INCOSE UK Chapter
Working Group on
Applying Systems Engineering to In-Service Systems
Supplementary Guidance to the
INCOSE Systems Engineering Handbook

The views expressed by the authors in this report do not necessarily represent the views of their employers.

ABBREVIATIONS

The following abbreviations are used within this report with the meanings described below:

CM	Configuration Management
DRACAS	Data Reporting, Analysis and Corrective Action System
FRACAS	Fault Reporting Analysis and Corrective Action System
ILS	Integrated Logistic Support
IM	Information Management
INCOSE	International Council on System Engineering
ISO	International Standards Organisation
ISSE	In-Service Systems Engineering
ISSWG	In-Service Systems Working Group
IT	Information Technology
ITIL	Information Technology Infrastructure Library
LRU	Line Replacement Unit
MoD	Ministry of Defence (UK)
MoE	Measures of Effectiveness
MoP	Measures of Performance
SE	Systems Engineering
UK	United Kingdom
V&V	Validation and Verification

GLOSSARY

The following terms are used within this report with the meanings described below:

Architectural Design	The synthesis of a system architecture baseline that is consistent with the requirements.
In-Service System Engineering	Systems engineering concerned with the sustainment of one or more systems that have entered service.
Measure of Effectiveness	A measure of the effectiveness of a service Measures of Effectiveness are the “operational” measures of success that are closely related to the achievement of the mission or operational objective being evaluated, in the intended operational environment under a specified set of conditions.
Measure of Performance	A measure of the performance of a system Measures of Performance define the key performance characteristics the system should have when fielded and operated in its intended operating environment.
Product System	The system that directly solves the underlying system needs.
Support System	The system that supports the Product System.
Requirements Development	To produce and analyse customer, product, and product component and work products [www.sei.cmu.edu/cmmi].
Requirements Management	To manage the requirements of the project’s products and product components and identify inconsistencies between those requirements and the project’s plans [www.sei.cmu.edu/cmmi].
System Architecture	The selection of the types of system elements, their characteristics, and their arrangement.
Systems Engineering	An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (From the SE Handbook).
Update	The act or process of bringing a system up to the current standard – roughly analogous to the term ‘Renew’ used by the rail industry.
Upgrade	The act or process of changing a system to bring it up to a higher standard (of performance) - roughly analogous to the term ‘Enhance’ used by the rail industry.
Upkeep	The act or process of keeping a system in good repair, especially over a long period – analogous to the term ‘Maintain’ used by (among others) the rail industry.
Validation	Confirmation that “we are building and have built the right thing”.
Verification	Confirmation that “we are building and have built the thing right”.

REFERENCE DOCUMENTS

[1]	INCOSE-TP-2003-002-03.1	INCOSE Systems Engineering Handbook	V3.1	August 2007
[2]	is-tr-001	INCOSE UK Chapter Working Group on Applying Systems Engineering to In-Service Systems: Final Report,	Issue 1.0	31 October 2008
[3]	ISBN 10 0-470-02554-9	Checkland, P.B. & Poulter, J. (2006) Learning for Action: A short definitive account of Soft Systems Methodology and its use for Practitioners, teachers and Students, Wiley, Chichester.		
[4]	*To be published in* the Proceedings of the 20th INCOSE International Symposium, 2010, Chicago	Duncan Kemp. So what is “in service systems engineering”?		
[5]	The introduction to the ITIL Service Lifecycle Book	IT Service Management, ISBN 9780113310616		August 2001
[6]	www.nspe.org/resources/pdfs/.../DocRetentionGuidelines0305final.pdf	Document Retention Guidelines		
[7]	Int. J. Product Lifecycle Management, Vol. 1, No. 1, 2005 Chris McMahon, Matt Giess and Steve Culley	“Information management for through life product support: the curation of digital engineering data",		
[8]	ISO 10303-239	Product lifecycle support		
[9]	ISO 10303-233	SE data exchange		
[10]	Def-Stan 21-59	Systems Engineering Guide for Naval Combat Systems - Section 5 in-service support although naval specific contains SE processes for modifications to in-service equipment		
[11]	Def Stan 00-79 Part 1	Provides a vocabulary for terms associated with the concept and application of Technology Insertion		
[12]	Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Volume 222, Number 8, 2008	Technology insertion in the defence industry: a primer, C I V Kerr, R Phaal, and D R Probert,		
[13]	www.aof.mod.uk	MoD Acquisition Operating Framework guidance – Support Solutions Envelope - Defines considerations for acquisition and integration of complex systems together with effective logistics support		
[14]	http://www.scnus.org/page.aspx?ID=101236 http://www.yourwindow.to/business-continuity/index.htm	Security Community Network		

[15]	ISBN 978-0-470-71240-5	Alexander, I & Beus-Dukic, L, “Discovering Requirements, How to Specify Products and Services”, Wiley, Chichester.	2009
[16]	ISO 10007	Quality Management - Guidelines for configuration management	2003

CONTENTS

.... 1
INCOSE UK Chapter.. 1
Working Group on.. 1
Applying Systems Engineering to In-Service Systems.. 1
Supplementary Guidance to the.. 1
INCOSE Systems Engineering Handbook.. 1
ABBREVIATIONS.. 2
GLOSSARY.. 3
REFERENCE DOCUMENTS.. 4
CONTENTS.. 6
1 Introduction.. 8
1.1 Purpose.. 8
1.2 Structure of this Document.. 8
2 Why Apply Systems Engineering to In-Service Systems?.. 9
2.1 Introduction to Different Perspectives on ISSE.. 10
2.2 Managing the System.. 12
2.3 Changing the System.. 13
2.4 Delivering the Service.. 13
2.5 Optimising the Supply Chain.. 14
3 How to Apply Systems Engineering to In-Service Systems.. 15
3.1 Introduction.. 15
3.2 Requirements, Validation and Verification.. 15
3.2.1 Handbook Processes Affected.. 15
3.2.2 General Guidance.. 15
3.2.3 Specific Guidance for Managing the System.. 17
3.2.4 Specific Guidance for Changing the System.. 18
3.2.5 Specific Guidance for Delivering the Service.. 19
3.2.6 Specific Guidance for Optimising the Supply Chain.. 20
3.3 Architectural Design.. 20
3.3.1 Handbook Processes Affected.. 20
3.3.2 General Guidance.. 21
3.3.3 What is the Current Architecture?.. 22
3.3.4 How Much Should the Architecture be Changed?.. 23
3.3.5 How is a Service Maintained while Changing the Architecture?.. 23
3.3.6 Specific Guidance for Managing the System.. 23
3.3.7 Specific Guidance for Changing the System.. 24
3.3.8 Specific Guidance for Delivering the Service.. 24
3.3.9 Specific Guidance for Optimising the Supply Chain.. 24

3.4 Implementation and Transition.....25
3.4.1 Handbook Processes Affected.....25
3.4.2 General Guidance.....25
3.4.3 Specific Guidance for Managing the System.....26
3.4.4 Specific Guidance for Changing the System.....27
3.4.5 Specific Guidance for Delivering the Service.....27
3.4.6 Specific Guidance for Optimising the Supply Chain.....28
3.5 Information and Configuration Management.....28
3.5.1 Handbook processes affected.....28
3.5.2 General Guidance.....28
3.5.3 Specific Guidance for Managing the System.....30
3.5.4 Specific Guidance for Changing the System.....30
3.5.5 Specific Guidance for Delivering the Service.....31
3.5.6 Specific Guidance for Optimising the Supply Chain.....31
4 Conclusions.....32
Annex A. Terms of Reference for the UK IN-SERVICE SYSTEMS WORKING GROUP.....33
Annex B. Acknowledgements and contributors.....34

1 Introduction

This document forms the output of the second UK Working Group on **In-Service System Engineering (ISSE**). It provides advice on how to perform System Engineering (SE) for systems that have entered service.

1.1 Purpose

The purpose of this document is to provide supplementary guidance to augment the guidance in the INCOSE SE Handbook , referred to as “the Handbook” below. This report implements some of the recommendations in a previous report .

The document explains why there is a need to review SE advice for in-service systems and explains why ISSE should examine four key perspectives developed from a Soft Systems Methodology approach to the problem. It goes on to give practical advice for implementation of ISSE in the identified areas.

Annex A records some of the historical background to the work and Annex B lists the contributors to it.

1.2 Structure of this Document

This document is presented in four main parts:

- Section 1 covers the introduction and purpose of the document.
- Section 2, ‘Why’ describes the four perspectives on ISSE:
  - Managing the system;
  - Changing the system;
  - Delivering the service; and
  - Optimising the supply chain.

  and explains why they are needed.
- Section 3, ‘How’ provides the guidance under four areas of concern:
  - Requirements, Validation and Verification;
  - Architectural Design;
  - Implementation and Transition; and
  - Information and Configuration Management.

  The guidance for each area of concern is structured according to the four perspectives.
- Section 4 presents conclusions.

2 Why Apply Systems Engineering to In-Service Systems?

Traditional SE concentrates on the early life cycle stages. There is good reason to believe that the potential return on the investment of SE is greatest in these early stages. The figure below, taken from the Handbook , illustrates the expected return.

