Managing an Internationally Distributed Software Project

Richard J. McGonegal and Stephen B. Wampler

The Gemini 8-m Telescopes Project
950 N.Cherry Ave., Tucson AZ 85726

ABSTRACT

The Gemini 8-m Telescopes Project is faced with the challenges of demanding scientific requirements, a fixed budget, and an aggressive schedule. In addition Gemini is an international project and the majority of the work itself is being done by groups within the partner countries. In order to meet the requirements on budget and on schedule Gemini has had to adopt a development methodology tailored to its specific circumstances. This paper will detail the engineering practices put in place to ensure quality, software and hardware standards adopted, and the management techniques and tools use to promote success. In addition this paper will describe some of the lessons learned and, most importantly, what the author would change if this were repeated.

Keywords: telescopes, development methodology, distributed management, software project

INTRODUCTION

The Gemini Project is an international partnership to build two 8-meter telescopes, one on Mauna Kea, Hawaii, and one on Cerro Pachon, Chile . The telescopes and auxiliary instrumentation will be international facilities open to the scientific communities of the member countries.The international partnership is made up of the United States, the United Kingdom, Canada, Chile, Argentina, and Brazil. The telescopes will be high performance, 8-meter aperture optical/infrared telescopes and have a planned completion date of 1998-2000.

CHALLENGES

Scientific Requirements

The goal of the telescopes is to exploit the best natural observing conditions and to undertake a broad range of astronomical research programs within the national communities of the partner countries. In order to reach this goal Gemini has set demanding requirements in terms of image quality, tracking, pointing, and availability (Table 1)

TABLE 1.

Specification	Requirement	Description
Image Quality	0.1 arcsec	increase in 50% encircled energy diameter
Tracking	0.044 arcsec	rms jitter in line of sight
Pointing	3.0 arcsec	in service pointing
Availability	98%	time collecting science photons

In addition Gemini has requirements to support a number of different observing modes.

- *Classical Observing* - user manually sequences the different telescope subsystems to acquire data
- *Preplanned Observing* - user plans detailed use of facility in advance and submits this to facility
- *Queue Scheduling* - parts of one observer's program are interspersed with those of others in order to make more efficient use of facility
- *Flexible Scheduling* - programs to be executed are selected depending on environmental conditions in order to make efficient use of existing conditions
- *Service Observing* - program is executed by a member of observatory staff - not the Principal Investigator

These different modes create a requirement for a programmable upper layer to the software - which Gemini calls the Observatory Control System. This system acts to synchronize and sequence the actions of the different subsystems which make up the observatory system.

The support for these different observing modes has been a very real factor in the planning for Gemini - the Gemini Science Committee has recommended that Gemini set as a goal that 50% of all observing be done in a “non Classical” manner in order to take advantage of the unique characteristics of the site.

Aggressive Schedule

The Software project started in 1992 and has the following milestones:

Sep’93 - Critical Design Review of Software System Design
Apr’95 - Delivery of User Interface Prototype
May’96 - Delivery of Alpha 1 Prototype
Aug’96 - Delivery of Alpha 2 Prototype
Jan’97 - Delivery of Alpha 3 Prototype
Aug’97 - Delivery of Beta System
Jan’98 - First Released Version of Software
Dec’98 - First Light; Mauna Kea
Apr’98 - Acceptance of Mauna Kea Software System
Jun’00 - First Light; Cerro Pachon
Oct’01 - Acceptance of Cerro Pachon Software System

One of the major differences between a new telescope project and adding a subsystem to an existing telescope is that everything is being done in parallel. This means that, quite often, it is only the interfaces which can be defined, the software on either side of the interface does not exist yet.

Fixed Budget

Gemini is a fixed price project - both of the telescope systems must be delivered within a price envelope of $176 million, including inflation. This means that the specification, the schedule, and the budget are all tightly interlinked and constrained. What this means in practice is that schedule is just as important as budget.

