

AIPS++ Technical Review

National Radio Astronomy Observatory
Socorro, NM
5-6 March 2003

REVIEW PANEL INTERIM REPORT

Roger Brissenden (CfA)
Hilton Lewis (Keck), Chair
Andrew Lumsdaine (U. Indiana)
Dave McConnell (ATNF)
Steve Scott (OVRO)
David Silva (ESO)
Doug Tody (NRAO)
Rick White (STSCI)

Table of Contents

Executive Summary	3
Introduction.....	8
Review Process	8
Major Recommendations	9
Appendix A: Charge to the Review Panel	16
Appendix B: Specific Questions to the Panel Posed by the Executive Committee.....	17

Executive Summary

A Technical Review for AIPS++ was held at the National Radio Astronomy Observatory in Socorro on March 5-6, 2003. The eight members of the review panel were: Roger Brissenden (CfA), Hilton Lewis (Keck), Chair, Andrew Lumsdaine (U. Indiana), Dave McConnell (ATNF), Steve Scott (OVRO), David Silva (ESO), Doug Tody (NRAO) and Rick White (STSCI).

The Panel congratulates the project team on the high degree of preparation for this review. The advance material was comprehensive and accessible, covering the major areas of interest to the Panel. The Panel also greatly appreciates the openness and candor of the project team in discussions at the review. We would also like to thank the project team and the NRAO for their hospitality during the intensive two days of the review.

Our charge was twofold: to review the technical suitability of AIPS++ for current and future radio astronomical data processing and to make specific recommendations on how to make AIPS++ a useful and user-friendly data reduction package. However, any meaningful review of these issues must, of necessity, involve some examination of the sociological environment of the project.

A number of significant technical issues have dogged the AIPS++ program from the earliest days, in particular, incomplete functionality and inadequate performance and robustness. In the past, a number of reviews have recommended changes to the program, but these recommendations have not always been adopted. A number of problems in management and technical development have persisted; however, within the last six months some very significant improvements have been made.

This Panel finds that AIPS++ has the *potential* to become a package suitable for general radio astronomical data processing, for current and future instruments, provided some significant changes are made to the management and development process. If these changes are adopted, there are excellent prospects for AIPS++ meeting the challenges of the next generation of facilities currently under development. We feel strongly, however, that continuing on the current path will not yield this result.

The Panel makes a number of recommendations that we feel are crucial to the future success of AIPS++. These recommendations are summarized below, and are explained in more detail in the *Major Recommendations* section that follows.

- 1. Change the focus of AIPS++ from a system aiming to be a general-purpose radio astronomy package to a system that is developed to meet the strategic goals of the Consortium.**

Until now, the aim of the AIPS++ project has been to develop a general-purpose package for the analysis of radio astronomy data. After 10 years of development, the system is still not yet at an appropriate level of maturity to deliver to the general user community. We believe that the major reason for the lack of delivery

has been the focus on producing a general-purpose system at the expense of producing software with specific science goals driven by actual radio astronomy analysis use cases and data.

In order to meet the AIPS++ goals, the approach to development must be driven explicitly by the needs of the current and planned major radio astronomy projects. This implies a modified development process consisting of

- science staff identifying use cases and developing requirements (including performance requirements) from the use cases
- development staff flowing the requirements to the software design and developing a build plan with content and delivery tied directly to the project milestones
- a test process that involves science testers using test data based on the use cases
- software acceptance by the relevant science or project lead

This approach is fundamentally different from the current AIPS++ development paradigm.

ALMA is a high-priority project and will be a major driver of the future software development. We see the task of completing sufficient functionality to demonstrate a full set of VLA use cases and a subset of complementary use cases from other Consortium instruments as high priority, in order to complete a significant part of the core functionality required for ALMA.

The support for external non-Consortium users should be de-emphasized until an accepted system can be completed. We recommend taking AIPS++ off-line until sufficient functionality has been demonstrated.

