LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	**LIGO-T950021-00 -**	**Cxx**	6/29/95

Pre-Stabilized Laser CDS Software Development Plan

R. Bork

Distribution of this draft:

R. Vogt B. Althouse
V. Schmidt A. Lazzarini
R. Spero A. Kuhnert

California Institute of Technology
LIGO Project - MS 102-33
Pasadena CA 91125
Phone (818) 395-2966
Fax (818) 304-9834
E-mail: info@ligo.caltech.edu
WWW: http://www.ligo.caltech.edu

TABLE OF CONTENTS

TABLE OF CONTENTS

1 INTRODUCTION

This Software Development Plan (SDP) describes the software management and development process for the Control and Data System (CDS) necessary to control and monitor the LIGO Pre-Stabilized Laser (PSL). Note that this document contains more general information and "boiler plate", which would typically reside in a higher level document, than is intended for most projects at this level. Since there is not presently a "Global" CDS SDP in place, and the project schedule does not allow time for one to be developed, this document is created with plans and procedures which could apply to all CDS projects. It is intended that sections herein which fall into that category of general policy later be incorporated into and superceded by a CDS SDP.

1.1. Project Overview

1.2. Project Deliverables

The project objective is to provide all software as necessary to control and monitor the LIGO PSL section of the LIGO Interferometer (IFO). This includes the real-time software as necessary to provide automated closed loop control, networking communications to move data in a distributing computing environment, and operator services, such as operator displays, alarm management, slow (10Hz or slower) data archival/retrieval, and system state save and restore capabilities. This does not include high speed (>10Hz) data acquistion and CDS remote diagnostics, which are covered elsewhere as separate projects.

1.3. Evolution of the SDP

This is intended to be a living, working document over the lifecycle of the project. It will be reviewed for accuracy prior to any formal reviews, whenever higher level LIGO management policies are published to ensure adherence to LIGO standards, and when plan changes are approved which affect this document.

1.4. Reference Materials

1. Pre-Stabilized Laser Control Requirements (PSLCR) LIGO-T950001-1-C

1.5. Definitions and Acronyms

CDS - Control and Data System

CI - Configuration Index

CIM - Computer Integrated Manufacturing

CSR - Pre-Stabilized Laser Control System Requirements

DRR - Design Requirements Review

EPICS - Experimental Physics and Industrial Control System

FDR - Final Design Review

IFO - Interferometer

PDR - Preliminary Design Review

PSL - Pre-Stabilized Laser

SCCS - Source Code Control System

SDL - Software Development Librarian

SMR - Software Maintenance Request

SRS - Software Requirements Specification

STP - Software Test Plan

TBD - To Be Determined

1.6. Document Precedence

In the event of conflict between this document and other LIGO documentation, the order of precedence, for this particular project, shall be:

1. LIGO Project Management Plan
2. LIGO Detector Implementation Plan
3. LIGO Project Cost and Schedule Documentation
4. Reference 1
5. This document

2 PROJECT ORGANIZATION

2.1. Process Model

The basic development process model is described in Section 4 of this document. Exact procedures and management processes will be in accordance with the LIGO Detector Implementation Plan. Project schedule and milestones are officially maintained by LIGO Project Management. Key milestone preferred target dates for this project are reproduced here for completeness only. Refer to LIGO Project Management maintained schedules for official dates.

1. Design Requirements Review (DRR) - February 3, 1995
2. Preliminary Design Review (PDR) - August 1, 1995
3. Final Design Review (FDR) - April 17, 1996
4. Initial Installation on LIGO IFO - July 31, 1998

In addition to these primary milestones, this particular development process will support prototype tests in the LIGO optics laboratory and LIGO 40M prototype. Key dates for this include:

1. Installation in optics lab system - May 4, 1995
2. Installation in CalTech 40M Prototype- September, 1995

2.2. Organizational Structure

The PSL is to be developed as a team effort within the LIGO Detector Group. This team is made up of members from both the CDS and IFO sections within the Detector Group. While CDS will be the primary provider of the PSL software, since the final product is the PSL itself, all members involved share responsibility in its successful development.

3 MANAGEMENT PROCESS

3.1. Management Objectives

The primary goal of this project is to provide quality software which is an integral part of a Detector Group team effort to provide a fully functioning PSL which meets the requirements of the LIGO detector. Management objectives toward meeting this goal are:

- Early guidance and planning of the project
- Risk Assessment and Analysis
- Incorporating configuration control procedures
- Establishing standard software procedures and coding areas.

3.2. Assumptions, Dependencies and Constraints

3.2.1. Schedule

This document assumes that the present LIGO schedule will be followed for implementation, which begins with a prototype phase in the LIGO Optics Lab, and later is installed in the CalTech 40M interferometer prototype.

3.2.2. PSL Configuration

This document assumes that the present PSL optics layout is to be used with only possible minor modifications and that this design is consistent for the multiple PSL subsystems which are to be installed at LIGO sites. This document does not take into account the possibility of multiple PSL subsystems per interferometer for redundancy.

3.3. Risk Management

Risk will be analyzed throughout the project lifecycle in terms of technical, cost and schedule risks. Risk analysis shall be presented at each review, along with mitigation techniques. Initial risk assessment is in the following paragraphs.

3.3.1. Technical

Technical risk is considered to be minimal. No new, beyond state of the art software is anticipated. Initial review of the PSL Control Requirements Specification (PSLCRS) document indicates no unattainable real-time constraints or general functionality. Where some real-time features may be incorporated in hardware or software, trade-off studies and/or software prototyping shall be performed early in the development process to minimize risk.

3.3.2. Cost

A cost estimate for the project has been resently submitted, and therefore cost risk is considered low. Tracking and minimizing risk shall be in accordance with the LIGO cost and schedule tracking policies.

3.3.3. Schedule

Schedule risk is considered medium. Reasons schedule risk are higher for this project:

1. Minimal in-house staffing when considered with the number of other concurrent LIGO CDS projects.
2. Some contract support will be required to upgrade the software tools required for implementation.
3. First LIGO application project undertaken by the CDS group and it comes early in the LIGO project. This means that not all CDS standards have yet been defined/adopted at higher system levels, which would typically come first prior to a lower level application such as the PSL controls. This results in development of standards during the project and perhaps some retrofitting when standards are adopted.

To reduce this risk, the following actions are to be taken:

1. Active recruitment of additional programming staff
2. Early contact with software contractor (already in progress) to update software tools.
3. Concurrent work to get CDS software/hardware standards in place as quickly as possible.

3.4. Personnel Safety and Equipment Protection

The CDS system will be analyzed from the point of view of personnel safety and machine protection throughout the system lifecycle. Items directly linked to personnel safety will never be resolved in software as the first line of protection. In those cases, hardware will always be the primary safeguard, with software systems only employed in a backup and monitoring role.

Equipment protection may be done in software, depending on the outcome of risk analysis and management decisions. In those cases, the software involved will undergo higher levels of scrutiny during the development and test cycles.

3.5. Monitoring and Controlling Mechanisms

Monitoring and control mechanisms shall be in accordance with LIGO project management plans.

As a minimum, software development will be reviewed at the DRR, PDR and FDR, as called for in the Detector Implementation Plan. Materials to be presented at each review are described in the following paragraphs. More information on activities leading to these reviews and specified documentation can be found in Section 4 of this document.

3.5.1. Formal Reviews

3.5.1.1 DRR

A DRR will be conducted prior to the preliminary design phase. Materials to be presented for review are:

1. Software Development Plan (DRAFT)
2. Initial concepts for software and the requirements that will be addressed/implemented in software to meet the PSL Control Requirements Specification.

3.5.1.2 PDR

A PDR will be conducted after the initial installation and operation on the PSL located in the optics lab, which, for the CDS, acts as a prototype. Materials to be presented at the PDR for software are:

1. SDP (Final)
2. Software Requirements Specification (SRS) (DRAFT)
3. Prototype Test Results
4. Preliminary Design

3.5.1.3 FDR

An FDR will be conducted at the conclusion of the final design phase. Materials to be presented at the FDR include:

1. SRS (Final)
2. Final Design
3. Software Test Plan (STP)
4. Release 0 code and documentation

3.5.2. Informal Reviews

The CDS group software developer's must work as an integral part of a team with other members of the CDS and the scientist's assigned to the PSL to provide a tightly integrated, functional product. As part of this interaction, informal reviews will be conducted within the "PSL team" of scientists and engineers.

