# Developing Safety Evidence Profiles for Software of Uncertain Pedigree

Dr Rick Vinter, Principal Safety Consultant
14th April 2016

SCSC

CGI



Experience the commitment®

# Agenda

1. Rationale
2. Guidelines
3. Methodology
4. Aim and Scope
5. Safety Argument Structure
6. Selection of Criteria
7. Checklist
8. Distribution
9. Acceptance Criteria
10. Trusted Software
11. Constraints on Use of Evidence
12. Managing Assurance Deficits

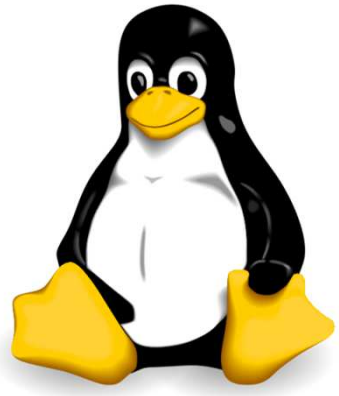


CGI

# Rationale (1)

- "*It is generally impractical to build complex systems from the most rudimentary parts"* and *"to make use of major subassemblies that have been previously developed*" (IEC 61508)
- Safety related systems development uses pre-existing software
  - E.g. OS, library, database, device driver
  - May have been developed for non-safety application
  - May be standalone COTS or integrated subcomponent
  - May not contain adequate mitigation for new application
  - May contain non-required functionality
  - May be changed frequently by many third parties







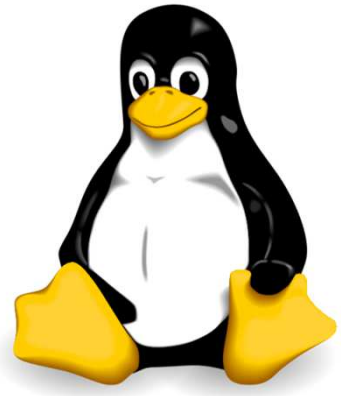
CGI

# Rationale (2)

- Pre-existing software can be treated as:
  - Software of Uncertain Pedigree (SOUP)
- SOUP can reduce development time and increase reliability if:
  - ✓ Used extensively in previous similar applications
  - ✓ Sufficient in-service history or field evidence is available
- But
  - Proven-in-use ≠ Fault-free
  - Proven for previous applications ≠ Suitable for new application









CGI

# Guidelines

- In 2001 Adelard developed guidance for safety assurance of SOUP
- HSE offers this as an illustration of a principled approach to:
  1. Gather evidence on the performance of SOUP
  2. Apply that evidence in the IEC 61508 framework
  3. Construct a systematic and transparent argument for the safety integrity of a specified safety function
- Originally intended for deriving MTBFs and SIL assessment
- CGI has adapted this approach to support safety assurance arguments for pre-existing software across industry sectors

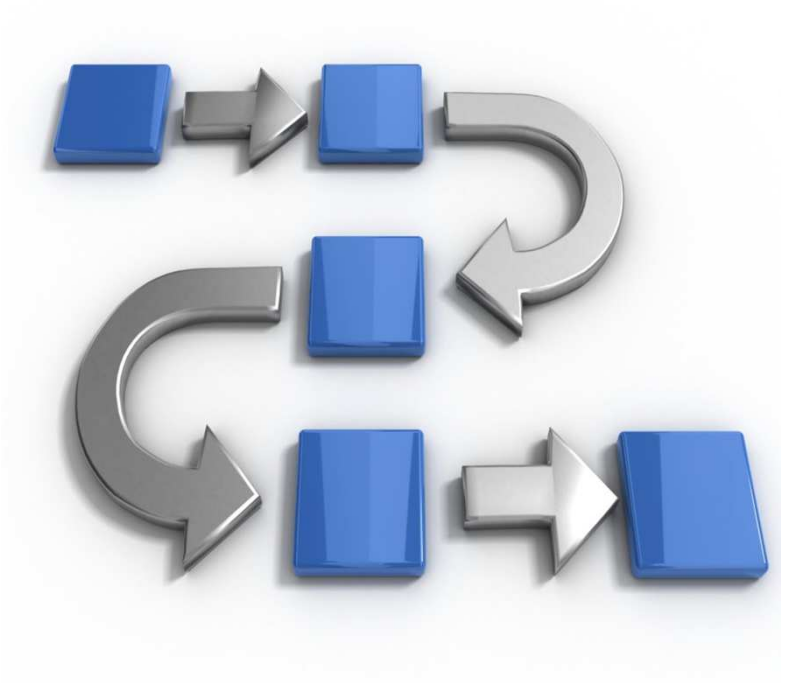



CGI

# Methodology

## 5 Step Approach

1. Define checklist criteria
2. Distribute checklist
3. Analyse completed response
4. Identify assurance deficits
5. Use evidence