2. Modify the Development Process

In order to deliver software tied to the specific milestones of the Consortium projects, the present software development approach needs to be modified. We suggest the incorporation of a number of developmental steps, detailed later in this report. Although many of these elements are currently present, they are not focused on the delivery milestones derived from the projects.

A critical aspect of developing the AIPS++ build schedule is creating a master schedule tied to the Consortium projects. The AIPS++ master schedule should be consistent with the major project schedules (e.g., GBT, ALMA, EVLA).

Explicitly tying the content and deliveries to a master schedule detailing the needs of the projects may result in the abandonment of the current 6-month routine delivery cycle. Such a delivery cycle is more suited to a system in long-term maintenance rather than development. The recent adoption of mini-deliveries each on 4-6 week timescales is a move in this direction.

3. Strengthen the Project Management and the Project Team

The project team is clearly highly talented, energetic and dedicated. Nevertheless, it is evident that there are a number of weaknesses in the management of the project that must be addressed.

Project management must be strengthened in a number of areas.

- More formal project management techniques should be adopted.
- The Project Manager role is a full-time position requiring a dedicated person; it cannot be shared with other duties and responsibilities. The current practice of sharing the Project Manager role between several individuals, none of whom is fully dedicated to the role, cannot work for a program of this magnitude.
- A clear customer focus must be fostered among the project team.
- Every effort must be made to fill current staff vacancies. The team should be strengthened by the addition of professional software engineers. One of the project staff positions should have the role of software architect, responsible for the overall integrity and consistency of the core classes.
- More effort must be made to utilize full-time FTEs. Fractional FTE allocations should be consolidated wherever possible into full-time staff.

Although this program could benefit from additional personnel, the Panel believes that the current team (utilizing all currently open positions) should first demonstrate their capability to use what resources they have more effectively. We also note that several Consortium partners are currently contributing significantly fewer staff than that required of them, in some cases to the point where their contributions are ineffective. This is a matter that the AIPS++ Executive Committee should resolve.

4. Further Strengthen the Project Scientist Role

The Panel applauds the appointment of a strong and active AIPS++ Project Scientist. The Panel also acknowledges the recent activities of the NRAO AIPS++ User Group (NAUG). The interaction of the NAUG and the project in recent months has been clearly beneficial and demonstrates the need and power of actively involving users in the development process.

The Panel believes the role of the Project Scientist is critical and needs to be expanded in a number of ways.

Analogous to the Project Manager, the Project Scientist should be appointed at the Consortium level and be responsible to the AIPS++ Executive Committee. Individual Consortium sites and projects should appoint individual Site Scientists whose activities would be coordinated by the Project Scientist.

In coordination with the Site Scientists, the Project Scientist should:

- Develop science-based use cases for all major Consortium instruments.
- Based on use cases, derive project science requirements (including performance requirements).
- Identify and appoint Consortium staff scientists as use case testers.
- Organize and prioritize tester feedback for project action.
- Verify that project deliverables fulfill use-case-derived requirements.
- Organize development of high-level User's Manuals ("cookbooks").

In coordination with the Project Manager, the Project Scientist should:

- Provide scientific prioritization for all project activity.
- Assure that general project activity is aligned with scientific requirements.
- Certify and approve for public release any new version of AIPS++.

Technical decisions should continue to lie with the Project Manager. The Project Scientist will generate use-case-based requirements, but it is up to the Project Manager to decide how to achieve those requirements.

5. Short-Term Priorities

Development efforts over the next 12 months should focus on the following:

- Significant improvements in reliability and stability
- Performance improvements approaching the performance of AIPS
- Synthesis imaging capability covering many common use cases of the VLA
- Reduced emphasis on graphical user interfaces

The Panel endorses the current systematic approach to investigation of performance issues by benchmarking of components. It seems likely that continuation with the same program will increase performance to the point where it approaches that of AIPS.

