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

Document Type	**LIGO-T050056-00-Z**	2 April 2005

PSU Software Development Process

L. S. Finn

Distribution of this draft:

LIGO Scientific Collaboration

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu/

Processed with LaTeX on 2005/04/03

Abstract

This note establishes the PSU group's three phase development process for software that contributes to the analysis and interpretation of LIGO data. The three phases are requirements definition and analysis, design, and implementation and testing. Entry and exit criteria are described for each phase. The goal of this process is well-documented, well-tested, and easily maintained software that satisfies a clearly-defined need. This goal shall inform every step in the implementation of the procedures described here.

$Id: T050056.tex,v 1.1 2005/04/03 23:17:19 lsf Exp $

Contents

1 Introduction

This note establishes the PSU group's development process for software that contributes to the analysis and interpretation of LIGO data. The goal of this process is well-documented, well-tested, and easily maintained software that satisfies a clearly-defined need. This goal should inform every step in the implementation of the procedures described here.

The process described here has three phases:

- requirements definition and analysis,
- design,
- implementation and testing.

The goal of the requirements definition phase is to capture the need that the software is to fill. It should be sufficiently detailed that it can be clearly determined whether a given software product meets the need. The precise nature of the requirements depends on the need. The requirements shall include a clear statement of goal and may include, as appropriate, precision, accuracy, performance, inter-operability, or other requirements. The requirements for a given software system are captured in a Software System Requirements (SSR) document.

The goal of the design phase is to produce a software design that meets the requirements captured in the SSR. The design phase has two sub-phases: a preliminary design phase and a detailed design phase. In the preliminary design a software architecture is identified that will meet the system specifications. All major subsystems are identified and the requirements described in the SSR are mapped onto the subsystems. All internal and external interfaces are defined to the subsystem level. The preliminary design is captured in a Preliminary Design Report (PDR).

The second design sub-phase is the detailed design. In the detailed design the preliminary design is elaborated, through successive refinements, down to the unit level. Unit functions are specified at the "code-to" level. The detailed design is captured in a Detailed Design Report (DDR). The DDR should describe in text the elaboration of the PDR. The detailed specification at the code-to level may be captured in "comments" that will form the head of every units actual code.

Implementation and Testing is the final phase of the development process. Implementation and Testing includes coding of the detailed described in the DDR, unit testing, integration and system testing against the requirements set-forth in the SSR.

Each of the three phases of software development has exit criteria: i.e., a phase is not complete until the exit criteria for that phase are met. The exit criteria for one phase are the entry criteria for the next phase; so, while phases may overlap in parts, no phase can be fully underway until the previous phase has been completed and no phase can be completed before the previous phase has ended. For example, the preliminary design cannot be completed before the system requirements are accepted, etc.

The burden of the process described here should be no larger than required to meet the goal of well-documented, tested and easily maintained software that satisfies a clearly-defined need. In particular, the SRD, PDR, and DDR documents called for need not be long nor formal; however, they should be consistent, clear and understandable to an informed reader not involved in the project, and complete enough to allow testing of the software at the unit and system level and act as a guide to the overall software design and the function within that overall design of each software unit.

The rest of this document describes the entry and exit criteria of each of the software development phases.

2 Requirements

The goal of the requirements definition phase is to capture the need that the software is to fill. It should be sufficiently detailed that it can be clearly determined whether a given software product meets the need. The precise nature of the requirements depends on the need. The requirements shall include a clear statement of goal and may include, as appropriate, precision, accuracy, performance, inter-operability, or other requirements. The requirements for a given software system are captured in a Software System Requirements (SSR) document.

To exit the Requirements phase the SSR must be accepted. To be accepted the SSR shall be delivered as a document to an external person or persons, who shall review it for clarity, consistency, and completeness. In particular, is the statement of the goal clear? Are the requirements consistent with that goal and will a software product that meets those requirements satisfy the goal? Are the requirements testable: i.e., can the software product be compared to the requirements and clearly and cleanly determined to have met the requirements?

