Software Safety Assurance System

A NATS Perspective

Roger Dillon (roger.dillon@nats.co.uk)

Robert Granville (robert.granville@nats.co.uk)

ES2 WS3-11

22nd September 2011

Agenda

- » What is Software; where is it in ATM systems?
- » What is Assurance; how does it differ from Development?
- » The NATS Software Safety Assurance System
- » Assurance approaches for different types of Software
- » How much Assurance do you need?
- » Using EUROCAE ED-109. Is it enough?
- » Conclusion

NATS

What is Software?

- The legal definition of software in (EC) No 482/2008, “Commission Regulation of 30 May 2008 establishing a software safety assurance system …”, is:
  - ‘Software’ means computer programmes and corresponding configuration data, including non-developmental software, but excluding electronic items, namely application specific integrated circuits, programmable gate arrays or solid-state logic controllers.
- This definition does not really go far enough; Ian Sommerville (Professor of Software Engineering in the School of Computer Science at St Andrews University, Scotland) says:
  - Software is not just the programs but also all associated documentation and configuration data which is needed to make these programs operate correctly.
- We have adopted the legal definition, but take “corresponding configuration data” to mean “associated documentation and configuration data”, as above.

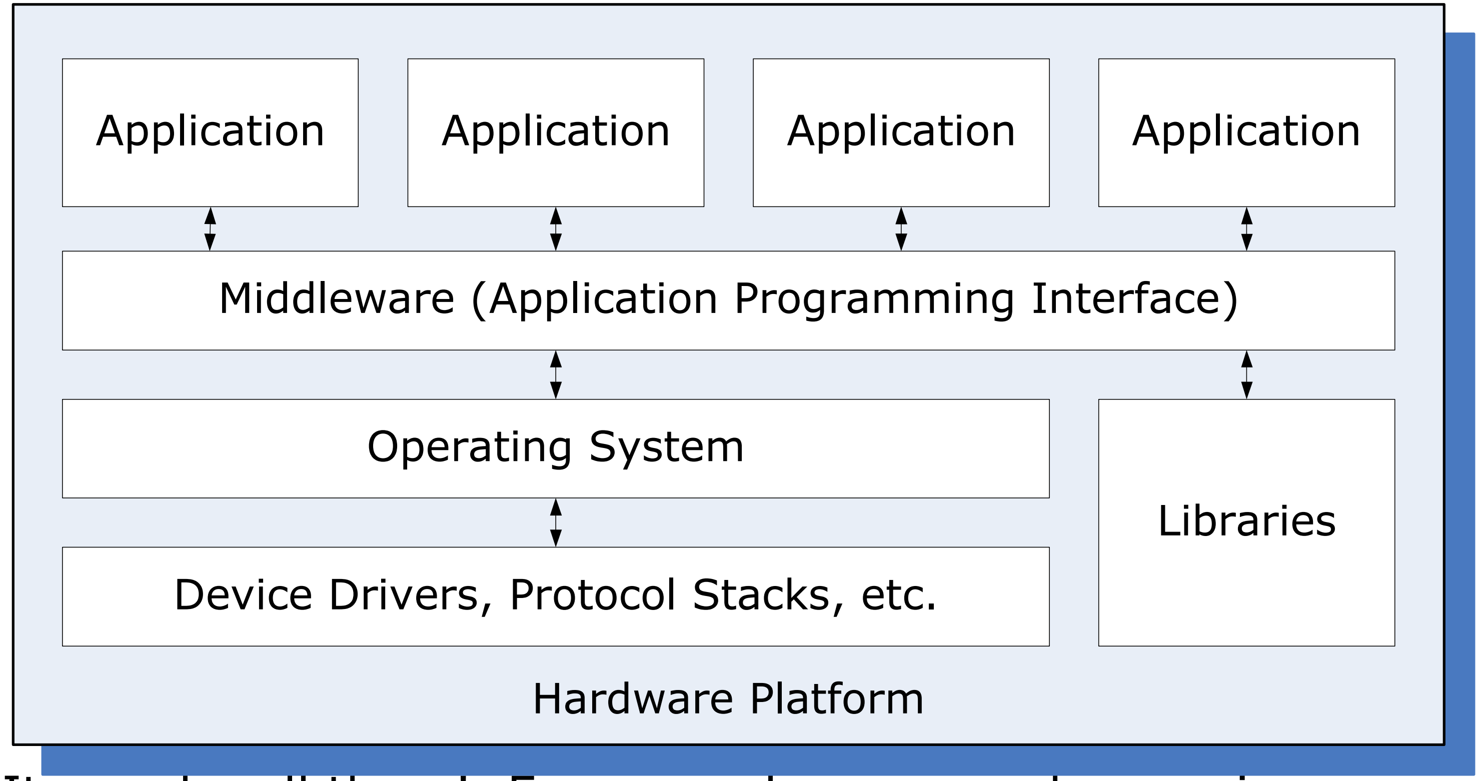
What is Software?

- Note: the legal definition of software has some explicit exclusions.
  - **…** excluding electronic items, namely application specific integrated circuits, programmable gate arrays or solid-state logic controllers.
- We interpret this to be saying, “If you design something using hardware techniques, and test it using hardware techniques, it is not software, even if it is implemented in a programmable device”.

- The Regulation identifies different types of software, e.g. New development, legacy, bought-in…

Different Types of Software

- What do you have in your equipment; is it new development, legacy, or bought-in?



Application

Application

Application

Application

Middleware (Application Programming Interface)

Operating System

Libraries

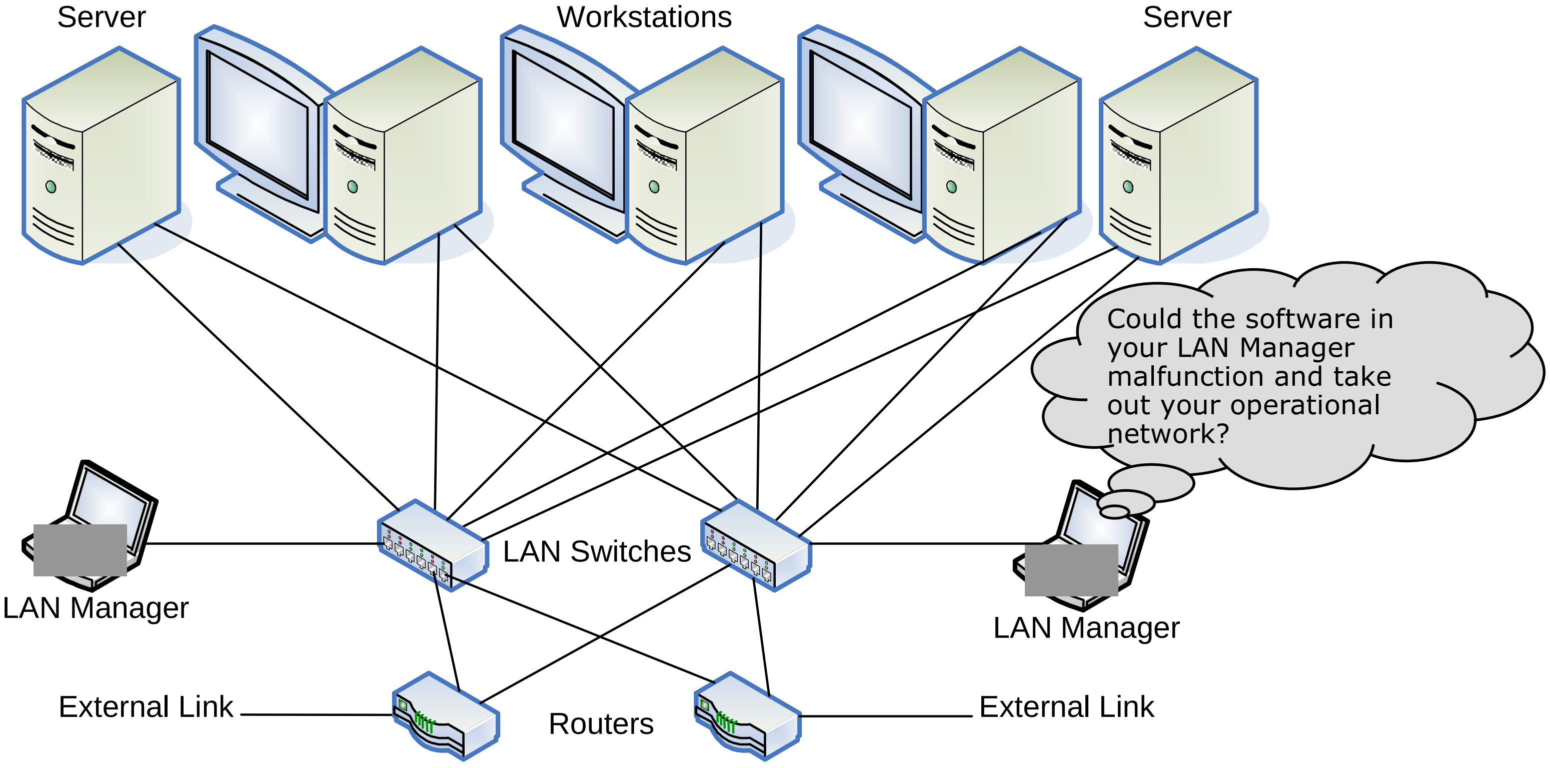
Device Drivers, Protocol Stacks, etc.

Hardware Platform

- It may be all three! For example you may be running new Applications, using a legacy library, over a bought-in Operating System and Middleware.

Different Types of Software II

» Where is the software?



Could the software in your LAN Manager malfunction and take out your operational network?

» It is everywhere, including the LAN Switches, Routers and maybe even the workstation displays.

ATM Software

- ATM Software includes:
  - Flight Data Processing
  - Surveillance Data Processing
  - Controller Workstation Management
  - Communications Management
  - And so on…
- Also critical to the operation:
  - Traffic Prediction, Controller Rostering, etc.
  - Periodic Maintenance Scheduling and Resourcing
  - Aeronautical Information Generation & Promulgation Tools
  - Embedded Software, for example in the Network Infrastructure
  - And so on…
- We require Assurance (to different degrees) for all of this software

What is Assurance?

- Assurance is the basis for justified confidence in something, for example that a system exhibits a required property; it should be presented in the form of logical arguments supported by pertinent evidence.

- You make a case; you are *not* proving anything.

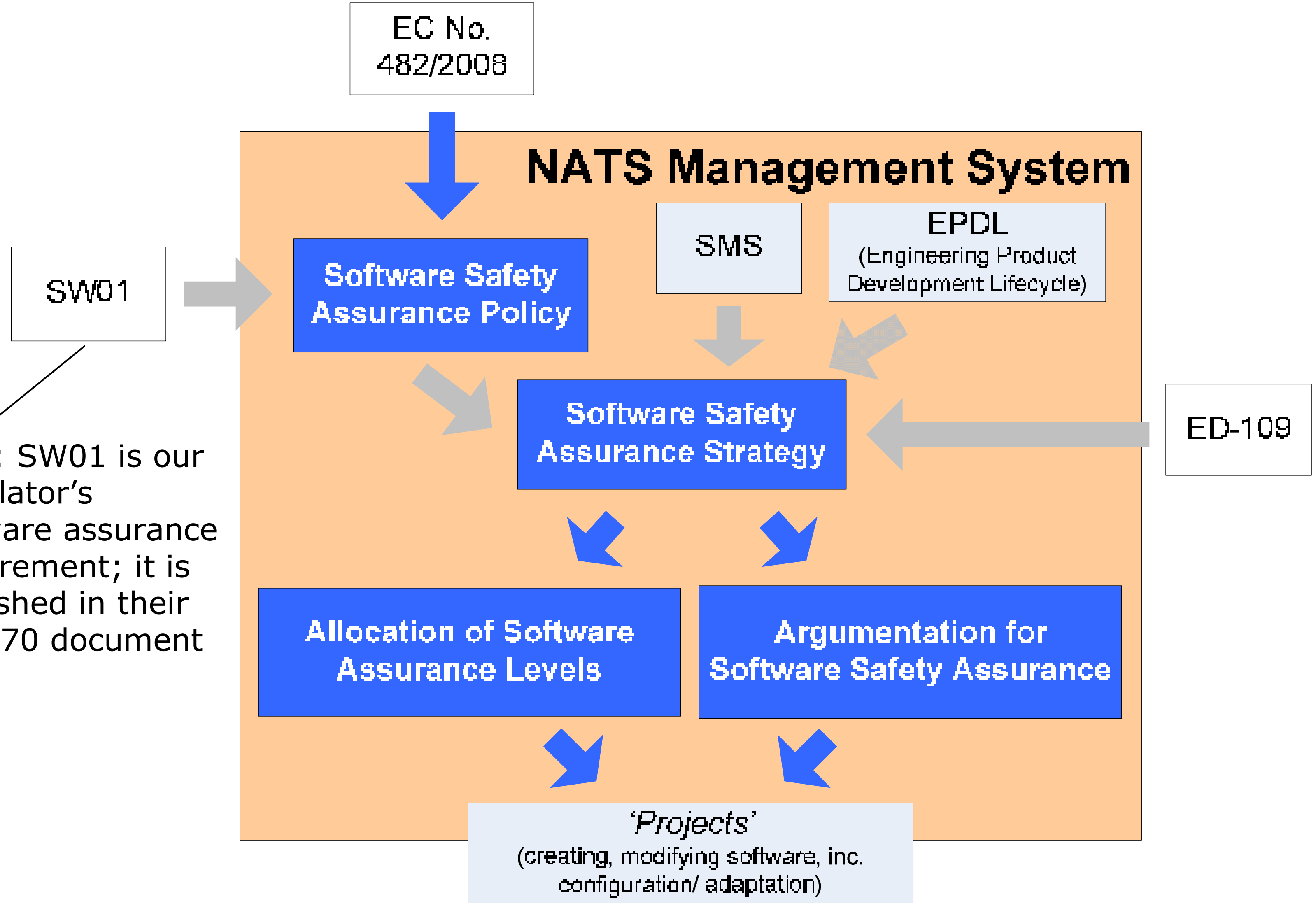
- Software Safety Assurance is the demonstration that the safety risks associated with the deployment of software in our systems have been reduced to a tolerable level for all stages of the operational lifecycle.

- This is to be achieved through a planned and systematic set of activities that provide confidence in the software conforming to constraints, requirements and, where pertinent, standards.

