LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO-M1000211-v3 *LIGO* 9-Sep-2014

Subsystem-Level and System-Level Testing Requirements

Dennis Coyne, David Shoemaker

Distribution of this document:
LIGO Scientific Collaboration

This is an internal working note
of the LIGO Laboratory.

California Institute of Technology
LIGO Project – MS 18-34
1200 E. California Blvd.
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – NW22-295
185 Albany St
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
P.O. Box 159
Richland WA 99352
Phone 509-372-8106
Fax 509-372-8137

LIGO Livingston Observatory
P.O. Box 940
Livingston, LA 70754
Phone 225-686-3100
Fax 225-686-7189

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

1 Introduction

Verification of the Advanced LIGO system to meet requirements is to be performed by functional and performance testing. The general sequence of testing, both prior to and during the integration phase, is described. The testing requirements and associated documentation requirements are also defined.

2 General Testing Requirements

1) Test Procedures:

   a. A test procedure must be written, and archived, for each test which is performed on every production (delivered) component/device, module/unit, subassembly and assembly or subsystem. These test procedures should be E-documents and must be filed in the LIGO Document Control Center (DCC).

   b. The test procedures should be complete step by step procedures (relying on referenced documents in needed) and have definitive pass/fail criteria defined. The pass/fail criteria should be in the form of acceptable values ranges (not just a nominal value).

   c. If a test procedure changes, then the same document number should be used and the version number incremented (-v1, -v2, -v3, …). This allows test results to be keyed to the specific version of the test procedure.

   d. The test procedure should be automated in whatever measure is productive, and it should be easy to store data and require a minimum of copying of numbers from screen to screen. For example use of EPICS test scripts and use of the Python scripts for automated electronics testing .

2) Test Reports:

   a. Test reports for every test procedure performed on every production (delivered) component/device, module/unit, subassembly and assembly or subsystem, must be archived into the DCC; These test reports should be E-documents, filed as Adobe AcroBat (*.pdf) files with the source document also filed (e.g. *.doc, *.tex, etc.)

   b. The test reports should be concise, but should indicate whether the unit has passed or failed. Do not include the test procedure or instructions within the test report; Cite the test procedure, including its version number (-vN).

   c. The test report(s) should have section or subsection numbers which are constant from unit to unit and which are cross-referenced to the test procedure.

   d. All of the test reports shall be hyper-linked together using the DCC's "related document" capability using the schema described in section 3 below.

   e. Acceptance of the test report should be indicated by DCC electronic approval by the subsystem leader and/or the leader's designee and by the System Engineer or the System Engineer's designee.

   f. If a unit fails, it is re-worked and re-tested, then the same DCC number should be used for the subsequent test report(s) by incrementing the version (-v1, -v2, -v3, …)

g. All electronics must have their documentation, by serial number, follow T0900520, "Documenting Electronics using the DCC and the E-traveler"

3) Test Results:

a. The test data are given in the Test Reports either as scalar numerical data (for example in tables or matrices) or plotted as vector data in the form of plots in format. Test data (or at least large sets of vector test data such as transfer functions) must also be archived in a native data format (*.xml, *.xls, *.txt, etc.) with sufficient information (e.g. calibration data, channel/signal names, filter coefficient sets, etc.) to enable subsequent use. For example, the electronics automated test results collected by the LIGO Python scripted test program should be archived for all circuit boards; the calibration response data for every ISI position sensor should be archived; the transfer functions for every ISI actuator to every ISI sensor should be archived, etc.

b. Test Results can be archived either using the DCC or the aLIGO Project Test Results SVN Archive (see section 3). It is not sufficient to archive these results in another archival system (such as the commissioning ilog/elog, a subsystem group's elog, one's home page, etc.) and then provide pointers or references to these other data storage systems. See section 3.

