E-MANAGEMENT AND QUALITY ASSURANCE

G. Milcinski, J. Dovc, J. Kamenik, A. Kmet, I. List, M. Plesko, K. Zagar, JSI and Cosylab, Ljubljana, Slovenia

Abstract

When control system development is done, one might ask: ‘Does it work?’, ‘Who implemented this component?’, ‘How much time had he spent doing it?’ and ‘Was the component implemented on time?’ In academically-minded development environments, such as the KGB Team [1] of the Jozef Stefan Institute, the usual answers to these questions would be: ‘It works just fine?’, ‘Given the colorful appearance of the buttons I’d say it was John?’, ‘Does it matter?’, and ‘No, but it is not a problem, since the whole project was late anyway.’ Due to the lack of up-front planning and accurate progress tracking, there usually remains little time for testing the control system thoroughly and documenting it comprehensively. Consequentially, only the author truly understands his work, and is, thus, irreplaceable. But what if he leaves the team? Who will take up his responsibilities? When a part of the KGB Team separated from Jozef Stefan Institute to start its own business as Cosylab [2], failure to do things right, well, on time and on budget had immediate effects on the paychecks. Thus, we were forced to introduce a low-overhead, user-friendly e-management system, as well as a quality assurance system, through which the processes, roles and quality controls within the company are formalized. Experience with both of these is shared in this paper. We are relying on Open Source tools, such as Request Tracker [3], Internet Calendar [4], CVS [5] and MrProject [6]. We have developed CosyDoc, an XML-based system for writing documentation, which allows us to write living documents more efficiently. Atop of that, we are defining our software and hardware development processes, which are based on Rational Unified Process [7], Microsoft Solutions Framework, Capability Maturity Model, ISO 9126 (software quality characteristics) and ISO 9000 Tick IT (quality assurance).

WHY STRICT PROJECT MANAGEMENT IS NEEDED?

Usually, project management is done in one of the two ways: corporations like McDonalds use the first one, where the procedure is written for just about everything. The other is based on improvisation. Both can be most easily described using a 5 W-questions system.

If we devote ourselves to IST sphere, these questions are for the McDonald-like system the following:

What exactly do we want to do?
Who is responsible?
Where should it be delivered?
Ho**W** should it be done?
When is the deadline?

Some individuals would ask another W-question: Why should we do it in such way – there is too much overhead? Especially in the academically minded sphere, where the progress is based on improvisation and the responsibility is scattered among large number of group members, people do not want to work in such a way. The problem appears when the group is involved in an important project. Suddenly W-questions appear again. This time they are slightly different:

What is wrong here?
Where is the guy who did this mess?
Ho**W** can we fix it?
When can I go on vacation?
Who am I kidding?

The fundamental distinction between both systems is the fact, that, in the first case, we can actually find answers.

We started as a group of students, who disliked more than anything else the bureaucracy related to work. We believed that the only important issue is a desire for programming. A moment came when all we did was bug fixing. We found out the answer to the last of five questions: we were kidding ourselves.

We introduced processes in our environment, one after another. We thought about responsibilities, dependencies, etc and found many important links in our projects along the way. We realised that we have to unite our thoughts and since the telepathy is not working properly we started discussions. It is true that we are able to describe just a small percent of our thoughts but the worse thing occurs – communication is being lost. Because of the rainforests slowly disappearing, piles of paper are not popular any more so we sought electronic tools.

HOW SHOULD I PLAY THIS GAME?

Request Tracker

There is a multitude of project management tools, many of them freely downloadable from the Internet. Some of these tools are specialised for customer relationship management (CRM), e.g., by taking care of customer’s support requests and helping them get resolved by the organization’s work processes. Others include address books, calendars, forums and to-do lists. We chose the Request Tracker (RT) because of one simple reason – this tool can manage e-mails really well – not only sending but also receiving. RT was picked as our main tool and we adapted all other ones to it. Every now and then, when experiencing problems with RT’s code written in Perl, a question appears why we did not rather write such tool by ourselves, but the final statement remains that RT is really good structured and extremely useful.

RT is (as described in its manual) an “enterprise-grade ticketing system which enables a group of people to intelligently and efficiently manage tasks, issues, and requests submitted by a community of users".

Its main unit is a ticket, which represents a specific task. A number of similar tickets is grouped in a queue (project). Specific ticket cannot be contained in more than one queue which is an issue we sometimes wished for. There is a lack of tree-like structure of queues but we are able to build such structure with tickets – each ticket can have one or more parents, children or brothers. Setting also dependencies, clear structure can be made. RT can warn us by email of a creation or modification of some ticket. The best feature of RT is the possibility of managing tickets via e-mail – every queue has its e-mail address to which we can send a request for creation, correspondence or comment and set just about every field of the ticket.

