

SOFTWARE QUALITY

CPTS 583

Quality Models and CMMI

Outline

- The McCall model
 - Hierarchy: factor, criteria, metrics
 - Contrast to the Boehm's model
- The ISO/IEC 9126 model
 - Characteristics and subcharacteristics
- Business/application-specific models
 - FURPS
 - GQM
- Capability Maturity Model (CMM)
 - Quality characteristics of various maturity levels

Software quality model



- A model with the objective to **describe**, **assess** and/or **predict** software quality.
- **Independent** of specific modeling techniques

WHY SHOULD
WE CARE?

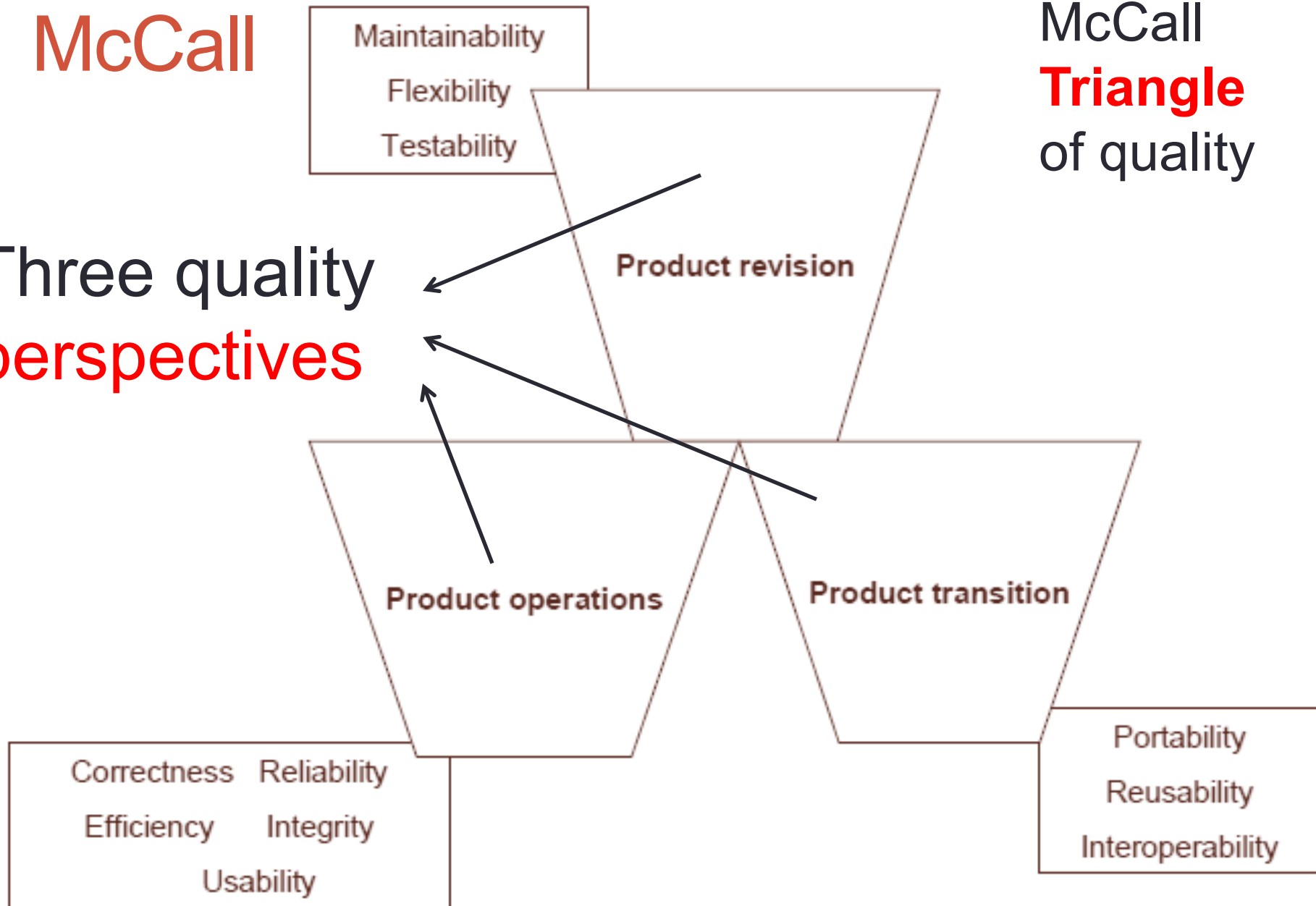
- Express quality **requirements** that customers agree with
- Form as the basis of software **design**
- Provides coding standards/guidelines for **implementation**
- Facilitate software **testing & validation**
- Serve communication/training purposes

