfill this function the tool must adapt to the software process used by the factory but be flexible enough to allow for variations required by customers. From this starting point, Zentipede provides a wide range of features. These characteristics are grouped according to the functions they fulfill.

On one hand, we have the following features whose aim is to optimize the use of the resources. The tool allows for a complete management of the workers and their workday. The tool can also collect information about all the projects. From this information it is possible to manage the participation of workers in the different projects and maintain a complete control of the number of hours spent by each of them on each project. Finally, it can control the unavailability of workers.

Moreover, the system has another series of characteristics to improve the quality of the software development process. Each project is divided into use cases and these, in turn, are divided in more fine-grained tasks, simpler to carry out and estimate their length and complexity. It is possible to assign these fine-grained tasks to one or more workers during the different iterations of the project so as to keep control of the part of the project in which a particular worker actively participates in each iteration. That control may be done in a more exhaustive way; it is even possible to associate both tasks and workers with fragments of source code. Finally, the system generates a large number of reports in different formats from the information stored in it. These reports are very useful to evaluate various aspects of a software factory, from a global level to that of a specific project or worker.

To conclude, the system has one last feature aimed at minimizing duplication of efforts and automating part of the process. To avoid the duplication of work the system helps generate the documentation of a project from the information it has collected. This is achieved through the use of a wiki as a collaborative repository for documentation of projects.

In addition to these functions, one of the key concepts that has dominated the creation of Zentipede is the simplicity of use. Throughout the entire development, the intention has been to minimize, as much as possible, the time required for data entry because this time is subtracted from that available for project development.

The following sections describe the various modules that make up Zentipede and which provide all the functions.

4. Zentipede Web Portal

In this section the most important module of Zentipede is described. This module is responsible for providing most of the functionality described in the previous section. The rest of modules that make up the tool are complements of this.

This web application is responsible for collecting data from the daily work done by all the members of the software factory. From these data a series of reports are generated, which give us a lot of information about the quality process being followed, especially of the areas of project planning and project monitoring and control described by CMMI[6].

The operation of the tool for the three roles involved in it is described below.

4.1 Developer

The application presents to the developer the task that he must perform daily. At the end of the day the developer should indicate the progress made and the hours spent on each task. This way a large amount of information about the work being done in a factory is obtained almost without interfering with the work of development.

4.2 Project manager

The users of this profile are the ones who must provide the most data to the tool. The application provides these users with the same functionality as that provided to developers

[6] CMMI. http://www.sei.cmu.edu/cmmi/models/index.html

and thus a part of the information to be provided is the same, specifically the time spent in the realization of the various tasks that they carry out.

In addition to this basic functionality, project managers have greater control over one or several of the projects being developed in the factory. This control is carried out through a workflow consisting of the steps shown below. For a better understanding of the workflow, Figure 1 shows it in the form of a business process. For greater clarity, tasks carried out by developers have been included in the business process.