How does Assurance differ from Development?

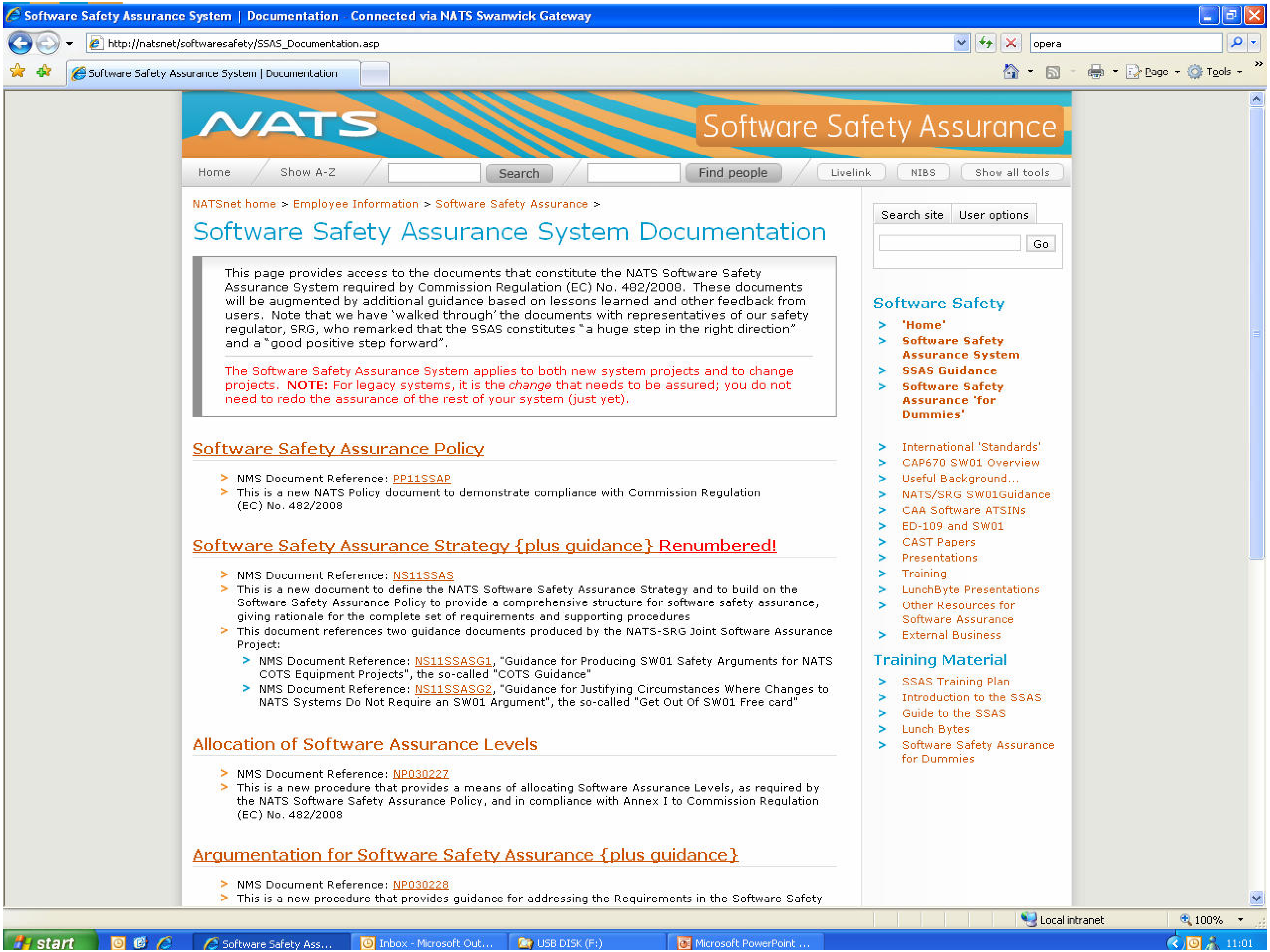
- In almost every respect!
  - Software development is the set of activities that results in software products (Wikipedia definition); whereas
  - Software assurance provides confidence in a software product's suitability for its intended purpose.
- They are aligned however, the development processes need to provide the evidence required to support the assurance argument.
  - Obtaining it retrospectively is difficult! The assurance argument may need to define a level of detail for the development processes.
- They also overlap; some activities serve both purposes.
  - For example the developer uses testing to show that the software behaves correctly under specified conditions; this behaviour can include the mitigations in which the assuror is interested.

Structure of NATS Software Safety Assurance System


EC No. 482/2008
NATS Management System
SW01
Software Safety Assurance Policy
SMS
EPDL
(Engineering Product Development Lifecycle)
Software Safety Assurance Strategy
ED-109
Allocation of Software Assurance Levels
Argumentation for Software Safety Assurance
'Projects'
(creating, modifying software, inc. configuration/ adaptation)

Note: SW01 is our Regulator's software assurance requirement; it is published in their CAP670 document

NATS


Software Safety Assurance System | Documentation - Connected via NATS Swanwick Gateway
http://natsnet/softwaresafety/SSAS_Documentation.asp
opera
Software Safety Assurance System | Documentation
Page
Tools
NATS
Software Safety Assurance
Home
Show A-Z
Search
Find people
Livelink
NIBS
Show all tools
Search site
User options
Go
Local intranet
100%
start
Software Safety Ass...
Inbox - Microsoft Out...
USB DISK (F:)
Microsoft PowerPoint ...
11:01

NATSnet home > Employee Information > Software Safety Assurance >

Software Safety Assurance System Documentation

This page provides access to the documents that constitute the NATS Software Safety Assurance System required by Commission Regulation (EC) No. 482/2008. These documents will be augmented by additional guidance based on lessons learned and other feedback from users. Note that we have 'walked through' the documents with representatives of our safety regulator, SRG, who remarked that the SSAS constitutes "a huge step in the right direction" and a "good positive step forward".

The Software Safety Assurance System applies to both new system projects and to change projects. **NOTE:** For legacy systems, it is the *change* that needs to be assured; you do not need to redo the assurance of the rest of your system (just yet).

Software Safety Assurance Policy

- NMS Document Reference: PP11SSAP
- This is a new NATS Policy document to demonstrate compliance with Commission Regulation (EC) No. 482/2008

Software Safety Assurance Strategy {plus guidance} Renumbered!

- NMS Document Reference: NS11SSAS
- This is a new document to define the NATS Software Safety Assurance Strategy and to build on the Software Safety Assurance Policy to provide a comprehensive structure for software safety assurance, giving rationale for the complete set of requirements and supporting procedures
- This document references two guidance documents produced by the NATS-SRG Joint Software Assurance Project:
  - NMS Document Reference: NS11SSASG1, "Guidance for Producing SW01 Safety Arguments for NATS COTS Equipment Projects", the so-called "COTS Guidance"
  - NMS Document Reference: NS11SSASG2, "Guidance for Justifying Circumstances Where Changes to NATS Systems Do Not Require an SW01 Argument", the so-called "Get Out Of SW01 Free card"

Allocation of Software Assurance Levels

- NMS Document Reference: NP030227
- This is a new procedure that provides a means of allocating Software Assurance Levels, as required by the NATS Software Safety Assurance Policy, and in compliance with Annex I to Commission Regulation (EC) No. 482/2008

Argumentation for Software Safety Assurance {plus guidance}

- NMS Document Reference: NP030228
- This is a new procedure that provides guidance for addressing the Requirements in the Software Safety

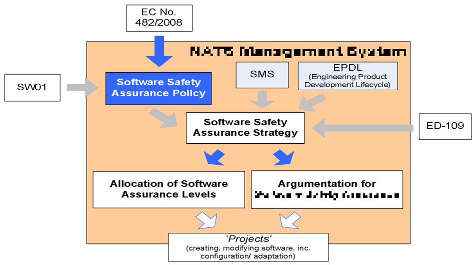
Software Safety

- 'Home'
- Software Safety Assurance System
- SSAS Guidance
- Software Safety Assurance 'for Dummies'
- International 'Standards'
- CAP670 SW01 Overview
- Useful Background...
- NATS/SRG SW01Guidance
- CAA Software ATSINs
- ED-109 and SW01
- CAST Papers
- Presentations
- Training
- LunchByte Presentations
- Other Resources for Software Assurance
- External Business

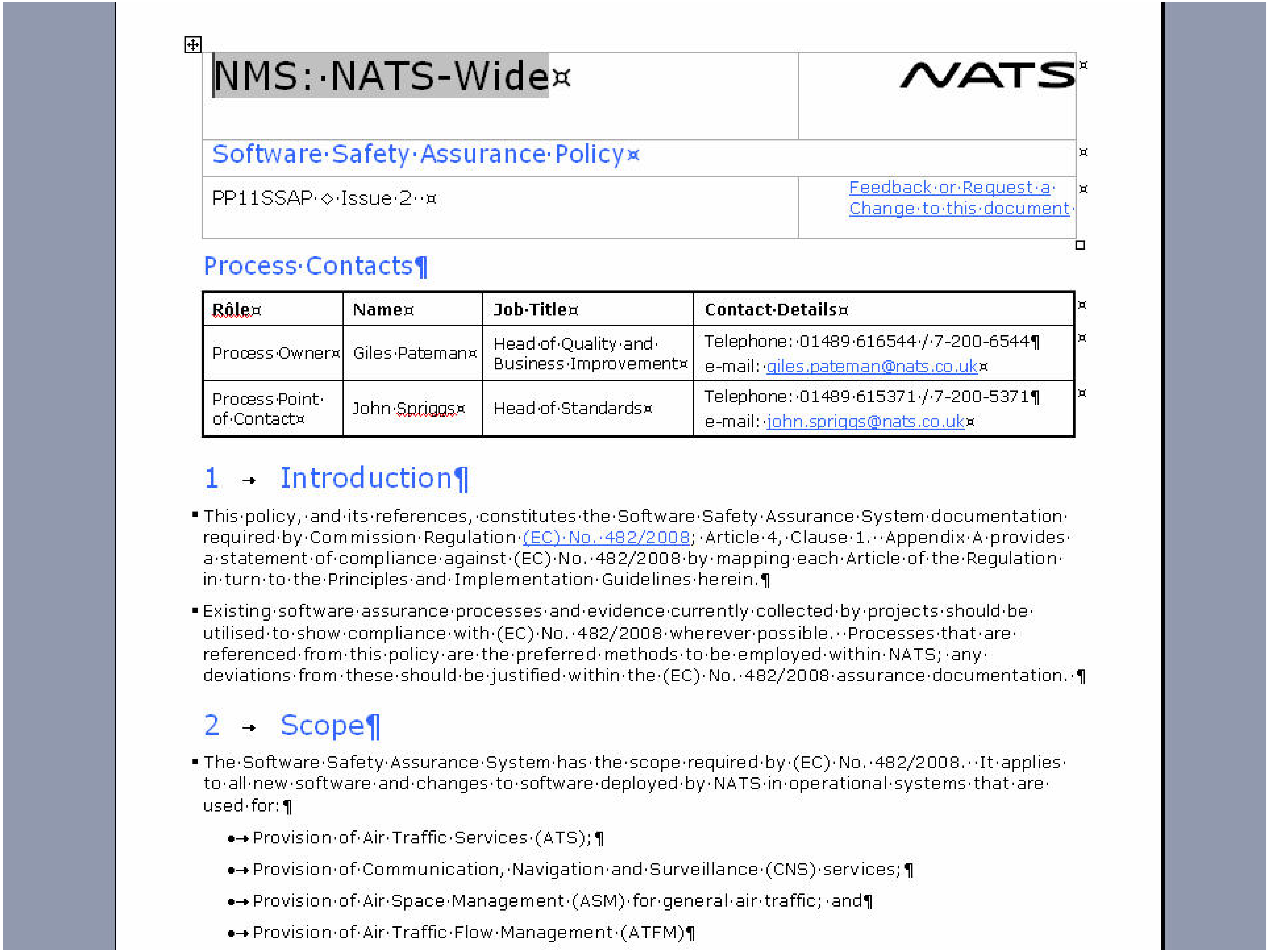
Training Material

- SSAS Training Plan
- Introduction to the SSAS
- Guide to the SSAS
- Lunch Bytes
- Software Safety Assurance for Dummies

Software Safety Assurance Policy


EC No.
482/2008
NATS Management System
SW01
Software Safety Assurance Policy
SMS
EPDL
(Engineering Product Development Lifecycle)
Software Safety Assurance Strategy
ED-109
Allocation of Software Assurance Levels
Argumentation for
'Projects'
(creating, modifying software, inc. configuration/ adaptation)

- Identifies Principles (x 9) & Implementation Guidelines (x 8)
  - Principles directly related to requirements of (EC) No. 482/2008
  - Implementation Guidelines provide interpretation for addressing Principles
- Traces the Principles, in the context of Implementation Guidelines, to requirements of (EC) No. 482/2008, so as to demonstrate compliance
- Provides a brief argument showing that, if the principles are fulfilled, our Regulator's CAP670 SW01 requirements are also met



NMS: NATS-Wide	NATS
Software Safety Assurance Policy	
PP11SSAP ◇ Issue 2	Feedback or Request a Change to this document

Process Contacts

Rôle	Name	Job Title	Contact Details
Process Owner	Giles Pateman	Head of Quality and Business Improvement	Telephone: 01489 616544 / 7-200-6544 e-mail: giles.pateman@nats.co.uk
Process Point of Contact	John Spriggs	Head of Standards	Telephone: 01489 615371 / 7-200-5371 e-mail: john.spriggs@nats.co.uk

1 Introduction

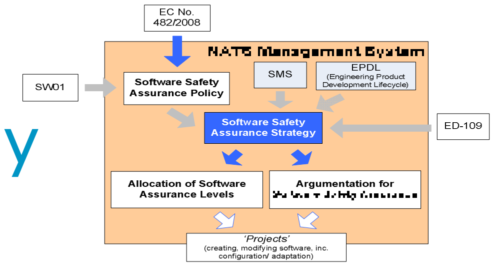
- This policy, and its references, constitutes the Software Safety Assurance System documentation required by Commission Regulation (EC) No. 482/2008; Article 4, Clause 1. Appendix A provides a statement of compliance against (EC) No. 482/2008 by mapping each Article of the Regulation in turn to the Principles and Implementation Guidelines herein.

