COSYLAB MANAGEMENT SYSTEM*

I. Verstovsek*, G. Milcinski, M. Plesko, K. Zagar, Cosylab, Ljubljana, Slovenia

Abstract

As a result of the free spirit in research institutions, project management is mostly considered a necessary but awkward task. We present a working solution tailored to academic projects that requires only a minimum of effort and discipline and results in huge benefits, which will be presented in this article. Cosylab is of academic origin, therefore the spirit, organization and work procedures are very much like in research institutes. In addition, we work on about a dozen projects simultaneously for customers on four continents, which requires a lot of travel and on-site work. Commercially available project management tools are not suited to manage such diversity. We have therefore adopted a set of open source tools, such as Request Tracker, Wiki, Sympa mailing lists, mySQL and maven, implemented some custom additions and integrated the tools into a coherent product to suit our purpose. It enables developers to track their work and communicate effectively; project managers to monitor progress of individual projects; and management to supervise critical parameters of the company at any time. In the article, the experiences gained by using the system are presented. As it has turned out in practice, the product is ideal also for research institutes, as it is demonstrated by its use at the Swiss Light Source (SLS).

HOW AND WHY TO MANAGE?

We have started as a team for control systems in the largest Slovenian research institute Jozef Stefan. After a successful completion of the control system for ANKA [1], we have established a spin-off company Cosylab [2].

Already as a research group, we became aware of the need to have some sort of project management and adequate electronic support for it. We adopted a set of open source tools to help us manage ourselves [3, 4]. Since then, the requirements on the project management were constantly increasing, and as a consequence, our tools were evolving - we began including new tools and wrote additional software, some for customization of existing software, but mostly for integration of different tools into a coherent whole.

In the last two years, the company has grown significantly, from a team of eight to a bunch of twenty full time employees and additional twenty part-time students. As it seems at the moment, the number will increase for a further 10 this year. Also the number of projects in progress has risen from a few at the time to about a dozen at a time. Although this certainly is good news from the business perspective, it adds another level to complexity of the organisation that is not trivial to handle.

*igor.verstovsek@cosylab.com

One of the key questions that arises is - have chosen the right set of tools for this transition? Will it scale when we grow for additional factor of two? The answer is hopefully "yes", otherwise we will be in an intriguing situation, to put it mildly.

All the time we try to keep in mind our vision from the early days - to retain the spirit of a research organisation that has infected us when we worked within a research institute, after all, most of us have scientific background and mentality. In other words, we never want to become a dull assembly-line kind of a company. It is true that in a company it is easier to see why is it good to keep within the budget and meet deadlines, mostly because the metric that tells you whether you are on the right track is trivially simple - the pay checks. But then again, in research institutes it also does not hurt being on time and on budget.

Therefore, the solution that we have created can be extremely useful also for research institutes. We have indeed proven this to be the case, by successfully introducing our project management system at SLS [5].