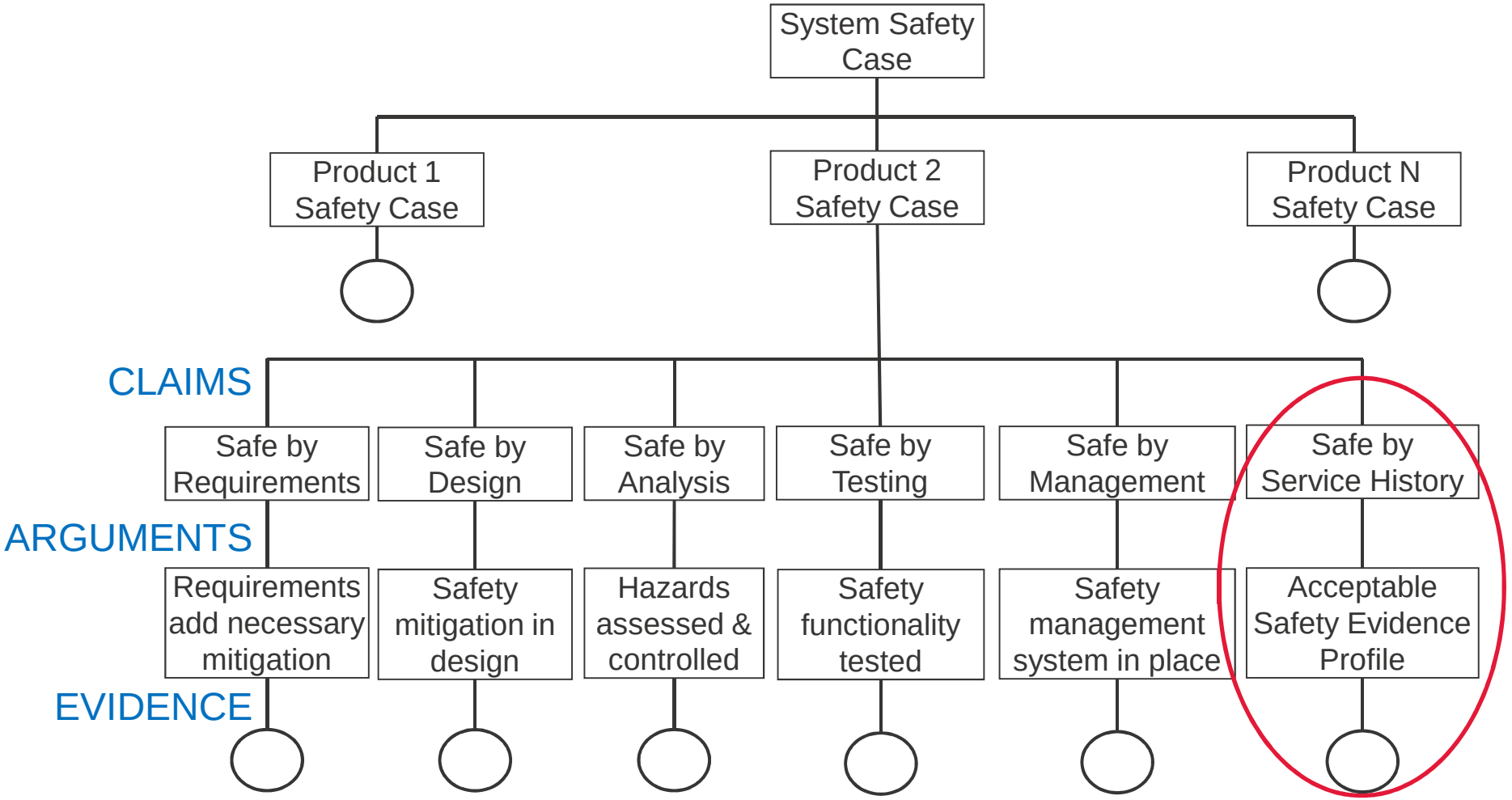
# Aim and Scope

- System Safety Case presents an independently auditable reasoned argument for a specific release of a system (comprising the product) in a specific environment based on objective evidence
- Checklist evidence supports claims for the product's safety functions
- Software Product Safety Qualification checklist
  - **Product** – Pharos
  - **Version** – Release 2.0
  - **Purpose** – High performance event storage and retrieval. Logging up to 50k events per second continuously
  - **Prior Applications** – Telecoms and security event logging
  - **Target Application** – DCC Smart Metering Implementation