Software process

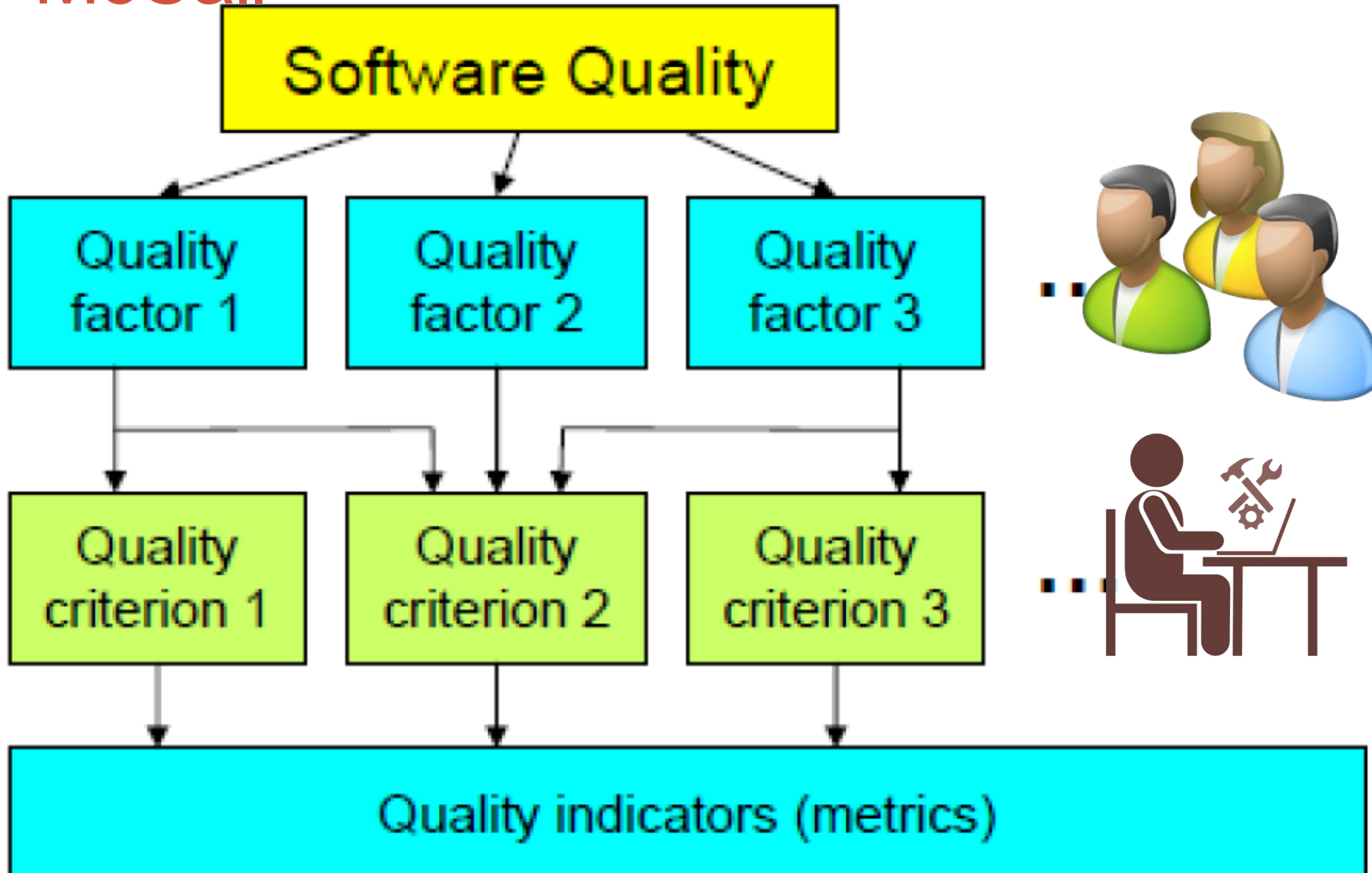
McCall

McCall **Triangle** of quality

Three quality perspectives



McCall



McCall

- Factors-Criterion-Metrics model

- Factors (11)

Behavioral characteristics
of software system

- Users **specify** quality requirements
- Users describe the **external** view of software quality

- Criterion (23)

Attribute of quality factor related
to software design and production

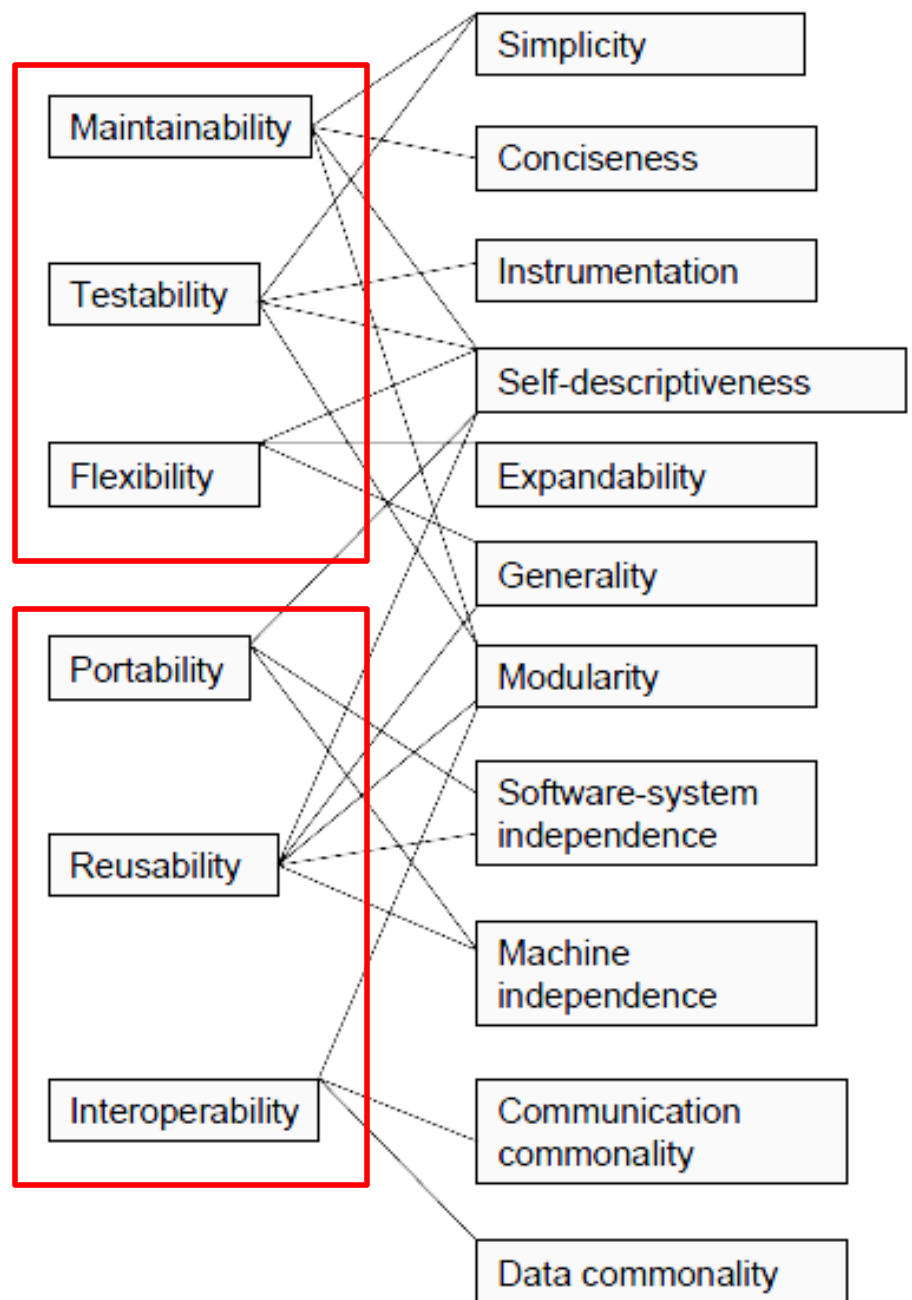
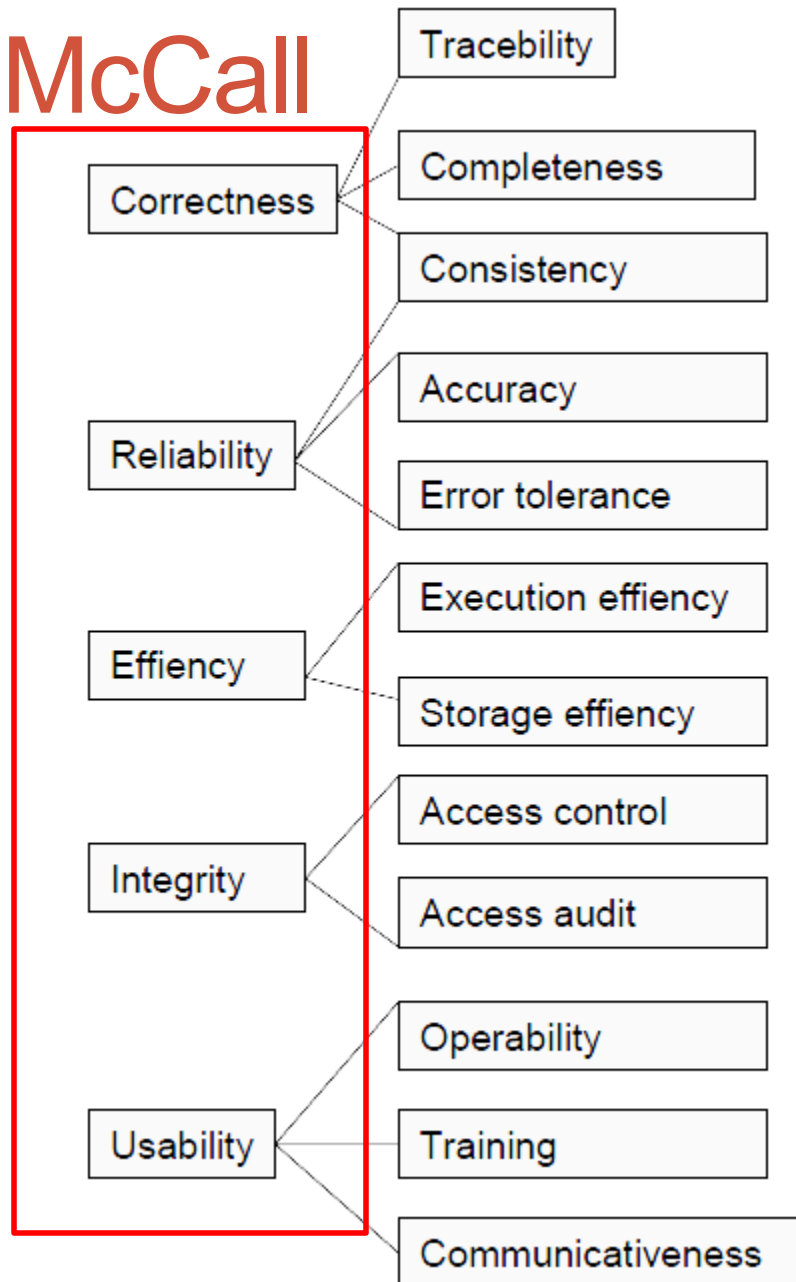
- Developers **build** software that meets the quality requirements
- Developers describe the **internal** view of software quality

