DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Joint Software Reviews - A Case Study from JP2030

Gina Kingston

DSTO-RR-0156

Joint Software Reviews - A Case Study from JP2030

Gina Kingston

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-RR-0156

ABSTRACT

Joint Software Reviews - between acquirers and developers - are an important approach in the acquisition of software intensive systems. These reviews are poorly understood and often conducted in an inefficient, *ad hoc* manner. This report describes some aspects of the design review for Project Llama (JP2030).

Information Technology Division were asked to provide input to this design review and formed a multi-disciplinary team to assess the design. This document describes the process attempted by the ITD review team and compares this to the actual process used. The benefits and limitations of the process are discussed as well as potential improvements.

A survey of participants at the design review meeting was conducted and the results of this survey are also included.

RELEASE LIMITATION

Approved for public release

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-011-020
July 1999

APPROVED FOR PUBLIC RELEASE

Joint Software Reviews - A Case Study from JP2030

Executive Summary

Joint Software Reviews are an important approach to quality control in the acquisition of software intensive systems. These reviews are poorly understood and often conducted in an inefficient, *ad hoc* manner. This report uses the architecture review for Project Llama (JP2030) as a case study to investigate how joint software reviews could be improved.

Information Technology Division (ITD) were asked to provide input to the architecture review for project Llama and formed a multi-disciplinary team with experts in Software Engineering, Human Factors and Geographical Information Systems to assess the design.

This paper uses information from: 1) the ITD review, 2) survey responses from the participants in the architecture review for Project Llama and 3) information from a literature search to identify the strengths and weaknesses of current, and proposed, review processes. One of the major limitations of current reviews is that the review process is not standardised, so many are *ad hoc*. Consequently, a document which addressed all of the issues raised in one review would not necessarily pass a subsequent review.

An alternative goal-driven approach is defined. This approach uses a three-dimensional evaluation framework. The three dimensions of the framework comprise Knowledge Domains, ViewPoints and Criteria. Knowledge Domains captures areas of expert knowledge, Criteria captures common evaluation criteria such as traceability. The final dimension, ViewPoints, captures different perspectives of the system. There are five perspectives: an Enterprise or holistic viewpoint, a Technology viewpoint, an Engineering viewpoint, a Computational viewpoint and an Informational or user driven viewpoint. The issues raised in the review of Project Llama were used to produce a preliminary checklist of questions, which may need to be addressed in other projects. This list will be refined in future case studies. It will need to be customised for other projects, taking into account the areas of technology, engineering, and management, which the project covers. Mechanisms for evolving this framework are also discussed.

Further concrete recommendations are given about other mechanisms to improve joint software reviews. These include: changes to the review procedures to allow for more flexible management of the attendees - particularly users - for shorter periods of time; improved training in review procedures; and developing procedures for integrating material from several information sources.

Authors

Gina Kingston
Information Technology Division

Gina has been employed in the Software Systems Engineering Group of the Information Technology Division of the Defence Science and Technology Organisation (DSTO) since graduating from the University of Tasmania with a BSc with First Class Honours in Mathematics in 1990. Her research interests include process improvement and empirical software engineering. She has conducted research into software costing, the analysis of software, and software acquisition. She is currently undertaking a PhD in Software Reviews through the School of Information Systems at the University of New South Wales and working on the JOint Reviews of Software (JORS) and Iterative Capability Development (ICD) tasks at DSTO.

Contents

Table of Abbreviations

Abbreviation	Description
ADF	Australian Defence Force.
ADI	Australian Defence Industries. A commercial organisation.
ADO	Australian Defence Organisation. This incorporates both the ADF and the ADOD.
ADOD	Australian Department of Defence.
COTS	Commercial-off-the-shelf.
CSS	Command Support System.
DSTO	Defence Science and Technology Organisation.
ITD	Information Technology Division.
JCSE	Joint Command Support Environment.
MILGEO	A situation monitoring system.

1. Introduction

Joint Software Reviews are an important approach to quality control in the acquisition of software intensive systems. These reviews are poorly understood and often conducted in an inefficient, *ad hoc* manner. This report describes some aspects of the design review for project Llama (JP2030).

Information Technology Division (ITD) were asked to provide input to this design review and formed a multi-disciplinary team to assess the design. This provided the opportunity to investigate joint software reviews and to consider how they might be improved. This document describes the goal-driven process attempted by the ITD review team and compares this to the process actually used by the ITD review team, where people automatically focussed on their areas of interest and expertise. The benefits and limitations of both processes are discussed as well as potential improvements.

A survey of participants at the design review meeting was conducted and the results of this survey are also included.

This document contains background information on Joint Software Reviews (Section 2); information on the project Llama case study (Section 3), including background information on project Llama, a discussion of the review process undertaken by the ITD review team and results from a survey of the design review participants; an alternative review method (Section 4) and a series of recommendations for future reviews (Section 5).

This document is the second DSTO paper in a series of studies on Joint Software Reviews. Information about other papers in this series can be obtained from the author.

2. Joint Software Reviews

> *"A process or meeting involving representatives of both the acquirer and the developer, during which project status, software products, and/or project issues are examined and discussed"* [MIL-STD-498, 1996].

Joint software reviews form an important part of the Defence Acquisition Process [MIL-STD-1521B, 1985; MIL-STD-498, 1994; CEPMAN 1, 1996; Gabb, 1997], and with the growing popularity of outsourcing, they are becoming more important in the commercial sector [ISO/IEC 12207, 1995].

Like other forms of software review, joint software reviews offer a means to evaluate the product being developed, to evaluate the development process and progress, and to identify risks early in the acquisition. Despite this, software-intensive systems are often delivered late, over-budget, and with sub-optimal functionality [Earnshaw, 1994; ADO, 1996; Mosemann II, 1995; Keil, 1995; Heemstra, 1992; Lederer and Prasad, 1995; Canale and Wills, 1995; Walsh, 1994].

Furthermore, anecdotal evidence obtained during interviews with DSTO (Defence Science and Technology Organisation) and ADF (Australian Defence Force) personnel, including those with considerable experience with joint software reviews, suggests that these reviews are considered to be inefficient by many of their participants.

2.1 Joint Review Processes

A Joint Technical Software Review is a software review process usually undertaken during system acquisition that aims to [MIL-STD-498, 1994]:

> "*a. Review evolving software products... review and demonstrate proposed technical solutions; provide insight and obtain feedback on technical effort; (bring to the) surface and resolve technical issues.*
> *b. Review project status; (bring to the) surface near- and long- term risks...*
> *c. Arrive at agreed-upon risk-mitigation strategies...*
> *d. Identify risks and issues to be raised at joint management reviews.*
> *e. Ensure on-going communication...*"

Very little has been documented on joint software reviews. Military standards discuss when, where, and why joint reviews should be held [MIL-STD-1521B, 1985; DOD-STD-2167A, 1988; MIL-STD-498, 1994]. There are, however, few guidelines on how to conduct joint software reviews: there is little discussion on the roles of participants and no debate on how issues should be raised or resolved. Where guidelines on how to conduct reviews are available, they are general in nature: recommending that minutes be taken and that the meeting is co-chaired and providing guidelines on what documents and activities should be assessed at the same review etc.

There is little evidence - either theoretical, or empirical - to support the conduct of joint software reviews in one manner over another or to support any of the existing guidelines.

Consequently, the joint software review process is often *ad hoc*. Interviews conducted by the author with Defence and DSTO personnel in early 1997 to gain insights into current review processes indicated that:

- the material under review may or may not be received prior to the meeting, it may or may not be complete, and the time available to review the documents may vary from a few days to a few weeks;
- prior preparation - either familiarisation or issue identification (as commonly found in inspections) - may or may not be conducted;
- people who review material prior to the meeting may either forward their comments or attend the meeting;
- people who review material prior to the meeting may or may not be given guidance as to the types of issues they are to try and identify;
- reviewers are normally not given guidance or training on how to review material but may use criteria which they have devised;
- there are usually more participants at joint software reviews than at typical inspections (for example, there were 7 participants from both the ADO and the contractor, as well as approximately 13 observers, at one review, while inspections typically have 3-4 reviewers);
- often participants are not happy with meetings as the meetings tend to drift from their main purpose;
- participants may include outside experts in technical areas, quality assurance and ordnance;
- usually the reviews are held at the contractors' premises and the contractors write the agenda which starts with actions outstanding;
- the agenda may include presentations and demonstrations which may be the focus of the review, or the review may be document-driven;

- reviews generally expand or contract to fit the allotted time which is often measured in days rather than hours;
- participants may or may not have an active role in the meeting, that is meeting observers are not clearly distinguished from participants.

2.2 Other Review Processes

Research on other forms of review suggests some guidelines that may be appropriate for joint software reviews. (See [Wheeler et al., 1996] for a collection of papers on software reviews and inspections.) However, even within the field of software inspections, there is not always consensus on what constitutes "best practice".

Most of the software inspection methods follow the same basic procedure (see Figure 1), with about 2 hours allocated for each of the preparation and meeting phases [Wheeler et al., 1996]. They tend to be document-driven reviews.

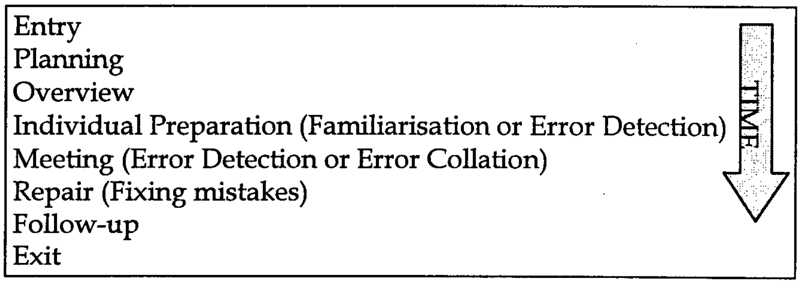
Entry
Planning
Overview
Individual Preparation (Familiarisation or Error Detection)
Meeting (Error Detection or Error Collation)
Repair (Fixing mistakes)
Follow-up
Exit

Figure 1: Phases of the Inspection Process

Guidelines for inspections [Brykczynski, 1994; Gilb, 1996; Grady and Van Slack, 1994; Shirey, 1992] suggest that inspections may fail because of:

1. High start-up costs including cultural change.
2. Poor planning, including introducing inspections on a project that is already in trouble; lack of resources; conducting inspections too late; or rushing inspections.
3. Lack of commitment to (the intent of) the process.
4. Lack of, or poorly defined, inspection goals.
5. Lack of, or differing, standards or quality goals.
6. Inappropriate or untrained reviewers.
7. Lack of entry and exit criteria.
8. Poor product stability.
9. Lack of historical information on defect distribution (insertion and removal by phase and by defect type), the cost of inspections and testing, the cost of rework and the cost of defects remaining in the system.