There are defects in the design of the many of the GUIs, some related to the underlying implementation of Glish and others to usability issues. However, script-driven processing is adequate to demonstrate the core functionality of AIPS++ at this stage. A Viewer GUI is essential for this mode of operation, but improving other graphical interfaces should have a lower priority for now.

Future releases must include user "cookbooks" for processing data with validated use cases.

The VLA provides an ideal test bed for AIPS++. The effort to identify the VLA scientific use cases needs to be completed along with their systematic testing. Successful achievement of this opens the door for an AIPS++ VLA pipeline, the

routine operation of which would give much needed credibility to AIPS++, as well as confidence that it could fulfill its Consortium obligations.

6. Proceed With the Proposed Technology Changes

The current architecture is built around Glish, which implements many features central to AIPS++. However, this is a custom system supported by just one developer; it is not easily able to benefit from the many developments in the rest of the software world. In fact, many of its functions might be better performed by standard components developed over the last 10 years.

A new technology has been proposed to change AIPS++ to an industry-standard open system, capable of using open-source software. This approach can address many issues of robustness, performance and usability that plague the current implementation.

The Panel finds that the case for changing the technology to support future development of AIPS++ is compelling. The proposed architectural change to the framework has technical merit and a proof of concept should be pursued at this time, in parallel with current development. However, it is critically important that this *not* be allowed to distract the AIPS++ team from completing the current development.

Since no design was presented, the Panel was unable to estimate the effort and cost required to make the change, nor indeed whether the candidate technology will in fact meet the needs of the Consortium projects. A design review will need to be held at some future time before the upgrade project can proceed.

Introduction

The team should be congratulated for the high degree of preparation for this review. The advance material was comprehensive and accessible, covering the major areas of interest to the Panel. The Panel greatly appreciates the openness and candor of the project team in discussions at the review. We would also like to thank the project team and the NRAO for their hospitality during the intensive two days of the review.

Our charge was twofold: to review the technical suitability of AIPS++ for current and future radio astronomical data processing, and to make specific recommendations on how to make AIPS++ a useful and user-friendly data-reduction package. However, any meaningful review of these issues must, of necessity, involve some examination of the sociological environment of the project.

A number of significant technical issues have dogged the AIPS++ program from the earliest days: incomplete functionality, inadequate performance and robustness, the inability to perform end-to-end data reductions and difficulty with usability and programmability. This Panel notes that there have been a number of major reviews of the AIPS++ program in the past, panels that have provided clear recommendations for change and warnings of problems that will arise if such change was not made. Several of the significant recommendations of the past have not been adopted and problems in management and technical development of the AIPS++ package have persisted. However, we also note that there have been a number of very significant improvements made within the last six months. In particular, the interactions and effectiveness of Steve Myers (Project Scientist) and Joe McMullin (Project Manager) have been particularly important in this regard.

This Panel finds that AIPS++ has the *potential* to become a package suitable for general radio astronomical data processing, for current and future instruments, provided some significant changes are made to the management and development process. If the recommended changes are adopted, there are excellent prospects for AIPS++ meeting the challenges of the next-generation facilities currently under development. We feel strongly however, that continuing on the current path will not yield this result.

The Panel makes a number of recommendations that we feel are crucial to the future success of AIPS++. The charge to the Panel and answers to questions posed by the project team are included in the appendix.

Review Process

The material for the AIPS++ Technical Review was made available on 21 February 2003. A number of email discussions were held prior to the review of 5-6 March 2003. The Panel convened in executive session on March 7 to produce a preliminary report; this was the basis of a verbal briefing to the AIPS++ Executive Committee later that day. An interim version of the report was submitted to the chairman of the AIPS++ Executive Committee, Dr. Fred K.Y Lo, on 17 March 2003.

Major Recommendations

1. Change the focus of AIPS++ from a system aiming to be a general-purpose radio astronomy package to a system that is developed to meet the strategic goals of the Consortium.

