NILE BASIN INITIATIVE
Initiative du Bassin du Nil

DHI

Project Handbook

Appendix A

NBI DSS Project

Project Handbook

Version 2.0

Revision and Signoff Sheet

Table 1 Change record

Date	Version	Description	Author

Table 2 Reviewers

Name	Version Approved	Position	Date
Niels K. Olsen	2.0	Software Quality Assurance Manager	<01/12/09>

Table of Contents

1. Introduction

The Project Handbook describes DHI’s internal work processes and project organization in software development projects. As such the Project Handbook is not written for a particular project or client. The Project Handbook does not assume any particular software development model (e.g. agile or waterfall), but may be adopted to any model.

It describes the project in terms of:

- Project organization – i.e. a description of the project in terms of roles, responsibilities and lines of references.
- Project life cycle – i.e. a description of the project in terms of phases and deliverables. The document deliverables falls in 2 categories:
    1. Documents identified as deliverables in the contract between the client and DHI. Hereafter referred to as external documents.
    2. The documents developed in order to internally manage the project. Hereafter called internal documents.

In the appendices A to G the handbook defines how the project works with:

A. Requirements and Incidents

B. Project Planning

C. Software Project Tracking and Oversight

D. Software Quality assurance

E. Configuration Management

F. Document Naming Conventions

G. Coding Practices and Standard

2. Overview

The diagram below (Figure 1) illustrates how DHI typically organizes a software development project.

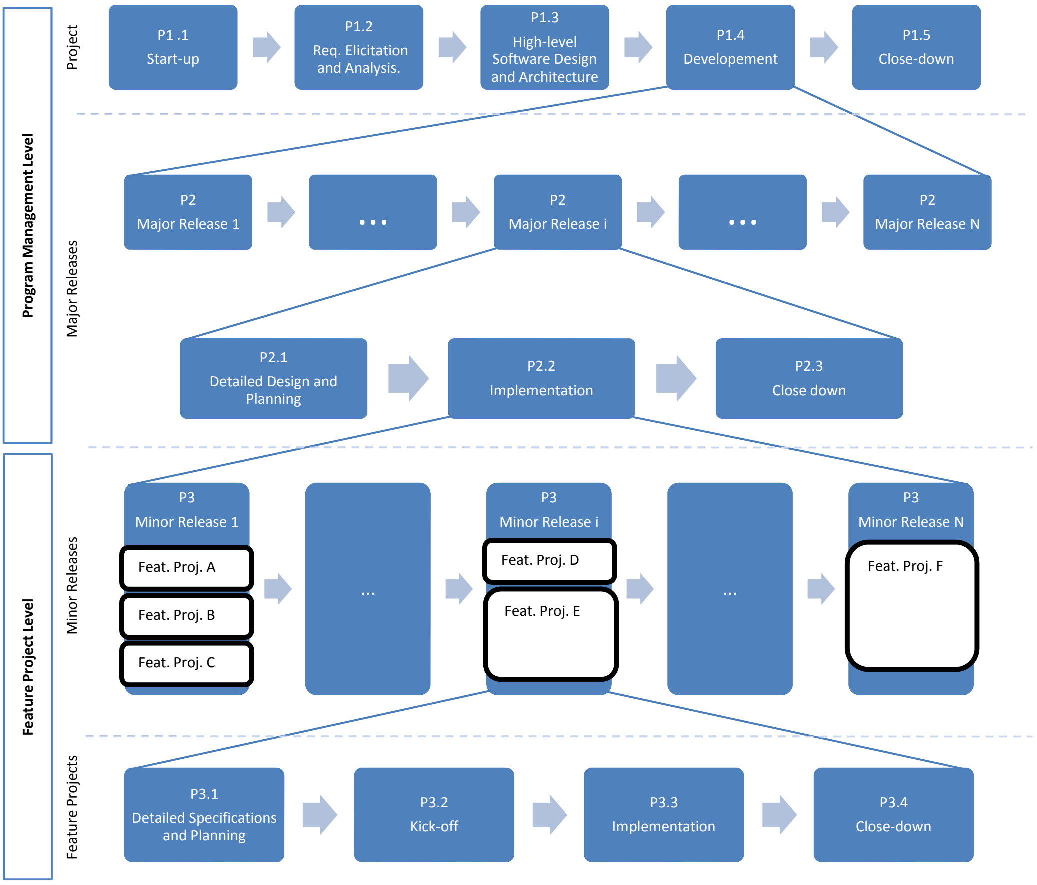


Figure 1 - Project Phase Overview

The diagram illustrates how the highest level, referred to as the project level, after start up, requirement elicitation and establishment of high level designs and project plan, is broken down into a number of major software releases (Level 2). The number of major- and minor releases will vary from project to project, mainly depending on the size, complexity and feature release plan of the development project.

During P1.2 Requirement Elicitation and Analysis, all requirements may, if feasible, be grouped in a number of Product Areas, which are logical functionality groupings, each managed by a Product Area Manager. The Product Area Manager shall have the appropriate domain- and IT expertice to efficiently manage the product area.

When planning the major release cycle (P2.1), the requirements, software design and -architecture are further refined and if appropriate the major release is broken into a number of shorter release cycles (Level 4).

The refined requiremens and designs are used to define a number of so-called *Feature Projects* which are assigned to the minor releases. Each Feature Project is responsible for implementing a set of requirements that naturally belongs together. The requirements are selected with due consideration of the release plan. During P2.1 the Product Area Managers will in close collaboration with program management define and plan the feature projects.

Feature projects lasts for an entire minor release (typically 4 weeks), and are executed in parallel. When planning the feature projects the total developer capacity of a given minor release cycle is divided between the

Feature Projects that shall be executed during that release.

Finally, each feature projects will go through the phases on the last level in the diagram, starting with a further refinement of requirements and specifications, planning of the feature project (can optionally be carried out before the start of a minor release cycle), followed by kick-off, implementation and closure.

3. Project Organization

As indicated in Figure 1, the project is organized in 2 levels.

1. A program management level – this level is responsible for the management of the project, release scheduling, project plans and processes. i.e. deals with doing the right project.

2. A feature project level – this level is responsible for detailed planning and design, implementation, testing and documentation. i.e. deals with doing the project right.

The organization on the two levels are described in the following sections.

3.1 Program Management Level

The project organization at the program management level is depicted and described in the following figure and table.

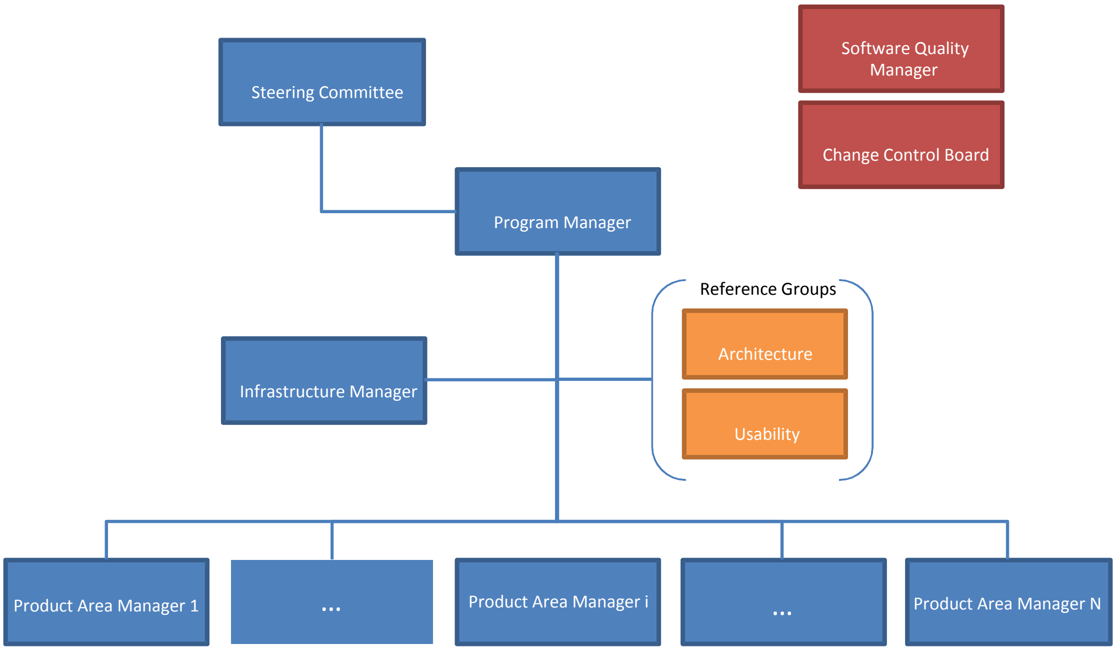


Figure 2 Project organization at Program Management level

The roles are summarised in Table 3.

.

