Evaluation paradigm selection according to Common Criteria for an incremental product development

Andreas Daniel Sinnhofer
Institute for Technical Informatics
Graz University of Technology, Austria
a.sinnhofer@tugraz.at

Wolfgang Raschke
Institute for Technical Informatics
Graz University of Technology, Austria
wolfgang.raschke@tugraz.at

Christian Steger
Institute for Technical Informatics
Graz University of Technology, Austria
steger@tugraz.at

Christian Kreiner
Institute for Technical Informatics
Graz University of Technology, Austria
christian.kreiner@tugraz.at

ABSTRACT

Today, agile product development techniques are widely used providing a rapidly and steadily progression of incremental product improvements. Traditionally, a product certification is issued in a late stage of the development process, although some Common Criteria evaluation paradigm would exists to support an agile or modular development process. The usage of such a paradigm would result in a beneficial certification process, since the evaluator gains experience through the maturing product. To provide a systematic way to integrate the evaluation process into the development process — and thus saving money and time — we have identified use case scenarios with the according evaluation paradigm, providing a selection scheme for the right paradigm.

Categories and Subject Descriptors

D.2.13 [**Software Engineering**]: Reusable Software—*Reuse models*

General Terms

Design, Security

Keywords

Common Criteria, Security Evaluation

1. INTRODUCTION

Today, agile product development techniques are widely used providing a rapidly and steadily progression of incremental product improvements, based on common parts and a modular product architecture [7]. This leads principally to a faster time to market and enables the ability to survive and compete in a competitive market. A problem with this flexible and adaptive development paradigm comes up when a certification of the product should be issued, since — traditionally — agile methods are already not used for the development and evaluation process of secure products.

At present, a common approach is to start the certification process of a product in a very late phase of the development, which can result in huge costs when the evaluation facility gives a negative attestation, because a redesigned must be issued. As identified by Boehm [9], the later changes are introduced in the development process, the higher the costs are.

Another problem with such an approach is the long period of time an evaluation process can take, even when the certification of the product is positive. E.g. the certification process of Microsoft Windows 7 took one year and eight months[1]. This can lead to a delayed release if a certificate is a condition for the disposal of a product (e.g. the CE certificate for resale within the EU) or a big gap between the date of release and the date a certificate is issued. Either way, both situations can potentially result in a loss of customers when a competitor is already selling a certified product.

To overcome these drawbacks, Raschke et al. [14] introduced two processes capable for a modular or agile product development, where the certification process is started in parallel. Furthermore he provides a method to automatically detect the actual impact set, so that only those modules are re-evaluated which have an effect to the security assurance of the product. This approach has the key benefit that the evaluator is integrated since the early stages of the process. In fact, the evaluator is gaining experience with the maturing system. Moreover, the feedback of the evaluator can be directly integrated in the next iteration step leading to lower redesign costs [8] [6]. The Common Criteria certification process itself is not further specified, which means that any possible paradigm can be chosen, such as the assurance

[1]see the 14th International Common Criteria Conference (ICCC) `https://www.commoncriteriaportal.org/iccc/ICCC_arc/presentations/T2_D2_2_30pm_Grimm_Evaluating_Windows.pdf`

continuity, a compositional evaluation or a delta evaluation, depending on the current development environment regarding the number of involved developing companies and the number of involved certification facilities.
The contribution of our paper is the identification of the appropriate evaluation scheme for a Common Criteria certification for an agile or modular product development which is applicable in combination with the processes from Raschke et al. [14]. The proposed selection scheme is also applicable for products which are based on previously certified products or modules (e.g. for bug-fix releases).
Section 2 gives a short introduction into the evaluation paradigms according to Common Criteria and the processes identified by Raschke et al. [14]. Section 3 gives an overview over the use case scenarios, providing further information on the according evaluation paradigm and Section 4 summarizes the findings from the use case scenarios in the proposed selection scheme. Finally the results of this paper are summarized and related work is presented.

2. BACKGROUND

2.1 Assurance Continuity

As proposed in Common Criteria Assurance Continuity [1], an evaluation paradigm for the maintenance and re - evaluation of already Common Criteria certified products exists. The flow chart of this approach is illustrated in Fig. 1. It can be seen that based on an impact analysis report (IAR) a decision is made whether the changes to the target of evaluation (TOE) is minor (does not affect the assurance baseline) or major. In the case of minor changes, the previously issued certificate is updated with a maintenance addendum and a maintenance report. The Common Criteria Assurance Continuity [1] states, that *"Maintenance may, in general, continue for up to two years beyond the certification date"*. Due to the fact, that we only consider major changes, the maintenance process is not further contemplated.