THE TOOLS

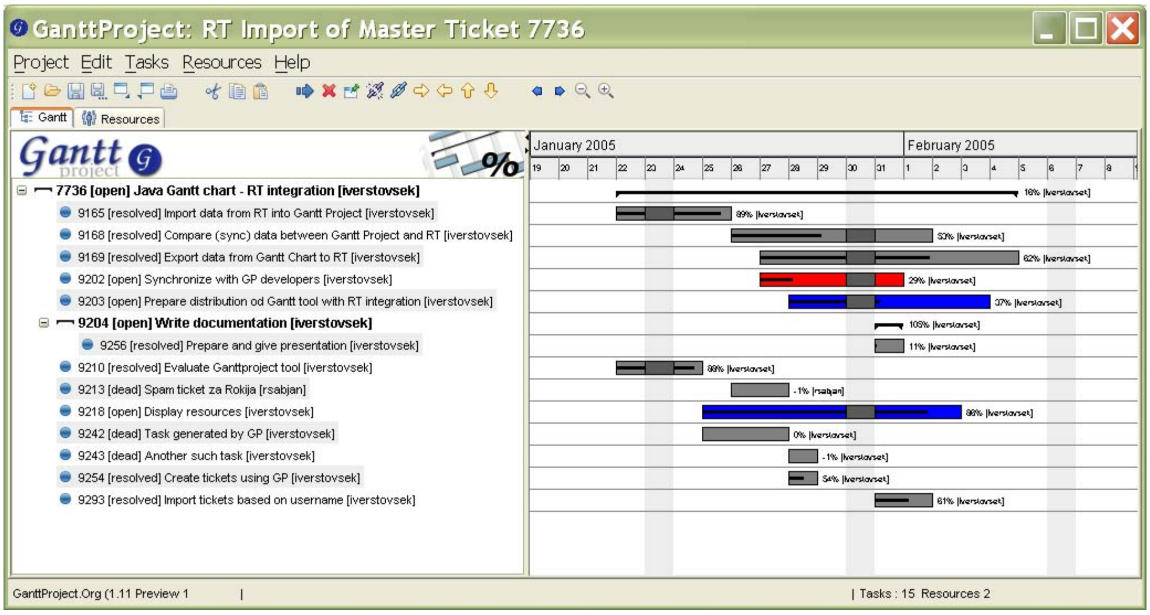
Request Tracker to Define Units of Work

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 well 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 to be done. It has several fields:

- status (new, open, resolved, stalled, dead)
- estimated time
- time spent (increases whenever a user reports a transaction)
- dates: start date, due date
- priority



7736 [open] Java Gantt chart - RT integration [iverstovsek]
- 9165 [resolved] Import data from RT into Gantt Project [iverstovsek]
- 9168 [resolved] Compare (sync) data between Gantt Project and RT [iverstovsek]
- 9169 [resolved] Export data from Gantt Chart to RT [iverstovsek]
- 9202 [open] Synchronize with GP developers [iverstovsek]
- 9203 [open] Prepare distribution od Gantt tool with RT integration [iverstovsek]
- **9204 [open] Write documentation [iverstovsek]**
  - 9256 [resolved] Prepare and give presentation [iverstovsek]
- 9210 [resolved] Evaluate Ganttproject tool [iverstovsek]
- 9213 [dead] Spam ticket za Rokija [rsabjan]
- 9218 [open] Display resources [iverstovsek]
- 9242 [dead] Task generated by GP [iverstovsek]
- 9243 [dead] Another such task [iverstovsek]
- 9254 [resolved] Create tickets using GP [iverstovsek]
- 9293 [resolved] Import tickets based on username [iverstovsek]

GanttProject.Org (1.11 Preview 1

Figure 1: Gantt Project with tickets, imported from the Request Tracker.

- keywords (severity of a bug, type of ticket: QA ticket, Master ticket for the project, etc.)

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, a 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 and bug tracking.

To illustrate the usage of RT in Cosylab - on March 14th we have reached RT ticket number 10000, which gave as a nice excuse to throw a party.

Gantt Project to Supervise Project Progress

RT and its web interface, however extremely useful, do not have everything one would wish for when managing and monitoring a project in progress, especially if this project is larger than a few man-weeks of work. The central entity of RT's web interface is a ticket, and just by browsing through different tickets, one can easily get lost by looking at the trees, and not seeing the entire forest.

In addition, one would also need a tool for oversight of how the project is progressing as a whole, how far is each of the tasks, which tasks are late, which task depend one on another, how much work is assigned to developers (resource utilisation), etc., in other words, to be able to take a "helicopter view" on the project. The functionality we look for is exactly the one of a Gantt chart.

There are plenty of commercial and open source tools that provide such functionality. We sought for an open source tool that would easily be integrated with RT. We have chosen Gantt Project (GP),written in Java. We have implemented a two way integration with RT by means of *Import* and *Export* features of GP. On import, one must enter a parent ticket number, and subsequently, this ticket and all its child tickets will be imported into GP. The mapping between the RT ticket and GP task is simple since they are conceptually the same.

After import, it is possible to see the progress of the project on a Gantt chart, where RT ticket status is colour coded on the Gantt chart. Another useful feature is the actual completion of the task in percents (calculated automatically from the data entered into RT). Here our solution adds value - there are probably not many project managers that would actually take care that accurate completion percentages are displayed on the Gantt chart.

Furthermore, it is possible to set starts and due dates of tickets, change ticket status, write comments or perform any other manipulation allowed by RT. It is even possible to make new RT tickets!

When finished with editing, a synchronisation with RT can be performed. Here, all the changes are enumerated for preview and it is possible to select a subset to commit back to the RT database. Synchronisation is performed via RT's mail interface - the program sends emails with appropriate commands to RT.

Integration of Gantt Project and RT brought something that most of the commercial tools lack - the ability to look on the project on different levels: from the level of a single ticket, to the level of several projects, where one can monitor the workloads on engineers also into the future.