As Gemini must pay for labor from the same pool of funds from which it pays for capital expenses purchasing commercial products makes good financial sense as they are, in general, quite inexpensive compared to the effort required to duplicate them and, especially, the effort required to maintain the duplicated software. It also means that it can cost more to be late than it does to hire more staff and be early - especially when inflation and currency exchange fluctuations are taken in to account.

INTERNATIONAL COLLABORATION

The Gemini system is being produced as a combination of internationally bid contracts and allocated work packages. In general the allocated work packages represent work in which the partners possess intellectual interest and existing skills. In the area of software/controls, the work is being done currently at five sites; Tucson, Arizona; Victoria, Canada; Edinburgh, Scotland; Chilton, England; and Cambridge, England. In addition the software/controls for the Scientific Instruments will be done at other, yet to be determined sites.

WHY DO THIS

A valid question is, “Why not just form a central project team and do the software in a conventional fashion at a single site?”. At first look this would seem to be sensible way to proceed and follows conventional wisdom.

Just as the telescopes are designed to exploit the best that the sites can offer the project structure is designed to exploit the best skills that the individual partners can bring to bear on the challenges represented by Gemini. The Astronomy community has a number of individuals with unique skills, expertise, and experience. The project benefits immensely from being able to tap these resources - indeed, access to these resources is required in order to build a telescope which satisfies the specifications, budget, and schedule requirements of Gemini.

The focus of the Gemini Project will migrate from Tucson (1992-1997), to Hawaii (1998-2000), and then to Chile (1999-2001) as will the project staff . Also, Gemini is a fixed length project with no guaranteed project positions past the year 2001.

Under these conditions it would be extremely difficult to convince a large number of these highly skilled individuals in the partner countries to give up their current positions to move to the Gemini Project. Indeed, it may be better for the long term health of the community if these people remain with their home institutions.

The Gemini solution to this in the area of software/controls was to form a distributed group. In order to make this successful it was necessary for Gemini to evolve a methodology for how to do this.

THE GEMINI DEVELOPMENT METHODOLOGY

Work Breakdown Structure

Gemini has followed two guiding principles, not always rigorously, in how the system was broken done into its component subsystems:

- try and partition the system to minimize the subsystem interfaces, and
- try to locate the software/controls at the same site which is building the mechanism.

The partitioning of the subsystems is complicated by the fact that it is a tightly coupled system. Unlike more conventional telescopes everything, including the orientation of the wind flow relative to the telescope, can affect the delivered image quality. This means that there exists a large degree of interconnection between the subsystems.

Locating the software/controls work at the mechanism fabrication site is not possible for some of the items such as the telescope structure and enclosure. In general the vendors with the skills to do this type of fabrication do not have ready access to the skills to do the software/controls.

TABLE 2.

Work Package Title	Organization	Site
Observatory Control	NOAO/Gemini	Tucson
Data Handling	DAO	Victoria
Communications	CONICYT + JACH	Chile + Hawaii
Core Instrument Controller	ROE	Edinburgh
Standard Control System	RGO	Cambridge
Telescope Control	RGO / DRAL	Cambridge + Oxford
Mount Control	RGO	Cambridge
Primary Mirror Control	RGO	Cambridge
Secondary Mirror Control	ROE	Edinburgh
Enclosure Control	DAO	Victoria

Engineering Practices

Gemini decided early on to follow three guiding principles:

- People who write software for big computer projects like to describe themselves as engineers. To deserve the name, they must behave as other engineers do.
- Successful software project require two things:
  —customers who can explain what sort of job needs doing
  —engineers who can deliver s system that will do the job on time and at a price that makes doing the job worthwhile
- When buying technology, it always pays to be second.

Gemini has adopted parts of the IEEE and ESA standards covering software development. There are used as guidelines for software developers following the Gemini Software Development model.

Gemini depends on a number of key elements in its model in order to keep a distributed group going. These are:

- Software Management Plan
- Overall System Design
- Formal Development Process
- Formal Review Process
- Established Standards

Each of these will be discussed in detail below.