In case of major changes, a re-evaluation needs to be performed regarding all affected parts and a new certificate is issued. This can be achieved using an informal modular evaluation scheme (i.e. Delta-Evaluation) through re-evaluation of only the changed and affected modules as stated in the Common Criteria Information Statement on the reuse of evaluation results (see Section 2.2).
The drawback of the assurance continuity approach is, that it is only applicable in those situations, where the evaluation facility is not changed and where a certificate was already issued. Therefore, this approach is intended to be used for bug-fix releases/revisions of old products.

2.2 Delta Evaluation

As stated in the "Common Criteria Information Statement on the reuse of evaluation results" [2] the following evidences must be shared to reuse previously created evidences:

- Product and supporting documentation
- New security target(s)
- Original security target(s)
- Original evaluation technical report(s)

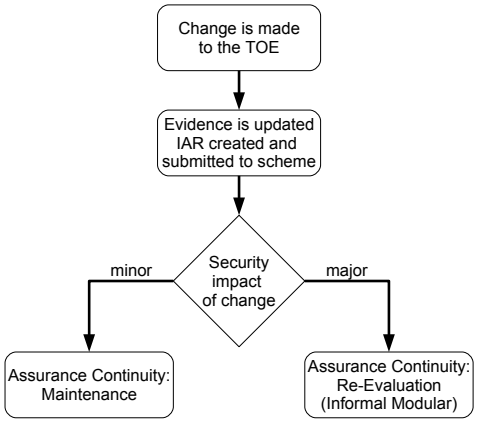

Figure 1: Common Criteria Assurance Continuity flow chart

- Original certification/validation report(s)
- Original Common Criteria certificate(s)
- Original evaluation work packages (if available)

It is specified that

> "*... the evaluation facility conducting the current evaluation should not have to repeat analysis previously conducted where requirements have not changed nor been impacted by changes in other requirements ...*"

where such changes are identified through a so called delta analysis:

> "*... The evaluation facility would be required to perform a delta analysis between the new security target and the original security target(s) to determine the impact of changes on the analysis and evidence from the original evaluation(s) ...*"

which is similar to an impact analysis.
As a result, a product re-evaluation can be performed by an analysis of the impacts of changes and through evaluation of only the changed and affected modules. Unaffected modules need not be reconsidered for the overall evaluation process. Drawback of this approach is that the evaluation technical report is typically generated by the evaluation facility and thus, in some cases, is considered as proprietary to that facility, which makes the interchange of evidences between different certification facilities difficult.

2.3 Composite evaluation

As stated in the Common Criteria Mandatory Technical Document on the composite product evaluation for smart cards and similar devices [3] a composite evaluation can be performed for all kind of products where

> "*... an independently evaluated product is part of a final composite product to be evaluated ...*"

and hence is not limited to smart cards only, but with the limitation that

> "*... The composite product is a product consisting of at least two different parts, whereby one of them represents a single product having already been evaluated and certified ... The underlying platform is the part of the composite product having already been evaluated ...*"

Thus it is applicable for example for an embedded system whereas an application runs on a certified OS, respectively the OS is running on a certified hardware. I. e. a layers pattern is used for the product, whereby trust is established through each layer. The lowest EAL of all components is the limiting factor of the composite product.

2.4 Composed evaluation

As stated in the Common Criteria part 3 (see [4]), the composed evaluation is intended for situations, where independently certified (or going through an independent certification process) products/modules are assembled to a new product which should be certified. It is applicable, where a composite evaluation is not suitable and a delta evaluation cannot be performed due to missing evidences (proprietary documents are not shared). At present, a composed evaluation for higher assurance levels (higher than CAP-C[2] is not supported through the composed scheme and hence a re-evaluation of the whole product is necessary. Due to this, composed evaluations have been performed much less successful than composite evaluations.

2.5 Informal: Identification of the impact set

Due to the fact that it is not necessary to perform unaffected evidences twice, it is meaningful to use change detection analysis to determine the actual affected modules so that only these modules need to be reconsidered in the evaluation. It is important to understand, that modules can interact with each other and hence not only the directly changed module but all other interacting modules need to be reconsidered. This can be achieved through the use of the change impact analysis process proposed by Bohner [10] or the refined processes by Raschke et al. [14]. Our work only mentions the processes proposed by Raschke et al. since he also describes a tool for an automatic change detection analysis, which is well-suited for an partially automatic generation of the Impact Analysis Report (respectively delta analysis), but every other approach is also applicable.
The change detection analysis is based on the so-called *Security Model*, which describes the properties and relationships of the developer evidences, based on the security target, the design documentation, the implementation and the tests (see Figure 2 explanatory graphical representation). Therefore it is applicable to trace and detect all dependencies between each module.