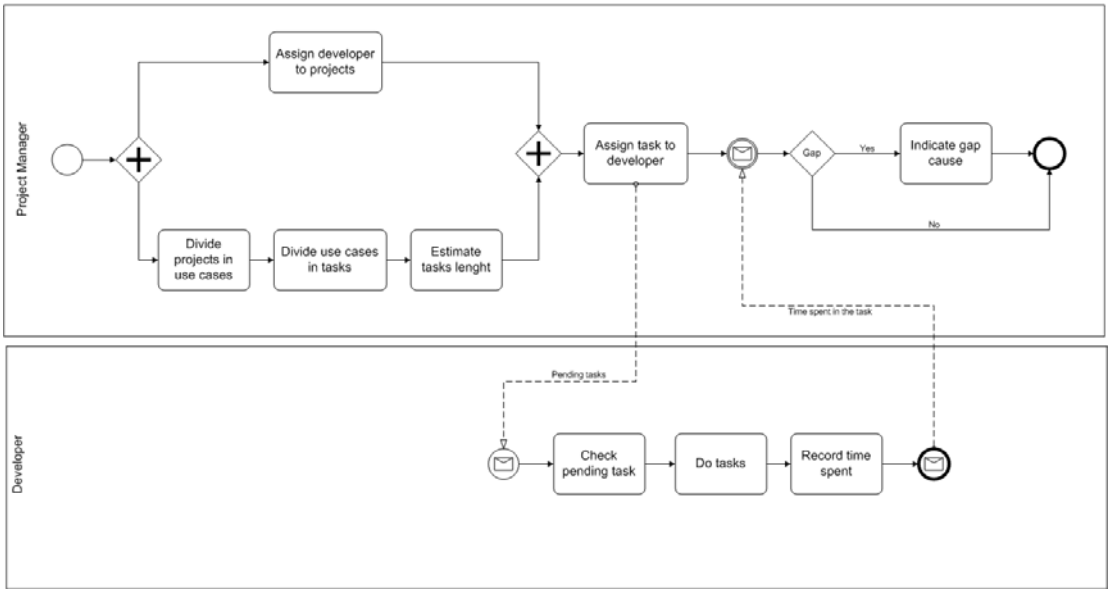

Figure 1. Business process followed by project managers.

The time that project managers must invest in the use of the tool is, as can be seen, considerably more than the time spent by the developer. But this dedication is still less than the time spent on management tasks before this tool. Moreover, the results are better than those that similar tools provide, as may be observed throughout this article.

4.2 Administrator

These users are those who benefit the most with the use of this application. They have at their disposal the same functionality as the project managers, expanded with a wide range of possibilities that give them complete control over the tool and the projects that are managed with it. These users also get lots of reports about the status of the factory and all the projects and workers involved at all times.

For the management of a project using this tool the only task that can only be done by the administrators, as such, is the inclusion of the project itself in the system. Once a project is created the rest of the necessary information can be provided, as has been seen, by project managers and developers.

Moreover, these users have a number of functions to modify some aspects of this tool to suit their specific needs.

But the biggest advantage of this application is the wide variety of data and reports it provides to administrators. These reports are obtained from all information provided. These reports also have the great advantage that many of them are connected to each other, allowing us to jump from one to another comfortably and thus find necessary information easily. Some of these reports are shown in later sections.

Administrators have all this information to improve the management of the development process and achieve a better use of available resources. Likewise, better control of the software process is achieved, which implies an improvement in quality.

5. Zentipede Eclipse Plug-in

This module was designed primarily to meet two goals. Its main function is to provide information concerning the code which the developers are working on. Therefore, it is possible to relate the tasks carried out by workers with the part of the code affected by these tasks. The other function provided by this module is to facilitate the daily use of the tool.

To achieve these objectives the most logical solution is to concentrate on the development environment. Because of this, this module was developed in the form of a plug-in for the Eclipse development environment[7]. This environment was chosen because it is one of the most used IDEs today, its architecture is perfectly prepared for the inclusion of plug-ins and it is licensed as FOSS.

The plug-in is connected with the web application, based on SOA, and the list of tasks assigned to the user is obtained. This task list is shown to the user. Once the assigned tasks are obtained, it is possible to register, in the system, the time spent in developing each of them from the development environment. This achieved one of the objectives of this

[7] Eclipse. http://www.eclipse.org

module, facilitate the daily use of the tool, so much so that developers can completely avoid the connection with the web application.

The mechanism for the registration of the changes to the code is more complex. The way to achieve this is to select, from the list of assigned tasks, the one the user is working on. By selecting one of the pending tasks the plug-in is responsible for monitoring the development environment to register the code that is changing.

This monitoring differs according to the files the user is working with. For files that do not correspond to Java code the only information that is recorded is that the user is modifying them. However, if it is a Java code file the environment provides a lot of information. When the user works with Java code files, the plug-in not only records the file that is being manipulated but also keeps a record of the packages that are being imported and classes that are changing within the file. Also, within a given class, the plug-in records what attributes or methods the user is working on.

Thanks to this monitoring, very precise information is obtained about the code areas in which the tasks are implemented and what users have done so. All this information is stored in the Zentipede Web Portal module and can be consulted later. This contributes very positively to the traceability of the product and the process and ultimately to the quality of both. Besides that, these data have another utility: when a user receives his list of pending tasks he also receives the areas of source code associated with them that have been modified during the realization of this task, so that a user who is assigned to a task can quickly find out what parts of the code other people have worked on.

