On Bringing Object-Oriented Software Metrics into the Model-Based World

Verifying ISO 26262 Compliance in Simulink

Lukas Mäurer, Torben Stolte

Tanja Hebecker, Michael Lipaczewski

Uwe Möhrstädt, Frank Ortmeier

Functional safety

> absence of unreasonable risk due to hazards caused by malfunctioning behaviour of E/E systems
>
> ISO 26262:2011

Car

Driver assistance	Active safety	Drivetrain	...
• ACC • Lane Assistance	• Airbag • ABS • ESP	• Automatic transmission • Drive-by-wire	• ... • ...

Chair of Software Engineering

1. Vocabulary

2. Management of functional safety

2-5 Overall safety management	2-6 Safety management during item development	2-7 Safety management after release for production

3. Concept phase

3-5 Item definition

3-6 Initiation of the safety lifecycle

3-7 Hazard analysis and risk assessment

3-8 Functional safety concept

4. Product development: system level

4-5 Initiation of product development at the system level

4-6 Specification of the technical safety requirements

4-7 System design

4-11 Release for production

4-10 Functional safety assessment

4-9 Safety validation

4-8 Item integration and testing

7. Production and operation

7-5 Production

7-5 Operation, service (maintenance and repair), and decommissioning

5. Product development: hardware level

5-5 Initiation of product development at the hardware level

5-6 Specification of hardware safety requirements

5-7 Hardware design

5-8 Hardware architectural metrics

5-9 Evaluation of violation of the safety goal due to random HW failures

5-10 Hardware integration and testing

6. Product development: software level

6-5 Initiation of product development at the software level

6-6 Specification of software safety requirements

6-7 Software architectural design

6-8 Software unit design and implementation

6-9 Software unit testing

6-10 Software integration and testing

6-11 Verification of software safety requirements

Core processes

8. Supporting processes

8-5 Interfaces within distributed developments	8-10 Documentation
8-6 Specification and management of safety requirements	8-11 Qualification of software tools
8-7 Configuration management	8-12 Qualification of software components
8-8 Change management	8-13 Qualification of hardware components
8-9 Verification	8-14 Proven in use argument

9. ASIL-oriented and safety-oriented analyses

9-5 Requirements decomposition with respect to ASIL tailoring	9-7 Analysis of dependent failures
9-6 Criteria for coexistence of elements	9-8 Safety analyses

10. Guideline on ISO 26262 (informative)

Chair of Software Engineering

Development process

Chair of Software Engineering

- Modularity
- Encapsulation
- Simplicity

ISO 26262-6:2011

	Methods	ASIL			
		A	B	C	D
1a	Hierarchical structure of software components	++	++	++	++
1b	Restricted size of software components	++	++	++	++
1c	Restricted size of interfaces	+	+	+	+
1d	High cohesion within each software component	+	++	++	++
1e	Restricted coupling between software components	+	++	++	++
1f	Appropriate scheduling properties	++	++	++	++
1g	Restricted use of interrupts	+	+	+	++

Chair of Software Engineering

Methods for the verification of the software architectural design

ISO 26262-6:2011

Methods		ASIL			
		A	B	C	D
1a	Walk-through of the design	++	+	o	o
1b	Inspection of the design	+	++	++	++
1c	Simulation of dynamic parts of the design	+	+	+	++
1d	Prototype generation	o	o	+	++
1e	Formal verification	o	o	+	+
1f	Control flow analysis	+	+	++	++
1g	Data flow analysis	+	+	++	++