[2]Attack potential "Enhanced Basic"; approximately comparable with EAL-4 (see[4] pages 38 and 47)

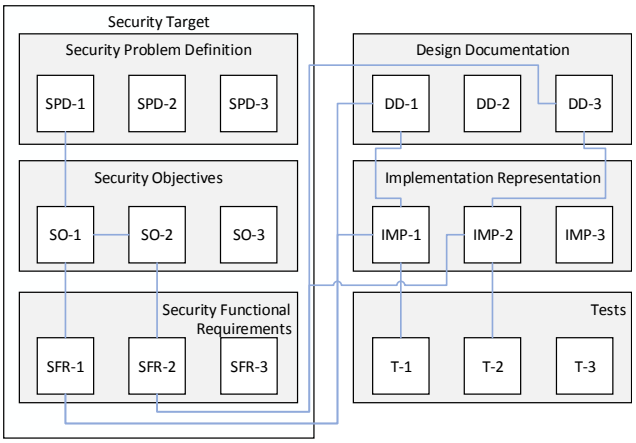

Figure 2: Explanatory Security Model, showing some exemplary artefacts and traces

3. PROPOSED USE CASES AND ACCORDING EVALUATION PARADIGM

Overall situation: Aforementioned, we consider an agile or modular product development process, where the (final) certified product is assembled using a number of modules. In each development iteration new modules can be added or old modules can be changed or removed. Various companies can be involved in the development process of the product and any number of evaluation facilities can be integrated in the certification process.
The selection scheme is applicable for the following scenarios:

- *Use case 1:* One company develops a number of modules which are all evaluated at the same evaluation facility. Since the evaluation facility has full access to all modules and all related evidences, an evaluation can be achieved by a simple informal modular evaluation. If during the development process the evaluation facility is changed, a formal modular paradigm would need to be chosen.

- *Use case 2:* One company develops a number of modules, whereas a number of evaluation facilities ($n > 1$) are involved in the certification process, interchanging all kind of evidences. Therefore, a delta evaluation can be issued.

- *Use case 3:* One company develops a number of modules, whereas a number of evaluation facilities ($n > 1$) are involved in the certification process, but unfortunately they do not interchange evidences. Depending on the architecture of the developed product a composite (Use case 3.a) evaluation or an composed (Use case 3.b) evaluation can be issued.

- *Use case 4:* Several companies are involved in the development process of the product, but one central evaluation facility is used. In this scenario an informal modular evaluation can be used since the certification facility has direct access to every contribution of every

Figure 3: Proposed paradigm selection scheme

company. If during the development process the certification facility is changed, a formal modular scheme would need to be chosen.

- *Use case 5:* Several companies are involved in the development process of the product and any number of evaluation facilities (n > 1) are included into the certification process (e.g. each company consults a different evaluation facility). A delta evaluation is possible if the different evaluation facilities interchange all kind of evidences, which can be a problem since the evaluation facilities would need to provide information on their evaluation process and their used methods. In practice, a composite (Use case 5.a) or composed (Use case 5.b) evaluation scheme is used, depending on the used architecture.

4. PARADIGM SELECTION SCHEME

Based on the activities during the assurance continuity process [1], a selection scheme for the presented use cases was created. The first steps towards the reuse of any evidence is the analysis of the impacts on the assurance of the current Target of Evaluation which is intended to be done by one of the processes proposed by Raschke et al. [14]. The selection scheme is split-up into two main leafs, where one is applicable if a product is developed from a single company and the other one for a product which is developed from many companies. As identified in the use cases, another factor which must be considered is the number of certification facilities and the fact if these certification facilities do interchange all needed evidences so that the evaluation results can be reused efficiently. Another criterion which needs to be reconsidered is derived from the composed evaluation scheme, whereby the developed product is structured in a layered approach. The lowest layer must be already certified.
Generally spoken an informal approach can be used if certification facilities do interchange evidences or a single certification facility is issuing the product evaluation and formal approaches must be chosen in all other cases which are usually more time and money intensive.
The next enumeration provides a short description of the according paradigms:

- *Informal Modular:* The certification facility has full access to all modules and evidences, therefore only the affected modules are re-evaluated (Delta Evaluation).
- *Formal Modular:* The certification facility was changed and hence, all modules need to be reconsidered in the evaluation process. Previously created evidences (e.g. certified modules) can be reused, if all needed information is available.
- *Composed Evaluation:* This evaluation is based on the Composed Assurance Package (CAP) of the Common Criteria part 3 (see Section 2.4). Drawback is that the highest achievable CAP level is CAP-C, which is comparable to EAL-4. Higher levels of assurance are only possible through a complete re-evaluation of the assembled product.
- *Composite Evaluation:* This evaluation paradigm is based on a layered product development, where trust is gained through the combination of all layers. In difference to the composed evaluation, the composite product is the final product for which an EAL level

certification is issued. This allows a direct comparison with similar products certified after a single evaluation. [3]

- *Delta Evaluation:* This is the delta evaluation as described in Section 2.2. A concrete process for the certification is not provided through the Common Criteria standard and thus the according certification facility needs to be consulted.

