Software Testing

L02. Software Quality Assurance (SQA)

Christian Millán

Autumn 2020

2.1. Intro SQA

Differences between software and other industrial products

2.1. Intro SQA

Characteristics	Software product	Other industrial products
Complexity	Usually, very complex product allowing for very **large number of operational options**	Degree of complexity much lower, allowing at most a **few** thousand **operational options**
Visibility of product	**Invisible product**, impossible to detect defects or omissions by sight (e.g. of a diskette or CD storing the software)	**Visible product**, allowing effective detection of defects by sight
Nature of development and production process	Opportunities to **detect defects** arise in only **one phase**, namely product development	Opportunities to **detect defects** arise in **all phases of development and production**

2.2. Software quality

IEEE Definition

- The degree to which a system, component, or process meets specified requirements.
- The degree to which a system, component, or process meets customer or user needs or expectations.

2.2. Software quality

Pressman's definition

- Software quality is defined as:
  Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

2.3. QA & QC

- **Quality assurance:** The planned and systematic activities implemented in a quality system so that quality requirements for a product or service will be fulfilled is known as *quality assurance.*

- **Quality control:** The observation techniques and activities used to fulfill requirements for quality is known as *quality control.*

2.3. QA & QC

Quality Assurance (QA)	Quality Control (QC)
1. It is process related.	1. It is product related.
2. It focuses on the process used to develop a product.	2. It focuses on testing of a product developed or a product under development.
3. It involves the quality of the processes.	3. It involves the quality of the products.
4. It is a preventive control.	4. It is a detective control.
5. Allegiance is to development.	5. Allegiance is not to development