Table 3 Roles at the Program Management level

Role	Abb	Description
Steering committee	SC	The steering committee has the overall project responsibility and shall handle resource and financial conflicts that may occur during the course of the project
Program Manager	PM	The Program Manager will manage the software development project on behalf of the steering committee.
Product Area Manager	PAM	Product Area Managers are responsible for distinct feature area sets within the solution including a breakdown of the feature area into a number of feature projects. PAMs will normally act as Project Leaders (PL) for feature projects, but may delegate this if appropriate.
Program Management Group	PMG	The Program Management Group consists of the PM, PAMs and IM
Software Quality Assurance Manager	SQAM	The Software Quality Manager overlooks the project with respect to process quality.
Infrastructure Manager	IM	The Infrastructure Manager is responsible for the development infrastructure.
Reference Groups	REFG	The reference groups are used by the project for managing internal stakeholders and as a means for reviewing ideas and designs • Architecture – forum for presenting and generating ideas for architecture related designs • Usability – forum for presenting and generating ideas related to functionality and functionality organization
Change Control Board	CCB	The Change Control Board is responsible for decision making in relation to change management

3.2 Feature Project Level

During a Feature Project, the Project Leaders will have access to a team consisting of a Lead Developer, a number of Developers, Testers and Domain Exports. The Project Leaders report to the Product Area Manager.

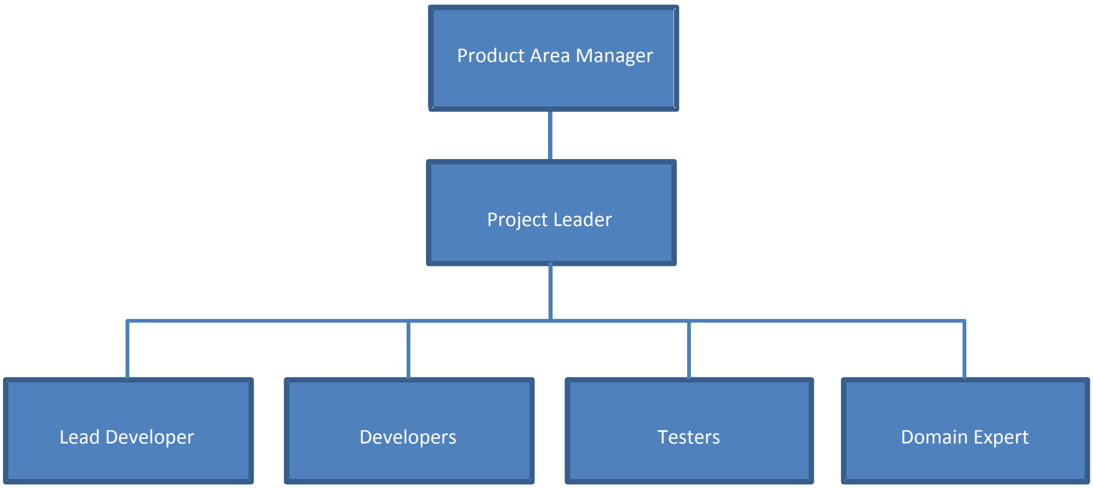


Figure 3 Project Organization at the Feature Project level

Table 4 Roles at the Feature Project level

Role	Abb	Description
Program Manager	PM	The Program Manager receives status reports and manages coordination issues.
Project Leader	PL	A group of Project Leaders will in close collaboration be responsible for detailed planning, tracking, and driving the implementation of the requirements during each feature project. The Product Area Manager may assume the role as Project Leader.
Lead Developer	LDEV	The Lead Developer is responsible for the implementation design, documentation of the implementation. He actively participates in coding and unit testing.
Developers	DEV	The Developers work with coding and unit testing.
Testers	TST	The Testers do regression testing and maintain the Issue Tracking System accordingly. The tester also performs dry-runs of the acceptance test plan.
Domain Expert	DEXP	The Domain Expert is responsible for understanding the requirements, for ensuring good usability and for executing the acceptance testing plan. Note: The Project Leader may assume the role as Domain Expert.

4. Project Life cycle

This chapter describes the details of each of the phases that were introduced in Figure 1.

4.1 Level 1: Project Life Cycle

The project follows the life cycle model shown in the figure below and described in the accompanying phase description tables.

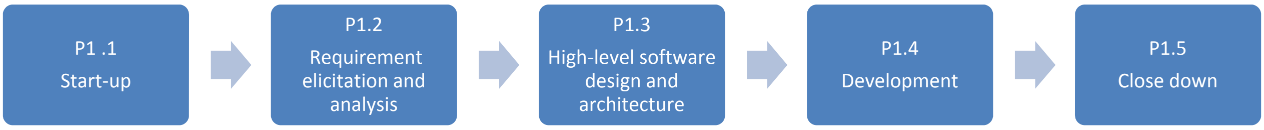

P1 .1
Start-up
P1.2
Requirement elicitation and analysis
P1.3
High-level software design and architecture
P1.4
Development
P1.5
Close down

Figure 4 - Main execution model

The following tables describe each of the above phases in terms of purpose, required criteria for entering the phase, activities to be managed during the phase and requirements for finishing the phase. The tables also defines the tangible deliverables from the phases.

Table 5 P1.1 Start-up

Name:	P1.1 - Startup	
Purpose:	Define the project in terms of processes, infrastructure, stakeholders and deliverables	
Entry criteria	**Activities**	**Exit criteria**
Project agreement signed	Establish product vision Establish infrastructure Establish project portal Establish processes Establish high-level Project Plan Establish document templates	Acceptance of the requirement management system and process by the client Acceptance of issue management system by the client Acceptance of all document templates by Program Management
Output:	The following list of external documents *None* The following list of internal documents 1. <Project-id>.PVI - Product Vision 2. <Project-id>.PHA – Project Handbook 3. <Project-id>.PPL – Project Plan 4. <Project-id>.SRS – Software Requirement Specification (template) 5. <Project-id>.SAD – Software Architecture Documentation (template) 6. <Project-id>.TPL – Test Plan (template) 7. <Project-id>.SUM – Software User Manual (template) 8. <Project-id>.<fp-id>.SRS – Feature Project Software	

	Requirement Specification (template) 9. <Project-id>.<fp-id>.SAD – Feature Project Software Architecture Documentation (template) 10. <Project-id>.<fp-id>.PPL – Feature Project Plan (template) 11. <Project-id>.<fp-id>.TPL – Feature Project Test Plan (template) The following list of infrastructure components 1. Requirement management and issue tracking systems 2. Project Portal

Table 6 P1.2 Requirement Elicitation and -analysis

Name:	P1.2 Requirement Elicitation and Analysis	
Purpose:	Collect, categorize and understanding the requirements	
Entry criteria	**Activities**	**Exit criteria**
P1.1 Completed	Requirement elicitation Assess the completeness of the requirements, and identify additional requirements if required. Breakdown in product areas. Establish product area visions. Transfer all requirements to the requirement management system system. Perform initial effort estimation and revise project plan.	Acceptance of the Software Requirement Specification (SRS) by the client
Output:	The following list of external documents 1. <Project-id>.SRS – Software Requirement Specification (new) The following list of internal documents (Program Management Level) 1. <Project-id>.PBD - Product Area Breakdown (new) 2. <Project-id>.PPL – Project Plan (revised)	

Table 7 P1.3 - High-level design and -architecture

Name:	P1.3 High-level Design and Architecture.	
Purpose:	Establish high level software design and architecture.	
Entry criteria	**Activities**	**Exit criteria**
P1.2 Completed	Establish a high-level software design and architecture. Identify additional requirements, and update DSS.SRS if required. Revise estimates Revise Project Plan Perform requirement baselining.	Acceptance of revised estimates by DHI management Acceptance of the Software Architecture Document by the Client (SAD) CCB approval of baseline requirements
Output:	The following list of external documents 1. <Project-id>.SAD – Software Architecture Documentation (new) 2. <Project-id>.SRS – Software Requirement Specification (Revised) The following list of internal documents (Program Management Level) 1. <Project-id>.PPL – Project Plan (revised)	

Table 8 P1.4 Development

Name:	P1.4 Development	
Purpose:	Implementation of the softeware system.	
Entry criteria	**Activities**	**Exit criteria**
P1.3 Completed	*See Phase P2.1 – P2.3*	Acceptance criteria defined in the Test Plan passed
Output:	*See Phase P2.1 – P2.3*	

Table 9 P1.5 Close-down

Name:	Close-down	
Purpose:	Finalize the project by consolidating the systems documentation and disassembling the team	
Entry criteria	**Activities**	**Exit criteria**
P1.4 Completed	Finalizing the Software User Manual Releasing the team members Report status	DSS.SUM finalized
Output:	The following list of documents 1. <Project-id>.SUM – Software User Manual (final)	

4.2 Level 2 – 3: Major Release Cycles

Each of major release development cycles will go through the phase P2.1 –P2.3 as shown in Figure 5.

