University of Oxford
Software Engineering Programme

Is Partnered Software Support appropriate for Military Aerospace Platforms?

A Dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science in Software Engineering

LB Cooper
Kellogg College
July 2007

ABSTRACT

Over the past 3 years the Royal Air Force (RAF) has added the Typhoon aircraft to its operational inventory, undertaken major software updates to both Tornado and Harrier, and is close to deploying the Nimrod MRA4 and Joint Strike Fighter. The traditional approach of supporting software with ad-hoc contracts and isolated organisations is now considered to be unsustainable. The approach is too expensive and operationally inefficient in that it takes too long for a software enhancement to be realised.

This Dissertation analyses the need to change from the present ad-hoc methods of supporting software, that were derived from those used for hardware development, towards a more cost and operationally effective approach for the future – termed Partnering. It looks at the credible options for software support and sustainability, taking into account the drive for “Bang-for-Buck” and whether Partnering in an unpredictable software environment is appropriate for the future.

The Dissertation concludes that by exploiting the strengths of individual organisations supported by an appropriate contract, costs can be reduced without compromising operational integrity. Indeed, if the correct skills and infrastructure are deployed then the time taken to implement software changes can be reduced.

The background analysis and recommendations within this dissertation are now being used to guide software support solutions for platforms undergoing change today and for guiding the approach for the future.

ACKNOWLEDGEMENTS

I would like to acknowledge my colleges and friends from the Royal Air Force Software Support Team for their encouragement and guidance in the production of this dissertation, also my wife for her tolerance when reviewing the many drafts put in front of her.

The author confirms that: this dissertation does not contain material previously submitted for another degree or academic award and is the author’s own work, except where otherwise stated.

The opinions documented within this dissertation are solely attributable to the author and do not necessarily represent the policies, or future direction, of the Ministry of Defence in anyway whatsoever.

TABLE OF CONTENTS

TABLES

FIGURES

CHAPTER 1 - INTRODUCTION

OVERVIEW

The software that is presently flying with the RAF ranges from the Nimrod Maritime Reconnaissance Mark 2 (MR2), which utilises CORAL 66 & SPIRIT III software stored on a 128K core processor, to the Typhoon aircraft that has just entered RAF service which uses Object Oriented Design (OOD) technology, C++ and ADA. This range of software technologies spans over 40 years and needs to be enhanced on a regular basis through software modification to maintain the different air platforms operational advantage. With such a range of technologies, architectures, languages and legacy hardware, the required level of support can prove problematic and costly to define and maintain. Due to their diversity any one solution cannot be assumed to be appropriate for all aircraft types. Software Support and the ability to change characteristics are vital to the continued effectiveness of platforms and the ability of the RAF to meet their ever-changing commitments. The purpose of this dissertation is to assess if a collaborative-partnered software solution could address the issues that are being experienced today. With partnering being defined as:

> *"Partnering is essentially the development of new, much more co-operative long-term relationships between MOD and Industry. Partnering is based on the following key principles: Joint Vision, Openness, Honesty and Trust. Partnering requires commitment throughout the business units of the MoD and its suppliers and is not an easy option. Proactive attitudes and contributions will be required on Partnering from Project Initiation in order to obtain improved performance and shared benefits."* [AMS07]

It should be recognised that the majority of modifications or enhancement to Military platforms do not fail. They are just late by differing amounts of time, or cost more than originally anticipated. This is supported by the fact that operational capabilities once authorised are realised eventually, you cannot have half a capability, either the software has the latest "wiz-bang-flash" or it does not. Partnering can be summarized as an approach to reduce time and money overruns, to reduce the probability of delay or failure. We just need to realise that software change throughout its life and that its how we plan for that changes that make the difference; this sentiment has been captured below:

> *"To introduce software to service effectively we must bow to the inevitable and admit that the software will change in service; only then we can concentrate on the real problem of putting in place an efficient [and cost effective] process to manage and implement that change.*
>
> - *It must be remembered that **Software** is called software because it is **SOFT**!*
> - *When software is working it is fulfilling its **FUNCTION***
> - *When **Software** is being changed it is fulfilling its **DESTINY !**"* [BED95*]*

BACKGROUND

The information for this dissertation was derived during a 2 year period of activity that defines the future requirements and direction for several Military projects that are undergoing changes in their Software Support solution. The major projects are as follows:

Harrier GR4 Ground Attack. This aircraft has just undergone a major avionics software update and has just entered Military service after the update. The past Software Support option is now not considered cost effective for the future.

Tornado GR9 Ground Attack. This aircraft is now in-service after undergoing a mid-life upgrade and the Software Support option selected, although initially agreed, is now not considered financially acceptable.

Typhoon Air defence. The Typhoon is presently undergoing acceptance by the Military and its support options are being derived. Due to the complexity of the platform the traditional support options are considered technically unacceptable or too expensive.

Joint Strike Fighter (JSF). The JSF is being developed by Lockheed Martin for the United States and the United Kingdom. It is considered to be the next generation of Military aircraft and the support is in its embryonic stage. There is now an opportunity to shape Software Support for the platform.

Tactical Unmanned Air Vehicle (T-UAV). The T-UAV project is presently defining its software solution and the information within this dissertation will be used to guide their options and way forward, both for software design and long-term support.

As can be seen from the air platforms identified, there are currently some major aircraft updates taking place. For this reason a task was embarked upon to assess if the traditional ad-hoc methods of supporting software is still acceptable for the future Software Support environment, or is a joint partnered solution now appropriate. This dissertation is based on work undertaken to analyse the platforms identified above, and takes into account present Government policies, past studies associated with Aerospace Software development or support, and interviews or workshops attended by the Author that are associated with the previously identified platforms.

DISSERTATION STRUCTURE

The structure of this dissertation and a brief description of the content is captured below:

Chapter 2 provides a common understanding of the Application Area with its constraints and the stakeholders that need to be managed. With Chapter 3 discussing what Software Support is and the inherent reliance on the initial design process and the factors that contribute to the definition of a supportable solution. It also recognises that 90% of functionality is now enabled through software and that software is the largest element of Whole Life Costs (WLC). These factors need to be balanced against both the expected change traffic and the unexpected changes of Urgent Operational Requirement that are experienced by platforms today.

Chapter 4 provides information on the present policies within the Military software environment and recognises that Integrated Logistical Support (ILS) is now the primary method of defining support solutions today with Chapter 4 capturing the main support solutions that are in use today across different platforms. Chapter 5 touches on the past and future trends in software with their perceived impact on software solutions. It also looks at why the present methods of providing software support are now unacceptable and how Industry and the Military perceive each others abilities.

Chapter 6 captures the recommendations from previous studies along with the need for the Military to remain an "Intelligent Customer" with its core capabilities and the Requirements that need to be met to take any proposed solution forward from today's method of contracting for software support to a partnered solution. Within Chapter 7 is a recommended solution that meets the Requirements and describing what the solution looks like. The proposed solution addresses the needs for Domain Knowledge retention and responsiveness, whilst recognising the drive to reduce WLC. These Requirements need to be balanced against the Risks and Benefits of a partnered solution identified in Chapter 7, with Chapter 8 providing information from both a Military and Industrial perspective on whether Partnering is an appropriate solution.

This Dissertation provides sufficient justification to conclude that by exploiting the strengths and removing the weaknesses of individual organisations as defined in Chapter 4, WLC relating to the support and sustainment of Airborne software can be considerably reduced, without compromising operational integrity. This conclusion is based on the information captured within the chapters outlined above and information gathered from workshops, interviews and reviews of software organisations.

From the taught element of the MSc, various courses have contributed to the content and direction of this Dissertation. They have provided guidance on Risk and Requirements analysis, how a methodical approach is required when tackling unfamiliar problems such as is applied to Safety, Managerial, or Quality assessments of organisations.

CHAPTER 2 - SOFTWARE APPLICATION AREA

INTRODUCTION

The application area to be investigated in this dissertation is based around the aircraft utilised today by the United Kingdom Royal Air Force (RAF) and the support of the software that is hosted on these air platforms. To maintain a universal understanding of this Software application area, this Chapter will define: the scope of Software Support; the Stakeholders who have a vested interest in Software Support; the inherent constraints, with the Chapter concluding with a summary, as represented by Figure 1 below.

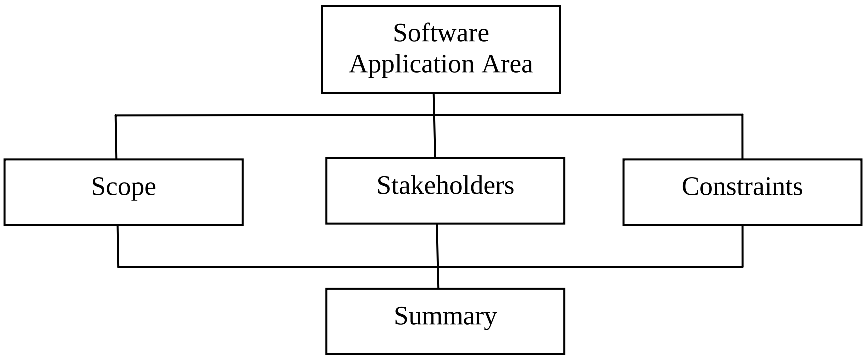

Figure 1 - Chapter 2 Structure

SCOPE

The RAF utilise large aircraft systems that are often integrated with distributed networks, these attract a high level of financial commitment from the outset. From this perspective RAF Software Support covers a wide spectrum, ranging from imbedded platform software to Information Technology (IT) and infrastructure that is used to support the development and exploitation of software systems. For the purpose of this dissertation, the software to be considered is limited to software located onboard an aircraft; loaded on a flight-by-flight basis to enhance mission capability. This software is commonly known as an Operational Flight Program (OFP) which historically has a life span of over 25 years throughout which it is modified.

Software Support for an OFP covers the whole “V” lifecycle [MCD94] from initial requirements capture and contracting through to development, qualification and finishing when the end product is put to its intended use. This process is supported by a Query Answering and Problem Evaluation services that is used for both investigation and rapid prototyping during the initial stages of assessing the need for a modification or enhancement to the software.

The scope of the organisations that directly change software range from the Original Equipment Manufacturer (OEM) generally termed Industry, with its reliance on subcontractors, to the Military’s own Software Support Teams (SST). This dissertation assesses their relationship, the challenges and the appropriateness of forming a single partnered team consisting of both organisations to support software development, sustainment and enhancement[1].

[1] Whenever software is changed through modification or enhancement, there is a change to Requirements.

STAKEHOLDERS

There are several stakeholder groups who have a vested interest in Software Support. These stakeholders have either an economic, political or operational interest with stakeholders often not considering each others interests due to their different priorities. Any change from the historic silo[2] positions towards partnering should be supported by policy and the financial levels needed to effect change. An overview of the different stakeholders and their position in the hierarchy is represented in Figure 2 below[3]:

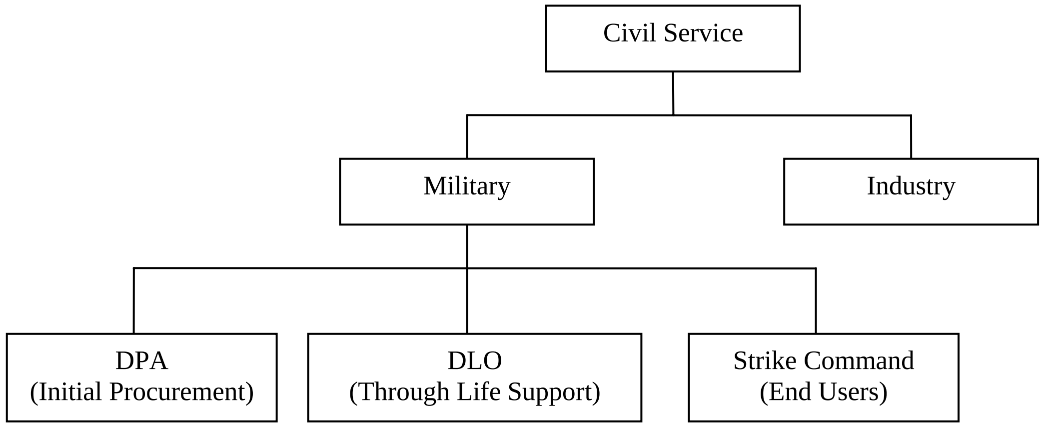

Figure 2 - Software Stakeholders

Civil Service. This stakeholder group consists of the Secretary of State for Defence, National Audit Office (NAO) and their associated agencies that directly interact with the Military. They provide direction on required capabilities that change the software characteristics and dictate the Military theatres of operation that the platforms are required to operate in. This group includes the Director for Equipment Capability (DEC), who has overall responsibility for Military capabilities to the government. It should be noted that the Military infrastructure has become increasingly financially and operationally accountable to the Civil Service agencies, therefore straining the relationship between the Military and the Civil Service. It is considered that this level of accountability is rightly justifiable, given the financial commitment that Software Support attracts.

Military - The Defence Procurement Agency (DPA). The DPA procures new major equipment and sponsors new technologies by the use of an Integrated Project Team (IPT) to coordinate change. An IPT is a multi-disciplined group containing people with a range of skills, from financial, contracting and engineering personnel through to the end-users who focus on a particular platforms needs. The DPA IPTs are mostly interested in meeting initial procurement costs and ensuring that a platform, or key technology, is available by a given In-Service Date (ISD)[4]. Past reports by the NAO have directly criticised the DPA for not addressing Whole Life Cost (WLC) requirements, they mainly considering the initial procurement needs not the total cost of ownership.

[2] Silo potions represents stakeholders that are comfortable in their present situation and are unwilling to change.
[3] On the 1 Apr 07 the DLO and DPA combined form Defence Equipment & Support (D&ES), the functional responsibilities still within DE&S.
[4] ISD in the point in time when new capabilities are transferred from development to the end-users.

Military - The Defence Logistics Organisation (DLO). The DLO is responsible for supporting platforms during their In-service life, they do this by the use of DLO IPT who's main focus is support and the sustainment of the platforms capabilities. Also to ensure that the software continues to be effective over its lifecycle, this support lifecycle is historically at least 25 years for an aircraft. Within this group, the Military has various Software Support Cells (SSC) that assist Industry and SST's that carryout the full software lifecycle [MCD94] in parallel with the Industry organisation. This duplication is against the principles of forming long-term partnerships and increases the financial footprint for through life support.

Military – Strike Command. This group represents the pilots and engineering personnel, who interact with the software on a daily basis on the Front-line[5]. These people rely on the continued efficiency and ability of the software to meet its stated, or implied, functionality to allow them to meet their operational commitments. Their prime concern is the end capability, not how the capability is supported or reaches the end-users.

Industry. This stakeholder consists mainly of the OEM that developed the aircraft and the sub-contractor organisations that assisted the OEM with the initial contracted software development. These industrial companies are primarily responsible to their shareholders but have a vested interest in the enhancements and continued airworthiness of the platform for their long-term revenue planning and financial success. The OEM is normally tasked by the DLO to provide support for the platforms throughout their operational life, this however can change this if Industry proved either too expensive or fails to meet support requirements, this issue is further elaborated on later in the dissertation.

It can be argued that by forming a IPTs, the DPA and DLO formed internal partnerships that have benefited from closer working relationships and bringing knowledge together benefiting their individual projects. Possibly now the forming of relationships needs to migrate from the IPT level down to the actual Software Engineering to allow greater benefits to be achieved. There have been previous software studies carried out that have looked at the issues with software development and support. These studies are assessed during Chapter 6 along with the need to retain core capabilities and for the Military to retain the ability to be an "Intelligent Customer" to meet its long-term aspirations. This information contributes to the debate on whether to increase the levels of communication between organisations

CONSTRAINTS

There are a number of constraints imposed on Software Support that need to be managed. These are not specific to the Military environment but need to be understood to appreciate why a partnered support organisation could be desirable. These constraints are categorised as follows with design and supportability factors captured in Chapter 3:

Financial constraints. As with any organisation there is financial accountability with spending being justified to higher authorities. For the Military this authority is the NAO. The Military budget for 2004/2005 was £32 billion [UKDS05/1] with the allocation for the support of platforms being in excess of £11 billion [UKDS05/2], this level of funding rightly attracts significant levels of scrutiny. However, this scrutiny increases the time from the identification of a software change to its embodiment, this time delay impacts on

[5] The Front-line is the Squadron organisation that operates the aircraft.

operational effectiveness and the ability to realise a capability when it is actually needed. If a financial case could be presented to the NAO by an organisation that reflects the software business, this could over time reduce financial scrutiny. The levels of financial commitment that software now attracts is expanded upon both in Chapter 3 under "Finance" and Chapter 5 under "Trend Implications". These Chapters demonstrate that software enables the majority of functional capability as well as attracting the majority of the cost.

Political change. Based on personal experience on a Project Team[6], whenever there is political uncertainty the level of directive guidance that the Military receives varies negatively in both quality and focus. This impacts on Military and industrial relationships which in turn affects local economies where Software Support activities are conducted. During the 2004/2005 financial period, the Military agreed to 37,000 contracts amounting to over £15.8 billion with UK Industry with the majority of funds directed towards support contracts. These contracts are directly responsible for employing over 307,000 civilians [UKDS05/3] in various technology areas and for these reasons whenever there is political uncertainty, or over zealous budgetary constraints, there is a risk to civilian jobs. This understandably strains relationships and can force organisation to become entrenched behind contractual barriers. Therefore a balance should be struck between accountability and meeting the operational needs, if stability can be provided from within an appropriate software organisation that represents both the interests of the Military and their industrial counterparts, this could help mitigate against political issues and uncertainties.

Changing Threats. Until the early 1990s there was one main threat that the Military directed its capabilities towards, the former Union of Soviet Social Republics (USSR). With the collapse of the USSR and the rise in international terrorism, the types of threats that need countering has changed. The threat has changed from a Super-power to small groups with variations in technology and capability, the effect is an increase in software change traffic[7] that is required to counter the diversity of threats. The types of modifications and frequency of release are now less predictable than during the Cold War, which is encouraging organisations to reassess how software is produced and supported.

The knock on effect is an increased difficulty in contracting Industry for software modifications; the use of traditional contracting and development methods where a single contract is let for a single enhancement no-longer meets the needs of the stakeholders – software release is now too slow and expensive. To address the need for an increased number of software modifications, at reduce cost; organisations are considering alternatives to traditional ways of developing and releasing software alternatives such as partnering which is considered more agile that ad-hoc contracting.

Obsolescence. Due to an in-service life in excess of 25 years, Military aircraft encounter obsolescence issues from day one, these relate to; hardware; software; tools and skills. An example is reduced Industry support, and trained people, for the ADA language. ADA is being used in Typhoon that has just entered service and the Joint Strike Fighter that's in development; both platforms are already encountering language obsolescence issues with ADA. This illustrates that Military aircraft are inherently obsolete due to their long lifespan, which makes them notoriously difficult to support through life as skills and

[6] A Project Team is responsible for developing a new aircraft and is part of an IPT.
[7] Change Traffic is the number of times that software is modified over a set period of time, e.g. 3 times/year. Levels of Change Traffic drives the Software Support solution.

knowledge transfer to the next new technology or platform. The challenge is to form an organisation that can combat obsolescence; this can be achieved through Partnering and is expanded in Chapter 7 under "Software Domain Knowledge".

Safety. One of the key considerations with aircraft software is safety. Any platform must be capable of operating over populated areas without injuring the crew or non-combatants, therefore capabilities that software enables undergoes formal safety analysis that takes time and costs more to develop than software that is not capable of endangering life. Presently there are three groups who assess software safety before its released, the original software development team, Industry Independent Safety Assessor (ISA) and the Military ISA. This could realistically be reduced to two groups as is common practice, but only by forming a relationship based on trust or a joint ISA organisation.

SUMMARY

In summary, the application area covers the sustainment of present and future Military capabilities through changing software, with a lifecycle of at least 25 years. To meet these commitments, various stakeholders need to be considered, budgets complied with, and an appropriate support organisation needs to be deployed. All this must happen whilst ensuring that the pilots, ground-crew, and non-combatants are not exposed to unacceptable levels of danger when the Military carry out their primary duties. This Chapter has also identified constraints that could be reduced through forming appropriate relationships, or partnerships between organisations. In the next Chapter we will look at the considerations for both software design and support that platforms need to consider during their lifecycle.

CHAPTER 3 - SOFTWARE SUPPORT CONSIDERATIONS

INTRODUCTION.

The purpose of this Chapter is to define the Software Support considerations that are applicable to the long-term support of aircraft software. The Chapter will concentrate on the interpretation of Software Support, initial design factors, and additional supportability factors that need to be considered during the definition of the support solution. This will conclude with a summary as shown on Figure 3 below.

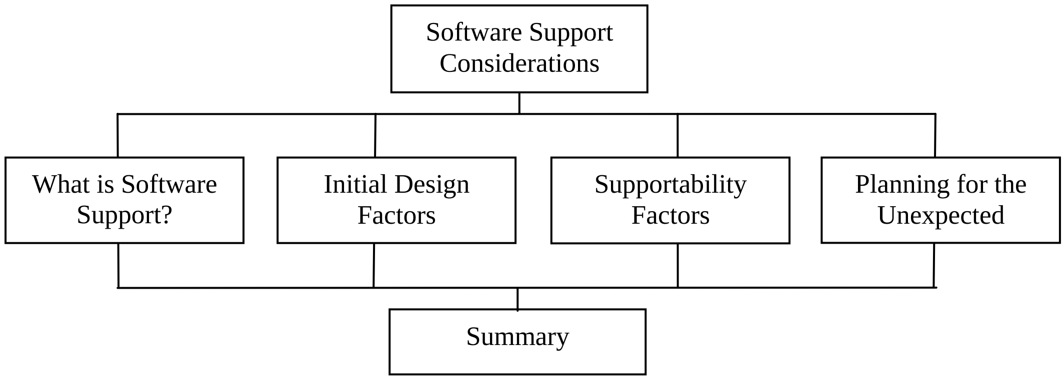

Figure 3 - Chapter 3 Structure

WHAT IS SOFTWARE SUPPORT?

