Project Management Systems at Gemini Observatory

Summary

This document links to all other documents provided for the review of Gemini's project development and management systems. A standalone PDF version is also available, sans the web links. Through this document and those linked to it, we explain how projects are developed, tracked, and executed at Gemini. This process tends to vary with the size and importance of projects, so particular emphasis is placed here on how Gemini handles large-scale "Transition" and high priority instrument development projects. Through this report we attempt to answer such questions as –

- How much effort is required to support Gemini's Transition plan and can Gemini's staffing model provide this level of effort?
- What is the basis for estimating the resources needed by various projects?
- How are projects scheduled and progress tracked?
- Are risks identified in a manner that they can be effectively dealt with?
- How is software developed and managed at Gemini?

Examples are given in many areas to explain methods used for project management, as well as fairly extensive datasets to illustrate current labor estimates referenced against our 2011 Observatory plan, which is still under development since the October annual planning retreat. It is important to note that the bulk of work done at Gemini consists of day-to-day operations. We are not a development lab but we do apply modern methods for project management to optimize the effort expended on projects being conducted against a persistent "backdrop" of operations dedicated to running a pair of 8 m telescopes on a 24/7 basis.

Management Controls

Project Definition

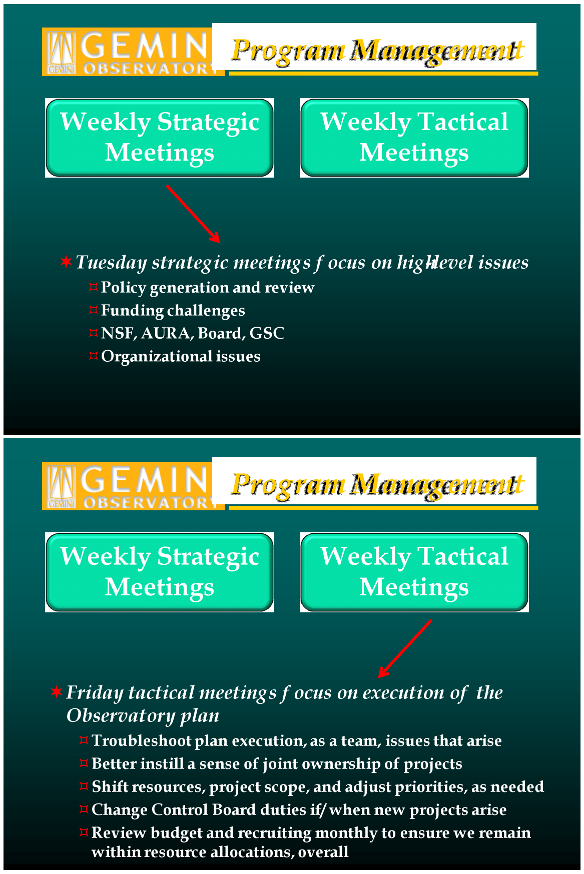
A variety of controls are in place to lead the development of activity within the Observatory and monitor the progress of its execution. While they collectively represent a continuum of activity, work is nominally parsed into annual components for the purpose of linking it to Gemini's finances (reconciled according to the calendar year), annual employee performance evaluation process (explicitly linked to goals during the January -> December period), and to first order the semester schedule associated with science operations (first semester starts Feb 1 each year). The starting point or on-ramp for activity onto this conveyor belt of activity is through an annual planning retreat that occurs in October of each year. Prior to that meeting proposed projects are developed at the level needed to assess their merit (scientific, technical, budget, etc.) and cost (labor and/or cash) to the Observatory. The ranking of these projects has Gemini Science Committee input through a GSC meeting that is timed to occur just prior to the planning retreat. Most recently, given the Board's priorities (listed below), the Directorate has pre-set projects for consideration in this process on the understanding that they are essential to Gemini's Transition proposal and work must proceed on them as a high priority, often on a multi-year basis. The overall process is summarized in the "Observatory Planning Guide" and "Guidelines for Developing New Projects".

The annual planning retreat is an important but not the only mechanism for new projects to be started within the Observatory. It is not unusual for plans to change over the course of a year due to unforeseen shifts in priorities. In that case the Change Control Board (CCB), which meets weekly in the context of other meetings, is in a position to either activate some projects or deactivate others. Examples of this in the past include cancelations of significant numbers of Administrative projects due to a heavy load by unforeseen NSF proposals or reviews. Within engineering, the damage to GNIRS and repairs needed by F-2 all led to changes in engineering plans to accommodate these new projects which placed significant liens on finite resources.

A web interfaced database called Project Insight (PI) is used to record a myriad of project details including project hours and tasks, work assignments, associated documentation, issue threads, etc. This program has been customized by Gemini to suit our specific needs, particularly in the area of reporting tools that can be used to track activity.

Strategic/Tactical Weekly Meetings

In recent years a combination of weekly AD level meetings and one-on-one meetings between the Director and each AD to discuss project execution issues within each branch were conducted. In recognition of the extra oversight needed by our Transition plans to monitor and collectively troubleshoot important projects, these meetings have been morphed into a weekly AD meeting for discussions of top-level strategic issues (new policies, broad staff issues, community interfaces issues, etc.) and much more focused weekly meetings dedicated to discussing issues arising with the execution of projects and/or operations activity. The latter also include on a monthly basis assessments of year-to-date budget information and current recruiting status. This helps ensure that our spending is consistent with budget constraints and positions which are open are discussed to determine if they should be filled with full time replacements, contract term replacements, outside contracted help, or terminated completely. This is an important consideration given our strategy of handling the bulk


GEMINI OBSERVATORY

Program Management

Weekly Strategic Meetings

Weekly Tactical Meetings

Tuesday strategic meetings focus on high-level issues
- Policy generation and review
- Funding challenges
- NSF, AURA, Board, GSC
- Organizational issues

GEMINI OBSERVATORY

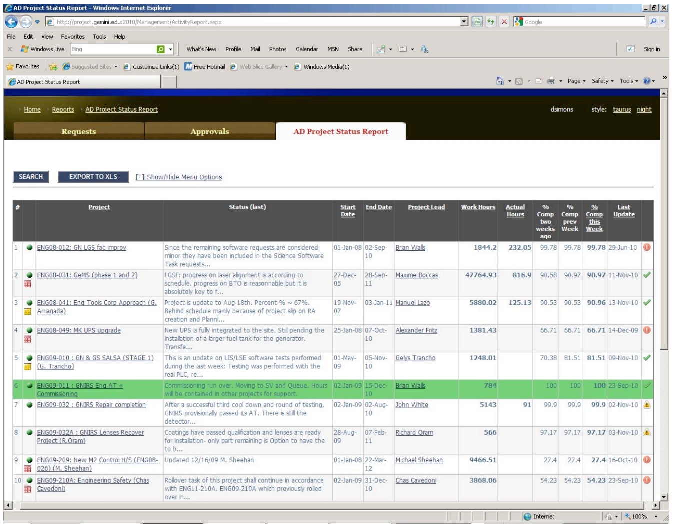
Program Management

Weekly Strategic Meetings

Weekly Tactical Meetings

Friday tactical meetings focus on execution of the Observatory plan
- Troubleshoot plan execution, as a team, issues that arise
- Better instill a sense of joint ownership of projects
- Shift resources, project scope, and adjust priorities, as needed
- Change Control Board duties if/when new projects arise
- Review budget and recruiting monthly to ensure we remain within resource allocations, overall

Figure 1 – The nature of and difference between so-called weekly strategic and tactical meetings is shown through a pair of slides recently presented to the Gemini staff to explain our new meeting structure.



AD Project Status Report - Windows Internet Explorer
http://project.gemini.edu:2010/Management/ActivityReport.aspx
Google
File Edit View Favorites Tools Help
Windows Live
Bing
What's New Profile Mail Photos Calendar MSN Share
Sign in
Favorites
Suggested Sites
Customize Links(1)
Free Hotmail
Web Slice Gallery
Windows Media(1)
AD Project Status Report
Page Safety Tools
Home Reports AD Project Status Report
dsimons style: taurus night
Requests
Approvals
AD Project Status Report
SEARCH
EXPORT TO XLS
[-] Show/Hide Menu Options
Internet
100%