6. Zentipede Documentation Center

In this section, the last module of Zentipede is presented. This module was designed to try to reduce one of the most common drawbacks in the development process: the duplication of data and the potential for inconsistencies.

A situation that occurs very frequently during the development process is the need to enter the same data in the different tools that are being used, with the risk of inconsistencies that this entails. The module that we present in this section is intended to mitigate, as far as possible, this problem.

In particular, this module tries to reduce these problems. To do so, the main web application has been integrated with a wiki whose main mission is to collect all the documentation of the project. The implementation of wiki chosen to integrate into Zentipede has been MediaWiki[8] because it is widely used and has a free software license. This integration has been done with the intention to reuse the information entered in the web application to be included automatically in the documentation associated with the project.

To integrate this module with the Zentipede Web Portal module, the Spring framework facility for aspect programming has been used. In this way, it is possible to incorporate the new functionality in a completely transparent way.The features added by this module are detailed below.

It is possible to define the desired structure for the documentation of the projects without being subject to any restrictions. A structure of documentation is associated to each project within the tool. This structure is generated automatically on the wiki.The structure of the documentation may be associated with project maintenance actions carried out in Zentipede Web Portal. So when one of these actions occurs in the web application, it affects to the documentation in the wiki. Because the wiki is accessible through the Web, the entire generated repository can be accessed directly, without using the main module. Taking advantage of the capabilities offered by MediaWiki, a historical trace of all actions carried out on the documentation of projects is maintained.

Through the use of this module, it is therefore possible to maintain the documentation of each project, with the necessary structure in each case, in a repository that allows for collaborative editing while taking advantage of the information entered in the main module to complete this documentation automatically.

7. Related work

There are other tools similar to Zentipede. In this section some of the best known ones, whose deficiencies have led to the creation of this system, are presented along with their advantages and disadvantages compared to our application.

[8] MediaWiki. http://www.mediawiki.org/

In general, most of these tools have two disadvantages, the first of which is that they are not oriented to agile and iterative development processes and, mainly, they have not been designed for use in software factories. The second disadvantage is that they focus their functionality on a single aspect of the development process, dissociating them from other aspects.

One of these applications that stands out is dotProject[9], one of the most widely used tools. However, it not only shows the disadvantages already mentioned, but also has problems with obtaining statistics about the information collected, which make it less useful for the management of an environment like a software factory.

Two other interesting tools in this field, although more limited than the previous one, are activeCollab[10] and BaseCamp[11]. However, they present a serious inconvenience: they are not licensed as free software and therefore cannot be easily expanded with the functionality needed.

Finally, there are other tools, such as LightHouse[12] and CruiseControl[13], but they are focused on more specific aspects of the development process, such as the management of the bugs that appear in the projects or the building process. This specification, however, introduces features that could be very interesting for the management of the development process.

8. Validation

The tool that is presented in this article has been used successfully in a software factory consisting of approximately 30 people. Throughout the period of use, more than ten projects were developed. The tool has recorded data from daily work in all these projects, so that after the completion of the tests the system has information on more than 7500 man-hours as shown in Figure 2.

[9] DotProject. http://www.dotproject.net/
[10] ActiveCollab. http://www.activecollab.com/
[11] Basecamp. http://www.basecamphq.com/
[12] LightHouse. http://lighthouseapp.com/
[13] CruiseControl. http://cruisecontrol.sourceforge.net/

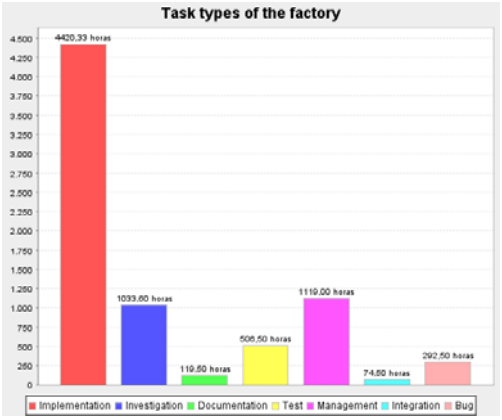

Figure 2. Worked hours during the use period of the tool, classified by task type.