Software Support covers the whole spectrum from initial requirements capture through development and qualification until the software is put to its intended use, it also covers problem evaluation, query answering and the management and authorisation of software changes. How the organisations that carryout software support are organised is amplified in Chapter 4, under "Present Software Support Organisations". The Society of Automotive Engineers (SAE) defines the requirement for Software Support as follows, with the definition of the customer in the Military context being the DLO IPT:

> *"There are two aspects to meeting a Customer's Software Supportability requirements. The first is ensuring the delivery of a product with the appropriate design characteristics to facilitate the expected demand for through-life change and enhancement. The second aspect is the provision of a support capability that satisfies the Customer's quality of service needs at an acceptable cost. These are interrelated goals that should be addressed through a co-ordinated approach to Software Supportability planning."* [SEI98]

Software Supportability is a step beyond Software Maintenance in that as well as addressing the continual operation of software and the introduction of enhancements, Software Supportability addresses the infrastructure, skills and design to facilitate maintenance activities through life. Software Support anecdotally attracts 60%-80% of through life costs, therefore the supporting infrastructure and organisation can be larger than the original development organisation, both in terms of size and capability. Software Maintenance activities need to be provided by an appropriate organisation that address cost, timescales and operational needs. This organisation should concentrate on controlling the day-to-day function of the software, Software Operational

Support (SOS), as well as controlling the modification process [PFL01]. This modification process should consider new enhancements as well as how to perfect the existing functions and preventing the original software from degrading to unacceptable levels.

The types of maintenance that are conducted within the software organisation are placed into five different categories which are universally accepted within the software community and explicitly stated by the ISO/IEC community. These categories are listed below with their associated representation in Figure 4 below:

- *Corrective maintenance: The reactive modification of a software product after delivery to correct discovered problems.*

- *Adaptive maintenance: The modification of a software product, performed after delivery, to keep a software product usable in a changing environment.*

- *Perfective maintenance: Modification of a software product after delivery to detect and correct latent faults in the software product before they are manifested as failures.*

- *Preventative maintenance: The modification of a software product after delivery to detect and correct latent faults in the software product before they become operational faults.*

- *Maintenance enhancements: A modification to an existing software product to satisfy a new requirement.* [ISO/IEC06]

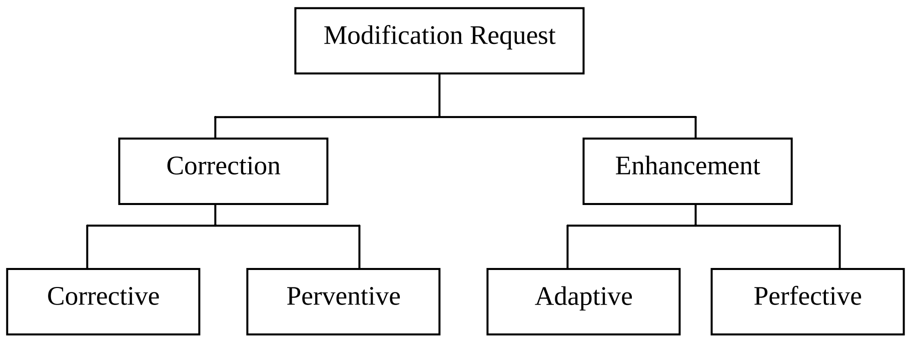

Figure 4 - SEC/IEC Modification Request categories

All of the activities required after initial deployment depend either on the end-users identifying a fault or functional failure, or the supporting organisation identifying and rectifying faults before they become apparent to the users. For a support organisation to provide an appropriate service there must be processes in place that are appropriate and usable. There are various software models in use today, but the Military were unable to locate a model that covered both Software Support and Data that matched their needs, they now use a Software and Data Support Model [JAP100A]. The model goes further than traditional development models in that it now covers Data, Qualification, the interface with External drivers and the platform itself which other identified models do not address. The model is used to determine **what** functions are required and not **who** will provide a support function, this concept is easy to understand but it's a change in perspective when contracting for support, which historically has the emphasis on **who**. This model is represented in Figure 5 below [JAP100D]:

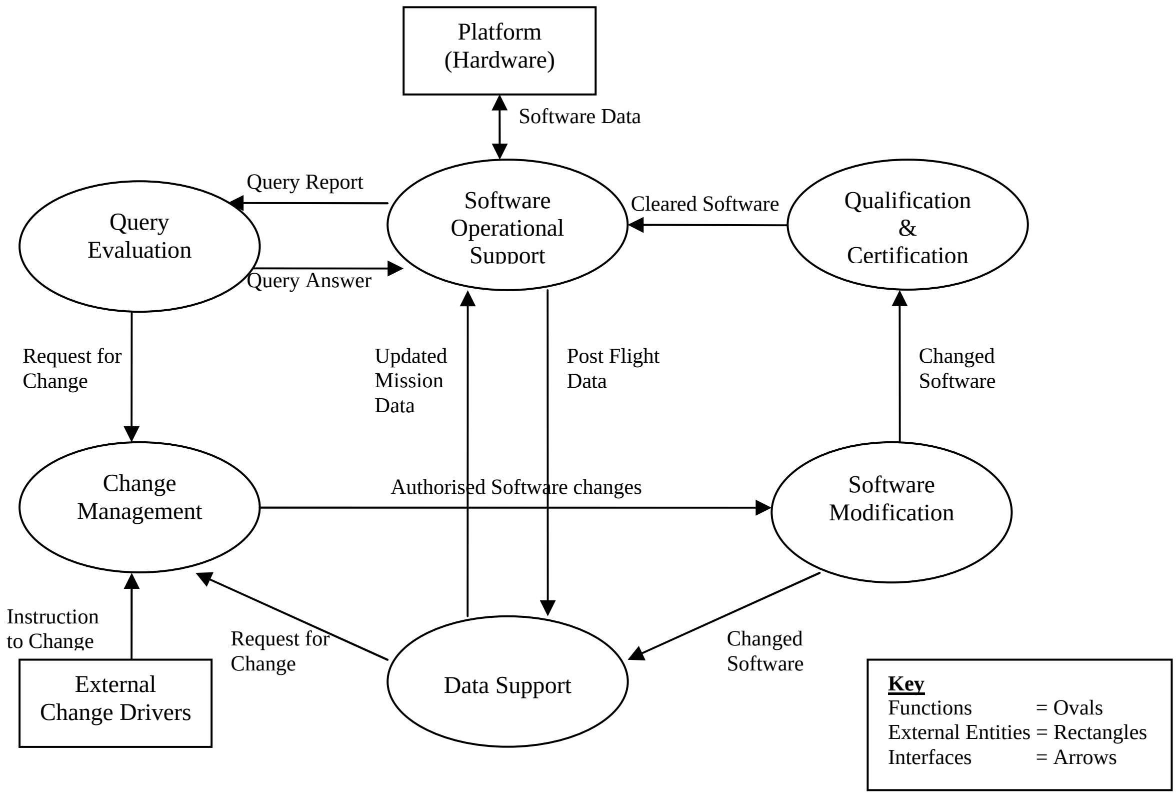

Figure 5 - Software and Data Support Model

The model in Figure 5 is functionally based rather than organisationally based, which allows individual functions to develop interfaces and capabilities that are interdependent. It is not considered appropriate to detail all the desired functionality of the model in this dissertation, but a high-level description is as follows. Also note that Figure 5 is used at various points in this dissertation representing different support options:

External Entities. The external entities provide inputs/outputs to the Support processes from both the Platform environment and the External Drivers. These are expanded upon below:

- Platform Hardware Host. The Platform host is the target hardware in which the software resides, with the software being loaded by a variety of means, some examples of loading methods used in-service are as follows:

    - Removable hard discs.
    - PCMCIA cards.
    - Tape cartridges.
    - Dedicated equipment.
    - On an Industry site.

- External Change Drivers. The term External Change Drivers refers to a source of change that originates from outside the users or modification environment. Examples of these change drivers include the need to [8]:

    - Enhance functionality due to changing threats.
    - Sustain a capability due to changing environments.
    - Change capability of a platform.
    - Reduce Whole Life Costs (WLC) due to reduced budgets.
    - Maintain Industry technology advances.

Software Operations Support. Software operation contains the activities that the users carry-out when interacting with the host platform. Examples of the functions are as follows:

- Actions necessary to "*load, re-load, replicate, label, store, distribute, recall and carry-out any handling activity on software, data or firmware*". [JAP100D]

- Data and software preparation, release and recovery.

Query Evaluation Function. The Query activities carry-out a vital role in the software lifecycle and are the prime interaction with the users. The main functions are as follows:

- Evaluate and filter queries and problems from the Software Operations function in order to identify the cause of any occurrence. This will allow people to become Subject Matter Experts (SME) due to the Domain Knowledge that they will acquire. This key knowledge provides an organisation with its capability to make informed engineering decisions.

- Investigates the nature of occurrence, removes any duplicates and justifies acceptance or rejection of the occurrence for all problems and queries. The evaluators must assess the operational benefits, costs and risk of adoption or rejection of the query or problem.

- Decide on the initial prioritisation for resolving any problems or queries, with some queries or problems generating a Request For Change (RFC) to be forwarded to the Management Function and others queries not requiring progression.

- The evaluators can solve many immediate problems by identifying and producing workarounds to prevent degrading the current capability. This is an interim solution to meet an operational, because software modification can be a lengthy process.

Change Management Function. The Management function manages operational capability and readiness through the control and prioritisation of change requests. Management deals with user-initiated changes that have been evaluated as well as externally driven change needs, such as major upgrades or addressing hardware obsolescence issues. It is common practise to use a Software Configuration Management Board (SCMB) comprising of IPT members supported by SME, from Industry and the

[8] These change drivers are forcing improvements in both software processes and contracting.

Military, who evaluate proposed software modifications. They work together across organisational boundaries to make informed decisions and to authorise changes where considered desirable. The Military brings Operational Domain Knowledge[9] with Industry providing in-depth systems and software Domain Knowledge and the relevant IPT controlling the budgets and representing the DEC.

Software Modification Function. The Software Modification function has responsibility for the implementation of authorised changes. In addition to this function, it has to assess its own capacity for tasking and communicate this capacity to the Change Management function, this capability information is used during the assessment and prioritisation of changes by management. There are presently different options for providing this function, Industry, The Military or a hybrid combining of the two groups.

Qualification and Release. The Q&R function is responsible for ensuring that the software is acceptably safe for use, and that the RFC requirements have been satisfied. The Integrated Project Team Leader (IPTL) is responsible for ensuring that a revised Safety Case is developed by the Q&R function and that it provides acceptable levels of evidence relating to airworthiness, as detailed in JSP553[10]. Q&R is an iterative activity that interacts with the full development or modification lifecycles. Its main functions include, but are not limited to; the evaluation of Aircrew workloads, platform performance and standards of the user documentation. This is complemented by ensuring that the required testing evidence was captured during each stage of the software lifecycles and that both the original Software Change Request (SCR) and any safety requirements have been satisfied during development.

Data Support Function. Data Support refers to information, both mission and engineering related, loaded to or downloaded from the software host. The Data Support activity captures all the activities necessary to create, preserve, modify, analyse and release data, examples of mission and engineering data are as follows:

- Electronic Warfare and intelligence data.
- Maps and whether data.
- Engine, fatigue, and component usage data.
- Weapons and sensor information data.
- Functional failures and faults data.

To summarise “What is Software Support”, the overall need is to provide an appropriate functional service that can analyse, prioritise and implement changes to the software under consideration. This capability need not be one organisation, but if it’s dispersed, the interfaces should be well defined and understood to prevent process and engineering errors occurring. This section looked at the supporting organisation and infrastructure; the next section will look at the design factors that need to be considered during initial software development.

[9] Domain knowledge – Relevant skills and experience on, or associated with, a platform.
[10] A JSP is a Joint Service Publication used by the Army, Royal Air Force and the Royal Navy.

INITIAL DESIGN FACTORS

With the best will in the world, Software Support cannot be designed in effectively after a product has been built and released. For this reason the following section outlines the design features that influence Software Support decisions and will stay with a product throughout its lifecycle. The past trends in software design are evaluated within Chapter 5, followed by a judgement on future trends and their impact on a partnered software solution.

Architecture. The architecture of a system is defined by the designers at the initial concept stages and is unlikely to change by any great degree throughout the life cycle. For this reason the designers and support organisation need to understand the concepts of Software Supportability and the circumstances which drive supportability. The architecture should facilitate the need for change and growth.

Identifying change traffic levels. By analysing the new systems "nearest neighbour"[11], you can identify the functions that are likely to change throughout the life of the software. For example, if you compare the software change traffic of an anti-skid control unit, that changes traditionally only on initial flight test to a main Head-Up Display (HUD) computer that has software that changes more than any other system, the change traffic would be vastly different. For this reason, the more effort should be spent designing the HUD software and its interfaces rather than the Anti-skid Control Unit because it is expected to experience constant change. This is one of the main reasons for carrying out change traffic analysis, the other being to determine which method of support you should adopt for a piece of software.

Modularity. Modularity has been widely taught within Universities as a method for designing software in manageable chunks, to promote re-use and the long-term supportability of software. One of the main drivers for modularity must be to design the software in a manner that will allow areas of expected change, identified by change traffic analysis, to be readily changeable.

Software Host update. The architecture of the software and physical design dictate the order that software can be updated. It is important to reduce the cost of physically updating the software on a host and the time it actual takes for a piece of hardware to be updated that is physically fitted to an aircraft, taking into account the secondary effects of possible removing the aircraft from operational use. The order of change method is listed below with the most desirable first. As you move down the list, the cost and time for changing software increases:

- Algorithms and data tables segregated from the code in specific modules. Software is loaded on to the target hardware by the users on a flight-by-flight basis when uploading the OFP.

- Algorithms and data tables segregated from the code in specific modules. Software is loaded on to the target hardware by the users using dedicated support equipment **connected** to the platform.

[11] This is a system that is presently in use that provides similar functionality or capability.

- Algorithms and data tables segregated from the code in specific modules. Software is loaded on to the target hardware by the users using dedicated support equipment **off** the platform.

- Algorithms and data tables segregated from the code in specific modules. Software is loaded on to the target hardware within the OEM environment.

- Algorithms and data tables embedded within the main software code. The only method of change software being to rewrite and qualify the software which is loaded on to the target hardware within the OEM environment.

Language selection. ADA is becoming increasingly difficult to support, where previously there were numerous programmers and analyses, now the language is no longer commonly taught by Civilian educational organisations and the Military have become the main user of the language. Subsequently, support has become costly as programmers and analysts are becoming hard to find, demanding premium salaries. Industry also passes on increased language support costs and is reluctant to commit to new support contracts for legacy languages. This means that aircraft like Typhoon, which is about to enter service, contains an already obsolete language before the platform has been put to its intended use. Whilst there is no desire to use the latest version of a language that's in its embryonic stage, there is also a need not to use languages that are sun-setting either, like ADA. Therefore a balanced judgement has to be made based on experience and taking into account the long-term supportability of a software driven system and the language acceptability, with Partnering being one approach to mitigating against language obsolescence, as described in Partnering Benefits in Chapter 7.

Safety features. Safety features are often seen as constraints when the initial design commences and safety is costly to reengineer at the latter stages of development. Within the Military environment high degrees of Safety Integrity levels are needed, this increases the level of effort, and therefore cost, during the design and maintenance phases. Using modularity to segregate safety functions can assist in reducing cost and increasing supportability of the software. If, by using modularity the safety functions are segregated, then more Software Support options can be considered as software can be changed without impacting on the Safety Argument.

SUPPORTABILITY FACTORS

If the initial design does not lend itself towards a supportable design, then this impacts on the overall costs and options for the future. The definition of the through-life support solution is carried out through Integrated Logistics Support (ILS), ILS analyses the support options and possible solutions and is expanded upon in Chapter 4. The factors that are beyond the initial design influence that contribute to the partnering and long-term support options are follows:

Intellectual Property Rights (IPR). IPR gives an organisation leverage to secure its intellectual property from exploitation by unscrupulous organisations or individuals. IPR can also be used as leverage over who can legally maintain the software. If a company is unwilling to release IPR to the Military this will prevent the software being maintained by the Military or to another third party, therefore creating a support monopoly.

Change traffic. Change traffic is the defining factor when determining whether to develop a Military support option or whether to return all the software to Industry for enhancements. If there is insufficient change traffic then unless extenuating circumstances exist, such as overriding policy or operational decisions, low levels of traffic will push a support solution toward Industry, and not a partnered solution.

Rigs. Rigs are used for three main purposes, firstly for initials design, secondly for fault investigation and thirdly to test and qualify software changes. Due to the cost of the present integrated avionics rigs these can prevent the optimum support option being chosen due to the short-term cost. The integrated software rig for a platform about to enter service is quoted as being in excess of £45 million to procure and £750 thousand a year to support [IPT05].

Data/Documentation. Initial design data is required to provide a basis for developing a support environment. Without this data; designs, test routines, qualification evidence, assumption and constraints, software would need to be reverse-engineered before support can be effectively conducted. Without appropriate documentation Software Support is problematic at best.

Skills/Experience. Initially whenever a software intensive project is launched there is limited Domain Knowledge with that particular platform, but engineers bring past experience from their past projects. This forms the basis for the new software project and the experience gained is vital to the long-term support over the life cycle of the aircraft. This experience provides Engineers with the ability to understand system requirements and to quickly understand the impact of changes outside their area of direct responsibility. This skill and specific Domain Knowledge underpins the long-term supportability of a platform. As people leave organisations for new challenges, the level of Domain Knowledge changes. This can be problematic if the platform is being modified and vital skills and knowledge are lost, leading to time and cost penalties, where partnerships are developed this can mitigate against the loss of skills..

Finance. One of the often overlooked aspects of Software Support is the financial commitment that is required once the initial procurement phase has passed. It is suggested by Boehm that an organisation needs to set aside at least the same funds for support as is required for the initial procurement [BOE81]. Following on from this recommendation, there is sufficient evidence to suggest that the 50% figure from Boehm was underestimated, to counter this recommendation information is provided below consideration [PIG97], with Figure 6 represents the historical increase in software maintenance cost, this is further expanded in Chapter 5. In the early 1973's 40% of total project cost was consumed on maintenance, by the early 90's this had increased to 90% of total project cost being spent on Software maintenance.

> *"Although there is no actual agreement on the actual costs, sufficient data exists to indicate that maintenance does consume a large portion of overall software lifecycle costs…*
>
> *A research marketing firm, the Gartmore Group, estimated that U.S. corporations alone spend over $30 billion annually on software maintenance, and that in the 1990s, 95% of lifecycle costs would go to maintenance]"* [MOA90]

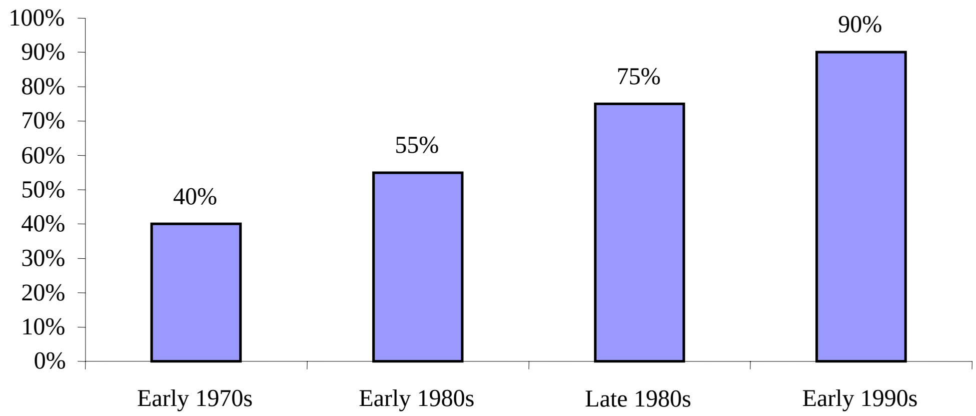

Figure 6 - Increase in Through-life Software Maintenance Cost

All of the supportability factors discussed above influence the Software Development Environment (SDE), the continued ability of this environment to function directly impacts on the long-term sustainability of the software. The SDE must be monitored and adjusted under a quality system to remain efficient, e.g. ISO 9000:2000 and ideally a process improvement method should be used to continually monitor and improve performance, Capability Maturity Model Integrated (CMMI) is the presently recognised method for process improvements and is being introduced across Military and Industry organisations but the implementation is different, which causes problems for organisations when evaluating each others performance or software processes.

PLANNING FOR THE UNEXPECTED

To assist in the forward planning of capability insertion programmes and budgetary forecasting Through Life Capability Management Plans (TLCMP) are produced, this TLCMP is used for near and long-term projections to allow both the IPT and its Industry counterparts to plan their support environments[12]. Due to the unpredictability of conflicts, there will always be occasions when a capability is required by the front-line in short timescales and before it was predicted in the TLCMP. This unpredicted capability need is termed an Urgent Operational Requirement (UOR). It is this level of unpredictability that helped drive the Military to develop its own in-house SST. These SSTs are used in “anger” whenever a conflict requires a UOR and the financial cost or timescales of using OEM are unacceptable to the IPT[13]. UOR are notoriously hard to contract for due to their unpredictability and the capability to cater for a UOR is often removed from Military contracts. During UOR activities there is an increased reliance on close working relationship between the Military and the OEM, with both bringing different levels of experience and Domain Knowledge to solve the software issues. But when a UOR is initiated there is no overarching contract, this is negotiated later in the process as the operational need takes precedence.

[12] Based on personal and SME experience of analysing software projects.

[13] The SST maintains its Domain Knowledge and skills by carrying out small software modifications as authorised by the IPT.

By adapting a partnered approach this would meet the need for an overarching contract and provide the correct level of service. This approach is expanded upon in Chapter 7 which looks at the preferred software solution taking into account Contacts, Domain Knowledge, cost and responsiveness.

SUMMARY

In summary there is a common interpretation of what Software Support is from the SAE and a generic software model is used today by the Military to define the functions that are required to support software. It has been shown that the design and supportability factors are presently understood and that there is an increased reliance on close working and utilisation of Domain Knowledge specifically when satisfying a UOR. This raises the question that if all this information is understood, why is there still a need to change the way software is supported today? The following Chapter will look at the present policies for software and the studies that have been conducted, this will allow for a picture of the present issues to be gained before Chapter 6 looks at why Software Support needs to change from being ah-hoc to a more appropriate approach.

CHAPTER 4 - PRESENT SOFTWARE SUPPORT

INTRODUCTION

Presently the Military support their software using a variety of methods that have been defined over many years, through differences in the political and economic factors experienced during the platforms lifecycle. For this reason there is no common support philosophy that can be defined. This Chapter will therefore present the main options that are being used today across various aircraft platforms, by the use of the model in Figure 5, and the present software policies. The Chapter will conclude with a summary as represented in Figure 7 below.