Until now, the aim of the AIPS++ project has been to develop a general-purpose package for the analysis of radio astronomy data. The intent has been to provide tools to allow the analysis and processing of data from current and future radio telescopes, including the VLA, GBT, ALMA and EVLA. After 10 years of development, the functionality of the system is approaching that required to analyze a subset of existing VLA science cases, but with performance, usability and some code defect issues. The system is not yet at an appropriate level of maturity to deliver to users.

We believe that the major reason for the lack of delivery has been the focus on producing a general-purpose system at the expense of producing software with specific science goals driven by actual radio astronomy analysis use cases and data.

In order for the Consortium to meet its analysis and data-processing software goals, the approach to developing AIPS++ must be driven explicitly by the needs of the current and planned major radio astronomy projects. This implies a modified development process consisting of

- science staff identifying use cases and developing requirements (including performance requirements) from the use cases
- development staff flowing the requirements to the software design and developing a build plan with content and delivery tied directly to the project milestones
- a test process that involves science testers using test data based on the use cases
- software acceptance by the relevant science or project lead

This approach is fundamentally different from the current AIPS++ development paradigm and implies tight coupling between a master schedule and the AIPS++ software deliveries.

ALMA is a high-priority project and will be a major driver of the future software development. In principle, it is preferable to examine high-priority use cases from all Consortium projects and derive an integrated set of requirements and build content from these. In practice, however, this diffuse focus, coupled with a small and distributed project team, is unlikely to succeed. We propose instead that the task of completing sufficient functionality to demonstrate a full set of VLA use cases be viewed as a high priority, not in order to satisfy users (who are generally using AIPS), but in order to complete a set of core functionality required for ALMA. A subset of complementary use cases of other Consortium instruments

(such as ATCA and BIMA) should also be incorporated in order to match remaining core ALMA requirements. Examples of these requirements are complex spectrometer specification and configuration, mosaicing and linear polarization.

We see the completion of the VLA use cases and a carefully selected subset of complementary use cases from other Consortium instruments as an essential part of the ALMA development.

The support for external non-Consortium users should be de-emphasized until an accepted system can be completed. We recommend taking AIPS++ off-line until sufficient functionality has been demonstrated per use cases. At present, VLA users are generally using AIPS, and the GBT users could be considered as the first target users of the system.

We believe that adopting these recommendations can, in time, lead to the AIPS++ goal of providing a general-purpose radio astronomy package, a package that will provide a general-purpose toolkit, cater for a variety of instruments and provide a good environment for algorithm development.

2. Modify the Development Process

In order to deliver software tied to the specific milestones of the Consortium projects, the present software-development approach needs to be modified. Specifically, we suggest considering incorporation of the following elements in future development:

- Develop use cases for the major project drivers
- Derive requirements from use cases (including performance)
- Develop build schedule with explicit release dates with incremental functionality tied to key project milestones
- Develop design and hold design reviews
- Develop tests, test data
- Develop code and documentation
- Peer review code
- Test using data tied to use cases
- Acceptance of release by Project Scientist

Many of these elements are currently present, but are not focused on the delivery milestones derived from the needs of the major Consortium projects.

Tying the content and deliveries to the projects may result in the need to abandon the current 6-month routine delivery cycle, which is more suited to a system in long-term maintenance than one undergoing significant development. The recent adoption of mini-deliveries each on 4-6 week timescales is a move in this direction.

A critical aspect of developing the AIPS++ build schedule is creating a master schedule tied to the needs of the Consortium projects. The resultant AIPS++ master schedule should be consistent with the major project schedules (such as GBT, ALMA, EVLA).

This approach would allow the answer to the question “What’s AIPS++ needed for” to be something like:

- Perform analysis of {BIMA, GBT, ALMA, LOFAR, etc.} data per the use cases and requirements documented in w, x, y and z
- Support pipeline processing of {VLA, ATCA, etc.} data per document x, y

The approach would have an associated set of processes that would be applied to all development projects, e.g., a process for generating a use case, by whom, how to ensure its part of full coverage, etc.