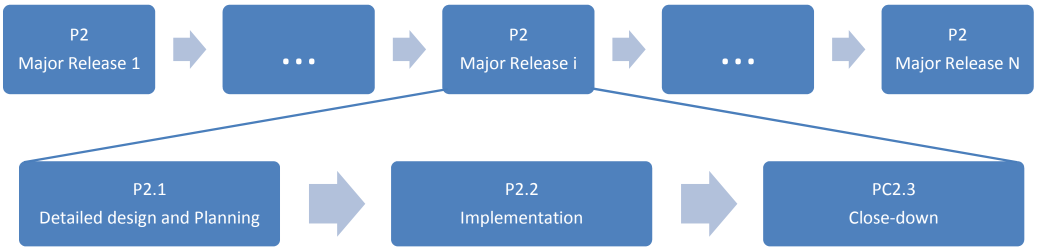


Figure 5 – Major Release Development cycle execution model

The following tables describe each of the above phases in terms of purpose, required criteria for entering the phase, activities to be managed during the phase and requirements for finishing the phase. The tables also defines the tangible deliverables from the phases.

Table 10 P2 Major Release Development Cycle

Name:	P2 Major Release Development Cycle	
Purpose:	Implementation of the softeware system.	
Entry criteria	**Activities**	**Exit criteria**
P1.3 Completed	*See Phase P2.1 – P2.3*	Acceptance criteria defined in the Test Plan passed *See Phase P2.1 – P2.3*
Output:	DSS.SRE Status Report <date> *See Phase P2.1 – P2.3*	

Table 11 P2.1 Detailed design and -planning

Name:	P2.1 Detailed Design and Planning	
Purpose:	Identify and plan the minor releases and feature projects to be executed in this development cycle.	
Entry criteria	**Activities**	**Exit criteria**
Previous phase completed	Further refinement of the requirements to be implemented in this release. Further refinement of the software design and architecture. Revise the estimates and project plan. Define Feature Projects. Plan implementation cycle Perform P2 baselining	Acceptance of the project plan by Program Management Revised Software Requirement Specification document accepted by the client Revised Software Architecture Document accepted by the client.

Output:	The following list of external documents 1. <Project-id>.SRS – Software Requirement specification (revised) 2. <Project-id>.SAD –Software Architecture Documentation (revised) The following list of internal documents 1. <Project-id>.PPL – Project Plan (revised)

Table 12 P2.2 Implementation

Name:	P2.2 Implementation	
Purpose:	Monitor the feature implementation projects, report progress and manage changes.	
Entry criteria	**Activities**	**Exit criteria**
P2.1 completed	*See Phase P3*	Acceptance criteria defined in the Test Plan passed.
Output:	*See Phase P3*	

Table 13 P2.3 Close-down

Name:	P2.3 Close-down	
Purpose:	Consolidate feature implementation deliverables and perform integration testing	
Entry criteria	**Activities**	**Exit criteria**
P2.2 completed	Perform integration testing. Update User Manual	
Output:	The following list of external documents 1. <Project-id>.SUM – Software User Manual The following list of internal documents *None*	

4.3 Level 4 - 5: Minor releases and Feature Projects

Each feature implementation project follows a project model as depicted and described below.

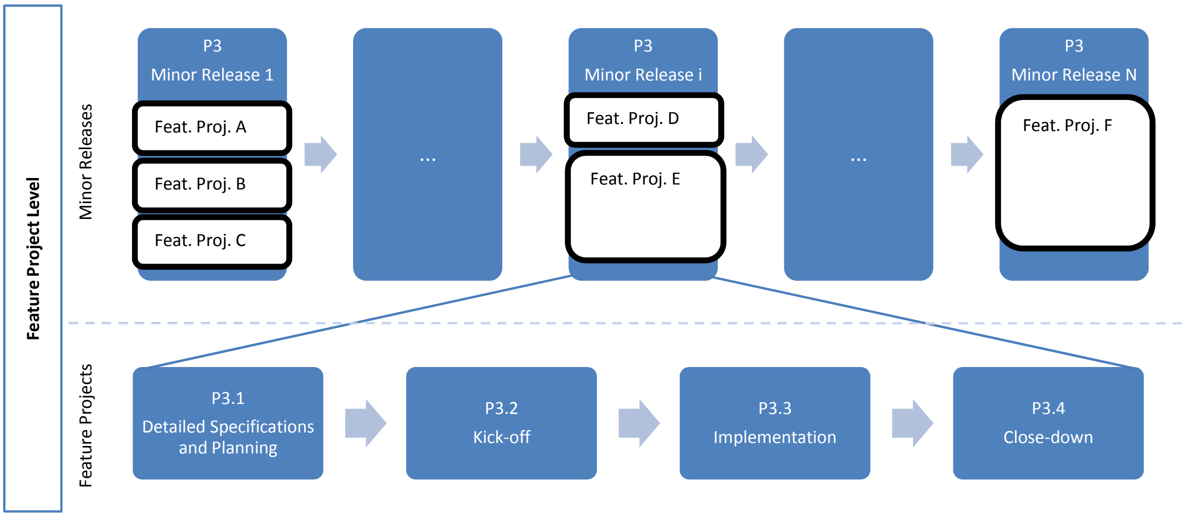


Figure 6 Minor Release development cycle execution model

The following tables describe each of the above phases in terms of purpose, required criteria for entering the phase, activities to be managed during the phase and requirements for finishing the phase. The tables also defines the tangible deliverables from the phases.

Table 14 P3 - Minor Release Development Cycle Phase

Name:	P3 Minor Release Development Cycle	
Purpose:	Plan and monitor the feature projects, report progress and manage changes.	
Entry criteria	**Activities**	**Exit criteria**
Previous phase completed	*See Phase P3.1 – P3.4*	Acceptance criteria defined in the Test Plan passed
Output:	*See Phase P3.1 – P3.4*	

Table 15 P3.1 Detailed specifications and planning

Name:	P3.1 Detailed Specifications and Planning	
Purpose:	Plan the feature projects	
Entry criteria	**Activities**	**Exit criteria**
P2.1 completed	Refine the associated requirements and establish feature project SRS Refine software design and architecture, and establish feature project SAD Establish feature project Test Plan Revise estimates and establish feature project plan. Perform Feature Project baselining. Assign staff	Feature Project Plan approved by Program Management. Feature Project SAD approved by Program Management. Feature Project SRS approved by Program Management Feature Project Test Plan approved by Program Management.
Output:	The following list of internal documents: 1. <Project-id>.<fp-id>.FPL - Feature Project Plan (new) 2. <Project-id>.<fp-id>.SAD - System Architecture Document (new) - *temporary document to be incorporated in DSS.SAD during P3.4* 3. <Project-id>.<fp-id>.SRS - Software Requirement Specifications (new) - *temporary document to be incorporated in DSS.SRS during P3.4* 4. <Project-id>.<fp-id>.TPL - Test Plan (new) - *temporary document to be incorporated in DSS.TPL during P3.4)* The following list of external documents: *None*	

Table 16 P3.2 Kick-off the project

Name:	P3.2 Kick-off project	
Purpose:	Start the project through on-site workshops or tele-conferences	
Entry criteria	**Activities**	**Exit criteria**
P3.1 completed	Gather the development team and establish a common understanding of what has to be achieved Let the development team estimate the assigned tasks. Revise project plan Establish re-current project meetings	

Output:	The following list of documents 1. <Project-id>.FPL Feature Project Plan (revised)

Table 17 P3.3 Implementation

Name:	P3.3 Implementation	
Purpose:	Implementation of the requirements assigned to the Feature Project	
Entry criteria	**Activities**	**Exit criteria**
P3.2 completed	Develop code Test Report progress	All specifications implemented Acceptance criteria defined in the Test Plan passed
Output:	The following list of documents 1. <Project-id>.<fp-id>.SRE – Status Report <date> (new) Software code	

Table 18 P3.4 Close Down

Name:	P3.4 Close down	
Purpose:	Close down feature project	
Entry criteria	**Activities**	**Exit criteria**
P3.3 completed	Report status Consolidate all feature project SRS documents in DSS.SRS Consolidate all feature project SAD documents in DSS.SAD Consolidate all feature project TPL documents in DSS.TPL Show-and-tell session	
Output:	The following list of documents 1. DSS. <fp-id>.SRE –Status Report <date> (new) 2. DSS.SRS Software Requirement Specification (revised) 3. DSS.SAD Software Architecture Document (revised) 4. DSS.TPL Test Plan (revised) Software code	

Appendix

Appendix A. Requirements and Incidents

A.1. Introduction

This Appendix describes the procedures that will be used by DHI for registration, analysis, and tracking of requirements, change requests and defects. The procedure describes the activities, methods, responsibilities and the documentation required during the process.