4) Pre-Integration Phase Testing:

a. All functions should be tested or demonstrated in the assembled condition, prior to installation/integration. However the following exceptions are permissible:

  - those functions which require integration into the system to demonstrate, (e.g. offloading from one system to another) if it is too burdensome to simulate the other subsystem(s);

  - functions which derive or follow from other demonstrated functions, can be shown to work in one instance of a major assembly or subsystem and not every one (but only for the pre-installed condition), e.g. isolation performance for the SEI ISI systems

b. All devices must be tested or demonstrated to be functional in the assembled condition. For example, it is not sufficient to show that one sensor works and argue that by similarity all others must also work. Likewise it is not sufficient to have demonstrated that a sensor worked in a stand-alone condition without also demonstrating that it works properly as installed including installed cabling.

c. If some aspects of performance or functional testing are compromised by the pre-installation condition (e.g. in air vs in vacuum) then those aspects of the testing can be deferred until in-situ testing, or accepted at a less than full performance measure, e.g. electro-static actuation, isolation performance at high frequency, etc. However, one wants a set of measures that can be shown to be consistent with, and 'promising' of performance per specifications. So, in addition to the test plan (which should be step-wise and have little flexibility), a backup analysis is needed that shows how success in this testing leads to confidence that the system can perform to its full specifications.

5) Integration Phase Testing:

a. All functions must be tested or demonstrated in the installed/integrated condition.

b. All devices must be tested or demonstrated to be functional in the installed, condition.

3 Archival Schema

3.1 DCC

The LIGO Document Control Center (DCC)[1] is the official archive for all LIGO project documentation. The hyperlinking, "related documents" feature of the DCC must be used to associate all relevant test data together in a tree as depicted in Figure 1. The root entry for this tree is a DCC entry entitled "aLIGO <subsystem name> Testing and Commissioning Documentation". From this DCC entry all test procedures and test reports for the subsystem can be reached. Note that this DCC entry is likely to not have any files directly associated with it (uploaded under this DCC number); it serves as an index, or list of hyperlinks.

The next level in the branched, hierarchical structure consists of DCC entries for the major subsystem assembly types for the subsystem. As an example for the Seismic Isolation (SEI) subsystem, the major assembly types are HAM-ISI, BSC-ISI, HAM-HEPI and BSC-HEPI. For the Suspension (SUS) subsystem, the major assembly types would be each suspension type (ETM/ITM Quad suspension, BS/FM triple suspension, HLTS and HSTS). This DCC entry is also likely to not have any files directly associated with it (uploaded under this DCC number); it serves as an index, or list of hyperlinks.

Tests for each component/device, module/unit, or subassembly of a single type should be collected under separate DCC entries as well. This DCC entry is likewise simply an index or list of hyperlinks to the documentation for each component/device, module/unit, or subassembly.

The DCC entries for the specific major subassembly tests, on a single assembly unit (instance), should have the test report filed as an Adobe Acrobat (*.pdf) file. In addition the source file(s) for the report (e.g. *.doc, *.tex, etc.) should be filed as an "other file" associated with the test report. Additional files (such as the raw data, associated spreadsheet files, Matlab files, etc.) should also be files as "other files" if/as appropriate.

"Related Document" links for each major subassembly test entry in the DCC should be provided for each specific serial numbered units which comprise the assembly (i.e. tests of component/device, module/unit, or subassemblies used to build the major assembly).

For a specific (and evolving) example, see the Seismic Isolation (SEI) test results entry in the DCC, E1000304-x0, which (at the time of writing this report) has the tree structure shown in Table 1. See also a summary presentation of the first HAM-ISI testing in G1000770-v1.

[1] See https://dcc.ligo.org/wiki/index.php/Getting_DocDB_Help

Table 1 SEI Subsystem Test Documentation Structure in the DCC

LIGO-E1000304: aLIGO SEI Testing and Commissioning Documentation

- LIGO-E1000305: aLIGO HAM-ISI Testing and Commissioning Documentation
    - LIGO-E1000341: HAM-ISI LHO test stand: software, electronic checks, and user guide
    - LIGO-E1000300: HAM-ISI LLO test stand: software, electronic checks, and user guide
    - LIGO-E1000309: aLIGO HAM-ISI, Pre-integration Testing Procedure - Phase I (post-assembly, before storage)
    - LIGO-E1000310: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LHO Unit #1
    - LIGO-E1000311: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LHO Unit #2
    - LIGO-E1000312: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LHO Unit #3
    - LIGO-E1000313: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LHO Unit #4
    - LIGO-E1000314: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LHO Unit #5
    - LIGO-E1000323: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LHO Unit #6
    - LIGO-E1000324: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LHO Unit #7
    - LIGO-E1000325: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LLO Unit #1
    - LIGO-E1000326: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LLO Unit #2
    - LIGO-E1000327: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LLO Unit #3
    - LIGO-E1000328: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LLO Unit #4
    - LIGO-E1000329: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LLO Unit #5
    - LIGO-E1000330: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LLO Unit #6
    - LIGO-E1000331: aLIGO HAM-ISI, Pre-integration Test Report, Phase I, LLO Unit #7
- LIGO-E1000306: aLIGO BSC-ISI Testing and Commissioning Documentation
- LIGO-E1000307: aLIGO HAM-HEPI Testing and Commissioning Documentation
- LIGO-E1000308: aLIGO BSC-HEPI Testing and Commissioning Documentation