These criteria and failure to maintain change management information should also be considered for joint software reviews.

2.3 Implications

Software inspections are generally conducted in a very different environment than joint software reviews. They are internal reviews, conducted by peers usually from within the same development team, and often with well-defined checklists or scenarios to aid in defect detection. They are conducted before joint reviews and the outputs of inspections may be used as inputs to joint software reviews.

In contrast, a joint software review is, as its name suggests, a joint process. It involves representatives from at least two organisations, or groups, normally the acquirer and the developer. These two groups may have very different functions and be made up of people with very different backgrounds, experiences, aims and objectives [Gabb et al., 1991; Fisher et al., 1997; Warne and Hart, 1995]. For example, the acquirer's group may include users of the system with little or no background or experience in software engineering.

Another significant difference is that inspections aim to *detect defects* while joint software reviews are aimed at *identifying and resolving issues*. While there may appear to be little difference between these terms, the implications are significant.

In simple terms, the main difference is that a defect is a current, definite problem, while an issue may relate to risks, future states of the system, or subjective opinions on the current state of the system. Issues include defects as well as issues related to risk mitigation (RM issues), and issues related to implicit requirements (IR issues). These concepts are discussed further in [Kingston, 1997].

The following quote from [Gabb et al., 1992] indicates how some of these issues arise:

"In many projects the refinement of requirements and detailed design can lead to a [sic] implementations that the customer regards as unsatisfactory. This is particularly likely in areas such as the definition of the user interface and in the specification of detailed performance (such as response times). More importantly, although the implementation may be unacceptable, it is often either compliant with the higher level requirements or the judgment of compliance is a subjective issue. While it might be claimed that this is the result of poorly specified requirements, this will frequently not be the case. Customers are encouraged to avoid detail in the requirements which might inhibit the design... The penalty for lack of detail is a development resulting in an unacceptable design."

Other issues arise due to differences in the backgrounds, experience and expectations of participants from the two organisations - the ADO and the contracting organisation. For example, *"There appears to be an almost universal difference of opinion between developers and customers regarding the suitability of delivered documentation"* [Gabb et al., 1991].

2.4 Summary

Joint software reviews have been poorly studied while other forms of review - such as software inspections have been widely studied. Many of the lessons learned about software inspections may be applicable to joint software reviews. However, many of the key concerns for joint software reviews (see Table 1) have not been addressed for software inspections. Furthermore, there are significant differences between the two review processes (see Table 2), which may limit the applicability of lessons learned for software inspections.

Table 1: Key concerns regarding joint software reviews

C.1.	Review processes are poorly defined and *ad hoc*. Different procedures are used for different reviews in an *ad hoc* manner.
C.2.	Little historical data is kept for joint software reviews. One difficulty with this is the need to cater for joint software reviews held between the ADO and different contracting organisation.
C.3.	Even if historical data were kept it would be of little value as there are no standard measures of efficiency and effectiveness and no guidelines on what information is important.
C.4.	There are no guidelines on how to identify and resolve issues, or lists of common issues that arise.
C.5.	Reviewers often receive little or no training on how to conduct reviews and identify issues.
C.6.	There are no guidelines on when information should be received before the review.
C.7.	Reviews are not seen to be cost-effective, reviews tend to last for a long time (days) and participants stay for the duration of the review.
C.8.	Many reviews contain presentations, demonstrations and documents. It is not clear how to navigate through the maze of information to identify, raise and resolve issues. That is, there are no standard methods: a) to integrate information in the various media, and b) to ensure that all the information and issues are covered.
C.9.	Software reviews often seem to get off-track and it is not clear how to manage the concerns of all participants.
C.10.	Joint software reviews often have no entry or exit criteria. The outputs of the review are not clear and the end of the review is identified by the end of the allotted time.

Table 2: Differences between reviews and inspections

D.1.	Software inspections are aimed at identifying defects while joint software reviews are aimed at identifying issues.
D.2.	Joint software reviews typically have more participants than inspections.
D.3.	Software inspections are an internal, peer review while joint software reviews are an inter-organisational review where management is often present. Joint software reviews may also be attended by outside experts.
D.4.	Participants at joint software reviews often have very different goals, and expectations. For software inspections goals are deliberately controlled and limited and poorly defined or differing goals are seen as a prime cause of failure.
D.5.	Software inspections tend to be document driven while joint software reviews may include presentations and demonstrations as part of the main review meeting.
D.6.	Joint software reviews allow reviewers to submit comments without attending the meetings, while all reviewers tend to attend inspection meetings.

3. Case Study

This section uses the review of project Llama to further explore the nature of joint software reviews, the related concerns, and possible mechanisms for improving joint software reviews.

The section contains three sub-sections; the first provides a brief description of project Llama. The second sub-section describes the review conducted by ITD before the main architectural review. The third sub-section describes the main architectural review of project Llama, which was held in November 1997. The main ideas contained in this section are summarised in Section 3.4.

3.1 Project Llama

Project Llama is part of JP2030 and is being developed by Australian Defence Industries (ADI) in Perth. The project aims to develop a replacement system for situation monitoring subsystem (MILGEO) of the Joint Command Support Environment (JCSE) Command Support System (CSS) [Hay, 1997].

In simple terms, the situation monitoring subsystem provides the display of information on maps. This includes static information such as roads and rivers as well as dynamic information such as troop deployments.

It was originally intended to deploy JCSE to a small number of headquarter facilities. Based on these requirements, the MILGEO system was designed to make use of available COTS products. Some of these hardware and software COTS components were expensive and received little use apart from their role in the MILGEO system. The architecture used required that additional software licences and hardware be acquired whenever new users or user sites were added. This resulted in a high COTS cost per user. It is now intended to deploy JCSE to many more sites and across more terminals than originally intended. The high cost of COTS components, and the support required for the current UNIX-based systems make this prohibitively expensive [Hay, 1997].

The replacement subsystem, project Llama, will provide a platform-independent situation monitor that will provide similar functionality to MILGEO with lower deployment costs. This will be achieved by minimising the use of expensive COTS software and hardware and by developing the system in Java [Hay, 1997].

3.2 ITD Review

Information Technology Division were asked to provide input to the joint architecture review for project Llama and formed a multi-disciplinary team with experts in Software Engineering, Human Factors and Geographical Information Systems to assess the design. The ITD review was neither a joint software review, nor an inspection. The review was not a joint software review as there were no participants from either the project office, or from ADI. Furthermore, the review was not an inspection because the team did not consist of the author's peers and there was no inspection checklist. Instead, the team attempted to elicit their goals for the review and to assign roles to participants to ensure that the entire document set (consisting of a software development plan, and a software design document) was covered and all the goals were addressed. The roles were to be determined from the structure of the documents and the areas of expertise of the participants.

The actual process used consisted of the participants familiarising themselves with the design and automatically focussing in on their areas of interest and expertise. Formal assignment of participants to roles was never achieved, and the goals were only determined at a very high level. A number of issues were raised from this review and the review team did not continue with the process as originally planned. They did not ensure that all their goals were met and that coverage of the documents was complete.

The reviewers stated that they believe the level of detail in the design was sufficient if the aim was to develop a concept demonstrator or prototype. They considered that the design would need to be reworked and re-reviewed if it the project intended to develop a production system, as the design was not sufficiently defined or robust under these circumstance. A number of issues relating to these areas were discussed.

The remainder of this section provides more information about how the review was conducted, the review goals and evaluation criteria used, and the limitations of the approach used.

3.2.1 Review Process

The review process used was developed and modified on the fly. The documentation for the review was received approximately one week before the review. A meeting to clarify ITD's role in the review and to plan the review process was held on the day that the documentation was received. The process consisted of a series of meetings interspersed with individual preparation sessions. The planned and actual activities conducted at each meeting are given in Table 3. The second meeting was conducted the day after the first meeting and the final meeting was conducted approximately 3 working days after the first meeting.

Table 3: Meeting Activities

Meeting	Planned Activity	Actual Activity
1	• Identify the goals for the review. • Assign people to areas of responsibility.	• The goals for the review were identified. • Assignment of people to areas was deferred until people were familiar with the document.
2	• Assign people to areas of responsibility.	• People indicated issues that they had uncovered. These tended to fall into their areas of expertise. • People were allowed to continue reviewing the document according to their interests.
3	• Check consolidated issues list	• A power failure meant that the meeting the consolidated list could not be printed or distributed. The meeting was postponed until power was restored, but due to time constraints the list of issues could not be distributed, and checked, before the meeting. • Consolidated issues list was checked for inconsistencies, errors of fact, and unnecessary replication.

It was planned to allocate responsibilities to participants at the first meeting. Mechanisms for doing this were discussed (see Section 3.2.2) but responsibilities were not allocated, as many participants wanted time to familiarise themselves with the documents. During the familiarisation phase, many participants also identified issues rather than broad areas of interest in the material. The second meeting tended to focus at this detailed level, rather than at the high-level of assigning responsibilities. A formal process of allocating people to areas of responsibilities was not used because people had tended to focus in on different areas and were already concentrating on identifying issues.

Review comments were circulated via email between the second and the third meeting, and a WWW site developed to enable ready access to the comments between the meetings.

3.2.2 Review Goals and Evaluation Criteria

Dr Lakshmi Narasimhan suggested that the Enterprise model may provide a suitable scheme for classifying issues raised by the ITD review team [Linington, 1994 #1197; Raymond, 1995 #1195]. A selection of more common evaluation criteria (eg [MIL-STD-1521B, 1985; MIL-STD-498, 1994; NASA, 1993; Sherif, 1992]) was also tabled (see Appendix C). Some attempts were made to combine the two sets of criteria, but it was acknowledged that in many ways these schemes provided two different dimensions for the evaluation.

The final scheme used by the ITD review team was based on the Enterprise model and is shown in Figure 2. This scheme was useful, but it was often unclear where to place issues within the classification scheme. Several issues were found in multiple areas of the consolidated issue list during the third meeting. Furthermore, the relationship between the various expert domains and the more common evaluation criteria is not clear. A three-dimensional framework, which addresses these problems, is proposed in Section 4.1.