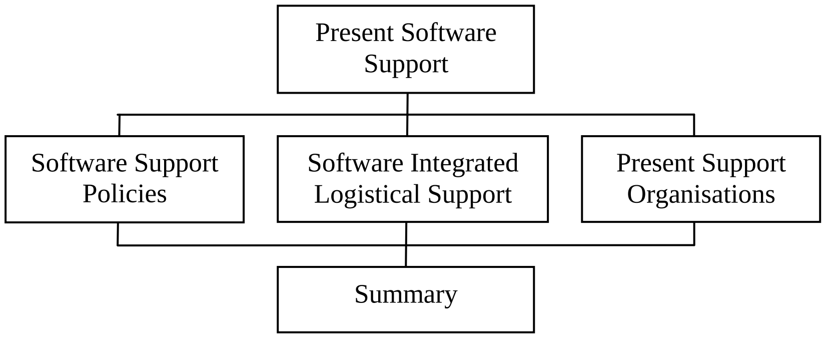

Figure 7 - Chapter 4 Structure

SOFTWARE SUPPORT POLICIES

The policies that relate to the definition of new Military capabilities, or the sustainment of present ones, are reproduced in Annex A. The information below summaries these policies:

MOD Policy Paper No 4 (Defence Acquisition). There are no specific software or hardware requirements, but the paper recognises the need to understand the strengths and weaknesses of Armed Service and Industry organisations, the use of gain-share[14] and incentivisation[15] as well as the need to acquire and support equipment more effectively.

MOD Policy Paper No 5 (Defence Industrial Policy). There are no specific software or hardware requirements, but the paper commits the Armed Services to developing appropriate closer relationships [a partnership] with Industry to equip the Front-line with the services that they require, without creating monopolies. It also recognises that competition remains the bedrock of acquisition and that operational risk will always remain with the Armed Services.

JAP100A-01 (Military Aviation Engineering Policy and Regulation). This policy document covers the support of software in Chapter 12.8, the specific needs are then referred out to AP100D-10 and DEF-STAN 00-60/3.

JAP100D-10 (Support For Mission Software In RAF Systems). This document provides the framework for a Software Support organisation but falls short of recommending a support solution; or addressing key issues on SST composition or location of infrastructure. These decisions should be derived by the correct application of DEF-STAN 00-60.

---

[14] Gain-share – A way for both the customer and supplier to receive a financial reward for reducing WLC. Designed to be used with close engineering and financial relationships.

[15] Incentivisation – A way for a supplier to receive a financial reward for decreasing cost or improving performance.

DEF-STAN 00-60/3 (Integrated Logistics Support – Guidance for Application of Software Support). This policy document is for the application of Integrated Logistics Support (ILS) with the results of Software Support decisions being captured within the Software Support Policy Statement. Unlike hardware which derives its support infrastructure from Mean Time Before Failure (MTBF), software derives its support organisation based on the level of change traffic and operational requirement. But, as the software section of DEF-STAN 00-60 is optional, it is not always carried out with the same level of commitment and understanding as hardware. This then impacts on the identification and realisation of an optimal financial and operational Software Support solution. The software aspects of DEF-STAN 00-60 are subject to much criticism and debate, but it still provides the MoD with its guidance and direction. ILS gained its pedigree through the influencing of hardware and has been adapted to address Software Supportability features; this does not make it ideal for providing the building blocks for the future.

SOFTWARE INTEGRATED LOGISTICS SUPPORT

Presently the MoD has mandated the use of Integrated Logistic Support (ILS) for the procurement of all major systems and upgrades. This ILS approach was authorised by the Chief of Defence Logistics (CDL) in November 1993 and is intended to provide quantative guidance on how individual systems and their component parts should be both procured and supported throughout their life. The overall aim being to reduce Whole Life Costs (WLC). DEF-STAN 00-60 provided guidance on ILS and is now used by all IPT to comply with the CDL mandate. ILS is intended to capture both design and supportability information, to guide the design phase, to influence WLC for both hardware and software. The main factors that contribute to WLC, from a software perspective, are as follows:

- Original design.
- Facilities.
- Levels of predicted change traffic.
- Costs of support environment.
- Tools and processes.
- Skills required.
- Loading methods.

Throughout the Software Support analysis phase there are a number of steps which are required to be systematically undertaken by ILS. These steps put the formality into the activities and document the results to both capture inputs for later stages and for prosperity, ideally the information is reused at a later date. The steps are called Logistic Support Analysis (LSA) Tasks and are identified as follows:

- Task 101 - Development of an Early LSA Strategy.
- Task 102 - LSA Plan.
- Task 103 - Programme and Design Reviews.
- Task 201 - Use Study.
- Task 202 - Software and Support System Standardisation.
- Task 203 - Comparative Analysis.
- Task 204 - Technological Opportunities.
- Task 205 - Supportability Related Design Factors.

- Task 301 - Functional Requirements Identification.
- Task 302 - Support System Alternatives.
- Task 303 - Evaluation of Alternatives and Trade-off Analysis.
- Task 401 - Task Analysis.
- Task 402 - Early Fielding Analysis.
- Task 403 - Post Production Support Analysis.
- Task 501 - Supportability Test, Evaluation and Verification.

The Software Support model recommended by the standard is shown in Figure 8 and forms the initial guidance for an IPT. Although this model has been updated by the RAF Software Specialists, who now recommend the use of the model in Figure 5, it is still contained within DEF-STAN 00-60 and used today for guidance.

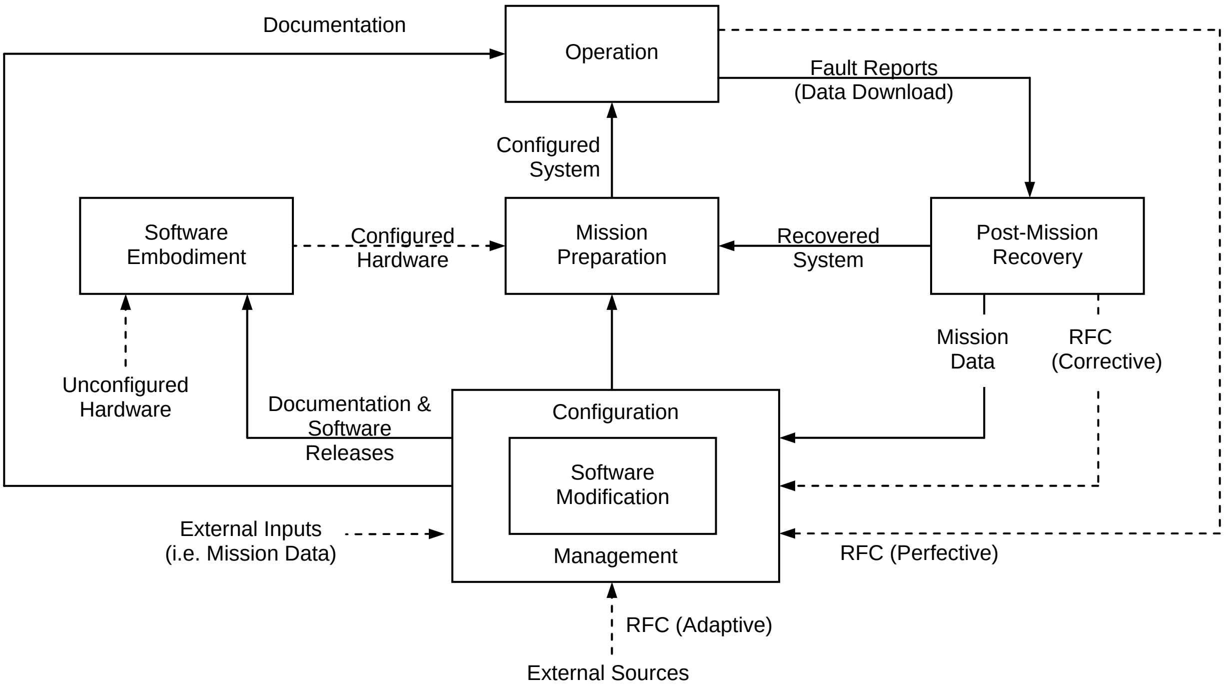

Figure 8 - DEF-STAN 00-60 Software Model

All IPT's should now use the ILS analysis approach to define their Software Support solution. There is however one failing of the systems, in that there are no inputs to ILS for either Operational needs or the need for flexibility in requirements and timescales. For these reasons Contractors are reluctant to agree cost for an unknown number of enhancements, and timescales, for software modifications during the early stages of a contract. Which impacts on the analysis required to develop an appropriate solution.

Figure 9 below represents the amount of software changes on a project that have been experienced by the Military over a 10 year period [DLO02] and is builds on the ISO/IEC definitions in Chapter 3, it can be seen that the majority of modifications fall into the enhancement category. The present Military contracting methods lean towards providing software enhancements on an ad-hoc basis, this contradicts to the aims of forming long-term relationships and partnerships. This ad-hock contracting increases WLC due to unpredictability of the level of enhancements and the constant contractual negotiations for new capability, contracts can take up to 3 years to be agreed [IPT05] with is considered unacceptable from an Operation view point. ILS analysis needs to recognise the amount of expected change traffic when defining a support solution in order to make the solution meaningful.

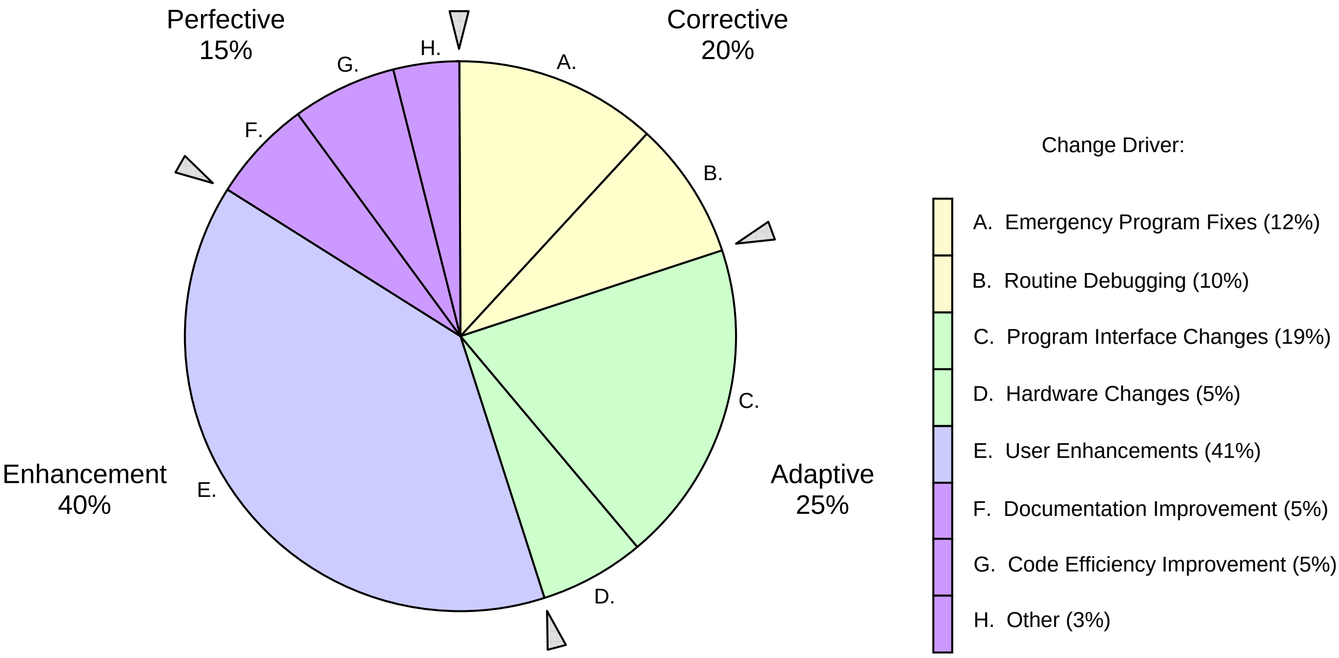

Figure 9 - Causes of Software Change

PRESENT SOFTWARE SUPPORT ORGANISATIONS

There are various permutations of Support Software used today within the Military. The two ends of the spectrum are where an IPT contracts just Industry to support all the software with little technical involvement from the IPT, to the other end of the support spectrum where an IPT tasks their own SST to carryout the enhancements. The list of present Software Support options across the Air Environment is contained in Annex B for completeness. The definition of each level of support that the individual IPT elects for in the majority of cases is based on past experience with an Industry contractor, emotions and financial considerations. This is not just a Military problem but was highlighted in 1990 when Bennett argued that *"based on empirical observation, that the activities undertaken during software evolution vary greatly"* [BEN99] it was seen that there was no single approach to software maintenance with some organisations stopping support post-delivery. The reason for this diversity of software solutions is that the majority of the decisions were made prior to ILS becoming mandatory during November 1993. The characteristics of the two ends of the Software Support spectrum, along with the middle ground of a Software Support Cell (SSC), are represented in the following text and Figurers 10-12 below, with these figurers being based on the model in Figure 5. These three variations; Industry, SST and SSC form the basis for Software Support today and are the main options considered by IPTs.

Industrial Support Option (OEM). For platforms such as the C130J that has entered Military service relatively recently the software updates are contracted for on an ad-hoc basis by the IPT. It can be seen in the Figure 10 that the Military retains its dominance over the Operational environment, which is to be expected as operations activities are carried out by the end-users. The only other involvement that the Military have is within the Management functions and decision making process. The Management Function dictates the Requirements, timescales and the acceptance criteria that the new capability will be judged against. The advantage of this approach is that the IPT can place a contract with Industry and leave the implementation and technical aspects to them, but these contracts are infrequent so Industry can loose some of its Domain Knowledge. This is perceived as transferring the development risk away from the IPT towards Industry, but the Military will always retain the Operational risk to their personnel and capability and there by default some development ownership.

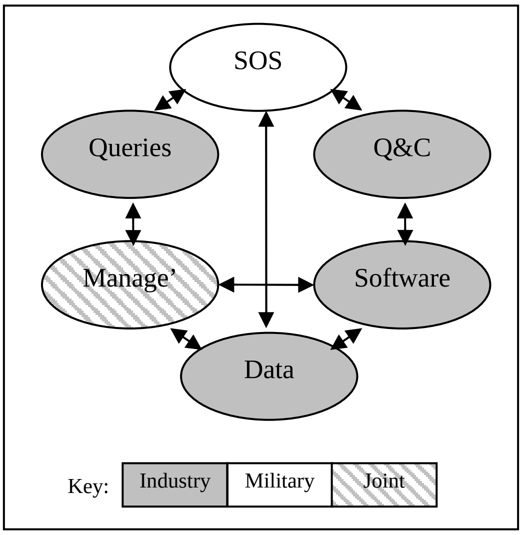

Figure 10 - Industry Software Functions

Software Support Cell. For platforms such as the Merlin III Helicopter which has also entered Military service relative recently, major software updates are still contracted to Industry on an ad-hoc basis. It can be seen in Figure 11 that the SSC option is similar to the Industry support Option outlined above but, located within Industry there is a small cell of Military people that carry out Initial requirements analyses, problem evaluation and the acceptance testing of the software. The IPT benefits, and advantage to both organisations, comes from the analysis of problems and requirements being carried out early, often before formal contract is let. This allows for the resolution of misunderstandings before they become a major issue. Presently these teams are used not just addressing Military IPT generated requirements but they also addressing problems that the contractor is experiencing internally. This has the spin off that Domain Knowledge is maintained and as the Military contribute to an internal modification, when this modification released to a wider market the IPT will be provided the enhancement at a reduced cost. This approach has an advantage over a full Industry solution as both parties gain from the relationship and that incorrect contract requirements can be addressed at an early stage, therefore reducing WLC. SSC are the first form of partnership that proved to be a success.

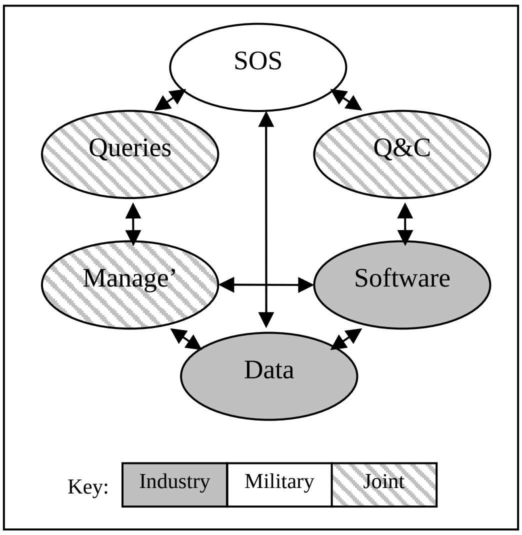

Figure 11 - SSC Software Functions

Software Support Team. For platforms such as the Harrier GR7 and Tornado GR4 the Military have their own software organisations, an SST, these are used exclusively for their aircraft type and they implement changes authorised by the IPT. It can be seen in Figure 12 that the SST carryout the full range of software activities from initial requirements analysis though to the qualification and release. The advantage of this approach is that no commercial contracts exist between the IPT and SST, they use a Formal Tasking Agreement (FTA). The route to forming a FTA is quicker than the Industrial contracting route, this decreases the time for a capability to reach the end-users when compared to the ad-hoc formal contracts with Industry. It would be expected that all the functions within Figure 11 would be allocated to the SST and not Industry, but this is not the case due to the approach that the IPT's take to risk management. It's a directive from the Secretary of State for Defence that all software from a SST needs to gain a Design Authority release certificate[16] for all software modifications. For this reason, Industry reincorporates all the software changes that have already been introduced by the SST, these are released to service during the next major Industry update. This duplication increases Software support costs by increasing the number of facilities, SDE and supporting infrastructure. This duplication is a major contributor to WLC, adding little value to the end product.

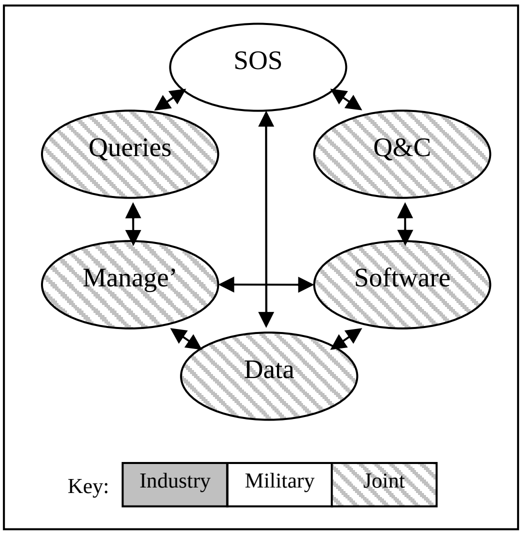

Figure 12 - SST Software Functions

There are advantages and disadvantages with all the three options above, with the three options for Software support being balanced by the IPT against financial constraints. A direct comparison of the Functions that different options carry out is represented against the "V-Model" [MCD94] in Figure 13 below. It can be seen that all three Support options are involved in Requirements Capture and Acceptance Testing, where the greatest influence lies, and that the SSC are also involved in developing specifications. Also the IPT SST carries out all the activities of the software life cycle from initial Requirements through to generating code and acceptance testing, but as stated above there is duplication and increased cost by using an SST when Industry has to implement duplicate changes. These changes are normally updated during major updates and due to there infrequency Industry has to reacquire its Domain Knowledge as personnel have migrated to new projects as workloads and contracts change.

[16] Design Authority release certificates are only released by Industry.

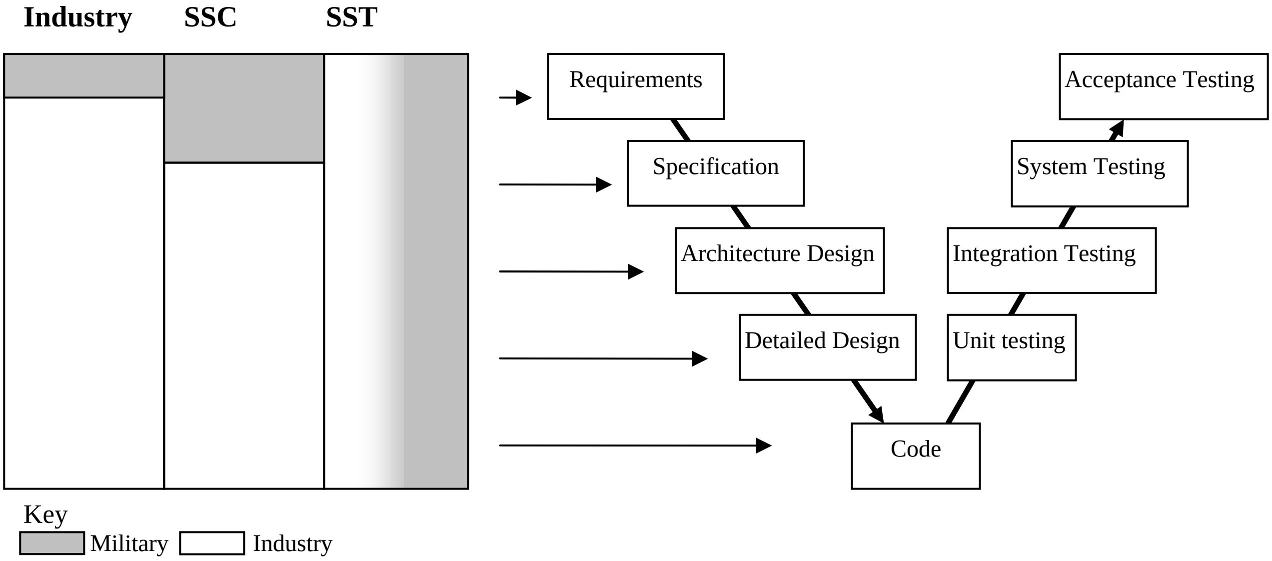

Figure 13 - Comparison of Support Options to "V" Lifecycle

There are good economic reasons to allow a single organisation to be constantly subjected to a flow of new requirements, as this prevents them loosing skills and Domain Knowledge. It is documented that 30% of software features change between major iterations [CUS97], if knowledge is allowed to mature through partnering instead of dwindling until a major update is required this would reduces the learning process, therefore reducing time and cost then major updates were inevitably required. Passing minor updates through a partnered organisation saves both time and money.