RT has easy-to-use search functionality, which allows one to quickly find a ticket. With all its features RT can be used for handling support requests, ordinary tasks or bug tracking.

Mr. Project

Mr. Project is an open-source project management tool running on Linux. It allows project managers to define tasks, establish dependencies between them, and assign resources to tasks. Given this input, Mr. Project is capable of preparing a Gantt chart and resource utilization report.

Mr. Projects files describing projects are stored in XML files, whose structure is very clear. This opens possibilities of integration of Mr. Project with other products (e.g., the RT).

Mozilla Calendar

An open-source calendar is included into Mozilla and is available as its plug-in. It doesn't have all the eye-catching features of competitive products (e.g., Microsoft Outlook) but there is all that we need for everyday use and some more. A useful feature for example is publishing calendar on the Internet or intranet, so that other members of the group and especially project manager can have an insight into one's schedule for the upcoming days and weeks. Not to worry – we can keep our private tasks for ourselves. Different calendars (work, private, etc) are being used as layers – we can have any number of them and choose which to display simultaneously. This makes it extremely easy to see how one's personal calendar is aligned with the calendar of organization's events, for example. It would be nice if you were able to move specific task from one calendar to the other (or also managing group operations – to copy all events from private calendar to the working one, changing the content to for example Busy) but we cannot have everything. Again, it is true that we do not cross to such problems if we are not using the calendar at all (we just do no have time for our private life…).

Another nice feature of the Calendar is iCalendar [10] file format in which individual layers are stored. iCalendar format is a standard, text, human readable format, which enables manipulation with various scripts.

Rational Unified Process (RUP)

As a framework atop of which we have defined our processes we have chosen the Rational Unified Process (RUP). RUP divides software construction in four phases (inception, elaboration, construction and transition), during which the software matures in iterations – each iteration building on the previous one to add new functionality, or to improve stability.

CVS

Concurrent Versioning System is, as the authors advertise, the dominant open-source network-transparent version control system. It does not matter which versioning system you are using but that you are using it. Without this basic tool no team software programming should be made. Although CVS is simple and free, it adequately addresses the needs of majority of software projects. Apart from that, it is greatly integrated into Eclipse [9], the Java developer's Integrated Development Environment of choice.

Automated Building and Testing

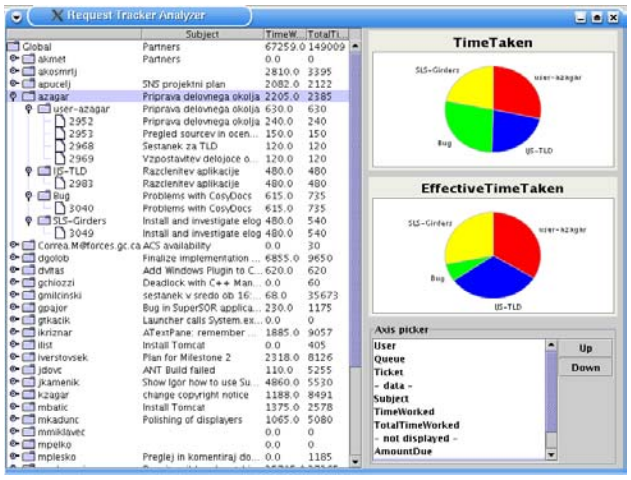
We are using automated tools for building and testing the software, not just at the end of development, but periodically (nightly). This way, integrity of the software can be guaranteed by fixing defects as they occur.

WHO HAS DONE THE MOST?

This question, and others like it (e.g., "how many times did X miss the deadline and by how much", "who is best at estimating required effort", …) are frequently asked by project managers.

After all the group members have been persuaded to use the RT for effort reporting, the only thing you need to get answers to such question is a tool for analysis. Sometimes it is other way around – you need such tool to show to people why the RT is so useful.

