Proposal for new SAR components and Packages in CC for Patch Management

Contents

Proposal for new SAR components and Packages in CC for Patch Management

FIGURES

TABLES

Proposal for new SAR components and Packages in CC for Patch Management

Introduction

This document deals with an ISCI-WG1 proposal for addition of SAR components for patch management in addition to the existing ones in Common Criteria [CC V3.1 R5]. It also adds new families (ALC_IAR, ALC_CMA and AGD_DEV), to include new components.

This document deals with a proposal for addition of SAR evaluation methodology in CEM [CEM V3.1 R5] for patch management.

This document includes also a proposal of CEM refinement for a given list of existing SAR components.

It also promotes new KAP packages aggregating new SAR components for product certification and KAP-G to be included in site certification.

Such proposal may request some adjustments for application in future release of Common Criteria.

Reference Documents

[CC V3.1 R5]	Common Methodology for Information Technology Security Evaluation Part 3: Security assurance components April 2017 Version 3.1 Revision 5 https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf
[CEM V3.1 R5]	Common Methodology for Information Technology Security Evaluation Evaluation methodology April 2017 Version 3.1 Revision 5 https://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R5.pdf
[Assurance Continuity]	Assurance Continuity: CCRA requirements [AC:2012-06-01] https://www.commoncriteriaportal.org/files/operatingprocedures/2012-06-01.pdf JIL- Assurance Continuity - Practical Cases for Smart Cards and similar devices https://www.sogis.eu/documents/cc/domains/sc/JIL-Assurance-Continuity-Practical-Cases-for-Smart-Cards-and-similar-devices-v1-0.pdf
[site certification]	Site certification : [CCDB-2007-11-001] https://www.commoncriteriaportal.org/files/supdocs/CCDB-2007-11-001-SiteCertificationProcessv1-0.pdf
[M_KAP]	Methodology for Known Assurance Process (KAP) [not yet available (to be produced from existing slides)]
[PP_Module with KAP]	Proposal of PP module for patch management using KAP packages
[G_INT]	Guidance for integration of new SAR components dedicated to additional code in initial TOE evaluation. [not yet available]

Proposal for new SAR components and Packages in CC for Patch Management

List of new SAR components and purpose

Find here the list of new components to add to Common Criteria.

Component	Component identifier	Family identifie	Class identifier
AGD_PRE.2	Preparative procedures for additional code management prior final TOE	AGD_PRE	AGD
AGD_DEV.1	Development guidance for application code and final TOE	AGD_DEV (New)	AGD
ALC_CMA.1	Labelling of the additional code	ALC_CMA (New)	ALC
ALC_CMA.2	Use of a CM system for additional code	ALC_CMA (New)	ALC
ALC_CMA.3	Production support, acceptance procedures and automation fo additional code	ALC_CMA (New)	ALC
ALC_DEL.2	Delivery of Additional Code	ALC_DEL	ALC
ALC_FLR.4	Systematic flaw remediation with additional code deployment	ALC_FLR	ALC
ALC_IAR.1	Impact analysis applied to non-security relevant change to the TOE	ALC_IAR (New)	ALC
ALC_IAR.2	Impact Analysis associated to Security Change in TOE Implementation (Binary)	ALC_IAR (New)	ALC
ALC_IAR.3	Impact Analysis associated to Security Change in TOE Implementation Representation (Binary and Source code)	ALC_IAR (New)	ALC
ALC_IAR.4	Impact Analysis associated to Security Change in TOE Design	ALC_IAR (New)	ALC

Table 1: List of new SAR components and associated purpose

Proposal for new SAR components and Packages in CC for Patch Management

Usage of components per item

The following table describes how the SAR components are applied per item (initial TOE, Additional Code, Final TOE).

Component	Component identifier	Applicable to Initi TOE	Applicable to Additional Code	Applicable to Final TOE
AGD_PRE.2	Preparative procedures for additional code management prior final TOE	YES (Procedure)		Final TOE evidences
AGD_DEV.1	Development guidance for application code and final TOE	YES (Procedure) SAF template	Self-Assessment Form completed	
ALC_CMA.1	Labelling of the additional code	YES (Procedure + AC sample)	AC evidences	Final TOE evidences
ALC_CMA.2	Use of a CM system for additional code	YES (Procedure + AC sample)	AC evidences	Final TOE evidences
ALC_CMA.3	Production support, acceptance procedures and automation for additional code	YES (Procedure + AC sample)	AC evidences	Final TOE evidences
ALC_DEL.2	Delivery of Additional Code	YES (Procedure + AC sample)	AC evidences	
ALC_FLR.4	Systematic flaw remediation with additional code deployment	YES (Procedure + Flaw sample)	AC evidences if flaw	Final TOE evidences
ALC_IAR.1	Impact analysis applied to non-security relevant change to the TOE	YES (Procedure + IAR1 sample)	IAR.1 + AC evidences	Final TOE evidences
ALC_IAR.2	Impact Analysis associated to Security Change in TOE Representation (Binary)	YES (Procedure + IAR2 sample)	IAR.2 + AC evidences + Binary	Final TOE evidences
ALC_IAR.3	Impact Analysis associated to Security Change in TOE Representation (Binary and Source code)	YES (Procedure + IAR3 sample)	IAR.3 + AC evidences + Binary & changes in IMP	Final TOE evidences
ALC_IAR.4	Impact Analysis associated to Security Change in TOE Design	YES (Procedure + IAR3 sample)	IAR.3 + AC evidences + Binary & changes in IMP, TDS, FSP	Final TOE evidences

Table 2: List of new SAR components per item

It is important to note that according to the level of selected assurance (Substantial or High), a choice is done between ALC_CMA. 2 or 3 for all additional code associated to an initial TOE.

Currently by default, the following level of assurance are defined:

- Substantial assurance level is associated to EAL2 and EAL 3 including AVA_VAN.2 and ALC_CMA.2,
- High assurance level is associated to EAL4 and higher associated to AVA_VAN.3 and higher and ALC_CMA.3 (augmentation to AVA_VAN.5 is a standard option)

If such default choice is not maintained, it may require some adjustment in SAR component definition.

The choice of ALC_IAR component is initially selected according to the level of assurance. It demonstrate your ability to manage different types of changes. Then, when a change is required, the type of evaluation of developer evidences is selected according to type of impact.

It is important to note that for each additional code and according to the type of change (functional or security), a choice has to be done when it is applied to a change but before product evaluation.

- Substantial assurance level is associated to EAL2 and EAL 3 including AVA_VAN.2 and ALC_IAR.3 allowing to produce an ALC_IAR.1 or ALC_IAR.3 for each additional code.
- High assurance level is associated to EAL4 and higher associated to AVA_VAN.3 and higher and ALC_IAR.4 allowing to produce an ALC_IAR.1 or ALC_IAR.3 or ALC_IAR.4 for each additional code.

Usage of components in TOE environment

The following figure is a representation on usage of SAR components in TOE environment and describes how SAR components are applicable to the different operations associated to initial TOE (TOE version N) and Additional code (AC1 and AC2 on TOE version N) and Final TOE (Final TOE 1: TOE version N + AC1 on TOE version N, Final TOE 2: TOE version N + AC1 and AC2 on TOE version N).

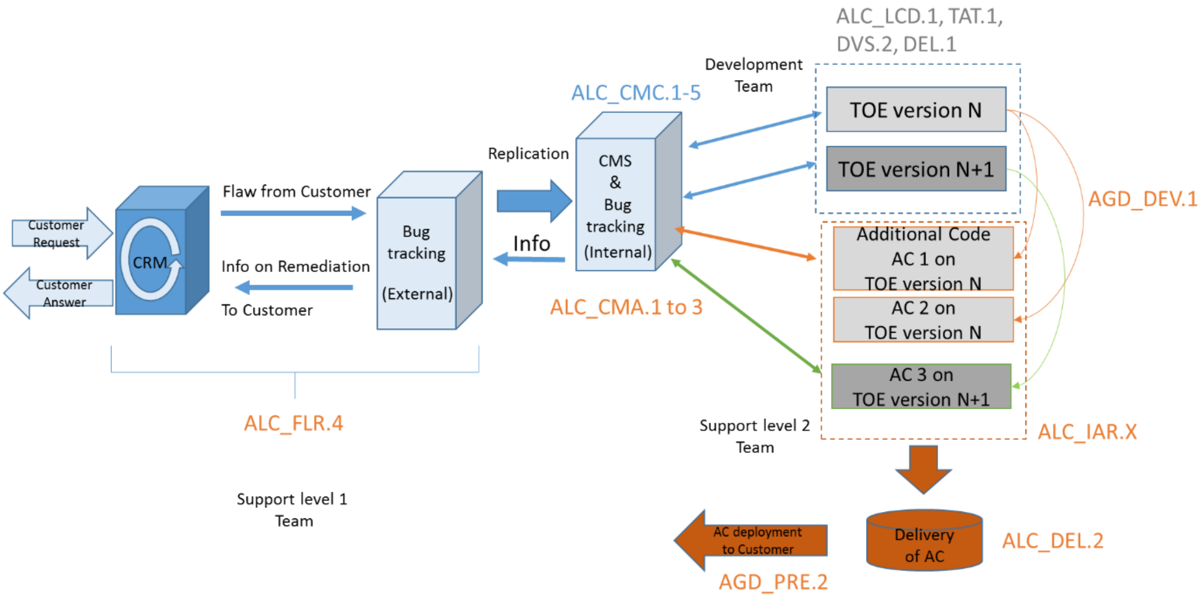

Figure 1: SAR components applicable to the different operations

Note: The figure represents that developer may decide for new TOE release (TOE version N+1) to introduce completely or partially or not existing additional code (AC1 and AC2 from TOE version N). On this new TOE release, it will be possible to add new additional code (AC3 on TOE version N+1).

But obviously, this activity on new TOE release will be not covered by evaluation of Final TOE 1 and 2.

List of existing SAR components requiring CEM refinement

Find here the list of existing components requiring an application note in CEM to cover additional code management.

Component	Component identifier	Comments
ALC_TAT.1	Well-defined development tools	may require an application note in CEM to cover additional code management
ALC_DVS.1	Identification of security measures	may require an application note in CEM to cover additional code management in addition to TOE and its parts
ALC_DVS.2	Sufficiency of security measures	may require an application note in CEM to cover additional code management in addition to TOE and its parts
ALC_CMC.4	Problem tracking CM coverage	may require an application note in CEM to cover additional code management in addition to TOE and its parts
ALC_CMC.5	Development tools CM coverage	may require an application note in CEM to cover additional code management in addition to TOE and its parts

Table 3: List of existing SAR components requiring CEM refinement for Initial TOE

Proposal for new SAR components and Packages in CC for Patch Management

Addition of components to Common Criteria version 3.1 release 5

Here after find the new SAR components to be added to Common Criteria to addressed additional code and Final TOE management.

AGD_PRE.2 Preparative procedures for additional code management prior final TOE usage

Dependencies: No dependencies

Objectives

Preparative procedures for additional code management are useful for ensuring that the additional code has been received and installed in a secure manner as intended by the developer to produce the final TOE. The requirements for preparation includes a secure transition of additional code to its operational environment. This includes investigating whether the additional code can be configured or installed in a manner that is insecure but that the user of the TOE would reasonably believe to be secure.