- Metrics

Measure that captures some
aspects of a quality criterion

- Used to **control** software quality
- Defined to provide a scale and method for **measurement**

McCall



McCall

- Quality metrics
 - Binary questions
 - Subjective

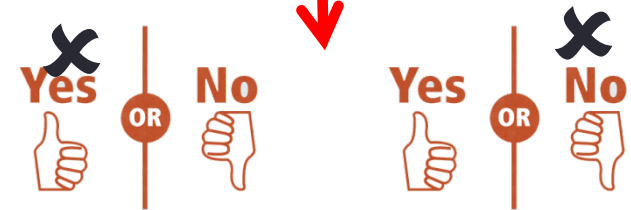
storage efficiency

Does the system consumption of 100GB disk space seem efficient to you?

Question on Quality criterion

Answer collection

Quality metric value



50%

Boehm's model

Level I characteristics

General Utility

As-is Utility

Maintainability

Level II characteristics

Portability

Reliability

Efficiency

Human Engineering

Testability

Understandability

Modifiability

Level III characteristics

Device Independence

Self Containedness

Accuracy

Completeness

Robustness/Integrity

Consistency

Accountability

Device Efficiency

Acessibility

Communicativeness

Self Descriptiveness

Structuredness

Conciseness

Legibility

Augmentability

<i>Criteria/goals</i>	<i>McCall, 1977</i>	<i>Boehm, 1978</i>
Correctness	*	*
Reliability	*	*
Integrity	*	*
Usability	*	*
Efficiency	*	*
Maintainability	*	*
Testability	*	
Interoperability	*	
Flexibility	*	*
Reusability	*	*
Portability	*	*
Clarity		*
Modifiability		*
Documentation		*
Resilience		*
Understandability		*
Validity		*
Functionality		
Generality		*
Economy		*

Boehm's versus McCall

Boehm focuses
on
maintainability

McCall focuses
on 'as-is' utility

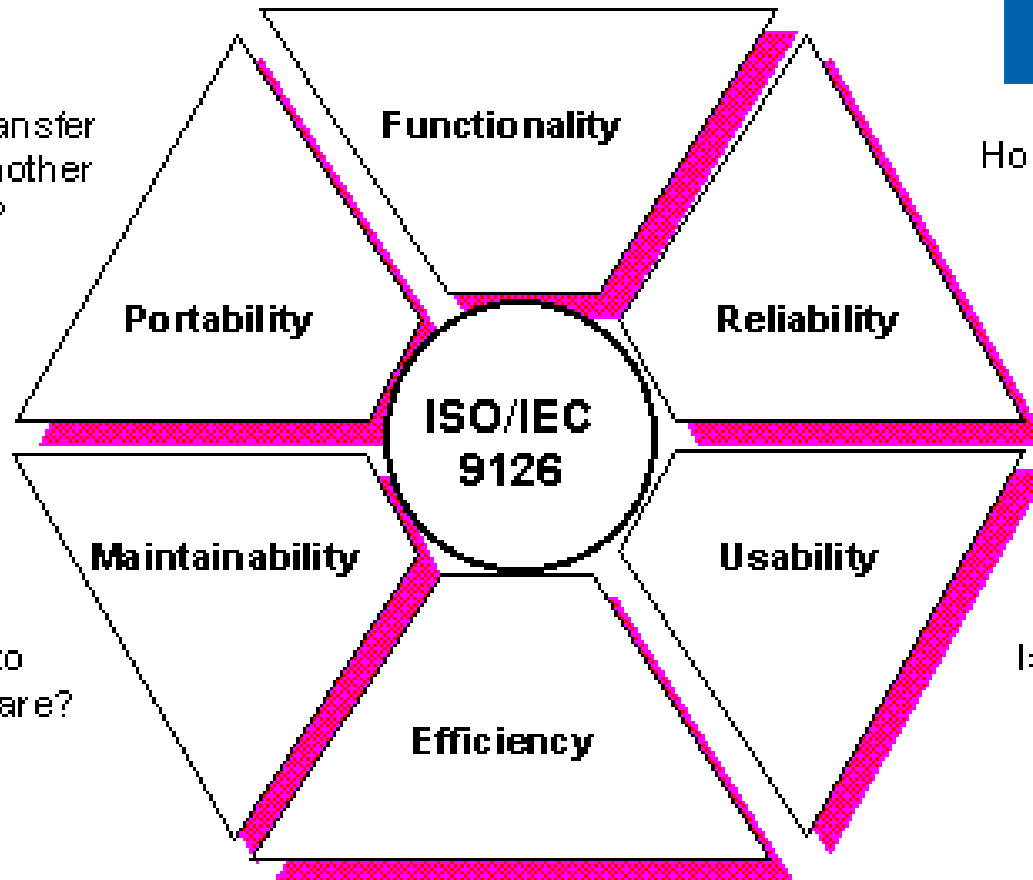
ISO/IEC 9126

Are the required functions available in the software?



How easy is to transfer the software to another environment?

How reliable is the software?