1 Enterprise
- 1.1 Goals
- 1.2 Supporting Evidence
- 1.3 General Quality
- 1.4 General
- 1.5 Evolution/Migration
- 1.6 Policies
- 1.7 Risk
- 1.8 Feasibility
- 1.9 Maintenance
- 1.10 Evaluation

2 Technology
- 2.1 Engineering Choices

Maturity

3 Engineering
- 3.1 Processes
- 3.2 Software Development
- 3.3 Standards
- 3.4 Quality Assurance

4 Computational
- 4.1 Design
- 4.2 Performance
- 4.3 Quality of Service

5 Informational
- 5.1 User Requirements
- 5.2 Functional Mappings

Figure 2: Evaluation Criteria

3.2.3 Benefits and Limitations

The process used by the ITD review team has a number of strengths. There was an attempt to clearly identify the goals and roles for the review, and an evaluation framework was proposed and used. However, it is also clear that the process used has some limitations (see Table 5). Furthermore, because of the limitation L.1, the review team deviated from the proposed review process (see Table 4).

Table 4: Limitations of the Proposed Process

L.1.	People wanted to focus quickly on their areas of interest, and on identifying issues, rather than identifying those areas of the documents they felt capable of reviewing.
L.2.	The evaluation criteria covered only a single dimension. (See Section 4 for a discussion on other evaluation dimensions.)

A brief meeting was held after the joint architecture review (see Section3.3). Participants at the meeting discussed the impact of the ITD review, and other limitations of the ITD team's process became clear. These limitations are shown in Table 6 and concern incompatibilities between how the ITD review and the joint architecture reviews were conducted.

Table 5: Limitations of the Process Used

L.3.	There was no attempt to ensure coverage.
L.4.	The document may have issues associated with areas that were not addressed during its initial review. A subsequent review, which either deliberately or coincidentally addressed these areas, could uncover these issues. Thus even if the document addressed all of the issues raised in this review, it would not necessarily pass a subsequent review.
L.5.	Detailed goals were never identified.

Table 6: Limitations of the ITD Process in Conjunction with the Joint Architecture Review

L.6.	Process Compatibility: Information from the ITD review was recorded according to the evaluation criteria that ITD developed. Discussions at the joint architecture review were based around key ideas and key sections of the documentation. This meant that information was difficult to find at the relevant time.
L.7.	Goals and Assumptions: Many of the comments made concerned the development of the system as a concept demonstrator versus a system which would be field instead of, or as a substitute for, MILGEO. This was outside the scope of the review, as it had already determined that the system would replace MILGEO in the field.
L.8.	Reviewer Assurance: Only one ITD reviewer attended the joint architecture review. While that reviewer was confident that the issues identified by the ITD review team were being addressed and were under control, he was not able to convince the other ITD reviewers that the issues were being addressed.

3.3 Joint architecture review

A survey of participants at the architecture review meeting was also conducted. The survey questionnaire can be found in Appendix A and the detailed results from this survey are shown in Appendix B.

The review consisted of several presentations, the demonstration of a prototype, questions throughout and the discussion and resolution of issues.

Fifteen people attended the review - 4 from the JP2030 Project Office, 1 outside expert from DSTO and 10 people from ADI. Six responses were received from across all these areas: two from the project office, one from DSTO (this person was the only member of the team discussed in Section 3.2 to attend the review) and three from ADI. The ADI respondents included two presenters and one person from quality assurance.

Some interesting observations can be drawn from these responses.

1. The review was well received. Participants had very high-level goals and generally felt that they achieved their goals.

2. The perceived benefits of the different activities in the review depend on the role/organisation of the participants.

Some of these results are intuitively obvious. For example, the developers gained little from viewing the presentations that they helped prepare and more from the questions that were asked.

Other results are more interesting. In particular, *the Project Office perceived little benefit from reviewing the documents*. The Project Office also perceived greater benefits from general questions by other people than perceived by the outside expert.

3. The balance of participants was almost right, but the review could have benefited from more users and less developers.

4. Most participants' roles stemmed from their position - for example project director or project manager. The review may have benefited from participants taking roles - eg system maintainer or user - particularly as, according to the data received, no users participated in the review.

5. The outside expert believed they would have been better prepared for the review if they had received more information about the nature of the project and its role in JP 2030 before attending the review.

These observations reflect the results of a small survey of a single case study, with a response rate of 30% and are therefore inconclusive. Surveys of other software reviews are planned and may shed more light on the nature of joint software reviews.

3.4 Summary and Recommendations

The study of the joint architecture review for project Llama provided a number of insights into the conduct of, and suggested a number of possible improvements to, joint software reviews. These recommendations are summarised in Table 7.

This case study was unusual in that a preliminary review was held by the ITD review team prior to the main review. However, this offered the opportunity to investigate how issues were identified. It highlighted the importance of having an evaluation framework - and of using that framework not just to identify and collate issues, but of using a compatible framework during the formal review. It showed the importance of knowing the scope of the review and having well-defined goals. It also showed that if the review is to provide exhaustive coverage of issues within a defined scope then someone must be responsible for defining areas of responsibility and ensuring that coverage is obtained. The team-based approach to ensuring coverage proposed by the ITD review team was not successfully implemented.

Table 7: Recommendations from Project Llama

P.1.	Increase participation by users and reduce participation by developers. Rather than simply restricting the number of developers who attend, this may be better implemented by having developers attend only relevant parts of the review.
P.2.	Assign roles to the participants - particularly in the absence of stake-holders.
P.3.	Ensure appropriate information is distributed before the review. Include background information for people outside the project office and development team. Limited distribution of documents to the project office may be possible, as not all members of the project office read the documents before the review.
P.4.	Ensure goals for the review are appropriate and sufficiently detailed to scope the review.
P.5.	Increase the flexibility of attendance at the review. The role of some of the participants means that they only contribute during parts of the review. By allowing participants to attend only those portions of the review of most benefit to them, some participants may be able to leave early, some may be able to arrive later and others may be able to conduct more beneficial work between the review sessions of most relevance to them.
P.6.	Provide a consistent evaluation framework and use this not only to review the system, but also to drive the discussion of the issues.
P.7.	Ensure coverage of all evaluation criteria within the scope of the review.
P.8.	Provide mechanisms for feedback to reviewers who are not present at the review.

Before the recommendations in Table 7 are implemented, it should be recognised that they are based on the results of a single case study, including a survey with a response rate of 30%. These recommendations may not be appropriate for all projects. Some of these recommendations also have support from other studies. Section 5 provides a complete list of recommendations based on information in this document and indicates sources of support for the recommendations.

Care needs to be taken when implementing some of these recommendations. In particular, P.1. recommends increased participation by users and this can have both positive and negative implications. Care needs to be taken to select users whose views are representative of the majority of users, and the selected users may need special training to understand the information presented at the review, and contained in the review documents. To avoid this second potential problem, during the review, the users' input could be restricted to the proposed user interfaces and the evaluation of prototypes during demonstrations. Alternatively or in addition, surveys of users' opinions could provide input to the review for consideration by all participants.

4. An Improved Review and Evaluation Process

The strengths and weakness of current review methods were summarised in Sections 2.4 and 3.4. Current review methods can identify numerous issues, however one problem with current review methods is that important issues can easily be missed. This can occur due to reviewers focusing on less important areas and types of issues.

An alternative method with three main components is proposed in this section. The first component of the review method is an extensible three-dimensional evaluation framework (Section 4.1). The framework identifies areas that may be of interest in the review. Over time, the framework should be populated with questions relevant to each area. To address each area in the framework would require a significant amount of effort and may not be worthwhile in many circumstances. The second component of the review method is a method of using and customising the framework so that the most important areas of the framework are considered and responsibility for them can be assigned to different individuals (Section 4.2). In this context, the framework is particularly useful in identifying issues before the review meeting. This forms part of the third component of the review method provides guidelines for other aspects of the review, including planning and conduct of the meeting, and how the framework can be used in these contexts (Section 4.2).

Examples from project Llama are used throughout this section and the section's main contribution is summarised in Section 4.3.

4.1 A Three Dimensional Evaluation Framework

Figure 3 shows a representation of the three-dimensional evaluation framework proposed in this section. The three dimensions of this framework are: Knowledge Domains (Section 4.1.1), Criteria (Section 4.1.3), and Viewpoints (Section 4.1.2).

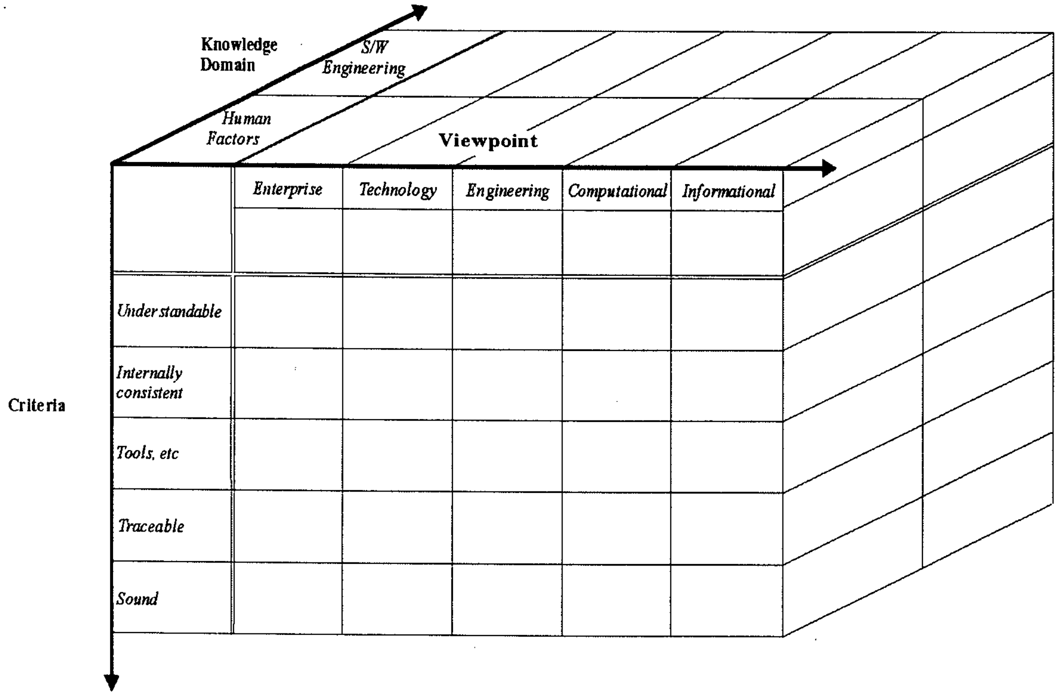

Figure 3: A Three-Dimensional Evaluation Framework

Several previous evaluation frameworks have used one or more of these dimensions. Many focus on the Criteria dimension. For example, standards that address evaluation needs focus on criteria such as useability, maintainability, portability and reliability [ISO/IEC 9126-1, 1996; MIL-STD-498, 1994]. The approach used by the ITD review team may be considered as a one- or a two-dimensional approach. Formally, the ITD framework consisted of only one dimension; Viewpoints based on the Enterprise model. However, there was some informal support for the Knowledge Domain, through the use of experts from different domains.