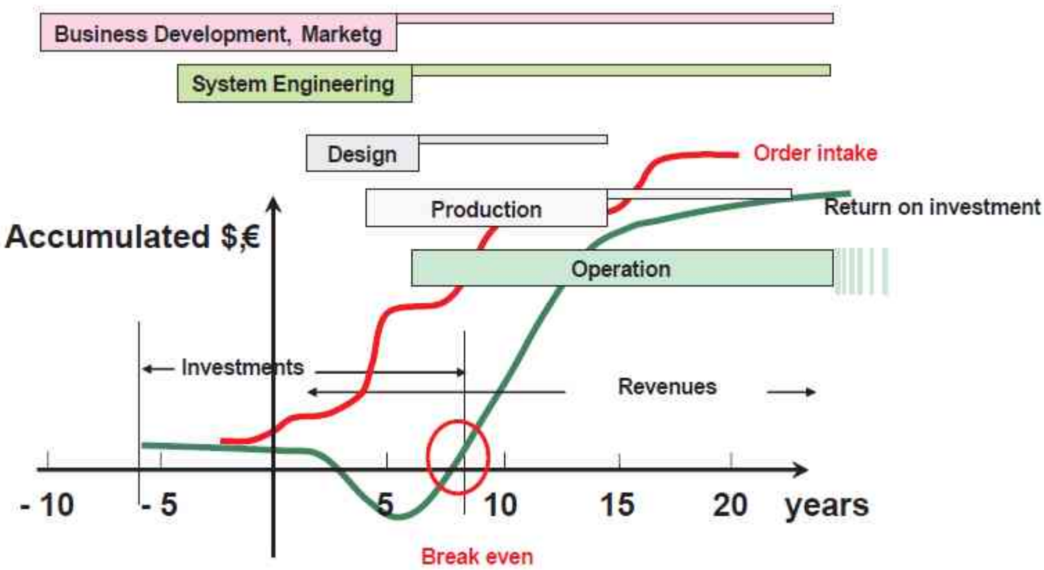

Figure 2-1: Generic business life cycle from the Handbook

This does not imply that SE is a low-value activity during service. It remains important for the following reasons:

- If those building the system have invested in good SE practice then it is essential to keep employing good SE practice during service in order to protect that investment – otherwise the intellectual 'grip' that has been obtained on the system will loosen over time.
- If those building the system have not invested in good SE practice during development then investing in good SE practice during the long service life becomes more important in order to 'get a grip' on the system. Sadly, this is the more common case in the authors' experience.
- The in-service stage of the system life cycle is usually longer than any other stage and implications of decision on SE will accumulate over this period.

Traditional SE guidance is largely written for those realizing new systems. The authors have found nothing to suggest that the principles of SE vary across the system life cycle. However the authors find that some adjustment to practices that work well for developing new systems is required when working on systems that have entered service, for a number of reasons:

- The existing system and its environment may provide significant constraints on how the system may be changed.

  New systems may be highly-constrained as well. We sometimes distinguish 'brown-field' systems that must be designed to fit within an existing context from less-constrained 'green field' systems. With this terminology, in-service systems are always 'brown field' systems and often so far to the extreme end of this spectrum that respecting constraints may be a bigger consideration in the specification and design of a change to an in-service system than the objectives of the change.

- The management and use of in-service systems does not fall within the project/programme management paradigm assumed by most systems engineering standards and textbooks. Rather in-service systems are managed using asset, operations or service management principles.

  This adds a language problem – the same concept may be named differently and different concepts may have the same name.

  Operations management has a different balance between reaction and planning. A 'sense and react' paradigm can often be more effective and efficient than a 'plan and execute' one.

- Determining the system of interest can be harder than in a new procurement.

This is partly inherent. In-service systems are often part of other systems. For example a radio system can be part of a train, a signalling system, a passenger information system and a rail service.

2.1 Introduction to Different Perspectives on ISSE

When a system enters service, there are typically a number of groups of people involved with the system who have different interests in it. The authors have used Peter Checkland's Soft Systems Methodology to identify four principal perspectives on an in-service system .

- **Managing the System –** maintaining or improving system performance. The phrase, **Measure of Performance** (**MoP)** is often used for measures of the technical performance of a system. The people with this perspective will be interested in keeping the delivered MoPs in line with requirements. They may work for a department with "maintenance" or "asset management" in its title.
- **Changing the System –** updating or upgrading the system in response to changing needs and circumstances. (A project is normally commissioned to do this, which will have its own life cycle.) If, as is traditional, this life cycle is pictured as a "V", then the updates and upgrades may be pictured as a series of small "V"s following the large "V" that represents the creation of the system, as pictured below.

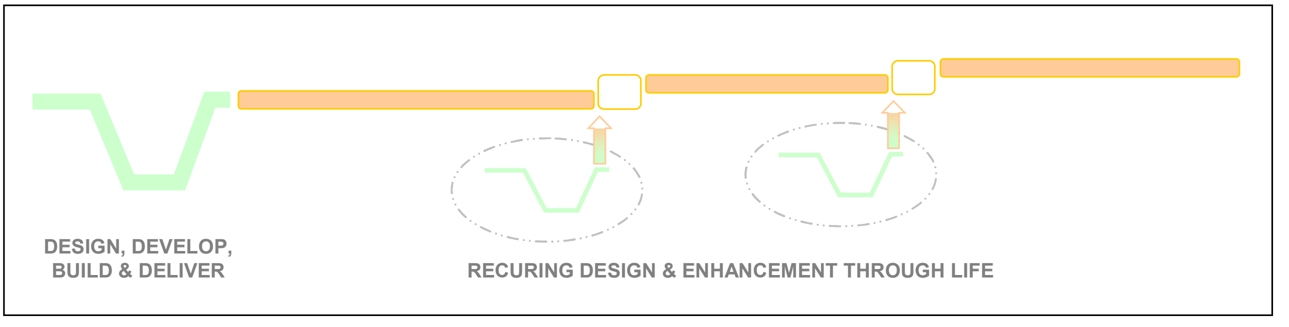

Figure 2-2: Multiple instances of the "V" life cycle.

The real situation is generally more complex than the figure indicates: components of the system based upon different technologies may be updated with very different frequencies and the "V"s may, in fact, overlap.

The people with this perspective will often be interested in enhancing the system's MoPs.

- **Delivering the Service** – there will normally be a department that uses the system in question, and possibly other systems, to deliver a service which advances the business objectives of the organisation that owns the system. The people with this perspective will normally be responsible for designing the manner in which the service is delivered and for adjusting this to optimise business/operational effectiveness. The phrase, **Measure of Effectiveness** (**MoE**) is often used for measures of service delivery. The people with this perspective will be interested in keeping the delivered MoEs in line with requirements.
- **Optimising the Supply Chain** - designing the right supply network to deliver effective support to the system in question at an affordable cost. Adjustments in this area may reduce cost of ownership and improve system availability.

It may be noted that there may be several 'Changing the system' viewpoints, each associated with a change project for the system, whereas the other viewpoints exist continuously through the service life of the system.

Each of these viewpoints is concerned with different but overlapping system boundaries. From these systems, we find it useful to highlight two:

- the **product** system that performs the desired function; and
- the **support** system (depots, docks, tools, maintenance procedures and so on) that maintains the product system.

These systems must be created in parallel and will evolve in parallel, as the figure below illustrates.

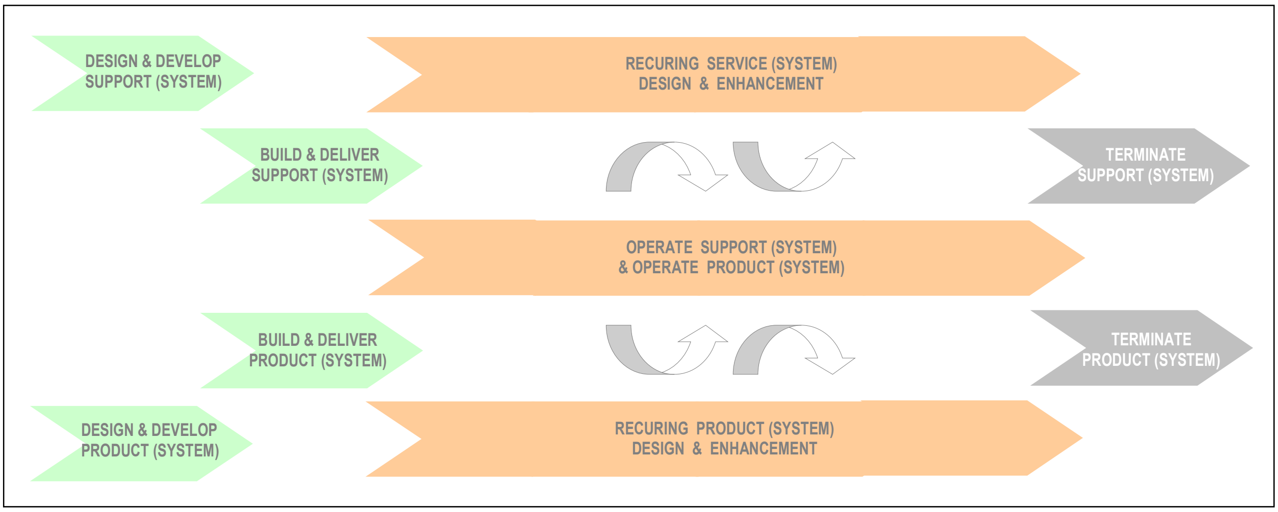

Figure 2-3: The Support System and the Product System through life

2.2 Managing the System

Once in-service, system performance must be continually monitored to check whether it continues to satisfy stakeholder needs. Where it does not, update or upgrade action may be necessary.

The gap between stakeholders' needs and performance may widen because the needs increase or performance falls, as illustrated below.