It is the intention that the DHI issue tracking procedure should comply with the Process Improvement Plan agreed with the NBI (Appendix H, Chapter 4 of the Contract). In regard to Defect and Requirement Handling the Process Improvement Plan recommends the following:

- Establish a requirement repository with requirement ID, origin and status
- Include requirement analysis
- Ensure bi-directional traceability
- All the people impacted by a requirement to verify the requirement analysis result and accept the solution (they might come from different domains).
- Create a Change Control Board for deciding on what requirements are valid and prioritize among them.
- Treat change requests the same ways as requirements.
- Treat defects as change requests.

Requirements, change requests and defects will be handled in a common Application Lifecycle Management System (ALM). Requirements will be handled in the ALM requirement repository, and defects and change requests will be handled as project incidents in the ALM incident repository. The three workflows will, however, be similar.

System Descriptions

DHI will use SpiraTeam, an ALM developed by Inflectra, for management of requirements, releases, tests, incidents and tasks.

SpiraTeam provides the ability to structure requirements in a hierarchical organization that resembles a typical scope matrix. In addition, each requirement is mapped to one or more test cases that can be used to validate that the functionality works as expected (the so-called Requirement Test Coverage).

Each test case consists of a set of test steps that represent the individual actions a user must take to complete the test. During the execution of the test case, each failure can be optionally used to record a new incident (defect), which can then be managed in the Defect Workflow. This provides complete traceability from a recorded incident to the underlying requirement that was not satisfied.

Requirements that have one or more children are referred to as Summary Items. The summary items simply display an aggregate of the worst-case assessment of their children’s status. Also, only detail items can be mapped against test-cases for test-coverage, the summary items simply display an aggregate coverage status of their children.

The lowest-level requirements can be further drilled down into a series of work items called ‘Tasks’. These tasks are the discrete activities that each member of the development team would need to carry out for the requirement to be fulfilled. Each task can be assigned to an individual user as well as associated with a particular major of minor release. The system can then be used to track the completion of the different tasks to determine if the project is on schedule.

Finally, SpiraTeam provides the ability to manage *incidents* that may occur during the life of an application. An inciden can be e.g. Defects, Change Requests and Risks. Each incident type may have a customized workflow associated.

A.2. Roles and Responsibilities

The most important purpose of the ALM is to ensure that all requirements, tasks, defects, tests, and releases are treated in a structured, transparent and safe manner, specifically in order to ensure:

- That all requirements, tasks, tests and incidents are uniquely identified and safely stored in a database.
- That all requirements and incidents are registered and analyzed before presented to the Change Control Board.
- That all requirements and incidents are accepted or rejected by a Change Control Board prior to implementation.
- That all requirements are covered by Test Cases.
- That all requirements and incidents are verified (tested) and approved before closure.
- That the project progress and status can be tracked.
- That bi-directional traceability of requirements, tasks, incidents, tests and releases is maintained (from registration to implementation in feature projects and source code and vice-versa).

Roles and Responsibilities in the Issue management procedure are listed in Table A- 1.

Table A- 1 Issue Management, Roles and Responsibilities.

Role	Abb	Description
Program Manager	PM	The PM initiates the procedure. Establish the requirement and incident management system in SpiraTeam. Ensure that relevant staff understands the procedure. Creates the Requirements and Incident Management Sheet (Appendix A.1). Manages the procedure to ensure that it is met.
Product Area Manager	PAM	Registration of requirements and incidents, analysis, and closure of requirements and incidents as described in the Requirements and Incident Management sheet.
Project Members	PM	Follow the Requirements and Incident Management procedures.
Change Control Board	CCB	Approve/reject issues in accordance with the described procedure.

A.3. ALM System Workflows

The ALM basically have two different workflows for handling requirements, and a third workflow for handling defects. Having two requirements workflows allows us to distinguish between requirements that origins from the natural breakdown and refinement of previously approved requirements and requirements that origins from changes.

These are:

- Requirement Workflow (RW)
- Change Request Workflow (CRW)
- Defect Workflow (DW)

Whenever a previously approved requirement is refined into one or more derived requirements (product breakdown), the derived requirements will have to go through the RW.

All changes to previously approved requirements, and new requirements that can not be derived from existing, approved requirements, have to go through the CRW.

Requirements that are handled with the RW are added directly to the ALM requirements repository, whereas changes requests are handled as project incidents that may, in case of CBB approval, result in additional requirements being added to the requirements repository.

Defects will be handled as Project Incidents of the type Defect, using the Defect workflow.

At the beginning of the project the ALM only contains the requirements that are listed in the ToR. Children of these requirements can be registered using the RW. Additional requirements will go through the CRW.

The workflows are illustrated in Figure 7.

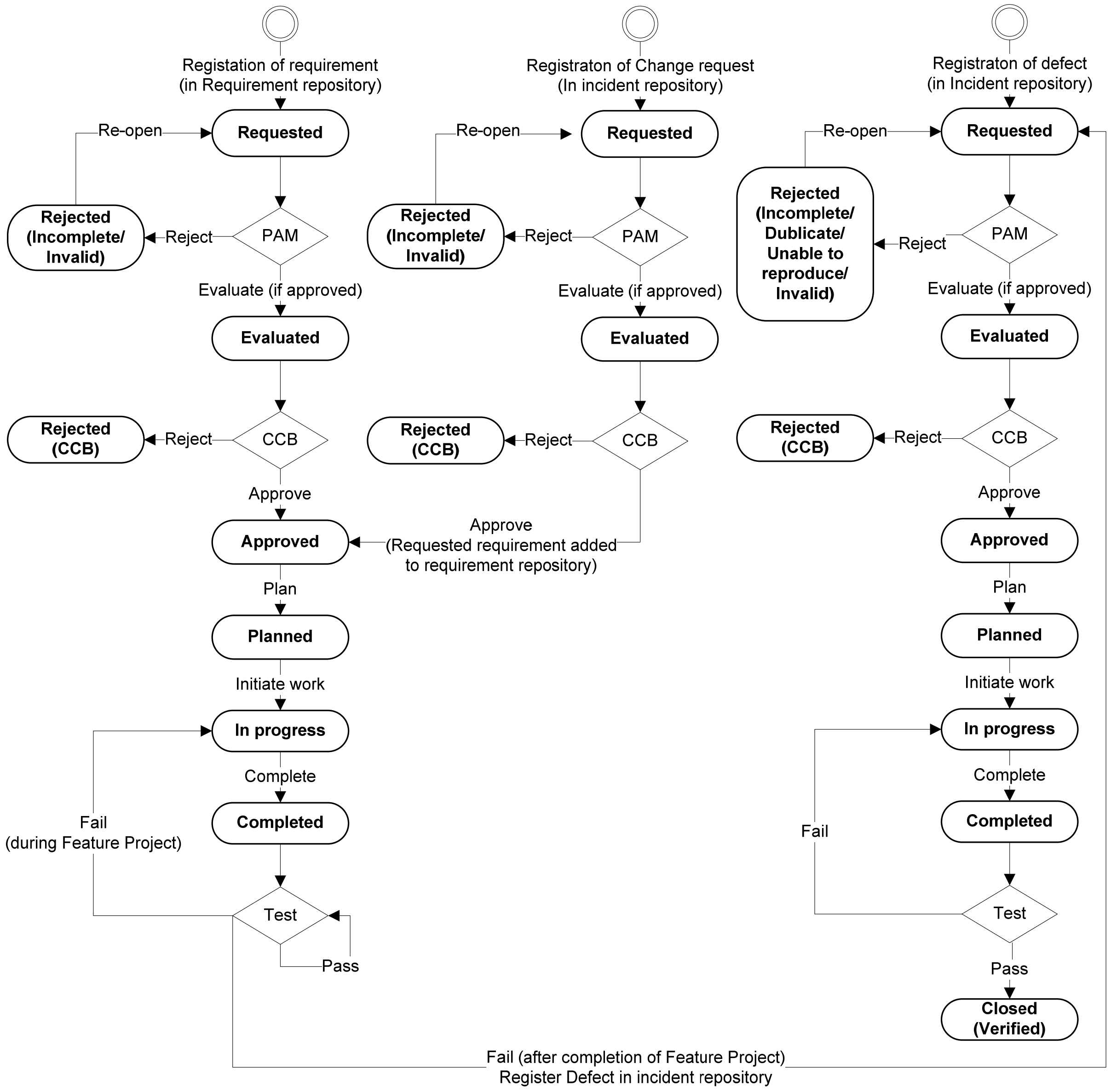


Figure 7 Issue Management Flow Diagram.

A5. The Requirement Workflow (RW)

This section describes the process for registering, evaluating and tracking the status for requirements.

The workflow is used for requirements that are defined as a result of the refinement of the ToR requirements, or refinement of previously approved requirements (i.e. requirement breakdown).