3.5.2.1 Design Walkthroughs

This is an informal method used to determine the completeness of a design. The designer conducts the review with attendees representing all affected interfaces. This would also be a forum for the verification of requirements and trade studies.

3.5.2.2 Code Walkthroughs

Verification of design is the primary goal of code walkthroughs. A secondary goal is to check compliance with the CDS adopted software style (TBD), which is important for long term maintenance of the code. Code walkthroughs are particularly important for complicated logic, often found in distributed and/or real-time systems. These walkthroughs will be typically held within the PSL team.

3.5.2.3 Component Informal Reviews

These reviews are frequent, and are intended to allow software to be seen "with a second set of eyes". They occur during work in progress to help verify, at each step, functionality and capabilities of the code, such that a long development has not used up a fair fraction of the schedule prior to a review. They also help ensure that more than one person is familiar with software components.

3.5.3. Software Configuration Management

Software configuration management includes the activities of configuration identification, change control, status accounting, and audits. Baseline software configurations will be provided by the CDS librarian to LIGO Configuration Control. Pre-baseline development configurations as well as baseline configuration will be controlled by the CDS librarian. The approved baseline documents for each component including SRS, STP, and SDP will also be controlled in accordance with LIGO configuration management procedures.

3.5.3.1 Flow of Configuration Control

The software and documentation developed for the PSL will move through distinct areas, as shown in the following figure, to help maintain configuration control. The general flow of software and documentation is:

1. Development Area: Area in which software engineers work on code in progress. This area has symbolic links to the Release Area, to ensure the developer is using the latest released versions of software operating systems and tools.
2. Proto/Test Area: Once a developer is satisfied that particular code is ready for release, the code and documentation is moved into a Prototype/Test area. Here the code is integrated and independently tested/operated as part of an overall system. Code may move back and forth between these first two areas as bugs/faults are detected and repaired. Faults/desired corrections are documented with a Software Maintenance Request (SMR) (discussed later), which travels with the code and is maintained in a database to track the history of software.
3. Software Development Librarian (SDL) Area: This is the repository for all code which has passed test and is ready for installation. The assigned librarian is then responsible for integration of all such code and coordinates the update into the Release Area.
4. The Release Area is where all installed LIGO operational systems derive their software.

Further expansion and definition of these areas is covered in the Technical Process section of this document.

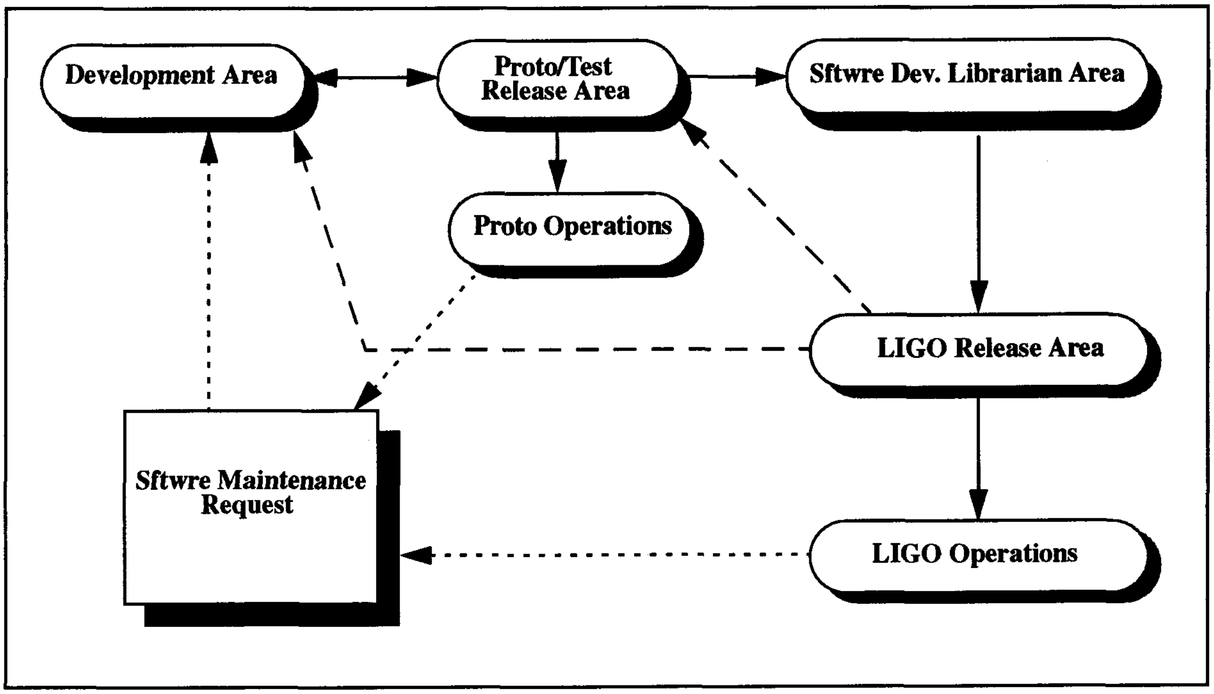

Figure 1: Software Configuration Control Flow

3.5.3.2 Configuration Control Tools

All software, in every area described above, will always be under source code control, using the Source Code Control System (SCCS), provided as a standard feature in UNIX.

3.5.3.3 Configuration Identification

The configuration identification for each code module will be the revision number assigned automatically by the SCCS. Once the SDL has integrated the various code modules, and it has been approved by the CDS Task Leader for release, the integrated code will be put under a unified SCCS revision number and released.

The release numbering scheme shall be a three number convention in the form x.xx, such as 1.23. The first number shall indicate a major release. Major releases are typically limited to when the code has undergone major core structural changes or a significant number of enhancements have been made. The second number is changed when a release has new features/enhancements. The final number indicates that bug fixes have been made without the addition of particular features.

3.5.3.4 Handling of Project Media

All CDS documents, source files and build instructions will be locally controlled by the SDL. Whenever a CDS product is approved and baselined (sent to the Release Area), a copy of all materials shall also be turned over to the LIGO Integration Group for LIGO Document Control.

3.5.3.5 Enhancements and Changes (Corrective Action)

3.5.3.5.1 Software Maintenance Request

Once software has left the development area, all requests for enhancements or corrections are documented in an SMR. An SMR has three basic parts:

1. Problem reporting/enhancement request area submitted by the software end user.
2. Analysis section, wherein the assigned software engineer analyzes the problem/request and provides recommendations to resolve the request.
3. Resolution Area: Information on how the request/problem was resolved.

A database will be kept of all SMR to help trend and monitor software development projects, which may point to certain areas which may need closer investigation for future software releases.

3.5.3.5.2 Corrective Action Process

Once an SMR is originated, it is submitted to the CDS Task Leader. He/she then assigns both a priority to the SMR and a person to be responsible for analyzing/resolving the request. Priorities are assigned according to the following table.

Table 1:

Priority	*Description*
1	The problem prevents LIGO from operating to its specified performance as a detector. The problem jeopardizes personnel safety or machine protection.
2	The problem adversely affects either an essential capability specified in the requirements or the operator's accomplishment of that capability, and no work-around is known.
3	Same as 2 above, but a work-around is known which may be put in place as a temporary solution.
4	The problem causes inconvenience or annoyance but does not affect a requirement.
5	All others not falling into a category above.

In the event an SMR is a request for enhancements or change in project scope, the CDS Task Leader will determine if this request must be processed through the LIGO Change Control System prior to further assignment to a software engineer.

Once the SMR has been analyzed and response returned to the CDS task leader, it is reviewed and assigned for implementation. Here it undergoes the same procedures as apply for new software development. Upon completion of test, the SMR is completed by the developer and returned to the CDS Task Leader for closeout.

From date of origin/receipt until closeout, the status of SMR's will be updated on a weekly basis, with a status page made publicly available such that end users and management can be kept apprised of SMR progress.

3.5.3.6 Configuration Management Documentation and Reporting

The primary reporting will be in the form of a Configuration Index (CI). During the design phase, software components will be identified, which are then tracked thoughout the software lifecycle. One CI is prepared for each of these components. The CI includes an historical record section, milestone data, list of associated documentation, and a list of applicable SMR or other project change requests (including status/disposition).

4 TECHNICAL PROCESS

4.1. Software Development Life Cycle

Software development will follow the standard waterfall life cycle as much as possible. This cycle is shown in Figure 2: Software Development Cycle. While this outlines the general flow of development, reiteration between certain phases will occur, for instance, prototype and test may indicate that requirements need to be changed/updated or new approach taken.