In practice this gives another goodie to research institutes: it is possible to start a small project by defining a few RT tickets which brings no significant overhead. If the project becomes interesting (i.e. there is a willingness to invest more work into it), a transition to a larger scale project requires no extra work, the Gantt chart and everything else is already there - just import the existing tickets into GP!

Mailing Lists for Communication

To handle communication on the project, relevant to all the developers, we use mailing lists. Typically, two mailing lists are activated for each project. The first one is a so-called external mailing list, the participants of which are the customer, CRM, project manager, interpreter and all important project members. It is used to keep customer up-to-date on the project progress and to consult with him on project details. The other mailing list is internal – recipients are project manager, all project members and also the interpreter. A content of this list is more tactical - discussions about how to handle this or that specific problem. A mailing list with exactly defined senders and receivers and with archived conversations is much better than a simple e-mail. By our experiences two mailing lists per customer are enough. In case of one mailing list per project (as opposed to one per customer, which may have multiple ongoing projects) confusion would arise, since it would be unclear as to which mailing list to use for a given problem – especially because certain issues span multiple projects anyway. If a need appears (for bigger projects), it is anyway trivial to establish a special list.

ANALISYS

When managing more than a few projects at a time, also higher level analysis of the projects and work spent on them is required. Simple statistical measures and tests can help the project manager to pinpoint potential sources of trouble in their early stages, thus acting proactively and avoiding the mode of "putting out the fires". Such tools are the *Early Warning System*, and the automated reports of the activity the project.

Early Warning System

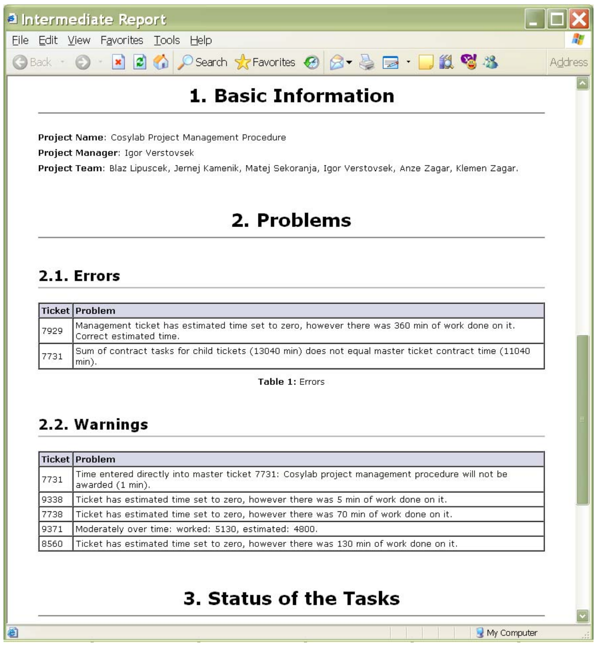
Given the structure and organisation of the RT database, it is relatively easy to detect some types of problems by asking very simple questions for every ticket on the project, such as

- Is the ticket past due? If so, how much?
- Does the ticket have estimated time set?
- Was there way too much work done on this ticket?

Instead of the project manager having to browse through several tens (sometimes even hundreds) of tickets, we are developing a Java program that traverses the ticket hierarchy and produces a HTML report of the problems.

The HTML is generated from CosyDoc XML [3] that we use for writing documentation. The XML is transformed by means of a XSLT transformation and ANT scripts into HTML and PDF format.

The report contains errors and warnings, depending on the severity of an issue. Errors require immediate action by the project manager and warnings might give him a hint that somewhere, somebody is doing something that might somehow not turn out fine for anybody.


Intermediate Report
File Edit View Favorites Tools Help
Back Search Favorites Address
My Computer

1. Basic Information

Project Name: Cosylab Project Management Procedure
Project Manager: Igor Verstovsek
Project Team: Blaz Lipuscek, Jernej Kamenik, Matej Sekoranja, Igor Verstovsek, Anze Zagar, Klemen Zagar.

2. Problems

2.1. Errors

Ticket	Problem
7929	Management ticket has estimated time set to zero, however there was 360 min of work done on it. Correct estimated time.
7731	Sum of contract tasks for child tickets (13040 min) does not equal master ticket contract time (11040 min).