Model
- Simulation
- Model reviews

Automatic code generation

Code
- Metrics
- Code-Review

Compilation

Software
- Tests

Chair of Software Engineering

Automatic code generation

Compilation

Model
- Simulation
- Model reviews

Code
- Metrics
- ~~Code-Review~~

Software
- ~~Tests~~

Chair of Software Engineering

Automatic code generation

Compilation

Model

- Simulation
- Model reviews
- **Metrics**

Code

Software

Chair of Software Engineering
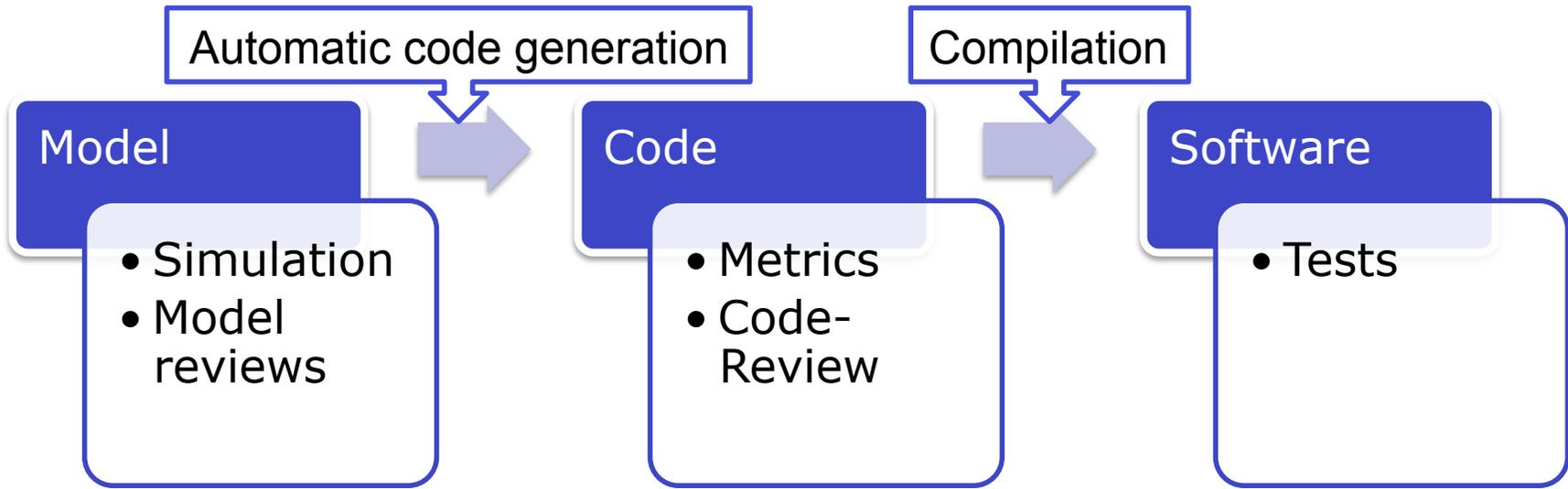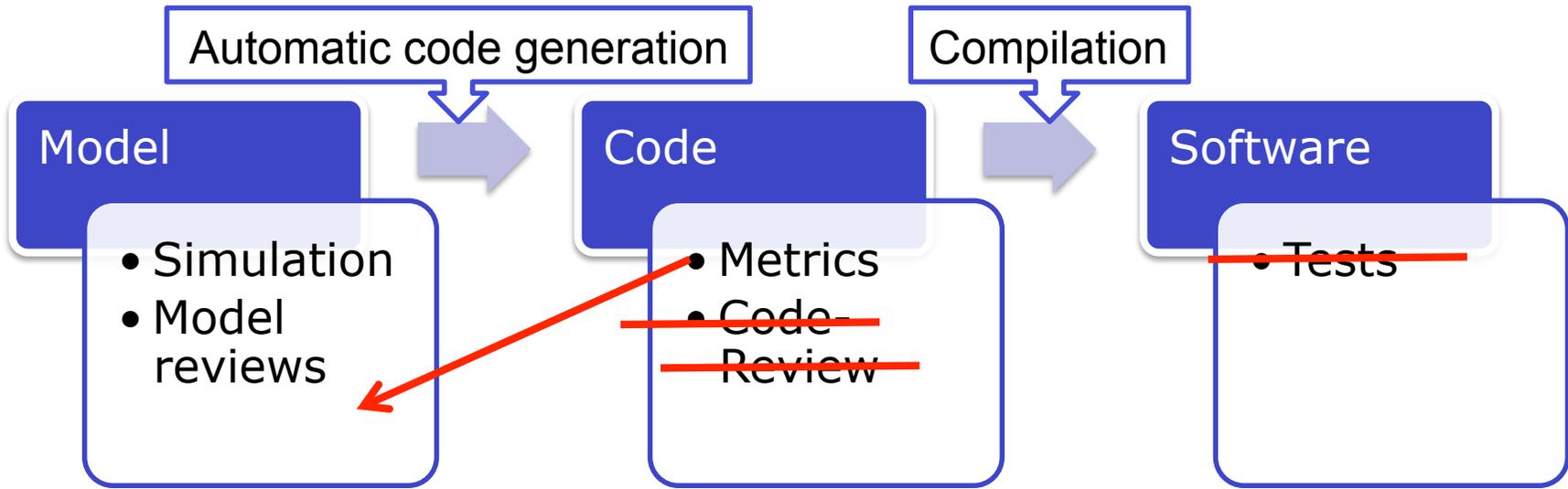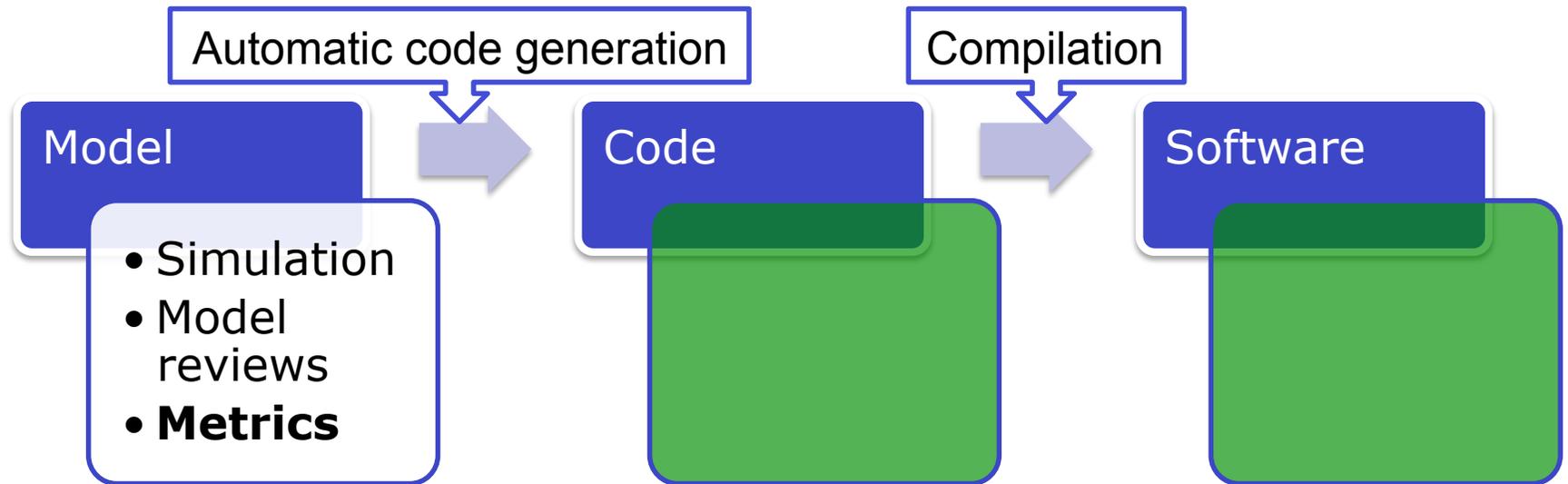
- **Widespread Metric suites**
  - Halstead, M.H.: Elements of Software Science (Operating and programming systems series). Elsevier Science Inc. (1977)
  - Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. Software Engineering, IEEE Transactions on, 20(6), 476-493.