- Existing software assurance processes and evidence currently collected by projects should be utilised to show compliance with (EC) No. 482/2008 wherever possible. Processes that are referenced from this policy are the preferred methods to be employed within NATS; any deviations from these should be justified within the (EC) No. 482/2008 assurance documentation.

2 Scope

- The Software Safety Assurance System has the scope required by (EC) No. 482/2008. It applies to all new software and changes to software deployed by NATS in operational systems that are used for:
  - Provision of Air Traffic Services (ATS);
  - Provision of Communication, Navigation and Surveillance (CNS) services;
  - Provision of Air Space Management (ASM) for general air traffic; and
  - Provision of Air Traffic Flow Management (ATFM)

Software Safety Assurance Strategy



EC No. 482/2008

NATS Management System

SW01

Software Safety Assurance Policy

SMS

EPDL (Engineering Product Development Lifecycle)

Software Safety Assurance Strategy

ED-109

Allocation of Software Assurance Levels

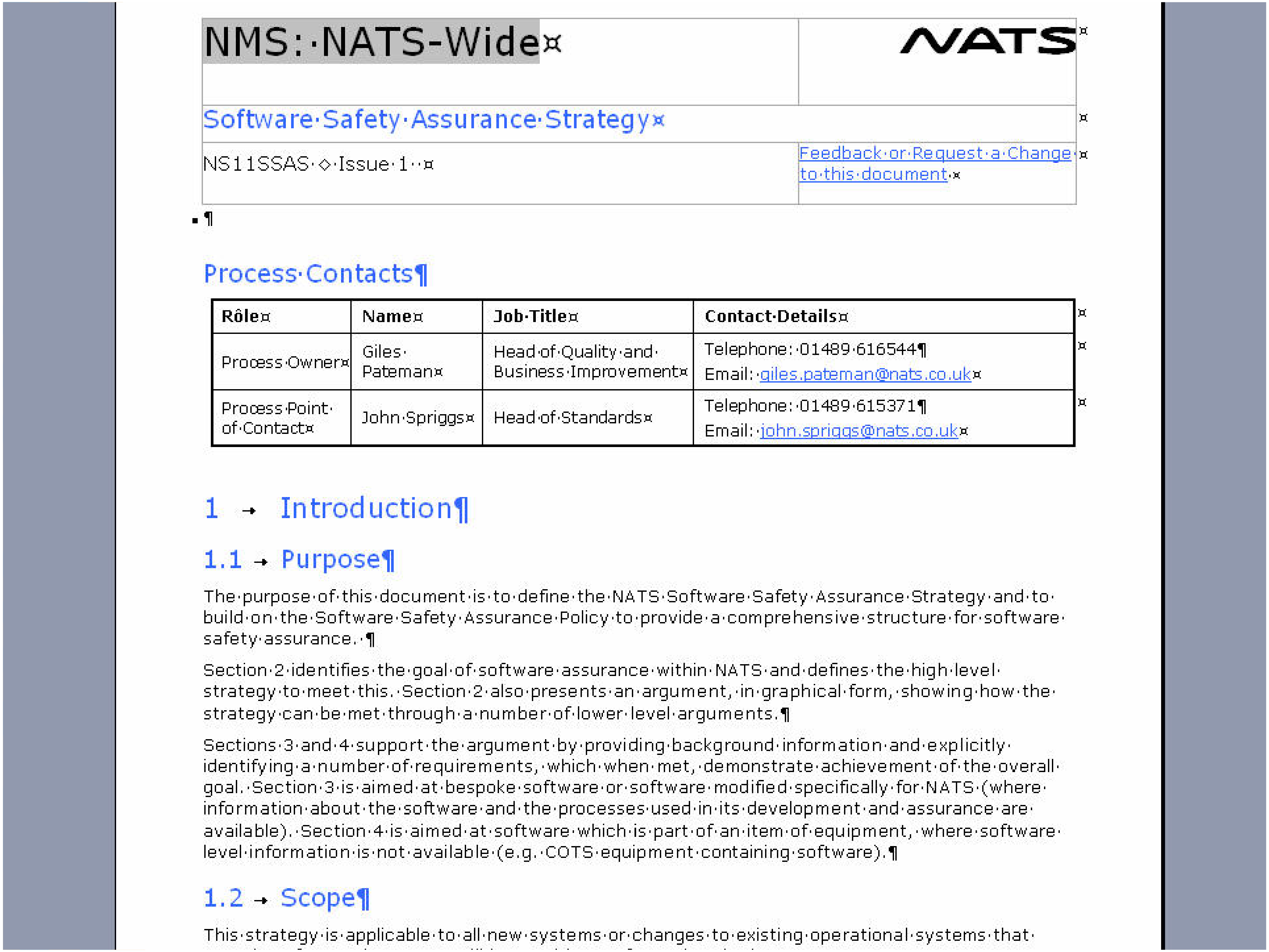
Argumentation for Software Safety Assurance

'Projects' (creating, modifying software, inc. configuration/ adaptation)

- Defines what is required to demonstrate that the *safety risk due to deploying the software is tolerable*

- Identifies a 'goal'; and a strategy to achieve that goal, from which 18 requirements are derived
  - Via an 'assurance argument'
  - Provides background and intent for each requirement

- Requirements addressed by:
  - Extant processes (e.g. Safety Management System, Software Development Processes)
  - Allocation of Software Assurance Levels (NATS SSAS Process)
  - Argumentation for Software Safety Assurance (NATS SSAS Process)



NMS: NATS-Wide

NATS

Software Safety Assurance Strategy

NS11SSAS ◇ Issue 1

[Feedback or Request a Change to this document]

Process Contacts

Rôle	Name	Job Title	Contact Details
Process Owner	Giles Pateman	Head of Quality and Business Improvement	Telephone: 01489 616544 Email: giles.pateman@nats.co.uk
Process Point of Contact	John Spriggs	Head of Standards	Telephone: 01489 615371 Email: john.spriggs@nats.co.uk

1 → Introduction

1.1 → Purpose

The purpose of this document is to define the NATS Software Safety Assurance Strategy and to build on the Software Safety Assurance Policy to provide a comprehensive structure for software safety assurance.

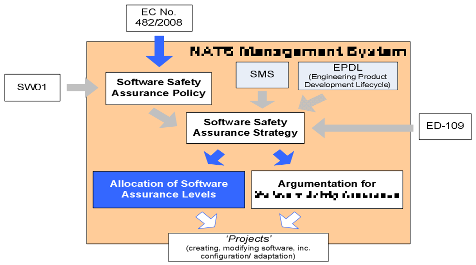
Section 2 identifies the goal of software assurance within NATS and defines the high level strategy to meet this. Section 2 also presents an argument, in graphical form, showing how the strategy can be met through a number of lower level arguments.

Sections 3 and 4 support the argument by providing background information and explicitly identifying a number of requirements, which when met, demonstrate achievement of the overall goal. Section 3 is aimed at bespoke software or software modified specifically for NATS (where information about the software and the processes used in its development and assurance are available). Section 4 is aimed at software which is part of an item of equipment, where software level information is not available (e.g. COTS equipment containing software).

1.2 → Scope

This strategy is applicable to all new systems or changes to existing operational systems that

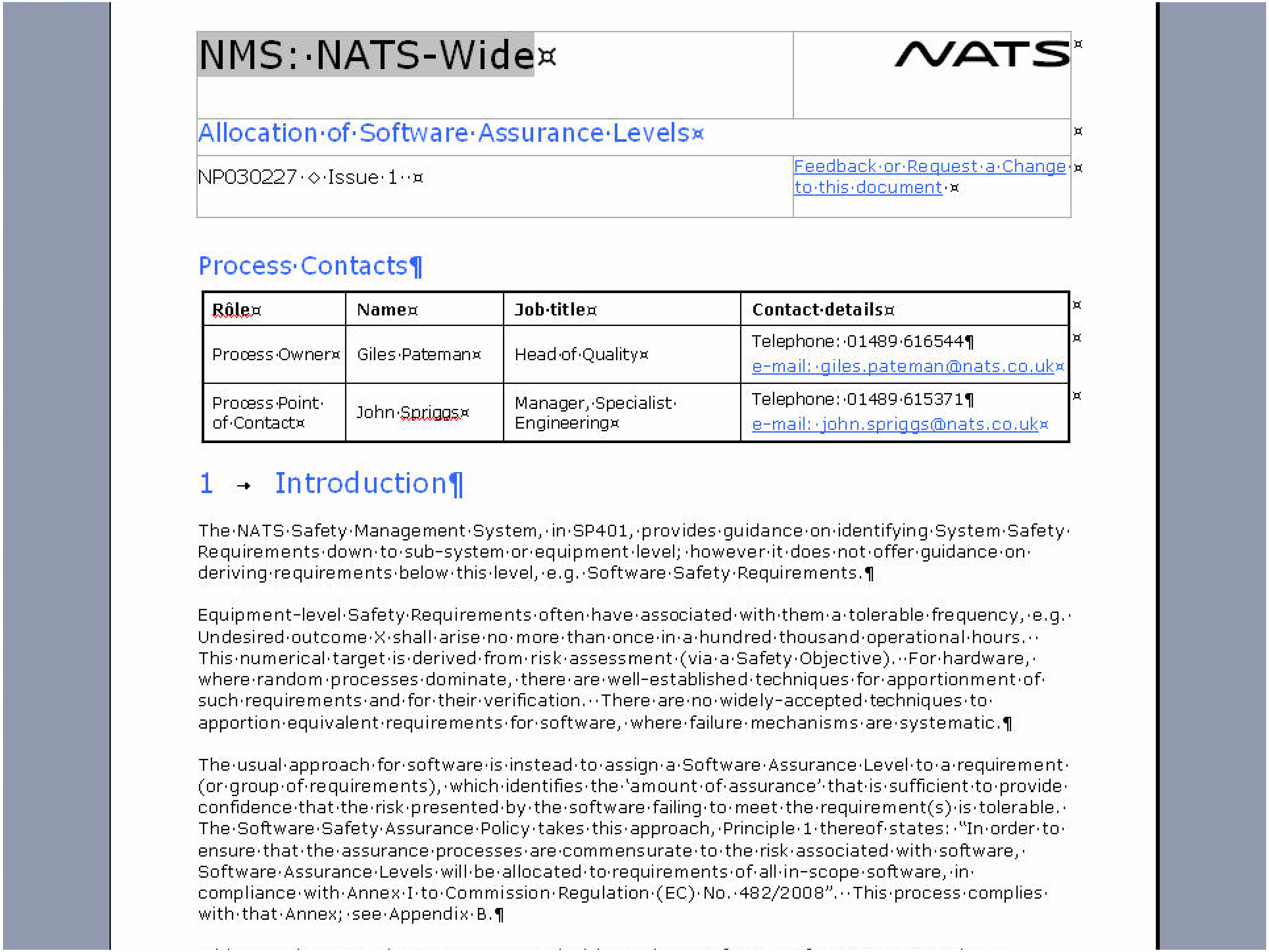
Allocation of Assurance Levels


EC No.
482/2008
NATS Management System
SW01
Software Safety Assurance Policy
SMS
EPDL
(Engineering Product Development Lifecycle)
Software Safety Assurance Strategy
ED-109
Allocation of Software Assurance Levels
Argumentation for
'Projects'
(creating, modifying software, inc. configuration/ adaptation)

- Provides a process for deriving the Software Assurance Levels
  - Aligned with EUROCAE ED-109 Assurance Levels (see later)

- Describes a two stage process
  - Stage 1: Identifies a 'worst case' Assurance Level, based on system level information (used for planning)
  - Stage 2: Allocates a Software Assurance Level, based on Software Safety Requirements
    - Would usually be stated to a supplier, but in some cases we may just specify what documents need to be provided
  - Stage 2 should be revisited in light of design decisions, changes, etc

- Assurance Levels are limited to AL3 and AL4 of ED-109
  - AL5 & AL6 are "too easy" for ATM; AL1 & AL2, are "too difficult"



NMS: NATS-Wide

NATS

Allocation of Software Assurance Levels

NP030227 ◇ Issue 1

Feedback or Request a Change to this document

Process Contacts

Rôle	Name	Job title	Contact details
Process Owner	Giles Pateman	Head of Quality	Telephone: 01489 616544 e-mail: giles.pateman@nats.co.uk
Process Point of Contact	John Spriggs	Manager, Specialist Engineering	Telephone: 01489 615371 e-mail: john.spriggs@nats.co.uk

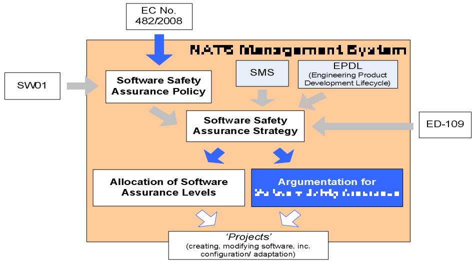
1 → Introduction

The NATS Safety Management System, in SP401, provides guidance on identifying System Safety Requirements down to sub-system or equipment level; however it does not offer guidance on deriving requirements below this level, e.g. Software Safety Requirements.