The previous one- and two-dimensional approaches help ensure coverage along the included dimensions, but do not ensure coverage in the remaining dimensions. The results from the ITD review of project Llama show that using the three dimensions independently is not sufficient to ensure coverage (Section 4.1.5). The three-dimensional approach addresses these limitations.

The framework provided in this document is only a beginning. To be most useful, it needs to be extended, populated, regularly updated and shared between projects (Section 4.1.4).

4.1.1 Viewpoints

Viewpoints offer a means of reviewing a document from different perspectives. The viewpoints proposed in the 3-dimensional evaluation framework come from the Reference Model of Open Distributed Processing (RM-ODP), also called the Enterprise Model [ISO/IEC 10746-1, 1995].

Methods similar to the viewpoint dimension have been used in other forms of review, such as inspections. One method is to assign participants to roles [Bisant, 1989], such as the user or maintainer role. A second approach is to use scenarios [Porter et al., 1995], or function-point scenarios [Cheng and Jeffery, 1996]. In the scenario approaches, each reviewer is provided with a different scenario. The scenario contains a set of questions, and a perspective from which the software should be reviewed. The original scenario approach is poorly defined, cannot readily be used for other projects, and there may be overlap between the scenarios and issues, which are not captured by the scenarios. The function-point scenarios approach was developed to address these concerns for Management Information Systems (MIS). This approach may not extend to software-intensive military systems, and even if it can be extended, was not found to be as effective as other methods of decomposing the system.

In contrast, RM-ODP was found by the ITD Review team to provide useful viewpoints (Section 3.2). The RM-ODP was developed to fulfil the need for "a co-ordinating framework for the standardisation of Open Distributed Processing (ODP)" [ISO/IEC 10746-2, 1995]. Significant effort has been invested in the development and continual updating of this framework. The five viewpoints are believed to be both necessary and sufficient for use in the development of ODP standards [ISO/IEC 10746-2, 1995]. Although the RM-ODP was designed specifically for ODP systems, the amount of thought that went into defining the viewpoints means that, conceptually, they are largely applicable for other systems.

The remainder of this section looks at the five viewpoints proposed in RM-ODP and used by the ITD review team in their evaluation of the project Llama architecture. These are the Enterprise, Computational, Informational, Engineering and Technology viewpoints. The RM-ODP descriptions is given first, and then an interpretation on how to extend the scope of the viewpoint for use in joint software reviews is given. These extensions combine the original descriptions with software and systems engineering knowledge and experience gained in using the model for the project Llama review. The RM-ODP descriptions combine information from [Linington, 1994] and [Raymond, 1995].

4.1.1.1 The Enterprise Viewpoint

RM-ODP

The enterprise viewpoint provides a high-level view of a system, its environment, and its requirements. According to Linington it "focuses on the purpose, scope and policies for the system".

Joint Software Reviews

The enterprise viewpoint for joint software reviews also provides a high-level view of a system, its environment and its requirements, and focuses on the purpose, scope and policies for the system.

For software and system acquisitions, there are several policy areas that may need to be addressed. Some of these are described below with examples of issues that were discussed during the ITD review for Project Llama.

- The nature of the system. Is the system to be a concept demonstrator, fielded and used in possible life- or mission-critical situations?

  Project Llama: The ITD review team believed that the design documentation provided for Project Llama was suitable for a concept demonstrator to explore alternative implementations to MILGEO. They did not believe that it would provide a suitable replacement for MILGEO.

- The purpose and scope of the system. Who are the intended users of the system? What will the system be used for? What assumptions have been made? Where is the boundary between when the system should and should not be used?

  Project Llama: Project Llama is much simpler than MILGEO. From the information received by the ITD review team it appeared that Project Llama was intended as a replacement for MILGEO. While many of the users of MILGEO find it unnecessarily complex, there are users who require more functionality than Project Llama will provide. Therefore, Project Llama should not be considered as a replacement for MILGEO.

- The evolution of the system. How will the system evolve? Is evolutionary acquisition being used? How will the system be transferred to the field? What training will be required - now and in the future? For how long will the system be fielded? How will the system be maintained? How will the system be retired or replaced?

  Project Llama: Project Llama is alternative system to MILGEO, which is already fielded in several locations. The ITD review team believed that the architecture documentation should have discussed how Project Llama would be fielded. For example, is it possible to operate both Project Llama and MILGEO simultaneously? Can both operate correctly if connected to the same network and used at the same time?

- The risks with the acquisition. What are the main high-level risks to the acquisition? How are they being managed?

  Project Llama: The architecture documentation identified several risks with the proposed development approach for Project Llama. However, in some circumstances, the documentation failed to discuss the likelihood of these risks eventuating, and the documentation did not discuss mechanisms for handling the risks if they eventuated.

A review from the enterprise viewpoint should ensure that policy areas, which are important for the review, are addressed, and addressed in an appropriate and satisfactory manner. The remaining viewpoints focus on lower level concerns about the system.

4.1.1.2 The Informational Viewpoint

RM-ODP
The information viewpoint focuses on information required by a distributed application. It is included in the model to ensure that applications "share a common understanding of the information they communicate when they interact" [Linington, 1994]. It looks at the "scope and nature of information specifications" [ISO/IEC 10746-2, 1995].

Joint Software Reviews
The informational viewpoint for joint software reviews is also concerned with ensuring a common understanding, but about the operation of the system rather than the information that the user needs from the system. This viewpoint looks at the system from the users' point of view and considers if the users' needs will be met. It is closely related to the scope of the system, which is addressed under the Enterprise viewpoint, but allows concerns about the users' requirements and the functionality of the system to be addressed at a more detailed level.

Project Llama
For example, the documentation of Project Llama provided a simple method for selecting images where bitmaps of all available images were displayed in a single scrollable window. This approach would quickly become unwieldy once a large number of images were available. A structured approach to selecting images would be preferable and make the system easier to use.

4.1.1.3 The Computational Viewpoint

RM-ODP
The computational viewpoint looks at how entities within a distributed system interact. It provides a functional decomposition of the system into objects that can be distributed throughout the system. This viewpoint covers a wide range of information about a distributed application, including information on:

1. the portability of objects,
2. failure control mechanisms and potential points of failure,
3. when and why objects interact, as well as information about their internal actions.

Joint Software Reviews
For joint software reviews, we extend the scope of the computational viewpoint slightly to look at all aspects of computation - design, performance and service quality. The design of the system contains point 3 in the RM-ODP description (when and why objects interact), but may also look at how interaction is achieved. (See also notes under the engineering viewpoint.) Performance includes consideration of failure control (point 2), timing constraints etc. Service quality includes consideration of how the design affects the quality of service the system provides - now and in the future including consideration of portability, safety, security etc.

Project Llama
Examples of computational issues that arose when the ITD team reviewed the Project Llama documentation included:

- lack of information about how fault tolerance would be achieved;
- a potential problem in meeting performance constraints without an operational profile, that is it is difficult, if not impossible, to optimise the performance of the system if you don't know how it is used - that is which operations, and sequences of operations are the most commonly used;
- an alternative design would be to implement map tools and vector maps as applets.

4.1.1.4 The Engineering Viewpoint

RM-ODP
The engineering viewpoint focuses on mechanisms for achieving distribution and allowing distributed entities to interact.

Joint Software Reviews
It is in the engineering viewpoint that we see the major differences between the viewpoints for RM-ODP and proposed viewpoints for joint software reviews. In general, the design of software systems includes consideration of how entities will interact (if necessary). The Computational viewpoint concerns the design of the system, including how the entities interact, and other concerns which at first sight appear similar to in the RM-ODP engineering viewpoint.

In determining what should fall under the joint software reviews engineering viewpoint, we need to consider the foci of the viewpoints. The focus of the RM-ODP is distributed systems. The focus of joint software reviews is the development (and acquisition) of software-intensive systems. That is joint software reviews do not just assess the product, they are also concerned with the whole development system including both the product, and the processes used to develop it. Thus while the engineering viewpoint of the RM-ODP focuses on the processes used to achieve distribution, the engineering viewpoint for joint software reviews should focus on the processes used to develop and acquire software-intensive systems.

The nature of this viewpoint will vary depending on the object being reviewed. If the review is focused on a software artefact or product, then this viewpoint should address:

- procedures for delivery including schedules, and costing approaches;
- the use and quality of the standards to which the product is being developed;
- whether common systems and software engineering processes are being used - for example quality assurance measures, version control, development methods, review processes;
- and whether or not these processes are suitable and compatible.

Project Llama: There were many development processes that were not fully specified for project Llama. For example, it was not clear whether JavaBeans or component-based software engineering (CBSE) technologies were being used, the choice of web development tools was not clear, and version-control tools were not specified.

If the review is focused on the processes themselves, then this viewpoint should address:

- the use and quality of standards for the development processes;
- and the processes used to modify, adapt and improve the development processes.

Project Llama: The process documentation identified the need to update standards to reflect the use of Java. However, the documentation did not describe how much effort would be required to convert the standards. (Note that from later discussions it appears that conversion of the standards was already well under way when the process documentation was delivered.)

4.1.1.5 The Technology Viewpoint

RM-ODP

The technology viewpoint focuses on the implementation of the system and the design choices that were made for the implementation. According to Linington, it "focuses on the choice of technology in that system".

Joint Software Reviews

The technology viewpoint for joint software reviews also focuses on the engineering or technology choices, for a system, and on the quality of those choices. It includes evaluating the choices of technology for maturity, suitability, durability and compatibility with other systems. It includes considering the choice of development language, the choice of COTS systems, and the selection of components for reuse.

4.1.2 Knowledge domains

Software-intensive military systems are becoming increasingly complex. Early systems tended to automate isolated functions while modern systems, such as command and control systems and weapons control systems, perform a much broader role and also need to be interoperable with a variety of other software-intensive systems [Deephouse et al., 1996; Gould et al., 1994].