#		Project	Status (last)	Start Date	End Date	Project Lead	Work Hours	Actual Hours	% Comp two weeks ago	% Comp prev Week	% Comp this Week	Last Update	
1	●	ENG08-012: GN LGS fac improv	Since the remaining software requests are considered minor they have been included in the Science Software Task requests...	01-Jan-08	02-Sep-10	Brian Walls	1844.2	232.05	99.78	99.78	99.78	29-Jun-10	!
2	● ▦	ENG08-031: GeMS (phase 1 and 2)	LGSF: progress on laser alignment is according to schedule. progress on BTO is reasonnable but it is absolutely key to f...	27-Dec-05	28-Sep-11	Maxime Boccas	47764.93	816.9	90.58	90.97	90.97	11-Nov-10	✓
3	● ▦	ENG08-041: Eng Tools Corp Approach (G. Arriagada)	Project is update to Aug 18th. Percent % ~ 67%. Behind schedule mainly because of project slip on RA creation and Planni...	19-Nov-07	03-Jan-11	Manuel Lazo	5880.02	125.13	90.53	90.53	90.96	13-Nov-10	✓
4	● ▦	ENG08-049: MK UPS upgrade	New UPS is fully integrated to the site. Still pending the installation of a larger fuel tank for the generator. Transfe...	25-Jan-08	07-Oct-10	Alexander Fritz	1381.43		66.71	66.71	66.71	14-Dec-09	!
5	● ▦	ENG09-010 : GN & GS SALSA (STAGE 1) (G. Trancho)	This is an update on LIS/LSE software tests performed during the last week: Testing was performed with the real PLC, re...	01-May-09	05-Nov-10	Gelys Trancho	1248.01		70.38	81.51	81.51	09-Nov-10	✓
6	●	ENG09-011 : GNIRS Eng AT + Commissioning	Commissioning run over. Moving to SV and Queue. Hours will be contained in other projects for support.	02-Jan-09	15-Dec-10	Brian Walls	784		100	100	100	23-Sep-10	✓
7	●	ENG09-032 : GNIRS Repair completion	After a successful third cool down and round of testing, GNIRS provisionally passed its AT. There is still the detector...	02-Jan-09	02-Aug-10	John White	5143	91	99.9	99.9	99.9	02-Nov-10	⚠
8	●	ENG09-032A : GNIRS Lenses Recover Project (R.Oram)	Coatings have passed qualification and lenses are ready for installation- only part remaining is Option to have the to b...	28-Aug-09	07-Feb-11	Richard Oram	566		97.17	97.17	97.17	03-Nov-10	⚠
9	● ▦	ENG09-209: New M2 Control H/S (ENG08-026) (M. Sheehan)	Updated 12/16/09 M. Sheehan	01-Jan-08	22-Mar-12	Michael Sheehan	9466.51		27.4	27.4	27.4	16-Oct-10	!
10	● ▦	ENG09-210A: Engineering Safety (Chas Cavedoni)	Rollover task of this project shall continue in accordance with ENG11-210A. ENG09-210A which previously rolled over in...	02-Jan-09	31-Dec-10	Chas Cavedoni	3868.06		54.23	54.23	54.23	23-Sep-10	!

Figure 2 – A typical report, generated by PI, showing various activity metrics is shown. These types of reports substantially serve as the basis for weekly meetings within the Directorate to identify issues with projects and address them in a timely manner.

of our labor reduction through attrition over the next several years.

These meetings use rolling agendas, generate action lists with individuals identified to address actions by agreed dates, and are recorded through a set of minutes. They are a continuing dialog within Gemini's Directorate that provides an important mechanism to track activity and take timely action to address issues that arise. A special reporting tool, built into PI is used during these meetings (see Figure 2). These reports include brief summaries updated weekly by project leaders and completion statistics for recent weeks to allow performance trends to be quickly spotted. In addition the meetings are structured to allow each AD to raise topics of broad interest and for everyone to ask questions of projects of particular interest. A good example of the type of discussions held and decisions reached at these meetings includes the "rewiring" of software effort in mid 2010 from the GMOS-N CCD project to MCAO, when it became clear that delays were going to stall progress with the GMOS-N CCD effort. Another example was the deferral of work on the new A&G replacement project when it became clear at the detailed level that we have resource conflicts with MCAO in early 2011. Beyond redirecting effort to when/where it is needed against a consistent set of priorities, this group also reviews cash and labor (current and planned) and represents the nexus of project monitoring and execution within the Directorate. Given how well these meetings function and the basis of information they rely upon, it is difficult to imagine that major Transition projects could languish for prolonged periods of time or suffer from inadequate resource allocation given this persistent level of oversight.

Regular Internal Reports & Meetings

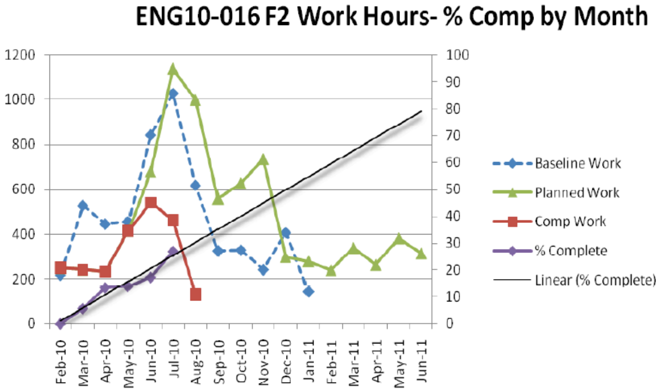
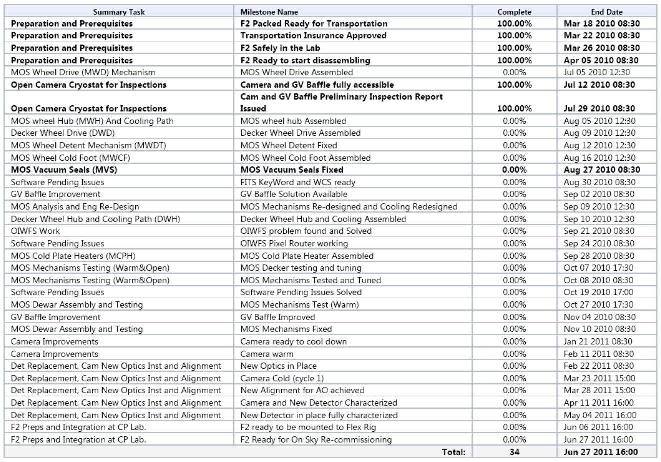
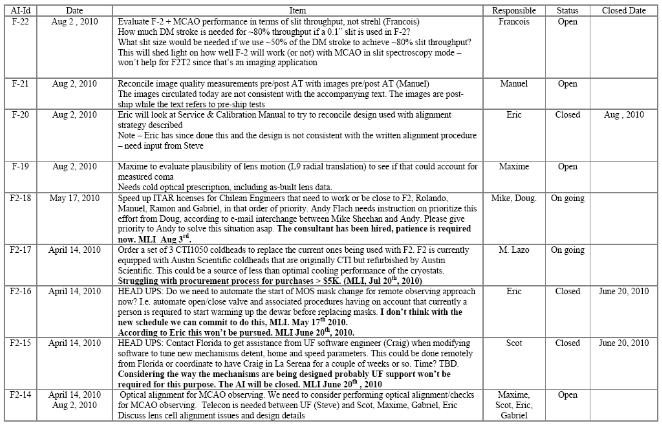


Figure 3 – An example of effort tracking is shown for the FLAMINGOS-2 rework project. This shows the original planned hours, the re-baselined plan after additional problems were found, the actual work completed, and the cumulative hours to date.

Beyond reporting of basic metrics like effort expended and recent milestones achieved, major projects also use internal reports to document findings and/or describe activity. These reports are used to keep an often broad audience within the Observatory up-to-date in a manner that is easily traceable and understandable. They pull together a wealth of project information and are represented by the monthly project reports used for the F-2 repair effort. Figure 3 shows an example of month-by-month effort expended on the F-2 repair project, benchmarked against the original baseline and a new baseline generated after a thorough examination of the instrument was completed. The cumulative effort, extrapolated forward gives an indication of when the project will be completed. Similar types of effort tracking, when applied to Transition projects, will be quite useful in monitoring their progress. Other metrics are posted within such reports as can be seen in Figure 4. These include running milestone summaries and action item lists. Beyond hours of effort expended vs. planned hours, planned milestones are a useful metric as they define tangible steps in the path to completion. They provide a related but still different perspective on the status of a project compared to overall percent completion.



Summary Task	Milestone Name	Complete	End Date
Preparation and Prerequisites	**F2 Packed Ready for Transportation**	**100.00%**	**Mar 18 2010 08:30**
Preparation and Prerequisites	**Transportation Insurance Approved**	**100.00%**	**Mar 22 2010 08:30**
Preparation and Prerequisites	**F2 Safely in the Lab**	**100.00%**	**Mar 26 2010 08:30**
Preparation and Prerequisites	**F2 Ready to start disassembling**	**100.00%**	**Apr 05 2010 08:30**
MOS Wheel Drive (MWD) Mechanism	MOS Wheel Drive Assembled	0.00%	Jul 05 2010 12:30
Open Camera Cryostat for Inspections	**Camera and GV Baffle fully accessible**	**100.00%**	**Jul 12 2010 08:30**
Open Camera Cryostat for Inspections	**Cam and GV Baffle Preliminary Inspection Report Issued**	**100.00%**	**Jul 29 2010 08:30**
MOS wheel Hub (MWH) And Cooling Path	MOS wheel hub Assembled	0.00%	Aug 05 2010 12:30
Decker Wheel Drive (DWD)	Decker Wheel Drive Assembled	0.00%	Aug 09 2010 12:30
MOS Wheel Detent Mechanism (MWDT)	MOS Wheel Detent Fixed	0.00%	Aug 12 2010 12:30
MOS Wheel Cold Foot (MWCF)	MOS Wheel Cold Foot Assembled	0.00%	Aug 16 2010 12:30
MOS Vacuum Seals (MVS)	**MOS Vacuum Seals Fixed**	**0.00%**	**Aug 27 2010 08:30**
Software Pending Issues	FITS KeyWord and WCS ready	0.00%	Aug 30 2010 08:30
GV Baffle Improvement	GV Baffle Solution Available	0.00%	Sep 02 2010 08:30
MOS Analysis and Eng Re-Design	MOS Mechanisms Re-designed and Cooling Redesigned	0.00%	Sep 09 2010 12:30
Decker Wheel Hub and Cooling Path (DWH)	Decker Wheel Hub and Cooling Assembled	0.00%	Sep 10 2010 12:30
OIWFS Work	OIWFS problem found and Solved	0.00%	Sep 21 2010 08:30
Software Pending Issues	OIWFS Pixel Router working	0.00%	Sep 24 2010 08:30
MOS Cold Plate Heaters (MCPH)	MOS Cold Plate Heater Assembled	0.00%	Sep 28 2010 08:30
MOS Mechanisms Testing (Warm&Open)	MOS Decker testing and tuning	0.00%	Oct 07 2010 17:30
MOS Mechanisms Testing (Warm&Open)	MOS Mechanisms Tested and Tuned	0.00%	Oct 08 2010 08:30
Software Pending Issues	Software Pending Issues Solved	0.00%	Oct 19 2010 17:00
MOS Dewar Assembly and Testing	MOS Mechanisms Test (Warm)	0.00%	Oct 27 2010 17:30
GV Baffle Improvement	GV Baffle Improved	0.00%	Nov 04 2010 08:30
MOS Dewar Assembly and Testing	MOS Mechanisms Fixed	0.00%	Nov 10 2010 08:30
Camera Improvements	Camera ready to cool down	0.00%	Jan 21 2011 08:30
Camera Improvements	Camera warm	0.00%	Feb 11 2011 08:30
Det Replacement, Cam New Optics Inst and Alignment	New Optics in Place	0.00%	Feb 22 2011 08:30
Det Replacement, Cam New Optics Inst and Alignment	Camera Cold (cycle 1)	0.00%	Mar 23 2011 15:00
Det Replacement, Cam New Optics Inst and Alignment	New Alignment for AO achieved	0.00%	Mar 28 2011 15:00
Det Replacement, Cam New Optics Inst and Alignment	Camera and New Detector Characterized	0.00%	Apr 11 2011 16:00
Det Replacement, Cam New Optics Inst and Alignment	New Detector in place fully characterized	0.00%	May 04 2011 16:00
F2 Preps and Integration at CP Lab.	F2 ready to be mounted to Flex Rig	0.00%	Jun 06 2011 16:00
F2 Preps and Integration at CP Lab.	F2 Ready for On Sky Re-commissioning	0.00%	Jun 27 2011 16:00
	Total:	**34**	**Jun 27 2011 16:00**



