“LR Guidelines for the Development of Dependable Computer-based Systems”

P. Katsoulakos F. Maclellan R. Shaw

Applied Information Engineering
Lloyd’s Register, Lloyd’s Register House
29 Wellesley Road, Croydon
CR0 2AJ, United Kingdom

Abstract

The paper describes the framework of the Lloyd’s Register Guidelines for Dependable computer-based Systems (LRGDS). It defines the development approach adopted by the guidelines, the industrial needs that they address and the technical requirements it attempts to satisfy. The LRGDS model focuses on how the management of software development should be handled through the production of appropriate plans. Based on these plans assessment of whether the development process has been carried out properly can be undertaken and fitness for purpose established.

1 Introduction

The capability of computer systems to offer ever increasing functionality is transforming all sectors of industry including those traditionally served by Lloyd’s Register (LR) – shipping, offshore and industrial. Without programmable electronics little on a modern ship, an offshore platform or chemical process plant would be able to function. LR is concerned with the assessment, approval and certification of ships and other engineering systems. Independent assessment or certification is carried out against the LR rules for the design, construction and operation of ships. Other international and national standards are also applied within industrial sectors other than shipping.

Software is increasingly used in control and management systems, where failure of the system to operate correctly could threaten life, damage the environment or cause large financial loss. There is a need therefore for LR to extend its traditional assessment, approval and certification services to cover computer based systems. For this purpose the LRGDS project was set up and the early results of its first phase are reported in this paper.

2 LRGDS Development Approach

The development of LRGDS may be viewed from the perspective of what has been called
a methodological pyramid [W+89], that is, a description based on a number of layers. The layers that we have found useful are as follows:

**Scope:** This defines the perspective of the domain to which LRGDS applies clarifying the boundaries and assumptions.

**Models and Theory:** Provides a framework for LRGDS by:
- Highlighting the characteristic properties of the guidelines
- Ensuring proper integration of a wide range of methods and techniques within the structure provided by the LRGDS framework.

**Methods and Techniques:** Describe in detail the appropriate methods and techniques derived from the LRGDS model.

**Tools:** They provide support for the practical application of LRGDS in software development.

**Case Studies:** They provide the means of guiding and testing developments.

In the following two sections we will outline the scope of LRGDS (that is the influences that have impacted upon its development and its potential areas of application), and outline an aspect of the development approach that is proposed.

### 3 Scope

#### 3.1 Software Focus

Software and digital systems in general provide solutions for complex functions often at the expense of introducing increasingly subtle and complicated failure modes. Further, the discrete discontinuous nature of software makes the prediction of behaviour difficult and is often counter to traditional engineering intuition. Many sophisticated software based systems are far too complex to test exhaustively and must be assessed by using a variety of specialist techniques linked to the development process in a suitable manner. Software is therefore at present the weak link of computer-based systems and is the area of attention for LRGDS.

#### 3.2 Dependability and Fitness for Purpose

LRGDS will be a set of guidelines to help achieve and assess fitness for purpose, that is safety, security, reliability, usability, performance, availability and maintainability. Special attention is placed on dependability defined as that property of a system which allows reliance to be justifiably placed on the service that it delivers. The service delivered by a system is its behaviour as perceived by its users. During the life of a system the user will witness proper service when the service conforms to its specification.

#### 3.3 Quality Approach

The concept of quality in LRGDS follows largely the principles identified in standards produced by the European Organisation for Quality Control (EOQC), the American Society of Quality Control (ASQC) and ISO/TC/176 on Quality Assurance. Quality is defined as "the totality of features and characteristics of a product or service that bears on its ability to satisfy given needs". Accepting this definition it follows that attaining a "high level of quality" does not mean the maximising of certain quality measures, but rather selecting quality criteria relevant to a specific appli-
cation and then achieving the required level in the corresponding quality metrics. Given this clarification, it is then important that quality is built into the deliverables that are to be produced and controls are defined for monitoring, and achieving adequate quality.