CGI

# Safety Argument Structure

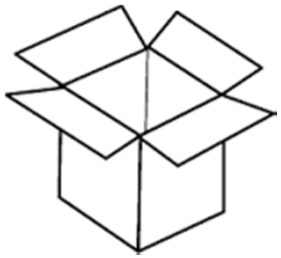





CGI

# Selection of Criteria (1)

- Checklist template tailored for target application
- Designed to generate evidence for product's required safety claims
- Black or white box criteria
  - Proportionate to safety risks and requirements
  - Design knowledge of internal product structure is available?
  - Higher risk or integrity ⇒ More white box evidence required

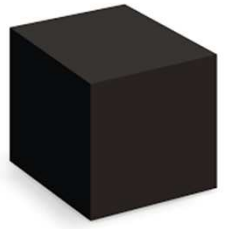




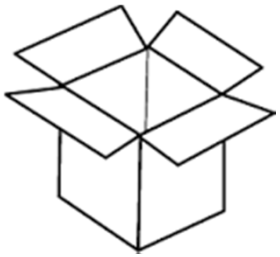
CGI

# Selection of Criteria (2)

- **Black box** test or field evidence may suffice for some claims
  - From observation in similar environments
  - E.g. certifications, operational service history, throughput capacity, failsafe attributes, integration test data, safety related incidents, reliability metrics



- **White box** analytic evidence fully clarifies pedigree
  - From analysis of internal product structure
  - E.g. open source code, known defects, development process documents, fully traceable requirements, module test data, non-required functions





CGI

# Checklist (1)

## Preliminary Questions

| Your Name and Role | Date |
|---|---|
| | |
| **Purpose of Product** | **Product Type (Bespoke / COTS / MOTS)** |
| | |

## Evidence Profile

| 1 Product Overview and Licensing |
|---|
| 1.1 Where was the product developed? (i.e. country of origin) |
| 1.2 Size of product? (i.e. approximate Lines Of Code) |
| 1.3 Does the license restrict use of the product as part of a safety related system? |
| 1.4 Supplier's experience in business sector? |
| 1.5 Supplier's experience with development of high integrity or safety related software? |
| 1.6 Could modification or configuration affect the supplier's warranty? |
| 1.7 Does the warranty allow for alerting of users to significant problems or urgent fixes? |



CGI

# Example: License Restrictions

***"In particular, the SOFTWARE PRODUCT is not designed for use in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation systems, direct life support machines, or weapons systems, in which the failure of the SOFTWARE PRODUCT could lead directly to death, personal injury, or severe physical or environmental damage. The entire risk as to the use, quality, and performance of the SOFTWARE PRODUCT is with the LICENSEE".***

"*3. RESTRICTIONS. You acknowledge that the Software is developed for general use in a variety of information management applications; it is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use the Software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle disclaims any express or implied warranty of fitness for such uses*".