SUMMARY

It can be summarised that there are policies in place for the development and support of software that facilitate through-life change and meet the customers' needs through ILS. As Software Support is not sufficiently addressed during the initial contractual stages due to its complexity, and the lack of understanding within the general DLO community, it is often carried out incorrectly or too late in the development process. This has lead to variations in methods for supporting software. This support ranges from a full Industry team through unpredictable and ad-hoc contracting to a full Military team that is duplicated by Industry during any major upgrade, significantly increasing cost. From the options presented the initial assessment is that the imbedded SSC provides a balance of influence and cost but this is also not ideal due to its ad-hoc contracting methods for updates, but it does have influence during the requirements stages to reduce cost due to ambiguity or misinterpretation. The next Chapter will look at the immerging need for change as software design activities change and the external drivers that are forcing IPT and Industry to look for alternatives to present Software Support contracting and relationships.

CHAPTER 5 - THE WIND OF CHANGE

INTRODUCTION.

The aim of this Chapter is to present the main drivers that have immerged over the past 10 years, which are encouraging both the Military and their industrial counterparts to consider a change from the present methods of supporting and contracting for software enhancements towards a joint approach, termed "Partnering". The main trends in software design and business issues have been recognised and are forcing this change along with changes in software technology and technologies, this chapter will consider these issues and conclude with a summary, as represented in Figure 14 below.

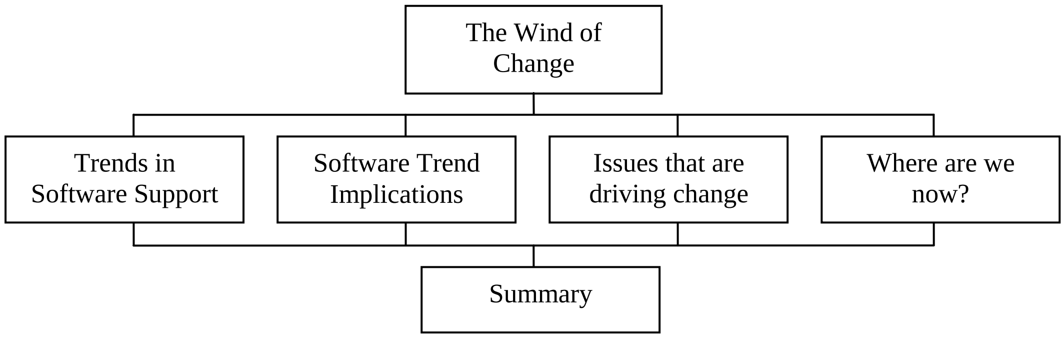

Figure 14 - Chapter 5 Structure

TRENDS IN SOFTWARE SUPPORT

If we look back over the recent history of aircraft software development, there have been dramatic changes in capabilities that are enabled through software; the comparison would be the changes that the IT Industry has experienced over the same time period has moved from standalone computers to integrated company networks that can communicate seamlessly across the world. By referring to Figure 15, we can see that the numbers of aircraft has dramatically decreased and the levels of integration have increased as technology has matured. Now new capabilities are achieved by enhancing present platforms and by not replacing aircraft, which has become the preferred option, based on cost. When aircraft such as the Nimrod MR2 entered service in 1972 one person could understand all the code and a small team could maintain the software, as it was contained within isolated systems or single functions. With aircraft such as the Nimrod MRA4, due to enter service in 2009, the levels of integration and battle space interoperability require an increased team size[17], its operational capabilities are now imbedded in technology and Domain Knowledge, rather than the number of aircraft you can put in the air.

[17] The permanent Software Support Organisation for Nimrod MRA4 is 53 people supported by sub-contractors.

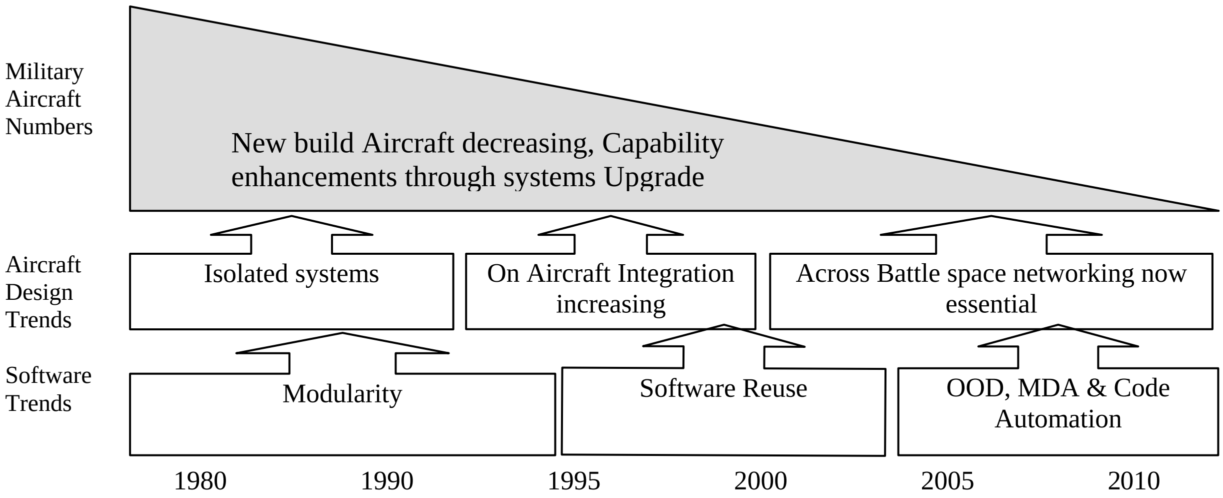

Figure 15 - Change in Trends

The future of software development and support is presently uncertain, but there are judgements that can be made based on the trends within both the Military and Civilian software communities. Over the past 10 years tools and techniques within the Military environment have stagnated whereas the Civilian community has morphed beyond all expectations. The Military software development has been stifled through a lack of investment and a limited amount of companies having an interest in changing their present approach, but changes are starting to filter through.

Technique Trends. Unified Modelling Language (UML) has become the preferred method for the development of new platforms, with UML taking the place of procedural structural design and development techniques. UML is now the preferred method of developing software, but there are still concerns for Safety Critical applications where the actions of software need to be fully predictable and verifiable. This need is understood by the software community and presently achieved through the use of Formal Methods and structural programming. It is considered that practices within the Automotive Industry that have moved away from structural programming, will migrate to the Aerospace community in due course, and UML will become universally accepted for all software.

Languages Trends. The Aerospace Industry is seeing the sun-setting of ADA, including SPARK, with C and C++ quickly becoming the languages of choice but there are still concerns with language selection. In 1986 Bjarne Stroustrup succinctly captured some of the concerns and is credited as capturing the difference between C and C++ as follows *"C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do it blows your whole leg off"* , the analogy is accepted which is maybe why Safety Critical systems designers are presently reluctant to use C++ and are moving from ADA to C#, therefore bypassing C and C++. It has become apparent with the increased use of Power PCs in larger aircraft, that manufacturers are pushing to use JAVA for systems that are non-safety critical or rely on human interpretation. Examples would be; Airborne Command and Control centres that rely on Data Fusion techniques to form a cohesive battle space picture, or systems that are reliant on real-time human interaction to make then operationally effective.

Future use of Auto-code Generation. Presently Auto-Code generation is frowned upon for the development of Safety Critical software or for displaying information that is presented to platform operators. As organisations work closer together [partner], they share their information, experience and expertise, this reluctance to use auto generated code is now decreasing. As this approach is being relied upon more and more within the commercial sector and it's gaining credibility, therefore studies are being conducted to challenge this reluctance to use Auto-generated code and forcing the Military Aerospace Industry to form a view based on facts, rather than emotion.

Bigger Picture. There are more and more platforms that can not fulfil their full capability without communication with a wider "Conciseness". This communication is already needed during the initial preparation of an aircraft that requires the latest Intelligence information, Electronic Countermeasures and the location of friendly forces and targets. During flight information is being updated in real-time therefore increasing the probability of survival, mission success or indeed providing the option to terminate a mission. This updating is now a bidirectional process where information is sent to the aircraft and onboard sensors can transmit information over a network to update commanders. This ability to exchange information is termed Networked Enabled Capability (NEC) and is allowing battle commanders, ships and Unmanned Air Vehicles (UAV) to communicate in real-time. With this ability to create this big picture also comes a big problem. If one person can not contain the knowledge of a single aircraft, what chance is there for fully understanding a capability that spans the Land, Sea and Air environments as well as different countries? The answer is you can't. But by the use of a partnered approach that draws experience from Industry, Academia and the Military, information from different areas can be exploited if the information is captured at the correct level on indenture, therefore it must be designed at the correct level. One approach is to use a Model Driven Architecture (MDA) that that can be used to develop a system independently of the platform presenting information at the correct level, this approach helps to specify the interoperability between systems with a variation been adopted by the Military in 2006.

We are now seeing a change in design methods and languages but this change is biased towards new platforms or major upgrades. But there are still many legacy systems that use older languages and structural programming, but these will reduce as platforms and systems are replaced or become too expensive to support. The change to UML is reducing diversity and increasing commonality across software boundaries, therefore assisting organisations that intend to form long-term relationships, partnerships, so reducing cost

SOFTWARE TREND IMPLICATIONS

The term "But I only changed one line of code" being said light-heartedly in the past now takes on a whole new meaning as the majority of functionality and capability are changed through software. With the increased reliance on interoperability, enabled by highly integrated software, now even a medium sized team cannot contain all the software just within their heads. This move towards increased interoperability was partly driven by the customers' expectations of what systems could achieve. The situation now is that an OFP in a single aircraft can change the characteristics and decisions of the whole battle space, not just a single aircraft as its information now contributes to the "bigger picture".

The early days of having a team that just "cut-code" have now disappeared and been replaced by highly trained Teams containing many people; with the requirements for a formal paper trail and the number of interrelated documentation increasing with each new contract. For the teams to understand their customers' needs and technical requirements there has been a change from pure Software Engineering towards Systems Engineering. This brings a need for appropriately skilled engineers that can understand the Domain that the aircraft works in, not just their piece of code and the skill to identify interoperability issues at an early stage to make sure that projects stay on track. Neither the Military nor Industry has sufficient Domain Knowledge to address the issues alone. There is a realisation that closer relationships are not just Nice-to-have, but essential for future software enhancements to be realised effectively.

Figure 16 below is a representation from the Future Offensive Aircraft System (FOAS) report that looks at the levels of functionality that will be enabled by the use of software in the future. It can be seen that in the 1970s hardware was the dominant factor that enabled 80% of the full platform capability, but over time software has increasingly become the major enabler and increased from 20% to 90%. This change means that both the Military and Industry have needed to update their skill sets to keep pace with changes in technology and the move from hardware enabled capability towards software. It can be surmised that as the main capabilities are now in software, software has become the largest cost magnet and is therefore being put under increasing levels of financial pressure to reduce WLC [DLO02].

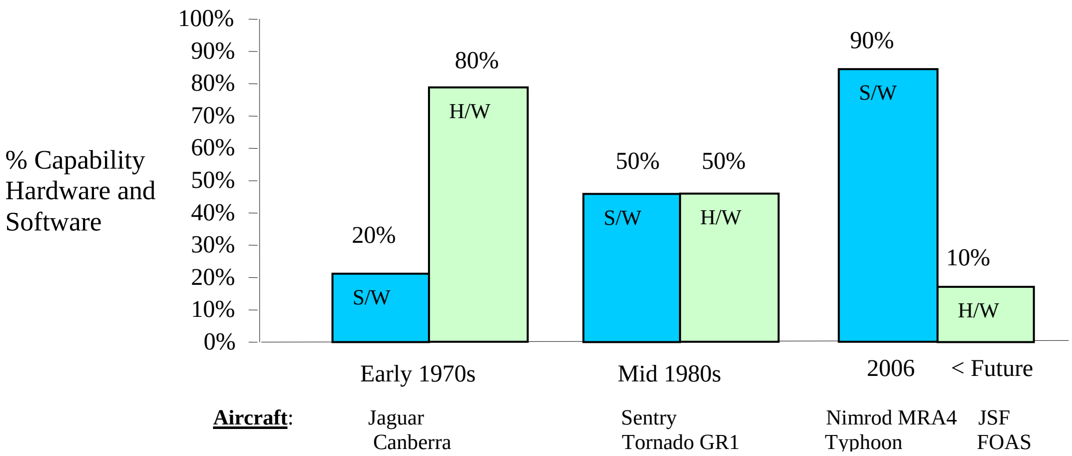

Figure 16 - Software enabled functions over time

ISSUES THAT ARE DRIVING CHANGE

There are issues that have been identified over recent months which need to be considered and risk mitigated measures taken before any major change in direction for supporting software take place. These issues have been touched upon previously in studies carried out by the Stakeholders, these studies are assessed in Chapter 6. These issues, either perceived or tangible, have immerged through difficulty in meeting customer or commercial requirements [DLO02]. The main issues are categorised as follows:

Quality. For systems to meet their aims there needs to be an appropriate quality system in place which if implemented correctly increased software reliability. It has been found that quality systems are seen as a tag-on by some organisations and are seen as blockers to meeting requirements. Quality Management Systems (QMS), when introduced correctly enhance the product that software teams develop and both its maintainability and reliability; with the introduction of ISO 9000 there has been an increase in documentation and development time. This is sometimes considered a negative result of a QMS. With the recent move towards CMMI the levels of quality activities have now stabilised, but questions remain about the level of legacy bureaucracy and documentation should be reduced by increased process efficiency by using best practice from either Industry or the Military.

> *"Software Maintenance [and support] are a collaborative, cultural activity that has shown major benefits, certainly in terms of software reliability and performance"* [BE/RA02]

Flexibility. There is an inherent inflexibility built into contracting and the assumption is that the contract will be correct first time, every time. This assumption is incorrect and supported by the NAO reports into major projects [NAO03]. This inflexibility causes resentment when requirements are clarified after the contracts are agreed, or there is a disagreement on the interpretation. There is then a risk that this circle of inflexibility, from both parties, repeats itself whenever enhancements are required as a level of mistrust has been established. The present software contracting methods are ad-hoc and considered to be inflexible; by their very nature they do not promote the correct attitudes for partnering. If software enhancements could reduce or remove the lengthy and inflexible contracting activities this could reduce capability realisation times and WLC.

Responsiveness. There is a lack of responsive in the way the Military and Industry communicate and interact with software queries and information exchange. This sluggishness is causes by the lack of standing agreements for information exchange and the use of contracting methods that do not add value. This ridged contracting routes causes frustration for all parties and adds little value to software development and support, an alternative approach to this ridged contracting is represented at Figure 20 (page 43) and discussed in Chapter 7.

Credibility. For any long-term relationship to exist there must be mutual trust and understanding. There is an increasing feeling that as new and exciting systems come along engineers will migrate from languages and techniques that are becoming obsolete, and therefore less marketable, towards the next generations. Some companies are renowned within the Aerospace Industry for not meeting software targets or promising more than they can realistically deliver, which impacts on their credibility and reduces Customer confidence in their abilities. But the Military is far from blameless in this relationship, the NAO have constantly highlighted where the Military have either contracted too early or changed their requirements dramatically soon after contract award. The credibility of both organisations needs to be re-established to allow a long-term relationship to develop.

Business needs. When the Cold War was still an issue for the world, there was a reliance on new technologies and the latest gadgets, with the Military in the business of acquiring them and Industry supplying them. There has been a change in perspective over the past 8 years after the Strategic Defence Review [SDR99] that reduced the manpower and funding for all arms of the Military and changed the emphasis from capability enhancements though new build towards value for money, reuse and major upgrades. For these reasons both Industry and the Military could not afford to continue with business as usual, but needed to take another more appropriate approach. The Cost of ownership[18] and Whole Life Cost has now become the main driver for Integrated Project Teams.

Resources. The Military enhancement and upgrade business by its very nature is predictably unpredictable. It is recognised both by Industry and the IPT, that there will be enhancements and when a new platform enters service the next upgrades are already being considered, but the actual requirements and available funding are the uncertainties. For this reason Industry cannot afford to have, for example, a team of 30 software and systems engineers sitting for 6-months waiting for the next contract, and quite rightly the Military should not pay for a Standing Team to do nothing for 6-months. Here lies a dilemma. For a Software Team to remain proficient they must be conversant with the relevant Software Domain. All aircraft types are different and there are types within types too, with different languages, different architectures, different hardware and varying levels of documentation. The challenge is balancing value for money against the loss of Domain Knowledge and the need to maintain a credible software modification capability.

Political and Economic factors**.** There was a perception within Industry that the Military could afford any software enhancement or solution that Industry defined. This was generally the case at the height of the Cold War, but there's been a shift toward value for money, or "Bang-for-buck" over recent years. Now the Military is under increased scrutiny concerning its procurement costs with the policy emphasis changing to value for money. This political change in emphasis emerged when Smart Procurement was launched during the early 90's, but the IPT have been increasingly criticised by the NAO for being noncompliant with Smart Procurement requirements and for not providing value-for-money [NAO03]. This is mainly because IPT do not understand Software Support requirements and will not accept that software is destining to change.

BREAKDOWN OF PLATFORM ENHANCEMENT COSTS

Cost has always been, and will continue to be, a major driver for change. To illustrate the amount of financial commitment that software support requires, historical information has been gathered from within the DLO. This relates to a recent major platform enhancement. The enhancement under consideration is typical of the type of modifications carried out throughout a platforms life; it necessitated replacing hardware and either modifying or developing software. This particular enhancement consisted of replacing the primary sensors, the main processing computer that analyse sensor data and then either provides the information to aircrew displays or is transmitted externally to support Network Enabled Capability. The cost information for this enhancement is presented in Figure 17.

It can be seen from Figure 17 that 71.7% of the cost was consumed by Software Development, 14.8% by Hardware Development and Production, with the remaining 13.5% on Platform Integration and Production, or the supporting Assurance and Project Management activities. This confirms the fact that software has now become the largest cost driver for platform sustainment and support.

[18] Cost of Ownership increases as platforms mature and mandatory regulations change, e.g. Health and safety.

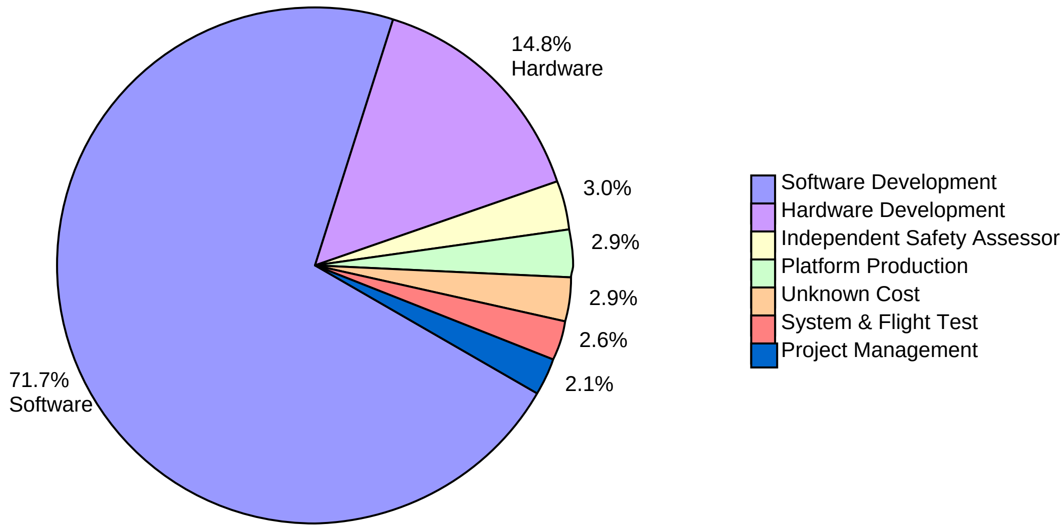

Figure 17 - Breakdown of Platform enhancement costs

By using the 4:1 Rule-of-thumb[19], it can be estimated that Nimrod MRA4 with its initial procurement cost of £2.4bn[20] will spend an estimated £9.6bn on through-life support and sustainment. If we accept the 71.7% from Figure 17, this means that £6.9bn will be spent on software support or sustainment. Therefore even a 1% increase in efficiency through partnering, will mean a £69m saving over the typical 25 year in-service life. This information is captured in Table 1 for consideration.

Activity	Cost
Nimrod MRA4 Initial procurement cost	£2.4bn
Expected Through-life support and sustainment cost	£9.6bn
Expected Through-life Software cost	£6.9bn
1% software through-life saving	£69m
Software saving per year	£2.8m

Table 1 - Representative Software Through-life cost

The information in Figure 17 and Table 1 are considered representative of through-life costs, with this information it can be understood why software development and support is coming under increasing levels of scrutiny. If partnering can increase efficiency by removing duplication, improving communications and software development techniques, these improvements in WLC terms will make a considerable financial difference.

WHERE ARE WE NOW?

For the same change drivers identified above, the Military had its own software revolution in the 1980's with the introduction of the Tornado and Harrier aircraft. They developed their own SST that could be trained and used for both minor and major enhancements that did not have the same level of constraints as their Industrial counterparts. The contractual constraints were removed and software could be developed and flown successfully. There was however one issue, the

[19] For every £1 spent during initial development, £4 will be spent on through-life support and sustainment.
[20] Nimrod MRA4 Initial Contract cost (1996).

issue of formal flight clearance that is required and that the Military decision makers were still not prepared to allow their SST to provide the Design Authority clearance. Therefore it was inevitable that whenever a major upgrade was introduced that was outside the capability or capacity of the in-house SST, it would be contracted back to Industry and all the changes that the SST had incorporated would also be added to the upgrade baseline. This increased the financial levels but the SST provided an acceptable level of flexibility and responsiveness, the SST were seen as being agile by the IPT with Industry being used to mitigate against the risks of software errors being introduced by the SST. The disadvantage of this approach to developing software is the duplication of processes, organisations and infrastructure, but there are advantages provided by both organisations, which are discussed in Chapter 7.

Recently with increasing levels of financial constraints and the treasury announcing a 20% reduction in support spending, the Software Support stakeholders held meetings to capture the key issues that needed to be addressed; the main issues are represented in Table 2 below. The major surprise was that the Stakeholders issues were very similar and that it was in everybody's interest to find a mutually agreeable solution.