➡ Overview over existing metrics

- **Evaluations of metrics**
  - Briand, L. C., Daly, J. W., & Wüst, J. (1998). A unified framework for cohesion measurement in object-oriented systems. Empirical Software Engineering, 3(1), 65-117.
  - Briand, L. C., Daly, J. W., & Wüst, J. (1999). A unified framework for coupling measurement in object-oriented systems. Software Engineering, IEEE Transactions on, 25(1), 91-121.
  - Mayer, T., & Hall, T. (1999, July). Measuring OO systems: a critical analysis of the MOOD metrics. In Technology of Object-Oriented Languages and Systems, 1999. Proceedings of (pp. 108-117). IEEE.

➡ Comparison of metrics, design flaws
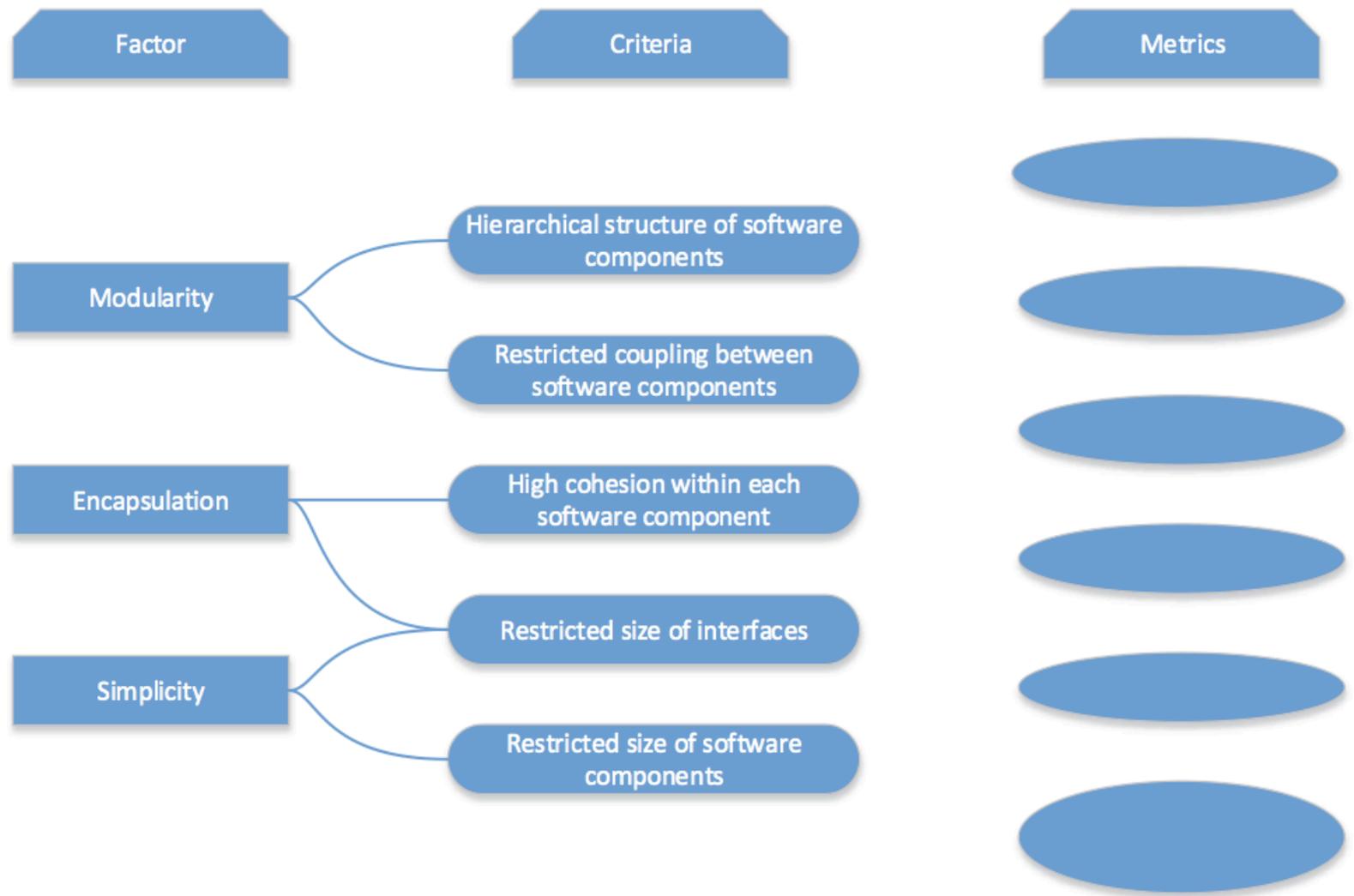
Factor	Criteria	Metrics
Modularity	Hierarchical structure of software components	
	Restricted coupling between software components	
Encapsulation	High cohesion within each software component	
Simplicity	Restricted size of interfaces	
	Restricted size of software components	

Cavano, J.P., McCall, J.A.: A framework for the measurement of software quality. In: Proceedings of the Software Quality Assurance Workshop on Functional and Performance Issues. pp. 133–139. ACM (1978)