CGI

# Checklist (2)

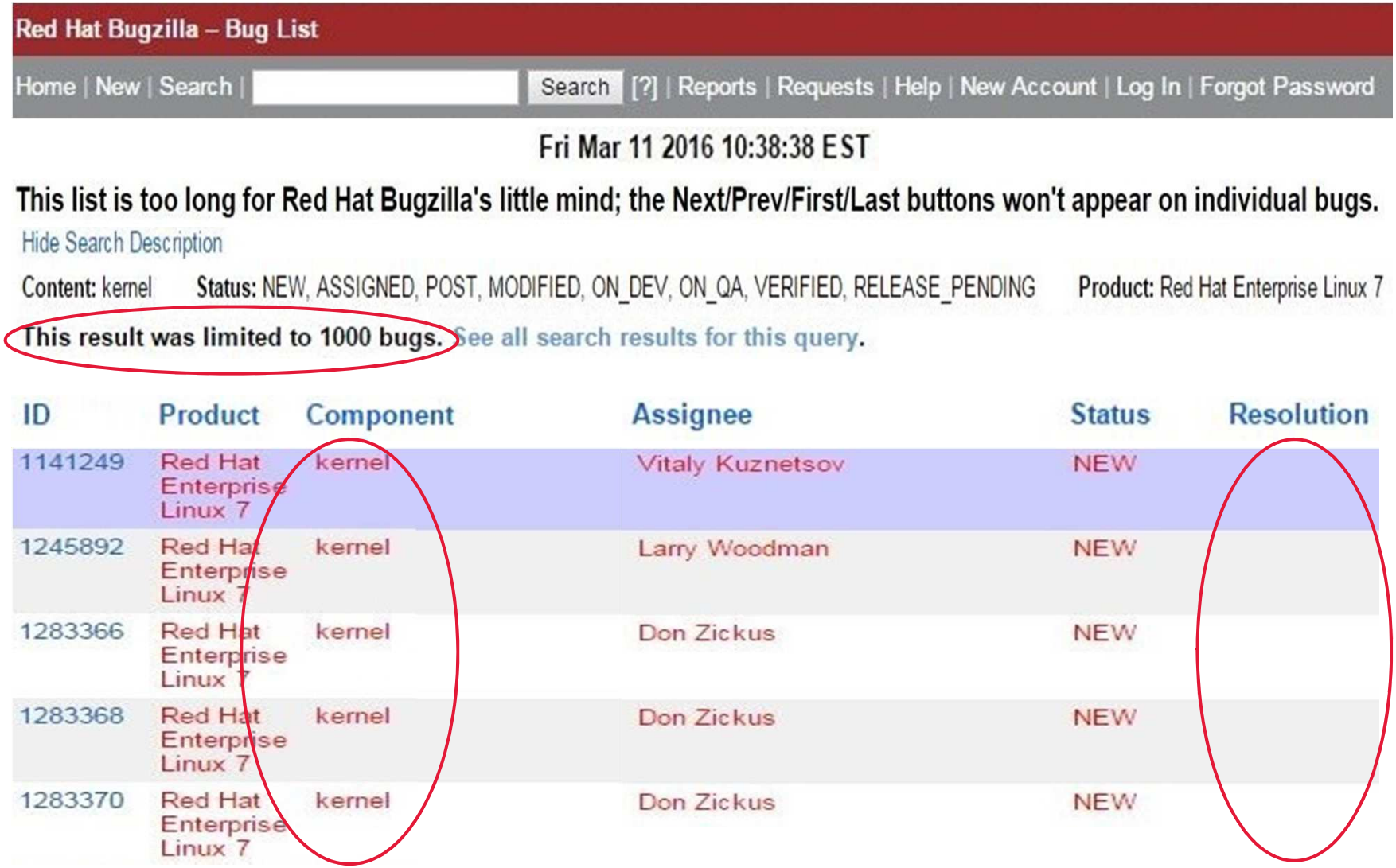
| 2 Field Experience |
| --- |
| 2.1 How long has the product been available? |
| 2.2 Approximate number of users or licenses sold? |
| 2.3 Used in a safety related system for similar client or environment? |
| 2.4 Is there a Problem Reporting, Tracking and Corrective Action system in place? |
| 2.5 Who is the point of contact for problem management? |
| 2.6 Are problem reports made available to users? (e.g. bug list available) |
| 2.7 Has the product been involved in any safety related incidents? |

| 3 Safety Attributes |
| --- |
| 3.1 What defensive coding measures are used? (e.g. determinism, comprehension) |
| 3.2 What fail-safe measures are included? (e.g. fault tolerance, watchdog) |
| 3.3 Robustness? (e.g. task segregation, resource shortages reported) |
| 3.3 Reliability or availability metrics? (e.g. MTBF, MTTF) |
| 3.4 Security measures? (e.g. passwords, encryption) |
| 3.5 Throughput capacity? (e.g. N events / messages per second) |
| 3.6 Ergonomics or usability is assessed? (e.g. HF task analysis, MMI test reports) |



CGI

# Example: Problem Tracking



Red Hat Bugzilla – Bug List

Home | New | Search | Search [?] | Reports | Requests | Help | New Account | Log In | Forgot Password

Fri Mar 11 2016 10:38:38 EST

**This list is too long for Red Hat Bugzilla's little mind; the Next/Prev/First/Last buttons won't appear on individual bugs.**