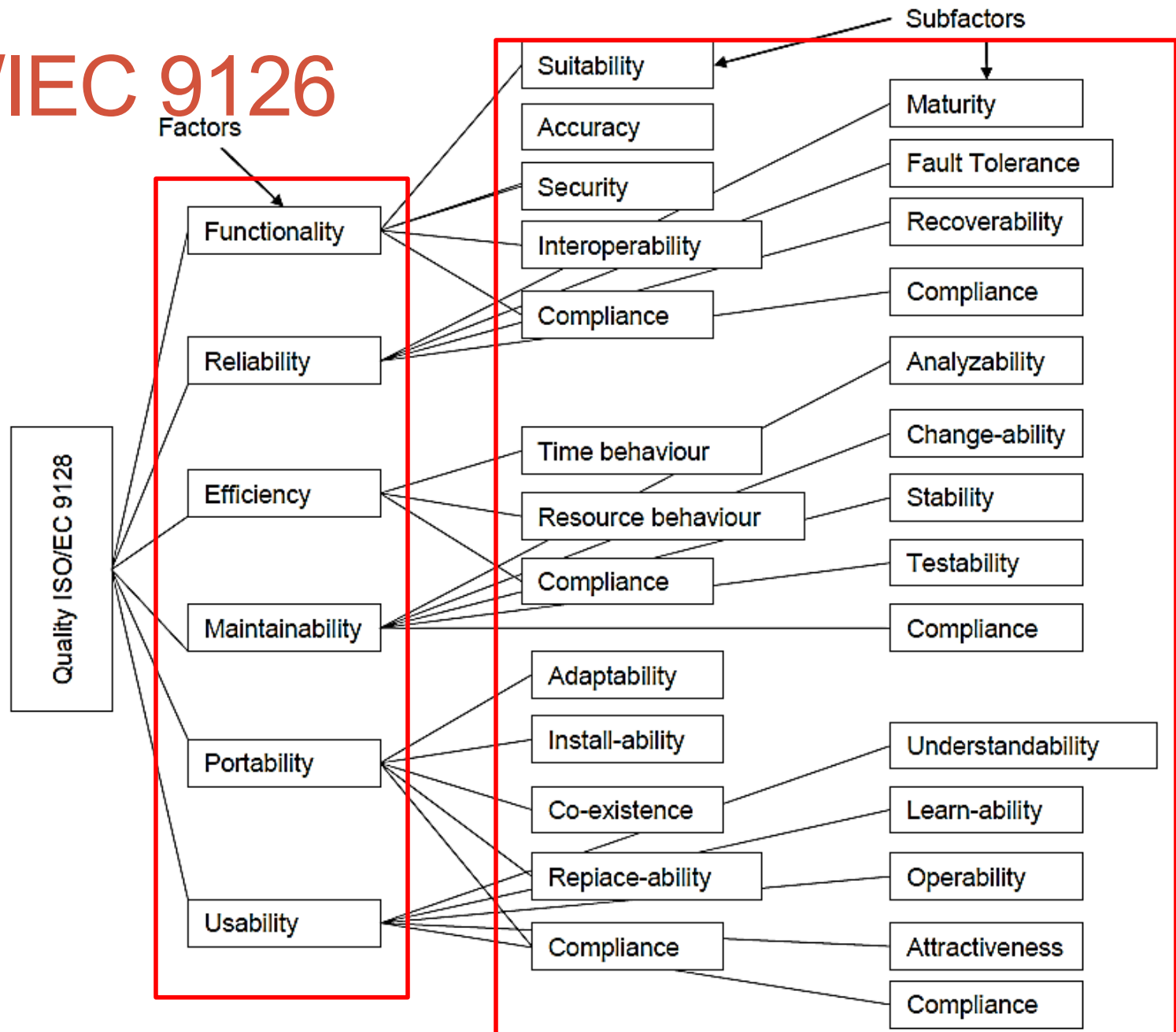


How easy is to modify the software?

Is the software easy to use?

How efficient is the software?

ISO/IEC 9126



ISO/IEC 9126

Characteristics	Subcharacteristics	Definitions
Functionality	Suitability	Attributes of software that bear on the presence and appropriateness of a set of functions for specified tasks.
	Accurateness	Attributes of software that bear on the provision of right or agreed results or effects.
	Interoperability	Attributes of software that bear on its ability to interact with specified systems.
	Compliance	Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions.
	Security	Attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs or data.
Reliability	Maturity	Attributes of software that bear on the frequency of failure by faults in the software.
	Fault tolerance	Attributes of software that bear on its ability to maintain a specified level of performance in case of software faults or of infringement of its specified interface.
	Recoverability	Attributes of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.
Usability	Understandability	Attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.
	Learnability	Attributes of software that bear on the users' effort for learning its application.
	Operability	Attributes of software that bear on the users' effort for operation and operation control.

ISO/IEC 9126

Characteristics	Subcharacteristics	Definitions
Efficiency	Time behaviour	Attributes of software that bear on response and processing times and on throughput rates in performances its function.
	Resource behavior	Attributes of software that bear on the amount of resource used and the duration of such use in performing its function.
Maintainability	Analyzability	Attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.
	Changeability	Attributes of software that bear on the effort needed for modification, fault removal or for environmental change.
	Stability	Attributes of software that bear on the risk of unexpected effect of modifications.
	Testability	Attributes of software that bear on the effort needed for validating the modified software.
Portability	Adaptability	Attributes of software that bear on the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered.
	Installability	Attributes of software that bear on the effort needed to install the software in a specified environment.
	Conformance	Attributes of software that make the software adhere to standards or conventions relating to portability.
	Replaceability	Attributes of software that bear on opportunity and effort using it in the place of specified other software in the environment of that software.

<i>Criteria/goals</i>	<i>McCall, 1977</i>	<i>Boehm, 1978</i>	<i>ISO 9126, 1993</i>
✓ Correctness	*	*	maintainability
Reliability	*	*	*
Integrity	*	*	
Usability	*	*	*
Efficiency	*	*	*
Maintainability	*	*	*
✓ Testability	*		maintainability
Interoperability	*		
Flexibility	*	*	
Reusability	*	*	
Portability	*	*	*
Clarity		*	
✓ Modifiability		*	maintainability
Documentation		*	
Resilience		*	
Understandability		*	
✓ Validity		*	maintainability
Functionality			*
Generality		*	
Economy		*	