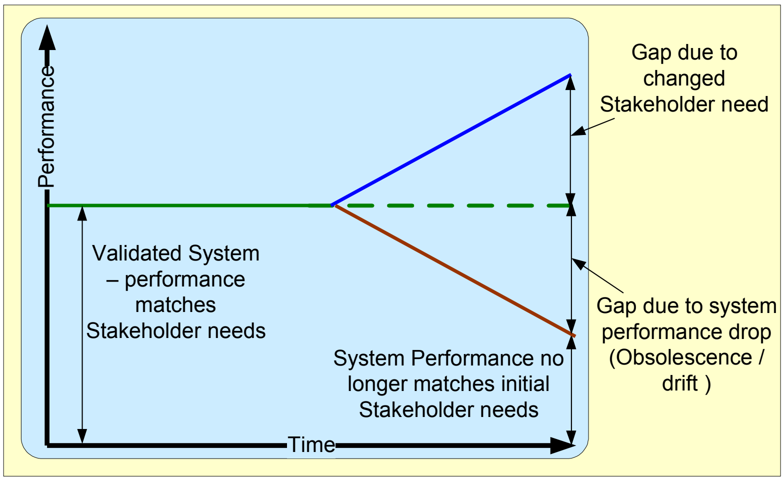

Figure 2-4: Gaps between performance and needs (part 1)

There may be a gap because a system may have been assessed as 'good enough to enter service' at acceptance but yet not meet all the current stakeholder needs. This assessment may be driven by the technical solution or by funding limitations – as shown in Figure 2-4 below. On the other hand, in order to better respond to perceived future requirements, the same drivers can lead to systems being delivered with performance above the current need.

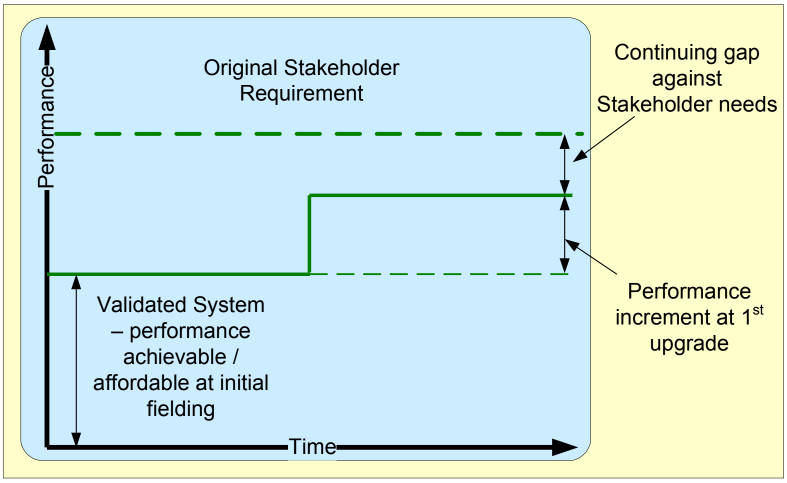

Figure 2-5: Gaps between performance and needs (part 2)

Managing the system can be split into four main activity groups as defined below:

- The management of the gap between the design intent and the material state of the system.
- The commissioning, oversight and acceptance of upgrade (new capability), update (replacing obsolete sub-systems) and upkeep (returning to safe and capable level). This involves concept exploration, requirements management and may also involve packaging several different upgrade, update and upkeep tasks into a major programme.
- The management of supplies – making sure that replacement parts (which could be complex engineered products or simple consumable items) are available when needed.

- The management of information. In theory this is a simple task. Coupling modern Information Technology (IT), support data sets and support processes will ensure information is captured once and used by everyone who needs it. In practice developing an equipment support information system is a complex, risky and expensive systems engineering task in itself!

2.3 Changing the System

Changes to the system fall, in the main, into three types:

- routine changes carried out as part of maintenance **(Upkeep),** for example replacing failed components,
- obsolescence-driven changes (**Updates**); and
- system enhancements to deliver better performance or better maintainability (**Upgrades**).

There will normally be a business sponsor for changes monitoring the cost and time aspects and a design authority monitoring technical performance.

To most traditional systems engineers this is the most straightforward of the four perspectives as it uses the project / programme management paradigm that they are familiar with. It follows the traditional SE life cycle (from the Handbook):

- **Concept** – This stage involves clarifying stakeholders' needs and proposing viable solutions. There may be a need to survey the existing system, if information about it is incomplete.
- **Development** - This is similar to the development of a new system but with three key differences:
  - the need to embody the change in the in-service systems;
  - the need for a 'make, buy or reuse' decision;
  - the need to undertake regression testing on the functions and performance of the system not being improved to ensure that they are not compromised.
- **Production** - This is similar to the production of a new system but regression testing is required and the implementation may be made more complex by the need to continue to deliver a service while the system is changed. This phase may include a complex transition process, as the change is introduced into the operational system.

An upgrade or update may also involve the **Retirement** of system elements being replaced.

The standard SE process can be applied but will require tailoring because the project is changing an existing architecture rather than creating a new one and because the need to keep the system in service places constraints on the project. For example, if the opportunities to change the system are infrequent and the changes being made must be introduced at a specific opportunity, the programme may be run under a 'fixed time - variable scope' rather than 'fixed scope – variable time' paradigm.

2.4 Delivering the Service

Systems rarely deliver benefit in their own right. Instead they are used to deliver services. For example:

- In transport: Aircraft are used by airlines to deliver flights to passengers. Integrated rail systems are used to deliver rail journeys to passengers.
- In defence: Unmanned Air Vehicles are used to provide surveillance services to commanders. Destroyers are used to deliver air defence services to commanders.
- In commerce: radio frequency identification tagging is used as a form of revenue protection for the organisation, and to help optimise supply chain operation.

Changing the way a system is used to deliver a service can deliver significant improvements in performance with minimal costs.

This perspective involves using the technical product/system as part of a business or operational system. It comprises five concurrent processes (from ITIL information service management):

- **Service strategy** – determining the services that will be delivered. This includes defining the functions, effectiveness and performance of the services.
- **Service design** – designing the service to deliver the agreed functions, performance and effectiveness.

- **Service transition** – introducing new components and services, including the necessary Verification and Validation.
- **Service operation** – delivering services to customers and recovering from service failures.
- **Continual service improvement** – improving all elements of the service based upon feedback.

This perspective is a mixture of the programme and operations management paradigm. It also operates concurrently over several different life cycles:

- The strategy life cycle – where changes to strategy are agreed, new services introduced into service and the impact on the business observed.
- The individual service development life cycles, where a new/upgraded service is defined and designed and new/upgraded (sub-)systems are procured and put into service.
- The individual (sub-) system life cycles.

This perspective requires the application of systems engineering to the business services that need to be delivered.

2.5 Optimising the Supply Chain

Section 9.0 of the Handbook (Specialty Engineering) contains most references to the supply chain, and uses the term Integrated Logistic Support (ILS) to collectively refer to the supply chain and a host of other related terms. In the main, the guidance provided is applicable to the design stages, and does not explicitly refer to the in-service stages of system deployment.

The sustained delivery of effective service using an in-service system is dependent on the supply chain supporting the system and the costs of sustaining the system depend upon the efficiency of this supply chain.

Cost savings and service improvements may be achieved by the three perspectives mentioned in the previous sections through:

- changes in maintenance practices (Managing the System);
- improvements to the design of the systems (Changing the System); and
- improvements to the service design (Delivering the Service).

  They may also be achieved by improving the efficiency of the supply network, through understanding and optimising:
- the cross-organisational value chains that underpin the supply chain; and
- the performance measurement, management, and incentivisation regimes used to drive the supply chain.

The people taking this perspective also need to:

- Determine the commercial approach to be adopted for the through life management of sub-systems and overall systems integration. Options range from the highly competitive to the highly collaborative (such as partnering or forming alliances).
- Monitor and react to obsolescence of system components.

3 How to Apply Systems Engineering to In-Service Systems

3.1 Introduction

Imagine two competent engineers: Engineer Green and Engineer Brown.

Case 1. Engineer Green sees an opportunity to apply the results of some recent research to produce a new system (product and/or service) and plans how to design and develop the system. Green describes the system to Engineer Brown. Brown sees in it the possibility of integrating the new system to achieve a better and cheaper solution to in a wider system to a problem.

Case 2. Engineer Brown has some problems sustaining the current system (product and/or service) to continue to meet the required capability and foresees cost, schedule, and performance criteria shortfalls. Brown describes this problem to Engineer Green. Green sees a solution in the application of some recent research and considers how to design and develop a solution.

In both cases ' … knowledge will be applied to a practical purpose … ' one from the system design viewpoint the other from the service delivery viewpoint.

These two engineers need to work as one to realise the new solution and its benefits. The issue here is that the eventual transition into the in-service stage of the new and termination of the old system prompts the engineers to think beyond 'green-field' engineering and become aware that the system must be placed into, and perform in, an existing environment, the 'brown field'.

The transition from 'green-field' engineering to 'brown-field' is more a change of philosophies than just a specific set of technologies. The key to adapting the guidance of the Handbook to in-service systems is to think 'brown-field'

The UK Ministry of Defence Acquisition Operating Framework (www.aof.mod.uk) offers some guidance on introducing a new system, product or service into a 'brown-field' environment. It advises that it is necessary to assess the impact of a new system, product or service on the 'defence lines of development': Training, Equipment, Personnel, Information, Concepts & Doctrine, Organisation, Infrastructure, Logistics and Interoperability Lines of Development.

3.2 Requirements, Validation and Verification

3.2.1 Handbook Processes Affected

- 4.2: Stakeholder Requirements Definition Process.
- 4.3: Requirements Analysis Process
- 4.7: Verification Process
- 4.9: Validation Process

3.2.2 General Guidance

3.2.2.1 Explanation of Terms

We define the terms **Validation** and **Verification** as follows:

- Validation: Confirmation that "we are building and have built the right thing".
- Verification: Confirmation that "we are building and have built the thing right".

The terms can also be applied to changing an in-service system: upgrades, improvements, and so forth. Validation and verification techniques therefore need to be applied during the engineering of these activities.

We should note that the definitions we have given above should be thought of as a **convention**. Some people/organisations reverse the allocation of these definitions to the two V words. This is not inherently wrong, it is just that they are using a different convention. Others talk about V&V, without a clear separation of meaning between the two, which is unhelpful. Still others would prefer to do away with the V words altogether (because of their potential ambiguity) and introduce other words with a clearer meaning. In any event, on a real project one is well-advised to agree these terms and their meanings with all stakeholders from the outset.