The name of the external party of parties who have reviewed and accepted the SSR shall be recorded in the SSR. The SSR shall be kept under version control with the software product. The first accepted version of the SSR shall be tagged as such.

3 Design

The goal of the design phase is to produce a software design that meets the requirements captured in the SSR. The design phase has two sub-phases: a preliminary design phase and a detailed design phase. In the preliminary design a software architecture is identified that will meet the system specifications. All major subsystems are identified and the requirements described in the SSR are mapped onto the subsystems. All internal and external interfaces are defined to the subsystem level. The preliminary design is captured in a Preliminary Design Report (PDR).

The second design sub-phase is the detailed design. In the detailed design the preliminary design is elaborated, through successive refinements, down to the unit level. Unit functions are specified at the "code-to" level. The detailed design is captured in a Detailed Design Report (DDR).

The entry criteria for the design phase is a complete and accepted SSR.

The PDR is a major milestone of the design phase. To pass this milestone the PDR and SSR are required to have been delivered as a document to an external person or persons, who shall review the PDR for clarity, consistency, and completeness. In particular, does the PDR clearly describe how the requirements map onto the design? Does the PDR describe a design that addresses the requirements described in the SSR? Does the PDR describe all the major subsystems and their internal and external interfaces?

The name of the external party of parties who have reviewed and accepted the PDR shall be recorded in the PDR. The PDR shall be kept under version control with the software product. The first accepted version of the PDR shall be tagged as such.

Following completion of the PDR milestone the detailed design can begin. The DDR is the main product of this sub-phase. The exit criteria of the design phase is an accepted DDR. To be accepted the DDR and the PDR from which it is derived shall be delivered as a document to an external person or persons, who shall review it for clarity, consistency, and completeness. In particular, is the elaboration of the preliminary design to the unit level clear? Is the detailed design consistent with the preliminary design? Does the DDR describe a complete elaboration of the preliminary design to the unit level?

The name of the external party of parties who have reviewed and accepted the DDR shall be recorded in the DDR. The DDR shall be kept under version control with the software product. The first accepted version of the DDR shall be tagged as such.

4 Implementation and Testing

Implementation and Testing is the third and final phase of the development process. Implementation and Testing includes coding of the detailed described in the DDR, unit testing, integration and system testing against the requirements set-forth in the SSR.

The entry criteria for Implementation and Testing is an accepted DDR.

The exit criteria for Implementation and Testing is the Acceptance Review. The Acceptance Review is a more elaborate and formal "gate" then the reviews that mark the end of the Requirements or Design phases. The Acceptance Review is a structured walkthrough of the software to insure that it

- is consistent with the design described in the PDR and DDR;
- conforms to the relevant coding standards, guidelines, and rules;
- meets the requirements set-out in the SSR; and
- has been adequately tested and is ready for release.

The Acceptance Review may require several meetings.

The rest of this section describes the Acceptance Review process in detail.

4.1 Participants

The Acceptance Review involves one or more formal meetings. Each participant in the meeting plays a specific role and has a specific responsibility. The participants and their responsibilities are

- Author: prepares and distributes copies of the code, documentation (including functional requirements, etc.), and any supporting material (e.g., flow-charts) to moderator and reviewers in advance of walkthrough. Presents the s/w being reviewed.
- Moderator: facilitates the walkthrough, ensures that the agenda is followed, encourages participation of all reviewers. The moderator may also be the scribe.
- Reviewers: evaluate the work to determine if it is technically accurate, assess whether project guidelines and standards are being followed, requirements met, and the s/w is properly prepared.
- Scribe: takes notes during walkthrough, opens tickets on defects and records other technical comments, suggestions and unresolved questions. The scribe and the moderator can be one and the same.