AI-Id	Date	Item	Responsible	Status	Closed Date
F-22	Aug 2 , 2010	Evaluate F-2 + MCAO performance in terms of slit throughput, not strehl (Francois) How much DM stroke is needed for ~80% throughput if a 0.1" slit is used in F-2? What slit size would be needed if we use ~50% of the DM stroke to achieve ~80% slit throughput? This will shed light on how well F-2 will work (or not) with MCAO in slit spectroscopy mode – won't help for F2T2 since that's an imaging application	Francois	Open	
F-21	Aug 2, 2010	Reconcile image quality measurements pre/post AT with images pre/post AT (Manuel) The images circulated today are not consistent with the accompanying text. The images are post-ship while the text refers to pre-ship tests	Manuel	Open	
F-20	Aug 2, 2010	Eric will look at Service & Calibration Manual to try to reconcile design used with alignment strategy described Note – Eric has since done this and the design is not consistent with the written alignment procedure – need input from Steve	Eric	Closed	Aug , 2010
F-19	Aug 2, 2010	Maxime to evaluate plausibility of lens motion (L9 radial translation) to see if that could account for measured coma Needs cold optical prescription, including as-built lens data.	Maxime	Open	
F2-18	May 17, 2010	Speed up ITAR licenses for Chilean Engineers that need to work or be close to F2, Rolando, Manuel, Ramon and Gabriel, in that order of priority. Andy Flach needs instruction on prioritize this effort from Doug, according to e-mail interchange between Mike Sheehan and Andy. Please give priority to Andy to solve this situation asap. **The consultant has been hired, patience is required now. MLI Aug 3rd.**	Mike, Doug.	On going	
F2-17	April 14, 2010	Order a set of 3 CTI1050 coldheads to replace the current ones being used with F2. F2 is currently equipped with Austin Scientific coldheads that are originally CTI but refurbished by Austin Scientific. This could be a source of less than optimal cooling performance of the cryostats. **Struggling with procurement process for purchases > $5K. (MLI, Jul 20th, 2010)**	M. Lazo	On going	
F2-16	April 14, 2010	HEAD UPS: Do we need to automate the start of MOS mask change for remote observing approach now? I.e. automate open/close valve and associated procedures having on account that currently a person is required to start warming up the dewar before replacing masks. **I don't think with the new schedule we can commit to do this, MLI. May 17th 2010.** **According to Eric this won't be pursued. MLI June 20th, 2010.**	Eric	Closed	June 20, 2010
F2-15	April 14, 2010	HEAD UPS: Contact Florida to get assistance from UF software engineer (Craig) when modifying software to tune new mechanisms detent, home and speed parameters. This could be done remotely from Florida or coordinate to have Craig in La Serena for a couple of weeks or so. Time? TBD. **Considering the way the mechanisms are being designed probably UF support won't be required for this purpose. The AI will be closed. MLI June 20th , 2010**	Scot	Closed	June 20, 2010
F2-14	April 14, 2010 Aug 2, 2010	Optical alignment for MCAO observing. We need to consider performing optical alignment/checks for MCAO observing. Telecon is needed between UF (Steve) and Scot, Maxime, Gabriel, Eric Discuss lens cell alignment issues and design details	Maxime, Scot, Eric, Gabriel	Open	

Figure 4 – Above, milestones as defined in the project plan, are itemized and tracked as an important metric of progress. Below, the monthly action item list is shown. This includes names of project members assigned to various actions and due dates for when each action item should be closed.

These reports are used to support various internal project meetings as they represent the focal point of many separate components of the project. Meetings occur at different intervals but major project meetings generally occur weekly, either group-wide, within a specific subset of the team (the laser group), between the systems group and project team, or between the AD for Engineering and the other engineering managers. In this manner meetings are tuned for content and frequency to optimally involve key project members and pull in non-project members of engineering, science, procurement, or the Directorate, as needed. The results of weekly team meetings and recent progress or issues are generally circulated through internal e-mail exploders (used by the GNIRS team) or through a number of internal blogs (used by the MCAO team). These electronic communication mechanisms provide a convenient platform for key staff members to become aware of issues, as they arise. For sufficiently high priority meetings, members of the Directorate participate as well. For example weekly GNIRS meetings were frequented by either the AD for Engineering, AD for Development, or Director when significant issues were planned to be discussed that stood to impact performance, cost, or schedule. By keeping informed through the aforementioned weekly Tactical meetings or various electronic communications, the timely involvement of senior management in key decisions is ensured. Today, either the Director or Deputy Director attends the weekly MCAO management meetings, which involve core members of the MCAO team and systems engineering. We do this to make sure that if/when resource bottlenecks emerge (e.g., operations issues unexpectedly arise), they are addressed immediately. Directorate involvement in these meetings also helps reinforce on the project team the importance of maintaining momentum in an often complex work environment.

2011 Project Information

A variety of information is available regarding Gemini's 2011 Observatory plan, which is still being refined prior to the beginning of its execution in January 2011. First, a spreadsheet is available that lists each project currently under consideration for activation in 2011. It is important to note that many of these "projects" are O&M related and are included primarily to ensure that we track effort available for high priority Transition projects accurately. For each project we list the project code (used for internal tracking), name, a summary description, the project team members (sponsor, manager/leader, systems engineer if applicable, and project scientist if applicable). The total FTEs required for each project is also listed to give an indication of scale for each type of activity. For the highest priority activity Project Data Sheets are available which list additional information about each project. These were generated soon after the October planning retreat and are still being updated but they give a snapshot of the nature and issues associated with many of the most important projects Gemini is either working on now or considering for the future. Immediately after the planning retreat Gemini's management team focused in defining and prioritizing the core or "top 10" projects that should be assigned effort, consistent with the Board and GSC priorities expressed earlier in the year. That list appears below.

1. **OBS11-008**: New Cooperative Agreement
2. **OBS10-006**: GMOS-N CCDs Project Overview (Scot K.)
    1. **OBS10-006A**: GMOS New CCDs (J. White) (ENG10-004)
    2. **OBS10-006A1**: GMOS-N CCD science commissioning (SCI10-244)
    3. **OBS10-006C**: GMOS DR for new CCDs (SCI11-606)
3. **OBS09-006A**: GeMS (Phase 3-4-5) (M. Boccas)
    1. **OBS09-006B**: GSAOI Science Commissioning (SCI11-209)
    2. **OBS09-006C**: GSAOI Data Reduction Software (SCI11-601)
4. Observatory Software
    1. **OSW11-200**: ITAC Phase (1+2) (D. Dawson)
    2. **OSW11-201:** Time Accounting Timeline (S. Walker)
    3. **OSW11-202**: SALSA/Sci Ops Software-LCH Clearances (A. Núñez)
    4. **SCI11-620**: QA Pipeline (K. Labrie)
5. **OBS10-007A**: F2 Fixes and Improvements
    1. **OBS10-007B**: F2 Science Commissioning
    2. **OBS10-007C**: Data Reduction Software for Flamingos 2

6. **OBS11-003**: GPI Observatory Project
7. Observatory Software
    1. **OSW11-207:** ODB Replacement Design Study (S. Walker.)
8. **OBS11-001**: New High-Resolution Optical Spectrograph (Scot K.)
9. **OBS11-005**: 2nd Generation Acquisition and Guidance Unit Observatory Project
10. **OBS11-501:** Base Facility Operations

A Gantt chart and task list is provided for each of these projects. Some are bundled together in several linked projects, like the GMOS CCD upgrade project, while others are defined through single project files. The level of detail reflected in these projects varies considerably. For example MCAO, which is a large multi-year project that has been split into 5 phases has nearly a thousand tasks in its project files. Others are more modest in scope and therefore detail. Several are still in a development phase (e.g. OSW11-207), having only been approved in October during the planning retreat. Despite the differences in maturity and details of these project plans, which to first order scale with complexity and age of each project, this list is extremely important in identifying for the staff which projects they should be focused on during 2011 (and beyond in some cases).