Development of complex software-intensive systems is known to require expertise from a variety of knowledge domains: eg Geographical Information Systems (GIS), Human System Interaction (HSI), software engineering and systems engineering (SSE) and the application domains [Butterfield et al., 1994; Rutherford, 1995; Lim, 1996; Noseck, 1994] - eg Radar, Navigation, Command and Control, Submarine Warfare and Intelligence.

The ITD review team consisted of experts from several knowledge domains including: Geographical Information Systems, Human Factors and Software Engineering. Specialities included Genamap, Java, system performance, MILGEO and Intranet expertise. Each expert tended to focus on his or her areas of expertise and interest. Some issues were raised by experts from all knowledge domains, but many issues were only raised by experts from a single knowledge domain. ITD had identified obvious knowledge domains that they thought they could contribute to the review, but made no attempt to ensure that all relevant knowledge domains were covered. (Nor was this possible given the time constraints and availability of information.) Other approaches to review have also used experts from multiple knowledge domains eg [NASA, 1993]. A systematic approach to identifying relevant knowledge domains and specialities, and appropriate experts to address the most important knowledge domains, would identify issues which may not otherwise be identified until much later in the development.

4.1.3 Criteria

Most evaluation models focus solely on the third dimension, criteria. This dimension covers issues such as maintainability, portability, reliability, correctness, scalability, understandability, soundness and completeness. These criteria are very important and need to be addressed at many levels.

Consider the understandability of a design architecture. It must address whether decisions were clearly identified, justified, and summarised; and whether alternatives were considered and complete.

The ITD review team identified many understandability issues in the Llama Project architecture. (See Appendix C for the full range of issues identified.) Issues were identified by experts from each of the knowledge domains and across all viewpoints.

4.1.4 Enhancing the model

The model can be represented using the two dimensional framework shown in Table 8 to capture the viewpoints and criteria dimensions. The third dimension, knowledge domain, can be represented using different fonts (or colours) for the different knowledge domains (as in Appendix C) or it can be represented using one copy of Table 8 for each area of expertise.

As given in Table 8, the framework provides a mechanism for determining review criteria, but provides very little guidance. To make the framework easier to use it needs to be enhanced. Common knowledge domains need to be identified and the framework needs to be populated with useful questions for each knowledge domain. As new knowledge domains and additional questions are identified, they need to be added to the framework.

Table 8: A Software Evaluation Template

	Enterprise	***Technology***	***Engineering***	***Computational***	***Informational***
	Strategy Evolution Integration Summary Cost/Schedule	Eng. Choices Maturity	Process Software Dev. Standards QA	Design Performance Service Quality	User Req. Functionality
Understandable					
Internally consistent					
Tools etc					
Traceable					
Sound					
Correctness					
Reliability					
Maintainability					
Flexibility					
Reuseability					
Interoperability					
Security					
Safety					
Scalability					
Terminology etc.					

Some work has already commenced on identifying standard knowledge domains for some types of systems - simulators (by Industry Involvement and Contracting Division, II&C, DAO), C2 systems from a high level perspective and C3 systems from a technological perspective (Information Technology Division, ITD, DSTO). The results of these studies could be used to enhance the framework, which should be considered as a living model that will evolve over time.

4.1.5 Project Llama and the Three Dimensional Evaluation Framework

It is not the purpose of this document to describe the details of the issues identified, or to provide a checklist for the evaluation of similar systems in the future. However, it is useful to consider the common criteria used by experts from a variety of knowledge domains to identify issues, with a view to providing an initial framework that may assist in the development of review goals and evaluation criteria for other projects.

Appendix C provides a breakdown of the issues that were identified by the ITD review team. The breakdown in the viewpoints dimension is based on the results of the review which used the Enterprise model to collate results. The Enterprise model bears a close relationship with the viewpoints dimension but the distinction between the viewpoints was not clearly defined before the ITD review. Consequently, some issues occurred in more than one location. Some of the issues have been moved from their original locations in the framework. However, most issues may still be found in the original viewpoints. The remaining dimensions were not used by the ITD review team and the positioning of elements within these dimensions was done by the author based on her experience. Some of the issues remain unclassified in the knowledge domains dimension.

Several questions that may be used to identify issues are shown in the criteria dimension. This shows questions that may be relevant for a criterion across all values of the remaining dimensions. For example:

- the Understandability criterion addresses the questions:
    - are all decisions identified?
    - are they justified?
    - are they summarised?
    - are alternatives considered?
    - are there other alternatives that should be considered? and
- the Tools criterion addresses the questions:
    - are tools and other products, standards and processes specified?
    - is the list complete or are tools etc missing?
    - are the tools currently available?
    - are the tools mature?
    - are they appropriate?

Questions can be associated with the individual viewpoints and for particular knowledge domains. Questions can also be associated with particular cells within the framework. For example, the cell addressing the tools criterion for the Engineering Viewpoint of the Software Engineering Knowledge domain may contain the following questions:

- is an automated version control system being used?
- is this system compatible with the development method being used?
- what support is available for the development method - is there an integrated software engineering environment or are individual tools being used?
- are the coding standards appropriate for the development method?
- if not, what effort is required to develop new coding standards?
- what are the procedures for selecting COTS products and reusable components?

These examples from project Llama show that questions can be attached at multiple places in the framework and can be used to provide guidelines on what to consider when evaluating a software-intensive system.

4.2 The Review Process and the Evaluation Framework

This section proposes a method for using the evaluation framework in the context of a review. This process has two main phases as shown in Figure 4, a planning phase where the reviewers and their roles are determined, and a conduct phase where the individual and group review activities are performed. The evaluation framework can be used in both phases of the review.

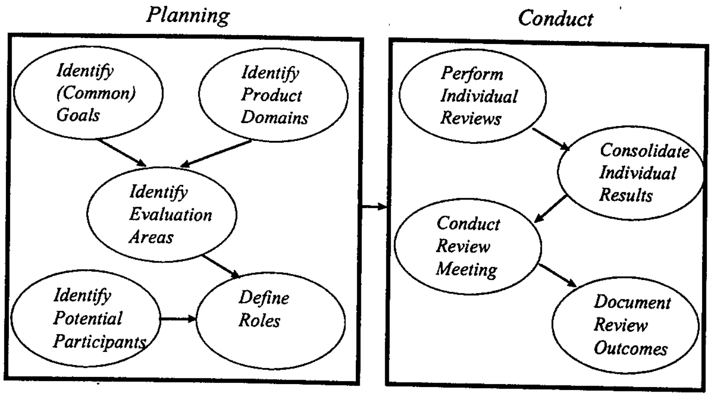

Figure 4: A Review Process

4.2.1 Planning

The three-dimensional framework presented in this document provides a basis for evaluating software-intensive systems in a systematic manner. However, the framework is large (and with enhancements, it will become larger) and it would take a significant amount of time to address each area of the framework. For most projects, it would not be necessary, or even desirable, to address each area of the framework. It is certainly not necessary for each reviewer to address each area of the framework.

The planning phase of the review process aims to assign areas of the framework to individual reviewers. This reduces the workload on individual reviews and provides them with more time to address the areas they have been allocated. By careful assignment of reviewers to different areas, the important areas of the framework can be addressed. Perhaps more importantly, by determining which areas of the framework will be addressed, the review participants, the project office and the contractors will know which areas have not been studied and will not assume that there are no issues in those areas.

Three factors need to be considered when determining the areas of the framework to address - the goals of the review, the nature of the product to be reviewed, and the expertise of the potential review participants. These need to be considered in a case-by-case basis when determining which evaluation areas will be addressed. Some examples are given below.

Goals
Identifying the goals for the review is necessary to focus the review on the areas that are considered to be important. Research in process improvement has shown the importance of identifying goals [Basili et al., 1994]. Some of the review goals will relate to the operation concept and the features that are most important to ensure it is

met. Some of the review goals will depend on the history of the project eg how the contract was written, the relationship between Defence and the contractor, and the results of previous reviews. Other goals may seek to address the high-risk areas.

Product

A nature of the product may affect how rigorous and detailed a review is conducted. A concept demonstrator may require a less detailed evaluation than a production system. Depending on the purpose of the concept demonstrator, it may be possible to focus on areas such as the user interface and ignore the performance and fault tolerance of the system.

The purpose of the product - and the knowledge domains it draws from - may help identify relevant knowledge domains from the evaluation model. There is little point in having a product evaluated against areas that it is not intended to address.

Project Llama

The ITD team reviewed two products for Project Llama - the software development plan [Hay, 1997] and a design document. The nature of these projects restricted the scope of the ITD review. For example, the dynamics of the user interface were not presented in the documentation and therefore could not be reviewed by the ITD team. A prototype demonstration was included in the joint architecture review and if the ITD team had attended the review, some of the dynamics of the user interface could have been evaluated during the demonstration.

Potential Reviewers

Having identified the desired areas of the evaluation framework, it is necessary to assign the areas to the potential reviewers. Where the potential reviewers do not fit the proposed evaluation areas a decision must be made: the help of outside experts may be sought; the review area may be flagged, but not addressed; or the review area may be addressed by someone with only limited expertise in the area. The choice will depend on the importance of the area, the available resources, and the characteristics of the potential reviewers.

Assigning Roles

Once these three factors have been addressed the review participants can be determined and each participant can be assigned roles and responsibilities - that is, given particular areas of the evaluation framework to address. If an area is very important, then more than one reviewer may be responsible for it. Less important areas will be covered by fewer reviewers.

4.2.2 Conduct

There are four phases in the conduct of reviews: individual review, consolidation of reports, the review meeting and documentation of the review.

Individual Review

Reviewers should be provided with information about the purpose and scope of the review and the system under review, as well as the information that is to be reviewed. They should address each area in the framework for which they are responsible. They should not rely on the questions within the framework to provide complete coverage of an area, until the framework has been enhanced. Instead, they should supplement the guidelines in the framework based on their own expertise. During this time, the reviewers should be able to evolve the framework by adding additional questions. It is anticipated that, with use, the framework will stabilise.

However, changes in technology and acquisition strategies will require the continued, controlled, evolution of the framework.

Consolidation

After the individual review, it is recommended that the results be consolidated so that Defence presents a unified outlook to the contractors. This consolidation has the added benefit that it might be possible to discard some issues, if someone has additional knowledge about the scope and nature of the system. The consolidation should also identify the most important issues. This includes issues that arose from consideration of different areas of the framework. It also provides a record of all the issues that were identified prior to the review.