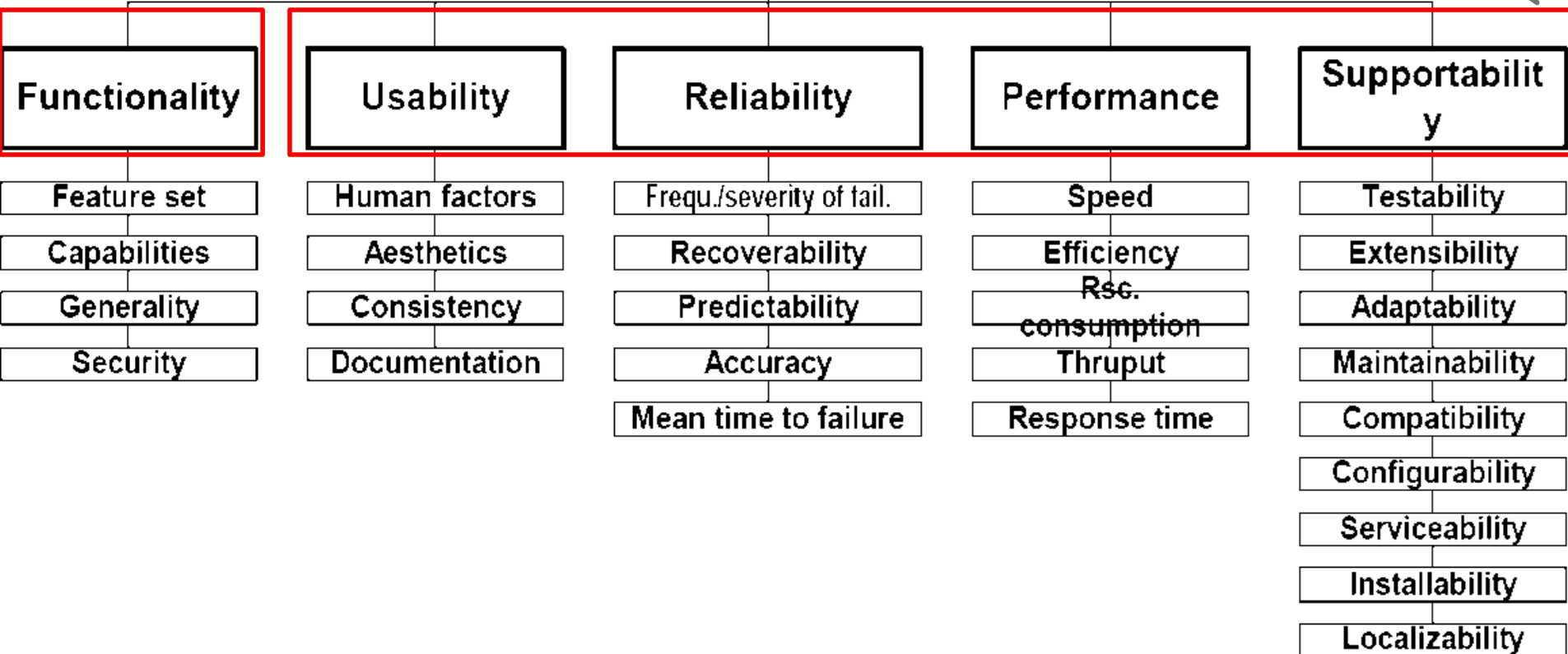
FURPS



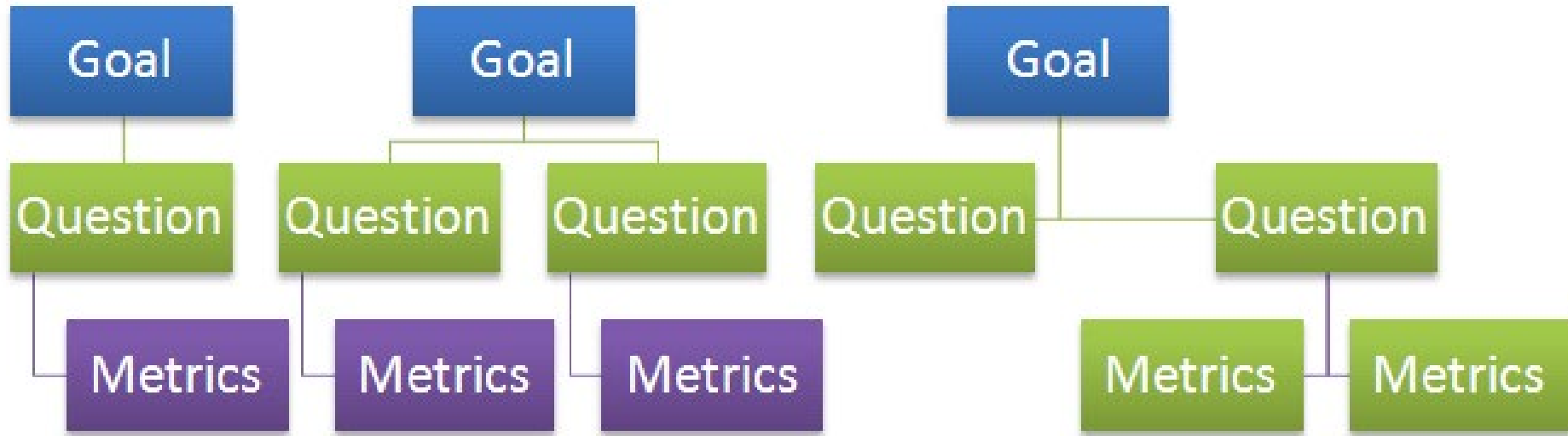
Functional

Non-functional

Quality



GQM: Goal-Question-Metric



- A **three-step** framework
- **Goal**: List the major goals of the development/maintenance project
- **Question**: derive questions from each goal that must be answered to determine if the goal is met
- **Metrics**: decide what must be measured in order to answer the questions

GQM: Goal-Question-Metric

Goal: evaluate the effectiveness of writing sw code using standards

Questions: who's using standard? What is coders experience? What is LOC quality?

Metrics:

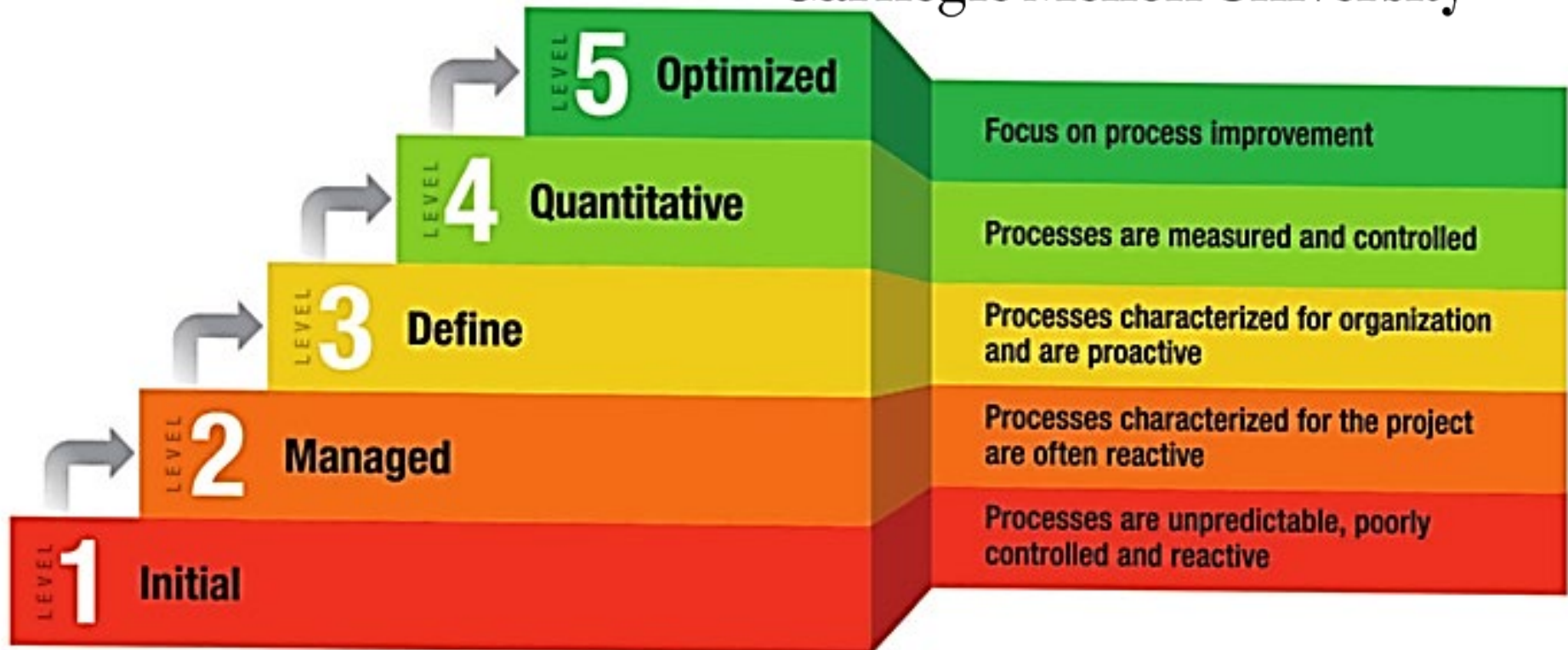
- Proportions of "coders"
 - using std
 - by Progr. Lang.
- Experience of "coders"
 - with std
 - with PL
 - by env
 - ...
- Sw size/length
 - LOC
 - FP/CFP
 - ...
- Effort
- Errors