5. RELATED WORK

Klohs [12] provides observations and thoughts on the modularisation concepts for the development of a smart card operating system according to Common Criteria. He points out that the JIL document [5] on the security architecture requirements for smart cards and similar devices, establishes a first starting point for the reuse of software components, based on a description of the security interface and the implemented security mechanism which is implemented from the component independent of a concrete security target.
The Assert4SOA[3] project focuses on the development of methods for the certification of service oriented architectures (SOAs), reusing existing certification processes to overcome the challenging tasks for an evolving software ecosystem. The project itself does not focus on the Common Criteria scheme, but provides a guidance to integrate the Common Criteria certification scheme into a service oriented architecture in [13].
The Euro-MILS[4] project focuses on providing a framework for trustworthiness by design and high assurance based on *Multiple Independent Levels of Security (MILS)* [11]. In fact, assurance of the whole product is gained through the composition of assurance arguments of its components and the system's security architecture. The developed framework is based on the Common Criteria evaluation schemes.

6. CONCLUSION

Today's industry is embossed through fast changing requirements regarding functional and security needs. These circumstances are tried to be solved through the usage of agile or incremental manufacturing techniques. We have identified a scheme for the selection of the appropriate evaluation paradigm to support an agile or modular development processes regarding the security certification to reduce the time shift between the successful certification and the time the product development finished. Furthermore the costs for re-evaluating the developed product/modules can be kept as low as possible since the most suitable paradigm is chosen, maximizing the reuse of already evaluated modules and providing a direct integration of the evaluation facility in the process so that the feedback is directly integrated into the next development iteration.

7. ACKNOWLEDGEMENT

Project partners are NXP Semiconductor Austria GmbH and the Technical University of Graz. The project is funded by the Austrian Research Promotion Agency (FFG).

[3] www.assert4soa.eu
[4] http://www.euromils.eu

8. REFERENCES

[1] Common Criteria. Assurance Continuity CCRA Requirements. Version 2.1 (June 2012).
[2] Common Criteria Information Statement. Reuse of Evaluation Results and Evidence. (October 2002).
[3] Common Criteria Supporting Document Mandatory Technical Document - Composite product evaluation for Smart Cards and similar devices. Version 1.2 (April 2012).
[4] Common Criteria for Information Technology Security Evaluation. Part 3 Security assurance compnents. Version 3.1 Revision 4 (September 2012).
[5] Common Criteria Supporting Document Guidance - Security Architecture requirements (ADV_ARC) for smart cards and similar devices. Version 2.0 (April 2012).
[6] S. Ambler. *The Object Primer: Agile Model-Driven Development with UML 2.0 - Third Edition.* Cambridge University Press, 2004.
[7] D. Anderson. *Agile Product Development for Mass Customization: How to Develop and Deliver Products for Mass Customization, Niche Markets, Jit, Build-To-Order and Flexible Manufacturing.* Irwin Professional Pub., 1997.
[8] K. Beck and C. Andres. *Extreme Programming Explained: Embrace Change (2Nd Edition).* Addison-Wesley Professional, 2004.
[9] B. W. Boehm. *Software Engineering Economics.* Prentice Hall, Englewood Cliffs, NJ, 1981.
[10] S. A. Bohner. Extending software change impact analysis into cots components. In *Proceedings of the 27th Annual NASA Goddard Software Engineering Workshop (SEW-27'02)*, SEW '02, pages 175–, Washington, DC, USA, 2002. IEEE Computer Society.
[11] H. Blasum, S. Tverdyshev, B. Langenstein, J. Maebe, B. De Sutter, B. Leconte, B. Triquet, K. Müller, M. Paulitsch, A. Söding- Freiherr von Blomberg, A. Tillequin. Secure European Virtualisation for Trustworthy Applications in Critical Domains - MILS Architecture, 2014.
[12] D. K. Klohs. Software modularisation and the common criteria - a smartcard developer's perspective.
[13] M. B. Samuel Paul Kaluvuri and Y. Roudier. Bringing common criteria certification to web services.
[14] W. Raschke, M. Zilli, P. Baumgartner, J. Loinig, C. Steger and C. Kreiner. Supporting evolving security models for an agile security evaluation, 2014.