Projects further down the draft list generated during the planning retreat are currently under final review for activation in 2011. To determine their final prioritization each Associate Director has been tasked with working within their groups and across groups as required to determine which projects will be pursued next year. Their recommendations will come before our Change Control Board over the remainder of 2010 to bring the process of defining activity for next year to closure. While the “top 10” list of projects is substantially driven by Board priorities, other projects are prioritized by second order constraints including detailed scheduling, GSC priorities, and available FTEs. An important tool used to round-out the 2011 projects is an FTE load-leveling tool built into PI. This reporting system generates a report for each member of the staff, broken down by labor hours needed by proposed projects for each month of the year. Staff members who are overloaded are easily identified in these reports and “tall poles” in the assigned project work identified quickly. With that information in hand Gemini’s managers can level the work load over the course of the year through a range of options including adjusting the start/stop dates for the activity, changing the scope of the effort, or moving projects into “Band 2”, where they are held until effort is freed up to activate them.

The output of the PI load-leveling reporting tool is shown here in the form of a set of spreadsheets reflecting major functional groups within Gemini’s org-chart (e.g., Gemini-N science, Admin, Cerro Pachon, etc.). These reports were generated in mid November and reflect the state of our load-leveling exercise at that time. Again, this process is on-going as we close out our annual planning exercise in the remainder of 2010. Good examples of where we have reasonable levels of FTEs assigned to O&M and new projects are generally found in the Administrative group. In contrast, some of the most excessive overloads detected through this mechanism (shown as red labor hours in the spreadsheet) are associated with the Electronics and Instrumentation Group (EIG). This group emerged in the Santiago planning retreat as a clear “tall pole” in nearly all project prioritization scenarios so it is not a surprise to see this issue remain, albeit in much greater detail, through this load-leveling report.

Finally, the total effort needed by the O&M projects on a month-to-month basis is available here. We provide this as an indication of the baseline effort needed to “keep the lights on” at Gemini. Since Gemini is of course primarily focused on day-to-day operations, the bulk of our available labor is sunk into this type of activity. The confidence of the labor estimates for this O&M effort is generally higher than it is for new proposed work simply because it stems from actual experience. Again, while this activity in some sense does not fit neatly into a “project” category, it is included in this analysis since it sets an important resource boundary condition for determining what non-O&M activity is pursued each year.

Contingency

All of the labor hours loaded into PI have some level of contingency built in. The exact level has been left for the ADs (work sponsors) to define for each project but a typical level is ~20%. This helps ensure that effort is available in the event projections are inaccurate and/or extra effort is needed to handle unanticipated work. Major projects generally have schedule contingency incorporated into their task lists (e.g., OBS11-003: Gemini Planet Imager) to identify when contingency is available. Minor projects (e.g. public outreach) are not required to have explicit contingency in their task lists simply because the nature/risk of these projects does not justify that level of project definition and management. In these cases, a primary purpose of Gemini’s planning system is to determine agreed performance goals for consideration in annual performance evaluations, and to increase the visibility of work done across the observatory through a shared database and the staff-wide dialog that accompanies the generation of these lower priority but still important projects.

In addition to labor contingency, cash contingency is held and managed by the Director. This has been earmarked to offset a number of possible liens on Gemini’s budget ranging from peso exchange rate fluctuations to temporary hires. In practice this budgetary line item (typically ~$350k/yr) has been managed very conservatively and not used very much in recent years. In addition to carrying contingency the Observatory adopted a strategy nearly two years ago to execute a partial hiring freeze, when the first indications of a serious budgetary shortfall became clear. This, along with significant savings elsewhere (e.g., electricity and travel) have provided a limited cash reserve that is being used to fund near-term components of our Transition plans including out-sourcing software projects, possible severance/retirement packages, etc. Based upon the most recent estimates of partner contributions (per the November 2010 Gemini Board meeting) and Gemini’s long-range (through 2015) resource estimates consistent with our Transition plans, a cash surplus of a few million dollars is projected by 2015 which currently remains unallocated. Significant uncertainty exists on these timescales but as an exercise in balancing projected contributions and expenditures, this budget forecast lends considerable credibility to the financial component of our transition plans.

Figure 5 shows historic and future labor (FTEs) and non-labor costs, consistent with our Transition plans. To hold labor costs essentially flat during the next 5 years will require a reduction in positions at Gemini that leaves a staff size comparable to its 2007 size. This allows for modest growth in salaries throughout this period to help ensure competitiveness in what is anticipated to be a better job market as the global economy recovers from the 2008 Wall Street collapse. In addition considerable savings need to be achieved with our non-labor expenditures. Once they are substantially realized in 2011, those up-front savings will be essentially fixed over time as we maintain a ~$11M budget for non-labor expenses.

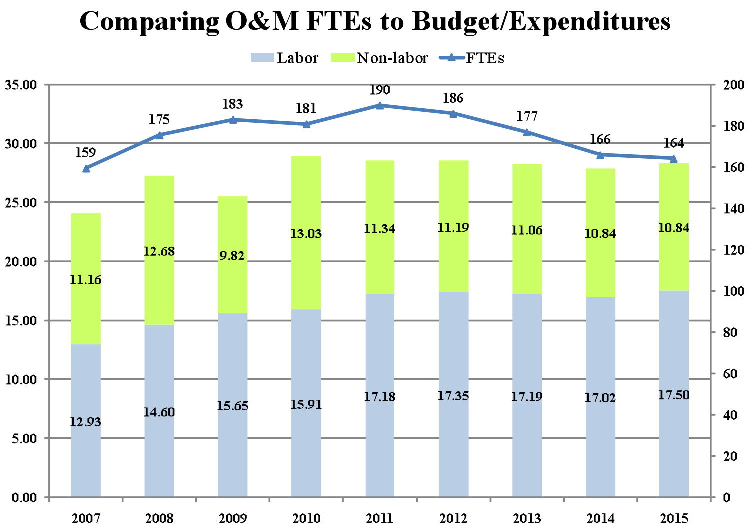


Figure 5 – Labor and non-labor expenditures, both past and future under Gemini's Transition Plan, as well as FTEs are shown in a single plot. Note that the long range (~2015) staff size is comparable to the 2007 level in this plan. It also requires holding essentially flat non-labor expenses through the next 5 years.

Off Ramps

In addition to the use of fairly rigorous resource management methods (FTE estimation and tracking, contingency allocation for schedule and cash, etc.) we are also evaluating off-ramps which effectively yield functional contingency in the event Transition projects are either not completed on schedule or, in some cases, not at all. In general these off-ramps translate into reducing the quality or quantity of our scientific product, or perhaps long-term higher costs for reduced near-term risk. At this point we are focused on identifying what forms of functional contingency we realistically have and when in the development lifecycle of Transition projects that contingency may be used as part of our broader risk mitigation strategy. One of our larger Transition projects is OBS11-501 (Base Facility Operations) which is designed to allow nighttime science operations to be conducted from Hilo or La Serena. Logical decision points in the design phase of this project occur at the CoDR, PDR, and CDR when the benefits of additional automation needs to be balanced against the long term cost of human presence. For example, given the likelihood of failure for and redundancy in our helium compressors, one could envision a trade being made between the need for remote on/off switches for these compressors vs. reliance upon summit based technicians to switch on backup pumps in the event of a failure. Likewise some failures may be of a nature that summit technicians should only place the telescope in a safe state instead of actively troubleshooting the system when a failure occurs, relying

Telescopes from Afar
An international conference on remotely operated, automated, and robotic ground based telescopes
February 28 – March 3, 2011
Waikoloa Beach Marriott, Hawai'i

Figure 6 – An indication of the migration of summit operations to base facilities around Mauna Kea is the "Telescopes from Afar" conference that CFHT is organizing in 2011.

upon Gemini engineering to arrive on-site the next day to work on the system. Additional trades will doubtless occur in the form of COTS equipment for remote diagnostics vs. custom systems that may offer greater ease of use or functionality but at greater cost. It is easy to envision these trades being outlined and reviewed within the structure of this project and then executed during the design phase to systematically retire risks.

It is important to note that this model is predicated upon the use of a joint-technical team that serves Gemini in conjunction with our neighboring observatories. In Hawaii, by the time Gemini is in a position to begin the “trial period” for this new operating mode, telescopes on the entire upper/eastern ridge of the summit will be operated from base facilities in Waimea, Hilo, or Honolulu. Experiences gained from these other observatories can be gleaned for use in Gemini’s base facility plans and shared through workshops like the planned “Telescopes from Afar” conference that CFHT is organizing in 2011. Also, the safety of the telescopes will be central to the review process, which will involve expert engineers from outside Gemini to evaluate risks and help the Observatory make reasonable and well informed decisions. While the cost savings for this project are impressive over time (~$500K/yr), they represent a relatively small fraction of the reduction in contributions Gemini must handle with the UK withdrawal. This initiative is being driven to leave the Observatory in a more robust engineering state while we have the larger staff size over the next few years. If it does not go forward, we would likely have to make further staff cuts which would be difficult but manageable compared to what is already necessary. Furthermore, *Gemini’s plan to use non-research staff to conduct queue observations does not rely on the success of this project*. The restructured science operations team will be capable of operating the telescope from either the summits or base facilities. This project has no reliance on the time critical software development and, in fact, could be stretched in duration using a smaller engineering staff, yielding the desired long term cost savings albeit at a later time. Given the range of off-ramps and re-scoping possible with this project and clear benefits of migrating to base facility operations (particularly in Hawaii), we feel confident that the goals of this project are achievable with an acceptable level of risk.