Industry Issues	**Military Issues**
• Contracts too slow and inflexible • Unable to accept SST software 'as-is' for approval • Want to make money • Military is inflexible • Requirements change late in development or are ambiguous • Lack of credibility • Lack of front-line Domain Knowledge • Mistrust of the Military	• Contracts too slow and inflexible • Industry can't accept the SST software 'as-is' for approval • Want to save money • Industry is inflexible • Industry software never flys' as its 'out-of-date' • Requirements change late in development or are ambiguous • Lack of in-depth systems knowledge • Mistrust of Industry

Table 2 - Common Issues that are driving change

SUMMARY

In summary there has been a marked change in software technology over the past 20 years and the levels of Domain Knowledge that is required to enhance future platforms is spread across both the Military and Industry organisations. There has been an increase in the levels of justification required for introducing capabilities and a major drive to reduce cost, not just initial procurement costs but through life support costs. And that contracting methods do not promote the attitudes of partnering. With the majority of the issues being either; commercial, financial or knowledge based, not technical development, the partnering approach will help reduce barriers and overcome these human issues. As the partnership develops the group will work closer together, as well as understanding each others problems, they will also help to solve problems to, for the good of the long-term relationship. In the next Chapter we will consider formal studies that have been carried out and the requirements that need to be addresses to form a solution to the issues captured during this Chapter.

CHAPTER 6 - FUTURE MILITARY SOFTWARE REQUIREMENTS

INTRODUCTION

The most important phase on any software development is capturing the requirements. This Chapter will meet that need, drawing from various studies and initiatives that have been conducted over the past 7 years within both the Military and Industry environments. After this activity was completed, the distilled requirements were presented back to the IPT for confirmation and clarification. This has allowed the dissertation to assess if a Partnered solution could meet the stated or implied needs of the Military and Industry organisations. The key activities from this Chapter are represented in Figure 18 below for consideration.

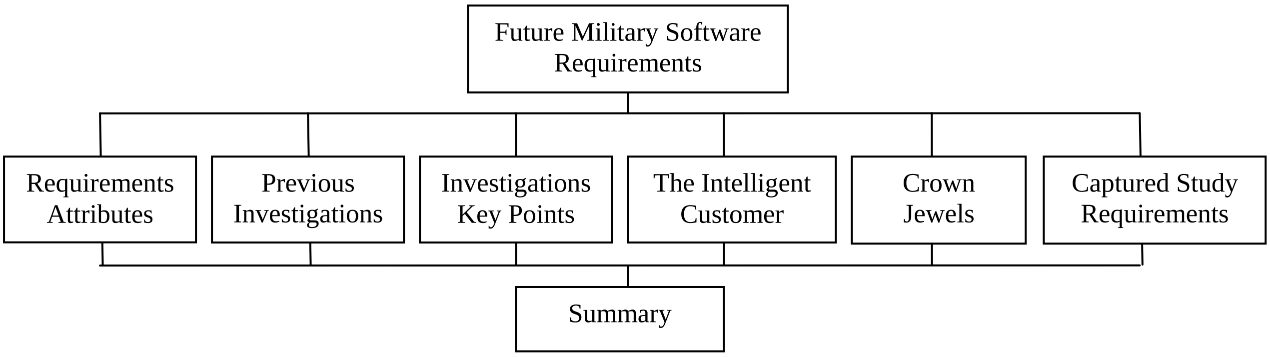

Figure 18 - Chapter 6 Structure

REQUIREMENTS ATTRIBUTES

For requirements to be managed and progressed, they need to have appropriate attributes. They should be "*consistent, complete and accurate*" [M880-02]. From requirements analysis documentation these attributes are further defined as follows:

Traceable. Traceability is required to allow assessments to be made on how requirements were derived and what the impact of change might be. Traceability also assists with trade-off analysis and contract compliance assessments.

Unique. All requirements should be uniquely identified to allow traceability to be achieved.

Unambiguous. Requirements that are ambiguous often cause debate and misunderstanding, if this is carried through to the design process the cost of change is increased when misunderstandings are highlighted. The longer the ambiguity exists the more it costs to correct; therefore ambiguity should be removed.

Testable. During different stages compliance against requirements needs to be assessed, either for commercial or technical reasons. For these reasons the method of gauging compliance and testing needs to be established. The definition of testing is preferably carried out during the initial requirements capture phase.

Types of Requirements. There are two categories of requirements, Functional and Non-functional, these are defined as follows:

- Functional Requirements are requirements that specify the functions that are to be achieved by the software; an example would be calculations or outputs.

- Non-Functional Requirements support the design process or impose constraints, examples would be quality standards, reliability of a system or security levels that are required to be achieved. Non-Functional requirements are harder to measure than functional requirements; therefore the methods of assessing compliance can be problematic to define.

PREVIOUS INVESTIGATIONS

There have been six main investigations carried out over the past seven years that are in general circulation within the Military software community. The report summaries are as follows, with their recommendations being reproduced within Annex C of this document:

“A Study into Software Teams” [DLO97]. This study was intended to assess the functionality and resources of the SST and concluded that:

- “There is, and will continue to be, a need for in-Service SSTs to address the future organisational needs of the Armed Services”.

- The DLO should form better relationships with Industry.

- The report also makes recommendations on the location of rigs, the support infrastructure and the need for ‘blue-suit’[21] employment within an SST.

“Logistics Programme Management Committee (LPMC) Paper: The Support of Software in Future RAF Air Systems” [DLO98]. This study was intended to address some of the points from “A Study into Software Teams” [DLO97] above, such as the post-Cold War Software Support requirements and major business areas affecting logistics support. It concluded that support should be assessed on a project-by-project basis with service/contractor teaming being the common basis for aircraft SSTs of the future.

DLO MCS Review – In-Service Aviation Design Support (ISADS) [DLO01]. This study was intended to establish the full extent of in-Service design within the Air Environment, and to assess the effectiveness of the aviation support organisations serving IPTs and Front Line Commands (FLC). It concluded that the in-Service SSTs provide value for money to the IPT, as they are cheaper than Industry and address the operational needs of the front-line customer. However, due to Strike Command cost savings, SSTs have had a net reduction in their capability to change software, which has in turn had a negative operational impact.

ES(Air) Software Support Strategy [DLO03]. The results were intended to provide guidance in the formulation of an ES(Air) software policy. The report raises many questions and issues to be resolved but does not provide a method to derive a solution from those questions. It recognised that software planning is ad-hoc, Industry is more expensive than an in-Service SST and that software must be updated in operationally significant time-scales. It concluded that a joint Industry and blue-suit solution would be the preferred option and provide an “intelligent customer” capability.

[21] Blue-Suits are; Uniformed Army, Royal Air Force or Royal Navy personnel.

Streamlining End to End Air and Land Logistics [DLO04]. This study was intended to deliver more cost effective logistic support to the Armed Services. It concluded that the future logistics support strategy should be based around the effects that the end-users need to achieve.

That "*Expeditionary operations are now the principle role of the UK Armed Services which therefore demands flexibility and reactiveness in its support infrastructure*".

It also recognised that benefits can only be achieved with considerable short-term investment to "*concentrate resources where they can deliver the required effects as affectively, flexibly and efficiently as possible*." [DLO04]

The reports summarised above raise points that any change to present Software Support methods should consider. The points of commonality from the studies have been distilled and captured in Annex D for consideration.

INVESTIGATIONS KEY POINTS

In the preceding discussions and analysis of the studies in Annex C, the requirements have been presented back to the IPTs. It has been confirmed that there are presently no activities that have taken the studies recommendations forward to a lower-level or proposed solutions to the problems identified within the studies. The main themes that immerged from the studies are summarised as follows:

The need for change. There are common themes within all the studies which these are represented within Annex D. These relate to the effectiveness and the definition of the most appropriate financial and operational Software Support solution to meet the DLO, End-User and Industry needs.

Reduced cost. If policy decisions based on the reviewed documents were implemented, this should reduce the up-front Lifecycle Costs and allow both Industry and the DLO to contract sooner for appropriate long-term Software Support.

Blue Suits. There is a need for planned and organised software capability within the armed services that actively involve blue suits and dictates the level of infrastructure and rigs required.

Baselines. With different organisations releasing software for the same aircraft, this causes problems because of the various baselines and functionality that is released.

Responsiveness. There is an ongoing need for any software solution to be released in timescales that impact on operational situations. Any solution must have a credible capability that can release UOR software as well as main software updates.

Optimise Domain Knowledge. Industry has a level of knowledge that is complimentary to that of the Military organisations. Industry has in-depth systems Domain Knowledge and the Military has the operational Domain Knowledge, any solution must exploit the positive attribute that both organisations have and reduce any shortfalls in knowledge.

Streamline interfaces. Whilst the software being released has to be safe, the certification activities are too slow and do not compliment the required level of response for a UOR.

> Therefore streamlining the dependencies and interfaces between organisations should have a positive effect, therefore streamlining the overall software development and support activities by establishing appropriate processes and interfaces.

The main observations from analysing these reports and discussions with current IPTs is that there has been a large amount of effort expended, both within Industry and the DLO, relating to the requirements for Software Support, blue-suit involvement, rig and infrastructure locations. It is evident from the studies analysed that there has been a lot of research carried out by the DLO and Strike Command, which has made specific recommendations relating to Software Support issues. These studies recommendations could form the basis of efficient policy, which would reduce duplication, save money, save time and produce an appropriate support solution for Front Line Commands (FLC). It is also noted that the recommendations from the LCMP and DLO Strategy reports are not as specific as the original analysis within Hyde and ISADS reports; therefore they add little value in developing an appropriate Software Support solution or policy and have actually watered down the recommendations allowing for more ambiguity.

THE INTELLIGENT CUSTOMER

With reference to studies carried out within the automotive Industry [TWG97] there is a case to be made for having a Guest Engineer within organisations, not independent Contractors but for a prime contractor to place one of their own experienced engineers into a subcontractor to provide guidance, clarify requirements and answer queries. Twigg defines the Guest Engineer as:

> *"A Guest Engineer is a technical specialist, usually employed by the supplier of the technology or design expertise, who is resident in the customers' organisation. It provides tacit knowledge throughout the product development process, from upstream activities (such as pre-concept, concept and design) to engineering validation and quality proving…its key role is to facilitate the effective integration of supplier' technology expertise with the need of the customer."* [TWG97]

The advantages of this approach are that when even a minor issue needs resolving there is a suitable representative who has the relevant experience, Domain Knowledge, contacts and empowerment to resolve the issue before it becomes a major problem for either party. Twigg identifies that the Rover motor company have being following this concept for many years and have provided their expertise to sub-contractors as a de-risking activity to development. This is mirrored within the Military community with personnel being positioned early within contracts to reduce long-term costs that could occur through ambiguous requirements or implied needs not being specifically stated or misunderstood. These people provide an "Intelligent Customer" input to projects and continuity, there is no firm documented requirement to maintain the Intelligent Customer status, this was implied from the studies in Annex C and stated by the IPTs [IPT05]. The challenge is how to maintain the level of Domain Knowledge required to be an Intelligent Customer. In a partnered environment both organisations will have the opportunity to draw from this knowledge for the benefit of the partnership. So any solution should address the Intelligent Customer need.

CROWN JEWELS

Within all organisations there are core capabilities that distinguish it from others, but over time these can becomes undistinguishable from capabilities that add little or no value to the business. When funds are reduced, as identified in Chapter 5, or business opportunities change, organisations often take a long hard look at themselves and reorganise their organisation. This can lead to the streamlining of organisations either through outsourcing or combining departments and functions. Whenever organisations consider change, there needs to be an assessment carried out on the levels of Domain Knowledge, skills or capability that should be retained within the individual organisation. The resultant output is termed the organisations Crown Jewels, the business critical functions that need retaining to maintain their market edge or core capabilities.

It can be argued that the only Crown Jewels for the Military are the front line aircraft and the pilots that control these aircraft, as these two parts form the end product that prosecutes a target. All of the supporting activities that initiate the planning or flight are therefore up for consideration to be outsourced or partnered. The only factors that have been voiced by the IPTs that prevent this level of partnering are as follows:

> Credibility. Industry has a lack of credibility in the Military's eyes - based on historical performance, as discussed in Chapter 5.
>
> Public Sector Comparator. The need to retain a credible public sector comparator for assessing costs – A government requirements carried out by the NAO.
>
> Intelligent Customer. The need to retain the "Intelligent Customer" position to be able to assess contracts and capabilities proposed by Industrial counterparts.

To date there have been no policies identified that defined the Military Crown Jewels, therefore it's possible to partner all Software Support activities, or indeed selling off its assets. This is seen as a failing of present policies. If the Military knowledge was eroded, there would be no credible "Intelligent Customer" and therefore no credible evaluation or comparison could be made against either new contracts or sustainment capabilities. This could create a monopoly, which is directly against the present intent of healthy competition and the End-to-End study [DLO04]. However the cost of maintaining a full Military SST capability needs to be balanced against the requirements of the IPT.

CAPTURED STUDY REQUIREMENTS

The specific requirements that have been distilled from the reviewed studies have been captured in the Table 3 with Chapter 8 assessing how a partnered solution meets the stated requirements. It can be seen that none of the requirements relate to the design of software, they are all non-functional. This indicates that the studies found no major issues with the development of software, its safety or quality. The software issues lie within the areas of relationships and the activities that support the development and support of Software, the human issues.

ID	Requirements	Origin	Functional / Non Functional	Test Method
Req-1	Software should be managed by the Military with the assistance of Industry	Hyde	Non Functional	Document Inspection
Req-2	Sustaining contracts should be used for Software	Hyde	Non Functional	Document Inspection
Req-3	Domain knowledge should be maintained across software organisations	Hyde ISADS	Non Functional	Document Inspection
Req-4	Software Teams should be manned by Civilians and Military personnel	Hyde LMPC ISADS	Non Functional	Document Inspection
Req-5	Quality Management needs to be appropriate across organisations	Hyde	Non Functional	Document Inspection
Req-6	Software Support should be flexible and responsive to operational needs	ISADS E2E	Non Functional	Document Inspection
Req-7	Comprehensive Management plans need to be produced	Hyde	Non Functional	Document Inspection
Req-8	ILS should be used for defining software requirements	LMPC	Non Functional	Document Inspection
Req-9	Metrics should be established that collect design costs	ISADS E2E	Non Functional	Document Inspection
Req-10	There should be one design and release authority for Software	ISADS	Non Functional	Document Inspection
Req-11	Support rigs should be provided for Query Answering and Problem Evaluation service	ISADS	Non Functional	Document Inspection
Req-12	Closer working relationships should be pursued between the Military and Industry	ISADS	Non Functional	Document Inspection
Req-13	UOR capability should be part of any partnered solution	ISADS	Non Functional	Document Inspection
Req-14	There is a need to maintain the Intelligent Customer capability	ES(Air) SSP	Non Functional	Document inspection
Req-15	A Reduced financial footprint for Software Support and sustainment is required	E2E	Non Functional	Document Inspection
Req-16	Software Capabilities are to be sustainable for the future	E2E	Non Functional	Document Inspection
Req-17	Support Functions should be centrally located, where possible	E2E	Non Functional	Document Inspection

Table 3 - Partnering Requirements

<u>SUMMARY</u>

It can be summarised that there have been studies carried out by both the Military and Industry that have looked at Software Support and sustainment, but they have only presented the issues and the need for change. There is now a need to propose a solution that can meet the requirements identified in the studies, and captured in Table 3 above. Namely, Software Support should use Sustaining contracts in preference to ah-hoc contracting, that any software organisation should comprise of both Military [Blue Suits] and Civilian personnel, to allow both organisations to retain their Domain Knowledge and the Intelligent Customer position, and that the financial footprint and WLC needs to be reduced.

Even though the above points have been captured in studies that are up to 7 years old, the recommendations are still considered valid. This is due to the length of time that the procurement and in-service phases associated with Military aircraft spans. These requirements have been confirmed as still valid by various IPTs who are either introducing software intensive platforms or are supporting them today. The next Chapter will look at the evidence captured within this Chapter; the studies; requirements and propose a solution that can be taken forward.

CHAPTER 7 - RECOMMENDED SOFTWARE SOLUTION

INTRODUCTION

This Chapter contains information on the main options for Software Support and why these options could be applicable to a partnered solution. Any solution must meet the requirements stated in Table 3 and needs to have the attributes to allow the solution to be "sold" to relevant stakeholders. For this reason it is considered pertinent to discuss the main characteristics of a single proposed solution, along with the risks and benefits associated with any change from the methods used today towards a single partnered software solution. The structure of this Chapter is represented in Figure 19 below.

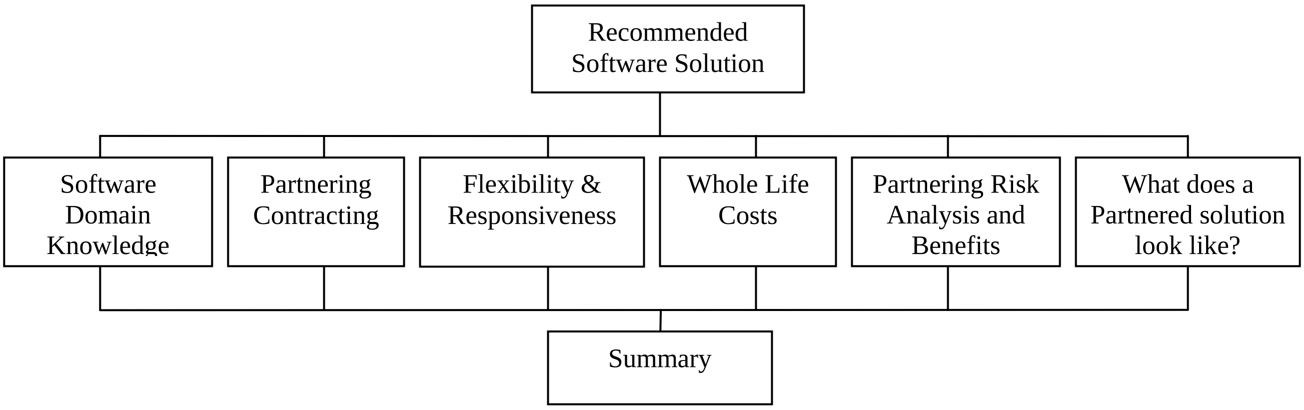

Figure 19 - Chapter 7 Structure

The information contained within this chapter needs to be considered carefully to allow assessors to evaluate if partnered solution is appropriate. The different levels and options for Domain Knowledge acquisition and retention need to be considered, how Contracting needs to change to take forward any change towards a Partnered Software solution; the needs for flexibility and responsiveness in meeting the users needs balanced against WLC, and finally managing the risks and quantifying the benefit of Partnering, only when all this information is captured can an informed judgement be made on whether a Partnered solution is appropriate. This judgement can either be from an Operational or Commercial perspective, indeed for some organisations that have cornered niche markets, it may not be desirable. But for larger organisations that rely on others there are benefits to be gained.

SOFTWARE DOMAIN KNOWLEDGE

In the previous Chapters there have been three main organisation structures that support software today. These are the pure Industry option, the pure Military SST and the hybrid SSC. All of the three support options have advantages that could be exploited for the benefit of future Software Support and disadvantages that, if eliminated, would allow for a better future for all stakeholders. Table 4 below shows the areas where Domain Knowledge resides today, it can be seen that no one organisation has all the knowledge to allow them to be fully effective at Software Support, but all areas where specific knowledge is required is addressed.

Domain Knowledge Areas	Industry	SSC	SST
User Domain Knowledge	Capability does not exist	Capability exists	Capability exists
In-depth systems knowledge	Capability exists	Capability partially exists	Capability partially exists
Organisation retaining Intelligent Customer	Capability does not exist	Capability partially exists	Capability exists
Duplicate organisation	Capability does not exist	Capability exists	Capability does not exist
Design Authority capability	Capability exists	Capability exists	Capability does not exist

Key
Capability does not exist — Capability partially exists — Capability exists

Table 4 - Domain Knowledge Areas

If the DLO passed all software changes to a pure Industry organisation, there would be problems, as Industry does not have the relevant Domain Knowledge of the users operational environment. This could mean that requirements are misunderstood, leading to the incorrect product being delivered or an increase in cost due to rework. But Industry is the only organisation with Design Authority approval; this approval is required by the Secretary of State for Defence for all software that operated over UK airspace, so there will always be reliance on Industry. Additionally, if the Military rely on Industry for full software development and support of software, where would the Intelligent Customers in the future come from? This is because the Military would not have the experience in the Software Support domain.

The SSC addresses the issue of user Domain Knowledge but they do not have the in-depth systems knowledge to maintain an Intelligent Customer capability; however they are used within commercial organisations and can address the need for DA knowledge. The SSC would be appropriate if the DLO were happy to become just requirement gatherers, arguably they are not, as the Intelligent Customer capability is still needed.

The SST addresses all the issues for Domain Knowledge except, for the in-depth systems knowledge that can only be provided by Industry. The main disadvantage is that by using a SST there is a duplication of facilities that increases WLC. If we were to rely on just a SST, then when major systems enhancements were required the in-depth knowledge that Industry provides today would not exist. This would increase the time to capability realisation, as this knowledge would need to be reacquired through reverse engineering.

The challenge would be to optimise the; Industry; SSC and SST organisations, then draw out the strengths and remove the weaknesses. The ideal solution would be for the users (the blue suits), to be involved throughout the whole support process. They would have greater involvement in the User facing functions of Problem Evaluation and Qualification. Industry would have involvement in all functions but only taking the lead in actual software modifications and the umbrella activities of QM and process control. This would allow Industry to retain assurance that their DA responsibilities were not being compromised. By using a single organisation, this would remove the duplication inherited by the use of a Military SST, therefore reduce WLC. A partnered solution that involved both organisations exploiting their Domain Knowledge would address the present shortfalls of a single organisation supporting software.

PARTNERING CONTRACTING

One of the issues highlighted in the documents and studies reviewed was that present contracting is "dictatorial and does not promote the attitudes for partnering". From NAO documentation reviewed, it has been established that initial contracting times can vary from 6-months to 3 years, with the majority of the main Software Engineering activities taking between 2-3 years in addition to the contracting phase. This contracting time is clearly unacceptable for capabilities that are required to meet operational realisation times, with up to 50% of the time being taken to negotiate the contractual hoops of Smart Acquisition and competitive tendering. Lengthy contracting is considered to be unavoidable for both new platforms and major modifications as it still adds value, but it is surmised that with relatively minor software modifications that are similar, continual contracting adds little value. For this reason, it is considered pertinent to consider options for changing the present contracting methods from Ad-hoc to an overarching agreement.