For the purposes of this document, let us accept the definitions above and expand them as follows:

Validation: Confirmation that "we are building and have built the right thing"

- Are the system requirements: known, understood, agreed?
- Are the system requirements the right ones?
- How do we know the system requirements are the right ones?
- Which system solution is the best one?

Ultimately, of course, we have to recognise that the acid test of whether we have built the right system comes when it is in service and the system's beneficiaries (that is those stakeholders who benefit in some way from the system and the services it provides) are pleased with what they have got (or not, as the case may be).

Verification: Confirmation that "we are building and have built the thing right"

- Is the developing/developed system compliant with the system's specification?
- How do we know? What will we measure?
- Have we got the means (that is, stage gate reviews, models, test plans, trial plans, acceptance plans, certification plans, test equipment, staff, budget, timescale, etc) to answer this question? How will we measure?

Note that verifying a system against its requirements is not sufficient to show that we have built the right system. If the requirements are wrong, we will have built the wrong system right!

3.2.2.2 Responsibility for Validation

This section deals with the potentially problematic issue of who is responsible for determining what the system's requirements should be. Note that this applies both to new systems and in-service ones.

If we are the ones identifying and specifying the system requirements, validation means:

- Using checklists based on our experience to ensure all relevant requirements categories have been considered (to minimise the chances of omitting any).
- Feeding results of the requirements identification process back to the stakeholders for agreement (or not) and clarification of ill-defined, ambiguous and conflicting aspects.
- Using suitable methods to communicate and obtain agreement with the stakeholders. These may include: scenarios and use cases, system architecture diagrams, demonstrators, prototypes, computer-generated visualisations and models.
- If the implications of getting the requirements wrong are very serious and the project has a feel of unease about how well-founded the requirements are, we must also modify the project life cycle by introducing risk-reducing stages with names such as 'concept exploration', 'proof of concept', 'user demonstration', 'option selection' and so forth.
- After fielding the system, putting in place a mechanism to collect, analyse and decide upon any discrepancies between the system's behaviour and performance as built/fielded and the expectations of the beneficiary stakeholders (those that emerged as the project progressed and/or were not captured in the requirements specification / data set). That is, in the final analysis, despite all our hard efforts to capture and manage the system requirements, did we get any wrong, miss any or not adequately control change, and if so what were they?

If we are **not** the ones primarily responsible for identifying and specifying the system requirements (for example because we are a contractor), and the system requirements are handed to us as a 'given', the question arises whether we have to validate the requirements, that is, check that they are the right ones.

This is largely a contractual issue (or at any rate has been a contractual issue). In earlier days, contractors took the view that they could hide behind whatever the contract stipulated and just do what they were contracted to do. In other words, they did not see it as their responsibility to check that the requirements were the right ones; that was the responsibility of whoever they were contracted with. They would, however, have to check the contract's small print in case there were contractually-binding words that assigned to them the responsibility of ensuring that their part of the system was 'fit for purpose', for example.

More recently, the trend has been for those awarding contracts to pass down this responsibility to the contractor.

So, care is needed with validation. Examine your contract carefully! Even if you think you can hide behind the contract, it is still worth asking whether the requirements make sense, or whether there are significant omissions or ambiguity (and this activity is a form of validation). In this regard, ask yourself how badly your reputation is likely to suffer if you are associated with a dysfunctional systems development project and your only line of defence is “we were just doing what the contract said”.

3.2.2.3 Applying Validation and Verification to ISSE

Summing up the implications of these definitions for ISSE: in the general case, through-life validation means that we need to ask the following question on an on-going basis: “are the requirements that the system embodies still the right ones, or have they changed (or are they likely to change)? In essence what requirement set defines the current in-service system and therefore can be used as a starting point (existing system requirements baseline) for the definition for the changes required (future system requirements baseline)?

Through-life verification means that we need to ask **a different** question from time to time: “is the system still compliant with its specification in terms of behaviour and performance?

It seems sensible to maintain this distinction between validation and verification in the case of ISSE. Just referring to ‘Through-life V&V’, blurs a subtle but important differentiation.

When considering the validity of requirements in the in-service stage, you need to think about both the current and the future requirements and constraints (especially at the insertion date for any new increment of the product or service), not forgetting:

- operator/user/maintainer needs that are fulfilled, emergent and still awaited;
- applicable guidance, standards and legislation at each increment of the product and/or service to be offered;
- peer systems (product and service) requirements and constraints at each increment of the product and/or service to be offered;
- target system (product and service) capability; and
- available solutions from the market place.

Existing systems may have a requirements set that does not reflect current needs or practice, and the architecture and standards may not be up-to-date or legal (consider the progress of legislation on safety).

Requirements for operational (business) services, technical services and products will be different but related – all three are needed.

The following should be borne in mind when considering requirements validation and verification for in-service systems:

- The need to identify both functional and non-functional requirements (including performance). This distinction can be useful: non-functional requirements are often the ones that are overlooked, and they can be difficult to specify and implement.
- The need to involve stakeholders throughout the process.
- Measures of Effectiveness (MoEs) and Measures of Performance (MoPs) (see section 2.1).
- Verification is the continual checking that the system is still compliant with its specification.
- Validation is the continual checking that the requirements baseline is still correct (that is: is the system capability as specified still sufficient for purpose and do we still have the right thing?).
- System performance/effectiveness will change continually through life as its use, environment, maintenance, and material state changes.
- Verification documentation (that is, evidence that the system is compliant with its specification) will need to be updated for the support change activity.

3.2.3 Specific Guidance for Managing the System

Requirements

- Does a properly-formed requirements baseline for the system and the services it provides exist? If not, one needs to be created. See for guidance. Note: photographs can be a useful method of recording what is actually in place in the case of physical assets.

- Is the in-service system's requirements baseline managed (that is understood, committed to and under change control)?

Validation

- Is a process in place to monitor and assess feedback from the beneficiaries of the system and the service it provides to establish satisfaction levels and areas for improvement?
- The feedback monitoring system should distinguish between feedback concerning the system and feedback concerning the service.
- Are the in-service system's baselined requirements consistent with current plans, products, stakeholder expectations and regulations?
- Extrapolate trends to provide predictions of future performance (note that this could also be part of verification).
- Is there a gap between the in-service system's design intent (elements of which may not be captured in the requirements baseline) and material state? See also section 3.5 on Information and Configuration Management.
- As requirements not in the baseline are discovered, conduct trade-off analysis on whether to include them in upgrades as part of the 'Managing the System' activity, not in upgrade itself.
- Maintain an accurate, ongoing record of validation activities and decisions to defend likely subsequent justifications.
- Where necessary, trigger update and upgrade activities and co-ordinate these with fixing non-compliances (including the prioritisation methods noted under verification above).
- As the use of the system evolves, the MoEs are likely to change. These need to be monitored as part of the validation activities.

Verification

- The actual performance of the system and its services may be more or less than the specified performance, so performance verification will be required (although in older systems performance verification measures may be out of date or the system may be incapable of matching modern measuring techniques).
- Extrapolate trends to provide predictions of future performance.
- Appropriate processes and resources should be allocated to perform in-service verification.
- Is the in-service system compliant with its specification? Is the in-service system certified as safe and legal?
- How serious are the non-compliances? A mechanism should be set up to adjudicate on non-compliances as they are discovered. This could for example be an arrangement in which non-compliances are categorised as: 'urgent' (that is they must be dealt with straightaway), 'fix in next upgrade', 'ignore it' (if the fix is judged to cost more than living with it) and 'to be assessed further' (that is it is not clear how serious the non-compliance is, so it needs to be analysed in more detail).

3.2.4 Specific Guidance for Changing the System

Requirements

- Are the requirements to be included in the system change documented and subject to configuration management and change control?
- Does this include changes to service requirements as well as to new system requirements?
- Are the requirements to be included in the change traceable back to the rationale for making this change?

Validation

- The high level trade-off analyses should be completed and associated decisions reached before the change programme proper starts.
- Constraints (including so-called 'buildability' constraints) may be more important than user needs, so look at the changes from the perspective of constraints and buildability (they tend to get ignored!).
- Review information archive to identify gaps.
- Engage with all stakeholders and elicit information to fill the information gaps.

- Identify what aspects are important to the users.
- Generate requirements for the replacement system and update the information archive accordingly.
- Assess the remaining information gaps and associated risks.
- Schedule in mitigation actions.
- Note: formal documentation will only give some information of the material state so you will need to survey the actual assets.

Verification

- Are the requirements associated with the system change written such that they are verifiable (for example, using methods such as test, demonstration, inspection, analysis and argument by similarity? Have the verification criteria been agreed with the stakeholders? Are MoEs and MoPs used appropriately?
- Are plans in place to maintain traceability between requirements and design of the changed system?
- Are plans in place to review compliance of design against requirements?
- Is the future (including transition, intermediate and final) system capable of being certified as compliant, safe, and legal?
- Are the necessary plans and resources (test plans, trial plans, acceptance plans, certification plans, test equipment and facilities, staff, budget, timescales, access) agreed and in place, or at least planned for?

3.2.5 Specific Guidance for Delivering the Service

Requirements

- Do the requirements identification process and the requirements baseline distinguish between the following?

  a. Requirements that relate to the system's beneficiary stakeholders. These are those who benefit in some way from the system and the services it provides. In other words: the people and organisations that the system and its services are 'for'. Note that not all stakeholders are beneficiaries. For example: a Technical Authority, a Regulator, a Subject Matter Expert are all stakeholders, but they are not system beneficiaries.

  b. The requirements that the system must comply with if it is to work correctly in its operational context and be safe and legal.

- Some sources refer to these two classes as 'user' requirements and 'system' requirements respectively (although neither of these terms is ideal).
- Requirements in class (a) can be thought of as defining the services that the system has to deliver.
- Have these services been explicitly articulated in the requirements baseline? If not, include them.
- Are such services subject to configuration management and change control in the requirements baseline?