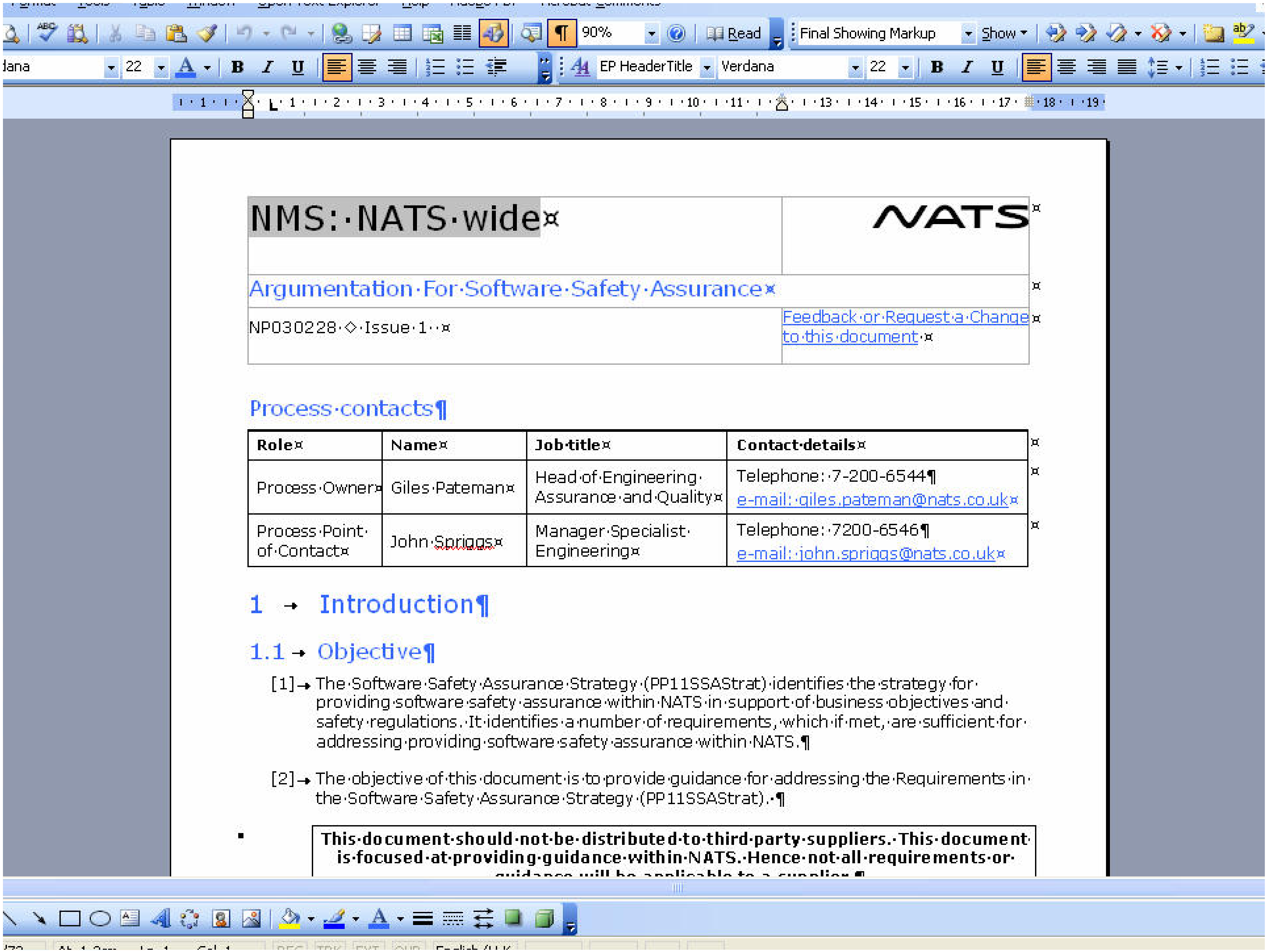
Equipment-level Safety Requirements often have associated with them a tolerable frequency, e.g. Undesired outcome X shall arise no more than once in a hundred thousand operational hours. This numerical target is derived from risk assessment (via a Safety Objective). For hardware, where random processes dominate, there are well-established techniques for apportionment of such requirements and for their verification. There are no widely-accepted techniques to apportion equivalent requirements for software, where failure mechanisms are systematic.

The usual approach for software is instead to assign a Software Assurance Level to a requirement (or group of requirements), which identifies the 'amount of assurance' that is sufficient to provide confidence that the risk presented by the software failing to meet the requirement(s) is tolerable. The Software Safety Assurance Policy takes this approach, Principle 1 thereof states: "In order to ensure that the assurance processes are commensurate to the risk associated with software, Software Assurance Levels will be allocated to requirements of all in-scope software, in compliance with Annex I to Commission Regulation (EC) No. 482/2008". This process complies with that Annex; see Appendix B.

Argumentation for Software Safety Assurance


EC No. 482/2008
NATS Management System
SW01
Software Safety Assurance Policy
SMS
EPDL (Engineering Product Development Lifecycle)
Software Safety Assurance Strategy
ED-109
Allocation of Software Assurance Levels
Argumentation for Software Safety Assurance
'Projects' (creating, modifying software, inc. configuration/ adaptation)

- Refines the assurance arguments of the Strategy for use by projects
- Provides the high level assurance argument using the Goal Structuring Notation (GSN)
- Identifies primary sources of evidence
  - Aligned with EUROCAE ED-109 evidence requirements
- Also suggests approaches for dealing with counter-evidence and the shortfalls in evidence that occur in real-world projects



NMS: NATS wide

NATS

Argumentation For Software Safety Assurance

NP030228 ◇ Issue 1

Feedback or Request a Change to this document

Process contacts

Role	Name	Job title	Contact details
Process Owner	Giles Pateman	Head of Engineering Assurance and Quality	Telephone: 7-200-6544 e-mail: giles.pateman@nats.co.uk
Process Point of Contact	John Spriggs	Manager Specialist Engineering	Telephone: 7200-6546 e-mail: john.spriggs@nats.co.uk

1 Introduction

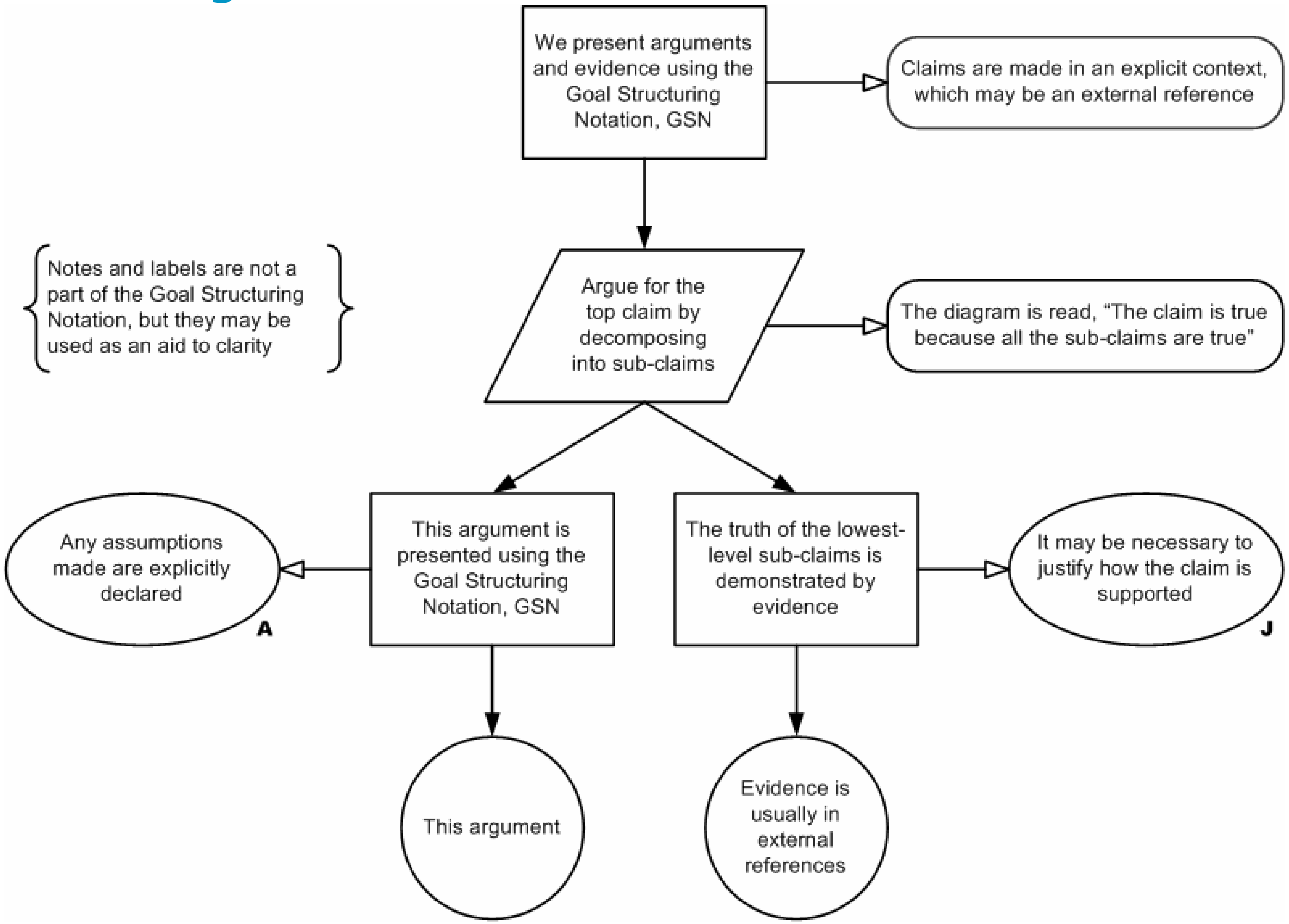
1.1 Objective

[1] The Software Safety Assurance Strategy (PP11SSAStrat) identifies the strategy for providing software safety assurance within NATS in support of business objectives and safety regulations. It identifies a number of requirements, which if met, are sufficient for addressing providing software safety assurance within NATS.

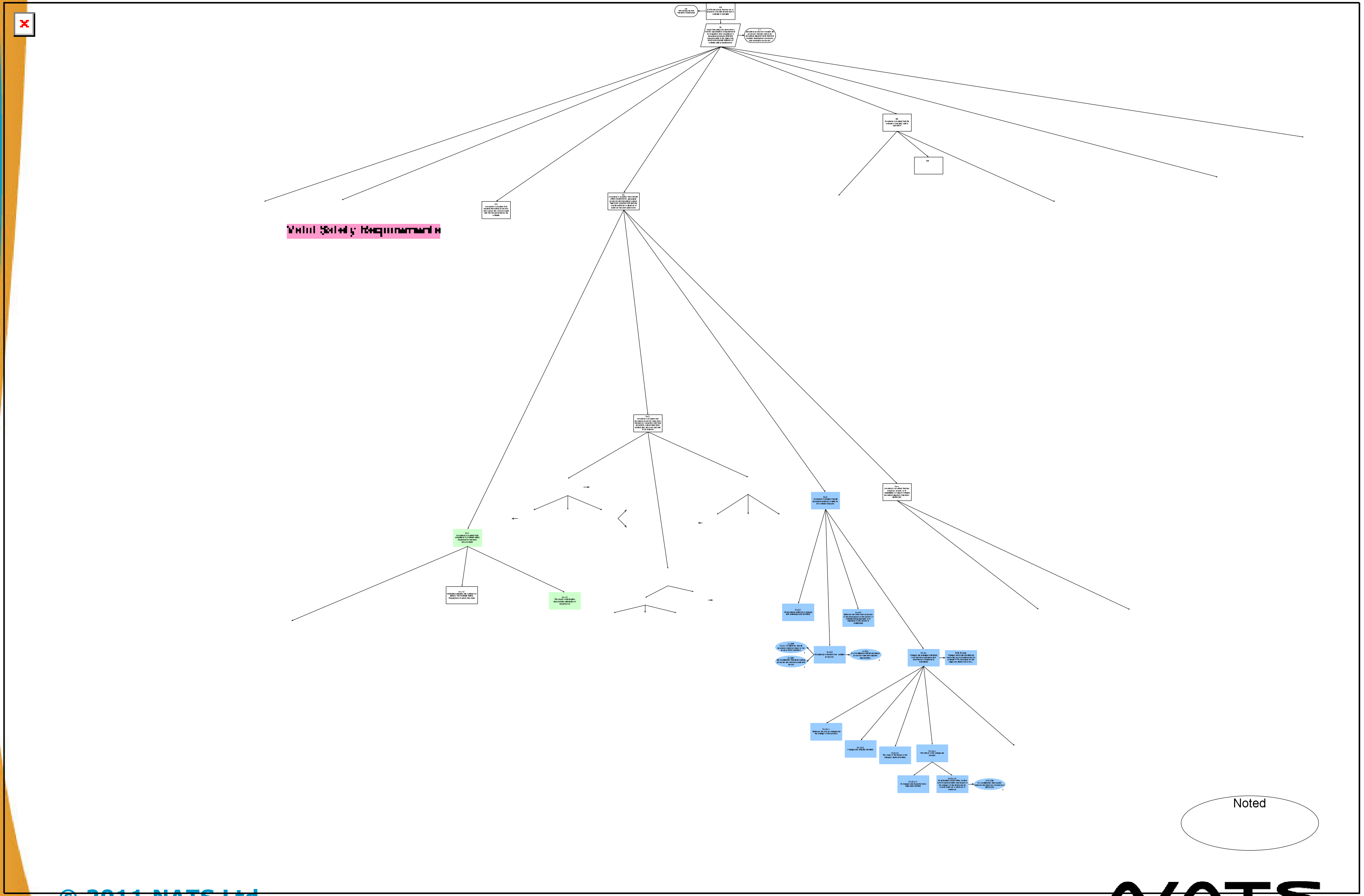
[2] The objective of this document is to provide guidance for addressing the Requirements in the Software Safety Assurance Strategy (PP11SSAStrat).

This document should not be distributed to third party suppliers. This document is focused at providing guidance within NATS. Hence not all requirements or guidance will be applicable to a supplier.