3. Strengthen the Project Management and the Project Team

The project team is clearly highly talented, energetic and dedicated. Although these are essential ingredients for success, by themselves they cannot result (and indeed have not resulted) in achieving the goals of the AIPS++ Project. In particular, it is evident that there are a number of weaknesses in the management of the project that must be addressed.

Project management must be strengthened in a number of areas. The Panel was struck by the lack of standard project management methodology and reports. Suitable formal project management techniques should be adopted, both for managing the project and communicating the state of the project to the Consortium management and partners. Key metrics such as earned value and progress against project milestones should be adopted, in order to show progress in a meaningful way and to allow for adjustment in scope and schedule as warranted.

The current practice of sharing the Project Manager role between several individuals, none of them on a full-time basis, presents a problem. For a project of this magnitude the Project Manager role must be recognized as a full-time position, one that cannot be shared with other duties and responsibilities.

Project management must foster a clear customer focus among the project team members, where the customers are the major Consortium programs requiring AIPS++. This is a distinct change from serving an amorphous general user community. The development of an AIPS++ master schedule tied to the Consortium projects will assist in this regard.

Every effort must be made to fill current vacancies. The team should be strengthened by the addition of professional software engineers; sufficient

astronomical expertise is already present in the existing project staff. One of the project staff positions should have the role of software architect, responsible for the overall integrity and consistency of the core classes. This is especially important, as significant further development will be required in the future.

More effort must be made to utilize full-time FTEs. Staff allocated at a small fraction of their time over an extended period are inefficient, and in the case of marginal involvement (<10%, of which there are several) they often amount to a drain on overall project resources.

Although this program could benefit from additional personnel, the Panel believes that the current team (utilizing all currently open positions, of which there are several) should first demonstrate their capability to use what resources they have more effectively. We also note that several Consortium partners are currently contributing significantly less staff than that required of them. This is a matter for the AIPS++ Executive Committee to resolve.

4. Further Strengthen the Project Scientist Role

The Panel applauds the appointment of a strong and active AIPS++ Project Scientist. It is clear that this has already had a significant and positive impact on the project. The Panel also wishes to acknowledge the recent activities of the NRAO AIPS++ User Group (NAUG). The interaction of the NAUG and the project in recent months has been clearly beneficial, and demonstrates the need and power of actively involving users in the development process.

The Panel believes the role of the Project Scientist is critical to the success of the project and, therefore, needs to be expanded in a number of ways.

Analogous to the Project Manager, the Project Scientist should be appointed at the Consortium level and be responsible to the AIPS++ Executive Committee. Individual Consortium sites and projects should appoint individual Site Scientists. The activities of these Site Scientists would be coordinated by the Project Scientist. This is similar to the user group structure presented by Steve Myers during the review, but formalized in the following ways.

In coordination with the Site Scientists, the Project Scientist should have the following responsibilities:

- Develop science-based use cases for all AIPS++ supported major Consortium instruments.
- Based on use cases, derive project science requirements. Note that science requirements should include baseline performance requirements.
- Identify and appoint Consortium staff scientists as use case testers. It is expected that these use case testers will be active scientists with research experience with their home instruments.
- Organize and prioritize use case tester feedback for project action.

- Verify that project deliverables fulfill use-case-derived requirements.
- Organize development of high-level end-user User's Manuals ("cookbooks") for major scientific threads.

In coordination with the Project Manager, the Project Scientist should have the following authority and responsibilities:

- Provide scientific prioritization for all project activity, based on clearly defined scientific use cases, and subject to the high-level project priorities established by the Consortium Executive Committee.
- Assure that project activity is aligned with the scientific requirements.
- Certify and approve for public release any new version of AIPS++, after the Project Manager is satisfied that a build is technically ready for release and after the satisfactory testing of the scientific use cases.

Technical decisions should continue to lie with the Project Manager. The Project Scientist will generate use-case-based requirements, but it is up to the Project Manager to decide how to achieve those requirements.