Validation

- System beneficiaries can be categorised under various headings, and which precise categorisation structure is used will be specific to the system.

  In a transportation system, the structure might include: passengers, the Treasury, the Transportation Authority, the Train Operating Companies, the environment, operational staff, maintainers, etc.

  In a defence system, the structure might include: military commanders, interoperating systems, the various relevant departments of Government, members of the fighting forces, logistics support, etc.

  So, ask the following questions:

  a. What categorisation structure has been used?

  b. Have all relevant beneficiary categories been included?

  c. Have any relevant categories been omitted?

d. Have any new ones emerged while the system has been in service?

e. Have any previous ones now become irrelevant while the system has been in service?

- Have appropriate techniques and sources been used to identify the required services as accurately as possible and any changes to them?

  The techniques and sources for identifying required services are the same when changing an in-service system as when building a new one and include: interviews, scenario workshops, working with people/observing how they go about things, fault reports, change requests, modifications to the system made by users, product user groups; Checkland's Soft Systems Methodology; Benefits Realisation Management; experimentation, simulation and Effectiveness/Performance Chain Analysis.
- Where relevant, do the services have associated MoEs?
- Have they been agreed with the stakeholders?

Verification

- Have the required services in the requirements baseline been formally verified as being delivered by the system?

3.2.6 Specific Guidance for Optimising the Supply Chain

Requirements

- Does your current supply chain structure meet your objectives adequately?
- Is there a better supply chain structure for your system? If so, do you have the flexibility to move to the better model? Is such a move viable?
- Does your supply chain 'buy into' the principles of Systems Engineering in general and good requirements practice in particular?
- Is your system life cycle correctly aligned with those in your supply chain?
- Is the right balance struck between your organisation identifying and specifying requirements and elements of your supply chain doing so?
- Is the right balance struck between a highly-competitive approach to enlisting members of your supply chain and 'partnering' relationships?
- Is the right balance struck between 'taut' contracting principles (which can lead to inflexibility and a lack of trust) and more relaxed ones (which can be inefficient and insufficiently disciplined)?
- Do you engage members of your supply chain at the most appropriate stage of the requirements process?

Validation

- Do you and your supply chain members share the same definition of 'Validation'?
- Do you involve your supply chain as appropriate in the identification and validation of the system's requirements?

Verification

- Do you and your supply chain members share the same definition of 'Verification'?
- How do you verify the product outputs from members of your supply chain?
- Do you have full visibility and control of verifying their outputs (for instance, by specifying verification criteria and by witnessing verification tests and signing off the results thereof), or do you have a 'hands off' self verification scheme (that is: you are happy to let your supply chain members verify their products on your behalf and trust that they do a professional job)?
- Have you got this balance right?

3.3 Architectural Design

3.3.1 Handbook Processes Affected

- 4.4: Architectural Design Process
- 8.2: Architectural Design

- Appendix K: System Architecture Synthesis

3.3.2 General Guidance

The guidance in the Handbook on Architectural Design remains of value from all perspectives on In-service Systems Engineering but does need to be applied with some circumspection.

The Handbook describes the **Architectural Design** task as that required to synthesize a system architecture baseline that satisfies the requirements and offers that **System Architecture** is defined as the selection of the types of system elements, their characteristics, and their arrangement.

In section 2.1 above, we introduced the concept of two complementary systems, the **Product System** and the **Service System**. Both systems act together to deliver a sustained and maintained solution to the client's need. In order to be successful, from a Systems Engineering point of view, both systems would need to progress through the six System Engineering Life Cycle stages (Concept, Development, Production, Utilization, Support, and Retirement) with the schedule and cost relationships between the two being programme dependent.

Recognising that the In-Service stage (which covers the Utilization and Support stages from the Handbook) usually occupies a longer time frame than the earlier stages we can simplistically represent the two systems thus:

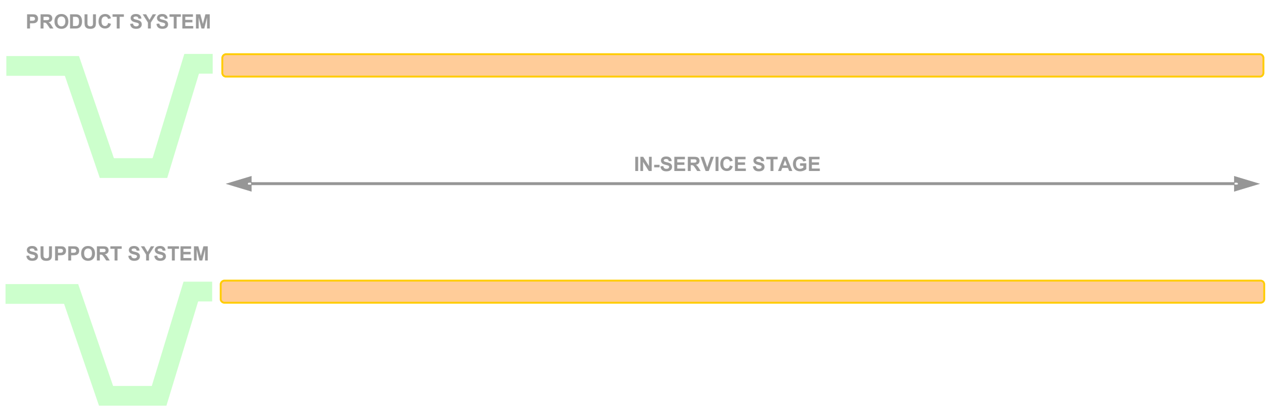

Figure 3-6: Co-evolution of product and service systems

During the In-Service Stage, in this simple representation, there would appear to be little need for Architectural Design. The two systems are in place and there is no indication of any issues with them or between them. The two systems combine to deliver a sustained and maintained solution to the client's need.

With this view In-service Systems Engineering is generally concerned with maintaining architectures rather than creating them. The guidance in the Handbook on an architectural baseline may be a better starting point than the guidance on that on the architectural design process. Section 4.4.4 of the Handbook lists the following main components of an architectural baseline.

- system element detailed descriptions with documented justification for concept selections;
- requirements assigned to system elements and documented in a traceability matrix; and
- interface requirements and a plan for system integration and Verification strategy.

In reality there are likely to be issues and consequentially the need to system engineer solutions and insert them into the existing Product-Service System Architectures.

Issues can be identified through observed deficiencies and inefficiencies in the *status quo* of the product and/or the service against the defined need. These can be generated from within the bounded Product-Service System (e.g. observed degradation in performance, high levels of spares usage, unacceptable levels of time/cost, etc.) as well as from outside the bounded Product-Service System through benchmarking against similar product-service solutions.

Shifts in legislation, standards, and client needs also represent potential sources of issues and changes that need system engineered solutions.

Thus we can reasonably predict that sometime during the In-Service stage there will be a need to insert a change to a System Architecture. When this happens there will be a need to design both the changes and the process of transition between the old and the near architectures.

Such effort would be part of the Concept, Development, and Production of the solution to the issue and the Concept, Development, and Production of the transition to the new *status quo*.

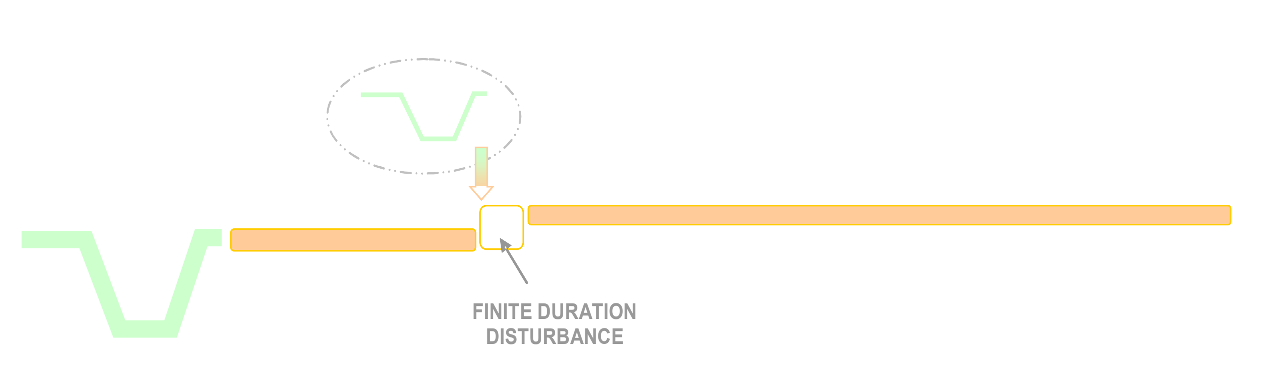

Figure 3-7: Insertion of change to system architecture

There are a number of problematic questions which arise during the resulting in-service architectural design and which affect several in-service perspectives:

- What is the current architecture?
- How much should the architecture be changed?
- How is a service maintained while changing the architecture?

We also need to recognise that changes can be to the Product or Service System Architectures or indeed necessitate changes to both.