GSN Argument


We present arguments and evidence using the Goal Structuring Notation, GSN
Claims are made in an explicit context, which may be an external reference
Notes and labels are not a part of the Goal Structuring Notation, but they may be used as an aid to clarity
Argue for the top claim by decomposing into sub-claims
The diagram is read, "The claim is true because all the sub-claims are true"
Any assumptions made are explicitly declared
A
This argument is presented using the Goal Structuring Notation, GSN
The truth of the lowest-level sub-claims is demonstrated by evidence
It may be necessary to justify how the claim is supported
J
This argument
Evidence is usually in external references

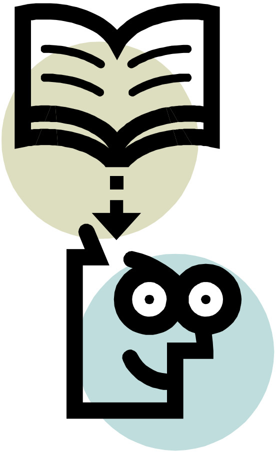
Overview of the NATS SSAS Assurance Argument

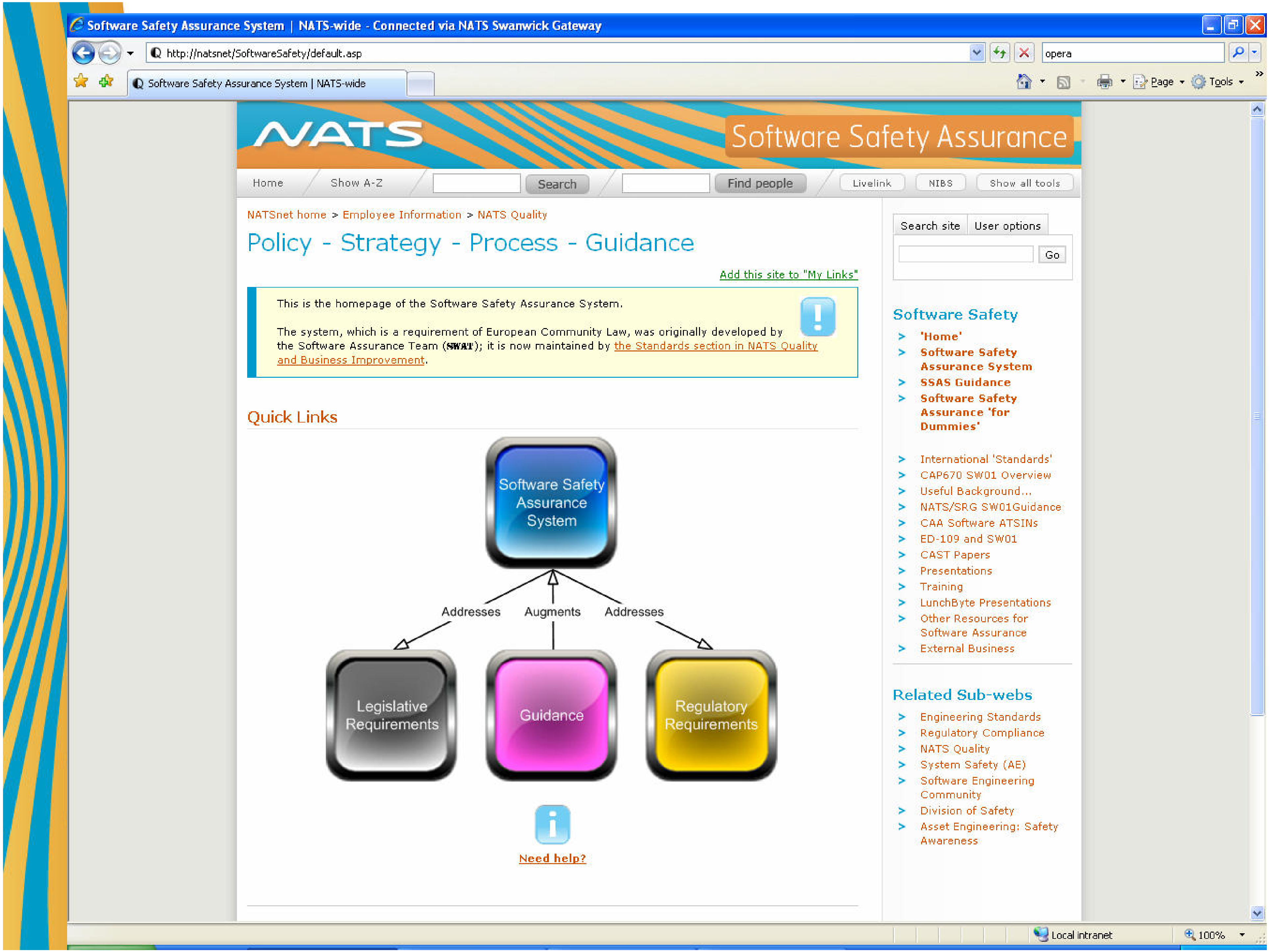

G0
NATS will ensure that the risk to provision of a safe service due to software is tolerable
C0
SMS defines the Risk Tolerability Classification
S1
Argue that safety risk assessment informs specification of requirements for mitigations and compliance to assurance processes that are commensurate to the safety risk which demonstrate fulfilment of software safety requirements
C1
assurance processes' includes all processes that are cited in an assurance argument and therefore includes development processes and verification processes
G3
Assurance is provided that required assurance processes and outputs are commensurate with the risk presented by the software
G4
Assurance is provided that software safety requirements, assurance processes and assurance outputs have been complied with and any counter evidence or absence of evidence has been addressed
G5
Assurance is provided that the software is tolerably safe in operation
G6
Valid Safety Requirements
G4.2
Assurance is provided that assurance processes have been followed by competent staff that assurance outputs have been verified and any tools used are fit for purpose
G4.4
Assurance is provided that any evidence counter to or unavailable to support software assurance argument has been addressed
Noted

© 2011 NATS Ltd

Training Provided Internally to NATS

- This is a complex subject; we introduced the SSAS to those affected by provision of training:
- Introduction to the Software Safety Assurance System
  - a single session aimed at managers and auditors
- Practitioners’ Guide to the Software Safety Assurance System
  - more than one session over several days for those who will produce the assurance
- Other short supporting lectures and papers to provide guidance on specific topics for a more general, non-specialist, audience
- Software Safety Assurance intranet sub-web




Software Safety Assurance System | NATS-wide - Connected via NATS Swanwick Gateway
http://natsnet/SoftwareSafety/default.asp
opera
Software Safety Assurance System | NATS-wide
Page
Tools
NATS
Software Safety Assurance
Home
Show A-Z
Search
Find people
Livelink
NIBS
Show all tools
Search site
User options
Go
Local intranet
100%

NATSnet home > Employee Information > NATS Quality

Policy - Strategy - Process - Guidance

Add this site to "My Links"

This is the homepage of the Software Safety Assurance System.

The system, which is a requirement of European Community Law, was originally developed by the Software Assurance Team (SWAT); it is now maintained by the Standards section in NATS Quality and Business Improvement.

Quick Links

Need help?

Software Safety

- 'Home'
- Software Safety Assurance System
- SSAS Guidance
- Software Safety Assurance 'for Dummies'
- International 'Standards'
- CAP670 SW01 Overview
- Useful Background...
- NATS/SRG SW01Guidance
- CAA Software ATSINs
- ED-109 and SW01
- CAST Papers
- Presentations
- Training
- LunchByte Presentations
- Other Resources for Software Assurance
- External Business

Related Sub-webs

- Engineering Standards
- Regulatory Compliance
- NATS Quality
- System Safety (AE)
- Software Engineering Community
- Division of Safety
- Asset Engineering: Safety Awareness

Assurance Approach for Different Types of Software

- New (bespoke) software
  - Developed specifically for NATS…
  - …explicitly covered by NATS SSAS…
  - …evidence required identified by assurance level.
- Software modified for NATS
  - Requires a modified approach (see next slide)…
- COTS
  - Assurance can be issue; but it depends what you mean by COTS…

Assurance Approaches for Software Modified for NATS

- Option 1. Treat as new.
  - When change is extensive, or very complex, it may be 'best', i.e. less time & money, to assure the whole as if it were New Software.

- Option 2. Define an Assurance 'Envelope'
  - Software is initially assured over a *range* of something (e.g. over a range of configurations)
  - Assurance evidence is provided for that *range*…
  - As long as software remains within this *range* after modification, the assurance remains valid…

- Option 3. Limit Scope of Impact

  1: Assure what has changed (as for New)

  2: Assure what has been impacted by the change

  3: Assure that there has not been any inadvertent regression

COTS

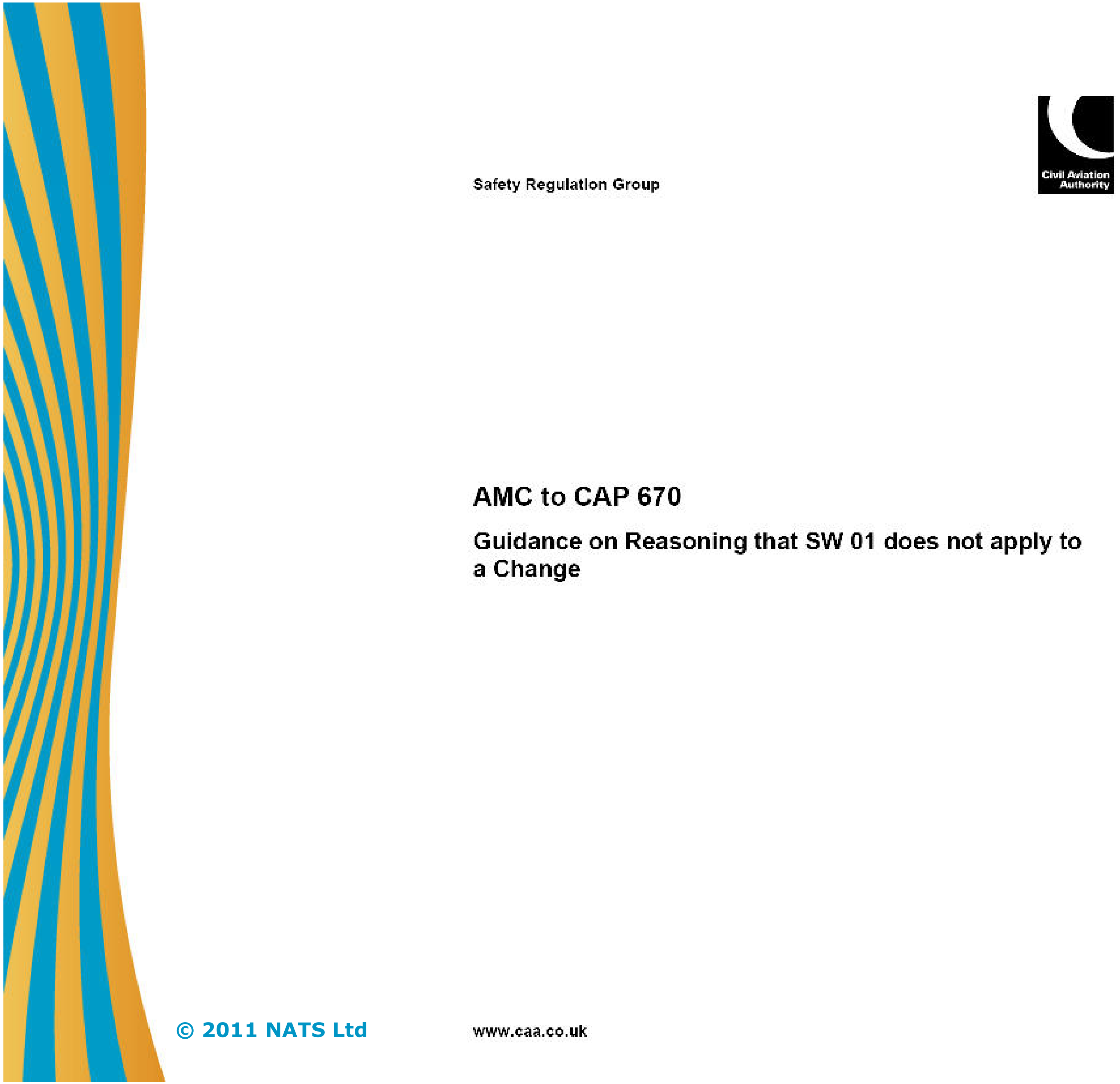
- "COTS" is a term open to widely varying interpretations
- It represents a continuum
  - From…high volume (many 1,000s), largely market independent
    - e.g. Network Routers, Operating Systems
  - …through…medium volume (many 100s), few defined markets
    - e.g. Radar Sensors, Navigation Beacons
  - …to low volume (10s), ATM market specific
    - E.g. Voice Communications System, Surveillance Data Processor
- Our main concern is availability of assurance evidence
  - High volume: rarely is any assurance evidence available; we are not a key Customer, the supplier is not interested
  - Medium volume: depending on the supplier, they *may* be prepared to provide information
  - Low volume: NATS likely to be a key customer, so opportunity to work with supplier is a viable option

Specific Guidance for COTS

- NATS and CAA have agreed a method of assurance
  - Limited to **equipment**, where most onerous ‘integrity’ requirement is $1x10^{-5}$ failures per operational hour per sector
    - Assurance is at the equipment level
    - Any software within the equipment is considered assured
    - Explicitly excludes assurance of the software in isolation
    - Meets regulatory & legislative requirements – does not mean that it is safe!
- To use it, we need to meet five pre-requisites
  - Safety Requirements are all expressed in terms of COTS equipment outputs
  - Safety Objectives have been set at an acceptable level of risk
  - The most onerous integrity requirement on the COTS equipment is no worse than 1×10^{-5} (per operational hour)
  - All equipment outputs mentioned in the Safety Requirements are observable
  - Equipment in-service monitoring requirements are specified in the associated System Safety Case

COTS Guidance Elements (cont.)