RT Analyzer



	Subject	TimeW...	TotalTi...
Global	Partners	67259.0	149009
akmet	Partners	0.0	0
akosmrlj		2810.0	3395
apucelj	SNS projektni plan	2082.0	2122
azagar	Priprava delovnega okolja	2205.0	2385
user-azagar	Priprava delovnega okolja	630.0	630
2952	Priprava delovnega okolja	240.0	240
2953	Pregled sourcev in ocen...	150.0	150
2968	Sestanek za TLD	120.0	120
2969	Vzpostavitev delojoce o...	120.0	120
IJS-TLD	Razclenitev aplikacije	480.0	480
2983	Razclenitev aplikacije	480.0	480
Bug	Problems with CosyDocs	615.0	735
3040	Problems with CosyDocs	615.0	735
SLS-Girders	Install and investigate elog	480.0	540
3049	Install and investigate elog	480.0	540
Correa.M@forces.gc.ca	ACS availability	0.0	30
dgolob	Finalize implementation ...	6855.0	9650
dvitas	Add Windows Plugin to C...	620.0	620
gchiozzi	Deadlock with C++ Man...	0.0	60
gmilcinski	sestanek v sredo ob 16:...	68.0	35673
gpajor	Bug in SuperSOR applica...	230.0	1175
gtkacik	Launcher calls System.ex...	0.0	0
ikriznar	ATextPane: remember ...	1885.0	9057
ilist	Install Tomcat	0.0	405
iverstovsek	Plan for Milestone 2	2318.0	8126
jdovc	ANT Build failed	110.0	5255
jkamenik	Show Igor how to use Su...	4860.0	5530
kzagar	change copyright notice	1188.0	8491
mbatic	Install Tomcat	1375.0	2578
mkadunc	Polishing of displayers	1065.0	5080
mmiklavec		0.0	0
mpelko		0.0	0
mplesko	Preglej in komentiraj do...	0.0	1185

Figure 1: Screenshot of the Request Tracker (RT) Analyzer, showing distribution of time the given user (azagar) had spent on his projects (IJS-Girders, bug fixing, IJS-TLD and other tasks).

As opposed to the other tools mentioned in this article, which have been only adapted to our needs, RT Analyzer was developed in-house entirely. It offers on-line analytical processing (OLAP) capabilities over the RT database. Through RT Analyzer, it is possible to extract and summarize distribution of effort (actual and estimated) over projects, resources (engineers) and time. The user requests results using simple drag-and-drop operations, and RT Analyzer swiftly responds with graphs and tables.

Project Reports

Every now and then people need reminders. The problem is that we get customised to repetitive events quite rapidly. Therefore we need to make reminders as interesting as possible.

There are numerous possibilities available to realize delivery of reminders to those they concern, such as e-mail notification, short messages via cellular phones, LED display over the desk, etc. The simplest solution is definitely e-mail.

What about the content? We have decided to remind the users know of the deadlines of tickets that are imminent. Another interesting report is a week report – we have one intended for every individual, telling him/her what did he/she do and how much time was spent for it. Similar report is being sent to project managers each week giving them an overview of the activities on their projects.

A true meaning of such reports is that we can quickly get the feeling about the areas that are OK and those that are not, and to identify which person did not do much during the week. After project manager finds out all these things there is just a small step needed to do something about it.

WHAT COMES NEXT?

Individually, previously described tools are already very useful, but once united they become indispensable. In the process of merging them we had to decide which database should be the primary one (had several been used, we would have run into problems with synchronization). Because of excellent structure of RT this decision was not difficult.

A connection of Mr. Project and RT is probably the first choice. We intend to use Mr. Project for quick and easily creation of tasks that are later synchronized with RT. We could also get feedback from RT about task realization and time spent on the project, and display in the Mr. Project's Gantt chart.

Next interesting connection is that of Calendar and RT. Calendars containing planned vacation, business trips, etc., could be inserted in the RT database. This would give project managers using RT a capability to determine the load and availability of their team members. This could also be done in the reverse direction – information about persons' tasks could be extracted from RT and used to create a to-do calendar which can be clearly presented in Mozilla, which is helpful for making decisions related to time allocation.

An interaction between RT and CVS or automated testing is not as straightforward. First one can be used for making estimation of time spent for one line of code. The other one would enable centralized collection of metrics (number of lines of code, number of build/unit test errors, …). Also, the person responsible for introducing an error detected during periodic automated builds could be automatically assigned a ticket to resolve it.

Last but not least is a connection of RT with the RUP-based process. All until now presented procedures can be united together and with addition of RUP, the project manager's job would be a much easier one. It is not so simple but one could automate many routine operations using RT scripting capabilities. The script would calculate required actions and present them to the project manager for signing, just as a secretary brings documents to the executive.

WHEN IS THE RIGHT TIME?

If the question is being asked regarding the deadline for introduction of similar process into your organisation, the answer is, most probably, that the deadline has already passed.

All that has to be done is to choose the tools and processes to be used. No tool (or process) fits perfectly at the beginning, but when some effort is invested into its adaptation to the organization's needs, soon the question will appear how something could have been done properly before.

REFERENCES

[1] http://kgb.ijs.si
[2] http://www.cosylab.com
[3] http://www.bestpractical.com/rt/
[4] http://www.mozilla.org/projects/calendar/
[5] http://www.cvshome.org
[6] http://mrproject.codefactory.se
[7] http://www.rational.com/products/rup/index.jsp
[8] http://bugzilla.mozilla.org
[9] http://www.eclipse.org
[10] IETF, RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar), http://www.faqs.org/rfcs/rfc2445.html