Review Meeting

It is recommended that the review meeting be structured around the key evaluations, which were identified in the planning stages of the review. The documentation could be supplemented with relevant presentations and demonstrations for each area, and the reviewers given the freedom to attend only those sessions on the areas they reviewed. This has the potential to reduce overheads to the individuals involved in reviews.

Review Documentation

It is recommended that the results of the review meeting be documented. Documentation should cover the issues raised, how they were resolved and clearly indicate any outstanding issues. This should cover issues raised both before and after the review.

The review documentation should be signed off by all the reviewers, who should check that all the issues they raised are included. The document should also indicate issues, which appeared in the consolidated issues list, but which were not raised in the meeting. If an issue was not raised, the reason for this should also be clearly identified. Two common reasons issues are not raised are that they were addressed by the presentations or demonstrations (how they were addressed should be specified) or that there was insufficient time to address the issues. In the later case, these issues should remain outstanding.

It is also desirable to keep historical information about reviews. As a minimum, this information should cover the nature of the review, the areas addressed, the effort of the reviewers in individually reviewing the documentation and attending the review, and the total duration of the review. A historical record of the review documentation is also desirable. Recommendations about how the review could have been improved could also be collected.

4.3 Summary

A three-dimensional framework was proposed along with a process for using the framework within reviews. The framework addresses many of the criticisms of other approaches to joint software reviews. However, it is still to be validated in practice. The three dimensions of the framework are viewpoints, knowledge domains and criteria. Viewpoints are different perspectives of the system, knowledge domains are expert and application domains applicable to a system and criteria are the more common evaluation criteria.

The identification of goals and customisation of the framework are recommended before it is used as both the basis of individual review and to drive the review meeting.

5. Recommendations

This section recommends procedures for use in joint software reviews.

Table 10 lists the recommendations. They appear in bold, are numbered and described, and material which supports the recommendation is identified. The third column indicates whether material from the (I)nspection literature, (J)oint software reviews literature, or the project Llama case study - either the study of ITD (R)eview Team or the (S)urvey at the joint architecture review. This key is summarised in Table 9. Each of the recommendations addresses one or more of the concerns C1..C10 or limitations L1..L8 where were identified in this document.

Table 9: Key to supporting information

Key	Meaning
I	Inspection literature
J	Joint software reviews literature
R	Project Llama - ITD Review Team
S	Project Llama - Architecture Survey

The first 8 recommendations may be addressed at least partially by the use of the review process presented in this paper. The ninth recommendation concerns the use of the three dimensional framework presented in this paper. The remaining recommendations were identified in this paper, but may require additional research before they are implemented. In particular, additional research may be required to implement recommendations R11 and R12.

Table 10: Recommendations and Supporting Evidence

Description		*Support*	*Location*
R1.	**Use a defined process.**	I, J C1	§ 2 (ie Section 2)
R2.	**Use compatible processes for individual evaluation of the product and for conducting the joint review.**	R, A L6	§ 3
R3.	**Provide guidelines on how to identify and resolve issues and develop lists of common issues.**	I C4	§ 2
R4.	**Use a process that identifies and manages the key concerns of participants and of the project.** The goals of the review need to be identified and the participants concerns in relation to these identified. This may result is the varying the participants during the review.	I, A C9, L1, L3, L5, L7	§ 2
R5.	**Ensure that suitable participants are present at appropriate times in the review and receive appropriate information before the review.** This recommendation has three parts: the selection of participants, their attendance at reviews, and the information they receive.	J, R, A C7, L7	§ 3
R6.	**Use a multi-dimensional review framework.**	R L2	§ 4

Description		Support	Location
R7.	**Use a process with well-defined entry and exit criteria.** Entry and exit criteria should include validation of the review minutes by all participants - including those who sent information but did not attend.	I, R, A L4	§ 2
R8.	**Use the review process and three-dimensional framework described in this paper.** This process has many desirable properties, which are given in recommendations 1-7. If this process is not used then an alternative method that also fulfils recommendations 1-7 should be used.	R	§ 4
R9.	**Customise and expand the three dimensional framework.** This extends the existing guidelines and is necessary for the full benefits of using the framework can be achieved.	R	§ 4
R10.	**Train people in how to conduct reviews and identify issues.**	I C5	§ 2
R11.	**Maintain historical data records.** If the three-dimensional framework is used, it can be kept in a common repository as a record of issues identified for each project. Information about the number of participants, their preparation time, and duration of the review should also be stored. This information can be used in determining effectiveness measures and refining the review process and the timing of events.	I C2, C3, C6	§ 2
R12.	**Develop procedures for integrating and navigating through various information sources.** The three-dimensional approach presented in this paper is independent of the sources of information sources for the review. However, it does not provide an explicit means for combining information from multiple sources. That is, the approach does not define a mechanism combining information from the review documents received before the review, with information in the presentations and demonstrations at the review. This is particularly important where similar topics are covered by many information sources.	I C8	§ 2

6. Conclusions

This paper provided a case study of the architectural review for project Llama, including the ITD team's preparation for the review. The case study was placed in the context of the software engineering literature. Together the case study and the literature were used to identify strengths and weaknesses in how software reviews are conducted. These are summarised in Tables 1 to 6.

Based on the weaknesses identified in the architectural review of project Llama, a new review process was proposed. This process features an extensible 3-dimensional evaluation framework which can be customised based on the goals of the review and the expertise of the reviewers. The framework can be used to help manage the review process and ensure coverage of key areas within the review. The framework combines three dimensions that have previously been used independently for software reviews. The first dimension covers the system from different viewpoints. This is second dimension covers the knowledge domains. These are domains of expertise relevant to the system of interest. The final dimension covers the more common evaluation criteria, such as completeness, correctness and traceability. A guide to using the framework in a review is provided and its benefits are discussed.

Two outstanding research issues were also identified:

- the development of measures of efficiency and effectiveness for joint software reviews; and
- the development of methods for integrating and navigating through information from different sources.

The work presented in this paper also requires on-going development, and research into its effectiveness.
The development of tools to support the process may facilitate its use in software reviews, it may facilitate the collection of historical information, and it may assist in the evaluation of review process presented in this paper.

Finally, some of the supporting evidence for the recommendations in this paper comes from the field of software inspections and the applicability of these results to joint software reviews also requires investigation.

The author is conducting further research into joint software reviews. This work is being conducted under ITD's Joint Software Reviews (JORS) task and draws on literature from the fields of negotiation and organisational behaviour as well as the fields of inspections and reviews.

Acknowledgments

This report describes work conducted in the Joint Architecture Review for Project Llama and the ITD preparation for this review. The ITD review team provided insights into the review process which contributed to this work. For example, it was Dr Lakshmir Narasimhan's idea to consider the use of the RM-ODP framework. Thanks to all the ITD review team for their contributions. Two members of this team require special mention: Scott Davis and Dr Rudi Vernik. Scott Davis was the ITD representative at the Joint Architecture Review, where he distributed the surveys and collected information discussed in this report. Dr Rudi Vernik provided valuable insights into the conduct of software reviews during numerous discussions and during the review of early drafts of this report.

The other reviewers of this paper have also contributed to its current form. Conn Copas, thanks for your suggestions.

References

[ADO, 1996] Defence Project Management: Pitfalls and Pointers (Proceedings). Canberra. **Australian Defence Studies Center**.

[Basili, V R et al., 1994] Goal Question Metric Paradigm. Encyclopedia of Software Engineering, John Wiley & Sons. pp528-532.

[Bisant, D B, 1989] "A Two-Person Inspection Method to Improve Programming Productivity." *IEEE Transactions on Software Engineering* 15(10):pp 1294-1304.

[Brykczynski, B, 1994] Why Isn't Inspection Used, Posting to comp.software-eng, 16 Mar 1994, 12:07:34 GMT, .

[Butterfield, J et al., 1994] "Resolving cognitive conflict in requirements definition: a blackboard-based model and system architecture". *1994 ACM SIGCPR Conference*, Alexandria, VA, USA. ACM.

[Canale, R and Wills, S, 1995] "Producing professional interactive multimedia: project management issues." *British Journal of Educational Technology* 26(2):pp 84-93.

[CEPMAN 1, 1996] The Capital Equipment Procurement Manual AL12 (No. Revision AL12 (8/8/96)), Acquisition and Logistics, Australian Department of Defence.

[Cheng, B and Jeffery, R, 1996] Apparatus for an Experiment of Function Point Scenarios (FPS) for Software Requirement Specification, Technical Report (No. 96/4), Center for Andvanced Empirical Software Research, University of New South Wales.

[Deephouse, C et al., 1996] "Software processes and project performance." *Journal of Management Information Systems* 12(3):pp 187-205.

[Earnshaw, A A P, 1994] The Acquisition of Major Capital Equipment by the Australian Department of Defence: A Comparative Analysis, PhD, University of Canberra.

[Fisher, P D et al., 1997] Lessons Learned from Software Engineering Support to JORN, Client Report (Limited Release) (No. DSTO-CR-0050), DSTO.

[Gabb, A, 1997] Critical Design Reviews - A Pragmatic Approach, DSTO-DP-0455, 23 July 1997.

[Gabb, A P et al., 1991] Tailoring DOD-STD-2167A - A Survey of Current Usage, Technical Note (No. WSRL-TN-57/91), DSTO.

[Gabb, A P et al., 1992] Recommendations for the Use and Tailoring of DOD-STD-2167A, Research Report (No. ERL-0637-RE), DSTO.

[Gilb, T, 1996] "Inspection Failure Causes." *Testing Techniques Newsletter* October(On-line Edition).

[Gould, R B et al., 1994] "Determining human systems requirements for new technologies". *Integrated Logistics and Concurrent Engineering*, Montpellier, France. EC2.

[Grady, R B and Van Slack, T, 1994] "Key Lessons in Achieving Widespread Inspection Use." *IEEE Software* 11(4):pp 46-57.

[Hay, D, 1997] Software Development Plan for the Java Situation Monitoring (Project Llama) (No. ADI-Perth-65), ADI.

[Heemstra, F J, 1992] "Software Cost Estimation." *Information and Software Technology* 34(10):pp 627-639.

[ISO/IEC 9126-1, 1996] Information Technology - Software quality characteristics and metrics - **Part 1: Quality characteristics and sub-characteristics**, Standard (No. ISO/IEC 9126-1).

[ISO/IEC 10746-1, 1995] Open Distributed Processing - Reference Model - Part 1: Overview and Guide to use, International Standard (No. ISO/IEC 10746-1).

[ISO/IEC 10746-2, 1995] Open Distributed Processing - Reference Model - Part 2: Foundations, International Standard (No. ISO/IEC 10746-2).