3.4 Genericity and Extensibility

LRGDS is intended to be a generic set of guidelines promoting fitness for purpose in software based systems. Much effort in software development methods and standards has been conducted in the context of particular application domains and industrial sectors. Military, nuclear power, aerospace, avionic and telecommunication driven developments have established a number of local and specific practices. However, it is believed that much commonality exists, which is desirable and feasible to exploit. LRGDS is therefore aimed at achieving such a goal. The generic nature of LRGDS should be based on appropriate classification of software. Once the initial guidelines have been produced, they should be expanded to cover different areas of emphasis of software by identifying appropriate concerns (ie. performance, safety, usability etc). For each concern, key factors that should be considered and managed will be defined. Different factors will have a different level of importance depending on whether the software is safely related, real time, security related, etc. These factors could then be matched to the relevant methods, techniques and tools which will be identified in a supplement to the guidelines which will provide suggestions on these topics.

LRGDS needs to be easily extendible to provide industry specific guidelines. The LRGDS guidelines will eventually be incorporated into the LR rule book for ships to provide the basis for software approval. In other areas LRGDS should be easily customised to incorporate the local requirements from standards or practices.

3.5 Relationship to International and National Standards

The standardisation process, Figure 1, involves the development of standards and techniques to evaluate these standards. Standards are produced by international and national bodies often helped by results of pre-standardisation activities. LRGDS is aimed at encompassing the requirements of existing standards. This places important flexibility requirements on the formulation of LRGDS. For example, different issues are stressed by current standards. Thus, for example, while there is an insistence on the use of formal methods and safety analysis techniques in Interim Defence Standards 00-5.5 and 00-56 [MODS89b, MODS89a] and the IEC Recommendations [IEC89a, IEC89b] the ESA standard [ESA87] places no such strictures on its users and concentrates on defining a purely procedural framework. This does not create a problem for LRGDS which while requiring that appropriate standards are employed lays stress on the objectives that the employed standards must address.

3.6 The need for Certification

There is a well established need for a seal of approval publicly establishing that a product has been produced according to a well recognised standard. This seal of compliance may be provided by a “certificate” which can be awarded by a recognised organisation. Lloyd’s Register is capable of providing such a certification service. For this purpose assessment and certification against LRGDS by LR should be
Figure 1: The Role of LRGDS in standardisation
recognised by the statutory authority responsible for supervising the relevant application specific standard.

4 Theory and Models

4.1 Philosophy

In developing the LRGDS we have not sought to focus on how software should be constructed, for instance by prescribing the use of specific methods and techniques. Rather the focus is on key issues that should be addressed and planned for in a project development. The principle behind this is that if it can be demonstrated that all important issues have been carefully considered and planned for then the project is more likely to be successful than it would otherwise be. It is equally important that the personnel involved are familiar with the strategy to be adopted, and the reasons for it, and are both aware of their individual responsibilities and sufficiently skilled to perform their allotted tasks.

Assessment of a project or product then involves checking that the issues identified have been addressed adequately, that the justifications for decisions made are valid and that the plans have been followed. There is an implicit need for these matters to have been recorded. This can only be performed by personnel with experience of the type of system being produced and hence the ability to make a judgement on the approach taken.

It is envisaged that a supplementary guide will be produced which will recommend particular methods, techniques and tools for certain classes of system. Obviously, such a guide will have to be updated regularly if it is to keep pace with changes in technology.

In order to capture, represent and manipulate the diverse range of concerns depicted in LRGDS, a knowledge-based representation method - the Concern Net Approach [Wil90] is being investigated. The concern net would encapsulate all the concerns of the various interested parties (the various members of the development team as specified in the plans and the customers) and help both to identify interacting concerns and to clarify the design-trade off decisions.

4.2 The Three-Layered Model

The model that describes the LRGDS view of computer-based system development is the three-layered model illustrated in Figure 2, the three layers being:

- Problem Definition
- Plans
- Activities, Deliverables and Quality Factors

Problem Definition: At the outset, the problem to be solved has to be defined. The characteristics of the system (e.g. whether it has safety or security implications, is real-time, a database application, has significant man-machine interaction etc.) and any risks and uncertainties (e.g. safety hazards, potential security violations, imprecise requirements, enforced use of unfamiliar technology) are identified.