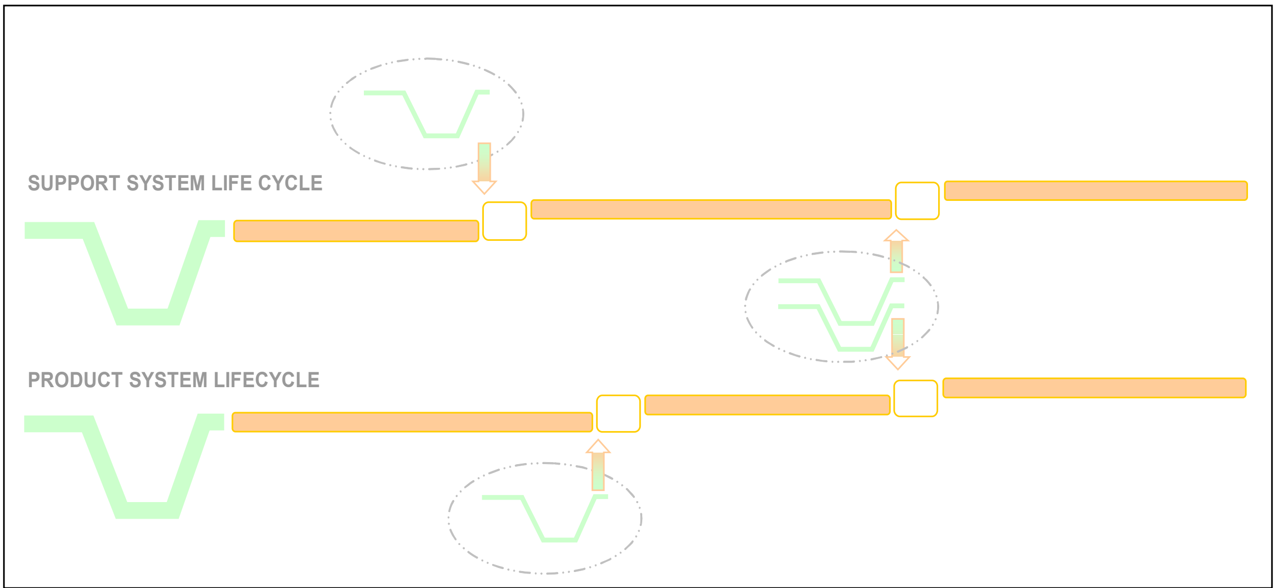

Figure 3-8: Insertion of change to product and service system architecture

3.3.3 What is the Current Architecture?

The actual architecture may not be fully captured; perhaps because the architectural design was done before a rigorous approach to Systems Engineering was adopted or because the architecture has changed since the architectural design and the architectural documents have not been updated.

If so, the first step must be to collect and collate whatever existing architectural documentation exists. If the system was built before a formal approach to systems engineering was put in place, there may be very little documentation of this type.

Some useful components of an architectural description may exist in documents that do not use the word "architecture" – system overviews or interface descriptions, for example. For long-established systems that have evolved over a period of time, many architectural decisions may be embedded in standards, which typically define the interfaces between the parts of the system.

If the architectural description is being created from scratch then the guidance offered by the Handbook is applicable. However, because of the difficulties in collecting and checking detailed information about the architecture of deployed systems, it may be necessary to simplify the methods and notations used.

Whether an explicit architectural description is inherited or created new from implicit evidence, there will be a need to check some aspects of it against the actual instantiation, typically by surveying the existing system.

If the system is geographically distributed or parts of it are hard to access, this process may have to be spread over a period of time. It may be cost-effective, for instance, to require maintenance engineers to take digital photographs of inaccessible components as part of their routine maintenance cycle so that the engineer can check them against the records. In that case, it may be necessary to build the architectural description incrementally or to associate an indication of reliability and/or references to survey results with some aspects of the description.

3.3.4 How Much Should the Architecture be Changed?

Successful systems evolve to deliver new benefits to their users and to cope with changes in the rest of the world. Systems architects try to create flexible architectures but they cannot foresee all the changes that will be made to the system so from time-to-time changes will be made that do not sit comfortably with the existing architecture. At this point there are generally two main options:

- 'Bolt On' / 'Work Around' - To make localized changes within the general framework of the architecture, perhaps creating an exception to an architectural rule. This is generally less expensive in the short-term but results in increasing complexity and inconsistency that, over time, can increase the costs of maintenance and further enhancement and decrease reliability.
- 'Embed' - To change the system architecture to accommodate the new functionality. This is generally more expensive in the short-term because it will require changes to elements that may not be involved in the new functionality but may pay dividends in the long run.

This is an investment decision and the most common strategy is to work within the existing architecture for a period of time and then to clear a backlog of desirable architectural change during a mid-life upgrade.

3.3.5 How is a Service Maintained while Changing the Architecture?

If it is necessary to keep the product system in service while it is being changed, it may be necessary to carry out the changes to the product and/or service in a series of phases, returning the system to service after each phase. This means that there is not just one final product/service architecture to be considered but several interim ones as well.

The architectural design problem now acquires a new dimension: time. The architectural baseline will extend beyond a description of the final architecture to include the migration path: the interim architectures and the transitions between them.

It is not sufficient just to show that the final architecture will allow the system requirements to be met with acceptable performance. It is also necessary to show that the interim architectures will support an acceptable, if possibly degraded, service and that the transition may be accomplished within the constraints set. For example, for a change to railway infrastructure, these constraints will include limitations on the closure periods during which the transition must be accomplished.

It may also be necessary to provide the ability to fall back from one architecture to the previous one, if something goes wrong with a change. This is discussed further in section 3.4.4 below.

The new dimension also introduces a new source of change: the migration path may change (e.g. because of a delay to one part of the project) even though the final system may not be affected and this must be controlled with as much rigour as changes to the final system are.

3.3.6 Specific Guidance for Managing the System

The general guidance given in 'What is the Current Architecture?' is applicable to this perspective, particularly the guidance on integrating surveys with routine maintenance activities.

The guidance on 'How much should the architecture be changed?' is applicable to this perspective, particularly the guidance on scheduling mid-life upgrades.

The guidance on 'How is a service maintained while changing the architecture?' is applicable to this perspective. Those responsible for managing the assets that make up the system will need to work with those changing the system to ensure that it remains maintainable in its interim configurations.

During the process of creating a new system, the architecture is often used as input to the creation of models to predict emergent properties such as reliability. These models must be maintained in step with the architecture. During the in-service stage these models remain of value and, therefore, must continue to be maintained in step with the architecture. However the theoretical models should now be compared with real-life experience in order to improve the models and in order to detect trends in performance that require action.

It is good practice to define a series of Measures of Performance (MoPs) to be collected in order to achieve this. These MoPs will need to be reconsidered whenever the system architecture changes.

3.3.7 Specific Guidance for Changing the System

The guidance on 'What is the Current Architecture?' is particularly applicable to this perspective, because an adequate understanding of the existing architecture is normally a pre-requisite to designing a change.

A project may have to proceed with an incomplete understanding of the architecture, focussing on the areas affected by the change. However it is important to ensure that the effect of the upgrade on shared resources such as power, cooling, space, bandwidth, computing and weight allocations is fully considered.

The guidance on 'How much should the architecture be changed?' is particularly applicable to this perspective, because it is a fundamental design question for the upgrade.

3.3.8 Specific Guidance for **Delivering the Service**

The System of Interest from this perspective will include operational processes and personnel as well as the technical system. An overall architecture for this system will need to be drawn up to describe how these components work together to deliver the required service. Changes to the architecture of the technical system will generally need to be associated with changes in this larger architecture. If a change is introduced in a number of phases then interim operational arrangements may need to be defined in order to operate the interim configurations of the technical system.

Suppose, for example, that in a control centre of some description, the workstations are being gradually replaced with new ones. For a period of time, new and old workstations may be in operation at the same time. If so, then there may need to be temporary operational arrangements for co-ordination between an operator at an old workstation and one at a new workstation that were not required before the migration started and will not be required after it has finished.

The guidance on 'How is a service maintained while changing the architecture?' is applicable to this perspective. Those responsible for using the system to deliver a system will need to work with those changing the system to ensure that it remains possible to deliver an adequate service while the system is in its interim configurations.

It is good practice to define a series of Measures of Effectiveness (MoEs) to be collected as measures of the service delivered and to link these to the MoPs associated with the product system. These MoPs will need to be reconsidered whenever the system architecture changes.

3.3.9 Specific Guidance for Optimising the Supply Chain

There will be a number of requirements that the service system / supply chain must meet if the Product System is to consistently meet its requirements in areas such as cost, capability dependability, responsiveness and the ability to cope with surges in demand.

Generally the techniques and models for this viewpoint are described and practiced within the ILS and Subcontract Management fields but again the focus needs to shift to the In-Service supply chain and working within the brown-field environment.

It will be valuable to use systems architectural techniques to describe the service system / supply chain as a system in order to describe and evaluate the current and possible futures.

The supply chain can be modelled to represent the key activities and steps required to maintain and sustain goods (parts) or services in order to understand where value is created, risk managed and costs consumed and to answer questions such as: Are these balanced? Is one part of the network consuming a disproportionate cost for the value generated/risk managed? Is one part of the network consuming too little?

For example, if a crucial component of a system is supplied by a sub-supplier to the system supplier then the supply chain architecture may show that discussions between the system operator and the component supplier to resolve an issue with the component would need to be mediated by two intermediaries. If this was inconsistent with responsiveness requirements than it might be desirable to create a direct relationship between the system operator and the component supplier.

The relationship between the supply chain, the service it supports, and the system performance and use can also be modelled to optimise systems performance against costs.

3.4 Implementation and Transition

3.4.1 Handbook Processes Affected

- 3.4.2 Incremental and Iterative Development
- 4.8 Transition Process.
- 4.11 Maintenance Process.
- 5.7 Configuration Management Process.
- 5.8 Information Management Process.
- 7.1 Decision Management Activity.
- 8.3 Configuration Management Activity.

3.4.2 General Guidance

At whatever stage of the project life cycle that a new/upgraded system or process is introduced into service, it is necessary to take a structured approach. This ensures that all risk factors that could affect the successful implementation and transition of the system/process are mitigated, managed or avoided.

The principle project approaches will include:

- Once-through: Plan, specify, and implement the complete system in one pass through the "V".
- Incremental: Plan and specify the system and then implement it in a series of well-defined increments.
- Evolutionary: Plan, specify, and implement an initial system capability. Gain experience with the initial system and define the next iteration to fix problems and extend capabilities.

These are described in the Handbook, Section 3.4.2.