Developer action elements:

AGD_PRE.2.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:

AGD_PRE.2.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the application code and the final TOE (when generated) in accordance with the developer's delivery procedures.

AGD_PRE.2.2C The preparative procedures shall describe all the steps necessary for secure installation of the application code in the initial TOE to obtain the final TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

AGD_PRE.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.2.2E The evaluator shall apply the preparative procedures to confirm that the application code and the final TOE can be prepared securely for operation.

AGD_DEV.1 Development guidance for application code and final TOE

Dependencies: ADV_FSP.1 Basic functional specification

Objectives

Development guidance for application code refers to written material that is intended to be used by all persons responsible of development of the additional code. Development guidance for application code provides instructions and guidelines (including warnings), helps to understand how to develop additional code in addition to initial TOE and how to create final TOE. Misleading and unreasonable guidance should be absent from the guidance documentation, and secure procedures for all modes of operation should be addressed. Insecure states should be easy to detect.

The evaluation of the development guidance for application code includes investigating whether the TOE can be used in a manner that is insecure but that the user of the TOE would reasonably believe to be secure. The objective is to minimise the risk of human or other errors in development resulting in an undetected insecure state of final TOE.

Developer action elements:

AGD_DEV.1.1D The developer of initial TOE shall provide a development guidance for application code and final TOE.

Note: AGD_DEV.1 will be also applied to final TOE, where the developer of final TOE shall provide operational user guidance for final TOE. It is not repeated here.

Content and presentation elements:

AGD_DEV.1.1C The development guidance for application code and final TOE shall describe how to develop application code to obtain final TOE without insecure TOE states.

AGD_DEV.1.2C The development guidance for application code and final TOE shall describe how to develop application code to be loaded on initial TOE to obtain final TOE.

AGD_DEV.1.3C The development guidance for application code and final TOE shall describe how to perform acceptance of additional code and procedure to test final TOE prior delivery of additional code.

AGD_DEV.1.4C The development guidance for application code and final TOE shall be clear and reasonable.

Evaluator action elements:

AGD_DEV.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ALC_CMA.1 Labelling of the additional code

Dependencies: ALC_CMS.1 TOE CM coverage

Objectives
A unique reference is required to ensure that there is no ambiguity in terms of which instance of the Additional code and Final TOE are being evaluated. Labelling the Additional code with its reference ensures that users of the Final TOE can be aware of which instance of the TOE they are using.

Application note:

If developer decides to only provide a reference and version of final TOE, information and label concerning additional code leading to final TOE should be provided to user by other means as documentation.

Developer action elements:

ALC_CMA.1.1D The developer shall provide the additional code to the initial TOE and a reference for the additional code leading to a reference to the final TOE.

Content and presentation elements:

ALC_CMA.1.1C The additional code shall be labelled with its unique reference and link is created with Final TOE.

Evaluator action elements:

ALC_CMA.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ALC_CMA.2 Use of a CM system for additional code

Dependencies: ALC_CMS.1 TOE CM coverage

Objectives

A unique reference is required to ensure that there is no ambiguity in terms of which instance of the Additional code and Final TOE are being evaluated. Labelling the additional code with its reference ensures that users of the Final TOE can be aware of which instance of the TOE they are using.

Unique identification of the configuration items of additional code leads to a clearer understanding of the composition of the final TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the additional code and final TOE.

The use of a CM system increases assurance that the configuration items of additional code and Final TOE are maintained in a controlled manner.

Application note:

If developer decides to only provide a reference and version of final TOE, information and label concerning additional code leading to final TOE should be provided to user by other means as documentation.

ALC_CMA.2.1D The developer shall provide the additional code to the initial TOE and a reference for the additional code leading to a reference to the final TOE.

ALC_CMA.2.2D The developer shall provide the CM documentation for additional code and how it leads to the final TOE.

ALC_CMA.2.3D The developer shall use a CM system for additional code and the final TOE.

Content and presentation elements:

ALC_CMA.2.1C The additional code and the final TOE shall be labelled each one with its unique reference.

ALC_CMA.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items associated to additional code leading to the final TOE.

ALC_CMA.2.3C The CM system shall uniquely identify all configuration items associated to additional code and link to final TOE.

Evaluator action elements:

ALC_CMA.2.1E The evaluator ***shall confirm*** that the information provided meets all requirements for content and presentation of evidence.

ALC_CMA.3 Production support, acceptance procedures and automation for additional code

Dependencies: ALC_CMS.1 TOE CM coverage
ALC_DVS.1 Identification of security measures
ALC_LCD.1 Developer defined life-cycle model

Objectives

A unique reference is required to ensure that there is no ambiguity in terms of which instance of the Additional code and Final TOE are being evaluated. Labelling the additional code with its reference ensures that users of the Final TOE can be aware of which instance of the TOE they are using.

Unique identification of the configuration items of additional code leads to a clearer understanding of the composition of the final TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the additional code and final TOE.

The use of a CM system increases assurance that the configuration items of additional code and Final TOE are maintained in a controlled manner. Providing controls to ensure that unauthorised modifications are not made to the additional code and final TOE (“CM access control”), and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the additional code and final TOE.

Developer action elements:

ALC_CMA.3.1D The developer shall provide the additional code to the initial TOE and a reference for the additional code leading to a reference to the final TOE.

ALC_CMA.3.2D The developer shall provide the CM documentation for the additional code and how it leads to the final TOE.

ALC_CMA.3.3D The developer shall use a CM system for the additional code and the final TOE.

Content and presentation elements:

ALC_CMA.3.1C The additional code and the final TOE shall be labelled each one with its unique reference.

ALC_CMA.3.2C The CM documentation shall describe the method used to uniquely identify the configuration items associated to additional Code leading to the final TOE.

ALC_CMA.3.3C The CM system shall uniquely identify all configuration items associated to additional code and link to final TOE.

ALC_CMA.3.4C The CM system shall provide automated measures such that no changes are made to the configuration items associated to the initial TOE and that only authorised changes are made to the configuration items associated to additional code, and the final TOE.

ALC_CMA.3.5C The CM system shall support the production of the additional code, and the final TOE by automated means.

Proposal for new SAR components and Packages in CC for Patch Management

ALC_CMA.3.6C The CM documentation shall include a CM plan for the additional code, and the final TOE.

ALC_CMA.3.7C The CM plan shall describe how the CM system is used for the restricted access to the initial TOE and the development of the additional code, and the final TOE.

ALC_CMA.3.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the additional code, and the final TOE.

ALC_CMA.3.9C The evidence shall demonstrate that all configuration items of the additional code, and the final TOE are being maintained under the CM system.

Evaluator action elements:

ALC_CMA.3.1E The evaluator ***shall confirm*** that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.2 Delivery of Additional Code for flaw remediation procedures

Dependencies: No dependencies

Objectives

In complement to TOE delivery procedures defined in ALC_DEL.1, this component requires to check procedures for additional code deployment. Guidance for additional code deployment ensures that TOE developers and TOE issuers are aware and able to manage additional code to obtain final TOE on the field.

Developer action elements:

ALC_DEL.2.1D The developer shall document and provide procedures for delivery of the additional code to the consumer.

ALC_DEL.2.2D The developer shall use the delivery procedures for the additional code.

Content and presentation elements:

ALC_DEL.2.1C The delivery documentation for additional code shall describe all procedures that are necessary to maintain security when distributing versions of the additional code to the consumer in order to obtain the final TOE.

Evaluator action elements:

ALC_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.4 Systematic flaw remediation with Additional Code deployment

Dependencies: No dependencies

Objectives

In complement to systematic flaw remediation procedure, this component requires to check procedures for additional code development and deployment. Guidance for Flaw remediation with additional code deployment user ensures that TOE developers and TOE issuers are aware and able to manage initial TOE and additional code to obtain final TOE on the field.

Developer action elements:

ALC_FLR.4.1D The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.4.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.4.3D The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.4.4D The developer shall document and provide “additional code” patch creation procedures to TOE developers to contribute to confidentiality, integrity and authenticity of “additional code” during development.

ALC_FLR.4.5D The developer shall document and provide procedures to TOE users for “additional code” deployment to maintain confidentiality, integrity and authenticity of “additional code” during deployment.

ALC_FLR.4.6D The developer shall provide evidences on “additional code” development to obtain similar assurance level on final TOE that the level provided for initial TOE.

ALC_FLR.4.7D The developer shall provide evidences on “Additional Code” deployment to obtain similar assurance level on final TOE that the level provided for initial TOE.

Content and presentation elements:

ALC_FLR.4.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.4.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.4.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.4.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.4.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.4.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.4.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.4.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.4.9C "Additional code" creation procedures shall demonstrate how the confidentiality, integrity and authenticity of the Additional code is assumed prior code delivery.

ALC_FLR.4.10C "Additional code" creation procedures shall demonstrate how the confidentiality, integrity and authenticity of the Additional code is maintained until code loading.

ALC_FLR.4.11C The flaw remediation guidance shall provide detailed instructions for users on how to check the availability of new "Additional Code" patches and how to apply them.

Evaluator action elements:

ALC_FLR.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.4.2E The evaluator shall exercise the "Additional Code" update process to verify that it is correctly handled including the documentation of the Impact of Changes.

ALC_FLR.4.3E The evaluator shall verify that a commensurate level of assurance is provided for the design, development, testing and delivery of "Additional Code".

ALC_IAR.1 Impact analysis applied to non-security relevant change to the TOE

Dependencies: No dependencies

Objectives

Impact analysis applied to non-security relevant change to the TOE requires that changes in initial TOE due to introduction of additional code to obtain final TOE be tracked and analyzed by the developer according to a defined procedure to be considered in evaluator actions for TOE assurance continuity.
This component is dedicated to non-security relevant change to the TOE limiting the expected evidences for evaluation.

Developer action elements:

ALC_IAR.1.1D The developer shall document and provide impact analysis procedures addressed to TOE developers.

Content and presentation elements:

ALC_IAR.1.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE

ALC_IAR.1.2C The impact analysis procedures shall require that a description of the nature and effect of each change be provided, as well as the status of security relevance of the changes and associated rationale.

ALC_IAR.1.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.1.4C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for non-security relevant modification as defined for application code and Final TOE development.

ALC_IAR.1.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for non-security relevant modification defined for application code development.

Evaluator action elements:

ALC_IAR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_IAR.2 Impact Analysis associated to Security Change in TOE Implementation

Dependencies: No dependencies

Objectives

Impact analysis applied to security change in the TOE implementation requires that changes in initial TOE due to introduction of additional code to obtain final TOE be tracked and analyzed by the developer according to a defined procedure to be considered in evaluator actions for TOE assurance continuity.
This component is dedicated to security change in the TOE implementation limiting the expected evidences relevant for TOE implementation for evaluation.

Note: TOE implementation means executable code or binary form of TOE.

Developer action elements:
ALC_IAR.2.1D The developer shall document and provide impact analysis procedures addressed to TOE developers.

Content and presentation elements:
ALC_IAR.2.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE.

ALC_IAR.2.2C The impact analysis procedures shall require that a description of the nature and effect of each change be provided, as well as the status of security relevance of the changes and associated rationale.