The figure shows the most common type of task in a software factory is, as might be expected, the implementation tasks. However the large number of hours spent on management tasks must be highlighted, as this demonstrates the need for a tool like Zentipede, which facilitates these tasks and helps increase productivity, in such an environment. Two curious aspects stand out in this figure. The first one is the large number of hours spent on research task motivated by the location of the factory in a university environment. The other one is the little time spent on tasks of integration or correction of errors, which was due to the fact that much of these tasks were performed by the founder company of the factory in its other locations.

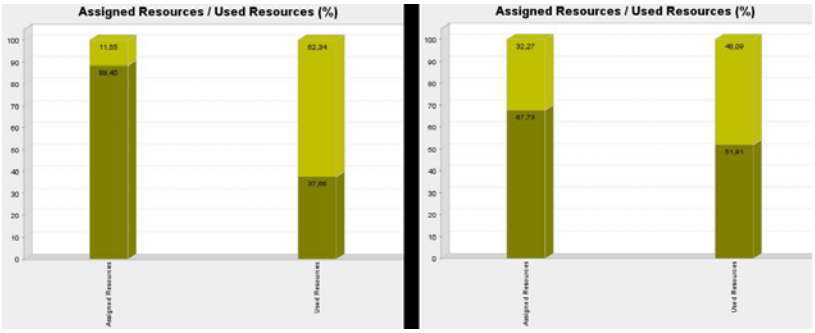

Figure 3. Data about the assigned resources compared to the used resources and their evolution.

The results have been clearly positive. The time spent on the management of the software process has been significantly reduced, while the information related to the process followed for each of the projects has increased remarkably. In addition, thanks to

the statistics provided by the tool, the accuracy of the time estimates increased and thus the utilization of available resources improved. This increase in the accuracy of the estimates is evident in Figure 3. This figure shows two equal graphs of the assigned resources and the resources actually used. The graph on the left is one of the first few weeks of use of the tool. In this graph the differential between resources allocated to those used is of more than 50 points. In the second graph, of the subsequent period after the experience gained through the use of the tool, the differential is reduced to less than 20 points.

These improvements in the estimates are largely due to information provided by the tool itself. In Figure 4, the difference between the estimated time and worked time on several projects is shown. This gap is further divided into positive gap (more time than needed was estimated) and negative gap (less time than needed was estimated).

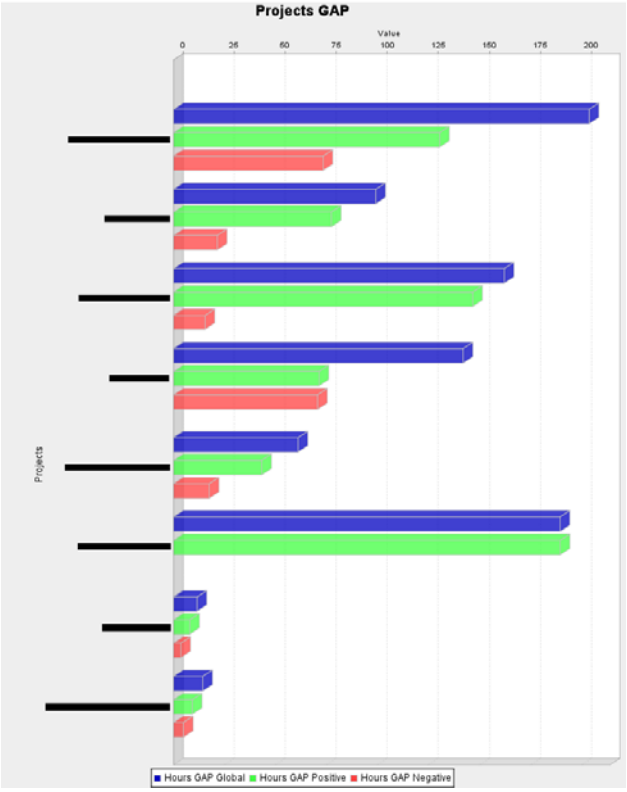

Figure 4. Difference between estimation work and realized work classified by project