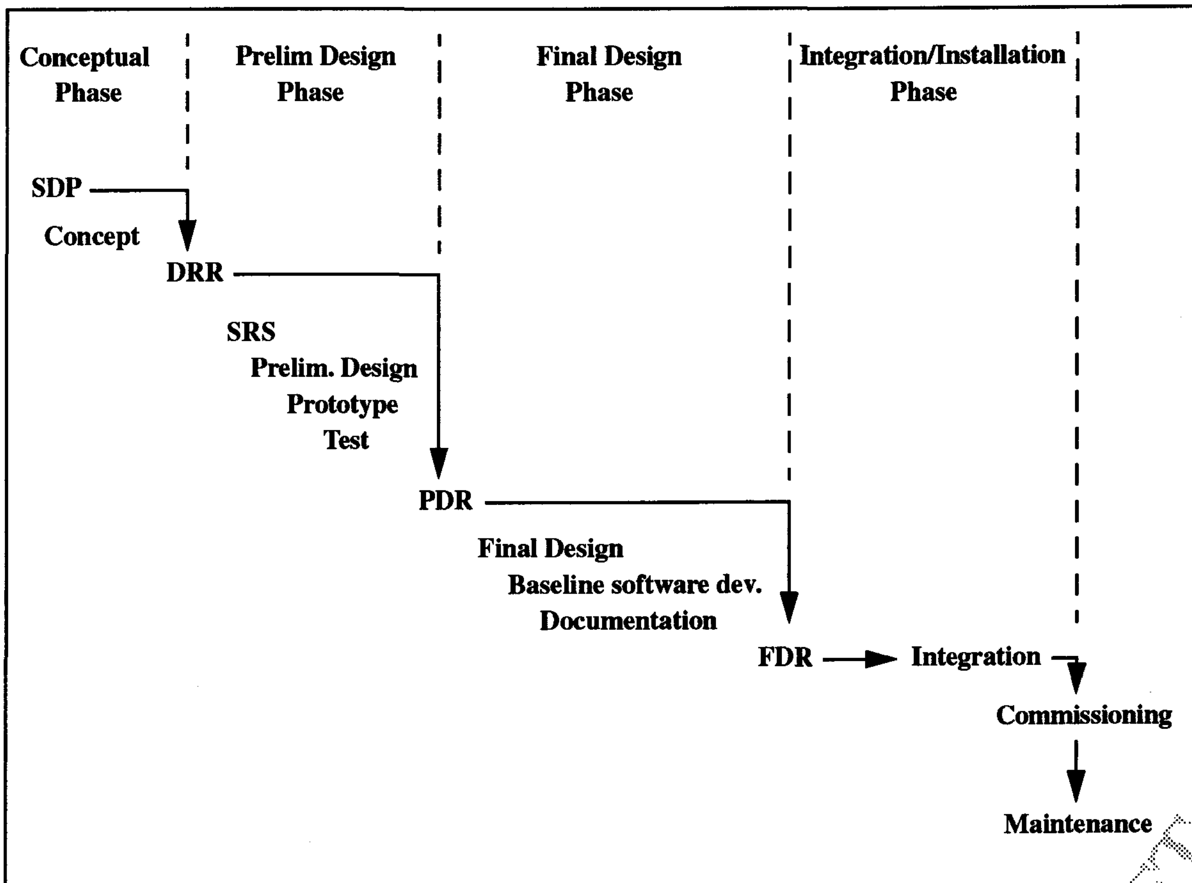

Figure 2: Software Development Cycle

4.1.1. Conceptual Phase

During the conceptual phase, a plan is developed (this SDP document), along with a conceptual design. This conceptual design is based on early extraction of requirements from the PSLCRS, which is an overall system performance specification. The Conceptual Design Phase ends with a formal DRR.

4.1.2. Preliminary Design Phase

During the preminінary design phase, software and electronic engineers will work closely to develop an overall system design, determining which parts of the system will be implemented in hardware and which in software. As this develops, requirements will be imposed on the software, which will be documented in a PSL SRS. These requirements shall be documented in a modular fashion to coincide with the CIM model developed for the PSL CRS such that design documentation and requirements can be closely coupled, both for ease of review to ensure designs meet requirements and also to develop a Software Test Plan (STP) which verifies that the software implementation meets the requirements. This model for the PSL is shown in Figure 3: PSL CIM Model. Further definition of PSL control levels can be found in the PSL CRS.

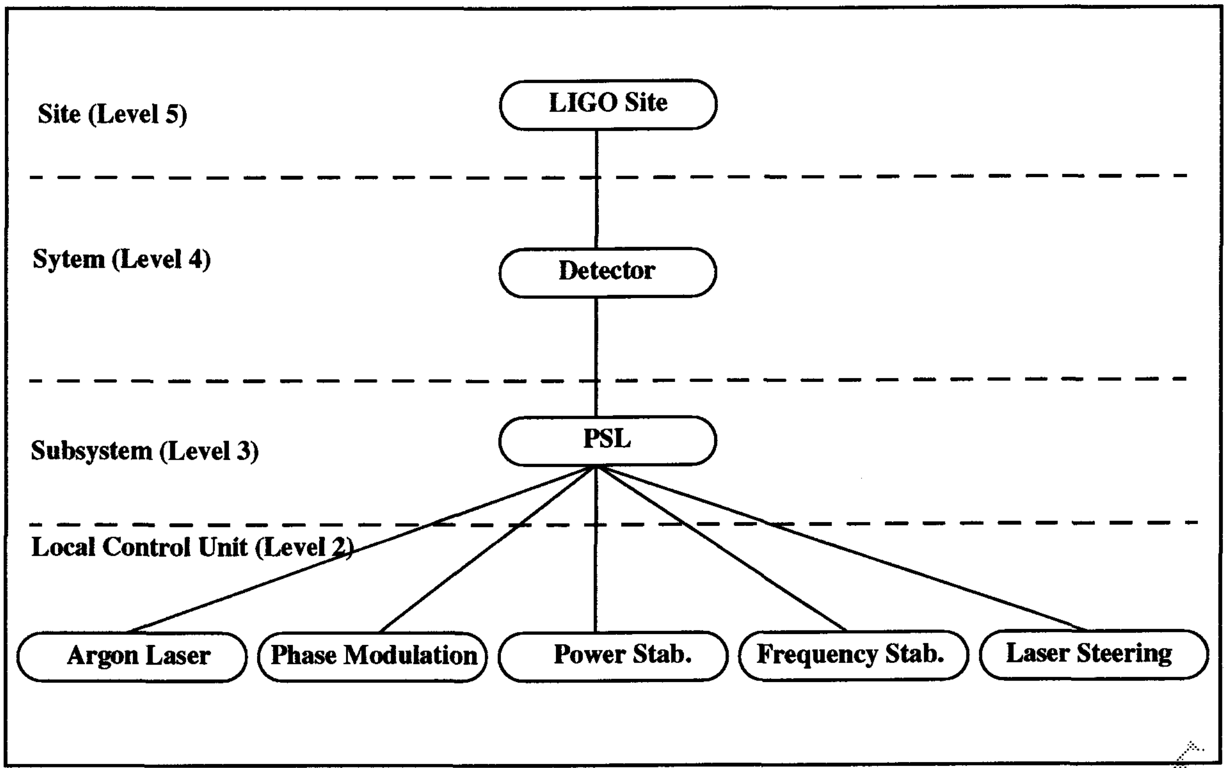

Figure 3: PSL CIM Model

During this phase, prototype software will also be developed and tested, with emphasis on those components which are determined to have a higher risk. This software will be component tested off-line by the developer, and then transferred to the PSL system in the optics lab for initial system testing.

At the end of this phase, all design issues should have been resolved and a complete design presented at a formal PDR.

4.1.3. Final Design Phase

In this phase, any outstanding design issues from the PDR are resolved, and final code implementation begins. A “Release 0” software package should be completed prior to a FDR at the end of this phase. “Release 0” is defined as that which meets the minimum requirements to begin installation and commissioning of LIGO. This distinction is made because the PSL software will undergo integration with other LIGO systems after the FDR, and certain changes and/or enhancements may be added during that time.

“Release 0” shall contain:

1. Code necessary to allow commissioning of the PSL system.
2. Supporting documentation of successful completion of testing defined in the STP.
3. The first set of software system documentation.