[ISO/IEC 12207, 1995] Information Technology Software Life Cycle Processes, Standard (No. ISO/IEC 12207), ISO/IEC.

[Keil, M, 1995] "Identifying and Preventing Runaway Systems Projects." *American Programmer*(March):pp 16-22.

[Kingston, G, 1997] Joint Software Reviews, Research Report (Draft), Information Technology Division, DSTO.

[Lederer, A L and Prasad, J, 1995] "Causes of inaccurate software development cost estimates." *Journal of Systems and Software* 31(2):pp 125-34.

[Lim, K Y, 1996] "Structured task analysis: an instantiation of the MUSE method for usability engineering." *Interacting with Computers* 8(1):pp 31-50.

[Linington, P F, 1994] "RM-ODP: The Architecture". *IFIP ICODP'94*. Chapman Hall.

[MIL-STD-498, 1994] Software Development and Documentation, AMSC (No. N7069), Department of Defense (USA).

[MIL-STD-498, 1996] MIL-STD-498 Overview and Tailoring Guidebook, Department of Defense (USA).

[MIL-STD-1521B, 1985] Military Standard for Technical Reviews and Audits for Systems, Equipments, and Computer Software, Military Standard (No. MIL-STD-1521B), US Dept. of Defense.

[Mosemann II, L K, 1995] "Software Development: Quo Vadis?" *CrossTalk: The Journal of Defense Software Engineering* 8(11):pp 2-3.

[NASA, 1993] Software Formal Inspections Standard (No. NASA-STD-2202-93), National Aeronautics and Space Administration, Office of Safety and Misson Assurance.

[Noseck, H, 1994] "Cost-Benefit Analysis for Software Reuse - A Decision Procedure". *Ada in Europe. First International Eurospace-Ada-Europe Symposium Proceedings,* Berlin, Germany. Springer-Verlag.

[Porter, A A et al., 1995] "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment." *IEEE Transactions on Software Engineering* 21(6):pp 563-575.

[Raymond, K, 1995] "Reference Model of Open Distributed Processing (RM-ODP): Introduction". *IFIP ICODP'95.*

[Rutherford, J, 1995] "Software estimating and metrics-a pragmatic approach." *GEC Journal of Research Incorporating the Marconi Review* 12(2):pp 66-75.

[Sherif, Y S, 1992] "The Characteristics of Efficient Software Formal Reviews." *Microelectronics and Reliability* 32(3):pp 415-422.

[Shirey, G C, 1992] "How Inspections Fail". *9th International Conf. Testing Computer Software.*

[Walsh, M, 1994] "Why software continues to cost more-and what the IS auditor can do." *EDPACS* 21(11):pp 8-15.

[Warne, L and Hart, D, 1995] "Organizational Power and Information Systems Development: Findings from a Case Study of a Large Public Sector Project". *6th Australasian Conference on Information Systems.*

[Wheeler, D A et al., 1996] Software Inspection: An Industry Best Practice. Los Alamitos, CA, USA. IEEE Computer Society Press.

Appendix A: Survey Questionnaire

A survey was conducted to determine the ways in which the participants in the Project Llama Joint architecture review felt that the review could have been improved. This section contains the questionnaire used to solicit this feedback. It has two parts: the first part was completed by the survey co-ordinator, Scott Davis from ITD; the second part was distributed to all review participants.

ITD Study of Joint Software Reviews

Post Review Questionnaire

Coordinator's Sheet

Thank you for agreeing to coordinate the distribution of the questionnaire at this review. Please complete the following information to assist us in interpreting the results of this survey. If you are uncertain of any information please either indicate a range of values (eg 1-2 hrs) or mark that you value is approximate, or is an estimate, with an asterisk (*).

Background:

Project:______________________________

Review: ______________________________

Contact Details: ______________________

Dates:

Notification of the Review ____________

Review Material Received ____________

Review Agenda Received____________

Review Meeting: ____________________

Review Minutes Received ____________

Previous Review (if known)__________

Next Review (if known) _____________

Facilities etc:

Please describe the facilities and seating arrangements (ie phone, fax, secretarial & tool support, table and relative locations of contractors and project office personnel.)

____

____

____

Material:

What material was received before the review?

____

____

What material was received at the review?

____

____

Participants (Number of):

Developers ________________________

Clients ____________________________

Outside Experts ____________________

Observers _________________________

Users _____________________________

Other _____________________________

____

____

Agenda:

Please attach a copy of the original agenda if one is available.

Indicate the amount of time devoted to the following:

Developers Presentations (OHPs) _____

Prototype Demonstrations __________

Questions_________________________

Discussion and Resolution of Issues ___

Other (Please Specify) _____________

____

Indicate the number of:

Developers Presentations (OHPs) _____

Prototype Demonstrations __________

Questions_________________________

Issues Discussed and Resolved _______

Other (Please specify) ______________

____

ITD Study of Joint Software Reviews

Post Review Questionnaire

Please take a few minutes to answer this questionnaire.

This questionnaire forms part of ITD's studies on Joint Software Reviews. These studies are aimed at improving the efficiency of software reviews. That is the studies aim to recommend improved practices which will identify potential problems earlier, and reduce the time (staff-hours) spent preparing for, and in reviews.

The results of this questionnaire will help identify strengths and weaknesses in the current review processes and will be used as input to other studies which will investigate alternative review practices. Completion of this questionnaire is your opportunity to influence future review practices.

Thank you for your participation.

Gina Kingston
Software Systems Engineering Group

Please return the completed questionnaire to:
Gina Kingston
KSB 2-C-60
SSE/ITD/DSTO
PO Box 1500
Salisbury, SA. 5108.

Part A: Background

Project: ____________________

Review: ____________________

Date(s): ____________________

Organisation: ____________________

Contact Details (Optional):

____

____

____

Part B: Survey

What were your goals[1] for the review?

(Please attach an additional sheet if required.)

____

____

____

Did you achieve your goals?

(Indicate if some, all or none of the goals were Completely / Satisfactorily / Partially / Not achieved)

____

What was your role in the review?

____

Why did you attend the review?

____

What material did you receive before the review?

____

____

How did you prepare for the review? (eg What did you do with the material before the review?)

(Please attach an additional sheet if required.)

____

____

____

Explain your contribution to the review. (eg What questions did you ask?)

(Please attach an additional sheet if required.)

____

____

____

How do you think reviews could be improved?

Activities (#)	**More**	**Same**	**Less**
Presentations by the developer			
Questions			
Demonstrations			
Activities (Duration)	**Longer**	**Same**	**Shorter**
Preparation time			
Presentations by the developer			
Questions			
Discussion and resolution of issues			
Demonstrations			
Participants	**More**	**Same**	**Less**
Developers			
Clients			
Users			
Observers			
Outside Experts			
Secretarial Support			
Planning	**More**	**Same**	**Less**
Frequency of reviews			
Information received before the review			
Notification of the review			
	Same	**Alternative/Better (Describe)**	
Facilities			
Seating allocations			
Clearly Defined Goals & Objectives			
Clearly Defined Roles			
Tool Support			

[1] Goals should include personal as well as corporate goals including, but not limited to: the relative priorities given to different quality aspects, functionality, cost, progress and risk management.

What were the most/least useful activities in achieving goals?

Activity	Most	Least
Reading the documents before the review.		
Other activities before the review (Specify)		
Presentations by the developer		
Demonstrations by the developer		
Questions by other people		
Responses to questions you asked		
Reviewing the documents during the meeting		
Discussion of issues and risks and how to resolve or mitigate them		
Other (Specify)		

Do you feel you had adequate training, knowledge and experience to participate effectively in the review?

____

____

Please provide any additional comments in the space below.

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

Appendix B: Survey Results

A survey was conducted to determine the ways in which the participants in the Project Llama Joint architecture review felt that the review could have been improved. The questionnaire used has two parts: the first part was completed by the survey coordinator, Scott Davis from ITD; the second part was distributed to all review participants. Table 11 contains the results from the first part of the questionnaire and Table 12 - Table 15 contain the results of the second part of the questionnaire. A total of 6 responses were received. However, one person's response was incomplete. They answered the questions in Part A and stated their goals and role. They stated that their training was appropriate, but did not answer any of the questions in the two tables.

Table 11: Llama Review Characteristics

Characteristic	Description	
Review	Project Llama Joint architecture review	
Participants	Project Office	3
	Developers	7
	Contractor's Project Management	4
	Outside Experts	1 (DSTO)
Material Received	Prior to the Review	Software Design Document (SDD) Software Development Plans (SDP)
	At the Review	Copies of overheads (36)
Activities (All values are approximate)	Presentations	3 hrs
	Demonstrations	1 hr - Prototype
	Questions	1 hr (throughout the meeting)
	Discussion and Resolution of Issues	3 hrs 20 issues

Table 12: Participants Goals

Characteristic	Description	Responses
Goals	Ensure product will be useable and functional	1
	Cost	1
	Suitability of architecture	2
	Risk assessment	1
	Process Development	1
	Approval to proceed	1
	Present the design	1
	Address issues in the design	1
Goals Achieved	Completely	2
	Satisfactorily (the decimals come from one	2.75
	Partially (participant with multiple goals	0.25
	Not achieved	

Table 13: Participants Roles

Characteristic	Description	Responses
Role	Identify issues / Ask Questions	1
	Architect	1
	Project director	1
	Knowledge domain expert	1
	Reviewee	1
Reasons for attending	Invited organisational representative	1
	Present and lead discussion	2
	Project Director	1
	Receive feedback	1
Preparation (All values are approximate)	Read Materials	1
	Collected Comments from Others	1
	Produced material	3
Contribution	Asked questions	1
	Answered questions (*Plus one who would have if needed)	1 *
	Explained concepts	2
	Chaired meeting	1

Table 14: Planning

Characteristic	Description		Responses
Planning	Frequency	Same	4
	Information Received Before the Review	Same More	3 1 (Background)
	Notification of the Review	Same	4
	Facilities	Same	3
		Larger display	1
		Air Conditioning	2
	Seating	Same	3
	Goals and Objectives	Same	3
		More specific	1
		Not sure	1
	Roles	Same	3
	Tool Support	Same	3
Training		Adequate	2
		Technical Information	1
		Background missing	1

Table 15: Project Llama Recommendations and Impressions

Activities	Name	Usefulness			Improvement	
		Type	+	-		#
(All values are approximate) approximate)	Preparation	Own	2	2	Less	
		Others	1		Same	3.5
					More	0.5
	Presentations		3		Less	
					Same	5
					More	
	Questions	Own	2		Less	
		Others	3	1	Same	4
					More	1
	Discussion and resolution of issues		2	1	Less	1
					Same	3
					More	1
	Demonstrations		3		Less	
					Same	5
					More	
Participants	Developers	Less			1.5	
		Same			3.5	
	Clients	Same			5	
	Users	Same			1	
		More			4	
	Observers	Same			5	
	Outside Experts	Same			5	
	Secretarial Support	Same			4.5	
		More			0.5	

Appendix C: Project Llama Issues

The ITD review team identified several issues with the documentation received for Project Llama. These issues have been classified according to the framework used for the review. The framework presented in this section is a precursor to that described in Section 4.1 and the meanings of the viewpoint dimensions differ slightly from that described in this document.