ALC_IAR.2.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.2.4C **The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for security relevant modification limited to the TOE implementation as defined for application code and Final TOE development.**

ALC_IAR.2.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of additional code has been performed consistently with rules for security relevant modification in TOE implementation defined for application code and Final TOE development.

ALC_IAR.2.6C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for TOE functional testing defined for Final TOE development.

Evaluator action elements:
ALC_IAR.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Proposal for new SAR components and Packages in CC for Patch Management

ALC_IAR.3 Impact Analysis associated to Security Change in TOE Implementation Representation

Dependencies: No dependencies

Objectives

Impact analysis applied to security change in the TOE implementation representation requires that changes in initial TOE due to introduction of additional code to obtain final TOE be tracked and analyzed by the developer according to a defined procedure to be considered in evaluator actions for TOE assurance continuity.
This component is dedicated to security change in the TOE implementation representation limiting the expected evidences relevant for TOE implementation and TOE implementation representation for evaluation.

Note: TOE implementation representation means source code of TOE.

Developer action elements:
ALC_IAR.3.1D The developer shall document and provide impact analysis procedures addressed to TOE developers.

Content and presentation elements:
ALC_IAR.3.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE.

ALC_IAR.3.2C The impact analysis procedures shall require that a description of the nature and effect of each change be provided, as well as the status of security relevance of the changes and associated rationale.

ALC_IAR.3.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.3.4C **The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for security relevant modification limited to the TOE implementation and the TOE implementation representation as defined for application code and Final TOE development.**

ALC_IAR.3.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of additional code has been performed consistently with rules for security relevant modification in the TOE implementation and the TOE implementation representation defined for application code and Final TOE development.

ALC_IAR.3.6C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for TOE functional testing defined for Final TOE development.

Evaluator action elements:
ALC_IAR.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_IAR.4 Impact Analysis associated to Security Change in TOE Design

Dependencies: No dependencies

Objectives

Impact analysis applied to security change in the TOE design requires that changes in initial TOE due to introduction of additional code to obtain final TOE be tracked and analyzed by the developer according to a defined procedure to be considered in evaluator actions for TOE assurance continuity.
This component is dedicated to security change in the TOE implementation and TOE representations requiring the evidences for all TOE items covered by evaluation.

Developer action elements:

ALC_IAR.4.1D The developer shall document and provide impact analysis procedures addressed to TOE developers.

Content and presentation elements:

ALC_IAR.4.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE.

ALC_IAR.4.2C The impact analysis procedures shall require that a description of the nature and effect of each change be provided, as well as the status of security relevance of the changes and associated rationale.

ALC_IAR.4.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.4.4C **The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for security relevant modification to the TOE implementation and all the different TOE representations as TOE implementation representation, TOE design or specification as defined for application code and Final TOE development.**

ALC_IAR.4.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of additional code has been performed consistently with rules for security relevant modification all the different TOE representations as TOE implementation representation, TOE design or specification as defined for application code and Final TOE development.

ALC_IAR.4.6C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for TOE functional testing defined for Final TOE development.

ALC_IAR.4.7C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules of TOE test coverage defined for Final TOE development.

ALC_IAR.4.8C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules of analysis of the TOE depth of testing defined for Final TOE development.

Evaluator action elements:

ALC_IAR.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Addition of Family to Common Criteria version 3.1 release 5.

CM capabilities for additional code (ALC_CMA)

Objectives

Configuration management (CM) for additional code is one means for increasing assurance that the final TOE meets the SFRs. CM establishes this by requiring discipline and control in the processes of refinement and modification of the initial TOE and the related information. CM systems are put in place to ensure the integrity of the additional code and the final TOE that they control, by providing a method of tracking any changes, and by ensuring that all changes are authorised.

The objective of this family is to require the developer's CM system for additional code to have certain capabilities. These are meant to reduce the likelihood that accidental or unauthorised modifications of the configuration items will occur. The CM system for additional code should ensure the integrity of the additional code and the final TOE from the early design stages through all subsequent maintenance efforts.

The objective of introducing automated CM tools is to increase the effectiveness of the CM system. While both automated and manual CM systems can be bypassed, ignored, or proven insufficient to prevent unauthorised modification, automated systems are less susceptible to human error or negligence.

Component levelling
The components in this family are levelled on the basis of the CM system capabilities, the scope of the CM documentation and the evidence provided by the developer.

Application notes
In the case where the TOE is a subset of a product, the requirements of this family apply only to the configuration items of additional code and the final TOE, not to the product as a whole.

Impact analysis (ALC_IAR)

Objectives

Impact analysis requires that changes in initial TOE due to introduction of additional code to obtain final TOE be tracked and analyzed by the developer according to a defined procedure to be considered in evaluator actions for TOE assurance continuity.

Although future compliance with Impact analysis procedures cannot be determined at the time of the initial TOE evaluation, it is possible to evaluate the policies and procedures that a developer has in place to analyze impact of change associated to a flaw or improvement, and to distribute the impact information and set of evidences associated to change to the evaluator allowing its impact analysis review.

Component levelling

The components in this family are levelled in a hierarchic order on the basis of the increasing extent in scope of the impact analysis procedures and the set of evidences to be produced and reviewed.

Application notes

This family provides assurance that the TOE will be maintained and supported in the future, requiring the TOE developer to perform impact analysis on TOE to maintain or update evaluation technical report. However, this family does not impose evaluation requirements beyond the initial TOE evaluation.

Guidance for development of additional code (AGD_DEV)

Objectives

Guidance for development of additional code refers to written material that is intended to be used by all persons in charge of development and acceptance of additional code to maintain or increase the security of the initial TOE.

Guidance for development of additional code describes the security functionality provided by the TSF, provides instructions and guidelines (including warnings), helps to understand the TSF and includes the security-critical information, and the security-critical actions required, for its development of additional code. Misleading and unreasonable guidance should be absent from the guidance documentation, and secure procedures for all modes of operation should be addressed.

The operational user guidance provides rules to be checked during development of additional code using self-assessment form and acceptance procedure to increase confidence that Final TOE will be at least secure that initial TOE. The objective is to minimise the risk of human or other errors in development of additional code that may deactivate, disable, or fail to activate security functionality, resulting in an undetected insecure state.

Component levelling

This family contains only one component.

KAP packages to address levels of assurance for additional code management.

The following table presents the 3 packages KAP-B, KAP-S, KAP-H addressing three level of assurance for additional code management.

, KAP-S package is associated to Substantial level assurance and KAP-H package is associated to High level assurance.

Assurance class	Assurance Family	Assurance Components by Known Assurance Package	
		KAP-S	KAP-H
Guidance documents	AGD_PRE	2	2
	AGD_DEV	1	1
Life-cycle support	ALC_CMA	**2**	**3**
	ALC_DEL	2	2
	ALC_FLR	4	4
	ALC_IAR	**3**	**4**
Security Target evaluation	ASE_CCL	1	1
	ASE_ECD	1	1
	ASE_INT	1	1
	ASE_OBJ	**2**	2
	ASE_REQ	**2**	2
	ASE_SPD	**1**	1
	ASE_TSS	1	1

Table 4: Assurance components included in Known Assurance Packages

Note: ALC_IAR.1 is not included in any described KAP package but it is applicable to functional change due to hierarchical principal in ALC_IAR family as explained in paragraph after Table 2.

Note: ALC_CMA.1 and ALC_IAR.2 are not included in any described KAP package because there are applicable only for low assurance package not available in this document.

KAP-S package is supposed to be used in complement with EAL2 or EAL3 package.

KAP-H package is supposed to be used in complement with EAL4 and higher packages.

Proposal for new SAR components and Packages in CC for Patch Management

KAP-G package to address generic assurance for additional code management in site certification.

The following table presents the KAP-G package addressing assurance generic assurance for additional code management in site certification.

Assurance class	Assurance Family	Assurance Components by Generic Known Assurance Package
		KAP-G
Guidance documents	AGD_PRE	**2**
	AGD_DEV	**1**
Life-cycle support	ALC_CMA	**3**
	ALC_DEL	**2**
	ALC_FLR	**4**
	ALC_IAR	**4**

Table 5: Assurance components included in KAP-G package

Note: ALC_IAR.1 is not included in KAP-G package but it is applicable to functional change due to hierarchical principal in ALC_IAR family.

KAP-G package is defined to reach high level assurance as done using site certification process and extends the set of SAR components already defined in [site certification] as described in following table.

Assurance class	Assurance Fam	Assurance Components included Site certification
Life-cycle suppor	ALC_CMC	**5**
	ALC_CMS	**5**
	ALC_DEL	**1**
	ALC_DVS	2
	ALC_FLR	**1 to 3**
	ALC_LCD	**2**
	ALC_TAT	**3**
Site Security Targ evaluation	AST_CCL	**1**
	AST_ECD	**1**
	AST_INT	**1**
	AST_OBJ	**1**
	AST_REQ	**1**
	AST_SPD	**1**
	ASE_TSS	**1**

Table 6: Assurance components included in Site Certification

KAP-G package is supposed to be used in site certification process and then be reused in product certification with EAL4 or higher within a scope included a KAP-H package.

Proposal for new SAR components and Packages in CC for Patch Management

Addition to CEM version 3.1 release 5.

The following paragraphs are a proposal of add-on to CEM for new SAR components introduced in previous chapter. Each paragraph can be studied separately.

Paragraph numbering is defined from current table of content of CEM to allow simple introduction in document (if required). Otherwise, it may remain independent as extension to a PP-module.

13.4.2 **Evaluation of sub-activity (AGD_PRE.2)**

Note: AGD_PRE.2 is a component written as a refinement of AGD_PRE.1 applied to the application code and the final TOE. Items in blue "highlights" the change vs AGD_PRE.1.

13.4.2.1 Objectives

The objective of this sub-activity is to determine whether the preparative procedure for additional code management have been documented and result in a secure configuration of final TOE.

13.4.2.2 Input

The evaluation evidence for this sub-activity is:

a) the ST;
b) the initial TOE and additional code including its preparative procedures for additional code management prior final TOE usage;
c) the description of additional code developer's delivery procedures;

13.4.2.3 Application notes

The preparative procedures refer to all acceptance and installation procedures that are necessary to progress from the initial TOE to the final TOE within a secure configuration as described in the ST.

13.4.2.4 Action AGD_PRE.2.1E

AGD_PRE.2.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the application code and the final TOE (when generated) in accordance with the developer's delivery procedures.

AGD_PRE.2-1 The evaluator shall examine the provided acceptance procedures to determine that they describe the steps necessary for secure acceptance of the application code and the final TOE (when generated) in accordance with the developer's delivery procedures.

If it is not anticipated by the developer's delivery procedures that acceptance procedures will or can be applied, this work unit is not applicable, and is therefore considered to be satisfied.

The acceptance procedures should include as a minimum, that the user has to check that the application code and the final TOE (when generated) as indicated in the ST have been delivered in the correct version.

The acceptance procedures should reflect the steps the user has to perform in order to accept the delivered application code and the final TOE (when generated) that are implied by the developer's delivery procedures.

The acceptance procedures should provide detailed information about the following, if applicable:

a) making sure that the delivered application code and the final TOE (when generated) is the complete evaluated instance;

b) detecting modification/masquerading of the delivered application code and the final TOE (when generated).