4.1.4. Integration and Commissioning Phase

These areas are outside of the scope of PSL construction. These phases are described in CDS document TBD.

4.2. Software Development Methodology

4.2.1. Software Standards

This software will be developed in accordance with the standards of the LIGO CDS group for all software development projects, as outlined in LIGO document TBD.

4.2.2. Design and Development Tools and Techniques

4.2.2.1 CASE and Development Tools

The PSL software will be designed and developed based on the present capabilities of the Experimental Physics and Industrial Control System (EPICS) software package, distributed by Los Alamos National Laboratory (LANL). For this project, EPICS will be used “as is” for the bulk of the activities, except to address those issues where use of EPICS is analyzed as not being able to meet particular PSL requirements.

Particular tools to be employed from the EPICS toolkit are:

1. Microsoft Access for EPICS database generation. An interface has been developed at LIGO to allow EPICS database generation with the commercial Microsoft Access product. This will be used for initial development. Later, Oracle RDMS will replace Access, as this product will be the LIGO standard.

2. The Motif EPICS Display Manager (MEDM) will be used for operator display development.

3. The ARchiver (AR) tool will be used to setup/initiate data archival; the tcl/tk ARchive Retrieval (ARR) tool will be used to extract and display archived data.

4. EPICS State Notation Language (SNL) will be used to develop real-time sequencing software.

5. The ALarm Handler (ALH) software will be used to develop alarm processing structures, along with the monitoring, reporting and archiving of PSL alarms.

4.2.2.2 Programming Languages

In those cases were EPICS will not meet a requirement, or other new software needs to be developed, the 'C' language will be used, designed with standard software design tools.

4.2.2.3 Documentation Tools

All documentation will be produced with the LIGO standard publishing packages. When approved, this documentation will be reproduced on the World Wide Web.

4.2.3. Target Systems

Target systems are defined as those hardware platforms and software systems on top of which the CDS software will operate.

4.2.3.1 Operating Systems

Two computer operating systems will be employed in this project:

1. Sun Unix 4.1.3 for code development and Sun workstation targets. It is intended that this be upgraded to Sun Solaris 2 once all development tools have been ported and tested by the EPICS collaboration.
2. VxWorks real-time operating system for the Motorola 68040 processor targets.

4.2.3.2 Hardware Resources

4.2.3.2.1 Development Hardware

The software development hardware will be Sun workstations.

4.2.3.2.2 Target Hardware

The project involves two types of target hardware:

1. Sun workstations for operator and file services.
2. VME-based Motorola 68040 processors for real-time control applications.

4.2.4. Security

Routine log-in procedures requiring user identification and password provides access control to CDS software areas. These areas will be open to all LIGO personnel for data "read", but will be write accessable only by the CDS group members. At minimum, weekly backups of all files will be maintained in order to prevent catastrophic loss of data.

4.2.5. Code Development Cycle

As software progresses through its lifecycle, it moves into different areas as shown previously in the Configuration Control section of this document. The contents of these areas and procedures by which it moves are outlined in the following sections.

4.2.5.1 Software Engineer's Work (Developer's) Area

The software engineer's work area is established prior to the beginning of software development. This area contains the current in progress code and documentation files. For the PSL project, this area will reside on the CDS group server (kater) under "/home/CDS/a" (the first of two 1GByte external disk drives on this server). The directory structure will be as shown in the following figure and described in the following subparagraphs. These basic structure will be continued as the software moves through the various areas of software development.

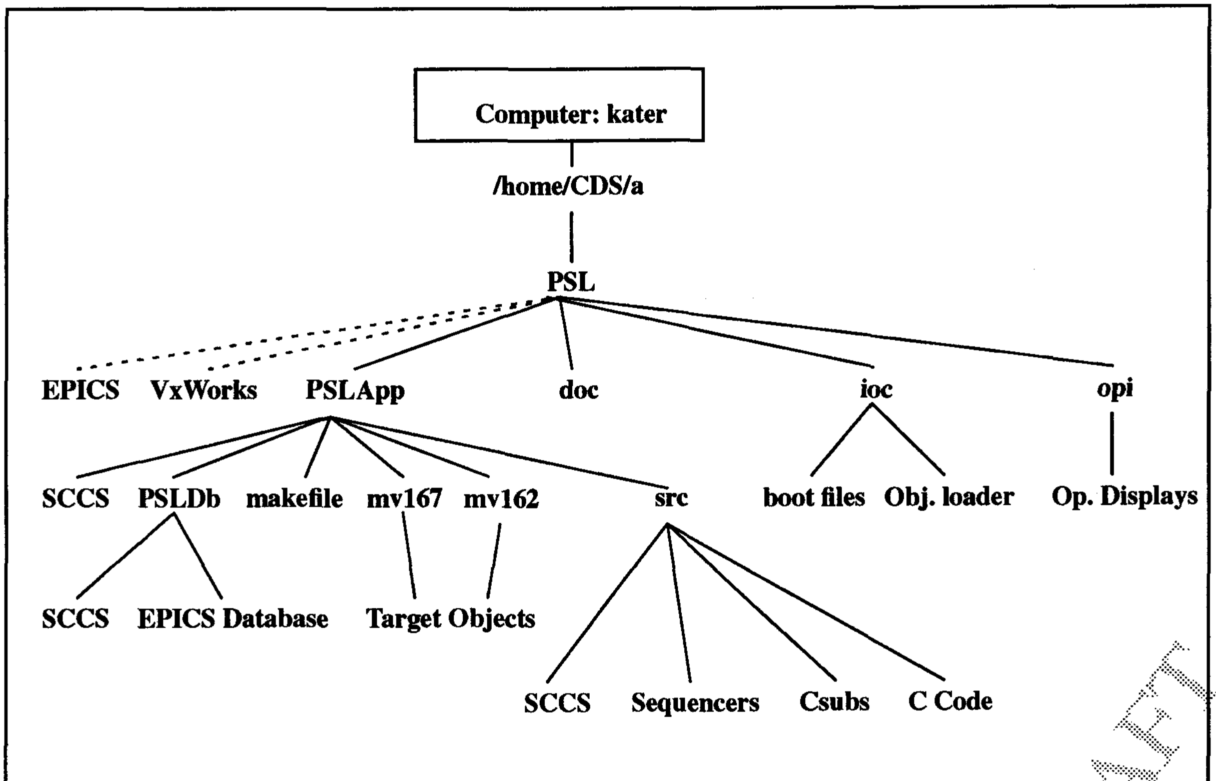

Figure 4: Development Area Directory Structure

4.2.5.1.1 EPICS and VxWorks

The dotted lines in the previous diagram denote that a copy of EPICS and VxWorks will not reside the the developer's area. To ensure that development is always with the latest and common version of this software, this area only contains symbolic software links back to the EPICS and VxWorks versions which reside in the LIGO Release Area.

4.2.5.1.2 PSLApp

This is the primary application development directory. It contains:

1. The SCCS directories.

2. PSLDb: All EPICS databases being developed.

3. A makefile which contains the computer instructions for compiling all application software within the development area.

4. Target directories for the real-time code objects generated by the makefile. These are mv167 and mv162 (Motorola VME model 167 and model 162), the two processor units presently supported by the CDS group.

5. src: Contains all SNL (*.snl) application files, and all C code, developed either into the EPICS database as CSUB records or stand-alone code.

6. doc: All documentation pertaining to this development project.

7. ioc: Subdirectories by development test target VME processor names. This area contains all of the startup and booting sequences required by development real-time processors.

8. opi: contains all operator displays developed for the project.

4.2.5.2 Software Prototype/Test Release Area

Once software is released by the developer for integrated testing, it is placed into SCCS and moved to the Prototype/Test Release Area. This area will mirror the developer's area directory structure, but be located on "kater" at "/home/CDS/b" (the second server disk drive). From here, code will be loaded to the TBD workstation and VME systems used to test the PSL in the optics lab and 40M Lab. The primary change in data in this directory will be new entries in the "ioc" subdirectory to cause the real-time software to load into the prototype/test target processors instead of the development area processors.

Once software in this area has completed successful testing, and endorsed by at the PSL FDR, it will be moved to the SDL area. To first order, at this point the PSL project described in this document is complete.

4.2.6. Documentation

Project documentation include those items previously described within this document, along with the documentation called for in CDS document TBD.

LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	**LIGO-T950021-00 - Cxx** 6/29/95
Pre-Stabilized Laser CDS Software Development Plan	
R. Bork	

Distribution of this draft:

R. Vogt B. Althouse

V. Schmidt A. Lazzarini

R. Spero A. Kuhnert

California Institute of Technology
LIGO Project - MS 102-33
Pasadena CA 91125
Phone (818) 395-2966
Fax (818) 304-9834
E-mail: info@ligo.caltech.edu
WWW: http://www.ligo.caltech.edu

TABLE OF CONTENTS

TABLE OF CONTENTS

1 INTRODUCTION

This Software Development Plan (SDP) describes the software management and development process for the Control and Data System (CDS) necessary to control and monitor the LIGO Pre-Stabilized Laser (PSL). Note that this document contains more general information and "boiler plate", which would typically reside in a higher level document, than is intended for most projects at this level. Since there is not presently a "Global" CDS SDP in place, and the project schedule does not allow time for one to be developed, this document is created with plans and procedures which could apply to all CDS projects. It is intended that sections herein which fall into that category of general policy later be incorporated into and superceded by a CDS SDP.

1.1. Project Overview

1.2. Project Deliverables

The project objective is to provide all software as necessary to control and monitor the LIGO PSL section of the LIGO Interferometer (IFO). This includes the real-time software as necessary to provide automated closed loop control, networking communications to move data in a distributing computing environment, and operator services, such as operator displays, alarm management, slow (10Hz or slower) data archival/retrieval, and system state save and restore capabilities. This does not include high speed (>10Hz) data acquistion and CDS remote diagnostics, which are covered elsewhere as separate projects.

1.3. Evolution of the SDP

This is intended to be a living, working document over the lifecycle of the project. It will be reviewed for accuracy prior to any formal reviews, whenever higher level LIGO management policies are published to ensure adherence to LIGO standards, and when plan changes are approved which affect this document.

1.4. Reference Materials

1. Pre-Stabilized Laser Control Requirements (PSLCR) LIGO-T950001-1-C

1.5. Definitions and Acronyms

CDS - Control and Data System

CI - Configuration Index

CIM - Computer Integrated Manufacturing

CSR - Pre-Stabilized Laser Control System Requirements

DRR - Design Requirements Review

EPICS - Experimental Physics and Industrial Control System

FDR - Final Design Review

IFO - Interferometer

PDR - Preliminary Design Review

PSL - Pre-Stabilized Laser

SCCS - Source Code Control System

SDL - Software Development Librarian

SMR - Software Maintenance Request

SRS - Software Requirements Specification

STP - Software Test Plan

TBD - To Be Determined

1.6. Document Precedence

In the event of conflict between this document and other LIGO documentation, the order of precedence, for this particular project, shall be:

1. LIGO Project Management Plan
2. LIGO Detector Implementation Plan
3. LIGO Project Cost and Schedule Documentation
4. Reference 1
5. This document

2 PROJECT ORGANIZATION

2.1. Process Model

The basic development process model is described in Section 4 of this document. Exact procedures and management processes will be in accordance with the LIGO Detector Implementation Plan. Project schedule and milestones are officially maintained by LIGO Project Management. Key milestone preferred target dates for this project are reproduced here for completeness only. Refer to LIGO Project Management maintained schedules for official dates.

1. Design Requirements Review (DRR) - February 3, 1995
2. Preliminary Design Review (PDR) - August 1, 1995
3. Final Design Review (FDR) - April 17, 1996
4. Initial Installation on LIGO IFO - July 31, 1998

In addition to these primary milestones, this particular development process will support prototype tests in the LIGO optics laboratory and LIGO 40M prototype. Key dates for this include:

1. Installation in optics lab system - May 4, 1995
2. Installation in CalTech 40M Prototype- September, 1995

2.2. Organizational Structure

The PSL is to be developed as a team effort within the LIGO Detector Group. This team is made up of members from both the CDS and IFO sections within the Detector Group. While CDS will be the primary provider of the PSL software, since the final product is the PSL itself, all members involved share responsibility in its successful development.

3 MANAGEMENT PROCESS

3.1. Management Objectives

The primary goal of this project is to provide quality software which is an integral part of a Detector Group team effort to provide a fully functioning PSL which meets the requirements of the LIGO detector. Management objectives toward meeting this goal are:

- Early guidance and planning of the project
- Risk Assessment and Analysis
- Incorporating configuration control procedures
- Establishing standard software procedures and coding areas.

3.2. Assumptions, Dependencies and Constraints

3.2.1. Schedule

This document assumes that the present LIGO schedule will be followed for implementation, which begins with a prototype phase in the LIGO Optics Lab, and later is installed in the CalTech 40M interferometer prototype.

3.2.2. PSL Configuration

This document assumes that the present PSL optics layout is to be used with only possible minor modifications and that this design is consistent for the multiple PSL subsystems which are to be installed at LIGO sites. This document does not take into account the possibility of multiple PSL subsystems per interferometer for redundancy.

3.3. Risk Management

Risk will be analyzed throughout the project lifecycle in terms of technical, cost and schedule risks. Risk analysis shall be presented at each review, along with mitigation techniques. Initial risk assessment is in the following paragraphs.

3.3.1. Technical

Technical risk is considered to be minimal. No new, beyond state of the art software is anticipated. Initial review of the PSL Control Requirements Specification (PSLCRS) document indicates no unattainable real-time constraints or general functionality. Where some real-time features may be incorporated in hardware or software, trade-off studies and/or software prototyping shall be performed early in the development process to minimize risk.

3.3.2. Cost

A cost estimate for the project has been resently submitted, and therefore cost risk is considered low. Tracking and minimizing risk shall be in accordance with the LIGO cost and schedule tracking policies.

3.3.3. Schedule

Schedule risk is considered medium. Reasons schedule risk are higher for this project:

1. Minimal in-house staffing when considered with the number of other concurrent LIGO CDS projects.
2. Some contract support will be required to upgrade the software tools required for implementation.
3. First LIGO application project undertaken by the CDS group and it comes early in the LIGO project. This means that not all CDS standards have yet been defined/adopted at higher system levels, which would typically come first prior to a lower level application such as the PSL controls. This results in development of standards during the project and perhaps some retrofitting when standards are adopted.

To reduce this risk, the following actions are to be taken:

1. Active recruitment of additional programming staff
2. Early contact with software contractor (already in progress) to update software tools.
3. Concurrent work to get CDS software/hardware standards in place as quickly as possible.

3.4. Personnel Safety and Equipment Protection

The CDS system will be analyzed from the point of view of personnel safety and machine protection throughout the system lifecycle. Items directly linked to personnel safety will never be resolved in software as the first line of protection. In those cases, hardware will always be the primary safeguard, with software systems only employed in a backup and monitoring role.

Equipment protection may be done in software, depending on the outcome of risk analysis and management decisions. In those cases, the software involved will undergo higher levels of scrutiny during the development and test cycles.

3.5. Monitoring and Controlling Mechanisms

Monitoring and control mechanisms shall be in accordance with LIGO project management plans.

As a minimum, software development will be reviewed at the DRR, PDR and FDR, as called for in the Detector Implementation Plan. Materials to be presented at each review are described in the following paragraphs. More information on activities leading to these reviews and specified documentation can be found in Section 4 of this document.

3.5.1. Formal Reviews

3.5.1.1 DRR

A DRR will be conducted prior to the preliminary design phase. Materials to be presented for review are:

1. Software Development Plan (DRAFT)
2. Initial concepts for software and the requirements that will be addressed/implemented in software to meet the PSL Control Requirements Specification.

3.5.1.2 PDR

A PDR will be conducted after the initial installation and operation on the PSL located in the optics lab, which, for the CDS, acts as a prototype. Materials to be presented at the PDR for software are:

1. SDP (Final)
2. Software Requirements Specification (SRS) (DRAFT)
3. Prototype Test Results
4. Preliminary Design

3.5.1.3 FDR

An FDR will be conducted at the conclusion of the final design phase. Materials to be presented at the FDR include:

1. SRS (Final)
2. Final Design
3. Software Test Plan (STP)
4. Release 0 code and documentation

3.5.2. Informal Reviews

The CDS group software developer's must work as an integral part of a team with other members of the CDS and the scientist's assigned to the PSL to provide a tightly integrated, functional product. As part of this interaction, informal reviews will be conducted within the "PSL team" of scientists and engineers.