In reality there are a limited number of companies that actually have the Domain Knowledge and capability to modify aircraft software. There are examples [IPT05] where a company has just been selected for a contract to modify software but by the time the Software Engineering starts the contract for the next update is being considered. If an organisation was to form a long-term partnership, supported by a commercial framework it could reduce this repeat cycle and therefore reduce the cost and timescales associated with software modification.

The alternative to ad-hoc contracting is for a Service Level Agreement (SLA) to be established between organisations. This approach has been used for levels of availability for IT services and has been trialled on a small scale on the Merlin Helicopter project. The Merlin IPT has an agreement in place that commits them to spend a specified amount of money each year on either sustainability or software modification. The advantage of this approach, is that the IPT can plan their long-term spend for a platform and the company knows in advance the level of funding they will receive each year, without having to renegotiate a contract each year. The company concerned can manage their infrastructure costs, allowing for the remaining funds to be allocated to a level of effort, or people. Figure 20 below represents that by using this approach the level of effort can be predicted into the future as the infrastructure costs can be apportioned upfront. There is no wastage on repetitive contract renegotiation after the initial contract agreement; the following two years after 2001 have increased software effort for the same amount of funds as no additional contracting is required, 2004 shows an increased level of effort requirement for the future, allowing Industry to recruit as needed. This is a benefit to the IPT and Industry.

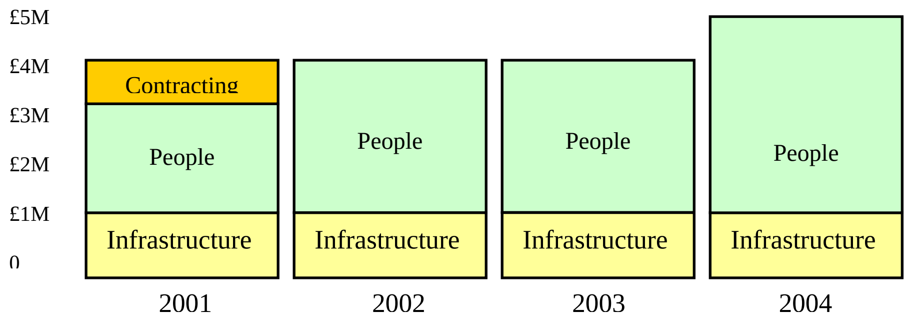

Figure 20 - Service Level Agreement Contracting

There is an example within Industry where the constant setting-up of a team and the cycle of contracting have been overcome. They have accepted that there will always be an initial cost for the establishment of an organisation and that is built into the initial business case for the software partnership, also the sustainment cost for the infrastructure and licences cannot be reduced by any significant amount. The only real variable that can impact the major costs is manpower. For these reasons the contracting method used is for a Standing Team that has secure employment for a set period of time. The MoD commits funds to the contractor three years in advance and as every year passes they assess their needs and available finances and commit to the next year beyond the present 3 years already agreed. A rolling contract with firm financial commitment for the next 3 years. This allows a Standing Software Team to be in place, which gives both partners the ability to plan for the future and it allows them the ability to terminate the contract 3 years in the future if either are unhappy with the others performance.

FLEXIBILITY AND RESPONSIVENESS

The issue of an organisation being flexible at appropriate stages during the development phases and responsive to the Customers needs has always been difficult for Industry to achieve when compared to the Militaries own SST, as highlighted in Table 2. The main issue preventing flexibility and responsiveness is the construction of ah-hoc contracts tied to payment milestones and the uncertainty of funding-lines. If the Customer would take responsibility for late changes in Requirements and not penalise Industry, this would allow organisations to react to changes more appropriately with the Customer managing the risks to their operational capability. These contractual and funding constraints experienced by Industry do not exist with the Customers SST, so there is a need for an overarching contract or agreement that allows for Requirements flexibility during appropriate development stages, with risk management being carried out by the Customer.

This present situation is that some platforms have software that is developed and maintained by two organisations, Industry and an SST, this approach is used to mitigate against inflexibility. These organisations have different processes that are intended to meet the same development objectives, with both organisations having strengths in different development areas, these processes include:

- Query Answering and Problem Evaluation.
- Change request prioritisation and Management.
- Requirement capture.
- Development processes.
- Testing processes.
- Release of software modifications.
- Quality Management Systems (QMS).

For these reasons the organisations should look at their present processes and draw out best-practice from each other, it was found that the Military SST have a better Requirements process with high levels of prototyping and end user involvement and that Industry have a better QMS based around CMMI. Both organisations have processes to offer the partnership.

WHOLE LIFE COST

There are fixed costs that will always be associated with software modification and it is irrelevant on the size of the modification, these costs cannot be removed. For software to be developed you need an appropriate development and support environment which consists of facilities that contain test rigs and software, and the people employed to develop any software solution. All of these attract costs, either for; heating, lighting, maintenance, licence costs or wages, but it needs to be recognised that these costs are unavoidable.

The only real variable between organisations, once the initial set-up costs have been quantified, is the charging rate for personnel and the number of people required to carryout the software changes. Figure 21 below represents a 25 year project with costs being plotted for man-power only, the cost represents a team of 38 people[22]. Using charging rates are from the NAO, when these costs were plotted over the 25 year lifecycle, it can be seen that when compared to Industry the Military are considerably more cost effective. In WLC terms a fully manned Military organisation would be cheaper, its Industry equivalent is more expensive by £105M over a 25 year project, and a partnered team would fall between the two extremes assuming a 50/50 split of Industry and Blue-suits.

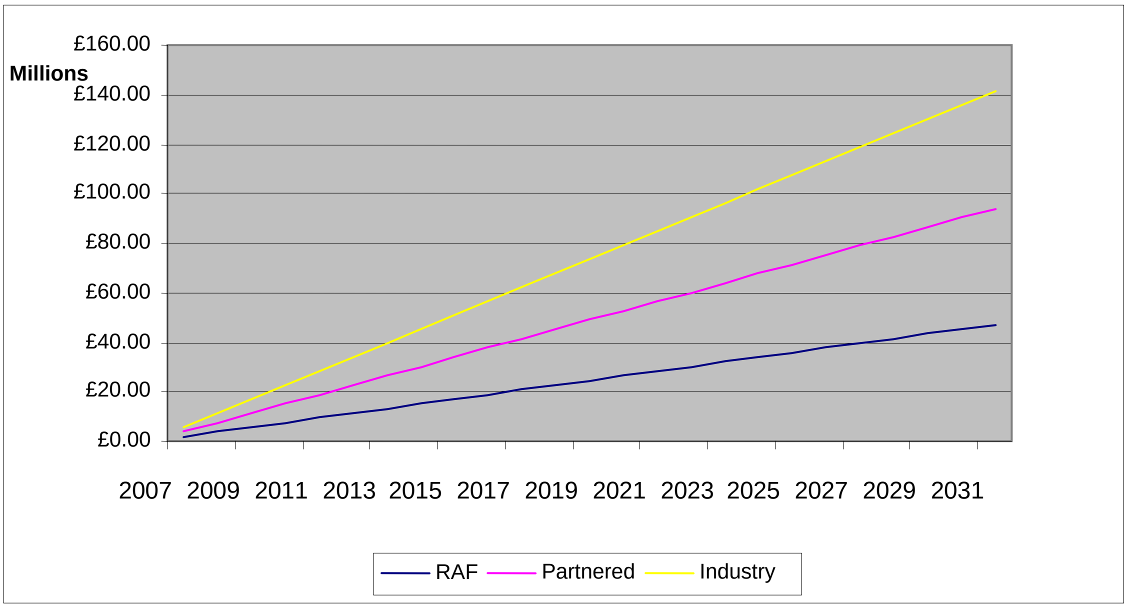

Figure 21 - Comparison of Military and Industry costs over time

If the decision for Software Support was based only on WLC, then limited analysis would be needed, but this is not the case. When assessing the capability of an organisation, due recognition needs to be taken of both the Domain Knowledge and long-term sustainability of the organisation that is supplying the service. A service may be cost effective, but if it lacks the knowledge or is not prepared to provide a capability for 25 years, the platform would be unsustainable. What the diagram above does not show is that if you have a fully manned Military team, you also need a team within Industry the provide DA approval for software. This will remain unchanged as long as Industry will not accept SST processes for software development, as was highlighted in Table 2, and this duplication increases the cost of using a single Military SST consideration.

[22] 38 people is the size of one of the present Military SST.

PARTNERING RISK ANALYSIS

During the analysis of the documentation available on Partnering along with the Requirements identified in Chapter 6, captured issues and information from the IPT, it became apparent that the Risks of creating the partnership have not been captured. For this reason a risk register was generated along with risk mitigation methods and the associated Cause and Effect analysis diagram. These can be found in Figure 24 and Table 12 in Annex E. Part of any Risk analysis activity is to prioritise the risks that need addressing first, for this reason the "Danger Slope", Table 11, at Annex E was produced that allocates a priority number based on the effects and likelihood of a risk occurring. A summary of the risks, by severity, is captured in Table 5 below, with the top nine risks being highlighted in the Cause and Effect diagram at Annex E and in Table 6 below.

Severity	Quantity
Low	0
	5
Medium	13
	9
	14
High	6
	3
Immediate	0

Table 5 - Risks summary

Risk Number & Description	Nature of uncertainty
5 - Costs too high	It may happen that the cost of the partnership will be too high
7 - Military needs not met	It may happen that the Military needs are not met
10 - Release too slow	It may happen that the software will not be released in the correct timescales.
12 - Sustainment too expensive	It may happen that the sustainment costs for the partnership are too high
15 - Costs too high	We are uncertain of the costs involved in the partnership
21 - Contract too slow	It may happen that the contract for the partnership will prevent software development starting at the correct time
28 - People too expensive	It may happen that the people cost id too expensive
35 - Not enough funds	In may happen that the Military are unable to secure the level of funding to maintain the partnership
36 - Partnership too expensive	We are uncertain of the full cost of the partnership

Table 6 - Top 9 Partnering Risks

The highest rated risks in Annex E are financial and relate to the MoD IPT ability to provide sufficient funds for the sustainment of the Partnership and the other six are either financial or the establishment of the initial contract. This confirms the experience of the IPTs and the information in Chapter 6, that the technical issues can be overcome, but if funding-lines and contracting methods are not adequately addressed, this can strain the relationships between organisations to the point were partnerships are dissolved.

The Risks captured above revolve around the assumption that a partnership is desirable, therefore the risks relate to a failed partnership. There are risks to forming the partnership that are expanded as follows:

Commercial risks. When forming partnerships, organisations communicate better and exchange more of their proprietary information. There is an inherent risk that information exchanged is valuable to other partners, beyond its intended purpose. For this reason adequate commercial provisions need to be in place to protect individual IPR and to allocate IPR to any software that is produced by the Partnership. Also, any partnership needs to be financially viable to make it a success; there is a risk to the partnership, and the individual organisations, that if enough funds are not made available to the partnership, it will be dissolved and the capability lost.

Technical Risks. With Partnering there is an inherent reliance on others, so if an organisation eroded their own autonomous software capability in favour of Partnering, what happens if the partnered agreement fails? Where do the Domain Knowledge and skills come in the future as essential skills would have been lost. About 13 years ago this situation occurred for the Royal Australian Air Force (RAAF), they contracted out their capability to neutralise Unexploded Bombs (UXB), but the contracted company collapsed after 5 years due to a lack of skills. At the time many experts from the RAAF left and were employed by the company concerned. The company had a good reputation but instead of training new people internally, they had always relied on people leaving the RAAF. But as the capability was now contracted out, the RAAF training stopped so the pool of expertise dried up, putting the company in an unsustainable situation so the contract was terminated. It has now taken 8 years for the RAAF to regain this UXB capability which was needed during recent conflicts. Although this example is not software related, the analogy holds true, in that the autonomous capability was lost and it took time to recover the specialist Domain Knowledge and skills, *"very often on software projects, the loss of knowledge is triggered by the loss of key personnel, and the projects slips [BE/RA02*], the need to retain Domain Knowledge is the key attribute for sustainable software support.

End-User Risks. The main risk to the end Military users is that the partnership will be unable to produce software to the right level of quality at the right time, therefore impacting on operational capability.

PARTNERING BENEFITS

There are tangible benefits that come out of software partnering; these need to be evaluated along with the risks that have been identified above. The main benefits are captured under the three categories below:

Commercial Benefits. When a long-term partnership is established there are commercial benefits for all the companies or organisations involved. These relate to the removal of competition, allowing partners to concentrate on their strengths and not on their old competitors; there is less commercial interaction for software changes to be realised, so saving money; and when partnerships go well their collective credibility is increased, therefore increasing the potential for more business.

Technical Benefits. From a technical perspective, partners can draw best-practice from one another, this can help to spread the burden of updating technologies or to mitigate against the obsolescence of either; tools, languages or processes. An example of this would be platforms that are just entering RAF service, like Typhoon, that contains ADA. The ADA language is becoming harder to support and will eventually become uneconomical from a commercial perspective, therefore making software difficult to update. It would be cost prohibitive for one organisation to unilaterally change from using ADA to, for example, C#. This is because the training burden and the cost of tool and process updates would be too expensive, the Return-on-Investment (ROI) would be hard to justify within a single commercial organisation. But with a Partnered organisation by using the economics-of-scale to purchase new tools or to employ consultants to change processes, could present a case for change therefore updating technologies. It is possible that one partner could take responsibility for each of the individual elements required to change the use of a language, e.g. training, process change or tool costs; therefore spreading the burden and making the ROI acceptable.

End-user Benefits. It can be argued that most end-users are uninterested in partnerships, costs, or the benefits to organisations. Their only interest is that their product works correctly and when it fails it can be fixed or updated quickly. The benefits of forming partnerships to end-users are that they get the correct produce when it enters service, as the Requirements are correctly understood, and when software modifications are required they are embodied quicker when compared to non-partnered modifications.

The benefits of partnering identified above are been seen on both the Harrier and Tornado software programs, but these can only be fully realised by forming better commercial relationships, allowing for flexibility and an increased understanding of the benefits to the collective from drawing the strengths of the individuals. These benefits can be realised across different disciplines, not just software:

> "*We are Borg, individual strength is irrelevant, resistance is futile, we wish to improve ourselves, we will add your biological and technological strengths to our own, your culture will adapt to service the collective*" - The Borg 2005.

WHAT DOES A PARTNERED SOLUTION LOOK LIKE?

From the information above, the characteristics of the partnered team can be quantified. The ideal partnered organisation must have sufficient Domain Knowledge to allow it to understand the requirements of the users, the actual software cost and system impact of proposed software changes, these can only be assessed with in-depth systems and operational knowledge. For this level of understanding to be achieved a partnered organisation should contain both experienced industrial people and blue-suit personnel that can understand [and interpret] requirements and have the ability to act as the Intelligent Customer.

To incentivise organisational level improvements, gain-share should be used supported by common processes that remove the possibility of inducing errors and to allow Industry to retain its Design Authority confidence in the software being produced.

There must be an appropriate contract in place that allows the organisation to remove non-valued activities of repeat contracting, but the contract should be flexible enough to allow requirements to change at appropriate phases of development, with the risks associated with changing requirements owned by the most appropriate organisation – this is assumed to be the IPT for Customer generated changes. This must be balanced against the WLC constraints and risks to both the Military and Industry positions, be these; economic, political or operational. This approach has worked on a small scale for the Merlin IPT, but not for full development and support activities highlighted in the Software and data Support model at Figure 5.

SUMMARY

In summary, there is sufficient information available within both the Military and Industry community to define the characteristics for a partnered Software Support organisation. These have been highlighted within this Chapter and the vision of what the organisation would look like captured. The following Chapter will compare the initial issues and partnering requirements and draw a conclusion as to whether partnering is an appropriate path to follow.

CHAPTER 8 - IS A COLLABORATIVE-PARTNERING APPROACH APPROPRIATE?

INTRODUCTION

This Chapter contains a reflection on the subject matter and conclusions based on the information presented by the author. It will also assess whether partnered Software Support is an appropriate approach for the Military or Industry organisations. The Chapter structure is a represented in Figure 22 below.

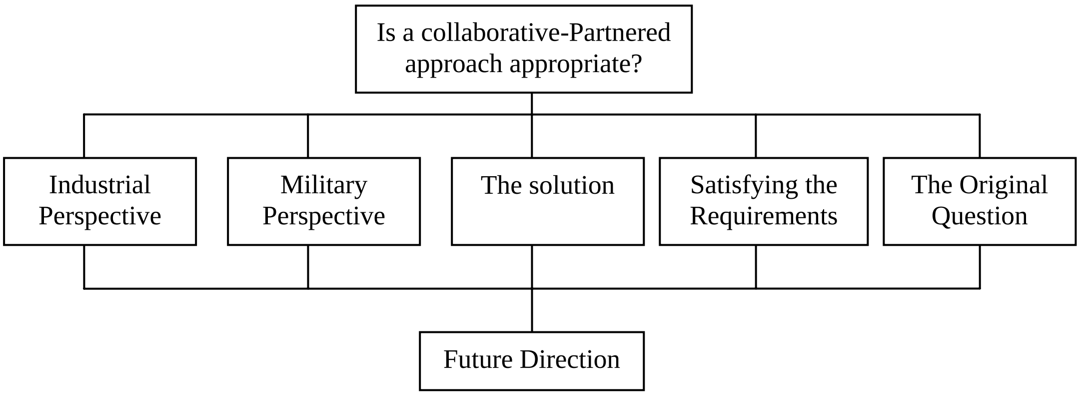

Figure 22 - Chapter 8 Structure

INDUSTRY PERSPECTIVE

From the research carried out there are different factors that drive the decision towards a partnered collaborative approach. The Industrial perspective of maintaining an on-shore capability to design future platform or upgrades and support present capabilities is highly desirable. Without a throughput of funding and research the UK Industry will not be in a technical position to address the national need in times of conflict. With the changes on the horizon there is a real possibility that all the industrial players will not survive to see the next generation of air platform. For these reasons and the items listed below, a partnered approach is considered desirable:

- Enhanced Customer perceptions of Industry abilities.
- Guaranteed long term revenue stream.
- Closer to the front line Customer, better operational Domain Knowledge.
- Better position to reduce costs.
- Transparency of Customer SSTs organisation and business activities.

MILITARY PERSPECTIVE

From a Military perspective a capability needs to exist that can support the national interest whilst being cost effective. Whilst Industry could provide the support capability it has been shown that this is cost prohibitive and the situation will not improve in the future. By adopting a platform approach this will lower the cost, improve requirements capture and retain the intelligent customer position. Additional benefits would also be:

- Reduced cost of ownership, no duplication.

- Planned support activities, including capability upgrades.
- Reduced time to contract.
- Faster response than traditional Software Support.
- More efficient use of resources and funds.
- An enhanced intelligent Customer position as Industry knowledge will be migrated to the Customer SSTs.
- Transparency of Industry organisation and business activities.

THE SOLUTION

By taking into account the different perspective and the maturity of platforms there are two solutions that should be taken forward for the future. One solution for present platforms, such as the Tornado GR4 and the Harrier GR9, and another for newer platforms such as Typhoon and JSF.

Legacy platforms. For legacy platforms the ideal solution should be to identify the areas of change traffic based on past operational capability upgrades and the TLCMP. The areas identified should then be partnered based on a through life business case. The long-term aim should be for the Military to take over support in full in the sun-setting years of the platform. As was experienced with the Jaguar and Sentry platforms, Industry drift away from supporting older platforms as funding-lines are reduced and technology stagnation occurs with the Military taking the lead role.

Future platforms. The platforms that are entering into service in the future are highly integrated when compared to the present in-service platforms. For this reason and the drop in the training given to new service personnel, as represented by the Army and RAF Apache experience, the Military will not have the skill sets to support platforms to the same degree in the future. Therefore they need to concentrate on core functions that ensure the in-service phase is as cost and operationally effective as possible. Partnerships should be pursued with vigour gaining in-depth knowledge where available, but Industry should take the lead role.

SATISFYING THE REQUIREMENTS.

Chapter 5 provided a list of 17 high level requirements and Chapter 7 captured the risks associated with a partnered solution to Software Support and sustainment. Table 7 below shows which Requirements and Risks have been addressed by the partnered approach, there are no Requirements or Risks that partnering, with appropriate contracted, does not address.

Solution	Addressed Risks	Addressed Requirements
An Industry/Blue suit Team	Risk-12, Risk-15, Risk-28, Risk-36, Risk-36	Req-1, Req-3, Req-4, Req-12, Req-14, Req-15, Req-16
Appropriate Contracts	Risk-5, Risk-7, Risk-15, Risk-21, Risk-35, Risk-36	Req-2, Req-6, Req-10, Req-12, Req-13, Req-15
Common processes	Risk-10, Risk-15	Req-5, Req-7, Req-8, Req-9, Req-10, Req-11, Req-13, Req-15, Req-16, Req-17

Table 7 - Requirements & Risk against the Solution

THE ORIGINAL QUESTION.

Partnering is a way to retain Intelligent Customers, prevent monopolies, and provide Industry with enough work to sustain its baseline minimal capability, which would provide Civilian employment and technology sustainment. Partnering is the correct way to process, but with Military involvement in key areas to retain its ability to support platforms when Industry does not have the capability, capacity or interest in the future. The Military are excellent at developing and prioritising operational requirements and problem evaluation and Industry is best placed to provide in-depth knowledge for major functionality and systems updates.