The benefits are not limited to a better use of available resources or a decrease in the time spent on management of the software process. In addition, thanks to the use of Zentipede, the factory has increased its database, with detailed data on the development of several projects. Such information will serve to take on new projects with more exact

criteria about the capacity and productivity of the factory. An example of this can be seen in figure 5[14] where number of assigned resources to several projects over a period of time is shown. These data can help managers to decide about the number of resources a new project will require or the ability of the factory to deal with new developments.

9. Conclusion and future work

In this paper, we have presented the Zentipede system, the motivations that led to its creation, its main characteristics and the different modules that compose it. Also, a study of existing systems that cover part of the same functionality has been done, paying special attention to the benefits of Zentipede. In conclusion, details of the results obtained from the use of the system in real software factories have been given.

The current state of the system corresponds to the one described in this article, however it is constantly being improved. The inclusion of the support for new programming languages, among those supported by the environment like C + +, is being considered for Zentipede Eclipse Plugin. The ability to connect Zentipede Web Portal through SOA is also being expanded. In addition, the Zentipede system is part of a much larger project in progress. The final system aims not only to cover the management of the software process, but also to assist and automate, as much as possible, its execution at all levels, from capturing requirements to obtaining source code.

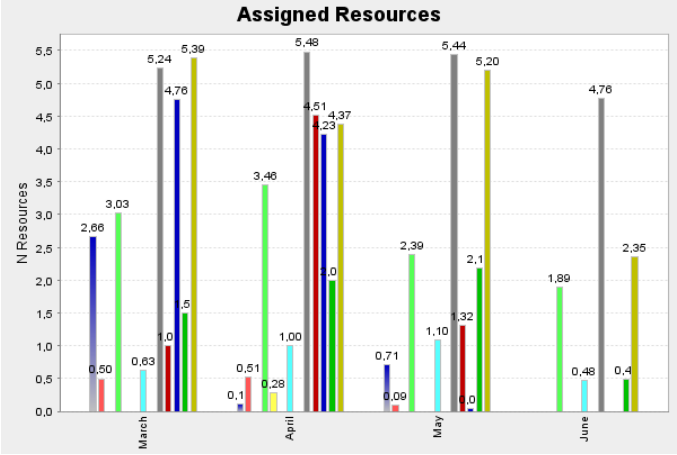

Figure 5. Assigned resources to different project during a period of time

[14] Projects names omitted for confidentiality purposes in figures 4 and 5.

Finally the authors want to highlight the availability of the system as free software and their disposal to address any doubts that may arise about the system or suggestions on possible improvements for future versions.

References

[1] Larman C. *Agile & Iterative development: a manager's guide*. Addison Wesley, 2005.

[2] Prikladnicki, R., Audy, J.L.N., Damian, D., y de Oliveira, T.C. "Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring", en: Paulisch, F., Kruchten, P. y Mockus, A. (eds.) *Proceedings of the Second IEEE International Conference on Global Software Engineering 2007*, pp. 262-274, 2007.

[3] Dunn, R. N., "KPIs for Agile Teams", *Agilejournal*, Vol. 3 Num.5, 2008, www.agilejournal.com (consultado en julio de 2008)

[4] Ambler S.W. y Nizami, K. "Agile Strategies for Geographically Distributed Quality Management", *Agilejournal*, Vol. 2 Num.11, 2007. www.agilejournal.com (consultado en julio de 2008)

[5] Sinha, V., Sengupta, B. y Gosal, S., "An adaptive tool integration framework to enable coordination in distributed software development" en: Paulisch, F., Kruchten, P. y Mockus, A. (eds.) *Proceedings of the Second IEEE International Conference on Global Software Engineering 2007*, pp. 151-155, 2007.

[6] Krishnamurthy, V., "Benefits Of Tool Integration In Distributed Agile Development". *Agilejournal*, Vol. 1 Num. 6, 2006. www.agilejournal.com (consultado en julio de 2008)