CMM

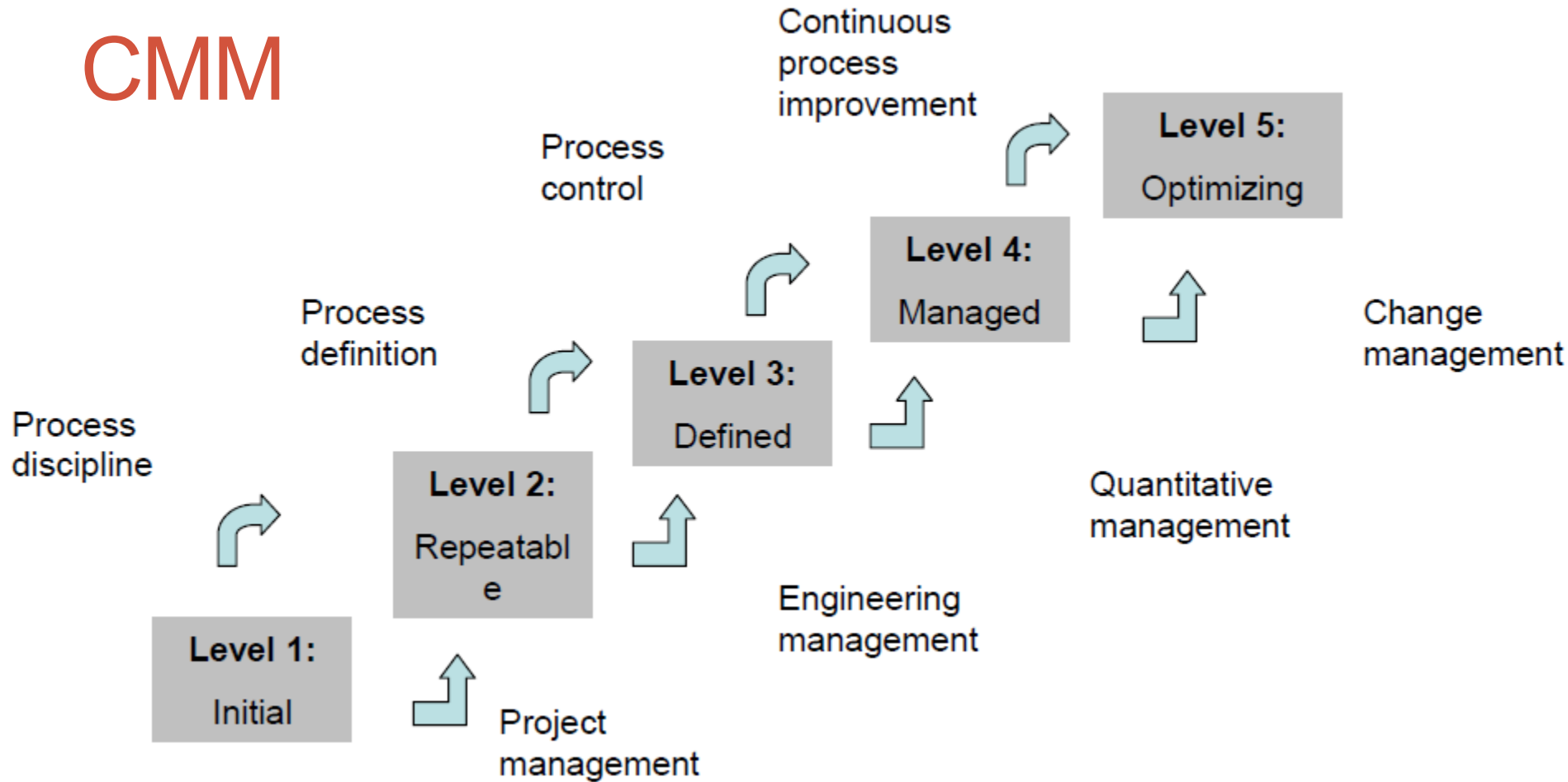
- Capability Maturity Model (CMM)