Factor	Criteria	Metrics
Modularity	Hierarchical structure of software components	Range of Demeter
	Restricted coupling between software components	FanIn/FanOut
Encapsulation	High cohesion within each software component	Tight Class Cohesion/ Loose Class Cohesion
Simplicity	Restricted size of interfaces	Element Hiding Factor
	Restricted size of software components	Halstead volume
		Number of Elements
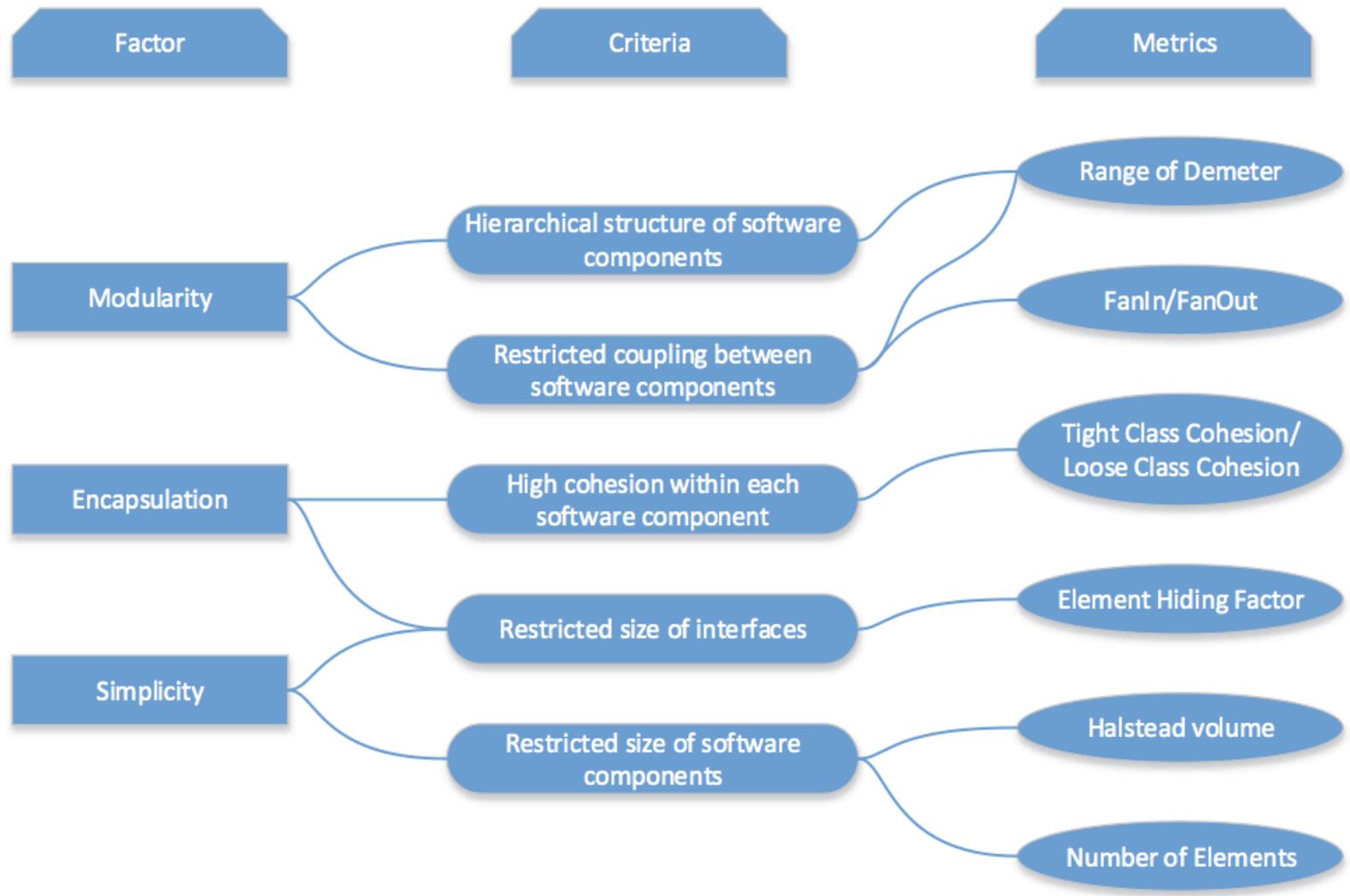
Cavano, J.P., McCall, J.A.: A framework for the measurement of software quality. In: Proceedings of the Software Quality Assurance Workshop on Functional and Performance Issues. pp. 133–139. ACM (1978)