Table 1: Errors

2.2. Warnings

Ticket	Problem
7731	Time entered directly into master ticket 7731: Cosylab project management procedure will not be awarded (1 min).
9338	Ticket has estimated time set to zero, however there was 5 min of work done on it.
7738	Ticket has estimated time set to zero, however there was 70 min of work done on it.
9371	Moderately over time: worked: 5130, estimated: 4800.
8560	Ticket has estimated time set to zero, however there was 130 min of work done on it.

3. Status of the Tasks

Figure 2: HTML report, created automatically by the Early Warning System. All data is gathered from the RT.

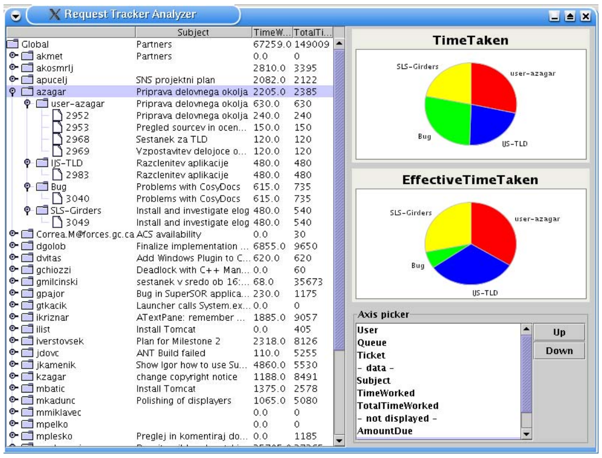
The program can be run automatically on a daily basis or on demand and the results of it will be displayed on the internal pages of the project.

RT Analyser and OLAP Cubes for Statistics

RT Analyser was developed entirely in-house. It offers on-line analytical processing (OLAP) capabilities over the RT database. The data is filled into the so-called OLAP cubes, multi dimensional entities where its dimensions can be whatever we find it to be useful, such as effort (actual and estimated) over projects, resources (engineers) and time.

Through RT Analyser, it is possible to extract and summarize the information from the OLAP cubes. The user requests results using simple drag-and-drop operations, and RT Analyser swiftly responds with graphs and tables. By looking at different "cross sections" of this cube, it is possible to gather many statistically useful data. For this purpose, also commercially available tools for viewing OLAP data can be used, such as the Microsoft Excel.

For example, when looking at the organisation as a whole, one question to ask is how much time spent went for production projects (these are the project that we get paid for), research ("basic" research, development of new products) and overhead (administration, sales and marketing, reading emails, drinking coffee, etc.).



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

This data is extremely useful for future planning and to pinpoint areas that need improvement.

Cosy Evaluator for Final Project Report

In Cosylab, in addition to the fixed part of the salary, the developers also get a variable part based on their performance on the project. Also here we use the data stored in the RT database - we have defined keywords that project manager can assign to grade how efficient a certain developer is on a certain task on a 1-5 scale. In addition, he or she can also evaluate a task qualitatively.

A Java program, an upgrade to the Early Warning System, goes through the entire project hierarchy when the project is finished and makes a validation of data: are all tickets closed and evaluated? If the validation is successful, the program generates an XML project report, with the distribution of awards and the most relevant statistics. This report is then used to evaluate the project manager and stored in the archive for future reference.

CONCLUSION

Every project management system is a trade off between flexibility and ease of use on one hand and order and strictness (that inevitably introduces some bureaucracy) on the other.

The project management procedure described in this article is an example of this, orientated more into the direction of flexibility, but it still gives enough order that it is possible to spot possible problems early, which is one of the goals of project management.

Each organisation must choose its place on this balance, based the nature of its work. Requirements of Cosylab and research institutes are very similar - we all like to do research, but we must still meet deadlines. Therefore we believe that our solution is ideal also for academic projects.

It requires only a minimum of effort and discipline and results in huge benefits, presented in this article.

REFERENCES

[1] I. Verstovsek et al., ANKA Control System Takes Control, PCaPAC 2000, Hamburg, Germany.
[2] Cosylab, http://www.cosylab.com
[3] G. Milcinski et al., e-Management and Quality Assurance, ICALEPCS 2003, Gyeongju, Korea.
[4] G. Milcinski et al., Developing a Control System from a Divan Bed, PCaPAC 2002, Frascatti, Rome.
[5] Swiss Light Source Controls Group, http://www.sls.psi.ch/controls/controls-home.html
[6] Request Tracker, http://www.bestpractical.com/rt/
[7] Gantt Project, http://ganttproject.sourceforge.net/