AGD_PRE.2.2C The preparative procedures shall describe all the steps necessary for secure installation of the application code in the initial TOE to obtain the final TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.2-2 The evaluator shall examine the provided installation procedures to determine that they describe the steps necessary for secure installation of the TOE and the secure preparation of the operational environment in accordance with the security objectives in the ST.

The installation procedures should provide detailed information about the following, if applicable:
a) minimum system requirements for secure installation;
b) requirements for the operational environment in accordance with the security objectives provided by the ST;
c) the steps the user has to perform in order to get to an operational final TOE being commensurate with its evaluated configuration. Such a description shall include - for each step - a clear scheme for the decision on the next step depended on success, failure or problems at the current step;
d) changing the installation specific security characteristics of entities under the control of the TSF (for example parameters, settings, passwords);

e) handling exceptions and problems.

13.4.2.5 Action AGD_PRE.2.2E

AGD_PRE.2-3 The evaluator shall perform all user procedures necessary to prepare the initial TOE to determine that the initial TOE and its operational environment can be prepared securely using only the supplied preparative procedures.

Preparation requires the evaluator to advance the initial TOE from a deliverable state to the final TOE in which it is operational, including acceptance and installation of the final TOE, and enforcing the SFRs consistent with the security objectives for the final TOE specified in the ST.

The evaluator should follow only the developer's procedures and may perform the activities that customers are usually expected to perform to accept the additional code and install additional code in the initial TOE to obtain the final TOE, using the supplied preparative procedures only. Any difficulties encountered during such an exercise may be indicative of incomplete, unclear or unreasonable guidance. This work unit may be performed in conjunction with the evaluation activities under Independent testing (ATE_IND).

If it is known that the final TOE will be used as a dependent component for a composed TOE evaluation, then the evaluator should ensure that the operational environment is satisfied by the base component used in the composed TOE.

13.3.1 **Evaluation of sub-activity (AGD_DEV.1)**

Note: AGD_DEV.1 is a component written as a refinement of AGD_OPE.1 applied to the rules for development of application code and the final TOE. Items in blue “highlights” the change vs AGD_OPE.1.

13.3.1.1 Objectives

The objectives of this sub-activity are to determine whether the development guidance for application code and final TOE describes how to develop application code for all persons responsible of development of the additional code, provides instructions and guidelines (including warnings), helps to understand how to develop additional code in addition to initial TOE and how to create final TOE.
It facilitates prevention and detection of insecure TOE states, or whether it is misleading or unreasonable.

13.3.1.2 Input

The evaluation evidence for this sub-activity is:
a) the ST;
b) the functional specification;
c) the TOE design, if applicable;
d) the TOE implementation representation, if applicable;
e) the development guidance for application code and final TOE guidance;

13.3.1.3 Action AGD_DEV.1.1E

AGD_DEV.1.1C The development guidance for application code and final TOE shall describe how to develop application code to obtain final TOE without insecure TOE states.

AGD_DEV.1-1 The evaluator shall examine the development guidance for application code and final TOE to determine that it describes, for all persons responsible of development of the additional code, the instructions and guidelines, that should be controlled in a secure processing environment, including appropriate warnings.

AGD_DEV.1.2C The development guidance for application code and final TOE shall describe how to develop application code to be loaded on initial TOE to obtain final TOE.

AGD_DEV.1-2 The development guidance for application code and final TOE to determine that it describes, for all persons responsible of development of the additional code, the instructions and guidelines for the loading of the additional code to obtain a final TOE without insecure TOE states.

The development guidance for application code and final TOE should provide advice regarding effective use of the TSF resources to allow loading on initial TOE.

AGD_DEV.1.3C The development guidance for application code and final TOE shall describe how to perform acceptance of additional code and procedure to test final TOE prior delivery of additional code.

AGD_DEV.1-3 The evaluator shall examine the development guidance for application code and final TOE to determine that it describes, for all persons responsible of development of the additional code, the way to perform acceptance of additional code and procedure to test final TOE.

The development guidance for application code and final TOE should contain an overview of the operation to be done on application code to allow loading of application code on initial TOE. It also includes description of security functionality that is visible at the user interfaces to load application code.

AGD_DEV.1.4C The development guidance for application code and final TOE shall be clear and reasonable.

AGD_DEV.1-4 The evaluator shall examine the development guidance to determine that it describes, for all persons responsible of development of the additional code, each type of security-relevant event relative to the user functions that need to be checked, including changing the security characteristics of entities under the control of the TSF and operation following failure or operational error.

AGD_DEV.1-5 The evaluator shall examine the development guidance for application code and final TOE to determine that it is clear.

The development guidance is unclear if it can reasonably be misuse by all persons responsible of development of the additional code, in a way detrimental to the initial TOE, or to the security provided by the final TOE.

AGD_DEV.1-6 The evaluator shall examine the development guidance for application code and final TOE to determine that it is reasonable.

The development guidance is unreasonable if instructions and guidelines makes demands on the initial TOE's usage or operational environment that are inconsistent with the ST or unduly onerous to maintain security (including operation following failure or operational error), their consequences and implications for maintaining secure operation of TOE.

14.X.1 **Evaluation of sub-activity (ALC_CMA.1)**

Note: ALC_ CMA.1 is a component written as a refinement of ALC_CMC.1 applied to Application code and Final TOE. Items in blue “highlights” the change vs ALC_CMC.1.

The initial TOE, the additional Code, and the final TOE may be covered by same CM system and different versions of same CM plan. In such case, ALC_CMC.2 and ALC_CMA.1 may be covered by same evaluation activity.
When development activity for are the additional Code, and the final TOE are performed by different teams, locations or tools, ALC_CMC.1 evaluation activity has to be done separately from ALC_CMC.1.

14.X.1.1 Objectives
The objectives of this sub-activity are to determine whether the developer has clearly identified the additional Code, final TOE and referenced the initial TOE.

14.X.1.2 Input

The evaluation evidence for this sub-activity is:
a) the ST;
b) the initial TOE suitable for non-regression testing, and the additional Code, and the final TOE suitable for testing.

14. X.1.3 Action ALC_CMA.2.1E

ALC_CMA.1.1C The additional code **shall be labelled with its unique reference and link is created with** Final TOE**.**

ALC_CMA.1-1 The evaluator shall check that the additional Code, and the final TOE provided for evaluation are labelled with its own reference.

The evaluator should ensure that the additional Code, and the final TOE contains the unique reference which is stated in the ST.

This could be achieved through a label provided in guidance associated to the additional Code, and the final TOE to be checked using final TOE interface. This is to ensure that it would be possible for consumers to identify the final TOE (e.g. at the point of purchase or use).

The final TOE may provide a method by which the additional Code, and the final TOE can be easily identified. For example, a software TOE may display its name and version number during the startup routine, or in response to a command line entry.

Alternatively, the unique reference provided for the final TOE may be the combination of the unique reference of the additional code and each component from which the final TOE is comprised (e.g. in the case of a composed TOE).

Application notes
If developer provides only version of Final TOE directly to the user, the additional code must be identified in relevant documentation to allow to identify the additional code on initial TOE to obtain final TOE.

ALC_CMA.1-2 The evaluator shall check that the final TOE references used are consistent.

If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the final TOE, that they have installed this version, and that they have the correct version of the guidance to operate the final TOE in accordance with its ST.

The evaluator also verifies that the final TOE reference is consistent with the ST.

If this work unit is applied to a composed TOE, the following will apply. The composed TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate final TOE reference. It would require further development for the composed TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

14.X.2 **Evaluation of sub-activity (ALC_CMA.2)**

Note: ALC_ CMA.2 is a component written as a refinement of ALC_CMC.2 applied to Application code and Final TOE. Items in blue “highlights” the change vs ALC_CMC.2.

The initial TOE, the additional Code, and the final TOE may be covered by same CM system and different versions of same CM plan. In such case, ALC_CMC.2 and ALC_CMA.2 may be covered by same evaluation activity.
When development activity for are the additional Code, and the final TOE are performed by different teams, locations or tools, ALC_CMC.2 evaluation activity has to be done separately from ALC_CMC.2.

14.X.2.1 Objectives
The objectives of this sub-activity are to determine whether the developer uses a CM system that uniquely identifies the initial TOE, additional Code, final TOE and its associated configuration items.

14.X.2.2 Input

The evaluation evidence for this sub-activity is:
a) the ST;
b) the initial TOE suitable for non-regression testing; and the additional Code, and the final TOE suitable for testing;
c) the configuration management documentation.

Application notes
This component contains an implicit evaluator action to determine that the CM system is being used. As the requirements here are limited to identification of the TOE and provision of a configuration list, this action is already covered by, and limited to, the existing work units.

14. X.2.3 Action ALC_CMA.2.1E

ALC_CMA.2.1C The additional code and the final TOE shall be labelled each one with its unique reference.

ALC_CMA.2-1 The evaluator shall check that the additional Code, and the final TOE provided for evaluation are labelled with its own reference.

The evaluator should ensure that the additional Code, and the final TOE contains the unique reference which is stated in the ST.

This could be achieved through a label provided in guidance associated to the additional Code, and the final TOE to be checked using final TOE interface. This is to ensure that it would be possible for consumers to identify the final TOE (e.g. at the point of purchase or use).

The final TOE may provide a method by which the additional Code, and the final TOE can be easily identified. For example, a software TOE may display its name and version number during the startup routine, or in response to a command line entry.

Alternatively, the unique reference provided for the final TOE may be the combination of the unique reference of the additional code and each component from which the final TOE is comprised (e.g. in the case of a composed TOE).

ALC_CMA.2-2 The evaluator shall check that the final TOE references used are consistent.

If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the final TOE, that they have installed this version, and that they have the correct version of the guidance to operate the final TOE in accordance with its ST.

The evaluator also verifies that the final TOE reference is consistent with the ST.

If this work unit is applied to a composed TOE, the following will apply. The composed TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate final TOE reference. It would require further development for the composed TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

ALC_CMA.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items associated to additional code leading to the final TOE.

ALC_CMA.2-3 The evaluator shall examine the method of identifying configuration items associated to the additional code, and the final TOE to determine that it describes how configuration items are uniquely identified.

Procedures should describe how the status of each configuration item associated to the additional Code, and the final TOE can be tracked throughout the life-cycle of the final TOE. The procedures may be detailed in the CM plan or throughout the CM documentation. The information included should describe:

a) the method how each configuration item associated to the additional code, and the final TOE is uniquely identified, such that it is possible to track versions of the same configuration item;

b) the method how configuration items associated to the additional code, and the final TOE are assigned unique identifiers and how they are entered into the CM system;

c) the method to be used to identify superseded versions of a configuration item associated to the additional code, and the final TOE.

ALC_CMA.2.3C The CM system shall uniquely identify all configuration items associated to associated to the additional code, and the final TOE.

ALC_CMA.2-4 The evaluator shall examine the configuration items associated to the additional code, and the final TOE to determine that they are identified in a way that is consistent with the CM documentation.

Assurance that the CM system uniquely identifies all configuration items associated to the additional code, and the final TOE is gained by examining the identifiers for the configuration items. For configuration items identified under ALC_CMS, the evaluator confirms that each configuration item possesses a unique identifier in a manner consistent with the unique identification method that is described in the CM documentation.