The RW may be initiated by a number of different sources including, but not limited to:

- The Client (designated staff)
- The Change Control Board.
- DHI project management
- DHI software management

Throughout the process Requirements may have one of the following states:

Status	*Process step*	*Required fields*
Requested	When entering the requirements database.	Name Description Author
Rejected(incomplete)	When insufficient information has been supplied during registration	Description – Reason for rejection added.
Rejected(invalid)	When the registered requirement is not a requirement, was reported on a wrong product, or was the result of a wrong operation	Description – Reason for rejection added.
Rejected(dublicate)	When the defects has already been registered	Reference to dublicate.
Evaluated	When an evaluation has confirmed whether or not this is a valid requirement. When a requirement has the status Evaluated, it can be presented to the CCB.	Importance Product Area Stakeholder Assiciations (if any) Planned effort
Rejected	When the CBB rejects a requirement.	Description – Reason for rejection added.
Approved	When the CBB accepts a requirement	Importance (changed if CCB decides to)
Planned	When the requirement is assigned to a future release (automatic).	Test coverage (min. one test case) Owner Release
In Progress	When a developer has initiated the work on one or more of the tasks that are associated to the requirement (automatic).	None
Completed	When all the child requirements and tasks has been completed (automatic).	None

A number of activities are associated with each process step. These activities are treated in detail in the following sections.

Registration

A CCB representative, the PM or a PAM can register requirements. The initial registration should include as much information about the requirement as possible and reflect the initial knowledge and requirement evaluation made by the person who enters the requirement.

Initial registration of requirements shall always be done directly in the ALM. An entry is not accepted until it has been entered in the ALM database.

At initial registration the database system shall automatically generate or force the entry of certain issue annotations including:

- Unique issue number (auto generated by SpiraTeam and never changed).
- Status of the issue (status will be “Requested” at Initial Registration)
- Date of the issue entry (auto generated by SpiraTeam)
- Name of the person who entered the requirement in SpiraTest (SpiraTeam user name).
- Links to related requirements, incidents, and/or the use cases from which a requirement is derived.

A complete list of annotations that are required when a requirement is registered is given in

Table A- 4.

Evaluation

After the initial registration every requirement is evaluated by the PAM. At this stage requirements may be ”rejected” by the PAM, if :

- The requirement is a duplicate of another issue.
- The requirement is not valid (is not a requirement, was reported on a wrong product, or was the result of a wrong operation)
- The requirement is incomplete

If the requirement is rejected, the reporter shall always be informed about the decision with a full explanation, and the option of re-opening the issue in case of a misunderstanding or other unknown circumstances.

In order to pass the *Evaluation* a requirement must meet the requirements listed in

Table A- 2 Criteria that must be met to pass the Evaluation Analysis.

Criteria	Description
Necessary	Do you need the requirement?
Unambigious	All stakeholders shall understand the requirement in the same way.
Complete	You should not need to guess any details.
Verifiable	It shall be possible to test/verify the requirement.
Consistent	No requirement may contradict other requirements.
Attainable	It shall be possible to fulfil the requirement within the scope of the project.

The *Evaluation* includes four steps:

- Clarification
- Traceability analysis
- Impact analysis and Estimation

Clarification

This part of the analysis shall make the requirement fulfil all the criteria listed in Table A- 2, both individually and as a set of requirements. It will often require changing the original description (wording) and adding extra information to the issue. It is therefore important that the reporter of the issue is made aware of the changes in order to have a chance to correct any misinterpretation. Within the project the *clarification* shall, to the extent possible, be a joint DHI, Client effort.

Putting the question: “How shall we verify this requirement?” may in some cases require a clarification of the issue. Therefore it is recommended to specify the verification method, whenever possible. SpiraTeam supports creation of Test Cases directly from the requirement dialog, and maintains a two-way tracability.

Traceability Analysis
Each requirement shall be tracked to any “higher” requirement, e.g. if one high-level requirement is resulting in five derived requirements, each of these shall contain the relation to the higher level .This two-way tracability is fully supported by SpiraTeam. If a requirement is derived from a Use Case, a two-way link between the use case and the requirement has to be established.

Impact Analysis and Estimation
During the analysis of requirements, it is often practical to consider how to solve these in order to help the planning of the implementation, or to estimate the cost of the implementation. Many issues may have alternative solutions implying very different costs, and in these cases it may be beneficial to indicate the chosen solution in the issue.

Any description about the envisaged implementation (impact analysis) is provided as text description with the issue (in SpiraTeam).

Estimation is done on all requirements by issue basis as described in Appendix B of the Project Handbook.

A complete list of annotations that are required when a requirement is evaluated is given in Table A- 5.

Requirement approval by the CCB

After obtaining the status of *Evaluated* every requirement shall be evaluated by the CCB. The outcome of this evaluation may be:

- Approved. The implementation may be started. Issue status changes to *“Approved”*
- Rejected, if the issue shall not be implemented. Requires a reason.

If the issue is rejected, the reporter shall always be informed about the decision with a full explanation, and the option of re-opening the issue in case of a misunderstanding or other unknown circumstances.

If the requirement is accepted, the CCB assigns a priority to the requirement.

A complete list of annotations that are required when a requirement is accepted or rejected is given in Table A- 6 and Table A- 7.

Planning

The state changes automatically to Planned when the PAM or PM assigns the implementation of a requirement to a release (i.e. a feature project). Annotations related to the Planning of a Requirement implementation is given in Table A- 8.

Progress

The state In Progress is automatically obtained when a developer starts to register time on one of the tasks that are associated to the given requirement (or child requirement). The system will track the progress and revised estimates for the tasks and display them against the corresponding requirement estimates so that risks to the schedule can be quickly determined.

Implementation of tasks will be done as part of a number of *feature projects*. A *feature project* implements a set of requirements and has a typical duration of approximately 2 calendar months. Feature projects are conducted as described in Appendix B of the Project Handbook.

Completion

The status Completed is automatically obtained when all child requirements/tasks has obtained the state Completed.

All requirements with the status Completed will continuously be tested by executing the associated Test Cases. If a test fails before signing off the feature project, the status of requirement is changed to In Progress until the issues that have prevented the successful execution of the test is resolved.

If a test is failing after the feature project has been signed off, and new incident of type Defect is registered in the incident repository.

A6 The Change Request Workflow (CRW)

This section describes the process for registering, evaluating and tracking the status for Change Requests.

A change request is handled as a project issues (in the ALM issues repository) until it has been approved by the CCB. After approval, it is converted to a requirement that will enter the RW with the status Approved. SpiraTeam maintains two-way tracability between the original Change Request incident, and the resulting requirement(s).

Change requests are treated as project issues of the type *Change Request.*

The Change Request Workflow shall be be used when:

- A change to a previously approved requirement is requested
- A new requirement, that does not origin from a breakdown of previously approved requirements is requested.

The CRW may be initiated by a number of different sources including, but not limited to:

- Client (designated staff)
- The Change Control Board.
- DHI project management
- DHI software management

Throughout the process Change request may have one of the following states:

Status	*Process step*	*Required fields*
Requested	When the Change request is entered in the ALM database.	Name Description Severity Detected by
Rejected(incomplete)	When insufficient information has been supplied during registration.	Description – Reason for rejection added.
Rejected(invalid)	When the issues is not a valid issue	Description – Reason for rejection added.
Evaluated	When an analysis has confirmed whether or not this is a valid change request, risk and required effort. When a requirement has the status Evaluated, it can be presented to the CCB.	Analysis report (text) Risk level Owner Estimated effort
Rejected	When the CBB rejects a requirement.	Analysis report – Reason added
Approved	When the CBB accepts a requirement	Priority

Registration

A CCB representative, the PM or a PAM can register Change Requests. The initial registration should include as much information about the requested change as possible and reflect the initial knowledge and requirement evaluation made by the person who enters the request.

Initial registration of change requests shall always be done directly in the ALM. An entry is not accepted until it has been entered in the ALM database.

A complete list of annotations that are required to register a Change Request is given in Table A- 9

Evaluation

After the initial registration every Change Request is evaluated by the PAM. At this stage requests may be ”rejected” by the PAM, if :

- The change request is a duplicate of another request.
- The request is not valid (i.e. it was reported on a wrong product, or was the result of a wrong operation, it is not a change)
- The request is incomplete.

If the requirement is rejected, the reporter shall always be informed about the decision with a full explanation, and the option of re-opening the issue in case of a misunderstanding or other unknown circumstances.

In order to pass the *Evaluation* a requirement must meet the requirements listed in

Table A- 3 Criteria that must be met to pass the Evaluation Analysis.

Criteria	Description
Necessary	Do you need the requested change?
Unambigious	All stakeholders shall understand the requested change in the same way.
Complete	You should not need to guess any details.
Verifiable	It shall be possible to test/verify the impact of the change.
Consistent	No changes may result in contradicting requirements
Attainable	It shall be possible to fulfil the change within the scope of the project.