Project SCI11-102 (Science Operations Training and Documentation) is central to our Transition plans as it is the core activity through which we adapt to the use of non-research staff to run ~3/4 of the nighttime operations (the queue). In this case risk mitigation is already underway as we are already training staff for this role to help make sure that we have an adequate sized team in place when we need it, recognizing the possibility of losing some fraction of our current staff (most likely the junior members) under the circumstances. Activating this part of the Transition plan now also leads to the early formulation of important documentation that will be used to cross-train additional non-research staff in the future, in essence allowing the program to accelerate in time. In the event we do not have non-research staff in place and the associated new software to enable this mode, off-ramp options include –

- Extend PhD staff contracts (budget implications)
- Enlist NGO and external scientist support
- Reduce DAS and/or SSA functional duties to increase effort available for nighttime operations

- Increase classical observing w/out significant science support (transfers costs elsewhere)

A related project is SCI11-620 (QA Pipeline) which is intended to provide on-line automatic data quality assessment to simplify nighttime operations. The design of this system is highly modular allowing flexibility in which instruments and modes are developed, i.e. allowing us to match their development to the arrival of new instruments and/or changing community demands of existing instruments. An obvious off-ramp in the event this software is not available in time is to only perform "spot checks" of data, as it is acquired, and to accept the higher incidents of data not meeting PI specified requirements. The Gemini Board has made it clear that this sort of trade between software capability, queue efficiency, and data quality is acceptable under the circumstances, meaning we may operate for a fixed time using spot checks while the QA pipeline is being completed without disastrous consequences.

As a final example of Transition software that has credible off-ramps we note OSW11-202 (Sci Ops Laser Clearing House Clearances). This software is intended to fully automate the laser interlock and shuttering system with telescope pointing and closure windows on the sky that the Laser Clearing House (LCH) has issued. This has the positive parallel effect of allowing Gemini to reduce its current 0.25 deg satellite avoidance cone to 0.1 deg, which we suspect will significantly reduce satellite closures. Switching to a fully automated system simplifies nighttime laser AO operations, which now rely upon manually shuttering the laser as a member of the night staff monitors telescope pointing and the clock. Off-ramps for this project include deferring it into the future and using the current (manual) system, or supporting fewer laser AO nights thereby reserving staff responsible for running the queue. No longer participating in the LCH program is also an option with NSF approval.

These and other examples represent of the types of functional contingency or off-ramps that are available to the Observatory in the event Transition projects are not completed in time. Key to these off-ramps is the Gemini Board's willingness to accept reduced efficiency or data product as a result of budget reductions. While the Observatory will strive to minimize the impact of these budget cuts to our community, the clear understanding between the Observatory and Board on this key point underpins much of our philosophy about off-ramps, which are an essential part of managing the entire program.

Software

Considerable attention is being paid to the development of new software under Gemini's transition plan. Beyond the aforementioned off-ramps in the event new software is not available at the times prescribed in our project schedules, we present in this section a variety of information about our software development system. The information listed below stem from a specific set of questions posed by Jim Fanson recently.

Plans for Developing and Reviewing the Software Architecture

As an example we offer the process used during the requirements gathering stage of the ITAC phase 1 software project. In this case user requirements were reviewed in detail by two developers and the software group manager. Follow-up discussions with the project scientists to clarify core issues were also conducted. The group identified several high level concerns that the architecture needed to support including software usability, rapid development, and system inputs and outputs. Other practical concerns that influence the system architecture include the projected people (both who and how many) and the rough schedule that guided the project's development. The group then proceeded over a 2-3 week period of white boarding sessions, email threads, technical advocacy, and research to settle several of the fundamental early questions in the project.

One such fundamental question revolved around how to manage the persistent storage of information. Options identified include the Observatory wide Science Program Database (the core of the high-level software), a standard industry relational database, or one of the newer data stores that are evolving to support high transaction environments. Each of the options was weighed against relevant factors including longevity of solution, "custom-ness" of solution (contributing to long-term maintenance burden and training efforts) and the speed with which it could be adapted to the project.

Another fundamental question was the choice of deployment platform. Rich client, thin client, and web applications are all possibilities and each solution offered a different level of support to the project objectives (client application versus web application which feed into trades of software usability and long-term maintenance costs). Research and discussion of the alternatives followed a similar set of steps as the persistence store discussion.

Following the architectural discussions, the group came into rough agreement and moved into a development stage that is ongoing. This process did not generate high-level documentation of the architectural choices made. That documentation is contained within the products unique to the project, ranging from source code (and accompanying documentation) to project description files to simple README files.

The choice of Maven as a central software project management tool couples the overall architecture to the source level products and supports the developer by providing a live architecture to generate the actual product. For more information, see the Appendix documenting the tools we have selected for use with the transition projects.

Metrics Used to Track Software Development

At the moment the tool we use to track development is PI, the same database used to store most project information. This makes it convenient to reconfirm the project plan on a monthly basis and make sure team members understand their responsibilities and provide their commitment during the next period. For example -

- Activities they agree to perform
- Dates they agree they will start and end these activities
- Amount of person-effort they agree they will need to perform these activities

During the performance period (monthly basis), team members record information on the following:

- Completed intermediate and final deliverables

- Dates to reach milestones
- Dates to start and end activities
- Number of work-hours needed for each activity
- Expenditures made for each activity

If necessary, corrective action is taken to bring the project's performance back into conformance with the plan. Furthermore, it is important to keep people informed by sharing achievements, problems, and future plan with all the project stakeholders.

In general our project management Information system is used to apply procedures, equipment and other resources for collecting, analyzing, storing and reporting information that describes project performance. Through it we consider inputs (raw data that describes selected aspects of project performance), processes (analyses of the data to compare actual performance with planned) and outputs (reports presenting the result of the analysis).

Key performance metrics include -

- **Monitoring Schedule Performance (Time):** Tracking activities, identified through the deliverables extracted from the project structure, their start/end dates and dates of project milestones. (PI provides a computer-based tracking system as a vehicle to support project schedule performance)
- **Monitoring Work Effort (Resources):** Comparing effort expended with effort planned is of course important. In this case a pending significant change at Gemini is to modify the way we uniformly track hours so that actual work instead of estimates of percent completion can be assessed.
- **Monitoring Project Expenditure (Cost).** At Gemini we generally don't charge actual hours to identify project labor cost but in some cases new or additional project costs (e.g., new infrastructure or contracted external developers) is defined up front.
- **Monitoring Project Scope**: It is not unusual to change to the scope of projects but these must be tracked and approved/rejected using change control procedures. Part of this includes system requirements traceability and PI auto-notifications whenever changes are made in project resources, requirements, or schedules.
- **Quality Assurance**: We will review and fix quality problems encountered during development iterations and testing phases of each project. Action items from this testing phase will also be monitored, with special attention paid to overdue closure on pending action items.

Software Testing

We are following fairly standard software testing practices. New software projects are developed alongside a complete suite of unit tests that verify the functionality of individual parts of the system. Integration tests checking the combination of larger modules are included as well.

For example, shown below are two test methods for parsing Gemini semesters that handle simple checks for correct parsing which consist of a four digit year, an optional dash, and a letter A or B.

```
@Test def testParseGood() {
  assertEquals(new Semester(2007, B), Semester.parse("2007B"))
  assertEquals(new Semester(2010, A), Semester.parse("2010-A"))
  assertEquals(new Semester(   0, A), Semester.parse("0000A"))
}

@Test def testParseBad() {
  val bad = List("207B", "20007B", "2007", "-2007B", "2007C", "2007--A")
  bad.foreach {
    b => try {
      Semester.parse(b)
      fail
    } catch {
      case ex: ParseException => // ok
    }
  }
}
```

For the transition projects we have adopted the industry standard “Maven” project management/build system (see Appendix). It automatically executes the full test suite on every build by default. The two test cases above are included in “SemesterTest” as shown below and executed every time the project is built. Any change that breaks the parsing will result in failed test cases and stop the build until it is fixed.