The next section contains some of the reasons to pick one strategy over another in the IS environment. In general, the once-through strategy is well suited for low-risk projects and the evolutionary strategy is more adaptable and well suited for higher-risk projects where there are significant unknowns.

Development Strategy	Opportunities (Reasons to Use)	Risks (Reasons to Avoid)
Once-Through	• All capabilities needed/desired at first delivery • Must phase out old system all at once • Efficient – If it is known exactly what is wanted	• Requirements are not well understood • Rapid changes to requirements possible • Large system with many unknowns • Limited staff or budget available now
Incremental	• Early capability is needed • System breaks naturally into increments • Funding/staffing will be incremental	• Requirements are not well understood • All capabilities needed/desired at first delivery • Must phase out old system all at once
Evolutionary	• User feedback is needed to understand full requirements • Early capability is needed • System breaks naturally into increments • Funding/staffing will be incremental	• All capabilities needed/desired at first delivery • Must phase out old system all at once

Table 3-1: Development Strategy Opportunities and Risks

3.4.3 Specific Guidance for Managing the System

For Managing the System, it is crucial to balance asset use with planned and unplanned operational availability in order to maintain the required system performance during implementation and transition. This tends to militate against Once-Through strategies for In-Service systems unless there is a compelling reason to make a step-change or there is a defined retirement point for the current system/process that cannot be extended.

Considerations include:

- Balancing potential lower initial availability of new equipment/services against the long-term benefits of the change and managing stakeholder expectations accordingly. This tends to attract the attention of Project Management as it impacts on system/process cost and performance.
- Managing the pace of change. This affects the speed at which the new system/process assets can deliver benefits. However, the potential impact on the ability of maintenance and support stakeholders to prepare for sustaining the new assets must be managed if they are going to be capable able of maintaining availability of both sets of assets until the retirement of current systems/processes.
- Recognising that people are assets, too. Change management is always most difficult when people are involved. The Human Resource assets of the system should be kept informed and involved. Transition training and operator processes must be in place and proven at the appropriate time during transition, and effectively and positively implemented.

Incremental (perfective) performance improvements could be achieved through: commissioning of upgrades, improving maintenance processes, changing operating procedures or improving operator performance. These may not include fundamental changes to the design, but are more focused on increasing the efficiency or cost-effectiveness of the existing system/process. A closed-loop process (see section 3.5.2 below) can be very effective in identifying and managing such changes. Considerations include:

- Trialling of the new system on a low risk part of the business. Off-line test and evaluation typically helps to reduce the risk of the new system failing to deliver against expectations.
- Effective implementation and transition planning, to identify and manage the risks, including consideration of:

- Upgrade cycle times. These should be similar to related technology cycle times (that is, 5 years for communications, 10-15 years for mechanical systems).
- The need to provision funding and fit opportunities for subsequent upgrades and updates 3-15 years beyond the current change event.
- Configuration management of the change to avoid loss of system performance with no available (and suitable) fall-back position defined.

Evolutionary (adaptive) changes mean moving generally towards more open system/process architectures as a long-term enabler for more cost-effective upgrades, with lower operational downtime. This leads to the ability to implement an Incremental or Evolutionary strategy for changes, without having to lose significant levels of operational availability during the process.

For Managing the System, this could mean implementation of apparently unconnected changes that are managed under a broad-ranging, long-term plan. This allows subtle changes to be made that have little immediate effect but cumulatively have a larger impact by changing the way that assets are managed.

3.4.4 Specific Guidance for Changing the System

There are 4 basic stages – creating the new components of the system, installing the changes, commissioning the changes and working up the capability. Note that the new components of the system may include hardware, software, documentation, processes or people.

Considerations are that:

- Acquisition strategy for or preparation of the modification kit may impose constraints due to (for instance) delivery schedules, cash-flow requirements, training design or physical characteristics.
- Fitting the modification needs to be programmed in to operations – may require operational down time.
- If there are a number of systems, Incremental fitting across the fleet means that more than one configuration is live at any one time – requires effective configuration management. Fit opportunities are likely to be the constraining factor in technology upgrades.
- For large systems, particularly infrastructure systems, it may be necessary to apply the change in a number of stages and to return the system to service after each stage is complete. Planning out the migration between the stages may be a significant task.
- Commissioning may require operational down (or quiet time). There is a risk that the new equipment will bring the whole system down – this risk needs to be managed as a contingency plan.

Good practice includes having a robust process for reviewing and checking operational and maintenance readiness. The measures that must be in place in order that the systems may be operated and maintained reliably (such as training, spares, updates to procedures) is drawn up well before entry to service. This checklist is often extended to included temporary measures needed for the transition phase (such as plans to fallback to the previous configuration if there are problems with the new one). A gate review is then convened shortly before entry into service to review the state of readiness and to decide whether to proceed into service of not.

3.4.5 Specific Guidance for Delivering the Service

The new/improved service needs to be transitioned into service at the same time as the upgraded/new system. This requires co-ordinated rollout of training and supporting infrastructure, which could affect delivery of the current service to customers and impact on performance as “bugs” are ironed out. Considerations are:

- Off-line test and evaluation facilities can decouple system and service transition – helping to enable demonstration of meeting service technical Measures of Performance before deploying in front of the Customer.
- Work-up may also require new operational approaches to be implemented. Need to work up existing staff in using the new service. This can be subject to resistance and also existing skills and knowledge will need to be up-graded, so there will be a requirement for effective communication and training.
- Use operational MoEs as a target for work-up of service operations.

3.4.6 Specific Guidance for Optimising the Supply Chain

The new/improved system must be adequately supported both during and after the transition period (Note that the supply chain itself could be the new system). The new system must be capable of being rapidly restored to use following a failure and must be cost effectively sustained for its required design life.

The system being retired must also be adequately supported so that it remains operational until the new system is fully rolled out. Considerations are:

- Implement the required commercial framework, ensuring that suppliers are able to deliver according to the plan. Programme-manage delivery of supply chain improvements against agreed performance framework
- Never under-estimate the cost and impact of logistics support on the system operational capability. Whole-life cost of ownership could be influenced more by logistics than the original cost of development and acquisition.
- Concurrent implementation of cost-effective logistics is the foundation of successful System operation and this must be sustained for the in-service life of the capability. This includes: Training; Facilities; Support and Test Equipment; Packaging, Handling, Storage and Transportation; Spares Provisioning; Disposal; System Reliability and Maintainability; Obsolescence Management.
- Analysis and support of the above logistic impact and requirements is carried out under the ILS discipline.

3.5 Information and Configuration Management

3.5.1 Handbook processes affected

- 4.3 Requirements Analysis Process
- 4.8 Transition Process
- 4.11 Maintenance Process
- 5.7 Configuration Management Process
- 5.8 Information Management Process
- 7.1 Decision Management Activity
- 8.3 Configuration Management Activity
- 8.4 Information Management

3.5.2 General Guidance

Information Management (IM) and Configuration Management (CM) for system realisation are covered comprehensively within the Handbook. Both activities are required throughout the whole system life cycle. However, the guidance provided is orientated to an idealised system life cycle.

The in-service stage often introduces additional complications such as:

- unavailability of comprehensive, reliable information about the existing system and its environment;
- a product may be manufactured or changed by supplier(s), but owned and used by others with the supply chain responsible for keeping the product up-to-date;
- a system may have numerous design changes and turnover of personnel over long in-service life cycle stages (60 years or more); amd
- management of information and the configuration over many versions/modifications.

IM is concerned with ensuring information is properly stored, maintained, secured and made accessible to those who need it. Currently the advice centres around IM as ‘product’ information however in-service systems require long term management of three generic sets of information:

- the design brief (summation of requirements and constraints);
- the design disclosure (including physical, functional, and operability information); and
- the design justification (evidence of compliance of the design to the brief).

In-service IM is not limited to the capture and storage of product information for system management and upgrade. It requires additional information (often missed when scoping IM early in the life cycle), in support of through-life supply chain and service delivery needs.

Effective systems engineering of an in-service system may require a broad range of information and data, including:

- data to manage customer service improvement;
- data to support enterprise information management and decision making in the long term;
- information to support on-going collaborative partnerships;
- information on technology, equipment, and business process development;
- data on process improvements such as enhanced workforce efficiency; and
- information to support Decision Analysis, leadership and governance.

Not all of this information may be available as product information. Some of it may be in individual's heads or personal records or not available at all. The term 'knowledge management' is a term used by some people to describe the business of collecting and maintaining all this information.

Like IM, CM may experience additional barriers during in-service support. Design Authority delegation during system realisation and update/upgrade creates CM interface issues. CM during system development often deals with bought-in subsystems as configured items. In-service systems are more concerned with CM down to Line Replacement Unit (LRU), (conceivably component level), to support maintenance, update and upkeep.

Representation of the in-service viewpoint early in the life cycle will allow handover of product and project information that is useful and appropriate for use in the in-service stage. The requirement is to avoid the risk inherent in continued management of information intended for one community that is often of little use to the next whilst managing both the existing and the information about the changing system.

Modern IM and CM systems can utilise the longevity of digital storage technology, enabling archiving and recovery of data over several generations of system hardware / software. However, there is a risk that, as the context of the information passes beyond the memory of one generation, information within the management systems may be devalued, misinterpreted or even lost over time. There then needs to be a trade off between the cost of sustaining and maintaining the information verses the cost of recovering lost information. Use of the survey techniques to revalidate information and the management of emergent work strategies can mitigate some of the costs. However there is always a risk that old information/data may never be fully recovered - even by survey.

Standards management is a specific instance of whole of life cycle CM – monitoring and maintaining the linkage between the configured item and the Standards to which the product was designed, built, tested, accepted, operated and maintained. The impact of a change to a Standard can then be considered. Here CM straddles the boundary between controlling the standards associated with the product and the standards associated with the processes used in its production and use.