14.X.3 **Evaluation of sub-activity (ALC_CMA.3)**

Note: ALC_ CMA.3 is a component written as a refinement of ALC_CMC.4 applied to Application code and Final TOE. Items in blue "highlights" the change vs ALC_CMC.4.

The initial TOE, the additional Code, and the final TOE may be covered by same CM system and different versions of same CM plan. In such case, ALC_CMC.4 and ALC_CMA.3 may be covered by same evaluation activity.

When development activity for are the additional Code, and the final TOE are performed by different teams, locations or tools, ALC_CMC.3 evaluation activity has to be done separately from ALC_CMC.4.

14.X.3.1 Objectives

The objectives of this sub-activity are to determine whether the developer has clearly identified the initial TOE, additional Code, final TOE and its associated configuration items, and whether the ability to modify these items is properly controlled by automated tools, thus making the CM system less susceptible to human error or negligence.

14.X.3.2 Input

The evaluation evidence for this sub-activity is:

a) the ST;

b) the initial TOE suitable for non-regression testing; and the additional Code, and the final TOE suitable for testing;

c) the configuration management documentation.

14.2.6.3 Action ALC_CMC.3.1E

ALC_CMA.3.1C The additional Code, and the final TOE shall be labelled each one with its unique reference.

ALC_CMA.3-1 The evaluator shall check that the additional Code, and the final TOE provided for evaluation are labelled with its own reference.

The evaluator should ensure that the additional Code, and the final TOE contains the unique reference which is stated in the ST.

This could be achieved through a label provided in guidance associated to the additional Code, and the final TOE to be checked using final TOE interface. This is to ensure that it would be possible for consumers to identify the final TOE (e.g. at the point of purchase or use).

The final TOE may provide a method by which the additional Code, and the final TOE can be easily identified. For example, a software TOE may display its name and version number during the startup routine, or in response to a command line entry.

Alternatively, the unique reference provided for the final TOE may be the combination of the unique reference of the additional code and each component from which the final TOE is comprised (e.g. in the case of a composed TOE).

ALC_CMA.3-2 The evaluator shall check that the final TOE references used are consistent.

If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the final TOE, that they have installed this version, and that they have the correct version of the guidance to operate the final TOE in accordance with its ST.

The evaluator also verifies that the final TOE reference is consistent with the ST.

If this work unit is applied to a composed TOE, the following will apply. The composed TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate final TOE reference. It would require further development for the composed TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

ALC_CMA.3.2C The CM documentation shall describe the method used to uniquely identify the configuration items associated to the additional code, and the final TOE.

ALC_CMA.3-3 The evaluator shall examine the method of identifying configuration items associated to the additional code, and the final TOE to determine that it describes how configuration items are uniquely identified.

Procedures should describe how the status of each configuration item associated to the additional Code, and the final TOE can be tracked throughout the life-cycle of the final TOE. The procedures may be detailed in the CM plan or throughout the CM documentation. The information included should describe:

a) the method how each configuration item associated to the additional code, and the final TOE is uniquely identified, such that it is possible to track versions of the same configuration item;

b) the method how configuration items associated to the additional code, and the final TOE are assigned unique identifiers and how they are entered into the CM system;

c) the method to be used to identify superseded versions of a configuration item associated to the additional code, and the final TOE.

ALC_CMA.3.3C The CM system shall uniquely identify all configuration items associated to associated to the additional code, and the final TOE.

ALC_CMA.3-4 The evaluator shall examine the configuration items associated to the additional code, and the final TOE to determine that they are identified in a way that is consistent with the CM documentation.

Assurance that the CM system uniquely identifies all configuration items associated to the additional code, and the final TOE is gained by examining the identifiers for the configuration items. For configuration items identified under ALC_CMS, the evaluator confirms that each configuration item possesses a unique identifier in a manner consistent with the unique identification method that is described in the CM documentation.

ALC_CMA.3.4C The CM system shall provide automated measures such that no changes are made to the configuration items associated to the initial TOE **and that only authorised changes are made to the configuration items associated to** the additional code, and the final TOE.

ALC_CMA.3-5 The evaluator shall examine the CM access control measures described in the CM plan (cf. ALC_CMA.3.6C) to determine that they are automated and effective in preventing unauthorised access and modification to the configuration items associated to initial TOE, and unauthorised access to the configuration items associated to the additional code, and the final TOE.

The evaluator may use a number of methods to determine that the CM access control measures are effective. For example, the evaluator may exercise the access control measures to ensure that the procedures could not be bypassed. The evaluator may use the outputs generated by the CM system procedures required by ALC_CMA.3.10C. The evaluator may also witness a demonstration of the CM system to ensure that the access control measures employed are operating effectively.

ALC_CMA.3.5C The CM system shall support the production of the additional code, and the final TOE by automated means.

ALC_CMA.3-6 The evaluator shall check the CM plan (cf. ALC_CMA.3.6C) for automated procedures for supporting the production of the additional code, and the final TOE.

The term “production” applies to those processes adopted by the developer to progress the additional code from the implementation representation to a state acceptable for delivery to the end customer.

The evaluator verifies the existence of automated production support procedures within the CM plan.

The following are examples for automated means supporting the production of the TOE:

a “make” tool (as provided with many software development tools) in the case of a software TOE.

ALC_CMA.3-7 The evaluator shall examine the TOE production support procedures to determine that they are effective in ensuring that the additional code, and the final TOE is generated that reflects its implementation representation.

The production support procedures should describe which tools have to be used to produce the additional code, and the final TOE from the implementation representation in a clearly defined way. The conventions, directives, or other necessary constructs are described under ALC_TAT.

The evaluator determines that by following the production support procedures the correct configuration items would be used to generate the additional code, and the final TOE. For example, in a software TOE this may include checking that the automated production procedures ensure that all source files and related libraries are included in the compiled object code. Moreover, the procedures should ensure that compiler options and comparable other options are defined uniquely.

The customer can then be confident that the version of the additional code delivered for installation is derived from its implementation representation in an unambiguous way and the final TOE implements the SFRs as described in the ST.

The evaluator should bear in mind that the CM system need not necessarily possess the capability to produce the additional code, and the final TOE, but should provide support for the process that will help reduce the probability of human error.

ALC_CMA.3.6C The CM documentation shall include a CM plan for the additional code, and the final TOE.

ALC_CMA.3-8 The evaluator shall check that the CM documentation provided includes a CM plan.

The CM plan does not need to The CM plan does not need to be contained within a single document, but it is recommended that there is a separate document that describes where the various parts of the CM plan can be found. If the CM plan is provided by a set of documents, the list in the following work unit gives guidance regarding the required content.

ALC_CMA.3.7C The CM plan shall describe how the CM system is used for the restricted access to the initial TOE and **the development of the** additional code, and the final TOE.

ALC_CMA.3-9 The evaluator shall examine the CM plan to determine that it describes how the CM system is used for the restricted access to the initial TOE and for the development of the additional code, and the final TOE.

The descriptions contained in a CM plan include, if applicable:

a) all activities performed in the restricted access to the initial TOE and the development of the additional code, and the final TOE that are subject to configuration management procedures (e.g. creation, modification or deletion of a configuration item, data-backup, archiving);
b) which means (e.g. CM tools, forms) have to be made available;
c) the usage of the CM tools: the necessary details for a user of the CM system to be able to operate the CM tools correctly in order to maintain the integrity of the additional code, and the final TOE;
d) the production support procedures;
e) which other objects (development components, tools, assessment environments, etc) are taken under CM control;
f) the roles and responsibilities of individuals required to perform operations on individual configuration items associated to initial TOE, the additional code, and the final TOE (different roles may be identified for different types of configuration items (e.g. design documentation or source code));
g) how CM instances (e.g. change control boards, interface control working groups) are introduced and staffed;
h) the description of the change management;
i) the procedures that are used to ensure that only authorised individuals can make changes to configuration items associated to the additional code, and the final TOE;
j) the procedures that are used to ensure that concurrency problems do not occur as a result of simultaneous changes to configuration items associated to the additional code, and the final TOE;
k) the evidence that is generated as a result of application of the procedures.

For example, for a change to a configuration item, the CM system might record a description of the change, accountability for the change, identification of all configuration items affected, status (e.g. pending or completed), and date and time of the change. This might be recorded in an audit trail of changes made or change control records;
l) the approach to version control and unique referencing of versions of the additional code, and of the final TOE.

ALC_CMA.3.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the additional code, and the final TOE**.**

ALC_CMA.3-10 The evaluator shall examine the CM plan to determine that it describes the procedures used to accept modified or newly created configuration items as parts of the additional code, and the final TOE.

The descriptions of the acceptance procedures in the CM plan should include the developer roles or individuals responsible for the acceptance and the criteria to be used for acceptance. They should take into account all acceptance situations that may occur, in particular:

a) accepting an item into the CM system for the first time, in particular inclusion of software, firmware and hardware components from other manufacturers into the TOE (“integration”);

b) moving configuration items to the next life-cycle phase at each stage of the construction of the TOE (e.g. module, subsystem, system);

c) subsequent to transports between different development sites.

If this work unit is applied to a dependent component that is going to be integrated in a composed TOE, the CM plan should consider the control of base components obtained by the dependent TOE developer.

When obtaining the components the evaluators are to verify the following:

a) Transfer of each base component from the base component developer to the integrator (dependent TOE developer) was performed in accordance with the base component TOE's secure delivery procedures, as reported in the base component TOE certification report.

b) The component received has the same identifiers as those stated in the ST and Certification Report for the component TOE.

c) All additional material required by a developer for composition (integration) is provided. This is to include the necessary extract of the component TOE's functional specification.

ALC_CMA.3.9C The evidence shall demonstrate that all configuration items of the additional code, and the final TOE **are being maintained under the CM system.**

ALC_CMA.3-11 The evaluator shall check that the configuration items associated to the additional code, and the final TOE identified in the configuration list are being maintained by the CM system.

The CM system employed by the developer should maintain the integrity of the associated to the additional code, and the final TOE. The evaluator should check that for each type of configuration item (e.g. design documents or source code modules) contained in the configuration list there are examples of the evidence generated by the procedures described in the CM plan. In this case, the approach to sampling will depend upon the level of granularity used in the CM system to control CM items. Where, for example, 10,000 source code modules are identified in the configuration list, a different sampling strategy needs to be applied compared to the case in which there are only 5, or even 1. The emphasis of this activity should be on ensuring that the CM system is being operated correctly, rather than on the detection of any minor error.

For guidance on sampling see A.2, Sampling.

ALC_CMA.3.10C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMA.3-12 The evaluator shall check the CM documentation to ascertain that it includes the CM system records identified by the CM plan.

The output produced by the CM system should provide the evidence that the evaluator needs to be confident that the CM plan is being applied, and also that all configuration items associated to the additional code, and the final TOE are being maintained by the CM system as required by ALC_CMA.3.9C. Example output could include change control forms, or configuration item associated to the additional code, and the final TOE access approval forms.

ALC_CMA.3-13 The evaluator shall examine the evidence to determine that the CM system is being operated in accordance with the CM plan.