The font of the issues indicates the knowledge domain in which they occur: those shown in italics are Software Engineering issues. Those in bold are Geographical Information Systems issues. Underliined issues come from the Human Factors knowledge domain and those in roman font are have not been allocated to a particular knowledge domain.

	Enterprise	***Technology***	***Engineering***	***Computational***	***Informational***
	Strategy Evolution Integration	Engineering Choices Maturity	Process Standards	Design Performance Quality of Service	User Requirements Functional Mappings
Understandable • decisions identified • decisions justified • decisions summarised • alternatives considered • alternatives complete	• *It is not clear how project Llama fits into the overall JCSE architecture, where it will used and where MILGEO will be used.* • The level of detail in the design document was not appropriate for a replacement system but would be okay for a concept demonstrator.	• *Supporting evidence for design choices is often lacking - eg for the choice of Java tools.* • *It is not clear why CORBA was chosen over RMI.* • The engineering choices are not justified in terms of the stated goals for the development.	• *The development approach not well-defined* • *It is not clear how much work will be required to update the development standards.*	• *The architecture is not well defined* • The difference between the proxy picture server and the snapshot server is not clear. • *A web application server is specified but its purpose is not clear.*	• **The justification for use of a GIS off-line is poor.**
Internally consistent		• *It is assumed that the Situation display is implemented as an applet sometimes and as an application at others.*			
Tools, etc • specified/complete • appropriate/ compatible • available • mature		• *The system relies on new Java features which may not be available in time.* • *The web browser was not identified.* • **Genamap may not meet internet and rendering requirements.** • **Is an overlay editor needed?**	• *What web development tools are proposed and why?* • *Version control tools were not specified.* • *Have the Java coding standards been defined?* • *Will JavaBeans or CBSE be used?*	• **Oracle may not meet performance requirements**	
Traceable • to requirements • to stated goals	• Requirements were not numbered so there was little tracability.		• The requirements were not testable.		
Sound • Are there better alternatives • Is anything missing • Will it be used	• The push for simplicity may have been taken too far. • Usage monitoring and analysis facilities would be beneficial. • Integration issues (eg training) not considered for MILGEO or project Llama.	• **The use of images as backdrops is limiting** • **Data management, storage and updates may be problematic.**	• *Was re-use of existing components considered?* • Procedures to manage the identified risks were not defined.	• **Map tools should be implemented as a map applet.** • **Vector maps could be implemented using an applet.** • Meeting performance requirements without an operational profile	• It may not be possible to find or convert data to the appropriate format. • Missing discussion of possible file formats for overlays. • + Lots more small issues • **Locatables consist of both**

	• The purpose and need for project Llama is not clear. • Has the Java/web approach been taken too far.			may be problematic. • The Intervals for generating transaction batches and snapshots are not defined.	**volatile and static data. If may be better if these ideas were separated.**
Correctness		• **Gzip only does compression.** • **ADG84 and WGS84 are similar.**			
Reliability		• **Use of document centric view rather than a database may introduce reliability issues.** • *At least a Transaction server is required.*		• *There is no discussion on how fault tolerance will be achieved.*	
Maintainability					• The implications of alerters need to be considered early.
Flexibility • sufficient • appropriate		• Range of mappings supported			• An ability to load maps independently of scenes restricts flexibility. • **Allowing extemsion of data schemas may be too flexible.**
Reusability					
Interoperability	• Project Llama is much simpler than the MILGEO system and should not be considered a replacement.				• Intergration between the decision aid module and the Situation Monitor appears to be lacking.
Security					
Safety					
Scakability		• **Large amounts of data may be necessary to store multiple copies.**		• **Use of thumbnail bitmaps will not scale.**	• Structured search to images and bitmaps is desirable.
Terminology (etc) • Appropriate • Internally consistent • Externally consistent • Understandable		• The distinction between scenes and sessions is not clear. • The use of Display & Business objects is unusual.			• Overlay is used where layering is more commonly used.

Joint Software Reviews - A Case Study from JP2030

Gina Kingston

(DSTO-RR-0156)

DISTRIBUTION LIST

	Number of Copies
AUSTRALIA	
DEFENCE ORGANISATION	
S&T Program	
Chief Defence Scientist	1 shared copy
FAS Science Policy	
AS Science Corporate Management	
Director General Science Policy Development	1
Counsellor Defence Science, London	Doc Control Sheet
Counsellor Defence Science, Washington	Doc Control Sheet
Scientific Adviser to MRDC Thailand	Doc Control Sheet
Scientific Adviser Policy and Command	1
Navy Scientific Adviser	Doc Control Sheet and distribution list
Scientific Adviser - Army	Doc Control Sheet and distribution list
Air Force Scientific Adviser	1
Director Trials	1
Aeronautical and Maritime Research Laboratory	
Director	1
Electronics and Surveillance Research Laboratory	
Director	Doc Control Sheet and distribution list
Chief of Information Technology Division	1
Research Leader, Military Computer Systems	1
Research Leader, Command & Control and Intelligence Systems	1
Research Leader, Command, Control and Communications	1
Research Leader, Advanced Computer Capabilities	Doc Control Sheet
Research Leader, Joint Systems	1
Head, Software Systems Engineering Group	1
Head, Information Architectures Group	1
Head, Information Warefare Studies Group	Doc Control Sheet
Head, Year 2000 Project	Doc Control Sheet
Head, Trusted Computer Systems Group	Doc Control Sheet
Head, Advanced Computer Capabilities Group	Doc Control Sheet
Head, Systems Simulation and Assessment Group	Doc Control Sheet
Head, C3I Operational Analysis Group	Doc Control Sheet
Head, Information Management and Fusion Group	1
Head, Human Systems Integration Group	Doc Control Sheet

Head, Distributed Systems Group	Doc Control Sheet
Head, C3I Systems Concepts Group	1
Head, Organisational Change Group	Doc Control Sheet
Gina Kingston	2
Publications & Publicity Officer, ITD / Executive Officer, ITD	1 shared copy

DSTO Library and Archives

Library Fishermens Bend	1
Library Maribyrnong	1
Library Salisbury	2
Australian Archives	1
Library, MOD, Pyrmont	Doc Control Sheet
US Defence Technical Information Centre	2
UK Defence Research Information Center	2
Canada Defence Scientific Information Service	1
NZ Defence Information Centre	1
National Library of Australia	1

Capability Development Division

Director General Maritime Development	Doc Control Sheet
Director General Land Development	Doc Control Sheet
Director General C3I Development	Doc Control Sheet
Director General Aerospace Development	Doc Control Sheet

Navy

SO (Science), Director of Naval Warfare, Maritime Headquarters Annex, Garden Island, NSW 2000.	Doc Control Sheet

Army

ABCA Strandardisation Officer, Puckapunyal	4
SO (Science), DJFHQ(L), MILPO, Enoggera, Qld 4051	Doc Control Sheet
NAPOC QWG Engineer NBCD c/- DENGRS-A, HQ Engineer Centre Liverpool Military Area, NSW 2174	Doc Control Sheet

Intelligence Program

DGSTA Defence Intelligence Organisation	1

Acquisition Program

Director General, Acquistion Management Systems	1
Director General, Command and Support Systems	1
Director General, Communications Systems	1
Director General, EW & Radar Systems	1
Director General, Strategic HF Systems	1
Director General, Underwater Warfare Systems	1
Director General, Surface Warfare Systems A	1
Director General, Surface Warfare Systems B	1
Director General, Ground Amphibious Warfare Systems	1
Director General, Aerospace Warfare Systems A	1
Director General, Aerospace Warfare Systems B	1
Director General, Aerospace Warfare Systems C	1

Corporate Support Program

OIC TRS, Defence Regional Library, Canberra	1

UNIVERSITIES AND COLLEGES

Australian Defence Force Academy	
Library	1
Head of Aerospace and Mechanical Engineering	1
Deakin University, Serials Section (M list), Deakin University Library, Geelong, 3217	1
Senior Librarian, Hargrave Library, Monash University	1
Librarian, Flinders University	1

OTHER ORGANISATIONS

NASA (Canberra)	1
AGPS	1
State Library of South Australia	1
Parliamentary Library, South Australia	1

OUTSIDE AUSTRALIA

ABSTRACTING AND INFORMATION ORGANISATIONS

Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts, US	1
Documents Librarian, The Center for Research Libraries, US	1

INFORMATION EXCHANGE AGREEMENT PARTNERS

Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1
SPARES	5
Total number of copies:	**68**

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

Joint Software Reviews - A Case Study from JP2030

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

Gina Kingston

5. CORPORATE AUTHOR

Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury SA 5108

6a. DSTO NUMBER DSTO-RR-0156	6b. AR NUMBER AR-011-020	6c. TYPE OF REPORT Research Report	7. DOCUMENT DATE July 1999

8. FILE NUMBER N8316/7/34	9. TASK NUMBER DST 97/243	10. TASK SPONSOR CITD	11. NO. OF PAGES 49	12. NO. OF REFERENCES 42

13. DOWNGRADING/DELIMITING INSTRUCTIONS

N/A

14. RELEASE AUTHORITY

Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Acqusition of computer software
Reviews
Computer programs

19. ABSTRACT

Joint Software Reviews - between acquirers and developers - are an important approach in the acquisition of software intensive systems. These reviews are poorly understood and often conducted in an inefficient, *ad hoc* manner. This report describes some aspects of the design review for Project Llama (JP2030).

Information Technology Division were asked to provide input to this design review and formed a multi-disciplinary team to assess the design. This document describes the process attempted by the ITD review team and compares this to the actual process used. The benefits and limitations of the process are discussed as well as potential improvements.

A survey of participants at the design review meeting was conducted and the results of this survey are also included.