Software Management Plan

An additional key item to distributed management is an overall plan. The project has found that the IEEE Standard 1058.1, *Software Project Management Plans*, is a useful mechanism for putting together this plan.

Overall System Design

The development of the Overall System Design was carried out by the Gemini Project Office using engineers drawn from the partner countries. This design went through a formal review process consisting of system design, preliminary design, and critical design where the review committee was made up of members of the partner countries and other large telescope projects.

The result of the System Design process was a formal Software Description Document[1] (SDD) which laid out the conceptual design of all of the subsystems as well as Interface Control Documents for the inter-subsystem and intra-subsystem interfaces in common.. The SDD followed appropriate IEEE and ESA standards.

Formal Development Process

A Work Package is defined by a formal Work Scope document which is signed by all parties concerned. This Work Scope describes the details of each phase of the development process, the deliverables from each phase, and the details of budget, schedule, and payment.

The individual phases and their deliverables are detailed below. The first phase, Work Package Development, takes place as part of the development of the Work Scope.

Work Package Development

The output of this phase of the development consists of a Work Scope which contains the following deliverables:

- Work Breakdown Structure
- Schedule
- Cost Breakdown by WBS element
- Capital Costs
- Travel
- Key and Supporting Personnel

System Design

The output of this phase consists of the requirements and a conceptual design of the system.

- Package Requirements Specification.
- Environmental and Behavioral Models.
- Plans for Design Process
- Plans for Simulations / Prototypes / Trade Studies
- Documentation Plan
- Proposed Equipment

Preliminary Design

The output of this phase is a

- Preliminary Package Software Design Description
- Results of Simulations / Prototypes / Trade Studies
- Initial Work Product documents

Prototype

For most of the software packages we do not produce a formal critical design. A critical design review would review all of the software design in advance of coding - the intent being to avoid significant rework if the design does not meet the requirements. Our belief is that the only reason to do significant levels of detailed software design is if the cost of producing the software, and hence the cost of redoing the software, is significant relative to the cost of doing the detailed design. With the tools which Gemini uses for development we believe that a series of rapid prototypes is a much better investment than reviewing a formal detailed design.

- Prototype Control System
- Package Software Design Description
- Update of Simulations / Prototypes / Trade Studies
- Preliminary Work Product Documents

Alpha Version

Alpha Control System
- provide the user and programmatic interface to the work package as seen by higher level users and systems
- respond correctly to these higher level inputs
- respond correctly to peer systems (such as the Data Handling, Instrument Control, and Interlock system).

Preliminary Package Test Procedure

Beta Version

Beta Control System
no Catastrophic Bugs
may contain Severe Bugs with or without Work Arounds.
may contain Cosmetic Bugs
Package Test Procedure
Preliminary Acceptance Test Plan

Final Version

Acceptance Test Plan
Final Control System
no Catastrophic Bugs
may contain Severe Bugs if Work Arounds exist.
no Cosmetic Bugs

Acceptance Testing

Acceptance Test Report

Specification Control System

hardware and software
documentation

Review Process

Each of the major phases of the Work Package have an associated review (Table 3). The goal of the reviews is not to have a large amount of presentation material and spend long periods of time listening to presenters. The current review format depends on whether it is a *real* or *virtual* review committee. The difference is that r*eal* review committees meet in person a single time with the person responsible for the work being reviewed while only a subset of a *virtual* committee ever meets in person.

TABLE 3.

Major Phase	Associated Design Review
System Design	System Design Review
Preliminary Design	Preliminary Design Review
Prototype / Critical Design	Prototype / Critical Design Review
Alpha Version	Alpha Review

TABLE 3.

Beta Version	Beta Review
Final Version	Acceptance Specification Review
Acceptance Testing	Acceptance Testing Review

The process that virtual review committees follow is:

- receive the documents ahead of time via email or ftp,
- send in comments ahead of time via email,
- reviewee prepares responses to comments
- subset of committee may/may not meet with reviewee to go over responses
- comments/responses sent via email to committee for comment
- final report issued to all concerned

Real review committees are very valuable during the early phases of a work package, especially for areas which have poorly defined requirements, difficult interfaces, or demanding functionality. *Virtual* review committees make much better use of everyone's time for areas which are well defined and reasonably non-controversial.