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running edu.gemini.qservice.skycalc.RaBinSizeTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.203 sec
Running edu.gemini.qservice.skycalc.DecBinSizeTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.01 sec
Running edu.gemini.qservice.skycalc.SemesterTest
Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.153 sec
Running edu.gemini.qservice.skycalc.HoursTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec

Results :

Tests run: 20, Failures: 0, Errors: 0, Skipped: 0
```

In addition, we are deploying a continuous integration system that automates a checkout of the updates from our revision control system on each commit by any developer, building and executing the test suite, informing developers when test cases have failed and reporting the level of coverage provided by the test cases.

End-users are also involved in testing, both during the development of the software and before deploying changes. In particular, for the transition projects we are using an iterative development process in which we incorporate testing and feedback from end-users throughout the course of the development cycle.

Number of Builds Planned

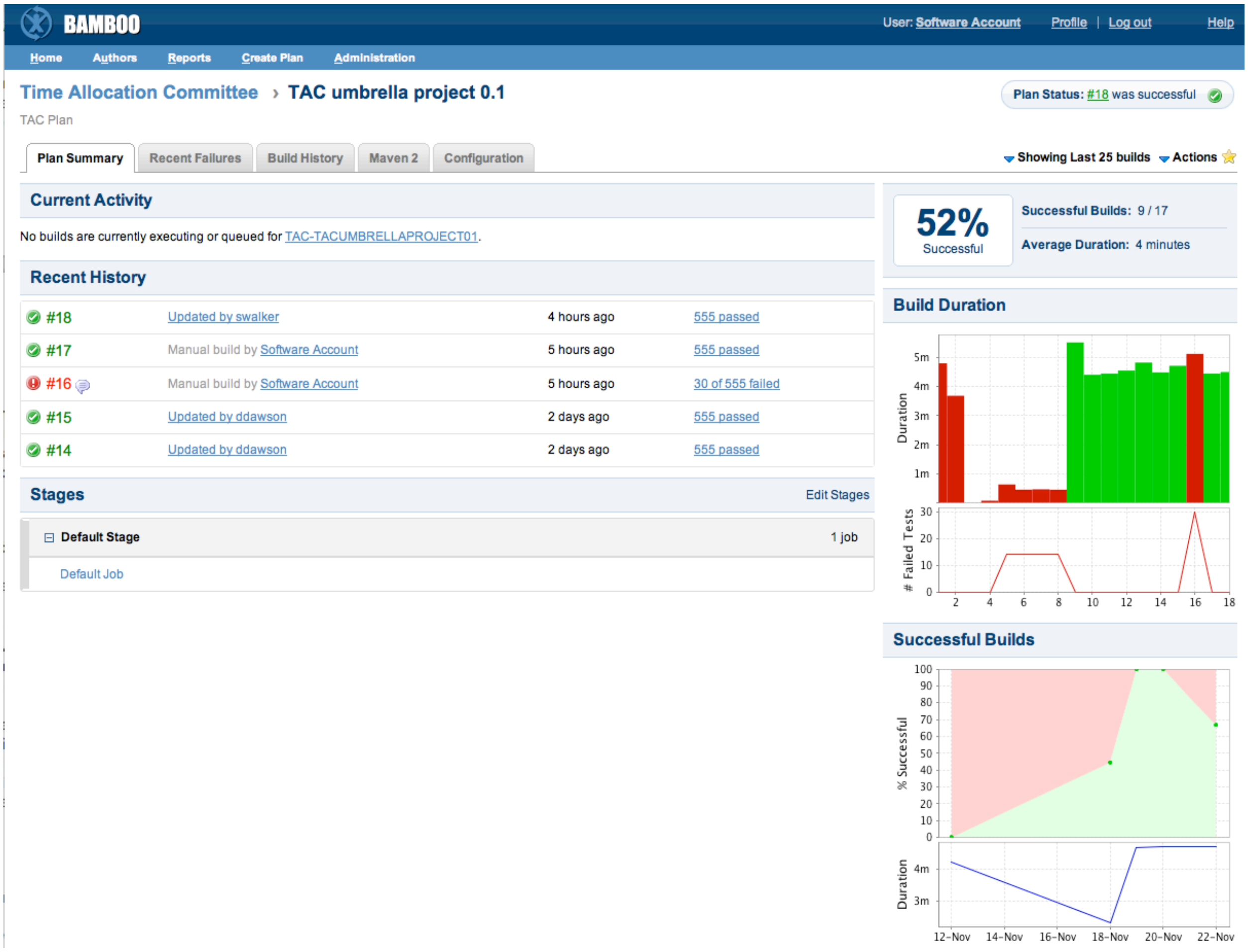
We perform builds on a designated build machine to ensure that official builds of our software are not inadvertently contaminated with local, uncommitted changes made by any given developer. At deployment time, we use the clean build taken from the build machine.

Builds are automated on the build machine using a continuous integration server (see Appendix). The continuous integration server is configured to perform a build whenever a change to the source code is detected in the revision control system. For this reason, the actual number of builds is not set or known in advance.

On the other hand, not every build will generate a new release of the software. As mentioned previously, we are adopting an iterative development style in which the development cycle will be broken into multiple stages on the order of 3-6 weeks per stage. The exact time is somewhat flexible and will depend upon the features to be incorporated into the upcoming test release. Each test release provides an opportunity for user testing and feedback and helps ensure that we remain on track towards delivering a solution that meets the user's needs.

Process for Tracking and Fixing Bugs

There are currently two mechanisms for collecting and tracking software bugs. Bugs in operational software are entered into a Remedy Fault Reporting System (FRS). Results of investigation and fixes are recorded in the FRS. Serious bugs that cause more than 30 minutes of time-loss are evaluated by a team of engineers and scientists to determine if an immediate fix is required. Bugs are also recorded in our general science software task list in Project Insight. This list can include bugs in operational software that have not caused problems during observing, issues and suggestions submitted by NGO staff and users via the HelpDesk, and any other problems reported to the lead and deputy Sci-Ops development scientists. If any bug is determined to be important enough then the science group can request that it be fixed at any time. We try to limit these "immediate" fixes to those that are high priority or time critical. As part of our release cycle the bugs in the FRS are reviewed and prioritized by our two lead System Support Associates. These are then merged and prioritized with the bugs and tasks from the other list by the Sci- Ops scientists. The work to fix these bugs is then scheduled in PI and tracked using the PI "issues tracker". Bugs that originated in the FRS must also be closed out in that system.



Time Allocation Committee › TAC umbrella project 0.1

Plan Status: #18 was successful

TAC Plan

Current Activity

No builds are currently executing or queued for TAC-TACUMBRELLAPROJECT01.

Recent History

#18	Updated by swalker	4 hours ago	555 passed
#17	Manual build by Software Account	5 hours ago	555 passed
#16	Manual build by Software Account	5 hours ago	30 of 555 failed
#15	Updated by ddawson	2 days ago	555 passed
#14	Updated by ddawson	2 days ago	555 passed

Stages — Edit Stages

⊟ Default Stage — 1 job

Default Job

52% Successful

Successful Builds: 9 / 17

Average Duration: 4 minutes

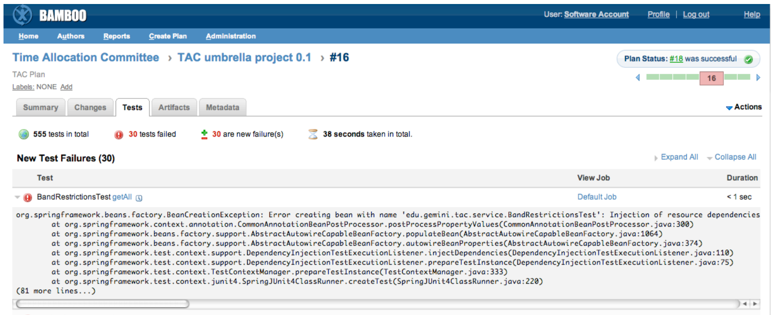
Figure 7 – Our continuous build system's summary view is shown. In this case 18 builds to date in the TAC project appear.

In the future the software group intends to investigate the use of a more specialized software bug tracking system such as bugzilla or JIRA [1.1].

Regression Testing

This is closely related to the answer to how testing is performed in general. During development unit tests are written whenever a new addition or change to the software is made. All new tests must execute and all existing tests must continue to function before changes are ultimately accepted in deployed software.

Final user testing uses "test matrices" (spreadsheets of test procedures) for each software application. Developers and the science group testing team update the test procedures for each release, both groups run though the tests and results are recorded in spreadsheets. If an application fails the tests and the offending feature cannot be fixed by the end of the testing and bug-fix phase, then either the new feature is backed out or the new version of the application is not deployed in the current release cycle. The test matrices are archived in the Document Management Tool (DMT) for the change request approval process that precedes operational deployment and for future reference. The user testing is done manually and is fairly time consuming. We are evaluating software for automating our testing procedures.



Time Allocation Committee › TAC umbrella project 0.1 › #16

TAC Plan

Labels: NONE Add

Plan Status: #18 was successful

555 tests in total 30 tests failed 30 are new failure(s) 38 seconds taken in total.

New Test Failures (30)

Test	View Job	Duration
BandRestrictionsTest getAll	Default Job	< 1 sec

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'edu.gemini.tac.service.BandRestrictionsTest': Injection of resource dependencies
        at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:300)
        at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1064)
        at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.autowireBeanProperties(AbstractAutowireCapableBeanFactory.java:374)
        at org.springframework.test.context.support.DependencyInjectionTestExecutionListener.injectDependencies(DependencyInjectionTestExecutionListener.java:110)
        at org.springframework.test.context.support.DependencyInjectionTestExecutionListener.prepareTestInstance(DependencyInjectionTestExecutionListener.java:75)
        at org.springframework.test.context.TestContextManager.prepareTestInstance(TestContextManager.java:333)
        at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.createTest(SpringJUnit4ClassRunner.java:220)
(81 more lines...)
```

Figure 8 – Gemini’s continuous build server reporting a test failure is shown.

Configuration Management

Requirements documents for projects are under version control using the DocuShare Document Management Tool (DMT). Test plans and results are also archived in DMT. We are currently collecting requirements for a requirements tracking database that will make it easier to record the history of requirements and test results.

Software code changes are tracked and controlled using Subversion. This is a widely used open source solution for software revision control. See the Appendix and “Observatory Software Configuration Management” for more information.