The evaluator should select and examine a sample of evidence covering each type of CM-relevant operation that has been performed on a configuration item associated to the additional code, and the final TOE (e.g. creation, modification, deletion, reversion to an earlier version) to confirm that all operations of the CM system have been carried out in line with documented procedures. The evaluator confirms that the evidence includes all the information identified for that operation in the CM plan. Examination of the evidence may require access to a CM tool that is used. The evaluator may choose to sample the evidence.

For guidance on sampling see A.2, Sampling.

Further confidence in the correct operation of the CM system and the effective maintenance of configuration items may be established by means of interviews with selected development staff. In conducting such interviews, the evaluator aims to gain a deeper understanding of how the CM system is used in practice as well as to confirm that the CM procedures are being applied as described in the CM documentation. Note that such interviews should complement rather than replace the examination of documentary evidence, and may not be necessary if the documentary evidence alone satisfies the requirement. However, given the wide scope of the CM plan it is possible that some aspects (e.g. roles and responsibilities) may not be clear from the CM plan and records alone. This is one case where clarification may be necessary through interviews.
It is expected that the evaluator will visit the development site in support of this activity.

For guidance on site visits see A.4, Site Visits.

14.4.2 Evaluation of sub-activity (ALC_DEL.2)

ALC_DEL.2 is a component written as a refinement of ALC_DEL.1 applied to Application code and Final TOE. Items in blue “highlights” the change vs ALC_DEL.1.

14.4.2.1 Objectives
The objective of this sub-activity is to determine whether the delivery documentation for additional code deployment describes all procedures used to maintain security of the additional code when distributing the additional code to the user and downloading it into the initial TOE to obtain the final TOE.
14.4.2.2 Input
1122 The evaluation evidence for this sub-activity is:
a) the ST;
b) the delivery documentation for additional code deployment.

14.4.1.3 Action ALC_DEL.2.1E
ALC_DEL.2.1C ***The delivery documentation for additional code deployment shall describe all procedures that are necessary to maintain security when distributing versions of the additional code to the consumer to obtain the final TOE.***

ALC_DEL.2-1 The evaluator **shall examine** the delivery documentation for additional code deployment to determine that it describes all procedures that are necessary to maintain security when distributing versions of the additional code to the consumer to obtain the final TOE.

The delivery documentation describes proper procedures to maintain security of the additional code during transfer of the additional code and to determine the identification of the additional code.

The delivery documentation should cover the additional code, but may contain different procedures for different parts of the additional code. The evaluation should consider the totality of procedures.

The delivery procedures should be applicable across all phases of delivery from the development environment to the deployment environment.

Cryptographic checksums or signature must be used by the developer to ensure that tampering or masquerading can be detected during transfer, storage, downloading and insertion in final TOE. Encryption must be used by the developer to ensure confidentiality of Additional code until its insertion in Final TOE.

Interpretation of the term “necessary to maintain security” will need to consider:
* The overall security level stated for the final TOE by the chosen level of the Vulnerability Assessment. If the final TOE is required to be resistant against attackers of a certain potential in its intended environment, this should also apply to the delivery of the additional code.

The evaluator should determine that a balanced approach has been taken, such that delivery does not present a weak point in an otherwise secure development process.

The security objectives provided by the ST. The emphasis in the delivery documentation is likely to be on measures related to integrity, as integrity of the final TOE is always important. However, confidentiality and availability of the delivery will be of concern in the delivery of some TOEs; procedures relating to these aspects of the secure delivery should also be discussed in the procedures.

14.4.1.4 Implied evaluator action

ALC_DEL.2.2D ***The developer shall use the delivery procedures for additional code deployment.***

ALC_DEL.2-2 The evaluator ***shall examine*** aspects of the delivery process for additional code deployment to determine that the delivery procedures are used.

The approach taken by the evaluator to check the application of delivery procedures will depend on the delivery process itself. In addition to examination of the procedures themselves, the evaluator seeks some assurance that they are applied in practice.

Some possible approaches are:

a) a visit to the distribution site(s) where practical application of the procedures may be observed;
b) examination of the additional code at some stage during delivery, or after the user has received it and installed in final TOE.
c) observing that the process is applied in practice when the evaluator obtains the final TOE.
d) questioning end users as to how the additional code was delivered.

For guidance on site visits see A.4, Site Visits.

It may be the case of a newly developed TOE that the delivery procedures have yet to be exercised. In these cases, the evaluator has to be satisfied that appropriate procedures and facilities are in place for future deliveries and that all personnel involved are aware of their responsibilities.

The evaluator may request a “dry run” of a delivery if this is practical. If the developer has produced other similar products, then an examination of procedures in their use may be useful in providing assurance.

14.6.4 Evaluation of sub-activity (ALC_FLR.4)

Note: ALC_FLR.4 is a component written as a refinement of ALC_FLR.2 applied to the application code and the final TOE. Items in blue "highlights" the change vs ALC_FLR.2.

14.6.4.1 Objectives

The objective of this sub-activity is to determine whether the developer has established flaw remediation procedures that describe the tracking of security flaws, the identification of corrective actions, and the distribution of corrective action information to TOE users. Additionally, this sub-activity determines whether the developer's procedures provide for the corrections of security flaws, for the receipt of flaw reports from TOE users, and for assurance that the corrections introduce no new security flaws. This component does not produce any information about additional code used to correct the flaw nor type of evidences associated to development or deployment of such additional code

14.6.4.2 Input

The evaluation evidence for this sub-activity is:

a) the flaw remediation procedures documentation;
b) the flaw remediation guidance documentation.

14.6.4.3 Action ALC_FLR.4.1E

ALC_FLR.4.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.4-1 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes the procedures used to track all reported security flaws in each release of the TOE.

The procedures describe the actions that are taken by the developer from the time each suspected security flaw is reported to the time that it is resolved. This includes the flaw's entire time frame, from initial detection through ascertaining that the flaw is a security flaw, to resolution of the security flaw.
If a flaw is discovered not to be security-relevant, there is no need (for the purposes of the Flaw remediation (ALC_FLR) requirements) for the flaw remediation procedures to track it further; only that there be an explanation of why the flaw is not security-relevant.
Note: Details and associated rationale concerning flaw security relevance will be covered by ALC_IAR family and components.

ALC_FLR.4.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.4-2 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would produce a description of each security flaw in terms of its nature and effects.
The procedures identify the actions that are taken by the developer to describe the nature and effects of each security flaw in sufficient detail to be able to reproduce it. The description of the nature of a security flaw addresses whether it is an error in the documentation, a flaw in the design of the TSF, a flaw in the implementation of the TSF, etc. The description of the security flaw's effects identifies the portions of the TSF that are affected and how those portions are affected. For example, a security flaw in the implementation might be found that affects the identification and authentication enforced by the TSF by permitting authentication with the password "BACKDOOR".

ALC_FLR.2-3 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would identify the status of finding a correction to each security flaw.
The flaw remediation procedures identify the different stages of security flaws. This differentiation includes at least: suspected security flaws that have been reported, suspected security flaws that have been confirmed to be security flaws, and security flaws whose solutions have been implemented. It is permissible that additional stages (e.g. flaws that have been reported but not yet investigated, flaws that are under investigation, security flaws for which a solution has been found but not yet implemented) be included.

ALC_FLR.4.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.4-4 The evaluator ***shall check*** the flaw remediation procedures to determine that the application of these procedures would identify the corrective action for each security flaw.

Corrective action may consist of a repair to the hardware, firmware, or software portions of the TOE, a modification of TOE guidance, or both. Corrective action that constitutes modifications to TOE guidance (e.g. details of procedural measures to be taken to obviate the security flaw) includes both those measures serving as only an interim solution (until the repair is issued) as well as those serving as a permanent solution (where it is determined that the procedural measure is the best solution).
If the source of the security flaw is a documentation error, the corrective action consists of an update of the affected TOE guidance. If the corrective action is a procedural measure, this measure will include an update made to the affected TOE guidance to reflect these corrective procedures.

Note: The scope is the flaw and correction action (as deploying additional code). Details about additional code correcting the flaw will be covered by ALC_IAR family and components.

ALC_FLR.4.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.4-5 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes a means of providing the TOE users with the necessary information on each security flaw.
The necessary information about each security flaw consists of its description (not necessarily at the same level of detail as that provided as part of work unit ALC_FLR.2-2), the prescribed corrective action, and any associated guidance on implementing the correction.
TOE users may be provided with such information, correction, and documentation updates in any of several ways, such as their posting to a website, their being sent to TOE users, or arrangements made for the developer to install the correction. In cases where the means of providing this information requires action to be initiated by the TOE user, the evaluator examines any TOE guidance to ensure that it contains instructions for retrieving the information.
The only metric for assessing the adequacy of the method used for providing the information, corrections and guidance is that there be a reasonable expectation that TOE users can obtain or receive it. For example, consider the method of dissemination where the requisite data is posted to a website for one month, and the TOE users know that this will happen and when this will happen. This may not be especially reasonable or effective (as, say, a permanent posting to the website), yet it is feasible that the TOE user could obtain the necessary information. On the other hand, if the information were posted to the website for only one hour, yet TOE users had no way of knowing this or when it would be posted, it is infeasible that they would ever get the necessary information.

ALC_FLR.4.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.4-6 The evaluator ***shall examine*** the flaw remediation procedures to determine that they describe procedures for the developer to accept reports of security flaws or requests for corrections to such flaws.

The procedures ensure that TOE users have a means by which they can communicate with the TOE developer. By having a means of contact with the developer, the user can report security flaws, enquire about the status of security flaws, or request corrections to flaws. This means of contact may be part of a more general contact facility for reporting non-security related problems.

The use of these procedures is not restricted to TOE users; however, only the TOE users are actively supplied with the details of these procedures. Others who might have access to or familiarity with the TOE can use the same procedures to submit reports to the developer, who is then expected to process them. Any means of submitting reports to the developer, other than those identified by the developer, are beyond the scope of this work unit; reports generated by other means need not be addressed.

ALC_FLR.4.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.4-7 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would help to ensure every reported flaw is corrected.

The flaw remediation procedures cover not only those security flaws discovered and reported by developer personnel, but also those reported by TOE users. The procedures are sufficiently detailed so that they describe how it is ensured that each reported security flaw is corrected. The procedures contain reasonable steps that show progress leading to the eventual, inevitable resolution.

The procedures describe the process that is taken from the point at which the suspected security flaw is determined to be a security flaw to the point at which it is resolved.

ALC_FLR.4-8 The evaluator ***shall examine*** the flaw remediation procedures to determine that the application of these procedures would help to ensure that the TOE users are issued remediation procedures for each security flaw.

The procedures describe the process that is taken from the point at which a security flaw is resolved to the point at which the remediation procedures are provided. The procedures for delivering corrective actions should be consistent with the security objectives; they need not necessarily be identical to the procedures used for delivering the TOE, as documented to meet ALC_DEL, if included in the assurance requirements.

Note: ALC_DEL.2 component has been create to address specifically such purpose of delivery of additional code. It is used in advance to measure the ability of the developer to perform a secure delivery of additional code. It will be also used during evaluation process to check evidences of effective delivery on changed TOE.

Information of any delivery of new additional code must be transfer to laboratory and certification body to be reviewed at the appropriate moment.

ALC_FLR.4.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.4-9 The evaluator shall examine the flaw remediation procedures to determine that the application of these procedures would result in safeguards that the potential correction contains no adverse effects.