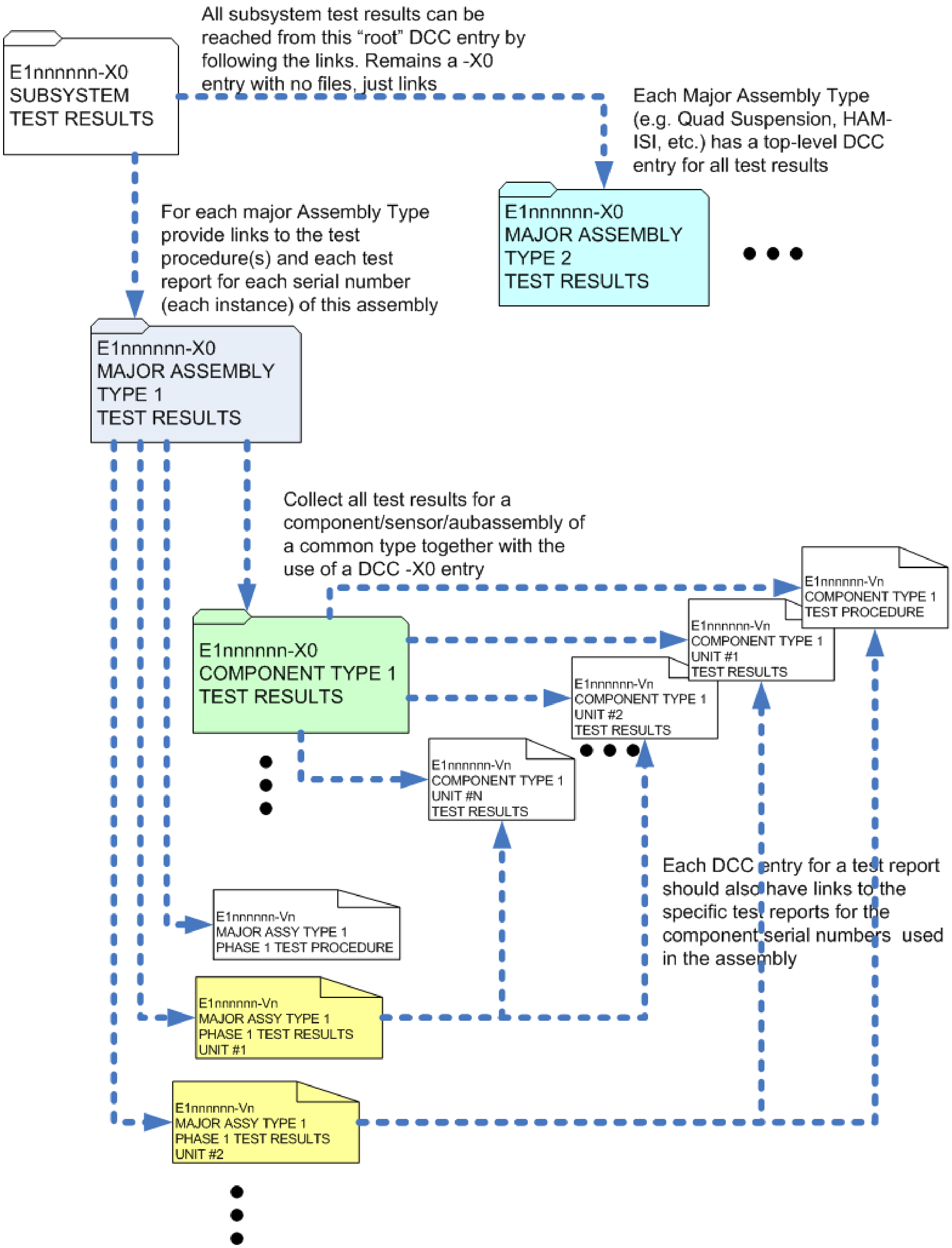

Figure 1 Schema for Organizing Test Data within the DCC

3.2 SVN

The test data in original or native format is to be archived. The preferred approach is to archive the data in an SVN[2] test data repository[3]. The intent of using SVN rather than filing the data sets as “other files” associated with the test reports in the DCC is to provide a richer and more flexible directory structure, more flexible tracking and rollback features, etc. However, for subsystems with relatively small and simple data sets, the benefits of using SVN may not outweigh the overhead associated with learning SVN. In these cases, the test data may be filed as “other file(s)” in the DCC entry associated with the test report.