Protecting Development Work From Ongoing Operations Work

Three new hires are already working for Gemini including Florian Nussberger, Larry O’Brien, and Carlos Quiroz to support mainly the new Transition Projects. Also actions are being taken to subcontract work equivalent to an additional ~4 FTEs over the next two years to augment the team on a temporary basis. The contractors are software engineers that have worked for Gemini in the past, giving us confidence in their future performance. We also have a small group of 4 engineers (2 per site) that are almost exclusively assigned to supporting operations. These engineers work on faults reported at night and also work in multidisciplinary teams troubleshooting problems related to other areas of engineering. Developers are only distracted with O&M related assignments when we are in an emergency that requires expert’s participation for a short period of time. The distribution of effort across the software team can be found here.

Queue Optimization Algorithms

The development process for the automated queue planning algorithms will follow the general process outlined above which has been shown to be successful with the initial Queue Planning Tool (QPT), user interface generation, manual drafting, and release cycle process. A group of experienced observers and queue planners will produce the functional and operational requirements. The requirements for the Transition project will be updated using the document “Queue Planning Tool Requirements”. The requirements give specific use cases and define Gemini-specific scheduling needs. The final version will include metrics for defining a successful queue plan. These metrics may include maximizing the observation weights in a plan or maximizing the number of available minutes scheduled. The requirements will be evaluated by the systems engineering group before being passed to software for design and implementation studies. There will be significant interaction between science and engineering groups in order to address all questions and discuss scheduling algorithm options. The algorithms used in an early prototype are described in the document "Queue Planning Software Requirements". Additional prototyping by the software engineering group will determine whether variations on these simple algorithms will meet the requirements or whether more sophisticated scheduling routines (e.g., constrained logic) or commercial solutions are needed. Engineering and science groups will review and agree on a design before implementation starts. Experienced queue schedulers will be involved with testing the new tools as they are developed and phased into operation.

Documentation and Training

We believe that the iterative development cycle we have adopted for the transition projects will help reduce the training burden. In particular, rather than delivering all requested functionality in one single release at the end of the development cycle when changes are expensive, users will have the opportunity to work with the software and provide feedback as it is under development. With each test release, a subset of the features and updates to the final product will be delivered. This will allow the user to focus on learning only the subset of features that are new or changed and to shape the product as it is developed. In other words, by the time the final version is released, many of the users will already be familiar with the product and will have played a role in shaping how it works.

As with existing software products at the Observatory, end user documentation will be created and made accessible online. This documentation is considered to be part of each project's set of deliverables. Examples of these include the Queue Planning Tool (QPT) Reference Manual, Sequence Executor, and "Getting SeqExec Running on Linux".

Skill Mix of Gemini's Software Team

Provided here are resume's for our entire software development team. They represent a blend of "veterans", working for AURA for >20 years, and relative newcomers who have only been recently hired to fill open positions. Collectively they have the required set of skills needed to develop real-time and high-level software for Gemini.

Past Performance Data of Gemini's Software Team

The best example of the development approach we have taken for the Transition software projects comes from the Queue Planning Tool, as mentioned above. In this case, a single developer worked closely with the science staff using an iterative development approach to deliver the initial major versions of this project in roughly a single man-year. This experience, and that gleaned from other projects conducted on a fixed schedule, gives us confidence in our ability to meet the necessary development objectives and schedules.

To demonstrate our track record for delivering time-critical software, listed below are 2 years of semester-by-semester summaries of updated releases of Gemini's Observing Tool (OT). The OT is released each semester to adapt to changing instruments, address bugs identified in the previous semester, and incorporate a progression of upgrades. This list of release notes gives an indication of Gemini’s ability to maintain and generate fairly sophisticated software on a fixed schedule basis.

2010B

- Modified instrument components and iterators
    - GMOS-N instrument component has the option to select either E2V (current) or Hamamatsu (planned upgrade) detectors
    - GMOS gain/read mode selection has been combined in the static components
    - Updates to the GMOS translation stage and ROI tabs
    - Add new Y and Z filters to the GMOS-N filter list
- User interaction and interface features
    - Magnitudes for targets are now entered as value:bandpass pairs instead of in a simple text box. Magnitudes for guide stars are parsed from guide star queries where possible.
    - Added UCAC3 guide star catalog query
    - Updates to the OT library function
- Bug fixes
    - Browser queries saved correctly
    - Improve the calculation of GMOS nod & shuffle detector offset based on the CCD Y binning

2010A

- Modified instrument components and iterators
    - Michelle component - correct behavior of the static component and Position Editor visualization for medN and echelle modes
    - GMOS component - added a Phase 2 check to ensure that imaging observations are binned 1x1, 2x2, or 4x4; pre-imaging should be 1x1 or 2x2
    - Updated telescope offsetting overheads for all instruments
    - Updated overhead calculations for NIFS, Phoenix, and NICI
    - Flamingos-2 component updated to support ongoing commissioning
    - GSAOI and Canopus guide probe visualization added to the Position Editor in preparation for commissioning
- User interaction and interface features
    - It is now possible to deactivate all guide stars for a given probe from both the target component and the Position Editor.
    - Guide probe states for different offset positions are remembered when all stars for a probe are deactivated and are returned to the previous states when re-activated. Note that the states are lost if all the guide stars are deleted from the target component.
- Bug fixes
    - Fix setting of Michelle observing wavelengths in offset iterators
    - Remove restriction on the number of entries displayed in the timing window table

2009B

- Modified instrument components and iterators
  - GMOS components
    - The damaged B600_5303 grating in GMOS-N has been replaced with B600_5307
    - Narrow band filters [SII], [OIII], [OIII]c, HeII and HeIIc have been added to the GMOS-N filter list
    - Narrow band filters HeII and HeIIc have been added to the GMOS-S filter list
  - NICI component - NDR selection removed and ISS port option added
  - Flamingos 2 component updated to support commissioning in 09B
  - GSAOI - basic component added in anticipation of commissioning
  - Added new controls for Michelle engineering
  - Updated Flamingos-2 observing wavelengths
- User interaction and user interface features
  - New target component features:
    - Available guide star types are context sensitive
    - Guide stars that are to be used and viewed in the Position Editor are selected in the target component rather than in the offset iterators
    - Guide star types for GSAOI and Canopus have been added
  - The offset iterator was modified to reflect the changes in the target component
  - Observations using "obsolete" instrument components (e.g. filters, gratings, etc that are no longer used) can now be searched for in the OT browser. By default these components are included in the component lists and they are indicated with an '*'. If the user does not want these components to be visible, then the option can be deselected using the Preferences option in the browser's File menu
  - Overheads for NIRI filter changes have been updated
  - Standard TOO triggers will no longer be given a timing window by default (it had been one week). If a timing window is desired then it can be added explicitly to the triggering observation.
  - For targets not defined in the Phase I information, the observing conditions selected in the conditions component will be compared with the default conditions in the Phase I information.
- Bugs fixed
  - Null pointer exception when working with observations without a sequence component.
  - The contents of an original note changes whenever one copy is edited. The copied notes are now independent again.
  - Michelle central spectroscopic wavelengths will not unexpectedly revert to the default values.
  - If all OIWFS guide stars are removed from the target component of a GMOS nod & shuffle observation then the OIWFS state is set to "park" in both nod positions in the nod&shuffle tab. These will remain at "park" even after a new guide star is added, you must go to the tab to set them to "guide" or "freeze". This is necessary now to avoid a problem in which the program will not export and may get lost from the database. This behavior will be improved in future releases.

2009A

- Modified instrument components and iterators
  - GeMS/GSAOI components added
  - The default wavelength for the Michelle lowN mode is 9.5um in order to move the spectra away from a currently unusable detector channel.
  - GNIRS has been moved to the GN instrument list to support commissioning at GN in 2009.
- User interaction and user interface features
  - Thesis and rollover programs are identified in the main program component.
  - A note has been added to the Phase II skeleton giving information and advice on defining the Phase II observations.
  - Various bug fixes related to drag/drop and storing changes
  - Coordinates from NED/Simbad queries are set to J2000 system
  - Improved communication with the GSA to avoid transfer bottlenecks and maintain rapid ingestion times.
  - Audible alarm for rToO triggers (on-site)
  - PIs can edit elevation/airmass constraints