Through analysis, testing, or a combination of the two, the developer may reduce the likelihood that adverse effects will be introduced when a security flaw is corrected. The evaluator assesses whether the procedures provide detail in how the necessary mix of analysis and testing actions is to be determined for a given correction.
The evaluator also determines that, for instances where the source of the security flaw is a documentation problem, the procedures include the means of safeguarding against the introduction of contradictions with other documentation.

Note: Details about non regression testing are covered by new ALC_IAR family.

ALC_FLR.4.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.4-10 The evaluator ***shall examine*** the flaw remediation guidance to determine that the application of these procedures would result in a means for the TOE user to provide reports of suspected security flaws or requests for corrections to such flaws.
The guidance ensures that TOE users have a means by which they can communicate with the TOE developer. By having a means of contact with the developer, the user can report security flaws, enquire about the status of security flaws, or request corrections to flaws.

ALC_FLR.4.9C "Additional code" creation procedures shall demonstrate how the confidentiality, integrity and authenticity of the Additional code is assumed prior code delivery.

ALC_FLR.4-11 The evaluator ***shall examine*** the delivery procedure for Additional code to determine that the security properties associated to additional code will be maintained prior Additional code delivery.

Note: ALC_DEL.2 component has been created to address specifically such purpose of verification of rules applied to delivery of additional code. It is used in advance to measure the ability of the developer to perform a secure delivery of additional code.

It will be also used during evaluation process to check evidences of effective delivery on changed TOE.

ALC_FLR.4.10C "Additional code" creation procedures shall demonstrate how the confidentiality, integrity and authenticity of the Additional code is maintained until code loading.

ALC_FLR.4-12 The evaluator ***shall examine*** the deployment guidance for Additional code and final TOE to determine that the security properties associated to additional code will be maintained during Additional code delivery.
Note: AGD_OPE.3 component has been created to address specifically such purpose of verification of rules applied to deployment of additional code. It is used in advance to measure the ability of the developer to define rules for secure deployment of additional code to obtain final TOE.
It will be also used during evaluation process to check evidences of effective deployment on changed TOE.
ALC_FLR.4.11C The flaw remediation guidance shall provide detailed instructions for users on how to check the availability of new "Additional Code" patches and how to apply them.

ALC_FLR.4-13 The evaluator ***shall examine*** the flaw remediation procedures documentation to determine that it describes a means of providing the TOE users with the necessary information on each Additional code.

The necessary information about each Additional code availability consists of its description and availability date and way to obtain it.

TOE users may be provided with such information about additional code in any of several ways, such as their posting to a website, their being sent to TOE users, or arrangements made for the developer to install the correction. In cases where the means of providing this information requires action to be initiated by the TOE user, the evaluator examines any TOE guidance to ensure that it contains instructions for retrieving the information.

14.X.1 Evaluation of sub-activity (ALC_IAR.1)

Note: ALC_IAR.1 is a component written from a base of ALC_FLR.2 applied to the non-security relevant change of initial TOE using application code to obtain a final TOE.

14.X.1.1 Objectives

The objective of this sub-activity is to determine whether the Impact analysis applied to non-security relevant change to the TOE have been documented and result in an accurate assurance level that additional code addition to initial TOE allow to obtain a secure final TOE.

14.X.1.2 Input

The evaluation evidence for this sub-activity is:

a) the ST of the initial TOE;
b) the ST of the final TOE;
c) the Impact Analysis report;
d) the evidences associated to developer's application code development procedures, if applicable;
e) the evidences associated to developer's application code deployment procedures, if applicable;
f) the evidences associated to developer's final TOE acceptance procedures, if applicable;
g) the evidences associated to developer’s additional code;
h) the evidences associated to developer’s final TOE;

14.x.1.3 Application notes

The impact analysis applied to non-security relevant change to the TOE refer to tasks that are necessary to document activities required by additional code development without security change to TOE and activities required to analyze such evidences, to provide enough assurance that addition of application code allows to obtain a secure final TOE.

14.x.1.4 Action ALC_IAR.1.1E

ALC_IAR.1.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE.

ALC_IAR.1-1 The evaluator ***shall examine*** the impact analysis procedures documentation to determine that it describes the procedures used to track all reported changes in each additional code associated to a release of the initial TOE.

The procedures describe the actions that are taken by the developer to analyze, to implement, to test to manage, to deliver and to deploy additional code.

ALC_IAR.1.2C The impact analysis procedures shall require that a description of the nature and effect of each change be provided, as well as the status of security relevance of the changes and associated rationale.

ALC_IAR.1-2 The evaluator shall examine that change implemented in an initial TOE as an additional code is characterized by its effect and its non-security relevance with a rationale of such decision.

The procedures identify the actions that are taken by the developer to describe the nature and effects of each change in initial TOE as additional code in sufficient detail to be able to reproduce the analysis. The description of the nature of a change addresses whether it is an update in the guidance documentation, a change in functional behavior versus the initial TOE limiting or extending a service, or changing the configuration of a service but without impacting the security.

ALC_IAR.1.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.1-3 Change in initial TOE may consist of a modification of software portions of the TOE, a modification of TOE guidance, or both. It is an improvement when the change is not associated to a customer request introduced as a flaw of the initial TOE.
It is a corrective action if it is introduced after a customer request or an internal discovery of a functional error that it is not security relevant.
Demonstration requires that such change is not linked with TSS implementing SFR.

ALC_IAR.1.4C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for non-security relevant modification as defined for application code and Final TOE development.

ALC_IAR.1-4 The evaluator shall examine the documentation associated to the development of application code to be used as change in initial TOE to determine if demonstration of non-security relevance is accurate, in particular that such change is not linked with TSS implementing SFR and there is no direct link between the change and any TSFI.

The necessary information about each change consists of its description as illustrated in IAR template provided in annex A.

ALC_IAR.1.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for non-security relevant modification defined for application code development.

ALC_IAR.1-5 The evaluator shall examine the documentation associated to the change to determine if rules defined in development guidance for additional code and final TOE have been followed, in particular concerning evidences of non-regression testing as part of ATE_FUN.

The evaluator shall determine if any part of independent testing requires to be executed as part of ATE_IND families based on evidences produced in TOE functional testing.

14.X.2 Evaluation of sub-activity (ALC_IAR.2)

Note: ALC_IAR.2 is a component an extension of ALC_IAR.1 applicable to the change of initial TOE using application code to obtain a final TOE, where the change is security relevant but it is limited to the TOE implementation.

14.X.2.1 Objectives

The objective of this sub-activity is to determine whether the Impact analysis applied to security change in the TOE implementation have been documented and result in an accurate assurance level that additional code addition to initial TOE allow to obtain a secure final TOE.

14.X.2.2 Input

The evaluation evidence for this sub-activity is:

a) the ST of the initial TOE;

b) the ST of the final TOE;

c) the Impact Analysis report;

d) the evidences associated to developer's application code development procedures, if applicable;

e) the evidences associated to developer's final TOE acceptance procedures;

g) the evidences associated to developer’s final TOE representation;

14.x.2.3 Application notes

The impact analysis applied to security change in the TOE implementation refer to tasks that are necessary to document activities required by additional code development including security change in the TOE implementation and activities required to analyze such evidences, to provide enough assurance to obtain a secure final TOE.

14.x.2.4 Action ALC_IAR.2.1E

ALC_IAR.2.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE.

ALC_IAR.2-1 The evaluator ***shall examine*** the impact analysis procedures documentation to determine that it describes the procedures used to track all reported changes in additional code and TOE implementation representing the final TOE.

The procedures describe the actions that are taken by the developer to analyze, to implement, to test to manage, to deliver and to deploy additional code.

ALC_IAR.2.2C The evaluator shall examine that change implemented in an initial TOE as an additional code is characterized by its effect and its non-security relevance with a rationale of such decision.

ALC_IAR.2-2 The evaluator shall examine that change implemented in an initial TOE as an additional code is characterized by its effect and its security relevance limited to the TOE implementation with a rationale of such decision.

The procedures identify the actions that are taken by the developer to describe the nature and effects of each change in initial TOE as additional code in sufficient detail to be able to reproduce the analysis. The description of the nature of a change addresses why only the TOE implementation is impacted, in particular why TSFI and security relevant internal interfaces are not modified.

ALC_IAR.2.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.2-3 Change in TOE implementation of initial TOE consists in a modification of software portions of the TOE that may require a modification of TOE guidance. It is an improvement when the change is not associated to a customer request introduced as a flaw of the initial TOE.
It is a corrective action if it is introduced after a customer request, a publishing of a new CVE impacting the TOE, or an internal discovery of a security error.

ALC_IAR.2.4C **The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for security relevant modification limited to the TOE implementation as defined for application code and Final TOE development.**

ALC_IAR.2-4 The evaluator shall examine the documentation associated to the development of application code to be used as change in initial TOE to determine if demonstration of security relevance limited to the TOE representation is accurate, in particular that such change is not linked with modification of any TSFI or security relevant internal interface.

ALC_IAR.2.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of additional code has been performed consistently with rules for security relevant modification in TOE implementation defined for application code and Final TOE development.

ALC_IAR.2-5 The evaluator shall examine the documentation associated to the change to determine if rules defined in development guidance for additional code and final TOE have been followed, in particular concerning evidences associated to TOE implementation.

ALC_IAR.2.6C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for TOE functional testing defined for Final TOE development.

ALC_IAR.2-6 The evaluator shall examine the documentation associated to the change to determine if rules defined in development guidance for additional code and final TOE have been followed, in particular concerning evidences of non-regression testing and changes introduced in security test plan, expected test results and test coverage as part of ATE_FUN.

ALC_IAR.2-7 The evaluator shall determine what part of independent testing requires to be executed as part of ATE_IND families based on evidences produced in TOE functional testing.

ALC_IAR.2-8 The evaluator shall determine if any part of penetration testing requires to be executed as part of AVA_VAN families based on evidences produced in TOE functional testing.

14.X.3 Evaluation of sub-activity (ALC_IAR.3)

Note: ALC_IAR.3 is a component an extension of ALC_IAR.1 applicable to the change of initial TOE using application code to obtain a final TOE, where the change is security relevant but it is limited to the TOE implementation.

14.X.3.1 Objectives

The objective of this sub-activity is to determine whether the Impact analysis applied to security change in the TOE representation have been documented and result in an accurate assurance level that additional code addition to initial TOE allow to obtain a secure final TOE.

14.X.3.2 Input

The evaluation evidence for this sub-activity is:

a) the ST of the initial TOE;
b) the ST of the final TOE;
c) the Impact Analysis report;
d) the evidences associated to developer's application code development procedures, if applicable;
e) the evidences associated to developer's application code deployment procedures, if applicable;
f) the evidences associated to developer's final TOE acceptance procedures;
g) the evidences associated to developer’s additional code;
h) the evidences associated to developer’s final TOE;

14.x.3.3 Application notes

The impact analysis applied to security change in the TOE representation refer to tasks that are necessary to document activities required by additional code development including security change in the TOE representation and activities required to analyze such evidences, to provide enough assurance that addition of application code allows to obtain a secure final TOE.

14.x.3.4 Action ALC_IAR.3.1E

ALC_IAR.3.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE.

