Business Intelligence applied to the ALMA Software Integration process

MAURICIO ZAMBRANO, CRISTIAN RECABARREN, VICTOR GONZÁLEZ,
ARTURO HOFFSTADT, TZU-CHIANG SHEN, RUBEN SOTO
ALMA Observatory, Santiago, Chile

ABSTRACT

Software quality assurance and planning of an astronomy project is a complex task, specially if it is a distributed collaborative project such as ALMA, where the development centers are spread across the globe. When you execute a software project there is much valuable information about this process itself that you might be able to collect. One of the ways you can receive this input is via an issue tracking system that will gather the problem reports relative to software bugs captured during the testing of the software, during the integration of the different components or even worst, problems occurred during production time. Usually, there is little time spent on analyzing them but with some multidimensional processing you can extract valuable information from them and it might help you on the long term planning and resources allocation. We present an analysis of the information collected at ALMA from a collection of key unbiased indicators. We describe here the extraction, transformation and load process and how the data was processed. The main goal is to assess a software process and get insights from this information.

Introduction

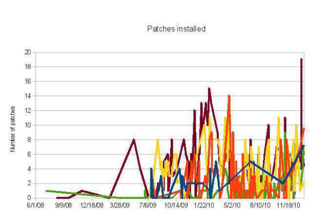
The Atacama Large Millimeter/submillimeter Array (ALMA) is an international partnership. It is the largest astronomical project in existence. ALMA will be a single telescope of revolutionary design, composed of 66 antennas located on the Chajnantor plateau at 5000 meters altitude in northern Chile.
The software development given the characteristics of ALMA is a complex task. Clear measurements and objective indicators are needed to make better informed decisions. This is where Business intelligence comes into action. We analyzed currently available historical data from the ALMA project, and we looked at the current and historical trends. We created key indicators, that will help to make decisions and detect ongoing trends.

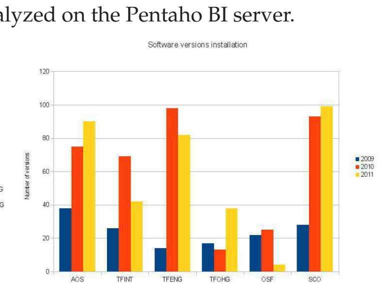
The Software process

The control software of the Array is build on top of the ALMA Common Software[1] (ACS) which is a container-component CORBA framework. The development is leaded and coordinated by the Computing Integrated Product Team (C-IPT). The system integration is done in a Standard Testing Environment (STE) that can be replicated anywhere. An STE is composed by a set of servers with specific roles to run the ALMA software. The hardware is simulated on this environment via special components that mimics the real one's behaviours[2]. Over the time the STE model has evolved to support the production environments.
The final software deployment and integration is done in Chile on STEs at the Operation Site Facilities (OSF) and at the ALMA Operation Site (AOS). In those places, the real hardware is being commissioned, constructed and used for science observations. The ALMA Department of Computing (ADC), coordinated with the C-IPT is in charge of the operation and support of the Chilean STEs.
A weekly regression tests allows to enforce stability, as new patches are being introduced[3].
Each time a software or hardware issue affects the operations, a ticket is created on two tracking system. Both systems are linked by the ticket number. There is also a third system that tracks the exact ALMA software version and patches available at a certain time.

Data extraction, transformation and loading

The data used for this analysis comes from our code repository (CVS) and from 3 systems. ON is a bug tracker (JIRA), the second one registers down times (events.aiv.alma.cl) and the third one keep track of the software versions in use on all the STEs (stesw.aiv.alma.cl).
The simplest extraction method is via scripts. Eventually more complex programs were used to extract the relevant parameters that were not directly available on the systems. With a small number of data we used OOCalc. In order to make the relevant extraction of data, the information was retrieved from multiple relational servers. Some of this information was analyzed on the Pentaho BI server.