As for the original question, “Is Partnering Software Support appropriate for Military Aerospace Platforms?” The answer is yes, but the degree of partnering depends on the levels of systems integration and the length of time a platform has left in-service. It is imperative that both Industry and the Military identify and retain their core capabilities and strengths, to allow for the exploitation of opportunities in the future. As represented in Figure 23, there are common problems that the organisation has, that were presented in Table 2, and common solutions that could pave the way for the future. Organisations should recognise that there should be; one Product, one Team and one software Baseline, the Military and Industrial needs are not mutually exclusive, we are actually reliant on each other

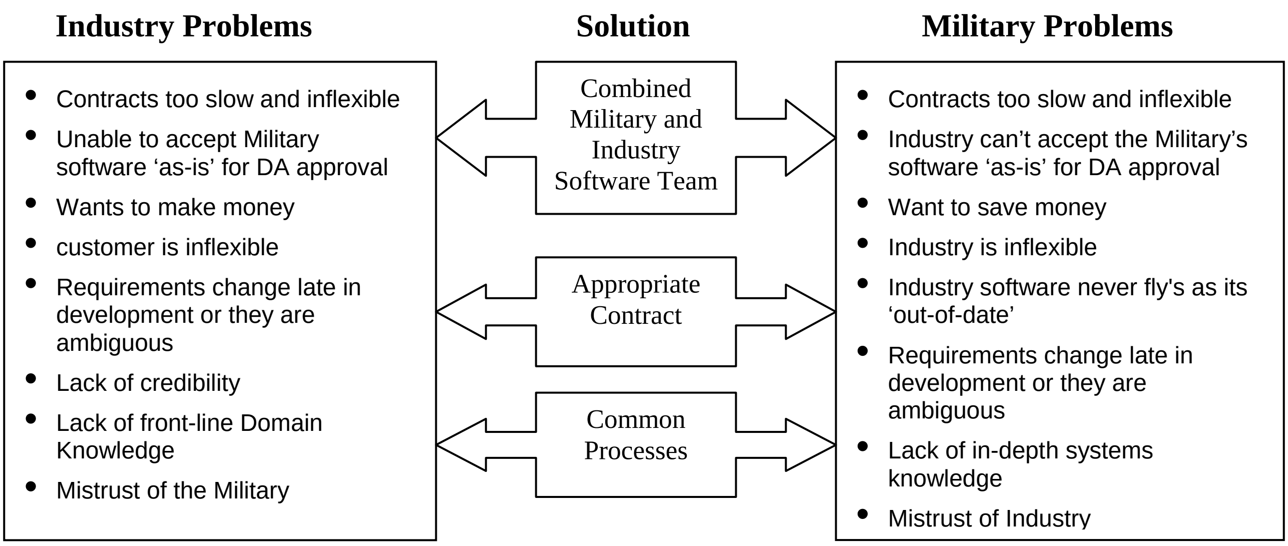

Figure 23 - Problems and Solutions

FUTURE DIRECTION

The future direction of the information captured and presented within this dissertation is to help define the Software Support solutions for the UK Military and collaborative projects. The information is now being used to direct Military IPTs and to assist them in identifying their actual needs, and to look realistically at their options for the support and sustainment of both platforms and employment throughout the Industry. By adopting a partnered approach, this provides stability for all stakeholders and the opportunity to reduce cost and provide operational capabilities in the correct timescales.

ANNEX A - PRESENT MOD, DPA AND DLO POLICIES

INTRODUCTION

This Annex contains a summary of the main policies for the MoD, DPA and DLO.

MINISTRY OF DEFENCE

The MOD policy relating to partnering is represented within the following documents:

- Policy Paper Number 4 (Defence Acquisition). A summary of the salient points from the Policy are as follows:
    - "The aim of this paper is to describe how the MOD goes about acquisition."
    - "The MOD used the word acquisition to denote the totality of setting the requirements for a equipment, facility or service: procuring that equipment, facility or service: and supporting it through life – all the way to its disposal."
    - "The aim of smart acquisition is to enhance defence capability by acquiring and supporting equipment more effectively in terms of time, cost and performance" …" acquiring capability progressively, at lower risk, and with the right balance between Military effectiveness, time and whole-life cost."
    - "The MOD is using PPP to develop better ways of supporting the Armed Services…" and "The MOD has no dogmatic preference for private over public, or vice versa."
    - "…it is necessary to understand the strengths and weaknesses of both public and private sector bodies. The MOD's aim is always to develop relationships that capitalise on the strengths…"
    - "Bringing about a profound change in the nature of the relationships between the MOD and its suppliers is a key feature of Smart Acquisition and a key aim of the DPA, the DLO and DE."
    - "To introduce both incentivisation and gain-share."
    - "A partnering arrangement is a form of long-term contract that establishes a framework within which the departments relationship with a contractor can grow with time."…"a Partnering Arrangement does not require the department to be so clear about the final outcome."
    - "…working together in agreed ways to introduce improvements in cost and performance, and to share the benefits of mutual success."
- Policy Paper Number 5 (Defence Industrial Policy). This recognises the importance of Policy Paper Number 4 with a summary of the relevant points from Policy Paper Number 5 as follows:

- “committed to public private partnerships to deliver benefits in the provision of defence services”.
- “...a new emphasis on closer co-operation and openness in our relationships with Industry.”
- “The Government’s Defence Industry policy is founded on the importance of equipping our Armed Forces efficiently with the tools they require to meet the challenges they face.”
- “Competition is the MOD’s primary means of achieving value for money, and any decision which would impact on the ability to compete future requirements…need to be considered very carefully.”
- “Competition will therefore remain the bedrock of our procurement policy.”
- “Whatever degree of risk is borne and managed by the contractor, the Armed Forces will always bear the operational risk of equipment or services that are not delivered on time or to the performance standard required.”
- “With Industry increasingly involved in providing long-term services to the MOD, we have recognised that a partnership approach, building reliable links with our suppliers, is often the best means of realising our goals.”
- “Partnering does not mean creating privileged or monopoly suppliers, which could stifle innovation and result in inflated prices for inferior equipment”.

DEFENCE PROCUREMENT AGENCY

The Defence Procurement Agency (DPA) policy relating to partnering are represented within the following documents [AMS06/1]:

“The Government has involved the private sector in Public Private Partnerships (PPPs), a procurement initiative which enables budget holders to make the best use of MOD and Industry capabilities, skills and experiences. Partnering provides an environment in which MOD and its suppliers work together from the earliest stage of a procurement process to satisfy MOD’s requirements in the most efficient and effective way”. This guidance addresses partnering with MOD in the three bullets below:

- Partnering in MOD. The main emphasis is as follows:
    - Competition remains the key tool in achieving value for money.
    - Appropriate benchmarking reassures MOD that partners remain competitive in the market place.
    - Taut contracts support-partnering arrangements.

  - Partnering arrangements are suitable for the procurement of services, equipment and material.

  - People are a key asset.

- Creating Partnering Relationships. This focuses on the attitudes to be addressed by those involved:

  - The decision to seek a partnering arrangement must be made after a study of all procurement options.

  - To flourish, the partnering arrangement must exist in a culture of openness and co-operation.

  - Partnering must have top-level MOD and Industry support.

  - Champions are appointed from both MOD and its partner, with the responsibility to advocate pro-active support, in word and deed, for a partnering arrangement.

  - A single integrated stakeholder team made up of representatives of MOD and its partner(s) manages the partnering arrangement.

  - The documents that support partnering are drafted by negotiation and agreement. All partnering arrangements are underpinned by legally enforceable contracts.

- Partnering Management. This provides an introduction to issues that need to be addressed and agreed at the outset of the relationship, and subsequently developed:

  - A change procedure.

  - Performance monitoring arrangements and the associated information requirements (benchmarking).

  - Mechanisms for problem solving and dispute resolution.

  - Intellectual property rights.

  - Termination and exit strategies.

  - Risk management.

  - Performance incentives including contract pricing and payment mechanisms.

  - Post-project evaluation.

DEFENCE LOGISTICS ORGANISATION

The Defence Logistics Organisation (DLO) policies relating to partnering are represented within the following documents [AMS06/2]:

- Constraints. The IPT leader shall ensure that the negotiations to structure and document a partnering arrangement and its subsequent operation avoid the creation of a Partnership as defined by the Partnership Act 1890.

- Authoritative Guidance:

    - When developing the Through Life Management Plan the IPT Leader shall explore all acquisition methods through a process of studies and, if appropriate, determine a partnering arrangement as the preferred procurement tool.

    - The option for a partnering arrangement should be included in all advertisements, tender briefings and in the invitation to negotiate.

    - The IPT Leader together with the Project Manager (Partner) shall jointly negotiate the Partnering Documents: The Partnering Principles; the Framework Agreement; the supporting contract(s) or tasking orders.

    - The IPT Leader together with the Project Manager (Partner) shall establish an Integrated Stakeholder Team to include skilled representatives from both MOD and Industry.

    - The IPT Leader together with the Project Manager (Partner) shall jointly commit the partners to the formal initiation of a legal relationship by the exchange of the Partnering Documents, signed by authorised representatives of both parties.

    - The IPT Leader together with the Project Manager (Partner) shall jointly manage the Partnering Arrangement by implementing policies and procedures negotiated during the production of the Partnering Documents.

ANNEX B - COMPARISON OF PRESENT SOFTWARE SUPPORT TEAMS

The table below contains a summary of the types of organisations that are supporting software today.

Platform	Support Team	Comments
Apache	IPT with SSC	SSC – Carryout query answering, problem evaluation and requirements capture activities
ASTOR	IPT with SSC	Will be located at RAF Waddington
Harrier GR7	SST	Possibly a Partnered Team supported by BAE SYSTEMS
JSF	TBD	Assumption that a Partnered Team will exist, location [commercial]
Merlin Mk3	IPT with SSC	SSC at GKN-WHL. Software study started in 2001 to look at the possibility of a SST
Nimrod MR2	SST	PDS with Customers SST
Nimrod MRA4	Embryonic Partnered Team	Driving towards a Partnered Team, supported by BAE SYSTEMS and its partners
Nimrod R	SST	PDS with Raytheon Field Service Representative (FSR) on site with Customers SST
Sentry	Team with SSC	Joint with Boeing with a FSR on site
Tornado GR4	SST	Joint Software Team
Typhoon	Embryonic Partnering	Driving towards a Partnered Team with BAE SYSTEMS and their partners

Table 8 - Comparison of Software Teams

ANNEX C - PREVIOUS STUDIES RECOMMENDATIONS

INTRODUCTION

Below are the main points extracted from the reviewed reports and studies referred to in Chapter 4 that have been evaluated for this dissertation.

HYDE REPORT RECOMMENDATION

- The recommendations from this report are as follows:
    - "An In-Service operational Software Support capability is maintained and that the Harrier Software Maintenance Unit (HSMU), Nimrod Software Team (NST), Air Defence Variant Software Maintenance Team (ADVSMT) and Tornado In-service Software Maintenance Team (TISMT) should continue as Service managed SSTs".
    - The SSTs should continue to be tasked to provide support for operational Software Support in both peace and crisis and war.
    - The Sustaining Engineering Contract (SEC) between HSMU and BAE is a good model of Service/Industry co-operation which should be adopted by other SSTs where they continue to have working relationships with Industry.
    - The officers and tradesmen employed in analysis, programming and testing duties within the SSTs should continue to have recent role and systems knowledge and experience.
    - Aircrew with current role and operational experience, and aptitude, should continue to be employed in management and, where appropriate, analysis, programming and test posts within the SSTs although some line management posts, currently filled by aircrew, could be filled by officers from the Operations Support or Engineer Branches.
    - All officers, all personnel involved in the analysis, programming and testing of operational software. And all bureau support personnel should be considered as core staff under Uniformed Regular Core Manpower Requirement (URCMR) Criterion 3, logistic support, and/or Criterion 5, essential skills.
    - Personnel involved in the programming of engineering or ground maintenance software, quality assurance, rig maintenance, configuration management and administration should be considered as non-core.
    - The four SSTs should be subject to a Manpower consultancy by the Headquarters Strike Command (HQ STC) Establishment Branch and that this should be preceded by a Constructive Cost Model (COCOMO).
    - A Manpower consultancy should take into account the relevant recommendations of this Report and of any relevant recommendations that may arise from the parallel Study by RIU on the impact of Logistics Trade Review (LTR) on SST manning.

- Civilianisation of non-core posts, in the first instance should be limited to a maximum of 5 posts per SST, and to the lower grades for which there is a wide skills base in the Civil Service.

- Contractorisation of non-core parts or functions should not be undertaken.

- A reduction of NST Acoustics Flight to off-set the costs of a new acoustics processor should not exceed 30 staff if continued Software Support to any significant level will be required.

- All operational SSTs should be given the right to assess all candidates for employment in core posts and to reject those who do not have the aptitude or motivation.

- The Software Configuration Management Board (SCMB)s should continue to operate as the principal means of assessing and controlling the tasking of both SSTs and Industry but should adopt a formal method of scoring the operational benefits, ease of change and the priority of software changes.

- The SCMBs should continue to control, monitor and recommend the release to service, including airworthiness considerations, of all operational software changes produced both In-Service and by Industry.

- The co-chairmanship of the Air Defence Variant (ADV) SCMB should be returned to the Operational Sponsor, or, failing that, be transferred to a member of the Tornado F3 operational community other than OC ADVSMT [23].

- The SCMBs for the Nimrod MR2 should operate under the co-chairmanship of the operational sponsor and the appropriate SA.

- The certified quality management systems used by each SST should be retained and maintained.

- The SSTs should remain under the full command of AOC in C STC.

- The Air Warfare Centre (AWC), or the appropriate AOC, should be given command and control of the SSTs including budgetary and functional control for the SSTs.

- If the AOCs are given command and control, the functional control should be exercised through the relevant aircraft role office which must be a member of the appropriate SCMB.

- The Nimrod MR2, Tornado F3 and Tornado GR1 Support Authority (SA)s should consider transferring the maintenance of computer hardware used by the SSTs to HQ STC Single Source Maintenance contract.

- Management Plans for each SST should be introduced as expeditiously as possible, they should cascade down from the management plan of the parent organisation, and the SSTs should collaborate in producing a set of common and consistent plans and PIs.

[23] ADVSMT was disbanded in 2006 to coincide with the Tornado F3 planned retirement date.

- o The efficiency of the SSTs should be reviewed by means of the reports normally required at the end of selected accounting periods and by a formal on-site functional review conducted annually at 1-Star level.

- o After TISMT has established itself at Boscombe Down, consideration should be given to amalgamating it and HSMU to form a Strike/Attack SST.

LPMC PAPER RECOMMENDATION

The recommendations from this report are as follows:

- "The application of Integrated Logistic Support and Logistic Support Analysis is the core process for defining Software Supportability requirements and the feasible support solutions."

- "The core strategy for the support of operational-critical software in future aircraft should be Service/contractor teaming , with the following criteria:

    - o RAF resources must exercise authority, as appropriate to their role, such that management and design activities for software change implementation are undertaken on an equal footing with contractor resources.

    - o SST manpower should compromise an appropriate combination of contractor staff, Ministry of Defence (MOD) civilians and the minimum necessary Service compliment. Service posts should be established on the basis of the URCMR criterion for Logistics Support."

- "Guidance should be provided for ILS Managers on the compilation and development of Software Support strategies. The guidance should cover the following aspects:

    - o Model arrangements for Service / contractor teaming.

    - o Criteria for the use of Service/MOD manpower.

    - o Harmonisation (where appropriate) of software and mission data support."

ISADS REPORT RECOMMENDATION

The recommendations from this report relate to systems and Software Support are as follows:

- Requirements and Customer Need.

    - o "An ES(Air) policy be established to set out the In-Service Aviation Design Support Policy. This policy should be the framework for a management strategy that sets out how to deliver the necessary design services. In turn, the strategy should be communicated in the DTech (Air) Management plan, and this should include the necessary metrics against which performance can be monitored."

- With all new projects and major equipment and platform upgrades, the responsible DEC, in close liaison with Customer 2, should ensure that the User Requirement Document contains clear guidance of the Software Support needs, particularly the required speed of response."

- Management and Organisation.
  - "A common design tool standard should be included in the future Information System strategy."
  - "A common approach to the collection and reporting of design costs should be developed in order that design services can be quantified accurately across the environment."
  - "All SSTs should be included in the AD/ADRP Design Approved Organisations Scheme."
  - "The STC A4 organisation should engage with other stakeholders to determine whether ownership of the ADVSMT, HSMU, TISMT and NST needed to re-assigned to satisfy better current Software Support needs."
  - "The STC A4 organisation should engage with other stakeholders to take a view on the ownership of SSTs supporting future platforms."
  - "A sponsor should be appointed for AP100D-10, and its content reviewed and re-distributed as necessary."
  - "SSTs should be provided with ready access to a collocated Stage 4 rig if they are to perform a design function adequately, including the problem evaluation and requirements capture task."

- Manning and Resources.
  - "It is recommended that civilianisation of Service posts within SSTs should not be pursued beyond the current levels."
  - "The Study Team concluded that under present circumstances, the TISMT and HSMU should remain as separate units at DERA Boscombe Down, but recommended that this situation should be revisited when the effects of DERA PPP and the policy on teaming with BAES is further developed."
  - "The current level of aircrew appointments within the SSTs should be maintained."
  - "It is recommended that a Wg Cdr post be established, ideally at AWC Waddington, to act as the SST Champion. It is recommended that a Wg Cdr be assigned to act as the SST Champion."

- Partnering.
    - “Closer working relationships should be pursued through the IPTs to improve the overall efficiency and effectiveness of the design processes.”
    - “Partnering arrangements should be considered between Industry and the SSTs to co-ordinate software releases and prevent duplication of effort.”
    - “The operational risk and life cycle cost implications should be assessed rigorously before deciding on teaming approaches with Industry which centre all of the design effort at the contractor's premises.”
    - “Partnering arrangements should allow Industry to employ Urgent Operational Requirements & Special Trial Fit procedures used by the SSTs to hasten the release of software issues.”
    - “That future teaming arrangements should provide software design capability both at contractor's premises and at a SST with appropriate cross fertilisation of manpower.”

ES(AIR) SOFTWARE SUPPORT STRATEGY

This report raises many question and issues to be resolved and makes only three recommendations, they are as follows:

- “The study sponsors endorse the in-service Software Support strategy.”
- “The Framework for Advice to IPTs becomes part of the Support Solutions Envelope.”
- “The strategy provides the basis for development of ES(Air)’s Software Support Policy.”

END-TO-END PAPER RECOMMENDATION

The recommendations from the End-2-End study are as follows:

- Air Equipment Support.
    - Minimise the deployed footprint.
        - “Conduct rigorous maintenance reviews for each platform with independent peer reviews to minimise activity levels, whilst maintaining airworthiness.”
        - “Adopt a systematic Lean approach to all remaining Forward support processes for each platform.”
        - “Redesign the Forward Force Element Table (FET) development process to ensure there is an appropriate incentive to seek minimum logistics footprint throughout.”

- Streamlining Depth Support.
    - "Systematically apply Lean techniques to all Depth support processes."
- Concentrate Depth Support infrastructure.
    - "In parallel with streamlining activity in Depth facilities, seek to concentrate activity in a single Depth facility."
    - "Carryout a cross-Defence rationalisation of airfield capability."

- Enablers.
    - Sustainability.
        - "On each occasion that a decision to allocate resources for sustainability is made, funding should be passed to the organisation best placed to execute the requirements."
    - Ownership of Logistics Personnel, Assets and Contracts.
        - "Transfer ownership of all Depth support infrastructure and all support contracts (including FSTA) to the DLO."
        - "Where service personnel are employed in Depth support, adopt the same command relationships as those in place under the Warship Support Modernisation Initiative (WSMi)."
    - Management of Logistics Support.
        - "Co-locate the day-to-day support and Technical management elements of the current IPT, Role office and Industry DA (Design Authority) functions with the Depth Logistics Hubs."
    - Performance Management and Outputs.
        - "Metrics for measuring logistics output and activity at all levels should be consistent with the RACERRS principles, and should reflect the CSAs. Where they do not already exist, such metrics should be developed, taking account of the current work on 2* and 4* CSAs. CDL, as process owner, should own the principles for developing logistics output metrics."
    - Contracting for support.
        - "In each procurement equipment equipment/service category, the DLO should develop an explicit strategy for improving supplier performance, both through the use of competition and associated performance-focused procurement strategies, and also through the targeted use of supplier development tools."

- “CDL to develop the overall logistics support architecture, including the joint supply chain process and infrastructure, and IS, and ensure the future support solutions are consistent with this.”

- “Embed the E2E logistics support principles and strategy into the SSE and into any support planning and contracting strategy in the DPA and DLO.”

- “Mandate the use of a Lean Public Sector Comparators (PSC) for all new investment appraisals & business cases.”

- “Set up a simple database to record the different incentive regimes for contracts, and the resulting behaviours, both anticipated and unanticipated.”

ANNEX D - GENERIC PARTNERING OBSERVATIONS

- If the in-Service SST are cheaper, more flexible and have proved that they can "produce the goods", why is emphasis being placed on partnering, the reduction of skills and the transfer of activities to Industry? This is considered to be a short-term financial vision with long-term WLC and flexibility implications that will be difficult to recover.

- As platforms increase in complexity levels, the emphasis on partnering will increase to reduce short term costs.

- Partnering can be beneficial for both the Armed Services and Industry, but it is not always appropriate or desirable.

- One size does not fit all; each project needs to be assessed independently, but overseen to ensure that long-term service needs are given due consideration.

- Partnering is undertaken for a variety of reasons:

    - Financial.
    - Tactical or strategic.
    - Directed by a higher authority.
    - Industry needs the skills that the Armed Services have to achieve their partnering commitments.
    - The Services need the DA expertise and systems knowledge.

- The 'Crown Jewels' are unclear within both the Armed Services and Industry.

- Fallback positions are sometimes unrealistic. When skills are passed to Industry they can't easily be re-established within the Armed Services.

- Common approaches across IPTs' are not apparent for:

    - Contracting levels or methods.
    - Partnering definition **–** Specific tangible information, not policy high-level terminology.
    - The rationale for individual partnered activities and the criteria for further pursuing a partnering arrangement or deciding to terminate one.
    - What levels of partnering are desirable for the Service?

- Gain-share identification and measurement are not universally understood and best practice is not captured.

- Partnering can reduce the "**intelligent customer"** position and the ability to be an informed "**decider**".

ANNEX E - RISK ANALYSIS

RISK APPROACH

The Risk Register has been populated below based on the initial Cause and Effect Analysis below in Figure 24. The headings for the Risk register are self-explanatory and commonly accepted within the Military organisations, for clarity the Probability, Impact and Danger-slope in Tables 9-11 have been added below. The Danger-slope table represents a numerical value based on a combination of relative Probability and Impact. This type of analysis provides assessors with the ability to identify the high-risk items from the Risk Register in Table 12 and apply resources as required to apply the Risk mitigation measures in Table 13.[24]

Probability definition	Abbreviation
Very likely	VL
Likely	L
Unlikely	U
Very unlikely	VU

Table 9 - Risk Probability Definition

Impact definition	Abbreviation
Life threatening	L
Project threatening	P
Expensive in time or cost	E
Some cost or time penalty	S
Negligible impact	N

Table 10 - Risk Impact Definitions

	Very likely	Likely	Unlikely	Very unlikely
Life threatening	8	7	6	5
Project threatening	7	6	5	4
Expensive in time or cost	6	5	4	3
Some cost or time penalty	5	4	3	2
Negligible impact	4	3	2	1

Table 11 - Risk Danger Slope

[24] Managing Software Quality and Business Risk – M Ould.