The *Evaluation* includes two steps:

- Clarification
- Impact analysis and Estimation

Clarification

This part of the analysis shall make the change request fulfil all the criteria listed in Table A- 3. It will often require changing the original description (wording) and adding extra information to the issue. It is therefore important that the reporter of the issue is made aware of the changes in order to have a chance to correct any misinterpretation. Within the project the *clarification* shall, to the extent possible, be a joint DHI, Client effort.

Putting the question: “How shall we verify the implementation of the requested changes?” may in some cases require a clarification of the issue. Therefore it is recommended to specify the verification method, whenever possible.

Impact Analysis and Estimation

During the analysis of a change request, it is required to consider the risk associated with the proposed change, and estimate the cost of the implementation. Many issues may have alternative solutions implying very different costs and risks. Any description about the envisaged solution (impact analysis) is provided as text description with the issue (in SpiraTeam). Information on risk, costs and envisaged solution must be available when the issue is presented for the CCB.

A complete list of annotations that are required when a change request is evaluated is given in Table A- 11

Approval by the CCB

After obtaining the status of *Evaluated* every change request shall be evaluated by the CCB. The outcome of this evaluation may be:

- Approved. The implementation may be started. Issue status changes to *"Approved"*
- Rejected, if the issue shall not be implemented. Requires a reason.

If the issue is rejected, the reporter shall always be informed about the decision with a full explanation, and the option of re-opening the issue in case of a misunderstanding or other unknown circumstances.

After approval, it is converted to a requirement that will enter the RW with the status Approved. SpiraTeam maintains two-way tracability between the original Change Request incident, and the resulting requirement(s).

If the change regards a requirement that has already been registered in the requirement repository, the existing requirement is closed, and a new requirement that includes the changes is added. SpiraTeam supports two-way tracability between the original requirement, the change request and the new requirement(s).

A complete list of annotations that are required when a change request is accepted or rejected is given in Table A- 12.

A7 The Defect Workflow (DW)

A defect can be defined as a problem that causes a program to produce invalid output, to crash, or failure to conform to specifications.

Defects are handled as Project Incidents of the type Defect, using the Defect workflow. A defect will, after evaluation, approval, fixing and testing be closed with the status Closed(Verified).

The DW may be initiated by a number of different sources including, but not limited to:

- Client (designated staff)
- The Change Control Board.
- DHI project management
- DHI software management

Throughout the process Defects may have one of the following states:

Status	*Process step*	*Required fields*
Registered	When the defect is registered in the incident repository.	Name Description Author
Rejected(incomplete)	When insufficient information has been supplied during registration.	
Rejected(dublicate)	When the defects has already been registered	Reference to dublicate.
Rejected(not reproducible)	When the defect can not be reproduced	
Rejected(by design)	When the reported defect is by design. The issue may instead be raised as a Change Request.	Reference to requirement
Evaluated	When an evaluation has confirmed whether or not this is a valid defect. When a defect has the status Evaluated, it can be presented to the CCB.	Severity Product Area Stakeholder Assiciations (if any)

		Planned effort Risk
Rejected (CCB)	When the CBB rejects a resolution.	Description – Reason for rejection added.
Approved (CCB)	When the CBB approves a resolution.	Priority
Planned	When the defect is assigned to a feature project.	Test coverage (min. one test case) Owner Release (Feature project)
In Progress	When a developer has initiated the work on the defect.	None
Completed	when the developer changes the state to Completed.	None
Closed(Verified)	When all tests have been executed successfully	

A number of activities are associated with each process step. These activities are treated in detail in the following sections.

Registration

The initial registration should include as much information about the defect as possible and reflect the detailed knowledge of the person who enters the defect.

Initial registration of defects shall always be done directly in the ALM. An entry is not accepted until it has been entered in the ALM database.

A complete list of annotations that are required to register a Defect is given in Table A- 15.

Evaluation

After the initial registration every defect is evaluated by the PAM. At this stage requests may be ”rejected” by the PAM, if :

- The defect is already registered.
- Insufficient information has been supplied at registration.
- The defect can not be reproduced.
- The request is not valid (i.e. it was reported on a wrong product, or was the result of a wrong operation)
- The reported behavior is by design. In those cases, the defect may instead be reported as a Change Request.

If the defect is rejected, the reporter shall always be informed about the decision with a full explanation, and the option of re-opening the issue in case of a misunderstanding or other unknown circumstances.

The *Evaluation* includes two steps:

- Clarification
- Impact analysis and Estimation

Clarification

This part of the analysis shall make the defect is:

- Reproducible
- Not already registered
- Not by design

Moreover, the associations to existing test cases and requirements should be established. If none of the existing test cases covers the defect, a new test case shall be registered in the ALM, and associated to the relevant requirement(s) (identified during the evaluation). The test case(s) will be used to verify the fix.

Impact Analysis and Estimation
During the analysis of a reported defect, it is required to consider the severity of the defect as well as the risk associated with resolving the bug (e.g. risk for side effects). If possible, the cost of resolving the defect should be estimated. Otherwise the initial estimate will be set to 4 hours. The developer will revise the estimate. Any description about the envisaged solution (impact analysis) is provided as text description with the issue (in the ALM).

Information on risks, costs and envisaged solution must be available when the issue is presented for the CCB.

A complete list of annotations that are required when a change request is evaluated is given in Table A- 17

Approval by the CCB

After obtaining the status of *Evaluated* every defect shall be evaluated by the CCB. The outcome of this evaluation may be:

- Approved. The implementation may be started. Defect status changes to *"Approved"*
- Rejected, if the defect shall not be resolved. Requires a reason.

If the issue is rejected, the reporter shall always be informed about the decision with a full explanation, and the option of re-opening the issue in case of a misunderstanding or other unknown circumstances.

A complete list of annotations that are required when a defect is accepted or rejected is given in Table A- 12 and Table A- 13.

Planning

The defect obtains the Planned state when the PAM or PM assigns the resolvement of a defect to a release (i.e. a feature project).

Progress

The state In Progress is obtained when a developer starts to register time on the defect. The system will track the progress and revised estimates for resolving the defect.

Completion

The status Completed is obtained when the developer changes the state to Completed.

Testing

When a defect has obtained the status of Completed, all test cases that are associated to the requirements that are affected by the defect is executed (identified during the evaluation). If all tests are executed successfully, the defect is closed. Otherwise, the status is changed to *In Progress*.

A.4. Issue Documentation and Annotations

When entered into SpiraTeam all issues will be associated with a number of annotations. These annotations are required in order to describe, classify, analyze and track every issue from entrance to closure.

Issue Documentation Styles
Issues may be documented in various ways:

- Text (see below)
- Tables (useful for large amounts of similar data)
- Models (when in file format, attach to database record)
- Tasks (scenarios or “use cases”)

Text entry will always be used in the database for parts of the issue, e.g. the title of an issue. Text should preferably be a single (or very few) sentences in natural language:

- Start with: ”The product …”
- Avoid synonyms – stick to a defined vocabulary
- Avoid subjective words (useful, high, easy)
- Avoid generalities like ”etc.”, ”and so on”
- Be aware of ”and” and ”or”
- Use ”shall”, ”should” and ”may” with care!

To make statement requirements more precise and verifiable, metrics should whenever possible be used. Add for example:

- Target
- Acceptable limits

Each issue may furthermore need to be classified (annotated) within a defined set of parameters in order to be ready for the analysis. Each annotation should always be limited to a defined set of values in order to be able to make queries and statistics from the database. Values should preferable not include “misc.”, “other” or similar.

Issue Annotations used in SpiraTeam
The SpiraTeam issue database is setup with a number of annotations that provides the necessary information about an issue.

In principle any annotation may be filled out already at *Initial Registration.* As such the person who enters the issue should add as much “knowledge” to the issue as possible. The *Initial Registration* should represent the best knowledge and evaluations of the person who entered the issue.

It will however be the PAM’s responsibility to review each and every annotation and change them as part of the *Registration* and/or during the *Evaluation.* Often this will take place in dialogue with the person who made the *Initial Registration* of the Issue.

Some annotations, such as *Issue state* are managed by SpiraTeam and can only change when the PAM works through the individual steps in the issue flow process.

The following sections lists the Issue Annotations used by DHI.

Requirement Workflow (RW) annotations

Annotations for the Requirement Workflow is provided in the tables below.