4.2 Before the Acceptance Review

Before the first meeting of the review panel the participants shall prepare for the review in the following ways:

- Author:
  - Determine if the size of the work product is appropriate for one walkthrough session. The duration of a walkthrough should not exceed 2 hours. If more time is necessary the work product should be divided into smaller portions and each portion reviewed separately.
  - Select reviewers (at least two) who are appropriate for the product. Reviewers should not be authors of the s/w under review.
  - Select the moderator and the scribe.

  - Schedule the meeting date, time and location and notify all participants.
  - Provide reviewers, moderator and scribe with the SSR, PDR, DDR and copies of all software to be reviewed at least 2 days prior to the walkthrough.

- Reviewers:
  - Review the material provided by the author;
  - Identify technical and non-technical (e.g., spelling) errors, note questions. Non-technical errors are not discussed at the walkthrough but should be provided to the scribe and author at the conclusion of the walkthrough.

- Moderator and Scribe:
  - Review the materials provided by the author;
  - Identify and record non-technical errors and provide to the author at the conclusion of the walkthrough.

4.3 During the Acceptance Review Meeting(s)

The Acceptance Review shall follow the process described below:

- The Moderator calls the Acceptance Review to order, reminds the participants of their roles in the review and the review purpose and ground-rules, and summarizes the goals for this review committee meeting.
  - The review's purpose is to identify inconsistencies between the coding and the design, or errors in the software, and the discussion in the walkthrough is limited to this purpose. The discussion of solutions or of "better" designs is not part of the walkthrough process. There should be no more than about 1 minute discussion per issue.
  - Each meeting of the review committee is limited to 2 hours. If the review exceeds 2 hours the moderator shall stop the review at a logical breaking point and schedule another meeting to continue the review

- The Author addresses issues that remain from a previous meeting of the review panel.

- The Author walks through the design at the level of the PDR. The purpose of this walkthrough is informational: this is not a review of the PDR. Discussion shall be limited to questions of clarification (unless a design error or misconception is identified, in which case the review is halted).

- The Author walks through each major subsystem at the unit level, reviewing the software. This walkthrough should be at the detailed level of Program Design Language (PDL) and associated code: e.g., describe in blocks of approximately a dozen lines of code what the code is supposed to do. Each block should have PDL and comments associated with it.

- The reviewers, moderator and scribe are to evaluate the work to

  - determine if it is technically correct;
  - assess whether the software reflects the PDR and DDR;
  - assess whether project guidelines and standards have been followed.

- The scribe is to record technical errors identified in the review.
- At the conclusion of the session the Moderator asks the reviewers to agree on a conclusion regarding the material reviewed. There are three conclusions that can be reached on each sub-system:
  - Accept as-is;
  - Accept with modifications;
  - Send-back for revisions, followed by re-review.
- The Moderator adjourns the walkthrough session at the scheduled time. If the agenda has not been completed a follow-up session is scheduled by the Author.

4.4 After each review committee meeting

After each review committee meeting the Scribe shall

- Record technical errors identified by reviewers in defect tracking system. Defects should be separated into logical groups and a separate ticket assigned to each group.
- Record non-technical errors in defect tracking system. A single ticket per document or program file will suffice for non-technical errors;
- Collect record of non-technical errors found by reviewers, moderator and see that these are provided to the author;
- Record suggested action items and other follow-up activities.
- Prepare the meeting record (day, time, participants, outcomes). The record shall include any action items identified by the reviewers and ticket numbers associated with identified defects.
- Circulate the meeting record to the participants for their review and comment;
- Update the meeting record as needed and distribute the revised meeting record to the author and moderator.

References

[1] Karl E. Wieger. 2002. “Peer Reviews in Software: A Practical Guide” Boston: Addison-Wesley.

[2] Office of the Chief Information Officer. 2002. ”Structured Walkthrough (SWT) Process Guide” Version 3. ¡http://cio.doe.gov/ITReform/sqse/publications.htm¿ , 30 March 2003.

[3] Software Engineering Laboratory, NASA Goddard Space Flight Center. 1992 “Recommended Approach to Software Development.” Revision 3. Goddard Space Flight Center SEL-81-305.