- Integrity Assurance
  - Testing [FAT, SAT, Soak, Training, Supplier Test]
    - Test script, Test Results, Traceability Matrix, Evidence of use in training
  - Field Service [equivalent usage]
    - Same system/same platform, Earlier system/same platform, Similar system/similar platform (OS/HW), Same system/previous platform (OS/HW), Similar system by same supplier (Build statement, Observed Failures, Environment)
  - Supplier Experience and Reputation
    - Same system type into ATC market (Evidence of Track Record)
    - Personnel involved have expert knowledge (CVs)
  - Supplier Software Design and Development Process
    - Demonstration of appropriate process/standard (Certificate of Conformance, Independent Audit)
    - Knowledge of internal design features (Design documentation)
- Functional Assurance
  - Testing [FAT, SAT]
    - Test Scripts, Test Results, Traceability Matrix

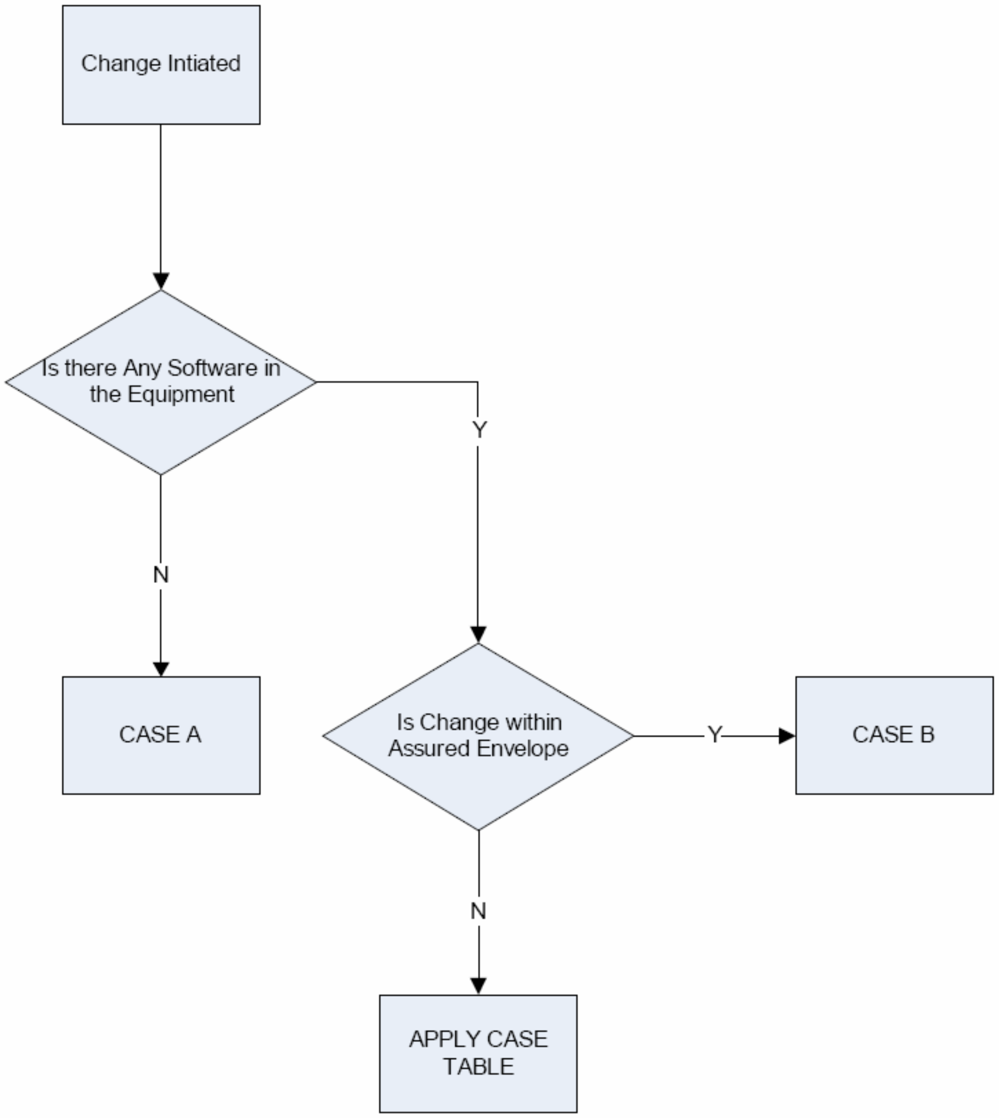


Safety Regulation Group

AMC to CAP 670

Guidance on Reasoning that SW 01 does not apply to a Change

www.caa.co.uk



Change Intiated

Is there Any Software in the Equipment

Y

N

CASE A

Is Change within Assured Envelope

Y

CASE B

N

APPLY CASE TABLE

Case Table

Not Safety Related		Safety Related		Case
HW change?	SW Change/ Impact?	HW change?	SW Change/ Impact?	
No	No	No	Yes	SW 01 argument[2]
No	No	Yes	No	C
No	No	Yes	Yes	SW 01 argument
No	Yes	No	No	E
No	Yes	No	Yes	SW 01 argument
No	Yes	Yes	No	D + E[3]
No	Yes	Yes	Yes	SW 01 argument
Yes	No	No	No	C
Yes	No	No	Yes	SW 01 argument
Yes	No	Yes	No	C
Yes	No	Yes	Yes	SW 01 argument
Yes	Yes	No	No	D + E
Yes	Yes	No	Yes	SW 01 argument
Yes	Yes	Yes	No	D + E
Yes	Yes	Yes	Yes	SW 01 argument

4 Case A: There is no Software in the Changed Equipment

4.1 Introduction

4.1.1 This claim can be used where there is no Software in the equipment.

4.1.2 Three alternative arguments are presented below. The first and second arguments are where an inspection/review is undertaken by a competent person. The third argument is where a formal statement from a supplier is available and the ANSP and its supplier are unable to complete the previous two arguments.

Note: It is preferable to comply with paragraph 4.2.1 or 4.2.2 by requiring the Supplier to provide the necessary documentation.

4.2 Arguments

4.2.1 Equipment has been Physically Inspected

Claim0 The change does not necessitate an argument that the SW 01 objectives have been met because there is no Software in the changed equipment.

Argument

IF A Competent Person [EVIDENCE2] has inspected the equipment and confirmed that the components cannot contain software [EVIDENCE1].

THEN An argument against SW 01 is not required.

Evidence1 Record of examination, by Competent Person, of physical equipment referencing items examined, including:

a) Model, version number and serial number of the equipment examined.

b) Statement signed by the Competent Person that the inspection was completed (components were identified down to a level where it is apparent that they cannot contain software).

c) Brief description of the examination and an overview of the components found.

Evidence2 Statement arguing that the person has relevant competence for the task. The argument should define relevant competence criteria (e.g. training, experience), including at least one from the following list, according to the task being undertaken by the Competent Person[1]:

a) Has a formal qualification in the discipline in which they are being asked to review, e.g. electrical engineering; or

b) Has experience in lieu of formal qualifications.

5 Case B: Change is within the Assured Envelope

5.1 Introduction

5.1.1 This claim can be used when the Safety Case identifies an Assured Envelope[5] and when the change has been implemented, the equipment remains within its Assured Envelope. This may include adaptation changes.

5.2 Argument

Claim0 The change does not necessitate an argument that the SW 01 objectives have been met because the change is within the Assured Envelope.

Argument

IF The equipment has an Assured Envelope identified [EVIDENCE1] and it has been adequately assured [EVIDENCE2].

AND The equipment remains within the Assured Envelope following the change [EVIDENCE3].

AND The equipment has been configured as intended [EVIDENCE 4].

AND The general behaviour of the equipment has not unexpectedly changed [EVIDENCE5].

AND All evidence presented in support of this argument either relates directly to the version of the equipment for which assurance is sought [EVIDENCE6] **OR** arguments are presented to justify why evidence from previous version(s) of the equipment remain valid [EVIDENCE7].

THEN An argument against SW 01 is not required.

Evidence1 A document which identifies the Assured Envelope parameters and their ranges. This may be the system safety case, other system documentation or, if such documentation does not currently exist, an agreed, documented, system expert's opinion. The source of the information should be stated.

If a system expert's opinion is used, this must be supported by an argument that the person has relevant competence for the task.

Evidence2 Evidence of assurance of the Assured Envelope. Currently this may include Test Plans, Test Scripts and Test Results from **previous** version(s) of the equipment, identifying the testing that has been completed along with its success/failure. [Refer to all appropriate plans, scripts and results, for example regression, site and installation][6].

Safety Regulation Group

Acceptable Means of Compliance to CAP 670 SW 01

Guidance for Producing SW 01 Safety Arguments for COTS Equipment

www.caa.co.uk

2 Assurance Approach

2.1 Overview

This section provides a description of the steps that an ANSP needs to follow when applying this Guidance. The steps are listed below and each one is expanded upon in the following sections:

Step 1: Set valid Safety Requirements

Step 2: Present Arguments that the Conditions for the use of the Guidance are met

Step 3: Present Arguments that the SW 01 objectives are satisfied

Step 4: Present evidence underpinning the argument

Step 5: Claim compliance with this AMC

2.4 Step 3 – Present Arguments that the SW 01 Objectives are Satisfied

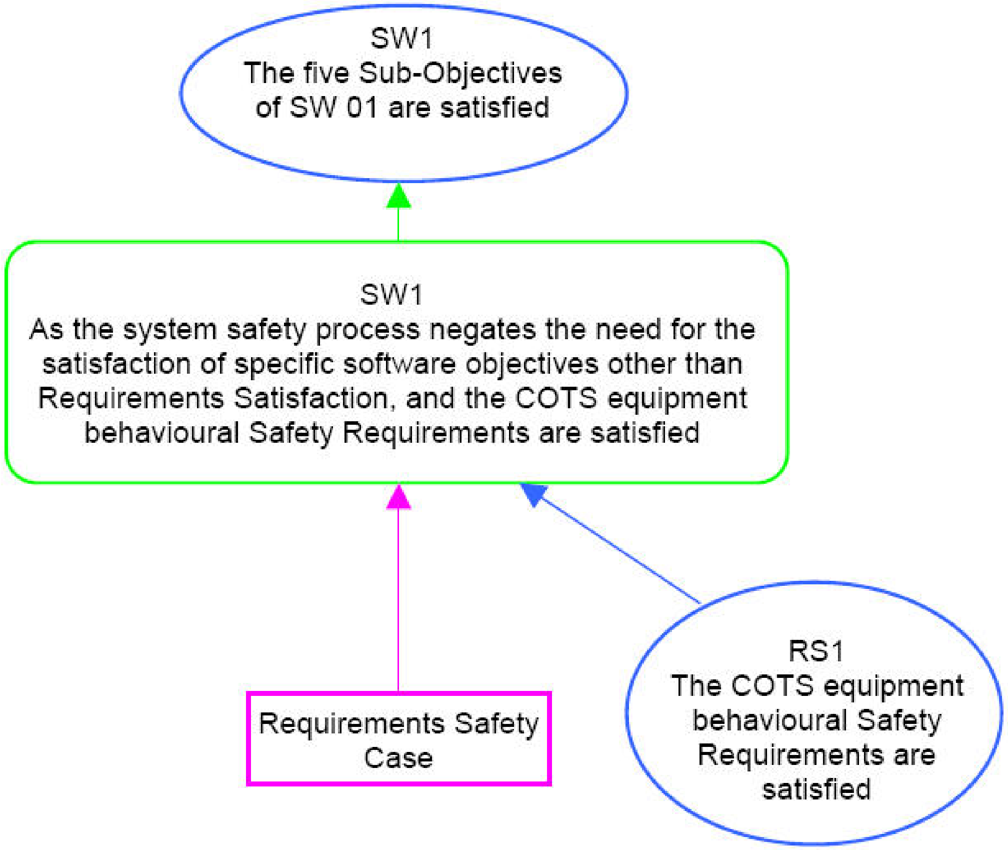
Using the templates provided in this guidance, the ANSP presents the argument that the objectives of SW 01 have been met. This Guidance provides arguments (the rationale for which can be found in Annex K of this document), illustrated by Claim, Argument and Evidence (CAE) diagrams as follows:

- Annex B covers the Arguments and CAEs for software in equipments with Safety Requirements no more onerous than 1×10^{-4}, for COTS equipment that meet the conditions of paragraph 2.3.

- Annexes C to G cover the Arguments and CAEs for software in equipments with Safety Requirements no more onerous than 1×10^{-5}, for COTS equipment that meet the conditions of paragraph 2.3.

ANNEX B ARGUMENT DIAGRAM TEMPLATE FOR 1 X 10^{-4} REQUIREMENTS

SW1 The Safety Objective of SW 01 is satisfied



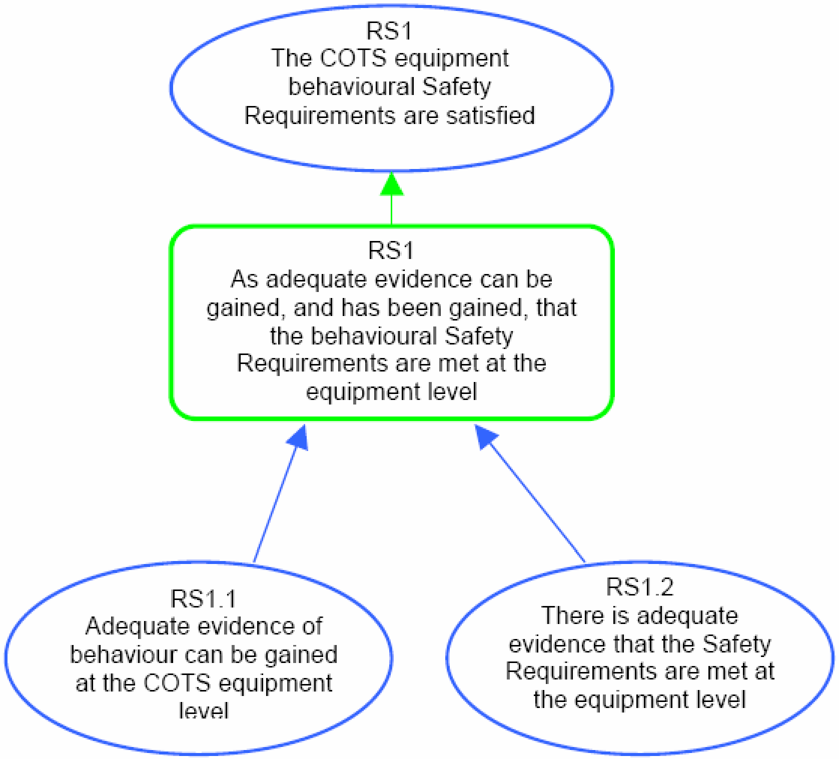
SW1
The five Sub-Objectives of SW 01 are satisfied

SW1
As the system safety process negates the need for the satisfaction of specific software objectives other than Requirements Satisfaction, and the COTS equipment behavioural Safety Requirements are satisfied