The organization of the data sets (by directory, branches or tags) should follow the structure indicated in Table 1. Note that each subsystem has its own SVN with subsystem provided code for generating control laws and filters. The infrastructure, or core control software, are in seprate SVNs.

Table 2 Example Subsystem SVN Organization

```
/
    <major assembly or subsystem 1, e.g. BSC-ISI>
        <Common>
        <H1>
        <H2>
        <L1>
            <instance 1, e.g. BS>
            <instance 2, e.g. ETMX>
                <Data>
                            <Figures>
                            <Spectra>
                            <TransferFunctions>
                            Etc …
                <FilterDesign>
                <Filters>
                <Medm>
                <Scripts>
                <SensorDefinitions>
                <Misc>
                Etc …
            <instance 3>
            Etc …
```

4 Testing Prior to Integration Phase

The nominal sequence of testing in the phase prior to installation and integration is depicted in Figure 2. Each potential test is described in the following subsections.

---

[2] B. Collins-Sussman, B. Fitzpatrick, C. Pilato, “Version Control with Subversion: For Subversion 1.5: (Compiled from r3305)”, http://svnbook.red-bean.com/en/1.5/svn-book.pdf

[3] The intent is to have a single project SVN repository accessible from anywhere and by any member of the LSC (perhaps LVC). Most SVN users would have read-only access.