3.5.2.1 Design Walkthroughs

This is an informal method used to determine the completeness of a design. The designer conducts the review with attendees representing all affected interfaces. This would also be a forum for the verification of requirements and trade studies.

3.5.2.2 Code Walkthroughs

Verification of design is the primary goal of code walkthroughs. A secondary goal is to check compliance with the CDS adopted software style (TBD), which is important for long term maintenance of the code. Code walkthroughs are particularly important for complicated logic, often found in distributed and/or real-time systems. These walkthroughs will be typically held within the PSL team.

3.5.2.3 Component Informal Reviews

These reviews are frequent, and are intended to allow software to be seen "with a second set of eyes". They occur during work in progress to help verify, at each step, functionality and capabilities of the code, such that a long development has not used up a fair fraction of the schedule prior to a review. They also help ensure that more than one person is familiar with software components.

3.5.3. Software Configuration Management

Software configuration management includes the activities of configuration identification, change control, status accounting, and audits. Baseline software configurations will be provided by the CDS librarian to LIGO Configuration Control. Pre-baseline development configurations as well as baseline configuration will be controlled by the CDS librarian. The approved baseline documents for each component including SRS, STP, and SDP will also be controlled in accordance with LIGO configuration management procedures.

3.5.3.1 Flow of Configuration Control

The software and documentation developed for the PSL will move through distinct areas, as shown in the following figure, to help maintain configuration control. The general flow of software and documentation is:

1. Development Area: Area in which software engineers work on code in progress. This area has symbolic links to the Release Area, to ensure the developer is using the latest released versions of software operating systems and tools.
2. Proto/Test Area: Once a developer is satisfied that particular code is ready for release, the code and documentation is moved into a Prototype/Test area. Here the code is integrated and independently tested/operated as part of an overall system. Code may move back and forth between these first two areas as bugs/faults are detected and repaired. Faults/desired corrections are documented with a Software Maintenance Request (SMR) (discussed later), which travels with the code and is maintained in a database to track the history of software.
3. Software Development Librarian (SDL) Area: This is the repository for all code which has passed test and is ready for installation. The assigned librarian is then responsible for integration of all such code and coordinates the update into the Release Area.
4. The Release Area is where all installed LIGO operational systems derive their software.

Further expansion and definition of these areas is covered in the Technical Process section of this document.

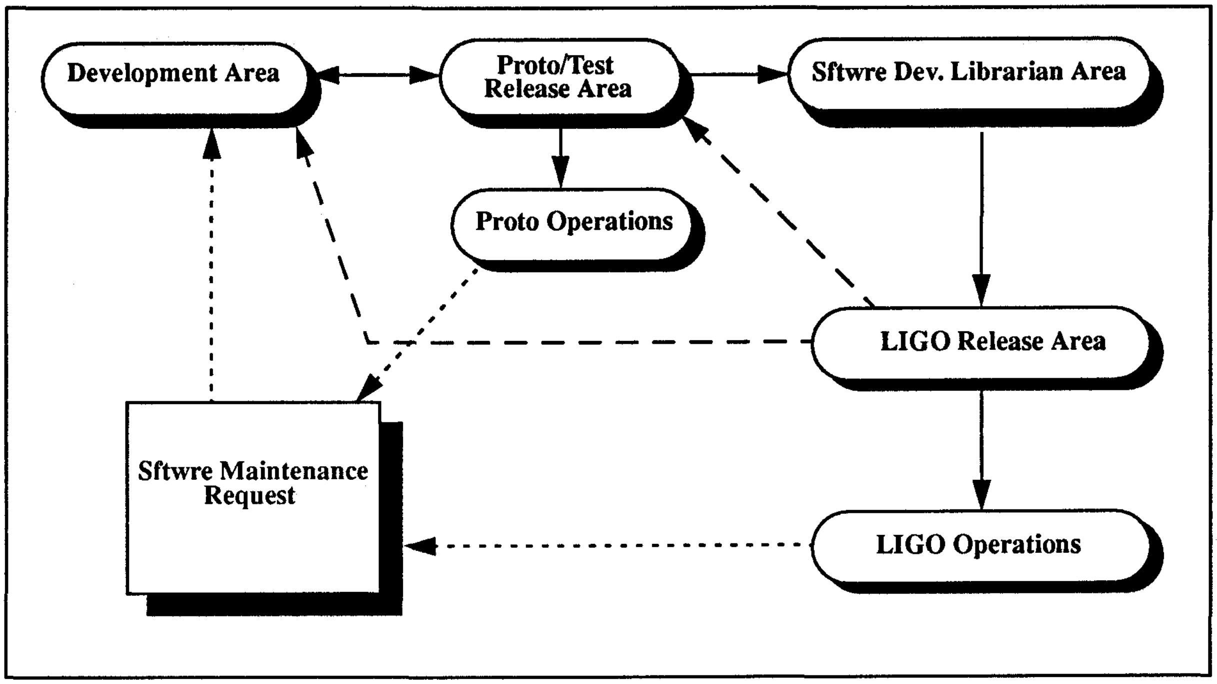

Figure 1: Software Configuration Control Flow

3.5.3.2 Configuration Control Tools

All software, in every area described above, will always be under source code control, using the Source Code Control System (SCCS), provided as a standard feature in UNIX.

3.5.3.3 Configuration Identification

The configuration identification for each code module will be the revision number assigned automatically by the SCCS. Once the SDL has integrated the various code modules, and it has been approved by the CDS Task Leader for release, the integrated code will be put under a unified SCCS revision number and released.

The release numbering scheme shall be a three number convention in the form x.xx, such as 1.23. The first number shall indicate a major release. Major releases are typically limited to when the code has undergone major core structural changes or a significant number of enhancements have been made. The second number is changed when a release has new features/enhancements. The final number indicates that bug fixes have been made without the addition of particular features.

3.5.3.4 Handling of Project Media

All CDS documents, source files and build instructions will be locally controlled by the SDL. Whenever a CDS product is approved and baselined (sent to the Release Area), a copy of all materials shall also be turned over to the LIGO Integration Group for LIGO Document Control.

3.5.3.5 Enhancements and Changes (Corrective Action)

3.5.3.5.1 Software Maintenance Request

Once software has left the development area, all requests for enhancements or corrections are documented in an SMR. An SMR has three basic parts:

1. Problem reporting/enhancement request area submitted by the software end user.
2. Analysis section, wherein the assigned software engineer analyzes the problem/request and provides recommendations to resolve the request.
3. Resolution Area: Information on how the request/problem was resolved.

A database will be kept of all SMR to help trend and monitor software development projects, which may point to certain areas which may need closer investigation for future software releases.

3.5.3.5.2 Corrective Action Process

Once an SMR is originated, it is submitted to the CDS Task Leader. He/she then assigns both a priority to the SMR and a person to be responsible for analyzing/resolving the request. Priorities are assigned according to the following table.

Table 1:

Priority	*Description*
1	The problem prevents LIGO from operating to its specified performance as a detector. The problem jeopardizes personnel safety or machine protection.
2	The problem adversely affects either an essential capability specified in the requirements or the operator's accomplishment of that capability, and no work-around is known.
3	Same as 2 above, but a work-around is known which may be put in place as a temporary solution.
4	The problem causes inconvenience or annoyance but does not affect a requirement.
5	All others not falling into a category above.

In the event an SMR is a request for enhancements or change in project scope, the CDS Task Leader will determine if this request must be processed through the LIGO Change Control System prior to further assignment to a software engineer.

Once the SMR has been analyzed and response returned to the CDS task leader, it is reviewed and assigned for implementation. Here it undergoes the same procedures as apply for new software development. Upon completion of test, the SMR is completed by the developer and returned to the CDS Task Leader for closeout.

From date of origin/receipt until closeout, the status of SMR's will be updated on a weekly basis, with a status page made publicly available such that end users and management can be kept apprised of SMR progress.

3.5.3.6 Configuration Management Documentation and Reporting

The primary reporting will be in the form of a Configuration Index (CI). During the design phase, software components will be identified, which are then tracked thoughout the software lifecycle. One CI is prepared for each of these components. The CI includes an historical record section, milestone data, list of associated documentation, and a list of applicable SMR or other project change requests (including status/disposition).

4 TECHNICAL PROCESS

4.1. Software Development Life Cycle

Software development will follow the standard waterfall life cycle as much as possible. This cycle is shown in Figure 2: Software Development Cycle. While this outlines the general flow of development, reiteration between certain phases will occur, for instance, prototype and test may indicate that requirements need to be changed/updated or new approach taken.