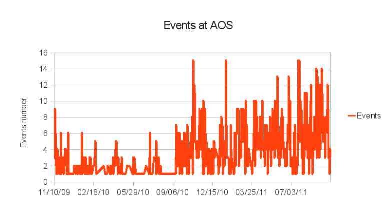


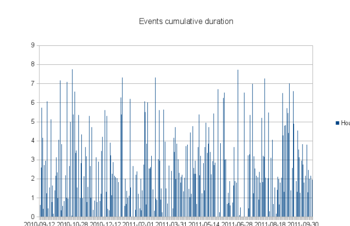
Key indicators

Choosing the right indicators is a complex task. Ideally it has to be a real value meaning something relevant from our Software development and integration processes. We will prefer unbiased indicator that can be easily tracked down to relevant issues.
To analyze the code, we took the source lines of code (SLOC) of C++, headers, idl and java files. They were computed from our CVS system and analyzed on a branch by time intervals. Our HEAD code base had in July 2011 17,151 files and 4,233,651 SLOCs. 44% of the code is java, followed by 15% of C++ and 11% of Python.
We extracted data from the "STESW" database, to get the number of different software versions and patches that where installed on all the STEs during the last 3 years.
Another endicator is the number of patches installed on a specific software version. Patches are directly related to bugs or added functions. Depending of the STE usage we will see a higher number of patches installed. This is specially the case of SCO and TFENG, the simulation and hardware testing STEs.

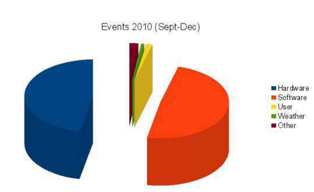
Events trends

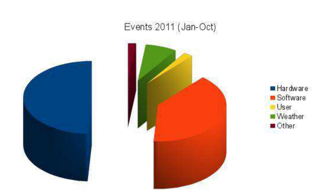
The event system records downtime at AOS during Science time (production). The below charts describe the number and total time of events per day since December 2009. The number of daily events has increased over the time. Although the daily total time lost on all those issues has been stable over the time. Specific long duration events like weather issues were removed from this plot.





The causes of these issues are mostly hardware and software problems. The current trend shows that the hardware issues are slowly increasing. This can be correlated to the fact that the number of antennas has also augmented.





Problems with metrics

It is clear that some metrics are not completely adequate. The priority and severity of the issues is not currently captured. Having a problem on a single antenna is very different from not being able to use the whole array. Although the present numbers are a valid starting point for extracting useful information, they also provide a base for improving the process analysis over the next years.

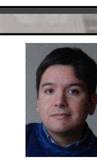
Conclusions

After our analysis, we found that specific changes that could lead to a better tracking of the current process. Patches are used as a generic term that covers bug fixes and new features. We believe that differentiating them will add more value to this information. Getting the relevant information of the CVS transaction would also improve the traceability of the bug fix process. The downtime could be computed based on the number of antennas affected by a failure. So a full observation period would be a total available time x available antennas.

- BI is a powerful analytical tool that can help your project to get insight about your processes. A clear definition on what metrics are relevant to be evaluated is important to design procedures that will help you tracking progresses and issues.
- Looking at the build installation and patches applied, we found a clear correlation between the chosen indicators, the data collected and our impressions.

We plan to continue improving our processes by analyzing in realtime the information from them. To achieve this, many processes have to be automatized on a BI platform.

References
[1] Chiozzi, G. et al., "CORBA-based Common Software for the ALMA project" in [*SPIE Advanced Telescope and Instrumentation Telescope Control Software II*], (2002).
[2] Mora, M. et al. "Hardware device simulation framework in the ALMA Control subsystem" in [*ADASS XVIII*], (2008).
[3] Shen, T.C. et al. "Status of ALMA Software", to be published in [*ICALEPCS XIII*], (2011).



Atacama Large
Millimeter/submillimeter Array

http://www.almaobservatory.org/

Contact e-mail: mzambran@alma.cl / For more information about the ALMA Department of Computing please visit our web site http://adcwiki.alma.cl