Taming the complexity of the CC

10th ICCC, Tromsø
First stop

The Common Criteria

The member organisations of the CCRA declare that defined assurance levels (EALs) between versions of the criteria are equivalent and can therefore be used without restrictions for composition activities.

CC v3.1, Release 3

Click here for information about the CC/CEM maintenance process.

CC v3.1 consists of three parts, make sure to download and use these files marked as 'Final':

- **Part 1**: Introduction and general model
  - PDF: [CCPART1V3.1R3.pdf](#)
  - XML: [CC3R3.dtd](#)

- **Part 2**: Security functional requirements
  - PDF: [CCPART2V3.1R3.pdf](#)
  - XML: [cc3R3.XML.zip](#)
  - Marked changes: [CCPART2V3.1R3 - marked_changes.pdf](#)

- **Part 3**: Security assurance requirements
  - PDF: [CCPART3V3.1R3.pdf](#)
  - Marked changes: [CCPART3V3.1R3 - marked_changes.pdf](#)

CEM v3.1 consists of one part:

- **CEM**
  - PDF: [CEMV3.1R3.pdf](#)
  - Marked changes: [CEMV3.1R3 - marked_changes.pdf](#)

See the new guides:

- Guide for the Transition from CC v2.3 to CC v3.1 for ADV requirements
- Transition Guide Summary of changes in the CC/CEM from v2.3 to v3.4
- Guidelines for Developer Documentation according to Common Criteria Version 3.1

CC v2.3

CC v2.3 is based on version 3.2, updated with a number of interpretations and further editorial changes.
The CC consists of +-1072 pages

Printed +-5kg of paper

Reaches observation deck of the Empire State building (+-318m)
The complexity is real

- Complexity in the process
  - Requirements setting
  - Inevitable changes

- Complexity in standard

- Coping strategies
  - For developer
  - For evaluator
  - For certifier
  - For end-user
Perceived security problem

Perceived assurance sought

Threats

Objectives

SFRs

Environment

SARs

PP

ST
Perceived security problem

Perceived assurance sought

"Risk analysis"

Common Criteria
Perceived security problem
Perceived assurance sought

Threats → Objectives → SFRs
Environment → SARs

PP

ST

“Risk analysis”

Common Criteria
Perceived security problem

Perceived assurance sought

Threats → Objectives → SFRs → Environment → SARs

PP

ST

“Risk analysis”

Common Criteria
Requirements setting phase: coping mechanisms

- Ignore need for alignment PP with perceived problem
  + no difficult discussions
  - evaluations against PP don’t address perceived problem

- Ignore need for clear and minimal requirements
  + no difficult discussions
  - more expensive evaluations, exclusion of products

- Create discussion forum and go through the process
  + alignment and consensus
  - hard, can lead to “wrong” end result
Process: coping mechanisms

- Consider significant process as unmanageable
  + no more worries
  - no schedule and cost control

- Greatest risk first
  + time to fix before critical path
  - dependencies complex*

- Address imprecise items immediately when they happen
  + avoid multiplier of fixing later
  - hard to be precise on the right items
Process: points of attention

- Be precise in the ST
  - TOE Scope: what is in, what is out?
  - TOE version and identification method
  - SFRs, and how to interpret them
  - Anything with “special” assurance methodology

- Understanding of the FSP and design

- Version management

Iterations have very high time costs
Biggest tool developer

- Risk reduction
- Predictable

- Drifts away from reality
- Duplication of work
- High cost for innovations
Design documentation

- Synchronization design documentation and code difficult

- Generation documentation from code
  
  Examples: Doxygen, JavaDoc
  
  + computer does tedious duplication
  
  - code must be close to design and SFRs
  
  Works well with libraries in C++/Java

- Generate code from documentation
  
  Examples: visual IDEs, Rational tools
  
  + code in synch with design
  
  - changes in code must feed back in documentation
  
  Works with rather strict top-down development
Version management

- Manual version management
  + already in place
  - human mistakes are difficult for evaluator

- Automated version management systems: code
  Examples: VSS, CVS, Subversion, etc
  + no mistakes, actually useful for programmers
  - ?

- Automated version management systems: documents
  Examples: Documentum, Sharepoint, KnowledgeTree, etc
  + no mistakes
  - integration Word difficult
Complexity of the standard

(perspective of the evaluator)
Dependencies

Hard to visualize:

- ~150 work units
- ~450 explicit links
- Fan-in and fan-out irregular
Complex
Which work units refer most and which are referred to?
Which work units refer most and which are referred to?
Understanding of the TOE design central
Testing depends on a lot (and is expensive)
Evaluator coping mechanisms

- Standardization of evaluation work
  - Re-use of approved reports
  - Formalize evaluation activities internally

- Avoidance of roundtrips to CB
  - Belt-and-suspenders analysis
  - Extra work cheaper than discussion
Certifier coping mechanisms

- Keeping assurance comparable is difficult
  - Avoidance of alternative assurance methods
  - Keeping close to literal CEM interpretation safe
End-user coping mechanisms

Common Criteria = ✔
Conclusion
I hope you are not dog-tired now and have some questions?
Wouter Slegers

Creative Solutions

Zwanendreef 2
2665 EM Bleiswijk
The Netherlands
+31-6-24721971
wouter@yourcreativesolutions.nl
Attributions

Common Criteria

“Empire State Building” by Matti Mattila, CC-BY @ flickr.com

Chris Satchwell, CC-BY @ flickr.com, http://www.chrissatchwell.com/

“An old book with a looking glass” by Est smiltis no ausim, CC-BY

Dimitri van Heesch, used with permission

wikimedia.org   EMVco, used as illustration

All other images © Wouter Slegers