Table A- 4 Annotations Related to Requirement Registration

Identification	Allowed Values	Explanation
Requirement ID	Number	Unique ID (Auto-generated)
Name	Free text	The Request title
Description	Free text	Narrative description of the requirement. Include attachments for supporting information. Attachments may include all document types. Often it will be use cases, activity diagrams and other information that refines the requirement. Verification/Verification method: how to test the requirement.
Author	[user name]	Name of the user who entered the issue.
Reference	Text	If relevant add a reference to a requirement document, ToR or similar contractual documents.
Date of Creation	Date	Auto-generated at the time of registration

Table A- 5 Anntations related to the Requirement Evaluation

Identification	Allowed values	Explaination
Stakeholder	Text	Name of the key stakeholder (e.g. the customer for a Business issue or DHI for an internal issue)
Owner	[user name]	The responsible PAM
Planned effort	Hours	The initial estimate of the time required to implement a requirement
Evaluation	Text	A textural description of the implications and risks that are associated to the implementation of the requirement, along with an envisaged solution. The Evaluation should provide the CCB with sufficient information to approve/reject the requirement.

Table A- 6 Annotations Related to Requirement Approval by the CCB.

Identification	Allowed Values	Explanation
Importance	Critical High Medium Low	Shall be implemented .. .. May be implemented if time

Table A- 7 Annotations Related to Requirement Rejection by the CCB.

Identification	Allowed Values	Explanation
Evaluation	Free Text	Evaluation field extended with reason for rejection.

Table A- 8 Annotations Related to the Requirement Planning.

Identification	Allowed Values	Explanation
Release	Release ID	Status changes automatically to Planned when a requirement is assigned to a release.
Test Coverage	Link to Test Case(s)	Each detailed requirement needs to a link to at least one test case, to ensure that the described functionality is covered by tests.

Tasks	Links to associated tasks	Each requirement is implemented according to a number of tasks descriptions that are assigned to the developers. All tasks have to be defined before the requirement can obtain the status of Planned

Progress

No required fields. Status and actual effort updated according to task progress.

Completed

No required fields. Status and actual effort updated according to task progress.

Change Request Workflow (CRW) annotations

Annotations for the Change Request Workflow is provided in the tables below.

Table A- 9 Annotations Related to a Change Request

Identification	Allowed Values	Explanation
Incident ID	Number	Unique ID (Auto-generated)
Name	Free text	The incident title
Description	Free text	Narrative description of the incident. Include attachments for supporting information. Attachments may include all document types. Often it will be use cases, activity diagrams and other information that refines the requirement. The description field may also include information on: Solution: Envisaged solution of the issue. Verification/Verification method: how to test the issue.
Detected by	[user name]	Name of the user who entered the issue.
Detected on	Date	Auto-generated at the time of registration
Severity	1 (Critical show-stopper) 2 (Major problem) 3 (Minor problem) 4 (Nice to have)	Initial assessment of the impact of a problem. May be updated later, when a solution has been identified.

Table A- 10 Annotations Related to Reject (Dublicate/Incomplete/Invalid)

Identification	Allowed values	Explaination
Resolution	Free text	Reason for rejection
(Associations)	Linkt to related issue(s)	Link to dublicates, or other related issues that may have influence on the decision to reject the change.

Table A- 11 Annotations Related to Change Request Evaluation.

Identification	Allowed values	Explanation
Severity	- Critical (Critical show-stopper) - High (Major problem) - Medium (Minor problem) - Low (Nice to have)	Initial assessment of the impact of a problem. May be updated later, when a solution has been identified.
Owner	[user name]	The responsible PAM
Estimated effort	Hours	The initial estimate of the cost that are associated with the implementation of a change.

Evaluation	Text	A textural description of the implications and risks that are associated to the implementation of the requirement, along with an envisaged solution. The Evaluation should provide the CCB with sufficient information to approve/reject the requirement.
Risk	- High - Medium - Low	The risk associated to implementing the proposed change.

Table A- 12 Annotations Related to a Change Request Rejection by the CCB

Identification	Allowed Values	Explanation
Evaluation	Free Text	Evaluation field extended with reason for rejection.

Table A- 13 Annotations Related to a Change Request Approval by the CCb.

Identification	Allowed Values	Explanation
Priority	Critical	Shall be implemented
	High	
	Medium	
	Low	May be implemented if time

Table A- 14 Annotations related to Planning of a Requirement Change

Identification	Allowed Values	Explanation
Start Date	Date	The planned start date
Associations	Link to requirement	At least one link to new requirements that have been established as a consequence of the requested change.

Defect Workflow (DW) annotations

Annotations for the Defect Workflow is provided in the tables below.

Table A- 15 Annotations Related to a Defect Registration

Identification	Allowed Values	Explanation
Incident ID	Number	Unique ID (Auto-generated)
Name	Free text	The incident title
Description	Free text	Narrative description of the incident. Include attachments for supporting information. Attachments may include all document types. Often it will be use cases, activity diagrams and other information that refines the requirement. The description field may also include information on: Solution: Envisaged solution of the issue. Verification/Verification method: how to test the issue.
Detected in release	Release ID	The release that the defect was detected in
Detected by	[user name]	Name of the user who entered the issue.
Detected on	Date	Auto-generated at the time of registration
Severity	1 (Critical show-stopper) 2 (Major problem) 3 (Minor problem) 4 (Nice to have)	Initial assessment of the impact of a problem. May be updated later, when a solution has been identified.

Table A- 16 Annotations Related to Rejection of a Defecect (dublicated, incomplete)

Identification	Allowed values	Explaination
Resolution	Free text	Reason for rejection
(Associations)	Linkt to related issue(s)	Link to dublicates, or other related issues that may have influence on the decision to reject the change.

Table A- 17 Annotations Related to Defect Evaluation

Identification	Allowed values	Explaination
Severity	Critical (Critical show-stopper) High (Major problem) Medium (Minor problem) Low (Nice to have)	Initial assessment of the impact of a problem. May be updated later, when a solution has been identified.
Owner	[user name]	The responsible PAM
Estimated effort	Hours	The initial estimate of the cost that are associated with resolving the defect.
Evaluation	Text	A textural description of the implications and risks that are associated to the resolution, along with an envisaged solution. The Evaluation should provide the CCB with sufficient information to approve/reject the defect.
Risk	High Medium Low	The risk associated to resolving the proposed defect.
Associations	Link to requirement	There has to be at least on link to a requirement that is affected by the defect. There shall be at least one test case that captures the defect. The test cases that are associated to the associated requirements shall be used to verify the fix.

Table A- 18 Annotations Related to Rejection of a Defect.

Identification	Allowed Values	Explanation
Resolution	Free Text	Reason for rejection

Table A- 19 Annotations Related to CCB Approval.

Identification	Allowed Values	Explanation
Priority	Critical	Must be resolved
	High	
	Medium	
	Low	May be resolved if time

Progress

No required fields.

Table A- 20 Annotations related to Defect Completion (fix).

Identification	Allowed Values	Explanation
Resolved in release	Release ID	The release that contains the fix.
Resolution	Free text	A textural description of the fix.
Closed Data	Date	Automatic set by SpiraTeam
Actual effort	Hours	The actual hours spend by resolving the defect.

Table A- 21 Annotations Related to a Failed Verification

Identification	Allowed Values	Explanation
Resolution	Free text	Reason for failure.

Table A- 22 Annotations Related to Verification

Identification	Allowed Values	Explanation
Resolution	Free text	Describe the verification

Appendix B. Project Planning

B.1. Estimation

Each time a new requirement is established in the requirement management system, the estimated effort shall be estimated.

When an exsisting requirement is decomposed into a number of refined requirements or low-level project tasks, each refined requirement/task is estimated. The sum of the estimates of the refined requirements is compared to the original estimate of the parent requirement and if the deviation is larger than an agreed tolerance the Program Management is informed and the project plan revised.

Hence, the project plan will at any time represent the level of detail to which the requirements has been refined.

The Product Area Manager shall together with the Lead Developer estimate the required effort based on the following formula:

- The development effort for each work package in the project is estimated in isolation (3-point estimation)

- The project management effort is estimated to 50 pct of the development effort

- The test and documentation effort is estimated to 50 pct of the development effort

Note: a feature project should generally not be planned for lasting longer than 2 calendar months and a work package within a feature project should not be planned for lasting longer than 2 calendar weeks.

At entry to each major development cycle the Project Plan shall be baselined.

B.2. Meetings

The following project meetings:

- W-meetings – weekly informal program management meetings. No minutes taken.
- M-meetings – monthly program management status meetings with fixed agenda. Minutes taken.
- F-meetings – weekly feature project meetings (Project Manager and Lead Developer). No minutes taken.
- SC-meetings – Steering group meetings. Minutes taken.

Appendix C. Software Project Tracking and Oversight

C.1. Program Management Level

The Program Management Group will in concert with the M-meetings (see Appendix B Project Planning)

- Monitor progress and milestones
  Assess the project progress and cost in relation to the planned values.

- Monitor resources provided and used
  Assess the performance and knowledge of the resources allocated to the project.

- Monitor project risks
  Evaluate previous identified risks and trigger mitigation plans if applicable.

- Take corrective actions
  Corrective actions can be taken at the following 3 levels

  a. The Program Management level, e.g. by re-shuffling resources or shifting requirement implementation from one feature implementation project to another.
  b. The Company level, e.g. by adding additional resources to the project for a shorter or longer period. This requires accept by the Steering Committee.
  c. The Project level by changing the Project Plan. This requires accept by the Change Control Board.