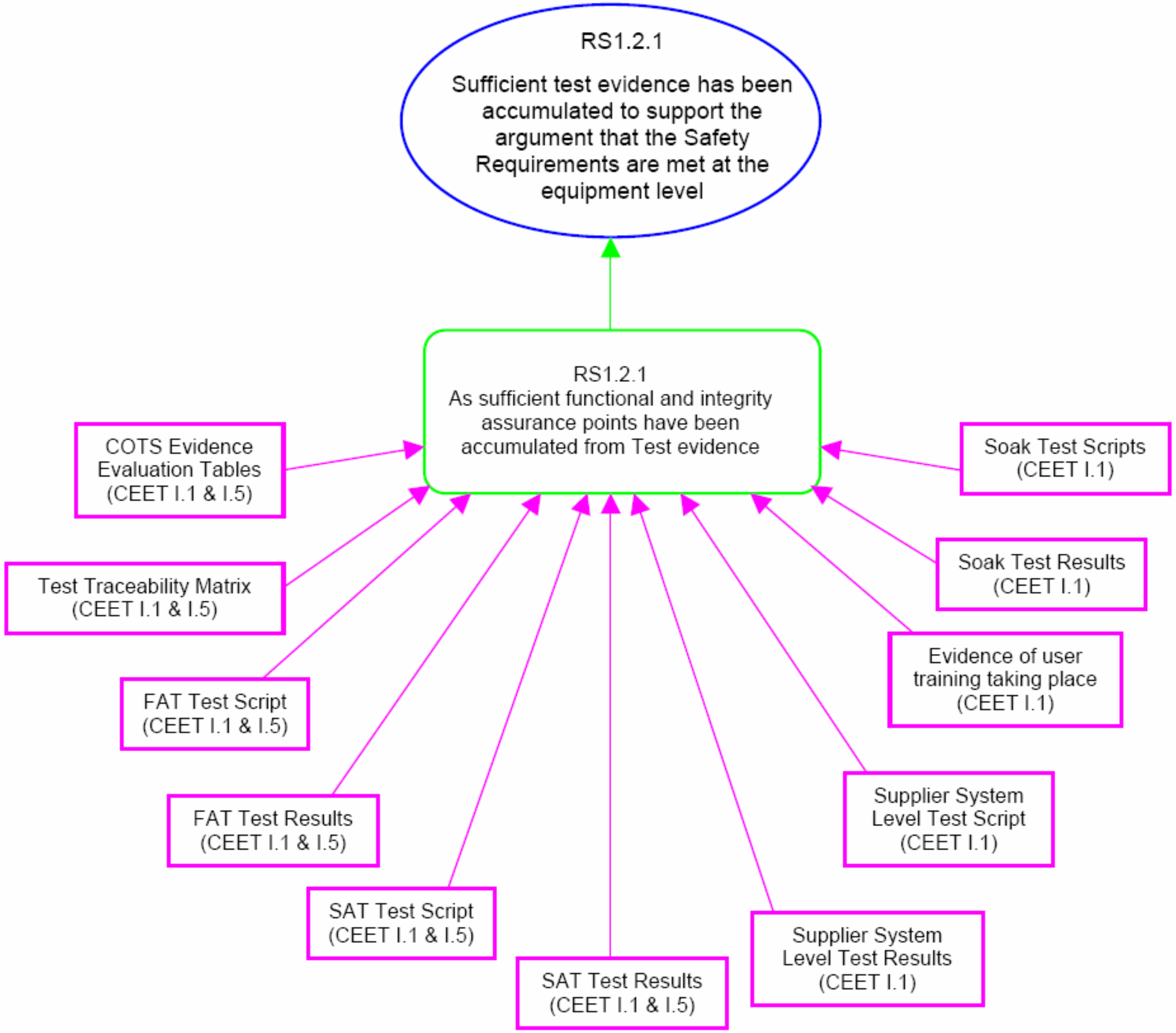
Requirements Safety Case

RS1
The COTS equipment behavioural Safety Requirements are satisfied

RS1 The COTS equipment behavioural Safety Requirements are satisfied


RS1
The COTS equipment behavioural Safety Requirements are satisfied
RS1
As adequate evidence can be gained, and has been gained, that the behavioural Safety Requirements are met at the equipment level
RS1.1
Adequate evidence of behaviour can be gained at the COTS equipment level
RS1.2
There is adequate evidence that the Safety Requirements are met at the equipment level

RS1.2.1 Test Evidence


RS1.2.1
Sufficient test evidence has been accumulated to support the argument that the Safety Requirements are met at the equipment level
RS1.2.1
As sufficient functional and integrity assurance points have been accumulated from Test evidence
COTS Evidence Evaluation Tables (CEET I.1 & I.5)
Test Traceability Matrix (CEET I.1 & I.5)
FAT Test Script (CEET I.1 & I.5)
FAT Test Results (CEET I.1 & I.5)
SAT Test Script (CEET I.1 & I.5)
SAT Test Results (CEET I.1 & I.5)
Supplier System Level Test Results (CEET I.1)
Supplier System Level Test Script (CEET I.1)
Evidence of user training taking place (CEET I.1)
Soak Test Results (CEET I.1)
Soak Test Scripts (CEET I.1)

ANNEX I CEET REQUIREMENTS NO MORE ONEROUS THAN 1 X 10^{-4}

The COTS Evidence Evaluation Table (CEET) for Requirements no more onerous than 1 x 10^{-4} is split into a number of tables that address the functional and integrity assurance aspects of the Safety Requirements. The tables presented are:

a) Integrity assurance:

- Testing: Table I.1
- Field service: Table I.2
- Supplier experience and reputation: Table I.3
- Supplier Software design and development: Table I.4

b) Functional assurance: Table I.5

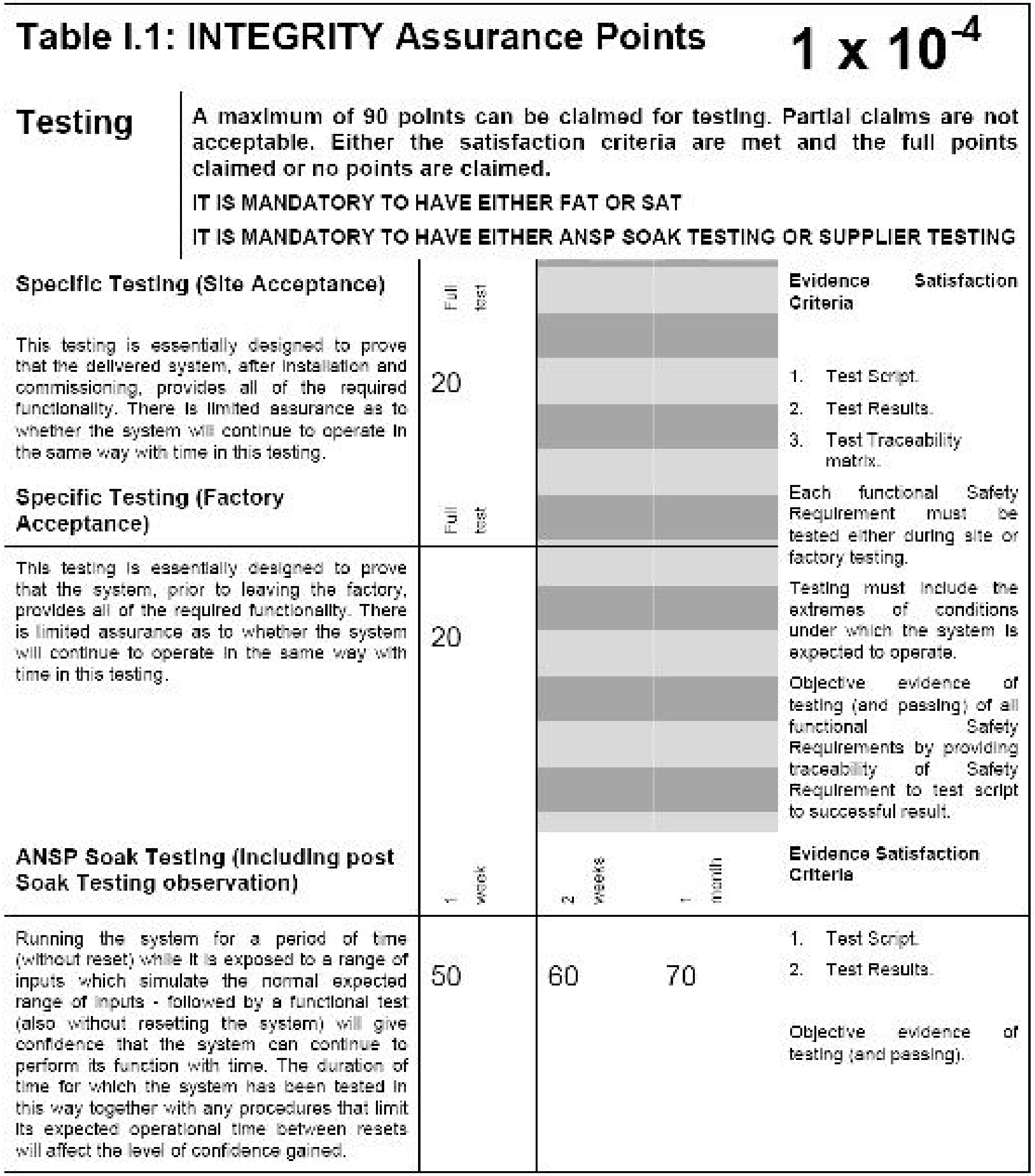


Table I.1: INTEGRITY Assurance Points 1×10^{-4}

Testing

A maximum of 90 points can be claimed for testing. Partial claims are not acceptable. Either the satisfaction criteria are met and the full points claimed or no points are claimed.

IT IS MANDATORY TO HAVE EITHER FAT OR SAT

IT IS MANDATORY TO HAVE EITHER ANSP SOAK TESTING OR SUPPLIER TESTING

Specific Testing (Site Acceptance)	Full test			**Evidence Satisfaction Criteria**
This testing is essentially designed to prove that the delivered system, after installation and commissioning, provides all of the required functionality. There is limited assurance as to whether the system will continue to operate in the same way with time in this testing.	20			1. Test Script. 2. Test Results. 3. Test Traceability matrix.
Specific Testing (Factory Acceptance)	Full test			Each functional Safety Requirement must be tested either during site or factory testing.
This testing is essentially designed to prove that the system, prior to leaving the factory, provides all of the required functionality. There is limited assurance as to whether the system will continue to operate in the same way with time in this testing.	20			Testing must include the extremes of conditions under which the system is expected to operate. Objective evidence of testing (and passing) of all functional Safety Requirements by providing traceability of Safety Requirement to test script to successful result.
ANSP Soak Testing (Including post Soak Testing observation)	1 week	2 weeks	1 month	**Evidence Satisfaction Criteria**
Running the system for a period of time (without reset) while it is exposed to a range of inputs which simulate the normal expected range of inputs - followed by a functional test (also without resetting the system) will give confidence that the system can continue to perform its function with time. The duration of time for which the system has been tested in this way together with any procedures that limit its expected operational time between resets will affect the level of confidence gained.	50	60	70	1. Test Script. 2. Test Results. Objective evidence of testing (and passing).

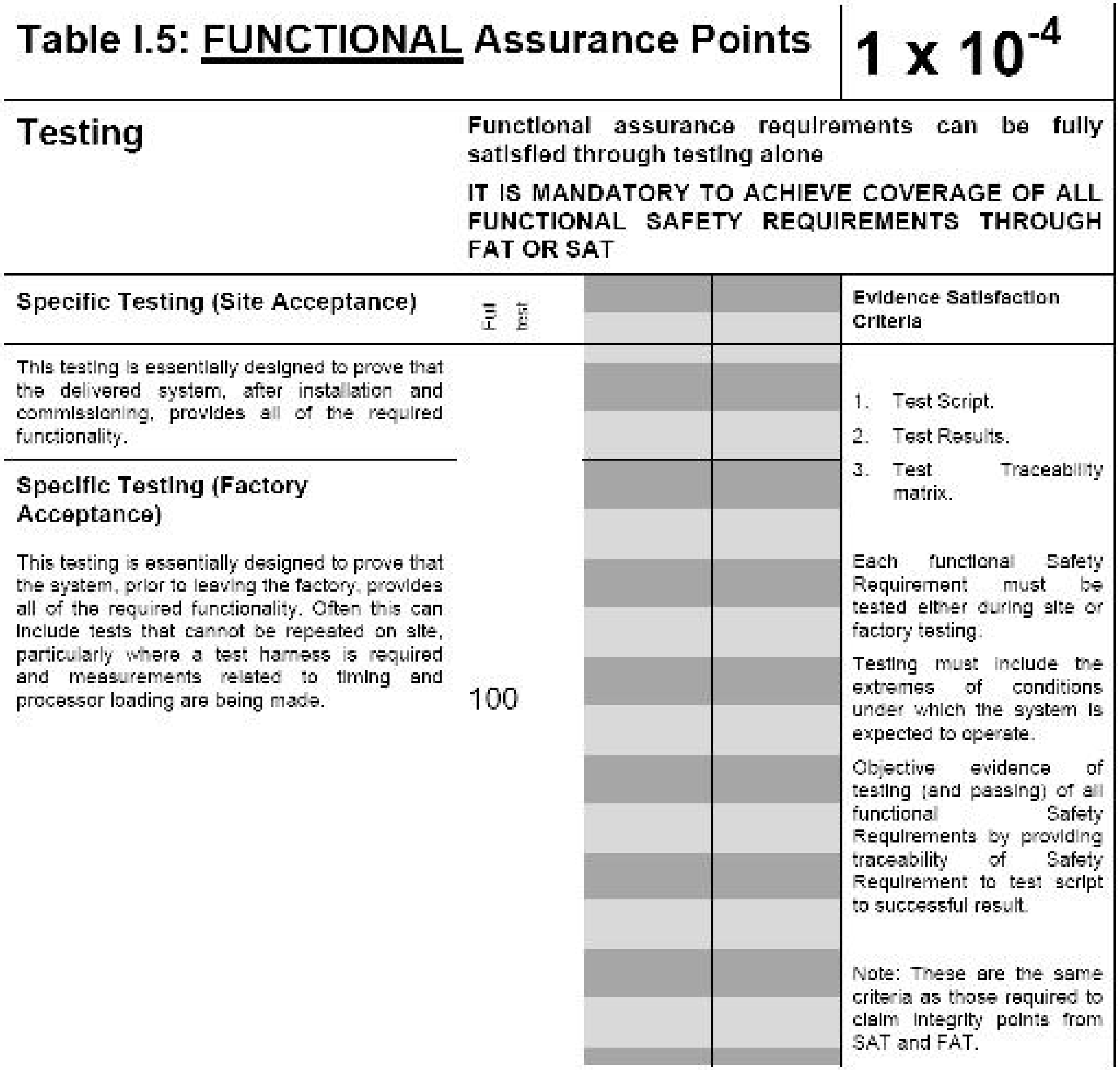


Table I.5: FUNCTIONAL Assurance Points				1×10^{-4}
Testing	**Functional assurance requirements can be fully satisfied through testing alone** **IT IS MANDATORY TO ACHIEVE COVERAGE OF ALL FUNCTIONAL SAFETY REQUIREMENTS THROUGH FAT OR SAT**			
Specific Testing (Site Acceptance)	Full test			**Evidence Satisfaction Criteria**
This testing is essentially designed to prove that the delivered system, after installation and commissioning, provides all of the required functionality.				1. Test Script. 2. Test Results. 3. Test Traceability matrix. Each functional Safety Requirement must be tested either during site or factory testing. Testing must include the extremes of conditions under which the system is expected to operate. Objective evidence of testing (and passing) of all functional Safety Requirements by providing traceability of Safety Requirement to test script to successful result. Note: These are the same criteria as those required to claim integrity points from SAT and FAT.
Specific Testing (Factory Acceptance) This testing is essentially designed to prove that the system, prior to leaving the factory, provides all of the required functionality. Often this can include tests that cannot be repeated on site, particularly where a test harness is required and measurements related to timing and processor loading are being made.	100			

Table I.5 Functional assurance from 100% test coverage of functional Safety Requirements

Customer and Supplier

- NATS has experience of safety in ATM
- Suppliers have experience of their systems
  - Although system integrators may have little detailed knowledge of the software in what they are integrating…
    - real example: "CCTV camera system has no software"
- NATS experience is that best assurance comes where strengths of both are used
  - NATS develops assurance argument
  - Suppliers deliver most of the assurance evidence
  - Need to work closely together to ensure no duplication of effort - and no gaps in the assurance!
- You can use the SSAS assurance argument structure to divide work between the ANSP and their suppliers
  - Some assurance may have to be supplied by all parties, e.g. fitness for purpose of their Configuration Management system

Amount of Assurance

- The Software Assurance Level identifies the ‘amount of assurance’ that is sufficient to provide confidence that the risk is tolerable.