Plans: A set of plans are produced which define the chosen strategy for the development (taking into consideration the identified risks and uncertainties), and the procedures and controls that will be used. There are a number of issues that these plans should address.
will be performed, their sequence and timing, the deliverables that will be produced and the quality factors that these should exhibit (some of which will be objective, others subjective). Some of the activities will be concerned with producing the deliverables, others with checking that the defined quality factors have been achieved in those deliverables. The deliverables include not only the specifications, designs and code of the system, but also the test specifications, test reports and review reports. The plans should define the quality factors for each of the deliverables.

4.3 The LRGDS Framework

The guidelines concentrate on plans that should be produced detailing how the project is to proceed, how it will be managed and how quality will be built into the system produced. They also highlight the importance of being able to justify decisions made.

The guidelines identify a number of plans that should be produced at the start of a project (see Figure 3). The three major ones are:

- the Management Plan
- the Technical Plan
- the Quality Plan.
The Management Plan details the costing and resourcing of the project. It is concerned with ensuring that the right personnel are available at the right times, having had appropriate training, and are supplied with appropriate facilities. It identifies key milestones within the project when certain of the project deliverables will be produced. It is built from the Technical and Quality Plans.

The Technical Plan details the technical strategy that will be adopted in order to ensure that quality is engineered into the system produced. It is concerned with choosing an appropriate process model and appropriate methods, techniques and tools, taking into consideration any risks and uncertainties in the project as well as the characteristics of the system to be produced.

The Quality Plan defines the quality features in the deliverables and the means of checking for them. It is concerned with the way in which verification will be performed, the review, configuration management, change control and fault management procedures.

Other plans that may be produced if the project is of sufficient criticality are an independent verification and validation plan, an independent assessment plan and special issue plans.

Figure 3: LRGDS plans
plans, for instance safety, security and usability plans.

In what remains of this paper we will limit the discussion to the technical plan as this illustrates sufficiently the LRGDS approach.

4.4 Technical Plan

As stated above, the Technical Plan details the technical strategy that will be adopted in order to ensure that quality is engineered into the system produced. That strategy should take into consideration any risks and uncertainties in the project as well as the characteristics of the system to be developed so that the chance of producing a system that is not fit for its purpose is minimised. The production, maintenance and distribution of the Technical Plan is the responsibility of the designated Chief Designer, who has overall responsibility for all the technical aspects of the project.

The Technical Plan should:

- describe the process model (see Section 4.4.1) to be adopted; the production of the project deliverables (see Section 4.4.2) and the way in which the identified risks and uncertainties will be managed within this process should be clearly indicated.

- identify the methods and techniques that will be used and the rationale for choosing them; each one should have been chosen because of the appropriateness of its computational model to a particular characteristic of the system, taking into account the level of formality of the method and its verification potential in relation to the criticality of the system.

- identify the tools that will be used to support the methods chosen and the programming language to be used for implementation.

- identify the target, development and maintenance environments in terms of both hardware and software components.

- identify the training that will be required to ensure that the personnel participating in the development have the requisite skills.

One of the activities that should be identified in the process model is that of Test Plan Generation. The Test Plan is supplementary to the Technical Plan and details the strategy for the integration and testing phase. The Test Plan should:

- define the strategy for integrating units of code into various levels of sub-system until the full system is integrated.

- define the methods to be used to test units of code, each level of sub-system and the full system; the method for a given level should be based on a functional and/or structural approach; methods for testing non-functional requirements (e.g. performance, reliability, maintainability) should also be defined.

- define what each level of integration is to be tested against and the coverage that should be achieved.

- define the test environment required for testing, that is the hardware, software (including any test harness), test tools, documentation, test data.

- specify what other test documentation is to be produced (in particular test specifications and test reports) and provide sufficient information to enable this to be done.
4.4.1 Software Development Process Model

The process model chosen in the Technical Plan will comprise a collection of activities that are linked together by dependence relations with each one generating outputs which are used as inputs to other activities. The exact “shape” of the process model will depend on the type of project being undertaken but it should normally include at least one instance of each of the following activity types:

- Concept formulation
- Feasibility and requirements capture
- Specification
- Design and implementation
- Test Plan Generation
- Integration and Testing

Other activities that might occur in the process model include prototyping.