5. Short-Term Priorities

Development efforts over the next 12 months should focus on the following specific issues:

- Significant improvements in reliability and stability
- Performance improvements approaching the performance of AIPS
- Synthesis imaging capability covering many common use cases of the VLA
- Reduced emphasis on graphical user interfaces

The Panel endorses the current systematic approach to investigation of performance issues in AIPS++ presented at the review. Benchmarking performance of components and feeding back problems to the developers for resolution has been quite productive. It is important that the planned "best practices" manual cautioning developers against performance-robbing practices be produced soon. It seems likely that continuation with this program will increase performance to the point where it approaches that of AIPS.

There are defects in the design of many of the present GUIs, some related to the underlying implementation of Glish and others to usability issues. However, script-driven processing is adequate to demonstrate the core functionality of AIPS++ at this stage. A Viewer GUI is recognized to be essential for this mode of operation. Other graphical interfaces are required in the longer term, but should have a lower priority for now. Redesign of existing GUIs will be required, but should await resolution of the underlying GUI performance issues. The project intends to achieve the needed level of performance through a re-implementation of significant parts of the current technology. However, other less ambitious options are also available (e.g., based on the Qt widget library), and should be

considered if the implementation of the new technology takes longer than planned.

Future releases must include user-level “cookbooks” for processing data with validated use cases. Given the toolbox nature of AIPS++, such cookbooks are the most effective way for non-expert users to use the system successfully and productively.

In the case of the NRAO, the VLA provides an ideal test bed for AIPS++, not only because of the many common observing modes, but also for organizational and sociological reasons. The Array Operations Center co-locates many of the AIPS++ developers and Project Scientist as well as a large pool of scientific users of the VLA. The recent increase in involvement of the scientific staff is very encouraging as is the quick turnaround on bug fixes. The effort to identify the VLA scientific use cases needs to be completed along with systematic testing of these use cases. Successful achievement of these efforts opens the door for an AIPS++ VLA pipeline, which would complement the proposed archive. Routine operation of such a pipeline would give much needed credibility to AIPS++ and confidence that it could fulfill its Consortium obligations.

6. Proceed With the Proposed Technology Changes

The current AIPS++ architecture is built around Glish, which serves as the software bus for task communications, the task control system, the scripting language for high-level applications, the command-line interface (CLI) for user interaction and the GUI system. Since the AIPS++ system has its own custom approach to each of these needs, it is effectively a closed system that cannot easily benefit from the many developments in the rest of the software world. In fact, many of these functions might be better performed by standard components developed in the software community over the last 10 years.

The proposed new technology appears to be a sound basis for changing AIPS++ to an open system that uses industry standards where appropriate (e.g., CORBA, Java) and allows the use of robust open-source software in place of AIPS++ software (e.g., Python instead of Glish). By moving Glish out of the core of the system, a major bottleneck for improving the performance of GUIs is also eliminated. Using the ALMA Common Software as the basis for the bus is also appealing, since that software is already in use within the project.

The case made for a new component-based distributed architecture is strong. A preliminary analysis of implementation technologies has been performed. This analysis should continue and should be the major focus of any proof-of-concept. In particular, the Alma Common Software (ACS) appears to be well aligned in terms of architecture and technology. However, the current ACS implementation was designed for a different purpose and may not be suitable for use within a data analysis framework without significant rework and extension. Collaboration with

ESO on a common "ACS-light" core could be beneficial to all concerned, but there may be difficulties in reconciling the development and release needs of the ACS in support of ALMA with the needs of the AIPS++ project. The project should prepare for the possibility that they will be responsible for the work to integrate ACS (or other technology identified by the feasibility study) into the AIPS++ system. A common software framework is desirable, but it may prove to be impractical to share this core software with other systems.