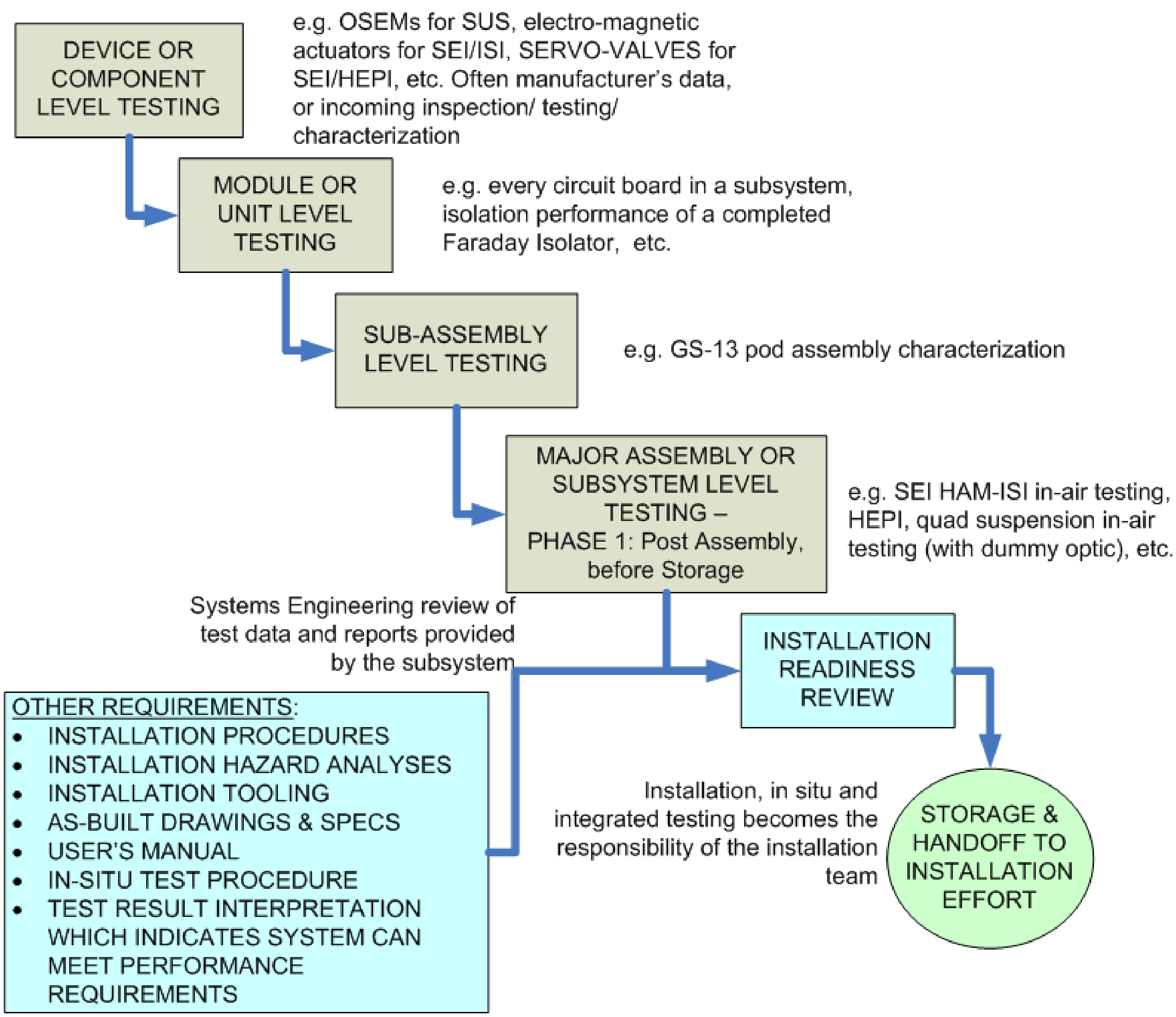

Figure 2: Pre-Installation/Integration Phase Testing

4.1 Device or Component Level Testing

Test reports and test data on devices or components (either performed by LIGO Laboratory or by the supplier) must also be archived. Often this test data is collected as part of the acceptance testing program prior to assembly.

4.2 Module, Unit or Sub-Assembly Level Testing

In some cases testing is performed on modules or sub-assemblies prior to final assembly. Examples include electronics modules.

4.3 Major Assembly or Subsystem Level Testing – Phase 1

A first step for each subsystem is to determine the appropriate 'handoff' unit for transfer to the installation/integration group. This will vary between subsystems, but is generally the highest level of assembly formed prior to installation. The consideration should be what is testable and represents a reasonable ‘chunk’ to move out of the subsystem responsibility and to the integration task.

Phase 1 testing is performed post-assembly and before storage.

4.4 Installation Readiness Review

The installation readiness review of the first, and perhaps second, instance of a major assembly or subsystem element (e.g. SEI HAM-ISI) will be a formal review with a committee reviewing the results. Subsequent reviews will be simply a sign-off by the Systems Engineer.

4.5 Handoff to Installation/Integration Effort

Once the major assembly, or subsystem, is deemed ready for installation, the responsibility for the equipment, software, documentation, and maintenance is transferred to the installation/integration group. Of course many/most of the personnel working on this subsystem continue to work on the subsystem in the installation/integration phase, but now under the direction and coordination of the installation/integration group.

5 Integration Phase Testing

The nominal sequence of testing in the phase prior to installation and integration is depicted in Figure 3. Each potential test is described in the following subsections.