For each of the activities identified in the process model the following should be specified:

- the inputs to and outputs from the activity.
- the entry conditions in order for the activity to proceed and the exit conditions which indicate completion of the activity.
- the preceding and succeeding activities.
- the reviews to be undertaken.
- the methods and tools to be used.
- the skills required by personnel performing the activity.
- the role responsible for the activity.

At the end of a development activity the deliverables produced or expanded upon should be verified against the deliverables taken as inputs, for instance a component specification should be verified against the system specification from which it is derived.

4.4.2 Deliverables

The system development should result in the production of a set of deliverables, most of which can be regarded as contributing to one of the following four system models:

- User Model
- Architectural Model
- Implementation Model
- Build Model

Each one describes a different view of the system.

The User Model describes the user’s view of the system and as such should comprise: such deliverables as the statement of requirements, system specification, the acceptance test specification, the operational system and any user guides, user manuals, user training etc.

The Architectural Model describes the system in terms of the logical structure of its control and data and how the functional structure of the system is mapped onto the target environment. It should comprise an architectural design, specifications of the major components, test specifications for these and the completed components.

The Implementation Model describes the way in which the architecture is actually implemented on the target hardware and software. It should comprise specifications of physical data structures, module specifications and test specification and tested modules.
The **Build Model** describes how the system of communicating processes is built from the source code and should comprise files required to construct processes/programs from source code, scripts for installation, system start-up, process initialisation and system control.

The choice of methods for the project will determine the items that will comprise the system models and hence how the quality of those items will be defined and verified to ensure the quality of the overall system.

Other deliverables that will be produced include test reports and review reports.

The quality factors that the deliverables should exhibit will be of one of two types, functional and non-functional. Functional factors and some of the non-functional ones will be derived from the system specification. However, the following non-functional quality factors should be exhibited by most of the deliverables:

- **Completeness** - in a specification, all functional and non-functional feature should be identified; a test specification should contain tests for all the features of the specification it is testing.

- **Precision** - the information in a deliverable should be unambiguous.

- **Consistency** - none of the features should be contradictory, and a specification should be functionally equivalent to the specification from which it was derived.

- **Clarity**.

- **Understandability** - a specification should be understandable to both implementors and users of the module, component or system; a test specification should be understandable by the personnel performing the tests.

- **Testability** - a specification should be written so that each feature can be tested.

- **Traceability** - a specification should identify the system requirements that are being met and a test specification should identify the system requirements being tested.

- **Compliance with relevant standards** - all specifications, designs, code and test specifications should comply with the standards defined for them.

- **Modifiability** - any deliverable should be well-structured with no redundant information.

- **Maintainability** - all design decisions, limitations and possible enhancements, and any dependencies should be detailed.

5 **The Remaining Levels**

Issues relating to specific methods and techniques that may be applicable in addressing the objectives laid down in LRGDS will be covered in a supplementary guide. It is not intended that the guide be a comprehensive survey of existing methods, techniques and tools, for example such as the NCC’s STARTS Guide [NCC89]. Rather it will contain specific information on methods and techniques that LR has investigated as part of its on-going research activities.

It is planned that case studies in the form of development projects using all or part of the guidelines will be employed to evaluate and improve the proposals.
6 Conclusions

LRGDS is a set of guidelines for the development of dependable systems which should enable Lloyd's Register to carry out formal assessment and certification of software based systems.

Because the guidelines are not aimed at a specific industrial sector or application the emphasis will be to ensure that objectives (planning, technical, quality, safety etc) are met, rather than to prescribe approaches that may address them. For this reason attention has been directed towards the planning of a project's development. The guidelines thus require the existence of a set of plans which should detail how these objectives will be satisfied. This also implies that assessment can begin at project inception and prevent wasted effort in using inappropriate methods.

Acknowledgements

The authors would like to thank the Committee of Lloyd's Register for permission to publish this paper.

References