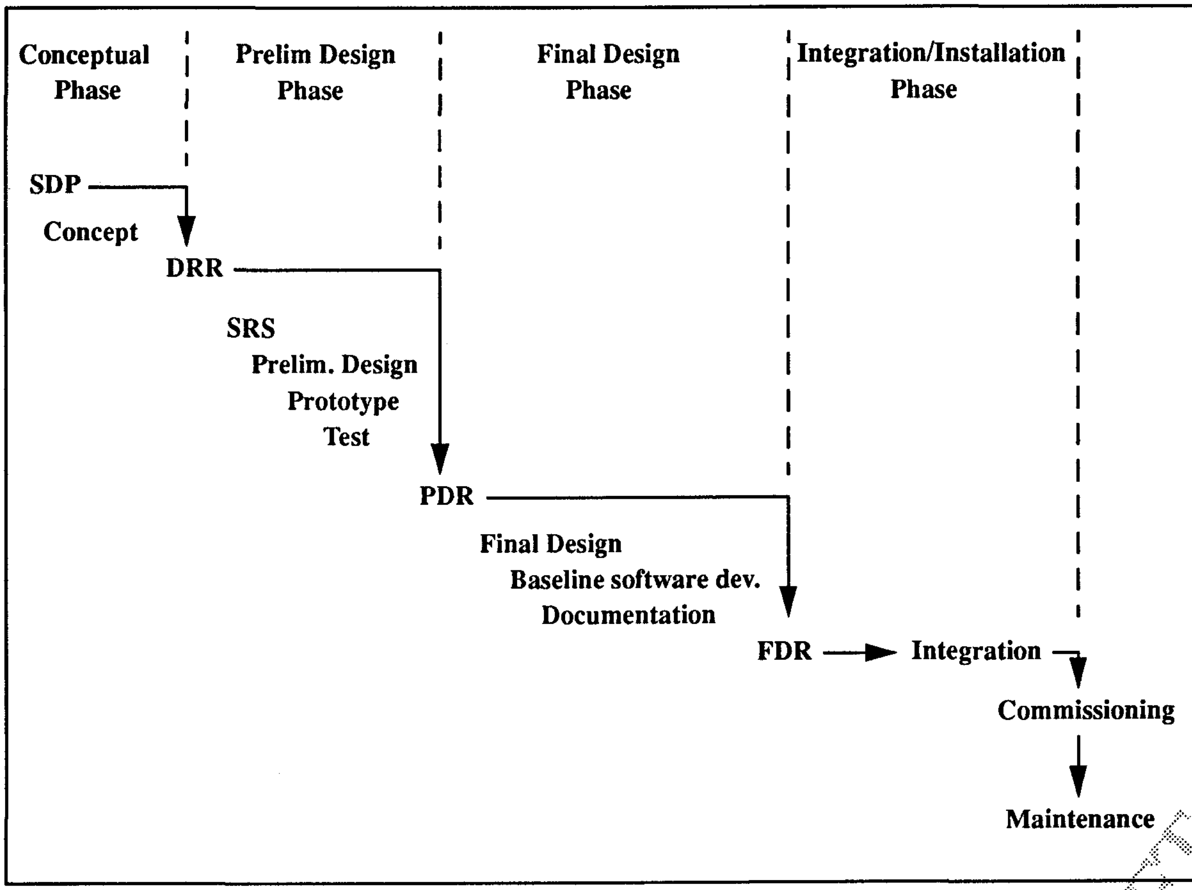

Figure 2: Software Development Cycle

4.1.1. Conceptual Phase

During the conceptual phase, a plan is developed (this SDP document), along with a conceptual design. This conceptual design is based on early extraction of requirements from the PSLCRS, which is an overall system performance specification. The Conceptual Design Phase ends with a formal DRR.

4.1.2. Preliminary Design Phase

During the premininary design phase, software and electronic engineers will work closely to develop an overall system design, determining which parts of the system will be implemented in hardware and which in software. As this develops, requirements will be imposed on the software, which will be documented in a PSL SRS. These requirements shall be documented in a modular fashion to coincide with the CIM model developed for the PSL CRS such that design documentation and requirements can be closely coupled, both for ease of review to ensure designs meet requirements and also to develop a Software Test Plan (STP) which verifies that the software implementation meets the requirements. This model for the PSL is shown in Figure 3: PSL CIM Model. Further definition of PSL control levels can be found in the PSL CRS.

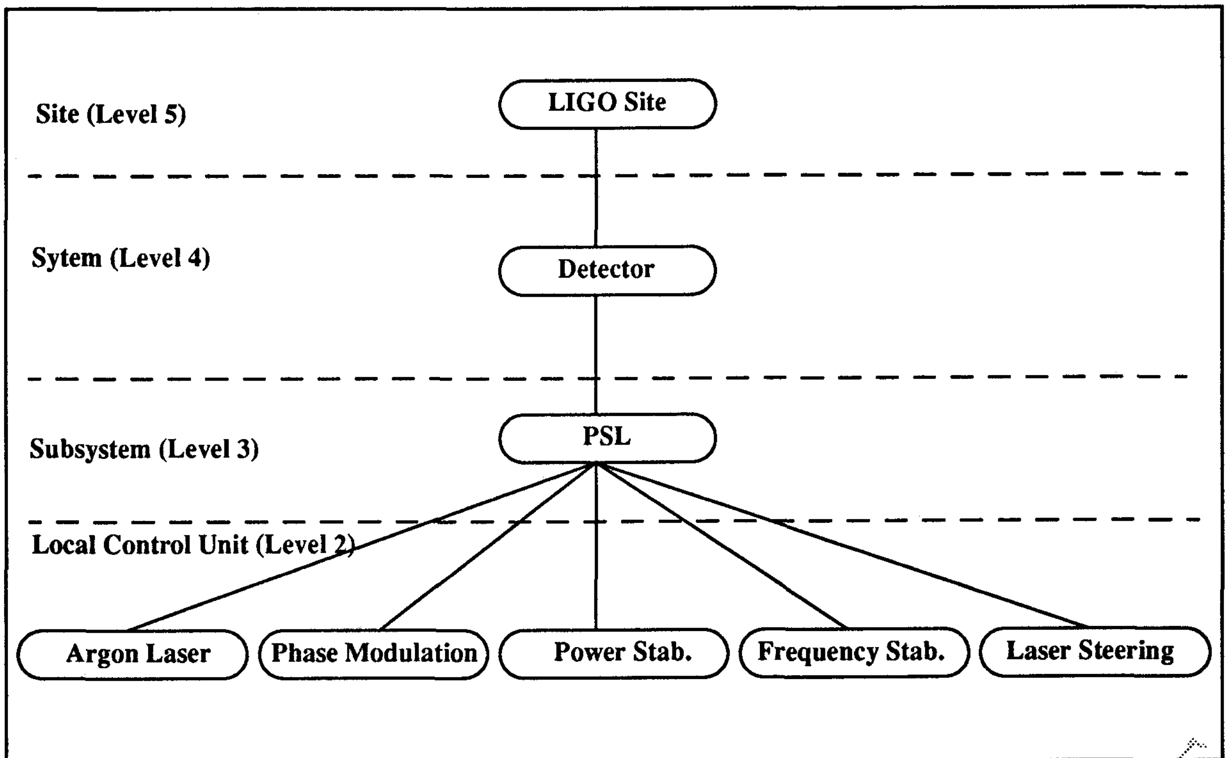

Figure 3: PSL CIM Model

During this phase, prototype software will also be developed and tested, with emphasis on those components which are determined to have a higher risk. This software will be component tested off-line by the developer, and then transferred to the PSL system in the optics lab for initial system testing.

At the end of this phase, all design issues should have been resolved and a complete design presented at a formal PDR.

4.1.3. Final Design Phase

In this phase, any outstanding design issues from the PDR are resolved, and final code implementation begins. A "Release 0" software package should be completed prior to a FDR at the end of this phase. "Release 0" is defined as that which meets the minimum requirements to begin installation and commissioning of LIGO. This distinction is made because the PSL software will undergo integration with other LIGO systems after the FDR, and certain changes and/or enhancements may be added during that time.

"Release 0" shall contain:

1. Code necessary to allow commissioning of the PSL system.
2. Supporting documentation of successful completion of testing defined in the STP.
3. The first set of software system documentation.