The terms Fault Reporting, Analysis and Corrective Action System (FRACAS) and Data Reporting, Analysis and Corrective Action System (DRACAS) are used to describe processes for collecting, analysing and acting upon faults and other in-service data. Some people draw a distinction between the two terms but, in the authors' experience there is not universal agreement on the distinction. To work effectively the DRACAS or FRACAS needs to be under CM.

There are numerous sources for IM and CM good practice. The following is considered to be good practice for IM and CM in-service systems:

- A 'multi-media' CM system may be required to record what is actually in place, providing a mixture of data, drawings, photographs, laser scans, video, etc.
- Lessons learned from previous similar projects should be taken into account.
- It may be useful to write a Knowledge Transition Plan or include IM and CM in the Project Transition Plan.
- Be wary of the 'garbage in – gospel out' problem - just because the information is in a formal system does not mean that it is reliable.
- Include programmed "weeding" of the information library throughout the in-service stage to limit the opportunity for confusion caused by redundant data.
- Be clear why information is being created, what it is being used for and how it needs to be managed.
- Recognise that different types and levels of information may be needed.
- Use agreed data standards for holding configuration information.

- A common repository with configuration control and workflow can be very useful.
- Accountability, together with boundaries and interfaces should be clearly defined.
- Remember the 'libraries on legs' – people are a great source of knowledge, particularly contextual.

In the context of IM there will be a need to address document (information) retention. Amongst other guidance the 'National Society of Professional Engineers Document Retention Guidelines' provides guidance.

Also worthy of consideration are emerging solutions to the IM challenges regarding:

- how data is organised to allow large collections of information to be accessed (mined) by diverse communities over long timescales;
- appropriate security; and
- identification of information provenance.

3.5.3 Specific Guidance for Managing the System

The parties taking the Managing the System viewpoint are particularly reliant upon CM. 'Product Lifecycle Support' (ISO 10303-239) provides a good start for a system management data set. System data will need updating following an upgrade, some data provided at handover will not be relevant, new information may not be complete.

Transfer of the configuration data from the realisation / upgrade project into the operational environment should be considered early in the project as well as near the scheduled time of transfer. This may dictate the timing and degree of CM applied in the project and the arrangements established for CM in the operations and maintenance environment. The organisation for CM for the whole of the life cycle must be considered from the start.

This whole life view must include an iterative, structured and auditable process for Standards Management.

3.5.4 Specific Guidance for Changing the System

When changing the system, control of the interfaces between the stakeholders' CM processes need to be established clearly and early enough to enable responsibilities to be clearly identified and effectively transferred where necessary. 'SE Data Exchange' (ISO 10303-233) is a good example of a data set for this viewpoint.

Good IM and CM during system change only works if embedded early in the project and addresses:

- Reviewing the information archive to identify gaps – the asset data will be a good starting point.
- Validating this data is critical for several reasons:
  - The system management data may not be of sufficient quality for the change project.
  - The process of validation will raise the domain knowledge of the upgrade team and identify 'unknown unknowns'.
  - The validation process will help build rapport between the upgrade project and system management teams.
- Engagement with all stakeholders and elicitation of information to fill any other information gaps.
- Identification of what aspects are important to the users.
- Generating requirements for the replacement system and update the information archive accordingly.
- Assessing the remaining information gaps and associated risks.
- Planning for the management of knowledge to help with the update project and mitigate future problems, including actions such as:
  - o updating drawings to reflect the in-use system;
  - o surveying technology and produceability, considering obsolescence; and
  - o surveying applicable legislation and regulation.
- Including preparations for update of the system configuration at embodiment.

3.5.5 Specific Guidance for Delivering the Service

The 'ITIL service delivery model' has a set of concepts and practices for managing IT services, IT development and IT operations and provided useful guidance on the delivery of an IT-based IM System. ITIL v3 comprises:

- Service strategy;
- Service design;
- Service transition;
- Service operation; and
- Continual service improvement.

Like all systems, an IM System may experience a serious incident at any time that prevents it from continuing normal operations. The ability to recover from such incidents in the minimum amount of time requires careful preparation, planning and potentially the development of a Business Continuity Plan. A primary function of the Business Continuity Plan will be to address continuity of Information and IM System service.

The IM and CM systems are not only used by the in-service community, but also provide an invaluable source of information for other life cycle stages. Architects and designers rarely work in a 'green-field' environment and so need to establish a working understanding of the 'brown-field' (in-service) environment and planned 'future' environments. This means IM and CM systems need to be able to offer information on the current and future system configurations and on the systems and service environments.

3.5.6 Specific Guidance for Optimising the Supply Chain

The supply chain is a system in its own right, which is subject to IM processes. The supply chain needs information and this information has to be managed.

For example; consider the supply of fuel. Knowledge of the user fuel usage rate and usage profile fluctuations is required to create a supply network architecture and performance framework that meets the user's needs. During the in-service state this is monitored and optimised as conditions change.

Accurate system CM is also crucial to the effectiveness of the supply chain. Multiple systems may be deployed across platforms at various modification states. Provision of the correct spares, data, tools, and maintenance, etc. to support each variant will be dependent a coherent system of CM data and effective liaison between the system delivery team and the supply chain team.

High quality information flow is critical to successful supply chain integration. Collaborative Working Environments can deliver significant reductions in cost and improvements in upgrade time. Implementation of a Collaborative Working Environment needs to focus on the specific benefits sought.

4 Conclusions

The authors consider that a well-managed extension of the SE performed during the original development of the system into the in-service stage is needed to protect the investment made in the system.

We subscribe to the emerging consensus within the SE community that, while the principles underpinning SE remain the same across the life cycle:

- some of the issues concerned with sustaining existing systems are more problematic than when realising new systems; and
- the existing SE Handbook and the competences of SE practitioners tend to be stronger on the issues that are more important when realising new systems.

In particular, we note that many aspects of In-Service SE have to be performed continually, outside the rhythm of a project, which is bounded in time.

We consider that there are four main areas where the Handbook needs to supplementary guidance for effective application during service:

- Requirements, Validation and Verification;
- Architectural Design;
- Implementation and Transition; and
- Information and Configuration Management

This report provides guidance on good practice in In-Service SE for these areas.

We have identified four distinct viewpoints for In-Service SE:

- Managing the system;
- Changing the system;
- Delivering the service; and
- Optimising the supply chain.

The guidance in this report is structured by these viewpoints.

In preparing this guidance, we have found that the new viewpoints demand new SE concepts and we have tried to define the vocabulary used for these concepts with care.

We know that this report cannot be the last word on the matter and are delighted that an international In-Service Systems Working Group has been established to take the work forward. We commend this report to those engaged in in-service systems engineering. We request that users of this guidance should pass their comments and experience of using the guidance to the international group so that they can be taken into account in future developments. Contact details for the internal In-Service Systems Working Group may be found at www.incose.org.

ANNEX A. TERMS OF REFERENCE FOR THE UK IN-SERVICE SYSTEMS WORKING GROUP

The following terms of reference were agreed with the INCOSE UK Board.

It is frequently necessary to change systems that are in service in order to sustain them and SE is just as important in this stage as it is when realising new systems. There is consensus within the United Kingdom (UK) SE community that, while the principles underpinning SE remain the same across the life cycle:

- some of the issues concerned with sustaining existing systems are more problematic than when realising new systems; and
- the existing SE Body of Knowledge and the competences of SE practitioners tend to be stronger on the issues that are more important when realising new systems than on the issues that are more important when changing existing systems.

This view was expressed at the INCOSE UK Advisory Board and confirmed at a workshop at the INCOSE UK 2007 Autumn Assembly. As a result, the INCOSE UK Board commissioned a working group (ISSWG) to advise it on:

- the difficulties encountered, in practice, in applying authoritative guidance on SE, including the INCOSE SE Handbook, to systems that are in service;
- best current practice in adapting SE guidance to overcome these difficulties; and
- additional work that the INCOSE UK Chapter might initiate to assist its members further in overcoming these difficulties.

This working group published a final report , and presented their findings at the INCOSE 2008 Autumn Assembly.

The conclusions of that report were, in summary:

- that the initial view, that the available guidance on SE in the in-service stage is capable of improvement, was upheld;
- six specific areas of guidance were found that could be significantly improved; and
- further work was needed to develop the guidance, which should be integrated into the SE Handbook.

The ISSWG recommended that, in summary:

- An International working group should be set up to improve and extend the work carried out by the ISSWG, to achieve a broader consensus on the conclusions and to establish arrangements for integrating additional guidance into existing INCOSE products;
- A UK working group should be set up to produce a supplement to the SE Handbook, providing guidance in the areas identified by the ISSWG to address the short-term, whilst the International working group generated its output.

These recommendations were accepted by the INCOSE UK Board and both working groups were commissioned.

ANNEX B. ACKNOWLEDGEMENTS AND CONTRIBUTORS

The members of the working group who contributed to the preparation of this report are listed below. Collectively, the members have significant experience of several sectors including the military, rail and air traffic services sectors. The views expressed by in this report do not necessarily represent the views of the member's employers.

- Andy Bourne, Tube Lines
- Lloyd Bridgewater, BAE Systems
- Ron Brittain, BAE Systems
- Doug Cowper, Cleave Systems Limited
- Bruce Elliott, Arbutus Technical Consulting
- Andrew Farncombe, John Boardman Associates
- Duncan Kemp, Department for Transport
- Alan Knott, Parsons Brinckerhoff
- Stewart Leinster-Evans, BAE Systems
- Peter Lister, Harmonic
- Ian McNeil, Boeing Defence UK
- Glenn Panter, Thales
- Colin Smith, Persides
- Matt Taylor, Babcock International Group

The working group would also like to acknowledge receipt of valuable comments on drafts of the report from Gordon Woods and members of the INCOSE International In-Service Systems Working Group.