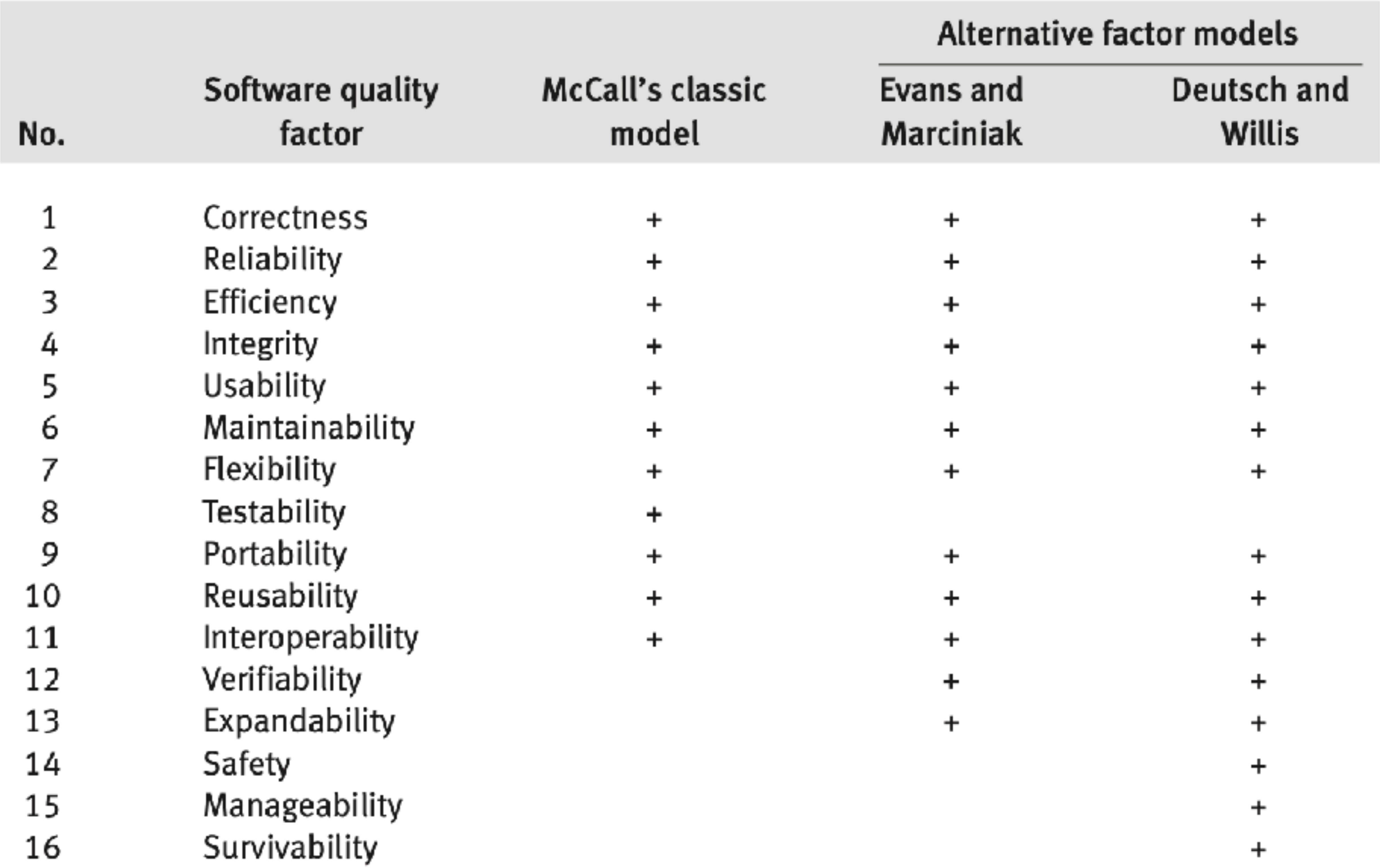
2.4. Verification & Validation

“a systems engineering process employing a rigorous methodology for evaluating the **correctness** and **quality** of software product through the software life cycle.”

2.4. Verification & Validation

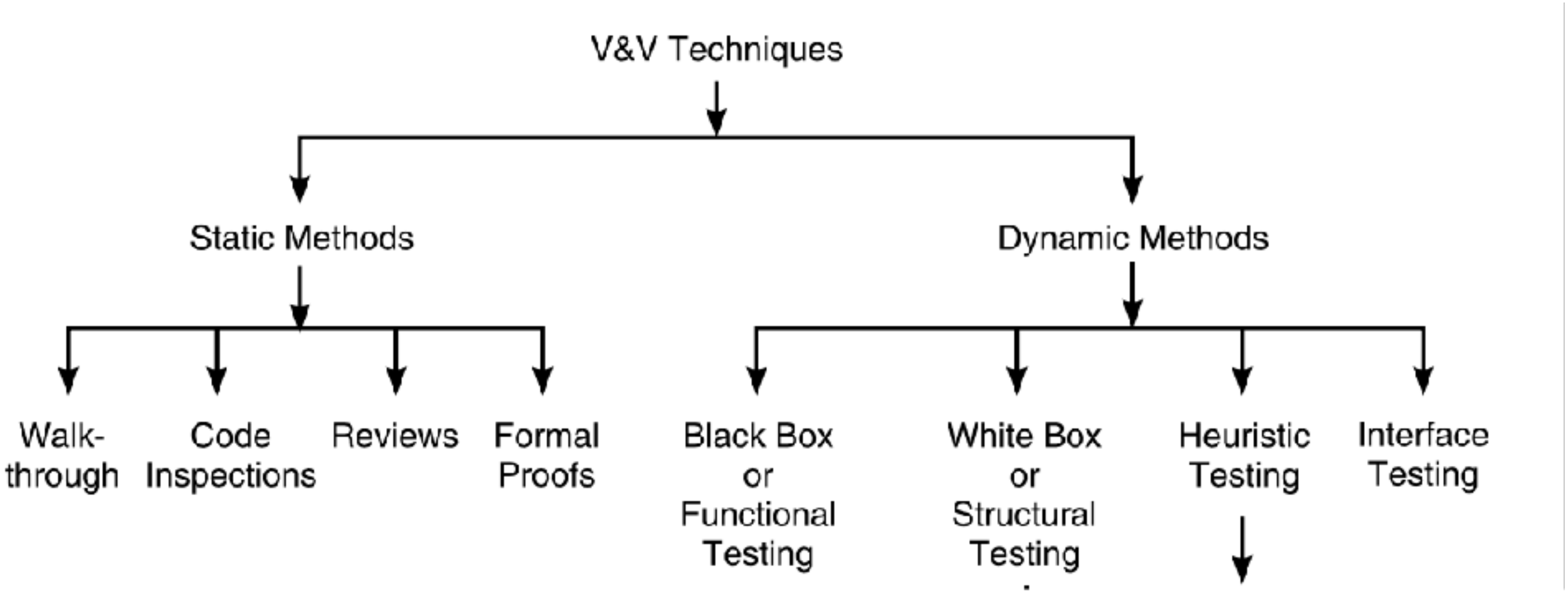
Verification	Validation
1. It is a static process of verifying documents, design, and code.	1. It is a dynamic process of validating/ testing the actual product.
2. It does not involve executing the code.	2. It involves executing the code.
3. It is human based checking of documents/ files.	3. It is the computer-based execution of program.
4. Target is requirements specification, application architecture, high level and detailed design, and database design.	4. Target is actual product—a unit, a module, a set of integrated modules, and the final product.
5. It uses methods like inspections, walk throughs, desk-checking, etc.	5. It uses methods like black-box, gray-box, and white-box testing.
6. It, generally, comes first—before validation.	6. It generally follows verification.
7. It answers the question—**Are we building the product right?**	7. It answers the question—**Are we building the right product?**
8. It can catch errors that validation cannot catch.	8. It can catch errors that verification cannot catch.