ALC_IAR.3-1 The evaluator ***shall examine*** the impact analysis procedures documentation to determine that it describes the procedures used to track all reported changes in each additional code associated to a release of the initial TOE.

The procedures describe the actions that are taken by the developer to analyze, to implement, to test to manage, to deliver and to deploy additional code.

ALC_IAR.3.2C The evaluator shall examine that change implemented in an initial TOE as an additional code is characterized by its effect and its non-security relevance with a rationale of such decision.

ALC_IAR.3-2 The evaluator shall examine that change implemented in an initial TOE as an additional code is characterized by its effect and its security relevance limited to the TOE implementation with a rationale of such decision.

The procedures identify the actions that are taken by the developer to describe the nature and effects of each change in initial TOE as additional code in sufficient detail to be able to reproduce the analysis. The description of the nature of a change addresses why only the TOE implementation is impacted, in particular why TSFI and security relevant internal interfaces are not modified.

ALC_IAR.3.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.3-3 Change in TOE representation of initial TOE consists in a modification of software portions of the TOE that may require a modification of TOE guidance. It is an improvement when the change is not associated to a customer request introduced as a flaw of the initial TOE.
It is a corrective action if it is introduced after a customer request, a publishing of a new CVE impacting the TOE, or an internal discovery of a security error.

ALC_IAR.3.4C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for security relevant modification limited to the TOE representation as defined for application code and Final TOE development.

ALC_IAR.3-4 The evaluator shall examine the documentation associated to the development of application code to be used as change in initial TOE to determine if demonstration of security relevance limited to the TOE representation is accurate, in particular that such change is not linked with modification of any TSFI or security relevant internal interface.

ALC_IAR.3.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of additional code has been performed consistently with rules for security relevant modification in TOE representation defined for application code and Final TOE development.

ALC_IAR.3-5 The evaluator shall examine the documentation associated to the change to determine if rules defined in development guidance for additional code and final TOE have been followed, in particular concerning evidences associated to TOE representation as part of ADV_IMP and ADV_INT families.

ALC_IAR.3.6C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for TOE functional testing defined for Final TOE development.

ALC_IAR.3-6 The evaluator shall examine the documentation associated to the change to determine if rules defined in development guidance for additional code and final TOE have been followed, in particular concerning evidences of non-regression testing and changes introduced in security test plan, expected test results and test coverage as part of ATE_FUN, ADV_COV.

ALC_IAR.3-7 The evaluator shall determine what part of independent testing requires to be executed as part of ATE_IND families based on evidences produced in TOE functional testing.

ALC_IAR.3-8 The evaluator shall determine if any part of penetration testing requires to be executed as part of AVA_VAN families based on evidences produced in TOE functional testing.

Proposal for new SAR components and Packages in CC for Patch Management

14.X.4 Evaluation of sub-activity (ALC_IAR.4)

Note: ALC_IAR.4 is a component an extension of ALC_IAR.2 applicable to the change of initial TOE using application code to obtain a final TOE, where the change is security relevant but it is not limited to the TOE implementation allowing change in TOE design and specification.

14.X.4.1 Objectives

The objective of this sub-activity is to determine whether the Impact analysis applied to security change in the TOE design have been documented and result in an accurate assurance level that additional code addition to initial TOE allow to obtain a secure final TOE.

14.X.4.2 Input

The evaluation evidence for this sub-activity is:

a) the ST of the initial TOE;
b) the ST of the final TOE;
c) the Impact Analysis report;
d) the evidences associated to developer's application code development procedures, if applicable;
e) the evidences associated to developer's application code deployment procedures, if applicable;
f) the evidences associated to developer's final TOE acceptance procedures;
g) the evidences associated to developer’s additional code;
h) the evidences associated to developer’s final TOE;

14.x.4.3 Application notes

The impact analysis applied to security change in the TOE design refer to tasks that are necessary to document activities required by additional code development including security change in the TOE design and activities required to analyze such evidences, to provide enough assurance that addition of application code allows to obtain a secure final TOE.

14.x.4.4 Action ALC_IAR.4.1E

ALC_IAR.4.1C The impact analysis procedures documentation shall describe the procedures used to track and to analyze all reported changes to initial TOE.

ALC_IAR.4-1 The evaluator ***shall examine*** the impact analysis procedures documentation to determine that it describes the procedures used to track all reported changes in each additional code associated to a release of the initial TOE.

The procedures describe the actions that are taken by the developer to analyze, to implement, to test to manage, to deliver and to deploy additional code.

ALC_IAR.4.2C The evaluator shall examine that change implemented in an initial TOE as an additional code is characterized by its effect and its non-security relevance with a rationale of such decision.

ALC_IAR.4-2 The evaluator shall examine that change implemented in an initial TOE as an additional code is characterized by its effect and its security relevance limited to the TOE implementation with a rationale of such decision.

The procedures identify the actions that are taken by the developer to describe the nature and effects of each change in initial TOE as additional code in sufficient detail to be able to reproduce the analysis.

The description of the nature of a change addresses why only the TOE implementation is impacted, in particular why TSFI and security relevant internal interfaces are not modified.

ALC_IAR.4.3C The impact analysis procedures shall require that each change is identified as correction of a flaw or as an improvement.

ALC_IAR.4-3 Change in TOE representation of initial TOE consists in a modification of software portions of the TOE that may require a modification of TOE guidance. It is an improvement when the change is not associated to a customer request introduced as a flaw of the initial TOE.
It is a corrective action if it is introduced after a customer request, a publishing of a new CVE impacting the TOE, or an internal discovery of a security error.

ALC_IAR.4.4C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of application code to be used as change in initial TOE has been performed consistently with rules for security relevant modification to the TOE implementation and all the different TOE representations as TOE implementation representation, TOE design or specification as defined for application code and Final TOE development.

ALC_IAR.4-4 The evaluator shall examine the documentation associated to the development of application code to be used as change in initial TOE to determine if demonstration of security relevance in TOE implementation and all the different TOE representations as TOE implementation representation, TOE design or specification and TOE representation is accurate.

ALC_IAR.4.5C The impact analysis procedures documentation shall describe the methods used to demonstrate that development of additional code has been performed consistently with rules for security relevant modification all the different TOE representations as TOE implementation representation, TOE design or specification as defined for application code and Final TOE development.

ALC_IAR.4-5 The evaluator shall examine the documentation associated to the change to determine if rules defined in development guidance for additional code and final TOE have been followed, in particular concerning evidences associated to the TOE implementation, TOE implementation representation, and in the TOE representations (if any) as part of ADV_FSP, ADV_TDS and ADV_IMP and ADV_INT families.

ALC_IAR.4.6C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules for TOE functional testing defined for Final TOE development.

ALC_IAR.4-6 The evaluator shall examine the documentation associated to the change to determine if rules defined in development guidance for additional code and final TOE have been followed, in particular concerning evidences of non-regression testing and changes introduced in security test plan, expected test results and test coverage as part of ATE_FUN, ADV_COV and ATE_DPT families.

ALC_IAR.4.7C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules of TOE test coverage defined for Final TOE development.

ALC_IAR.4-7 The evaluator shall determine what part of independent testing requires to be executed as part of ATE_IND families based on evidences produced in TOE functional testing.

ALC_IAR.4.8C The impact analysis procedures documentation shall describe the methods used to demonstrate that testing of change in TOE has been performed consistently with rules of analysis of the TOE depth of testing defined for Final TOE development.

ALC_IAR.4-8 The evaluator shall determine what part of penetration testing requires to be executed as part of AVA_VAN families based on evidences produced in TOE functional testing.

Annex A - Refinement of IAR structure

1. INTRODUCTION
1.1. Document Objectives
1.2. TOE and ST References
2. DESCRIPTION OF THE CHANGES
2.1 Summary of Product Changes
2.1.1 Rationale for Type of change *(based on §7.1)*
2.1.2 How Functional Regression is mitigated? (for IAR.1 to 4)
2.1.3 How Security Regression is mitigated? (for IAR.1 to 4)
2.2. Description of the Issues and Proposal
2.2.N. Issue [Identifier N]: Issue Title N (for each corrected issue) *(see details in §7.3)*
2.3. Environment Changes
2.3.1. Environment of Development
2.3.2. Environment of Production
2.3.3. Environment of Delivery
2.3. Environment Changes
2.3.1. Environment of Development
2.3.2. Environment of Production
2.3.3. Environment of Delivery
2.4. Process Changes
2.4.1. Development Process
2.4.2. Delivery Process
2.5. Recommendation Changes
2.5.1. Delivery Recommendations
2.5.2. Preparation Recommendations
2.5.3. Operation Recommendations
2.6 Modification Management
2.6.1. The Roles
2.6.2. Process of Acceptance for the Modification
3 IMPACT ANALYSIS ON DEVELOPER EVIDENCES
3.1. List of Modified TOE Deliverables *(Based On §2.1.1 and §7.2)*
3.2. List of Unmodified TOE Deliverables *(based on §2.1.1 and §7.2)*
4. DESCRIPTION OF EVIDENCE CHANGES
5. CONCLUSIONS
6. REFERENCES
6.1. External References
6.2. Internal References
7 ANNEX
7.1 Typology of security change
7.2 Strategy of Deliverable update per type of security change
7.3 Detailed structure for Issue description
7.3.1. Issue [Identifier N]: Issue Title N
7.3.1.1 Root Cause
7.3.1.2 Why Issue has not been seen before?
7.3.1.3 Technical Solution
7.3.1.4 Implementation of Solution
7.3.1.5 How to be sure that new implementation solves the issue?
7.3.1.6 How to be sure that no security weakness has been introduced?

Annex B – Self-assessment Form

A self-assessment form template has to be built in accordance of structure of rules and recommendations introduced development guidance for additional code and Final TOE.

When an additional code is developed, it has to follow rules and recommendations introduced development guidance for additional code and Final TOE.

An evidence such rules have been followed is generated by use of self-assessment form template as an output of the development of the additional code.

When rules are not followed, it is mandatory to obtain a waiver from an internal authority based on a precise rationale prior any possible acceptance of such additional code.
In such case, such waiver must be reviewed by ITSEF chosen to perform reassessment.

Rules from AGD_DE	Application [YES / NO/NA]	Location in evidences	Validation [OK/NOK]
Rules 1	YES	IAR.2 (binary) IAR.3 (name of source file) IAR.4 (name of module / TSFIs)	[OK]
Rules 2	NO	Rationale	Waiver [OK]
Rules 3	NA	Rationale	Waiver [OK]
…			

Table 7: Example of self-assessment form already completed and validated

Additional Code development team completes the self-assessment form based on rules defined in Guidance for additional code development.
Validation of self-assessment is done by entity responsible of additional code acceptance.
If rule is not applied or considered as not applicable, a rationale is mandatory to authorize additional code deployment.

Note: For IAR.2 where only TOE implementation is distributed to ITSEF, It will be not possible to the lab that rules are followed at level of TOE implementation representation.

Annex C – Vocabulary

AC	Additional Code
AEP	Asynchronous Evaluation Process
IAR	Impact Analysis Report
KAP	Known Assurance Package
NA	Not Applicable
SAR	Security Assurance Requirement
SAR	Security Assurance Requirement
SEP	Synchronous Evaluation Process
TOE	Target of Evaluation
TSFI	TSF interface