4.1.4. Integration and Commissioning Phase

These areas are outside of the scope of PSL construction. These phases are described in CDS document TBD.

4.2. Software Development Methodology

4.2.1. Software Standards

This software will be developed in accordance with the standards of the LIGO CDS group for all software development projects, as outlined in LIGO document TBD.

4.2.2. Design and Development Tools and Techniques

4.2.2.1 CASE and Development Tools

The PSL software will be designed and developed based on the present capabilities of the Experimental Physics and Industrial Control System (EPICS) software package, distributed by Los Alamos National Laboratory (LANL). For this project, EPICS will be used "as is" for the bulk of the activities, except to address those issues where use of EPICS is analyzed as not being able to meet particular PSL requirements.

Particular tools to be employed from the EPICS toolkit are:

1. Microsoft Access for EPICS database generation. An interface has been developed at LIGO to allow EPICS database generation with the commercial Microsoft Access product. This will be used for initial development. Later, Oracle RDMS will replace Access, as this product will be the LIGO standard.

2. The Motif EPICS Display Manager (MEDM) will be used for operator display development.

3. The ARchiver (AR) tool will be used to setup/initiate data archival; the tcl/tk ARchive Retrieval (ARR) tool will be used to extract and display archived data.

4. EPICS State Notation Language (SNL) will be used to develop real-time sequencing software.

5. The ALarm Handler (ALH) software will be used to develop alarm processing structures, along with the monitoring, reporting and archiving of PSL alarms.

4.2.2.2 Programming Languages

In those cases were EPICS will not meet a requirement, or other new software needs to be developed, the 'C' language will be used, designed with standard software design tools.

4.2.2.3 Documentation Tools

All documentation will be produced with the LIGO standard publishing packages. When approved, this documentation will be reproduced on the World Wide Web.

4.2.3. Target Systems

Target systems are defined as those hardware platforms and software systems on top of which the CDS software will operate.

4.2.3.1 Operating Systems

Two computer operating systems will be employed in this project:

1. Sun Unix 4.1.3 for code development and Sun workstation targets. It is intended that this be upgraded to Sun Solaris 2 once all development tools have been ported and tested by the EPICS collaboration.
2. VxWorks real-time operating system for the Motorola 68040 processor targets.

4.2.3.2 Hardware Resources

4.2.3.2.1 Development Hardware

The software development hardware will be Sun workstations.

4.2.3.2.2 Target Hardware

The project involves two types of target hardware:

1. Sun workstations for operator and file services.
2. VME-based Motorola 68040 processors for real-time control applications.

4.2.4. Security

Routine log-in procedures requiring user identification and password provides access control to CDS software areas. These areas will be open to all LIGO personnel for data "read", but will be write accessable only by the CDS group members. At minimum, weekly backups of all files will be maintained in order to prevent catastrophic loss of data.

4.2.5. Code Development Cycle

As software progresses through its lifecycle, it moves into different areas as shown previously in the Configuration Control section of this document. The contents of these areas and procedures by which it moves are outlined in the following sections.

4.2.5.1 Software Engineer's Work (Developer's) Area

The software engineer's work area is established prior to the beginning of software development. This area contains the current in progress code and documentation files. For the PSL project, this area will reside on the CDS group server (kater) under "/home/CDS/a" (the first of two 1GByte external disk drives on this server). The directory structure will be as shown in the following figure and described in the following subparagraphs. These basic structure will be continued as the software moves through the various areas of software development.

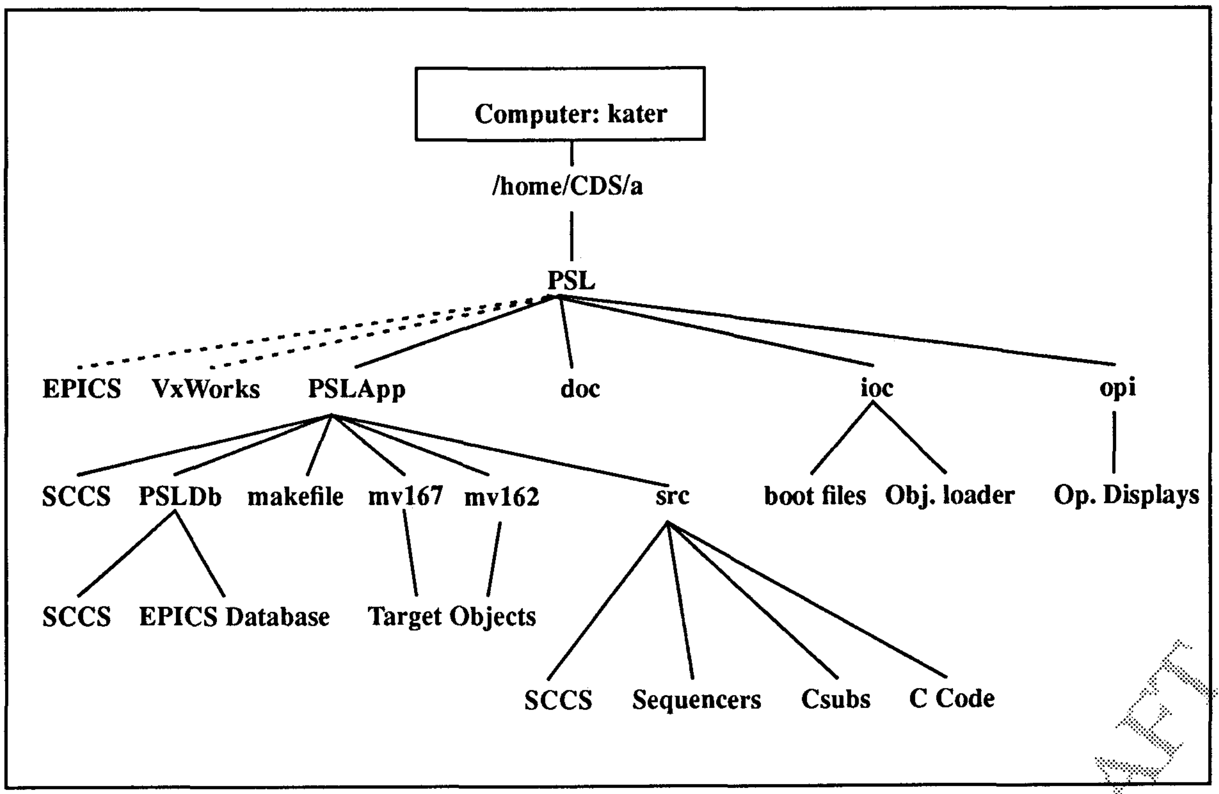

Figure 4: Development Area Directory Structure

4.2.5.1.1 EPICS and VxWorks

The dotted lines in the previous diagram denote that a copy of EPICS and VxWorks will not reside the the developer's area. To ensure that development is always with the latest and common version of this software, this area only contains symbolic software links back to the EPICS and VxWorks versions which reside in the LIGO Release Area.

4.2.5.1.2 PSLApp

This is the primary application development directory. It contains:

1. The SCCS directories.

2. PSLDb: All EPICS databases being developed.

3. A makefile which contains the computer instructions for compiling all application software within the development area.

4. Target directories for the real-time code objects generated by the makefile. These are mv167 and mv162 (Motorola VME model 167 and model 162), the two processor units presently supported by the CDS group.

5. src: Contains all SNL (*.snl) application files, and all C code, developed either into the EPICS database as CSUB records or stand-alone code.

6. doc: All documentation pertaining to this development project.

7. ioc: Subdirectories by development test target VME processor names. This area contains all of the startup and booting sequences required by development real-time processors.

8. opi: contains all operator displays developed for the project.

4.2.5.2 Software Prototype/Test Release Area

Once software is released by the developer for integrated testing, it is placed into SCCS and moved to the Prototype/Test Release Area. This area will mirror the developer's area directory structure, but be located on "kater" at "/home/CDS/b" (the second server disk drive). From here, code will be loaded to the TBD workstation and VME systems used to test the PSL in the optics lab and 40M Lab. The primary change in data in this directory will be new entries in the "ioc" subdirectory to cause the real-time software to load into the prototype/test target processors instead of the development area processors.

Once software in this area has completed successful testing, and endorsed by at the PSL FDR, it will be moved to the SDL area. To first order, at this point the PSL project described in this document is complete.

4.2.6. Documentation

Project documentation include those items previously described within this document, along with the documentation called for in CDS document TBD.