2.5. SQA



			Alternative factor models	
No.	Software quality factor	McCall's classic model	Evans and Marciniak	Deutsch and Willis
1	Correctness	+	+	+
2	Reliability	+	+	+
3	Efficiency	+	+	+
4	Integrity	+	+	+
5	Usability	+	+	+
6	Maintainability	+	+	+
7	Flexibility	+	+	+
8	Testability	+		
9	Portability	+	+	+
10	Reusability	+	+	+
11	Interoperability	+	+	+
12	Verifiability		+	+
13	Expandability		+	+
14	Safety			+
15	Manageability			+
16	Survivability			+

2.6. Static & Dynamic methods



V&V Techniques

Static Methods

Dynamic Methods

Walk-through

Code Inspections

Reviews

Formal Proofs

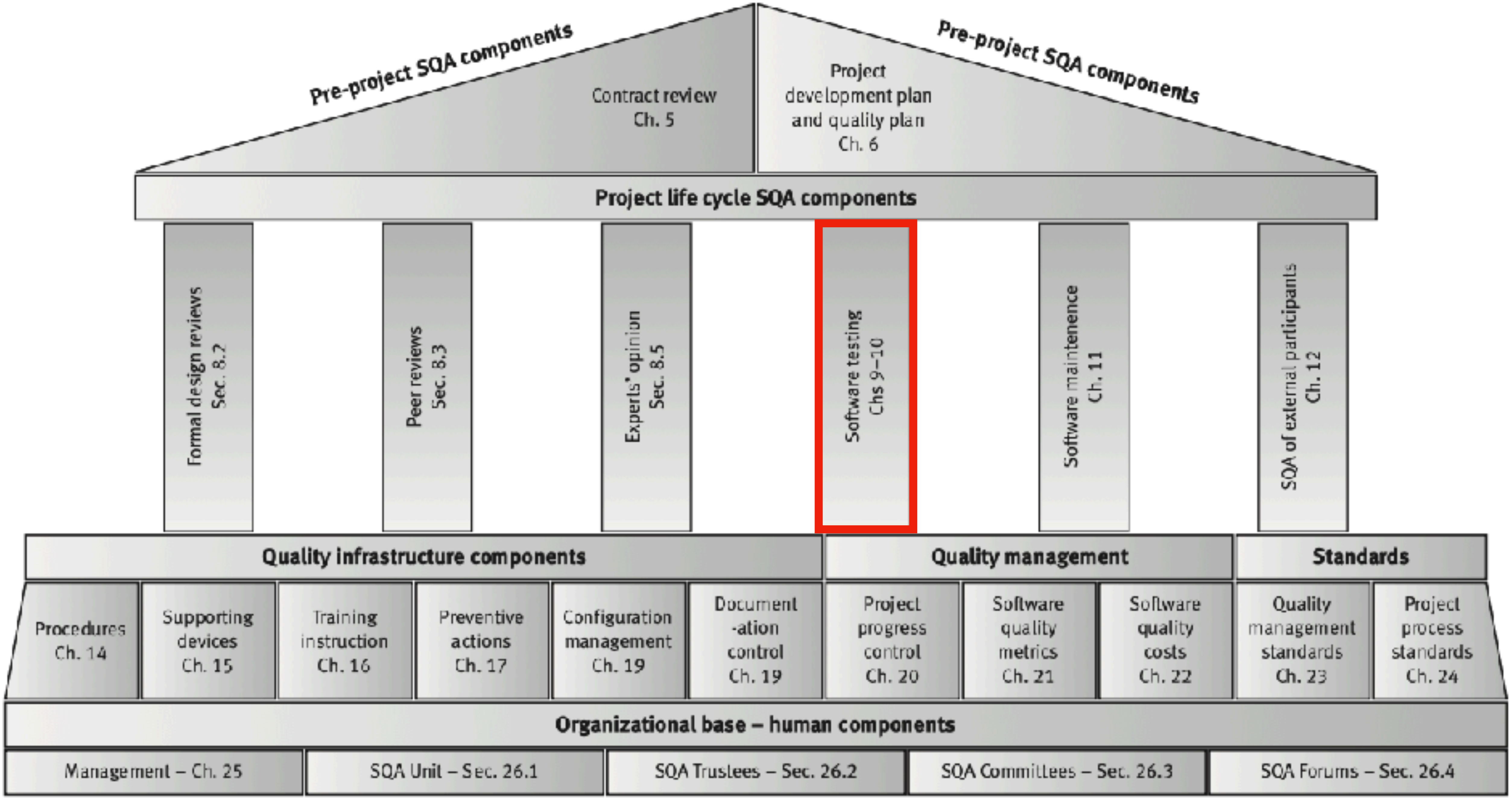
Black Box or Functional Testing

White Box or Structural Testing

Heuristic Testing

Interface Testing

2.7. Components of the SQA system



Pre-project SQA components

Contract review
Ch. 5

Pre-project SQA components

Project
development plan
and quality plan
Ch. 6

Project life cycle SQA components

Formal design reviews
Sec. 8.2

Peer reviews
Sec. 8.3

Experts' opinion
Sec. 8.5

Software testing
Chs 9–10

Software maintenence
Ch. 11

SQA of external participants
Ch. 12

Quality infrastructure components

Procedures
Ch. 14

Supporting
devices
Ch. 15

Training
instruction
Ch. 16

Preventive
actions
Ch. 17

Configuration
management
Ch. 19

Document
-ation
control
Ch. 19

Quality management

Project
progress
control
Ch. 20

Software
quality
metrics
Ch. 21

Software
quality
costs
Ch. 22

Standards

Quality
management
standards
Ch. 23

Project
process
standards
Ch. 24

Organizational base – human components

Management – Ch. 25

SQA Unit – Sec. 26.1

SQA Trustees – Sec. 26.2

SQA Committees – Sec. 26.3

SQA Forums – Sec. 26.4

2.8. Development and quality plans

Elements of a software quality plan

1. List of quality goals
2. Review activities
3. Software tests
4. Acceptance tests for software externally developed
5. Configuration management tools and procedures

2.8. Development and quality plans

Recommended elements of development and quality plans for **small projects**

The development plan:

1. Project products, indicating “deliverables”
2. Project benchmarks
3. Development risks
4. Estimates of project costs

The quality plan:

1. Quality goals