Appendix

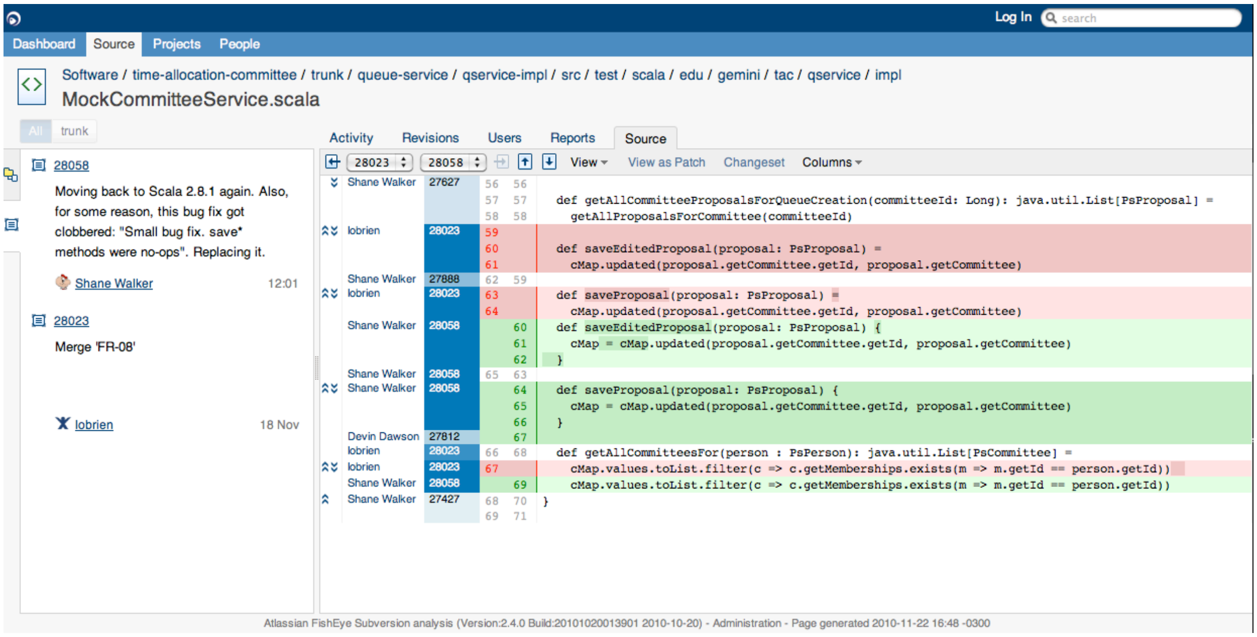
Gemini's Software Development Tool Chain

This section documents the tools we have chosen to support the development of the Transition software projects.

Software Revision Control System

We use Apache Subversion (SVN - http://subversion.apache.org) for software revision control. Subversion enjoys wide support in the software industry and is free and open source. In general revision control systems make it possible to track changes in software over time, share code and updates across the entire development team, and ensure that the exact configuration of particular releases can be recreated at will. Subversion is a standard choice for revision control and most development tools (integrated development environments, continuous integration systems, etc.) come with built-in subversion support.

To fully exploit Subversion, we use a web-based repository browser from Atlassian called “FishEye”. This application (http://www.atlassian.com/software/fisheye) provides “an efficient, consistent way to view change sets, revisions, branches, tags, diffs, annotations and much more from any Web browser”.



Software / time-allocation-committee / trunk / queue-service / qservice-impl / src / test / scala / edu / gemini / tac / qservice / impl

MockCommitteeService.scala

28058

Moving back to Scala 2.8.1 again. Also, for some reason, this bug fix got clobbered: "Small bug fix. save* methods were no-ops". Replacing it.

Shane Walker 12:01

28023

Merge 'FR-08'

lobrien 18 Nov

```

  def getAllCommitteeProposalsForQueueCreation(committeeId: Long): java.util.List[PsProposal] =
    getAllProposalsForCommittee(committeeId)

  def saveEditedProposal(proposal: PsProposal) =
    cMap.updated(proposal.getCommittee.getId, proposal.getCommittee)

  def saveProposal(proposal: PsProposal) =
    cMap.updated(proposal.getCommittee.getId, proposal.getCommittee)
  def saveEditedProposal(proposal: PsProposal) {
    cMap = cMap.updated(proposal.getCommittee.getId, proposal.getCommittee)
  }

  def saveProposal(proposal: PsProposal) {
    cMap = cMap.updated(proposal.getCommittee.getId, proposal.getCommittee)
  }

  def getAllCommitteesFor(person : PsPerson): java.util.List[PsCommittee] =
    cMap.values.toList.filter(c => c.getMemberships.exists(m => m.getId == person.getId))
    cMap.values.toList.filter(c => c.getMemberships.exists(m => m.getId == person.getId))
}

```

Atlassian FishEye Subversion analysis (Version:2.4.0 Build:20101020013901 2010-10-20) - Administration - Page generated 2010-11-22 16:48 -0300

Figure 9 – A Fisheye change set view is shown.

Project Management and Build Automation

Apache Maven (http://maven.apache.org) has been selected for building and managing software projects. Maven is a widely used solution for documenting project dependencies and automating the build process. Using Maven, a software project is described in a collection of standard "project object model" files. Software developers familiar with Maven can move from one project to another with minimal effort because it standardizes both the definition of dependencies between modules that make up a software project and the collection of products that are generated. Like Subversion most development tools come with built-in support for Maven, obviating the need to configure build information in multiple tools.

By default, Maven builds and executes all test code each time it compiles a change to the source code. This helps ensure that bugs are not unexpectedly introduced to working code. Maven also automates test-case code coverage analysis (using a plugin such as Atlassian Clover - http://www.atlassian.com/software/clover), which helps to guarantee that source code has been adequately tested. In the reports that it creates, sections of code that have not been exercised by test cases are highlighted allowing the developer to identify where additional test code is needed.

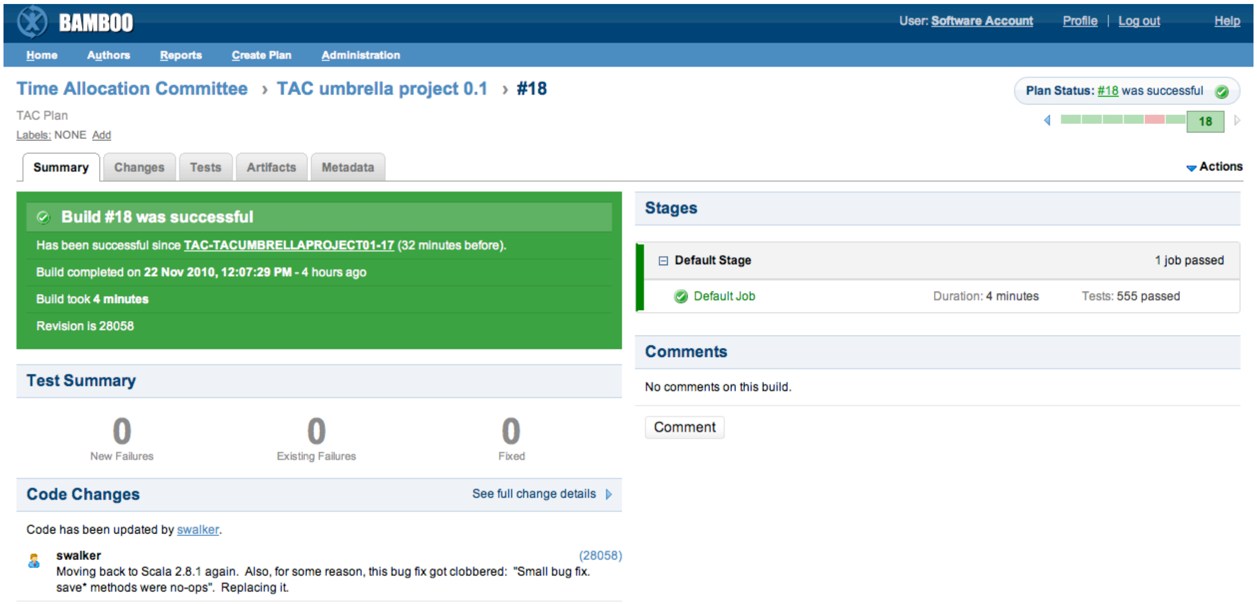
Maven is configured to find both in house and third party software library dependencies in a shared software repository. We use the "Artifactory" (http://www.jfrog.org) repository manager to cache dependencies and support the deployment process.

Continuous Integration

We have a designated build machine that contains the official latest build of each project. When we deploy a new release, the build is taken from the build machine to ensure that it is not contaminated with uncommitted changes that any developer might have made while developing the software.

Builds are automated on the build machine using the Bamboo [1.2] continuous integration server. Bamboo is configured to automatically download changes that have been submitted to our revision control system and execute a build and its associated test cases. When there is a build or test-case failure, developers may be notified by email, RSS or instant messenger.

In the screen shot shown in Figure 10, the code change that prompted the build is displayed along with the successful test results.



Time Allocation Committee › TAC umbrella project 0.1 › #18

TAC Plan

Labels: NONE Add

Plan Status: #18 was successful

Build #18 was successful

Has been successful since TAC-TACUMBRELLAPROJECT01-17 (32 minutes before).

Build completed on 22 Nov 2010, 12:07:29 PM - 4 hours ago

Build took 4 minutes

Revision is 28058

Stages

Default Stage		1 job passed
Default Job	Duration: 4 minutes	Tests: 555 passed

Comments

No comments on this build.

Test Summary

New Failures	Existing Failures	Fixed
0	0	0

Code Changes

See full change details

Code has been updated by swalker.

swalker (28058)
Moving back to Scala 2.8.1 again. Also, for some reason, this bug fix got clobbered: "Small bug fix. save* methods were no-ops". Replacing it.

Figure 10 – Continuous integration server build details are shown.

Deployment

The deployment process to be used with the Transition software projects is an area that is still under investigation. The Transition project applications are being built as collections of modules that are packaged into applications running on the OSGi service platform (http://www.osgi.org). OSGi encourages a development style that focuses on assembling loosely coupled modules. It has emerged as the industry standard component model for JVM-based applications. One benefit of using OSGi is that it provides a number of options to help simplify application management and deployment. For example, applications themselves may provide a remote management interface that allows individual modules to be started, stopped, and updated remotely. Third-party tools like Apache ACE (http://incubator.apache.org/ace) and Equinox p2 (http://wiki.eclipse.org/Equinox/p2) support remotely managed software distribution frameworks that allow the user to configure and distribute updates to target applications.