NTNU


# Evaluation - the Main Road to IT Security Assurance

CC Part 3

NTNU

# Assurance definition


**Asssurance**

that the claimed security measures of the TOE

are

**effective**

and

**implemented correctly**

is derived from knowledge about the

**- definition**

**- construction**

**- operation**

of the TOE

NTNU

# Measuring Assurance


by: Active investigation

of the: TOE

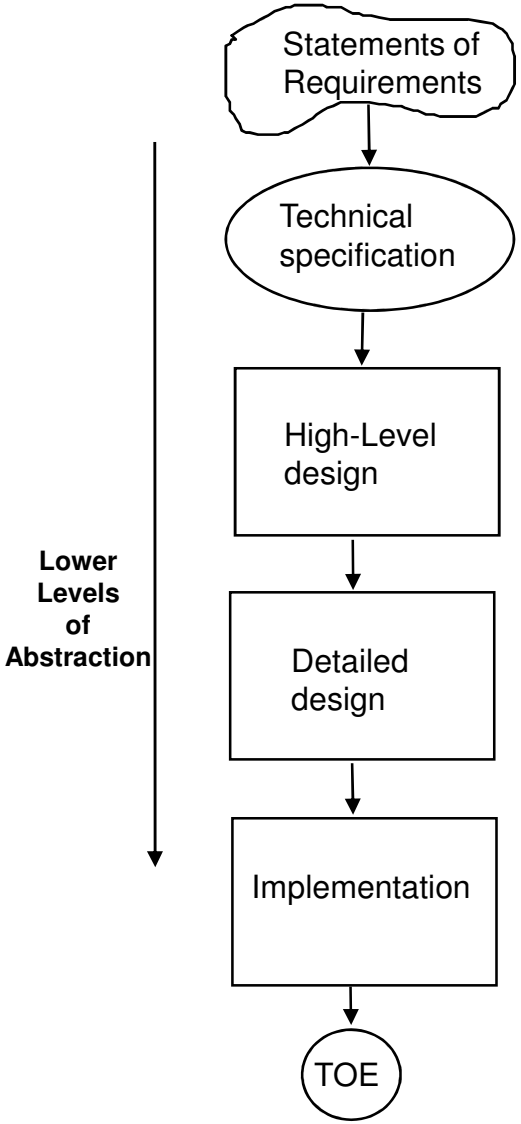
by: Expert evaluators

with increasing emphasis on:

- scope
- depth
- rigour

NTNU

# Assurance Structure




## Each Assurance Component Consists of:

### Developer Actions (.D)

Activities to be performed by the developer - shall use, shall provide

### Content and Presentation of Evidence (.C)

Evidence required for evaluation, what the evidence must demonstrate, and what information the evidence must convey - include, identify, describe, show, demonstrate

### Evaluator Actions (.E)

Analysis implied by the evidence provided, and by the targeted level of assurance - confirm, determine

NTNU

# Vulnerabilities


Vulnerabilities can arise through failures in:

- requirements -- that is, an IT product may possess all the functions and features required of it and still contain vulnerabilities that render it unsuitable or ineffective with respect to security
- development -- that is, an IT product does not meet its specifications and/or vulnerabilities have been introduced as a result of poor development standards or incorrect design choices
- operation -- that is, an IT product has been constructed correctly to a correct specification but vulnerabilities have been introduced as a result of inadequate controls upon the operation

NTNU

# Handling vulnerabilities


Vulnerabilities should be:

- eliminated -- that is, active steps should be taken to expose all exercisable vulnerabilities and remove or neutralise them
- minimised -- that is, active steps should be taken to reduce the potential impact of any exercise of a vulnerability to an acceptable residual level
- monitored -- that is, active steps should be taken to ensure that any attempt to exercise a residual vulnerability will be detected so that steps can be taken to limit the damage