- Original:
  - „don't talk to strangers"
  - ¬ Object1.object2.method1()
  - ✓ Object1.method2()
- Model-based derivation:
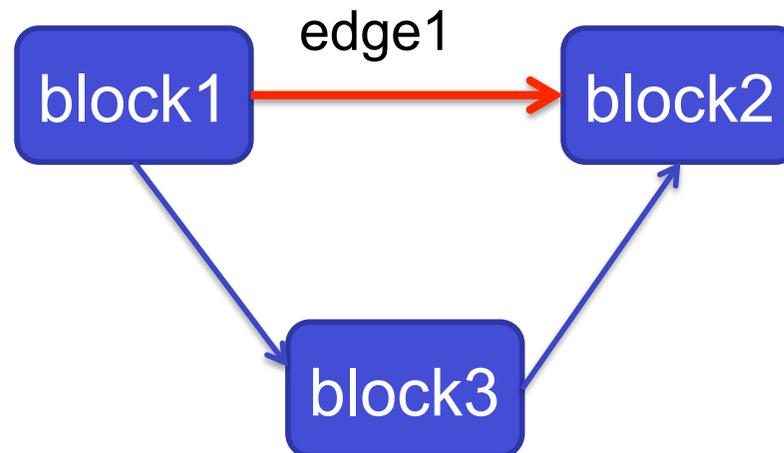
Lieberherr, K., Holland, I., Riel, A.: Object-oriented programming: an objective sense of style. In: Conference Proceedings on Object-oriented Programming Systems, Languages and Applications. pp. 323–334. OOPSLA '88, ACM (1988)

Chair of Software Engineering

- ## Model-based derivation:
  - If two blocks block1 and block2 are indirectly connected, they must not be directly connected
- ## Metric:
  - Count number of skipped blocks

Chair of Software Engineering
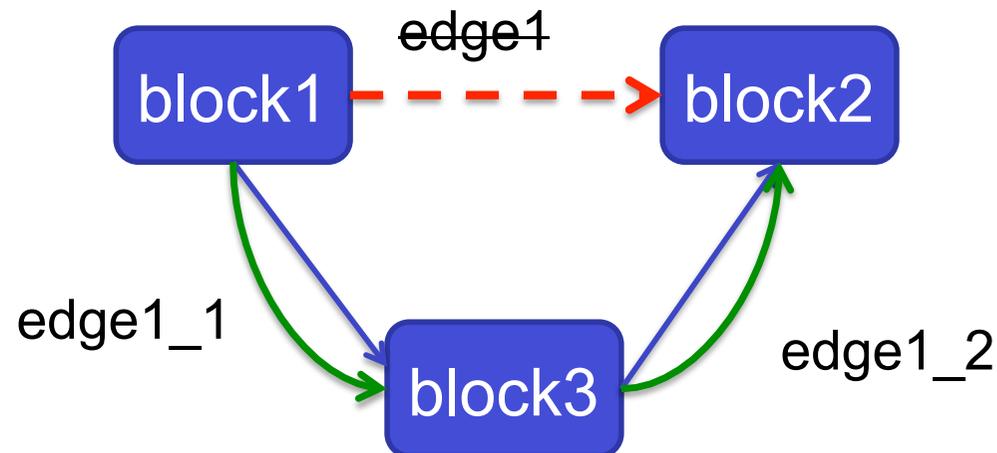
- Original:
  - Method Hiding Factor

  $$MHF = \frac{\#HiddenMethods}{\#Methods}$$

  - Attribute Hiding Factor

  $$AHF = \frac{\#HiddenAttributes}{\#Attributes}$$

Abreu, F.B., Carapuça, R.: Object-oriented software engineering: Measuring and controlling the development process. In: 4th Int. Conf. on Software Quality (1994)

- Model-based derivation:

$$EHF = \frac{\#HiddenElements}{\#Elements} = \frac{6}{13} = 0.46$$

Chair of Software Engineering
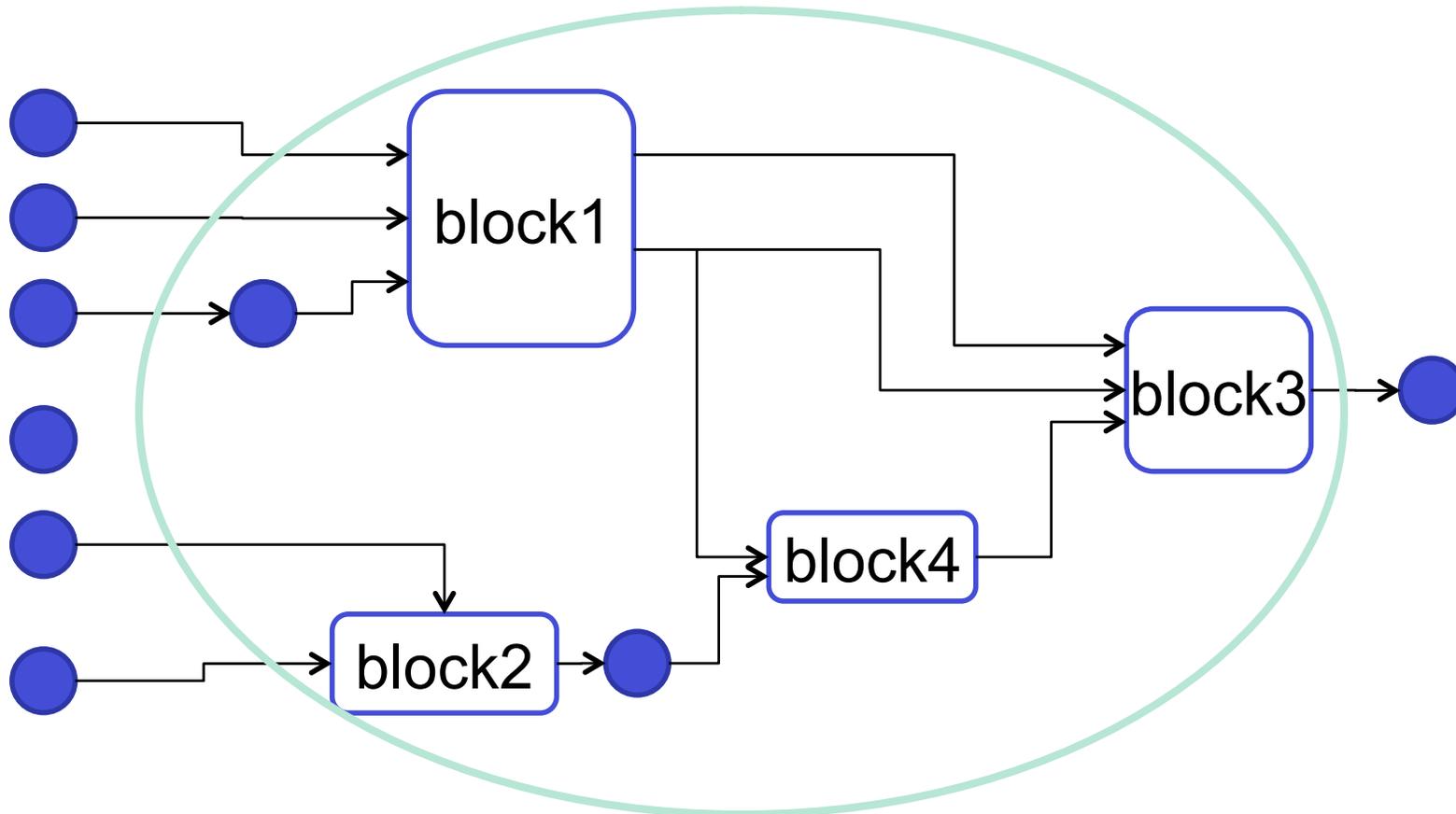
http://www.pitchvision.com/files/image/!stream/brain_shutterstock_154870703.jpg

Chair of Software Engineering

Metric	Target Value	Model SC	Model BMS
Halstead Volume	minimal	16554.93	53904.3
Number of Elements	minimal	2411	8099
Loose Block Cohesion	1	0.84	0.96
Tight Block Cohesion	>0.75	0.57	0.77
Element Hiding Factor	>0.75	0.51	0.55
Range of Demeter	0	14	0
FanIn (FI)	1<=FI<=3	2.18	1.83
FanOut (FO)	1<=FO<=3	1.94	1.57

Chair of Software Engineering

- **What do we have?**
- Approach for an automatical evaluation of ISO 26262 requirements for model-based software architecture (for dataflow languages)

- **What did we want?**
- Evaluation of ISO 26262 requirements
  - ✓ Fast (<30s)
  - ✓ Cheap
  - ✓ Objectiv

Chair of Software Engineering

- Expert review
- Case study for threshold definitions

- More metrics to cover special cases
- More input languages

- Visualisation and support for integration into the development process

Chair of Software Engineering

Questions?

lukas.maeurer@st.ovgu.de