Software Engineering Institute
Carnegie Mellon University

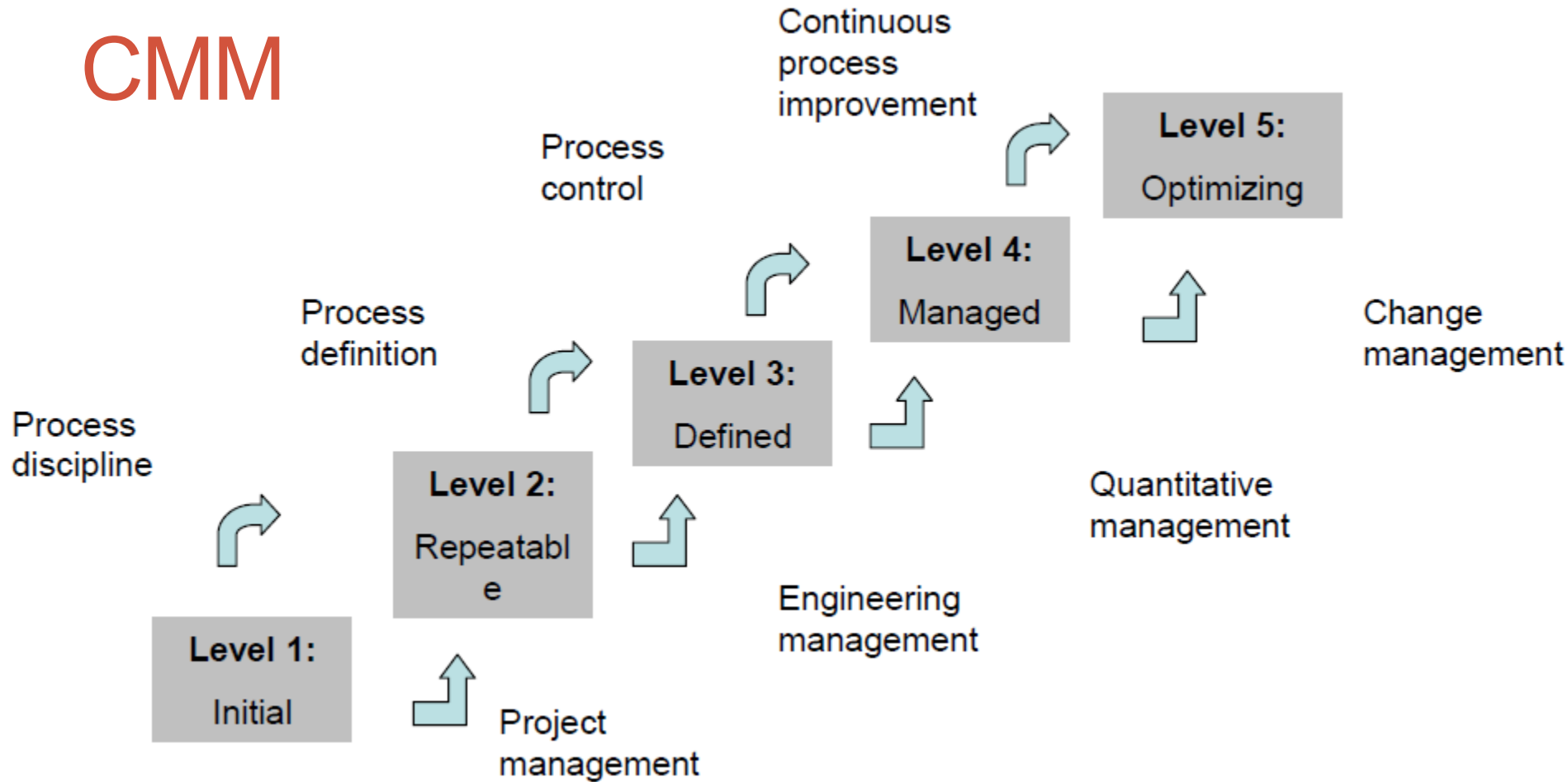


CMM



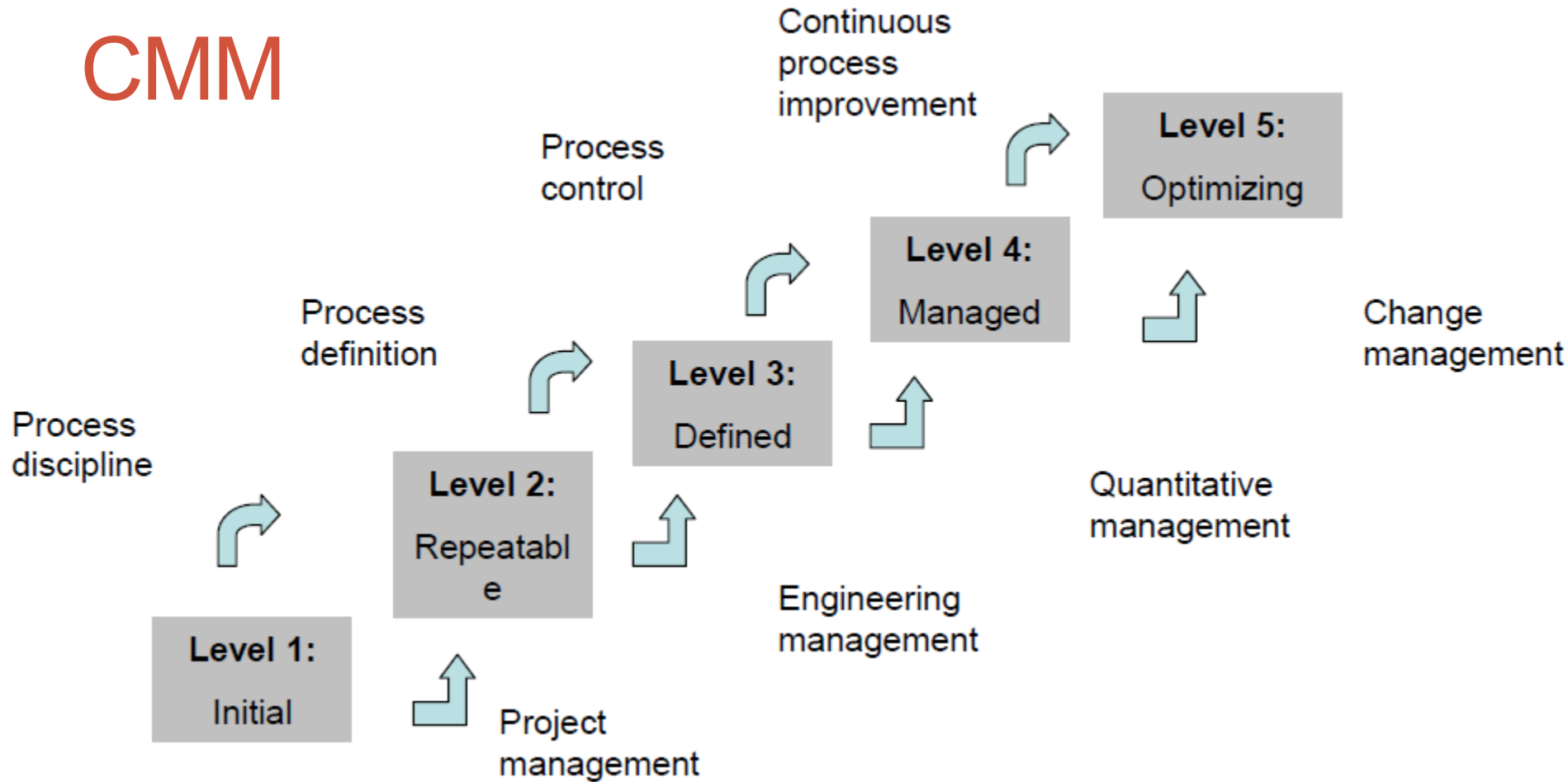
- **Level 1 (Initial Level):**
 - software process is **disciplined**, but unpredictable, poorly controlled and reactive.

CMM



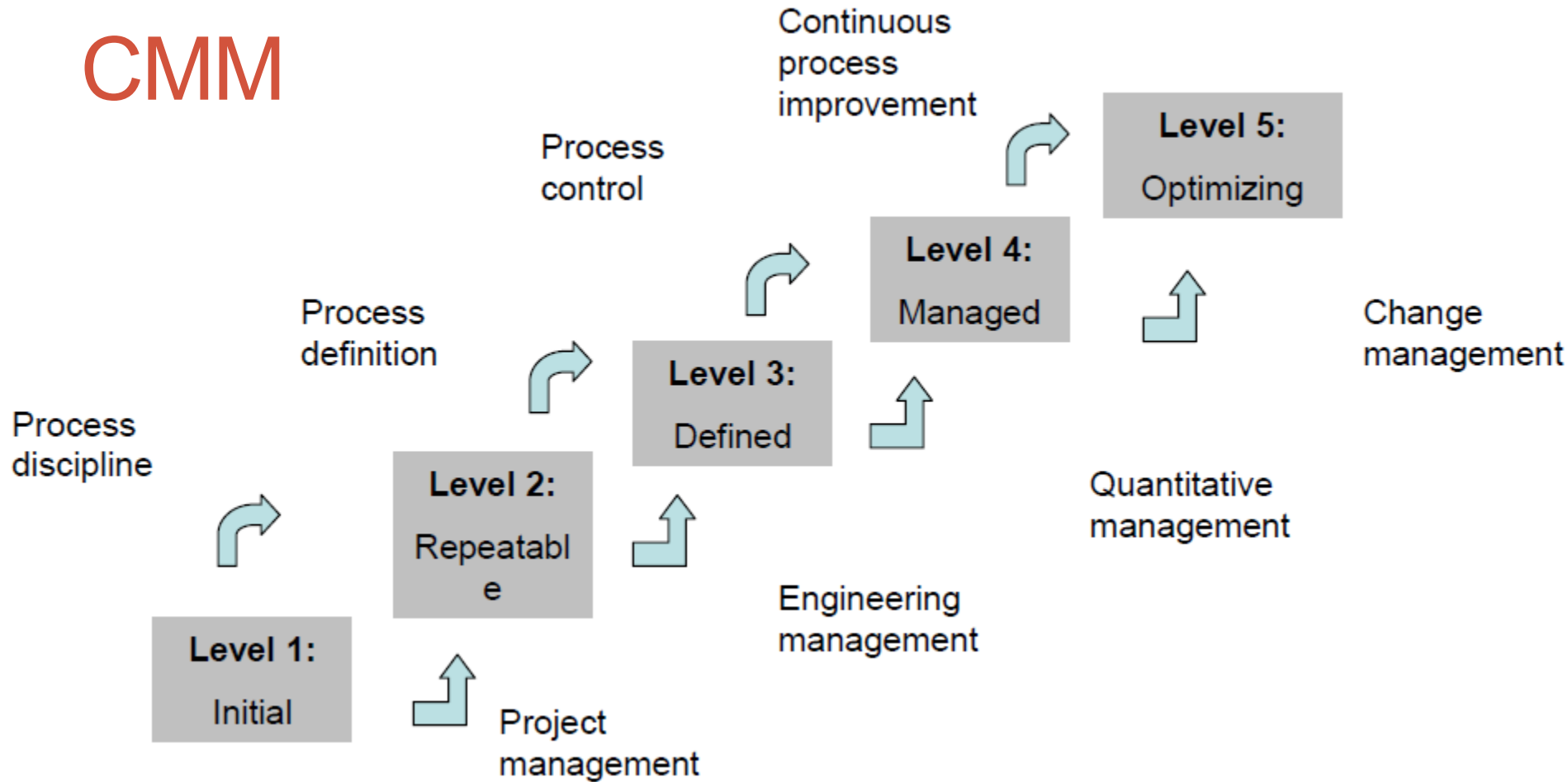
- Level 2 (**Repeatable** Level):
 - software process is **repeatable**, but is **reactive** and characterized for specific projects.