CAUSE AND EFFECT ANALYSIS

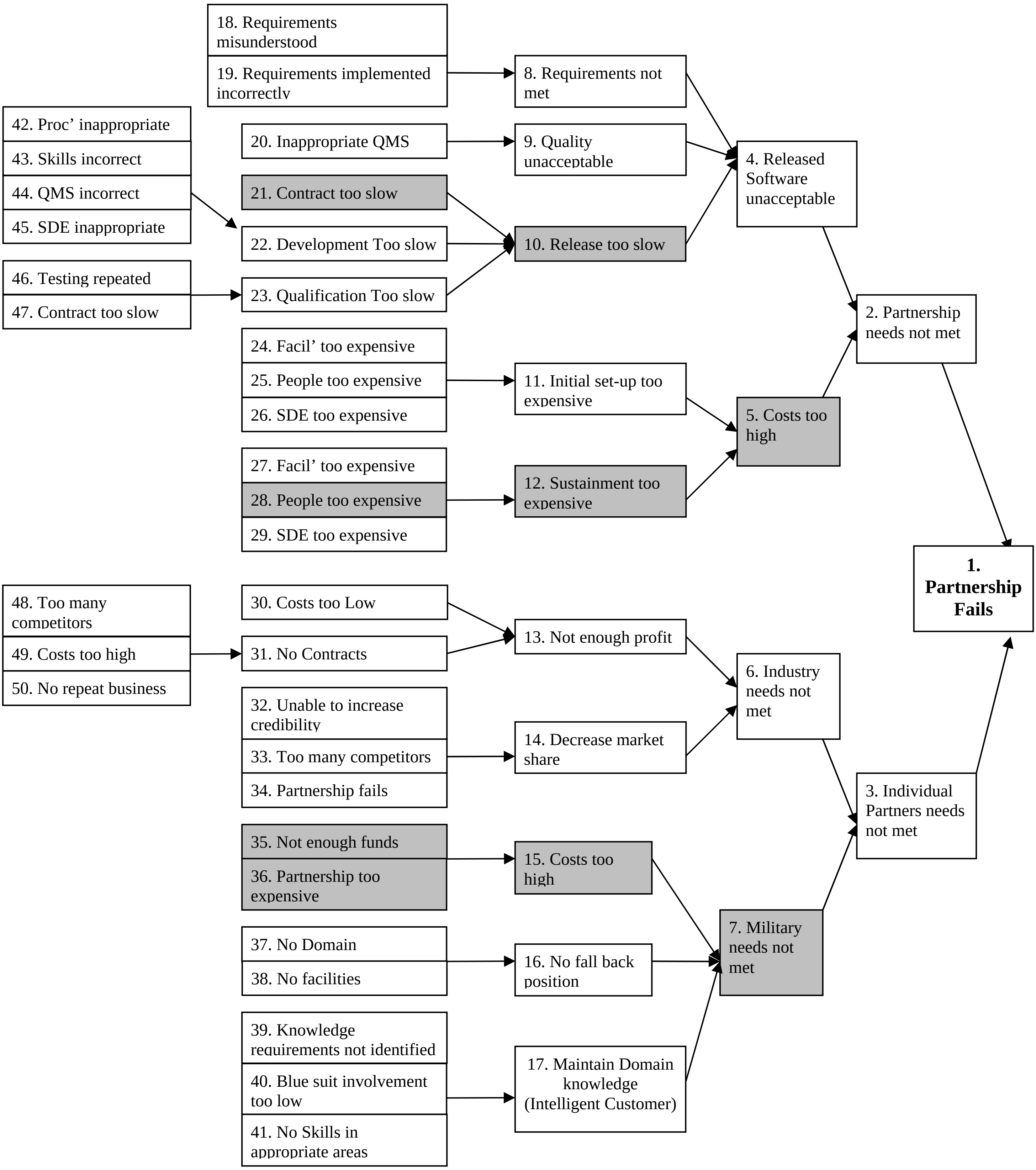

Figure 24 - Cause and Effect Analysis

No	Risk description	Causes	Sources of uncertainty	Nature of uncertainty	Probability	Impact	Risk owner
1	Partnership Fails		Event	It may happen that the partnership will fail	U	P	MIL
2	Partnership needs not met	1	Event	It may happen that the partnership needs will not be met	L	E	MIL
3	Individual Partners needs not met	1	Event	It may happen that the individual Military or Industry needs are not met	L	E	MIL
4	Released Software unacceptable	2	Event	It may happen that the released software is unacceptable to the partnership	L	E	MIL
5	Costs too high	2	Event	It may happen that the cost of the partnership will be too high	L	P	MIL
6	Industry needs not met	3	Event	It may happen that the Industrial needs are not met	U	E	IND
7	Military needs not met	3	Event	It may happen that the Military needs are not met	VL	E	MIL
8	Requirements not met	4	Event	It may happen that the software requirements are not met	VU	E	MIL
9	Quality unacceptable	4	Event	It may happen that the quality of the software is unacceptable	VU	S	MIL
10	Release too slow	4	Event	It may happen that the software will not be released in the correct timescales.	VL	E	MIL
11	Initial set-up too expensive	5	Event	It may happen that the initial costs of the partnership are too high	U	P	MIL
12	Sustainment too expensive	5	Event	It may happen that the sustainment costs for the partnership are too high	L	P	MIL
13	Not enough profit	6	Estimate	We are uncertain of the level of profit in the partnership	VU	P	IND
14	Decrease market share	6	Estimate	We are uncertain if the marker share will change	U	S	IND
15	Costs too high	7	Estimate	We are uncertain of the costs involved in the partnership	VL	P	MIL
16	No fall back position	7	Estimate	We are uncertain if a fall back position can be maintained	L	N	MIL
17	Maintain Domain knowledge (Intelligent Customer)	7	Event	It may happen that the Military Domain Knowledge will be unsustainable	U	E	MIL
18	Requirements misunderstood	8	Event	It may happen that the requirements will be misunderstood	VU	E	MIL
19	Requirements implemented incorrectly	8	Estimate	We are uncertain if we can achieve the partnership requirements	VU	E	MIL
20	Inappropriate QMS	9	Event	It may happen that the QMS does not meet the needs of the partnership	VU	S	MIL
21	Contract too slow	10	Event	It may happen that the contract for the partnership will prevent software development starting at the correct time	VL	E	MIL
22	Development Too slow	10	Estimate	We are uncertain if the development of software will be quick enough for the partnership	U	S	MIL
23	Qualification Too slow	10	Estimate	We are uncertain how long qualification will take	L	E	MIL
24	Facilities too expensive	11	Event	It may happen that the facilities are too expensive	U	P	MIL
25	People too expensive	11	Event	It may happen that the people cost id too expensive	U	P	MIL
26	.SDE too expensive	11	Event	It may happen that the cost of the SDE is too expensive	U	P	MIL
27	Facilities too expensive	12	Event	It may happen that the facilities are too expensive	U	P	MIL
28	People too expensive	12	Event	It may happen that the people cost id too expensive	L	P	MIL
29	SDE too expensive	12	Event	It may happen that the cost of the SDE is too expensive	U	P	MIL
30	Costs too Low	13	Event	It may happen that we charge too little to make a profit.	VU	P	IND

No	Risk description	Causes	Sources of uncertainty	Nature of uncertainty	Probability	Impact	Risk owner
31	No Contracts	13	Event	It may happen that there are no follow on contracts when the partnership is established.	U	P	IND
32	Unable to increase credibility	14	Estimate	We are uncertain if the partnership will lead to an increased level of credibility	U	N	IND
33	Too many competitors	14	Estimate	We are uncertain of the level of competition will decrease after the partnership is established	U	N	IND
34	Partnership fails	14	Event	It may happen that the partnership will fail	U	S	IND
35	Not enough funds	15	Event	In may happen that the Military are unable to secure the level of funding to maintain the partnership	VL	P	MIL
36	Partnership too expensive	15	Estimate	We are uncertain of the full cost of the partnership	VL	P	MIL
37	No Domain Knowledge	16	Estimate	We are unsure if we will be able to retain the required level of Domain Knowledge within the partnership	L	N	MIL
38	No facilities	16	Event	It may happen that the facilities for the fall back position will be unavailable	L	N	MIL
39	Knowledge requirements not identified	17	Event	It may happen that we do not correctly identify the Domain knowledge requirements	U	E	MIL
40	Blue suit involvement too low	17	Estimate	We are uncertain of the level of blue suit involvement required to maintain a credible amount of knowledge	L	E	MIL
41	No Skills in appropriate areas	17	Event	We are uncertain of the present Military skills or the required ones for the future	U	E	MIL
42	Processes inappropriate	22	Event	It may happen that the processes are inappropriate for the development of the software	U	S	MIL
43	Skills incorrect	22	Event	It may happen that the people skills are inappropriate for the development of the software	U	S	MIL
44	QMS incorrect	22	Event	It may happen that the QMS is inappropriate for the development of the software	U	S	MIL
45	SDE inappropriate	22	Event	It may happen that the SDE is inappropriate for the development of the software	U	S	MIL
46	Testing repeated	23	Estimate	We are uncertain of the level of testing that the qualification organisation will require	U	E	MIL
47	Contract too slow	23	Event	It may happen that the contract with the Independent Safety Authority will be too slow to set-up	L	E	MIL
48	Too many competitors	31	Event	It may happen that the level of competitors will not reduce, allowing for a monopoly	U	N	IND
49	Costs too high	31	Event	It may happen that we charge too much therefore making the partnership too expensive for the Military	L	S	IND
50	No repeat business	31	Event	It may happen that there will be no repeat business after the partnership has been initially formed.	VU	P	IND

Table 12 - Partnering Risk Register[25]

[25] Oxford University MRQ Project Headings – LB Cooper.

No	Risk description	Chosen Risk mitigation measure	Risk owner	Residual Risk	Best case value (Days)	Chosen case value (Days)	Worst case value (Days)
1	Partnership Fails	Manage all other risks	MIL	None	0	15	30
2	Partnership needs not met	Manage all partnership risks	MIL	None	0	10	20
3	Individual Partners needs not met	Manage all individual risks	MIL	None	0	10	15
4	Released Software unacceptable	Actively Monitor the development and sustainment costs and progress	MIL	Contract too slow	-5	5	15
5	Costs too high	Make sure that there are good cost estimation techniques in place and that costs are readily visible	MIL	Partnership too expensive	0	4	8
6	Industry needs not met	Compare profit against market share and project future levels	IND	None	0	3	6
7	Military needs not met	Monitor available funds against future funding requirements	MIL	Costs too high, loss Intelligent Customer position	10	30	50
8	Requirements not met	Use appropriate techniques to capture requirements and monitor compliance	MIL	None	0	5	10
9	Quality unacceptable	Implement an agreed QMS	MIL	None	0	2	4
10	Release too slow	Actively monitor development and project the release date against relevant milestones	MIL	Software not released in "Operationally significant" timescales	2	4	8
11	Initial set-up too expensive	Make costs visible to partners to allow cost-benefit analysis to be conducted.	MIL	None	0	2	4
12	Sustainment too expensive	Make costs visible to partners to allow cost-benefit analysis to be conducted against variations on costs through time	MIL	None	0	3	6
13	Not enough profit	Make costs visible to partners to allow cost-benefit analysis to be conducted	IND	None	0	1	2
14	Decrease market share	Project present market share against predicted	IND	None	0	1	2
15	Costs too high	Provide visibility of partnership costs and make sure that required funds are available at the correct time	MIL	Military cannot provide to support the partnership. Variable beyond IPT control.	10	30	50
16	No fall back position	Monitor and manage the level of Domain Knowledge and the cost/availability of facilities	MIL	None	0	3	6
17	Maintain Domain knowledge (Intelligent Customer)	Monitor and manage level of required blue suits and the actual people in post	MIL	Military cannot provide correct quantity or skilled personnel. Variable beyond the IPT control	2	8	14
18	Requirements misunderstood	Capture requirements through prototyping and use software best practice.	MIL	None	0	3	6
19	Requirements implemented incorrectly	Use an agreed QMS and Development practices	MIL	None	0	2	4
20	Inappropriate QMS	Use an agreed system	MIL	None	0	2	4
21	Contract too slow	Identify best practice and use an appropriate contracting method	MIL	Partners will not agree to chosen contracting method	5	25	45

No	Risk description	Chosen Risk mitigation measure	Risk owner	Residual Risk	Best case value (Days)	Chosen case value (Days)	Worst case value (Days)
22	Development Too slow	Use Best Practice and an appropriate QMS	MIL	None	0	2	4
23	Qualification Too slow	Bring Qualification organisation into partnership	MIL	ISA may not agree to chosen contracting method	5	10	15
24	Facilities too expensive	Identify duplicate or non-value organisations and rationalise were possible	MIL	None	0	2	4
25	People too expensive	Identify actual needs for skills and remove non-value organisations	MIL	None	0	1	2
26	SDE too expensive	Identify needs and rationalise where identified	MIL	None	0	1	2
27	Facilities too expensive	Identify duplicate or non-value organisations and rationalise were possible	MIL	None	0	1	2
28	People too expensive	Identify actual needs for skills and remove non-value organisations	MIL	None	0	1	2
29	SDE too expensive	Identify needs and rationalise where identified	MIL	None	0	1	2
30	Costs too Low	Identify the costs of maintaining the partnership and calculate the actual cost to the business	IND	None	0	3	6
31	No Contracts	Identify best practice and use an appropriate contracting method	IND	None	0	2	4
32	Unable to increase credibility	Fully commit to the partnership and only commit to what can actually be achieved	IND	None	0	2	4
33	Too many competitors	Increase Domain Knowledge and produce competitive costs	IND	None	0	1	2
34	Partnership fails	Manage all risks	IND	None	0	1	2
35	Not enough funds	Proactively bid for funds when required and create appropriate business cases with Risk contingencies built in.	MIL	The availability of funds is outside the control of the IPT.	5	20	35
36	Partnership too expensive	Monitor initial set-up costs and sustainment costs, add these to business case as needed.	MIL	Cost of partnership exceeds available Funds	2	4	6
37	No Domain Knowledge	Ensure knowledge requirements are captured and that resources are in place at the appropriate times	MIL	None	0	2	4
38	No facilities	Capture facility requirements and locate on Military sites	MIL	Locate facilities on a Military site	3	10	17
39	Knowledge requirements not identified	Capture requirements for knowledge early drawing from appropriate expertise	MIL	None	0	2	4
40	Blue suit involvement too low	Make sure that there are Blue Suits involved in every stage that is needed for an autonomous solution should the partnership fail.	MIL	Military cannot provide quantity or skilled personnel. Variable beyond the IPT control	5	20	35
41	No Skills in appropriate areas	Make sure that the appropriate areas are identified and populated with people.	MIL	None	0	2	4
42	Processes inappropriate	Use Best Practice	MIL	None	0	2	4
43	Skills incorrect	Identify partnership skill requirements	MIL	None	0	2	4
44	QMS incorrect	Implement an agreed QMS	MIL	None	0	1	2

No	Risk description	Chosen Risk mitigation measure	Risk owner	Residual Risk	Best case value (Days)	Chosen case value (Days)	Worst case value (Days)
45	SDE inappropriate	Identify an appropriate SDE for the software	MIL	None	0	1	2
46	Testing repeated	Get ISA involved early on the testing activities to reduce repetition	MIL	None	0	5	10
47	Contract too slow	Identify best practice and use an appropriate contracting method	MIL	Partners will not agree to contracting method	5	10	15
48	Too many competitors	Manage costs and knowledge	IND	None	0	1	2
49	Costs too high	Manage costs	IND	None	0	2	4
50	No repeat business	Manage costs and knowledge	IND	None	0	4	8

Table 13 - Residual Partnering Risk Register

ANNEX F - ASSUMPTIONS

During the composition of this dissertation there were a number of assumptions made. There have been captured below for consideration. These will need to be confirmed at an appropriate time during the satisfaction of a Partnering solution, by the relevant Stakeholders.

No	Assumptions
1	That the MoD will follow the Governments' directive and form partnerships with their industrial counterparts.
2	That the MoD can secure sufficient funds to change from the traditional support solutions to a partnered software solution.
3	To maintain the Intelligent Customer position, Military personnel will need to be seeded throughout and resultant organisation. It is assumed that the present drawdown in Military personnel will not prevent this.
4	That Industry are willing to commit to long-term partnering agreements

Table 14 - Partnering Assumptions

BACK MATTER

ABBREVIATIONS

Abbreviation	Interpretation
CDL	Chief of Defence Logistics
CMMi	Capability Maturity Model Integrated
COCOMO	Constructive Cost Model
DA	Design Authority
DE&S	Defence Equipment & Support
DEC	Director Equipment Capability
DLO	Defence Logistics Organisation
DPA	Defence Procurement Agency
EW	Electronic Warfare
FLC	Front Line Command
FOAS	Future Offensive Air System
FSR	Field service Representative
FTA	Formal Task Agreement
HUD	Head-up Display
ILS	Integrated Lifecycle Support
IPR	Intellectual Property Rights
IPT	Integrated Project Team
IPTL	Integrated Project Team Leader
ISA	Independent Safety Auditor
ISADS	Is-Service Avionics Design Study
ISD	In-Service Date
IT	Information Technology
JSF	Joint Strike Fighter
JSP	Joint Service Publication
LCC	Lifecycle Costs
LSA	Logistics Support Analysis
MDA	Model Driven Architecture
MoD	Ministry of Defence
MR2	Maritime Recognisance Mark 2
MTBF	Mean Time Between Failure
NAO	National Audit Office
NEC	Networked Enabled capability
OEM	Original Equipment Manufacturer
OFP	Operational Flight Program
OOD	Object Oriented Designed
PPP	Public Private Partnership
QMS	Quality Management System
RAAF	Royal Australian Air Force
RAF	Royal Air Force
RFC	Request For change
ROI	Return on Investment
SAE	Society of Automobile Engineers
SCMB	Software Configuration Management Board
SCR	Software Change Request
SDE	Software development Environment
SLA	Service Level Agreement
SME	Subject Matter Expert
SOS	Software Operational Support

SSC	Software Support Cell
SST	Software Support Team
Strike	Strike Command
TLCMP	Through Life Capability Management Plan
UAV	Unmanned Air Vehicle
UML	Universal Modelling Language
UOR	Urgent Operational Requirement
USSR	Union of Soviet Socialist Republics
UXB	Unexploded Bomb
WLC	Whole Life Costs

Table 15 - Abbreviations

BIBLIOGRAPHY

BAE Systems Software Support Strategy - Introduction, Cooper 2004.
DLO Study into Software Support Teams - 2003.
Future Offensive Air System - Software Study (Brooks/Usher/Cooper) - 2004.
Logistics Support Analysis Guidance for Software – Bailey 2003.
Maturing the Software Logistics Support Analysis Process – Brooks 2005.
MOD Policy Paper 4 - 2001.
MOD Policy Paper 5 - 2002.
Software Book of Knowledge, IEEE Computer Society - 2004.
UK Defence Procurement Policy - 2003.

REFERENCES

[AMS06/1] http://www.ams.mod.uk/ams/content/docs/toolkit/ams/policy/other_guidance/partnering/summary.htm, 25 Jul 06.
[AMS06/2] http://www.ams.mod.uk/ams/content/topics/topic.htm, Jul 06.
[AMS07] http://www.ams.dii.r.mil.uk/default.htm 15 Jun 07.
[BED95] Planning for Change – A personal view on Introducing software to service in the Royal Air Force, Sqn Ldr Baz Beddoes, May 95.
[BE/RA02] Software Maintenance and Evolution: a Roadmap, 2002.
[BEN97] A New perspective on Software evolution, Bennett, 1999.
[BOE81] Software Engineering Economics. B Boehm, 1981.
[CUS97] Microsoft Secrets, Cusumano & Selby, 1997.
[DLO01] D/DLOHQ/15/535/5/4, DLO MCS Report 32/00 28 Feb 2001.
[DLO02] Defence Logistics Organisation, 2002, ES(Air)(Wyt)SS/2306/01/AD Sys, J Brooks, M Usher, LB Cooper.
[DLO03] ES(Air)/ADAS(ABW)/25/1 dated 31 Mar 2003.
[DLO04] AVM AJ Smith, End-to-End Air and Land dated 01 Jul 2003.
[DLO97] STC/155247/11 Wg Cdr Hyde dated 25 Mar 1997.
[DLO98] LC/165118/1/7 Air Cmdr R Brumpton dated 7 Jan 1998.
[IPT05] Information from IPT commercial – Due to commercial reasons the IPT cannot be referenced.
[ISO/IEC06] Software Engineering Software life cycle processes – Maintenance BS ISO/IES 14764:2006.

[JAP100A] Joint Air Publication 100A-01 Chapter 12.8.
[JAP100D] JAP100D-10, 2004, Royal Air Force.
[M880-02] M880-2002 – Open University course in Software Engineering.
[MCD94] McDermid, J.A. (1994), Software Engineer's Reference Book, Butterworth-Heinemann Ltd.
[MOA90] Maintaining The Competitive Edge", Moad J, DATAMATION , 1990.
[NAO03] NAO Report 2003, Office of Statistics.
[PFL01] S.L Pflegger, Software Engineering: Theory and Practice, second Edition, 2001.
[PIG97] Practical Software Maintenance – Best Practices for Managing Your Software Investment", Thomas M. Pigoski, John Wiley & Sons, Chichester, 1997.
[SDR99] Strategic Defence Review, 1999.
[SEI98] SAE JA 1004, Software Supportability Program Standard, Jul 1998 – 6.2 Meeting Customer Requirements.
[TWG97] The Role of Supplier "Guest Engineer" in the Customers Produce Team, David Twigg, 1997.
[UKDS05/1] UK Defence Statistics 2005 – Finance.
[UKDS05/2] UK Defence Statistics 2005 – MOD Resource by Department.
[UKDS05/3] UK Defence Statistics 2005 – Defence Contracts.