Hide Search Description

**Content:** kernel **Status:** NEW, ASSIGNED, POST, MODIFIED, ON_DEV, ON_QA, VERIFIED, RELEASE_PENDING **Product:** Red Hat Enterprise Linux 7

**This result was limited to 1000 bugs.** See all search results for this query.

| ID | Product | Component | Assignee | Status | Resolution |
|---|---|---|---|---|---|
| 1141249 | Red Hat Enterprise Linux 7 | kernel | Vitaly Kuznetsov | NEW | |
| 1245892 | Red Hat Enterprise Linux 7 | kernel | Larry Woodman | NEW | |
| 1283366 | Red Hat Enterprise Linux 7 | kernel | Don Zickus | NEW | |
| 1283368 | Red Hat Enterprise Linux 7 | kernel | Don Zickus | NEW | |
| 1283370 | Red Hat Enterprise Linux 7 | kernel | Don Zickus | NEW | |



CGI

# Checklist (3)

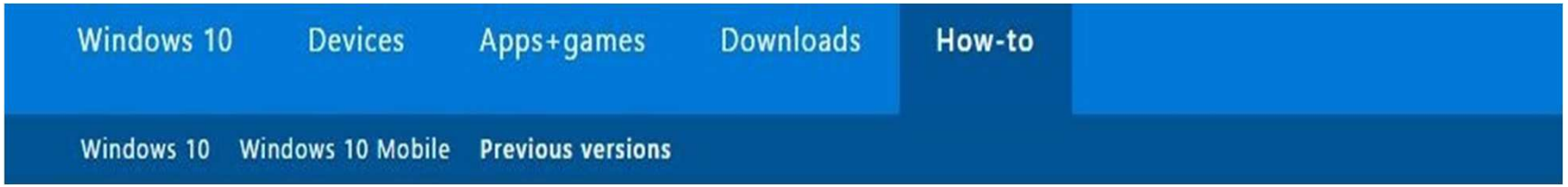
| 4 Safety Assessments |
| --- |
| 4.1 Has been assessed against relevant standards? (e.g. ISO 9001, IEC 61508, DS 00-55) |
| 4.2 Could any of the following be made available: Hazard Analysis, Hazard Log, Risk / Issue Log? |

| 5 Compatibility |
| --- |
| 5.1 Any compatibility issues with other COTS software? (e.g. Windows XP / Vista / 7 / 10) |
| 5.2 Compatible with memory resident software? (e.g. anti-virus, malware protection, firewall) |
| 5.3 Could interfere with other applications sharing the same computing resource? |
| 5.4 Can select which parts of the product are installed? |
| 5.5 Portable to other operating systems or hardware platforms? |
| 5.6 Any foreseeable problems with target environment or interfacing systems? |



CGI

# Example: Compatibility




Making older programs compatible with this version of Windows

Most programs created for earlier versions of Windows will work in this version of Windows, but some older programs might run poorly or not at all.

**Warning**

- Don't run the Program Compatibility Troubleshooter on antivirus programs, firewall software, backup software, disk utilities, or on system programs that came with Windows. This might cause data loss or create a security risk.



CGI

# Checklist (4)

| 6 Product Documentation |
| --- |
| 6.1 Where is the source code maintained? Any escrow arrangements? |
| 6.2 Where is design and development documentation maintained? |
| 6.3 Are the following available: Operating Manual, Installation Guide, Release Notes, Design? |

| 7 Software Planning and Requirements Traceability |
| --- |
| 7.1 Which software development methodology was applied? |
| 7.2 Development processes are documented in: Software Quality Plan / Configuration Management Plan / Coding Standards / Software Development Plan? |
| 7.3 Where the development environment defined? (e.g. tools, compiler, hardware, languages) |
| 7.4 What is the lowest level to which requirements can be traced? (e.g. design, code, tests) |
| 7.5 Required effort to integrate or modify the product for this application is identified? |



CGI

# Example: Product Documentation





CGI

# Checklist (5)

| 8 Test Evidence |
| --- |
| 8.1 Can test plans or results be made available? |
| 8.2 How is functional correctness demonstrated? (e.g. test coverage) |
| 8.3 Estimated fault density? (i.e. predicted residual defects based on code coverage testing) |
| 8.4 Are there any specific tests recommended to verify the product for this application? (e.g. data integrity, performance, availability) |

| 9 Product Support |
| --- |
| 9.1 What is the planned support lifetime for the product? |
| 9.2 What support options are available? (e.g. 24x7 service desk, webchat) |
| 9.3 Availability of developers / technologies to provide long-term support? |
| 9.4 Is support provided for earlier versions of the product? |
| 9.5 Training materials or courses are available for the product? |