CMM



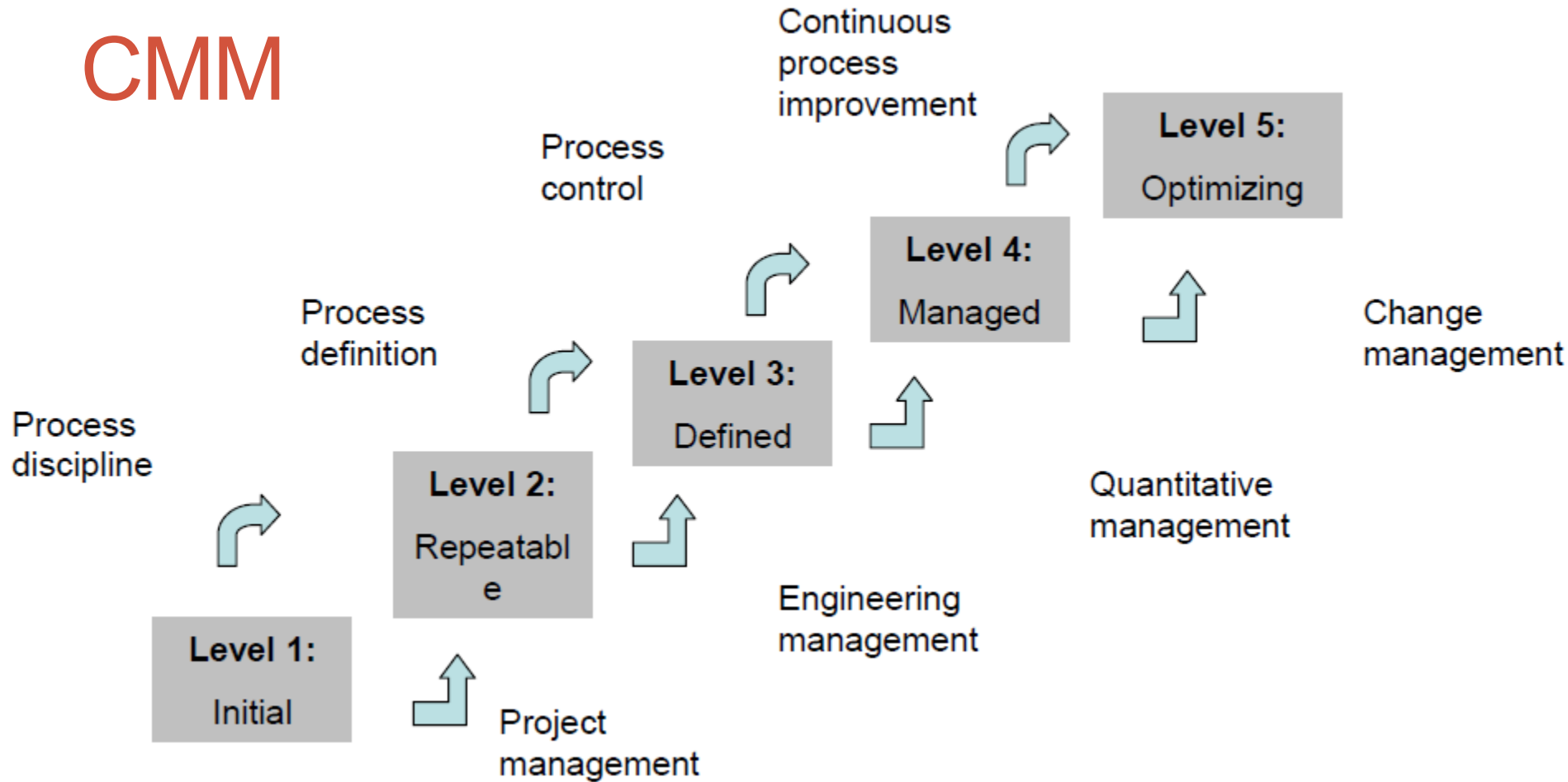
- Level 3 (**Defined** Level):
 - software process is **defined**; no longer reactive but **proactive**; also process is characterized not for specific projects but for the entire organization (business)

CMM



- Level 4 (**Managed Level**):
 - software process is **quantitatively managed**; process is measured and controlled.

CMM



- Level 5 (**Optimizing** Level):
 - software process adapts to **changes** and is continuously **improved**.

CMM

Table 1: Maturity levels with corresponding focus and key process areas for CMM.

Level	Focus	Key Process Area
Level 5 – Optimizing level	Continuous improvement	Process Change Management Technology Change Management Defect Prevention
Level 4 – Managed level	Product and process quality	Software Quality Management Quantitative Process Management
Level 3 – Defined level	Engineering process	Organization Process Focus Organization Process Definition Peer Reviews Training Program Intergroup Coordination Software Product Engineering Integrated Software Management
Level 2 – Repeatable level	Project Management	Requirements Management Software Project Planning Software Project Tracking and Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management
Level 1 – Initial level	Heroes	No KPAs at this time

Summary

- Concepts of software quality model
- The McCall quality model
 - Quality factors - criterion - metrics
- The Boehm's quality model
 - Three levels of quality characteristics, in contrast to the McCall
- The ISO/IEC 9126 quality model
 - Factors and subfactors
 - Compared to McCall and Boehm's models
- Application/company specific models
 - FURPS and GQM
- CMM
 - Process quality: five maturity levels