- For example, you have three building projects:
  - A dog kennel – a failure may injure some family members
  - A private house – a failure may kill someone and injure several more people
  - A skyscraper – a failure may kill many people and injure many more

- Would you approach the assurance of all these buildings the same way?

Choice of approach

- There are three methods from which to choose:
  - Do it yourself
    - Analogous to a Low Integrity Development Process

  - Use a local builder
    - Analogous to a Medium Integrity Development Process

  - Use an architect and a contractor experienced in the required building systems
    - Analogous to a High Integrity Development Process

- Which would you use for each project?

How to vary 'Amount of Assurance'?

- The three main things that you can vary are:
  - What you do, e.g.:
    - Requirements → Code **OR** Requirements → Design → Code
    - Analysis and focussed regression testing after a change **OR** 100% retest
  - How you do it, e.g.:
    - Measure code coverage as statement coverage **OR** as branch coverage
    - Implement in C++ with bought-in libraries **OR** use a tightly-controlled subset of a strongly-typed language
  - The degree of independence of those who check you have done it, e.g.:
    - Test team drawn from your development team, as the project focus changes **OR** from an independent external test company
    - Use your suppliers' own internal audits **OR** send in your auditors

NATS

EUROCAE ED-109

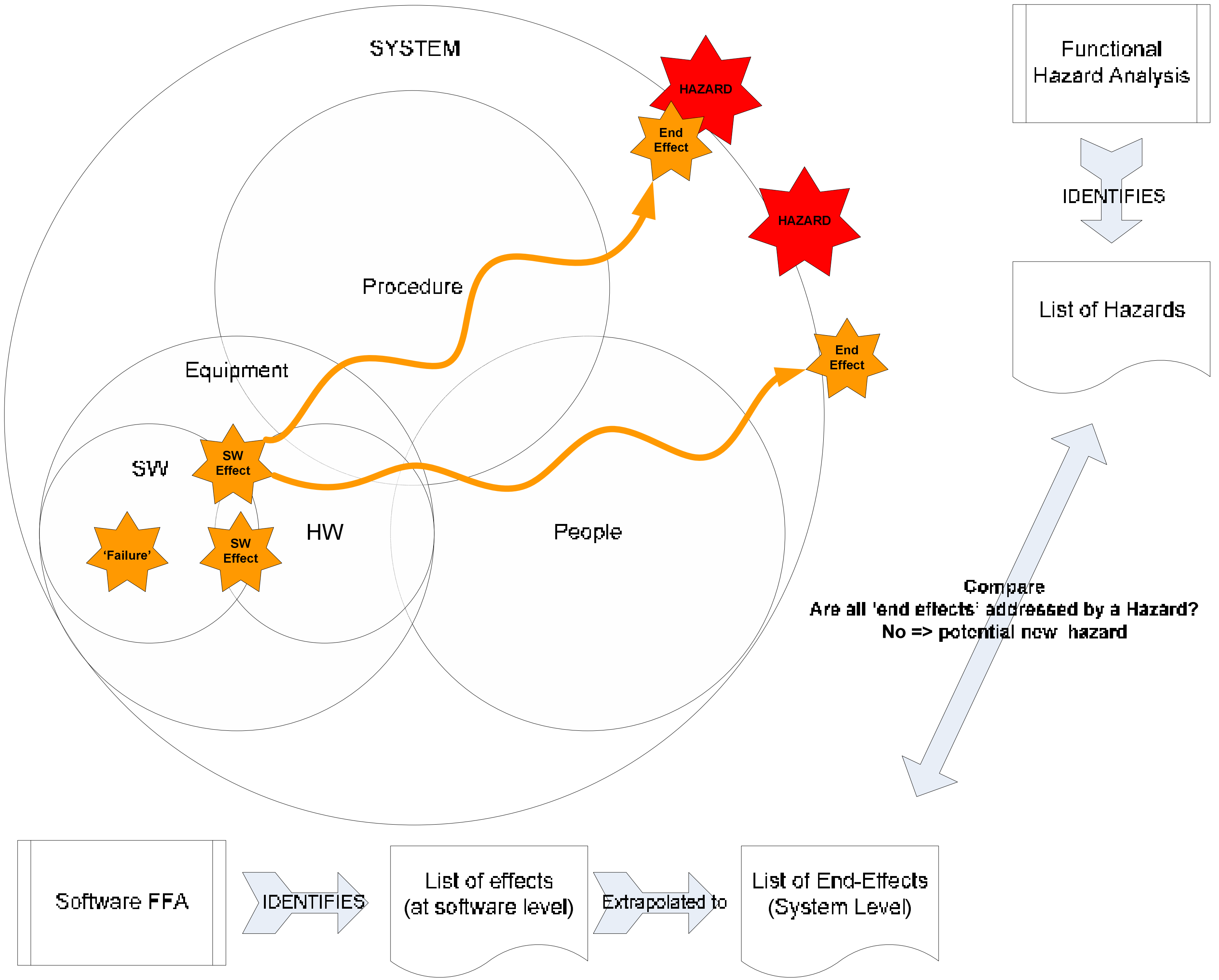
- NATS has aligned its Software Safety Assurance System Assurance Levels with those of EUROCAE ED-109
- ED-109 provides guidance, for each Assurance Level, about the software development processes required to produce the necessary evidence
- ED-109 provides a level of guidance that, if complied with, will provide ~80% of the assurance evidence required by the NATS SSAS…
  - … but it does not provide everything needed by the requirements of (EC) No. 482/2008 and CAP670 SW01 …
  - … however, the NATS SSAS identifies the 'missing' elements and provides additional requirements for:
    - Functional Failure Analysis
    - Staff Competency

NATS SSAS and ED-109 Assurance Levels

Assurance Level	ED-109 Assurance Level requirements (SSAS additions in bold)	
AL5	• Software plans defined • High-level software requirements defined and traceable to system level requirements • Executable compliant and robust to high-level software requirements • Adaptation data defined • Test coverage of high-level software requirements achieved • SCM and change control processes applied • SQA processes applied • Software Approval processes applied • Tools qualified	Outside scope of SSAS [Insufficient assurance for Air Traffic Services]
AL4	AL5 AND… • Software development standards are defined • High-level, derived software requirements and software architecture conform to Software development standards • Test procedures shown to be correct • Test discrepancies addressed • Test coverage of software structure achieved • **Software Architecture Functional Failure Analysis** • **Competency of key staff addressed**	In scope of SSAS
AL3	AL4 AND… • Low-level software requirements defined, conform to design standards and traceable • Source Code complies with architecture, conforms to coding standards and is traceable • Executable is compliant and robust with respect to low-level software requirements • Test coverage of low-level software requirements is achieved • 100% Statement coverage achieved • **Software Design Functional Failure Analysis**	
AL2	AL3 AND… • 100% decision coverage demonstrated • Many objectives satisfied with greater independence	Not used [too costly]
AL1	AL2 AND… • 100% Modified condition / decision coverage demonstrated	

Functional Failure Analysis

- Functional Failure Analysis provides a means of validating the system-level hazard identification
  - Note that this is neither a Failure Modes & Effects Analysis nor a Hazard & Operability Study, but combines elements from both
- For each operating mode of the software, Functional Failure Analysis systematically addresses each function to be implemented, and identifies credible malfunctions using ‘prompts’
  - These prompts were derived both from the behavioural attributes specified in (EC) No. 482/2008 and from a UK Defence Standard
  - The analysis assesses the effect of each identified malfunction at the software boundary
- This approach captures effects arising from multiple failures, or from unknown sources, and so is better than Failure Modes & Effects Analysis, which only considers one failure mode of one component at a time, and assumes everything else is as required



SYSTEM

Procedure

Equipment

SW

HW

People

‘Failure’

SW Effect

SW Effect

End Effect

End Effect

HAZARD

HAZARD

Functional Hazard Analysis

IDENTIFIES

List of Hazards

Compare
Are all 'end effects' addressed by a Hazard?
No => potential new hazard

Software FFA

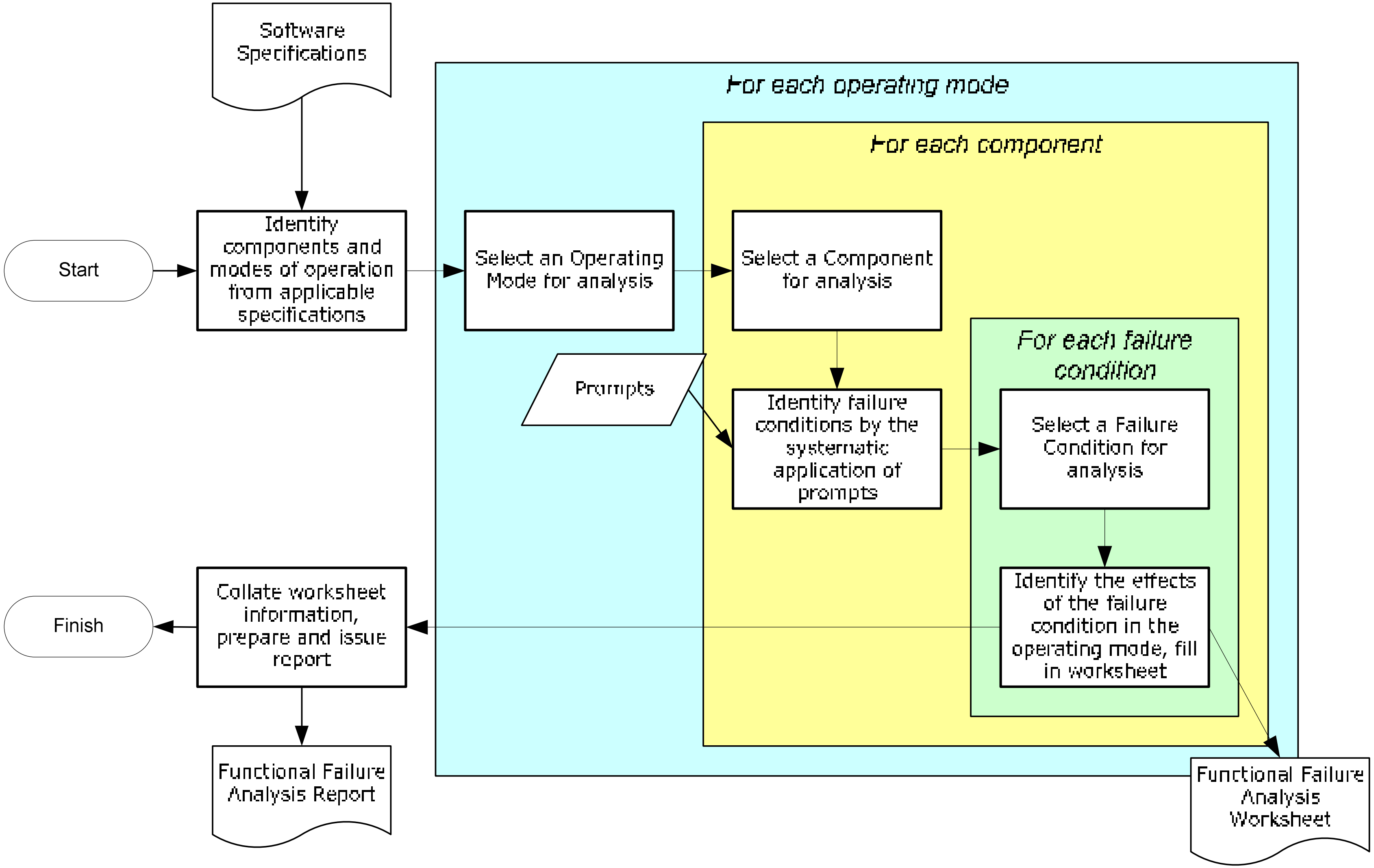
IDENTIFIES

List of effects (at software level)

Extrapolated to

List of End-Effects (System Level)

FFA Process


Software Specifications
Start
Identify components and modes of operation from applicable specifications
For each operating mode
Select an Operating Mode for analysis
Prompts
For each component
Select a Component for analysis
Identify failure conditions by the systematic application of prompts
For each failure condition
Select a Failure Condition for analysis
Identify the effects of the failure condition in the operating mode, fill in worksheet
Functional Failure Analysis Worksheet
Collate worksheet information, prepare and issue report
Finish
Functional Failure Analysis Report

In conclusion…
The Benefits of a Software Safety Assurance System

» It gives us a higher level of confidence that we have controlled the risk of deploying complex software in the operational system

» It enables us to claim compliance to Commission Regulation (EC) No. 482/2008, and provide assurance to the National Supervisory Authority as required

» It enables relatively simple read-across arguments to persuade our national Regulator that we have met their requirements for software assurance (CAP670 SW01)

» Rather than spending time 're-inventing the wheel' for each project, we can concentrate on controlling risk…

NATS

Questions?

DOES ANYONE HAVE A QUESTION DESIGNED TO INCREASE MY WORKLOAD FOR YOUR ENTERTAINMENT?
www.dilbert.com
scottadams@aol.com
©2007 Scott Adams, Inc./Dist. by UFS, Inc.
5-10-07