The above will be documented in status reports submitted to the Steering Committee.

C.2. Feature project Level

The Project Leader of a Feature Project will issue bi-weekly status reports to the Program Management Group. The status reports will contain:

- Highlights of activities performed during the reporting period
- Risks and mitigations
- Qualitatively progress status
- Quantitatively progress status

The Program Management Group will use the Feature Project status reports as input to the Program Management Level status reports.

Appendix D. Software Quality Assurance

The project follows:

- The standard DHI quality assurance guidelines for consultancy work.
- A software quality assurance plan that shall verify that the project is executed in accordance with the process documented in said document and produce deliverables of acceptable quality

The software assurance plan consist of:

1. Figure 2 on page 5) having the following mandate

    a. Can at any time perform check of project deliverables – internal as well as external
    b. Has direct reporting links to the DHI Project Director and DHI Quality Assurance manager

2. A Modus Operandi of the Software Quality Assurance function defining

    a. Random check on production of deliverables as defined in said document
    b. Random check on content deliverables

        i. Does the deliverables appear as being constructed in a thorough manner
        ii. Does software implementation follow the standard defined in Appendix G. Coding Practices and Standard
        iii. Is the software implementation in accordance with the produced documentation
        iv. Is the software implementation in accordance with standard best practices

3. The Software Quality Assurance function shall produce a minimum of (referring to bullet 2 above)

    a. Eight inspections of type a per development cycle
    b. Four inspections of type b.i per development cycle
    c. Four inspections of type b.ii per development cycle
    d. Two inspections of type b.iii per development cycle
    e. Two inspections of type b.iv per development cycle

4. The Software Quality Assurance function shall without delay report detected deviations to the Team Leader and Senior Software Engineer.

5. The Software Quality Assurance function shall produced an inspection report to the Team Leader and Head of Steering Committee once every 4 months

6. Appointing the role as Software Quality Assurance Manager to Technology Lead Niels K Olsen of the DHI Software Products department (the MIKE software). Mr Olsen has been with DHI for approximately 20 years and has during this period worked with both the standard products and custom development. Prior to working at DHI Mr Olsen has worked as EN 45001assessor.

Appendix E. Configuration Management

The purpose of Configuration Management is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

The table shown below identifies the artifacts that shall be placed under configuration management. The table also identifies the systems used for controlling the products.

Table E 1 -Configuration items

CI Group	Specifics	System
Requirements	All requirements **Changes** Changes to requirements shall be handled with the Change Request Workflow Responsible: Relevant PAM on behalf of the CCB. **Baseline** Requirements shall be baselined at the following events: • After CCB meetings – all requirements targeted to be included in the final version • At entry to a new development cycle – all requirements targeted to be included in the cycle • At entry to a feature development project – all requirements targeted to be included in the feature development project **Responsible**: Relevant PAM	SpiraTeam
External documents	Documents submitted to the customer as part of the delivery plan **Naming convention**: See Appendix F. Document Naming Conventions on page 40. **Baselining:** When accepted by the customer **Responsible:** Team Leader.	SharePoint
Internal documents at the Program Management level	Documents related to the control of the project **Naming convention**: See Appendix F. Document Naming Conventions on page 40 **Baselining:** When entering a new development cycle or as consequence of CCB changes. **Responsible**: Team Leader	SharePoint

CI Group	Specifics	System
Internal documents at the Feature Project level	Documents related to the control of a single feature implementation project **Naming convention:** See Appendix F. Document Naming Conventions on page 40 **Baselining:** At feature project entry and exit **Responsible:** Program Manager	SharePoint
Code	All code **Changes** Developers check-in and out from the system. **Coding style:** See Appendix G. Coding Practices and Standard. **Baselining:** The code shall be baselined at the following events: • When a feature project is closed – all files created or modified during the project • When a prototype or final version is delivered to the customer – all the code • When a prototype or final version is accepted by the customer – all the code **Responsible** • Relevant PAM after feature project close • Senior Software Engineer after delivery of prototypes and final versions.	Surround
3rd party components **Tools** **Build specifications**	All 3rd party components All tools used during construction like compilers and Object-Relational mapper All specifications used in building the software These configuration items are managed in the MZVC system. **Changes** Changes happen according to CCB decisions. **Baselining:** After each change **Responsible:** Program Manager.	MZVC

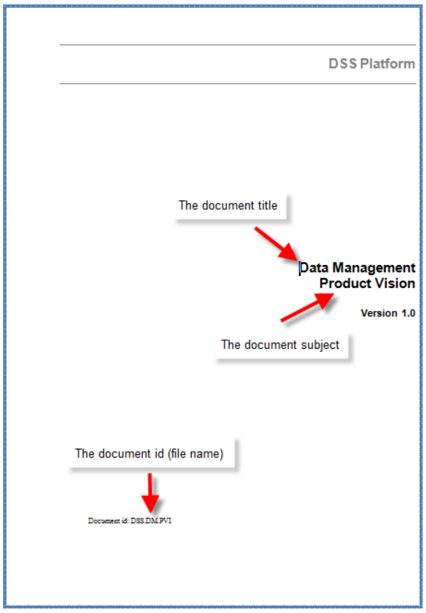
CI Group	Specifics	System
Requirement traceability	Tracing between requirements and produced derivatives. 1. Requirement and test plans and test case document – i.e. a mapping of where test specification for a specific requirement is documented 2. Requirement and source code component – i.e. a mapping of the source code that constitutes the implementation of a specific requirement 3. Requirement and feature project – i.e. a mapping of the feature projects that implemented a specific requirement **Baselining:** At exit of each development cycle. **Responsible**: Program Manager	SpiraTeam
Release Numbering	DHI uses a four number software release numbering scheme: (Major).(Minor External).(Minor Internal).(Iteration) The major version increments one each time a major release is handed over to the client. The Minor External number increments by one each time a minor release (i.e. a service release) is handed over to the client. The minor external is incremented by one for each internal minor release (typically used for test purposes e.g. at the end of a minor release cycle). Finally, the last digit represents the release of an internal test version on a sub-minor release time scale.	

Appendix F. Document Naming Conventions

The document naming standard covers the following document properties:

- The file name
- The document title
- The document subject

The figure below illustrates the location of the three properties on the document frontpages.



F.1. Project-level Documents

Project documents follow the name convention:

- *File name:* *<Project-id><Type-id><type-name>*
- *Title:* *<Project-id>*
- *Subject:* *<type-name>*

- Type ids and type names are defined in the following table

Table F1 – Document Type-ids

Type-id	Type-name	Short Description
SRS	Software Requirement Specifications	Requirement analyses
SAD	Software Architecture Document	Conceptual and logical design
PVI	Product Vision	Description of the development groups vision for the system or product area
PBD	Product Breakdown	Breakdown of the system or product area into functional areas
PHA	Project Handbook	This document
PPL	Project Plan	The project plan. Refined and revised throughout the project life cycle.
SRE	Status Report *<date>*	
TPL	Test Plan	Test plan incl. test specifications
SUM	Software User Manual	

Example: *"DSS.SRS – Software Requirement Specifications"(if Project-id = DSS)*

F.2.Feature Project Level Documents

The documents at this level follow the naming convention:

- File name: <Project-id>.<fp-id>.<Ftype-id><Ftype-name>
- Title: <fp-name>
- Subject: <Ftype-name>

where *<fp-id>* and *<fp-name>* are feature project specific short and long names. The feature project names are defined by the respective product area manager.

Predefined type-ids and type-names are defined in the following table.

Table F 2 – Feature Project ids and names

Ftype-id	Ftype-name
PPL	Feature Project Plan
SRS	Software Requirement Specification
SRE	Status Report *<date>*
SAD	System Architecture Documentation
TSP	Test Specification

Example: “*DSS.FP42.SRS – Software Requirement Specifications*” (if Project-id = DSS and fp-id = FP42)

Appendix G. Coding Practices and Standard

General coding standards to be followed:
http://www.idesign.net/idesign/download/IDesign%20CSharp%20Coding%20Standard.zip

Specific conventions to be followed:

1. Use 4 spaces for indentation, i.e. no tabs in the source code

2. Namespace convention

   ```
   <root>.<Software component>.<tier-name>
   ```

   - `Root` is always `DHI.Solutions`
   - `Software component` is `the name of the software component that the namespace represents.`
   - `Tier-name` is one of the following: `Data, Data.Services, Business, Business.Services and UI`
   - Below `DHI.Solutions.System` is located a number of system namespaces. They might or might not include the tier-name part in their namespace name.

3. Each source file shall include the following copyright note.

   ```
   /*
    * Copyright © 2009, DHI
    * All rights reserved.
    * http://www.dhigroup.com
   */
   ```