The first step in considering the technology change should be a feasibility study that evaluates alternative technologies relative to the long-term project requirements. It is essential that this evaluation not interfere with the ongoing improvements in functionality, robustness and performance required for the AIPS++ tools and applications. Parallel efforts focusing separately on completing the current AIPS++ science data-processing capabilities and prototyping a new system framework are therefore called for. However, the staffing for the current AIPS++ applications group must not be reduced; moreover the effort of evaluating and implementing the new technology will require staff with a strong software engineering background, rather than that of astronomical data processing. These requirements point to the need for additional resources, probably carried out by a group that is independent of the existing project team.

Appendix A: Charge to the Review Panel

The AIPS++ package has been under development as a general radio astronomical data analysis package for about 10 years and has established some significant capabilities. Nevertheless, the AIPS++ package has not reached the desired final state of being the data analysis package of choice, and there are some adverse perceptions of AIPS++ in the user community. These perceptions include:

- AIPS++ runs too slowly.
- AIPS++ is hard to learn and is difficult to use.
- After 10 years of development, AIPS++ functionality still significantly lags that of older packages.
- It is difficult to program in AIPS++, leading to long development timescales and initial reliability problems.

Objectively, the changes that need to be made to the project to arrive at the final state of AIPS++ can range from minor modifications to major revisions, or even to seeking an alternate approach altogether, but any proposed solution to the current problems cannot be open-ended in time and resources. Consequently, the AIPS++ Executive Committee has agreed that a Technical Review of AIPS++ is needed as soon as possible to inform decisions.

The purpose of this AIPS++ Technical Review is thus twofold:

1. To evaluate the technical content and suitability of the AIPS++ software package as a radio astronomical data processing program for current and future instruments; and
2. To make specific recommendations, wherever possible, on how to make AIPS++ a useful and user-friendly data reduction package.

In its report, the panel is asked to estimate the consequences in schedule and resources of any substantial changes recommended. The panel's final report will be used by the AIPS++ Consortium Executive Committee to help determine the future status and course of development for the AIPS++ project. The panel is asked to present its final report by no later than 2 months following the review meeting.

AIPS++ Executive Committee
January 13, 2003

Appendix B: Specific Questions to the Panel Posed by the Executive Committee

Q1 Goals: Are the project goals (*e.g.* the mission statement – see below, development plans) clear and appropriate for the needs of the AIPS++ consortium and the international community? Are the near and long term goals well defined and aligned with the needs of the current and future customer base? Are the goals achievable on timescales acceptable to the customers? Does the AIPS++ Project have the correct vision for the future evolution of the package?

Q2 Design: Is the design of AIPS++ sound, such that it will fulfill the requirements of the projects of the consortium partners over the projected timescale of the current decade?

Q3 Architecture: Is the underlying software architecture sufficiently flexible and maintainable? Does it allow easy upgrade paths, such as parallelization, use over the Internet, or use of grid computing? Will AIPS++ be flexible enough to handle heretofore-unforeseen data processing problems, such as those posed by consortium projects such as ALMA, EVLA, LOFAR, CARMA, and GBT? Will the AIPS++ component parts be inter-operable over the immediate lifetime of the project? If there are architectural or design deficiencies, what level of rework is required to correct them?

Q4 Components: Are the core components of AIPS++ (*e.g.* Glish, Table system, MeasurementSet, MeasurementEquation, User Interface) suitable for the current and future requirements of the package? Is the core calibration and imaging software design sufficient to handle the increased demands, *e.g.* dynamic range, of the new instruments? If the components are unsuitable, what changes are recommended?

Q5 Usability: There are serious concerns about the usability of the AIPS++ package, including performance, ease of use, and programmability. What specific actions should be taken to verify and quantify these concerns and to remedy the real problems?

Q6 Process: Are the project management structure and development methodology suitable for the development, construction, and maintenance of a software package the size and scope of AIPS++? Are there sufficient tools, processes and management in place for quality assurance of the package deliverables? Is the management and process structure appropriate for the distributed nature of the project? Does the development process identify and adequately incorporate the input of its current and future customer base?

AIPS++ Executive Committee
January 13, 2003