CGI

# Example: Product Support

## What is Windows XP end of support?

Microsoft has also stopped providing Microsoft Security Essentials for download on Windows XP. If you already have Microsoft Security Essentials installed, you'll continue to receive anti-malware signature updates for a limited time. However, please note that Microsoft Security Essentials (or any other antivirus software) will have limited effectiveness on PCs that do not have the latest security updates. This means that PCs running Windows XP will not be secure and will still be at risk of infection.

## What happens if I continue to use Windows XP?

If you continue to use Windows XP now that support has ended, your computer will still work but it might become more vulnerable to security risks and viruses. Internet Explorer 8 is also no longer supported, so if your Windows XP PC is connected to the Internet and you use Internet Explorer 8 to surf the web, you might be exposing your PC to additional threats. Also, as more software and hardware manufacturers continue to optimise for more recent versions of Windows, you can expect to encounter more apps and devices that do not work with Windows XP.



CGI

# Checklist (6)

| 10 Product Updates |
| --- |
| 10.1 Are updates included? Do they include change descriptions? |
| 10.2 How are updates distributed? Could upgrading cause an outage? |
| 10.3 Are there any planned updates for known problems? |
| 10.4 Is there a means for alerting users to unresolved problems that may impact safety? |
| 10.5 Is there a means for prioritising urgent fixes to maintain safety? |
| 10.6 Is a change history available? Can changes be tracked to particular product versions? |
| 10.7 Are metrics maintained for faults found in service or their resolution? |



CGI

# Example: Product Updates

MICROSOFT SOFTWARE LICENSE TERMS

WINDOWS OPERATING SYSTEM

**6. Updates**. The software periodically checks for system and app updates, and downloads and installs them for you. You may obtain updates only from Microsoft or authorized sources, and Microsoft may need to update your system to provide you with those updates. By accepting this agreement, you agree to receive these types of automatic updates without any additional notice.

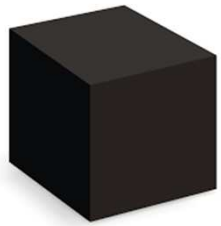


CGI

# Distribution

- **Option 1 – Submit checklist to developer for completion**
  - Access to white box evidence
  - Normally in developer's commercial interest to provide
    - Could be approached under support contract
  - Responses may be commercially sensitive
    - Assurance for safety assurance purposes only
    - NDA may be required to disclose information to third parties

- **Option 2 – Complete using knowledge within the organisation**
  - May be limited to black box evidence
    - Unless organisation is also the developer
    - There may be gaps in responses
    - Additional analysis or testing may be required
  - Review by knowledgeable staff – e.g. Design Authority





CGI

# Acceptance Criteria

- IEC 61508 criteria for proven-in-use
  - ✓ Design specification does not require modification
  - ✓ Applied in different applications
  - ✓ At least 1 year service history
  - ✓ Operational lifetime or demands commensurate with SIL
  - ✓ Operating experience relates to known demand profile
  - ✓ No safety related failures
- DO-178C / ED-12C criteria for service history
  - ✓ Configuration management
  - ✓ Relevance of service history environment
  - ✓ Length of service history
  - ✓ Effective problem reporting
  - ✓ Actual error rates
  - ✓ Impact of modifications
- If evidence meets criteria, reasonable to expect similar integrity, reliability and performance in new application?