NTNU

# Organising the requirements


**Class** - share a common intent
different coverage of security objectives

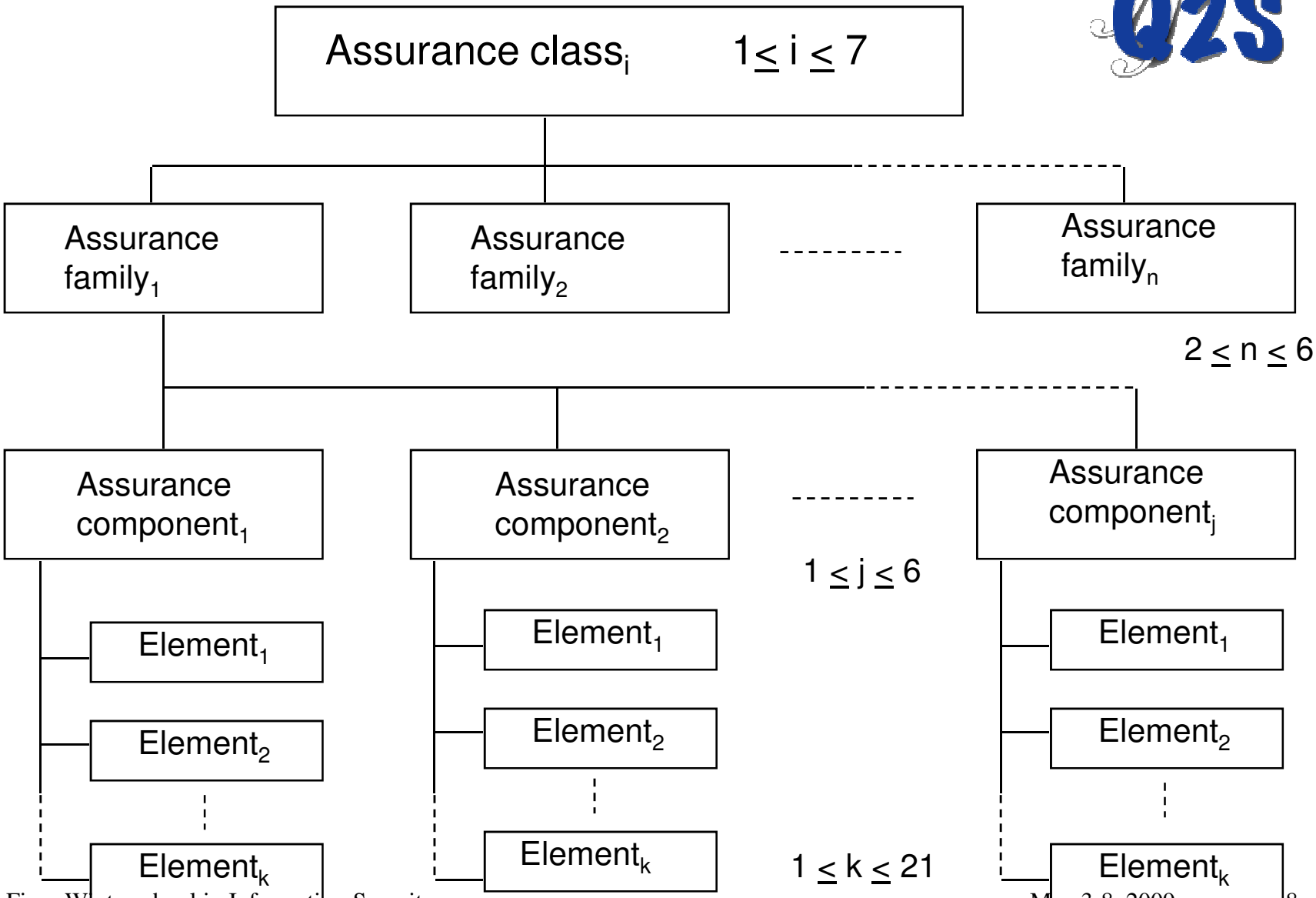
**Family** - share security objectives
different in emphasis or rigour

**Component** - addresses a set of security requirements

**Element** - indivisible security building blocks

NTNU

# Class hierarchy



NTNU

# EALs, Ver. 3.1


| Assurance class | Assurance Family | Assurance Components by Evaluation Assurance Level | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | EAL1 | EAL2 | EAL3 | EAL4 | EAL5 | EAL6 | EAL7 |
| Development | ADV_ARC | | **1** | 1 | 1 | 1 | 1 | 1 |
| | ADV_FSP | **1** | **2** | **3** | **4** | **5** | 5 | **6** |
| | ADV_IMP | | | | **1** | 1 | **2** | 2 |
| | ADV_INT | | | | | **2** | **3** | 3 |
| | ADV_SPM | | | | | | **1** | 1 |
| | ADV_TDS | | **1** | **2** | **3** | **4** | **5** | **6** |
| Guidance documents | AGD_OPE | **1** | 1 | 1 | 1 | 1 | 1 | 1 |
| | AGD_PRE | **1** | 1 | 1 | 1 | 1 | 1 | 1 |
| Life-cycle support | ALC_CMC | **1** | **2** | **3** | **4** | 4 | **5** | 5 |
| | ALC_CMS | **1** | **2** | **3** | **4** | **5** | 5 | 5 |
| | ALC_DEL | | **1** | 1 | 1 | 1 | 1 | 1 |
| | ALC_DVS | | | **1** | 1 | 1 | **2** | 2 |
| | ALC_FLR | | | | | | | |
| | ALC_LCD | | | **1** | 1 | 1 | 1 | **2** |
| | ALC_TAT | | | | **1** | **2** | **3** | 3 |
| Security Target evaluation | ASE_CCL | **1** | 1 | 1 | 1 | 1 | 1 | 1 |
| | ASE_ECD | **1** | 1 | 1 | 1 | 1 | 1 | 1 |
| | ASE_INT | **1** | 1 | 1 | 1 | 1 | 1 | 1 |
| | ASE_OBJ | **1** | **2** | 2 | 2 | 2 | 2 | 2 |
| | ASE_REQ | **1** | **2** | 2 | 2 | 2 | 2 | 2 |
| | ASE_SPD | | **1** | 1 | 1 | 1 | 1 | 1 |
| | ASE_TSS | **1** | 1 | 1 | 1 | 1 | 1 | 1 |
| Tests | ATE_COV | | **1** | **2** | 2 | 2 | **3** | 3 |
| | ATE_DPT | | | **1** | **2** | **3** | 3 | **4** |
| | ATE_FUN | | **1** | 1 | 1 | 1 | **2** | 2 |
| | ATE_IND | **1** | **2** | 2 | 2 | 2 | 2 | **3** |
| Vulnerability assessment | AVA_VAN | **1** | **2** | 2 | **3** | **4** | **5** | 5 |