Software and Hardware Standards

Gemini has followed a policy of adopting or adapting to commercial and community standards wherever possible. The philosophy behind this is to spend the time building the application, not the infrastructure for the application. Towards this end the following are considered Gemini standards.

TABLE 4.

Area	Standard
Host Workstation	Sun
Host OS	Solaris
GUI Builder	not decided
Windows	X11
Command Language	TCL
GUI	Tk
Visualization	PvWave
Bulk Data transfer	Self Defining Structures (SDS)
Observing Database	not decided
Archival Database	Sybase
Data Processing	IDL
Data Reduction	IRAF
Version Control	CVS
Trouble Reporting	TkGnats
Real Time Target	68040
Real Time Target OS	VxWorks
Real Time Database	EPICS
RT Database Programming	CapFast

One of the truisms of software development is that it is often the infrastructure rather than the application which is the more interesting project. In a number of areas Gemini encountered substantial resistance to the adoption of standards, especially when the standards were used to displace existing infrastructure developed by that site.

Gemini's position is that it is only by adopting standards that Gemini can stay ahead of the user driven application demand - after all, users want applications. Users also want applications that work the way their other applications work, and that have the same bells and whistles. By leveraging our applications off of widely used infrastructure standards Gemini can take advantage of the ongoing development efforts as well as the large programming pool available.

MANAGEMENT TECHNIQUES AND TOOLS

Work Scopes

Once the overall System Design and the Standards are in place it is necessary to have an agreement in place about what work needs to be done, how that work will be done and reviewed, and how much effort and money is to be used to perform the work. In Gemini's system all of the Partner Countries have agreements in place which cover most of the boiler plate of a more traditional contract. The details of each Work Package are contained in a Work Scope developed by the staff responsible for performing the work.

The management of a work package is via a matrix management approach. The individual at the site who is responsible for the work remains an employee of his/her existing organization and reports to and takes direction from his/her normal supervisor. However, in all areas to do with the work package, the individual is responsible to the Gemini Controls manager. This means in practice that no changes to the work package which affect specification, budget, or schedule can be taken without prior agreement of Gemini .

National Project Offices

Each of the Partners have a National Gemini Project Office which is generally staffed with a Project Manager, Project Engineer, and Project Scientist. The Project Offices perform a very valuable function of being the major point of contact between the national communities and the Gemini Project. They also act to form a national consensus on issues that will be discussed by Gemini.

The National Project Manager is one of the signatories, in addition to the Work Package Manager, who signs the Work Scope.

LESSONS LEARNED

A number of lessons have been learned over the preceding three years and there will certainly be new ones. Some of the more important ones are:

Ownership

It is extremely important that the groups responsible for the work have a real sense of ownership of the work they are doing. This can only be developed by involving the groups in as much of the process as is possible. In the Gemini case this required starting a System Design phase with members of the partner countries involved, having members of the partner countries sit on the review committees, and having the work package responsibles develop the work breakdown structure, schedule and costing on their own.

Personal Contact

It is very much easier to do productive work with people whom you know well on a personal as well as professional level. It is imperative, especially during the startup of the project, to have regular face to face meetings. Gemini had someone from the controls group at the remote sites for one week per month for the initial year.

Delegation

You have to let go - once you involve outside groups you have to be willing to give them the freedom to make design decisions and choices. This means that the system you are building will not be done the way you would have done it. If you manage carefully enough though, it will probably be done better than you were capable of on your own.

Moving Target

The requirements have to change - although it would be easy to insist on having all the requirements iron clad at the start this is unrealistic. To be successful the project must meet the user's expectations, of which the requirements are only an attempt at expressing. The waterfall model of software/control development is not appropriate in this environment.