CGI

# Trusted Software

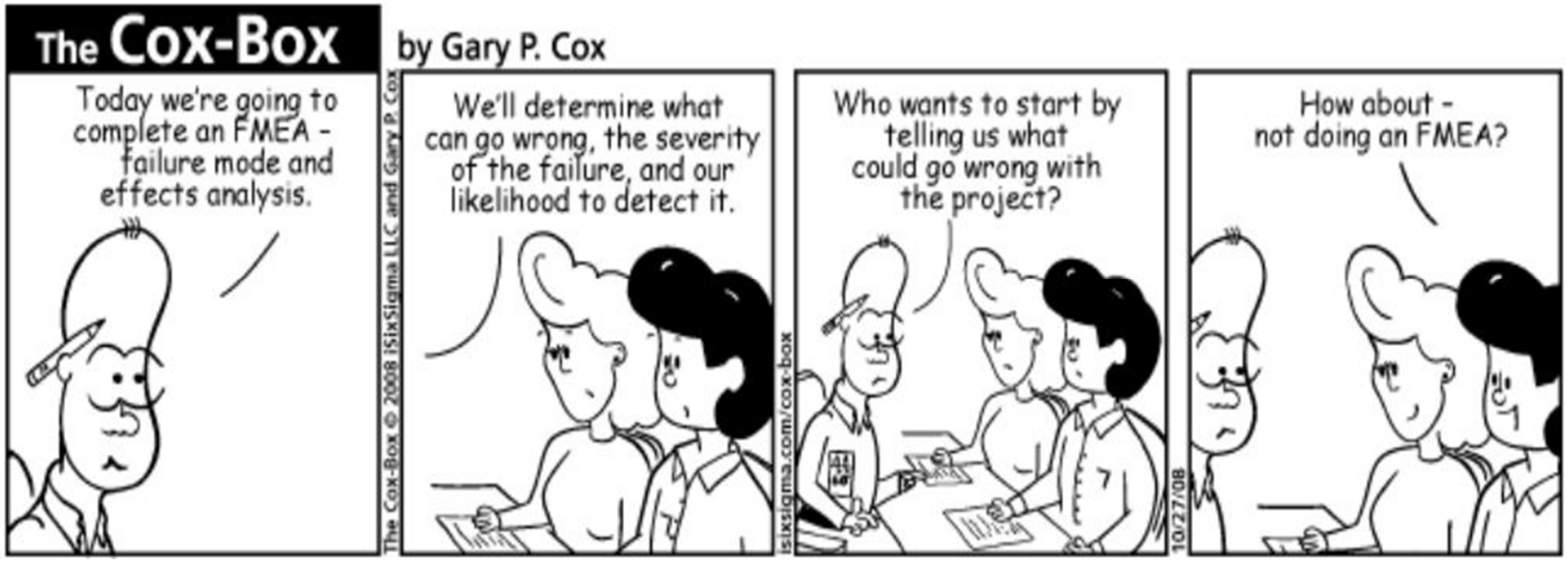
- "*To take advantage of designs which have not been formally or rigorously verified, but for which considerable operational history is available*" (IEC 61508)
- "*Compliant Element Safety Manual*" identifies:
  - Software including version numbers
  - Users and time of application
  - Operating time
  - Procedure for selection of user-applied systems and application cases
  - Procedure for detecting, reporting failures, and for removing faults
  - Capabilities and limitations of reusable elements
- "*Only in rare cases will 'proven-in-use' be a sufficient argument that a trusted software element achieves the necessary safety integrity*"



CGI

# Constraints on Use of Evidence

- Support specific safety claims for application of specific software release in specific operating environment
- Base safety claims on objective evidence
- Justify any extrapolation from checklist responses
- Unsupported leaps of faith may not be acceptable
- No substitute for functional safety analysis



# Managing Assurance Deficits

1. What evidence is required to support safety claims?
2. What evidence has supplier provided?
   - Shortfalls may weaken the safety argument

- Compensatory measures:
  - Functional analyses – e.g. FMEA, FTA
  - Static analysis of source code
  - Hazard-driven testing
  - Fixes, workarounds, user caveats for known problems
  - Application monitoring / Event logging
  - Data replication / Redundancy
- If assurance gap still too wide then consider product replacement




CGI

# Further Reading


[1] IEC 61508, Functional safety of electrical / electronic / programmable electronic safety related systems.

[2] DO-178C, Software Considerations in Airborne Systems and Equipment Certification.

[3] Justifying the Use of Software of Uncertain Pedigree in safety-related applications.

[4] Clear SOUP and COTS Software Can Reliably Serve Safety-critical Systems.

[5] Blogs: MS License: Using Java Could Lead to Death.

[6] Red Hat Bugzilla Bug Tracking System.

[7] Hazard-driven Testing of Safety-related Software.

[8] How can I use Windows XP safely now it's no longer supported?

[9] Windows 10 updates to be automatic and mandatory for Home users.

[10] Windows Updates that shut down computer without warning.




CGI

# Our commitment to you

We approach every engagement with one objective in mind: to help clients succeed

CGI

Experience the commitment®