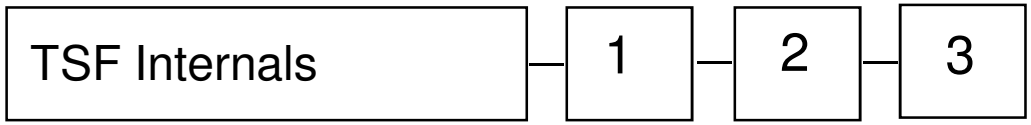
NTNU

# Assurance class ADV


| | | | | | | |
|---|---|---|---|---|---|---|
| ADV_ARC: Security Architecture | 1 | | | | | |
| ADV_FSP: Functional specification | 1 | 2 | 3 | 4 | 5 | 6 |
| ADV_IMP: Implementation representation | 1 | 2 | | | | |
| ADV_INT: TSF internals | 1 | 2 | 3 | | | |
| ADV_SPM: Security policy modelling | 1 | | | | | |
| ADV_TDS: TOE design | 1 | 2 | 3 | 4 | 5 | 6 |

NTNU

# Assurance Family ADV_INT

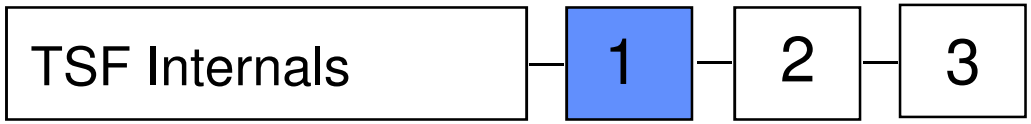




- deals with the internal structure of the TSF

| | | |
|---|---|---|
| Objectives | - | modular construction, layering of software, minimization of circular dependencies, minimization of non-TSP enforcing software |
| Component Levelling | - | based on the amount of structure and minimization required |
| Application Notes | - | “portions of the TSF”, interfaces, sub-systems, modules implementation units |

NTNU

# Assurance Family ADV_INT




## ADV_INT.1 Modularity

Dependencies:

**ADV_IMP.1 Subset of the implementation of the TSF**
**ADV_LLD.1 Descriptive low-level design**

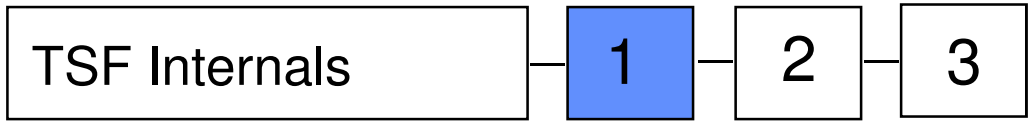
Developer Action Elements:

**1.1.D The developer shall the design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design**

**1.2.D The developer shall provide an architectural description**

NTNU

# Assurance Family ADV_INT




## ADV_INT.1 Modularity

Content and presentation of evidence:

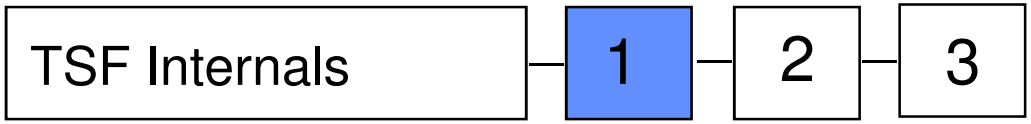
**1.1.C The architectural description shall identify the modules of the TSF**

**1.2.C The architectural description shall describe the purpose, interface, parameters and effects of each module of the TSF**

**1.3.C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions**

NTNU

# Assurance Family ADV_INT




## ADV_INT.1 Modularity

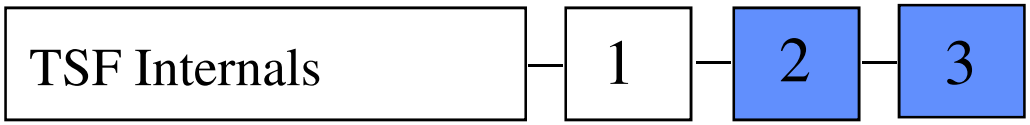
Evaluator actions:

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

**1.2.E The evaluator shall determine the both the low-level design and the implementation representation are in compliance with the architectural description**

NTNU

# Assurance Family ADV_INT


ADV_INT.2 Reduction of complexity

ADV_INT.3 Minimisation of complexity

NTNU

# Assurance Levels


- EAL1 - Functionally tested
- EAL2 - Structurally tested
- EAL3 - Methodically tested and checked
- EAL4 - Methodically designed, tested, and reviewed
- EAL5 - Semiformally designed and tested
- EAL6 - Semiformally verified design and tested
- EAL7 - Formally verified design and tested