Communications

The most important aspect of keeping a distributed group on schedule would appear to be good communications. Gemini uses a mix of the following communication methods.

TABLE 5.

Method	**Pros**	**Cons**
email	effective for frequent in writing instructions and communications	not always read, still gets lost or delayed more than would like real problem with standards for attaching binary files between Windows and Unix
email exploders only way to make sure that everyone sees everything	very good for making sure no one is left out of the loop on key topics	volume of traffic means some people will arbitrarily delete email if title has no hook
site visits	only way to make personal contact must precede all other forms of communication required for testing and some reviews	very expensive ($500/day) for foreign sites hard on schedules, staff and families
ftp mirrors	keep all documents at all sites	file naming conventions differ between Windows and Unix
teleconference	good for weekly meetings	quality of voice and audio levels make multiple person teleconferences difficult
video conference	very good for small meetings with good agenda allows access to body language	end point equipment expensive compression/decompression delays in audio difficult some of partners have very expensive charges still not quite “plug and play”
fax	sometimes only way to get document there you know it arrived	quality / missing pages differences between letter and A4 sizes
FedEx/UPS	still are people with no net connection	expensive
Postal Mail	still best way to guarantee that it was received	slow and expensive for large documents

THINGS TO CHANGE

The one area which the author would change significantly in a similar project would be to involve more of the partner countries principal software/controls staff in a central team for the overall System Design phase. The best option would be to get one or two staff from each of the institutions to spend a period of 6 months at a central site to do the system design as a coordinated team. This would have made things much easier at the point in time at which the subsystems are sent out as work packages as there would already be staff on-site with in-depth experience with the design. Although this might seem to contradict the usefulness of distributed management there is a learning curve associated with bringing the group up to sped on a design - perhaps improvements in video conferencing will remove this problem in the future.

In Gemini’s case any such effort would have been difficult due to the span of time over which the partners joined the project - some 3 years. This meant that it was difficult to get commitments from all the potential partners to send staff to the central site.

Although it might seem easy to insist on all the partners being signed up before starting the requirements process, and then to have all the requirements agreed to before proceeding to do a system design, this is not the way projects like this happen. It is necessary that large projects proceed as much in parallel as possible in order to shorten the time scale to delivery. I believe that this is just an extension of the concept rapid prototyping to the level of including all the phases of a large project.

CONCLUSIONS

The Gemini Project's Software Development Methodology effectively meets the challenges posed by Gemini's distributed nature. Although this methodology is more challenging and more expensive than centralized management this does not necessarily translate into more difficulty and more expense for the project as a whole. Indeed, it is the author's experience that distributed management allows one to access much more qualified people to work on the project than a hiring process would result in. This is not to say that the project cannot recruit and hire highly skilled staff - just that the pool is much larger if one is willing to do it in a distributed fashion. This results in cost savings, schedule shortening, and improved system performance relative to a centralized staff - however such improvements are difficult to quantify up front. Our current metrics show that staff, in general, are meeting their milestones with only 70-80% of the effort they estimated.

A methodology makes it is possible to successfully manage a large, distributed software/controls development effort - although Gemini has a few more years to go before we can lay claim to ultimate success. However the Gemini software/controls task is currently on budget, to schedule, and has yet to miss a specification.

A distributed project is the only way large science projects will be done in the future so we need to find methodologies that work and not be afraid to try new models. The advent of high bandwidth communications at reasonable rates will bring more and more of these types of group interactions into the forefront. The ability to quickly form and disperse inter-organizational project teams in a distributed fashion will be key to keeping costs and schedules in check while at the same time meeting the challenging specifications with low risk solutions.

ACKNOWLEDGMENTS

The author would like to thank all of the Gemini work package responsibles for their effort and forbearance as Gemini worked towards a model which would work.

REFERENCES

[1] S.Wampler, K.Gillies, S.Beard, A.Johnson, R.Laing, C.Mayer, P.McGehee, R.McGonegal, P.Wallace, Software Design Description, Gemini Project Doc. No. SPE-C-G0037