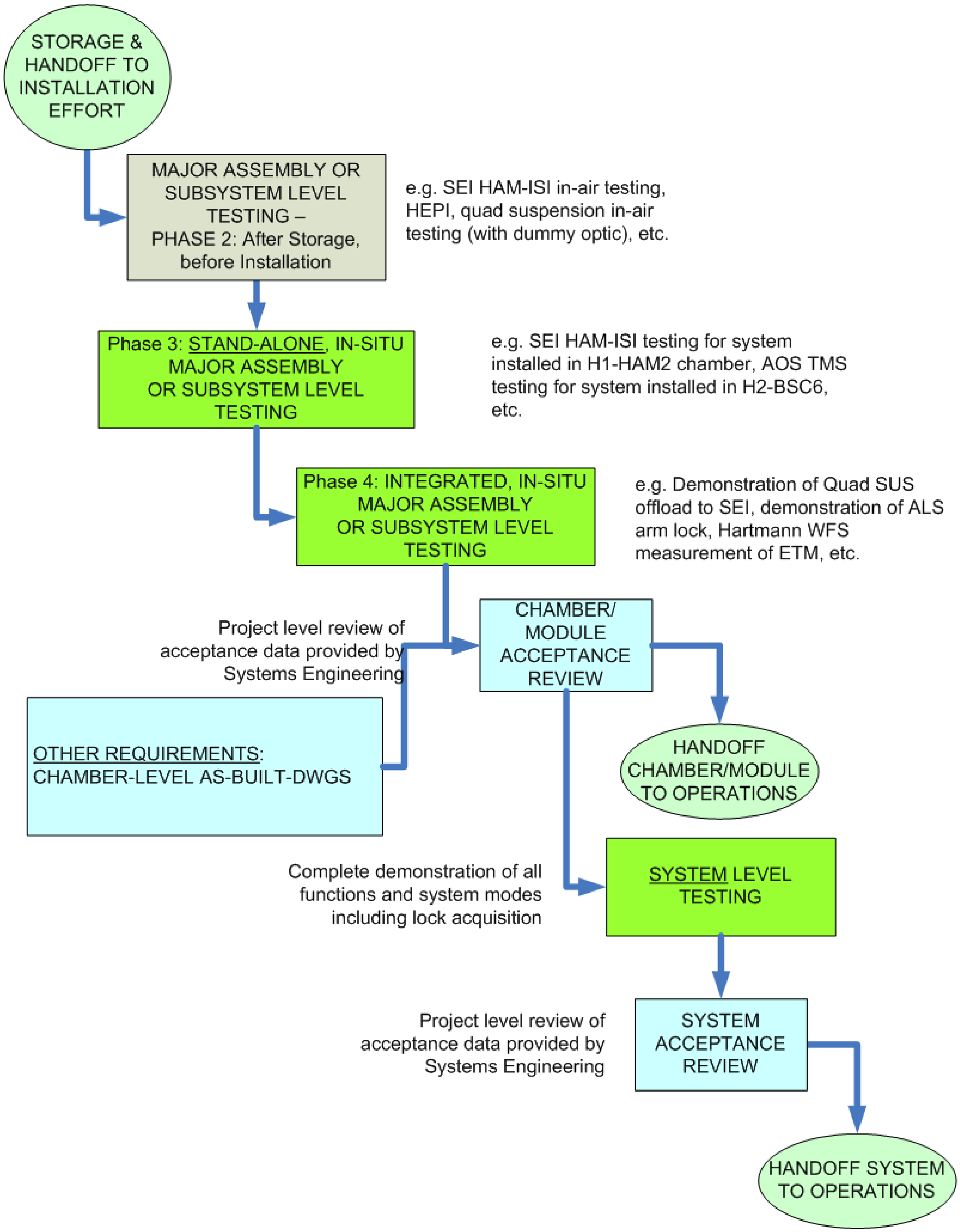

Figure 3 Integration Phase Testing

5.1 Major Assembly or Subsystem Level Testing – Phases 2, 3 and 4

Phase 2 testing is performed after storage and just prior to installation. This is an abbreviated set of tests to assure that nothing was damaged during storage and transport and the assembly is ready for installation.

Phase 3 testing involves stand-alone, in-situ testing to see that the assembly behaves properly after the installation procedure and with the field-installed cabling, etc.

Finally phase 4 testing is of the assembly fully integrated. Testing is limited to inter-operability functions such as off-load commanding, isolation performance in the presence of payload dynamics, etc.

5.2 Chamber/Module Acceptance Review

A first step for each subsystem is to determine the appropriate 'handoff' unit from integration to operations. This will vary between subsystems. Some examples of potentially useful units:

- The equipment in one vacuum chamber (e.g., SEI ISI, the SUS in a chamber, the AOS in a chamber); with its associated support equipment (cabling, electronics, software)
- Another natural unit like the PSL, or the IO equipment external to the vacuum
- One subsystem's infrastructure element -- e.g., the VacEq modifications, or the DAQ

The consideration should be what is testable and represents a reasonable 'chunk' to move out of the subsystem responsibility and to the integration task. The Systems Engineering group will work with the subsystem and commissioning teams determine the appropriate acceptance packages.

5.3 System Level Testing

The plans and objectives for testing and integration of the aLIGO detector system are defined in T1200437, "Advanced LIGO Interferometer Integration". In particular this document defines the five interferometer sub-configurations that are planned as intermediate steps to full system integration:

- Single resonant arm cavity, or "one arm test" (T1100080)
- Resonant Input Mode Cleaner cavity (T1100201)
- Dual Recycled Michelson Interferometer (DRMI) (E1300631)
- Arm Cavity with Power Recycling Cavity (PRC), or Half-Interferometer (HIFO) (T1300174)
- Full Arm Length Stabilization (both arms and PRC) or HIFO-XY
- Full Interferometer (T1200437)

In the list above, the document references are to the test plans and objectives for each of these integrated systems tests.