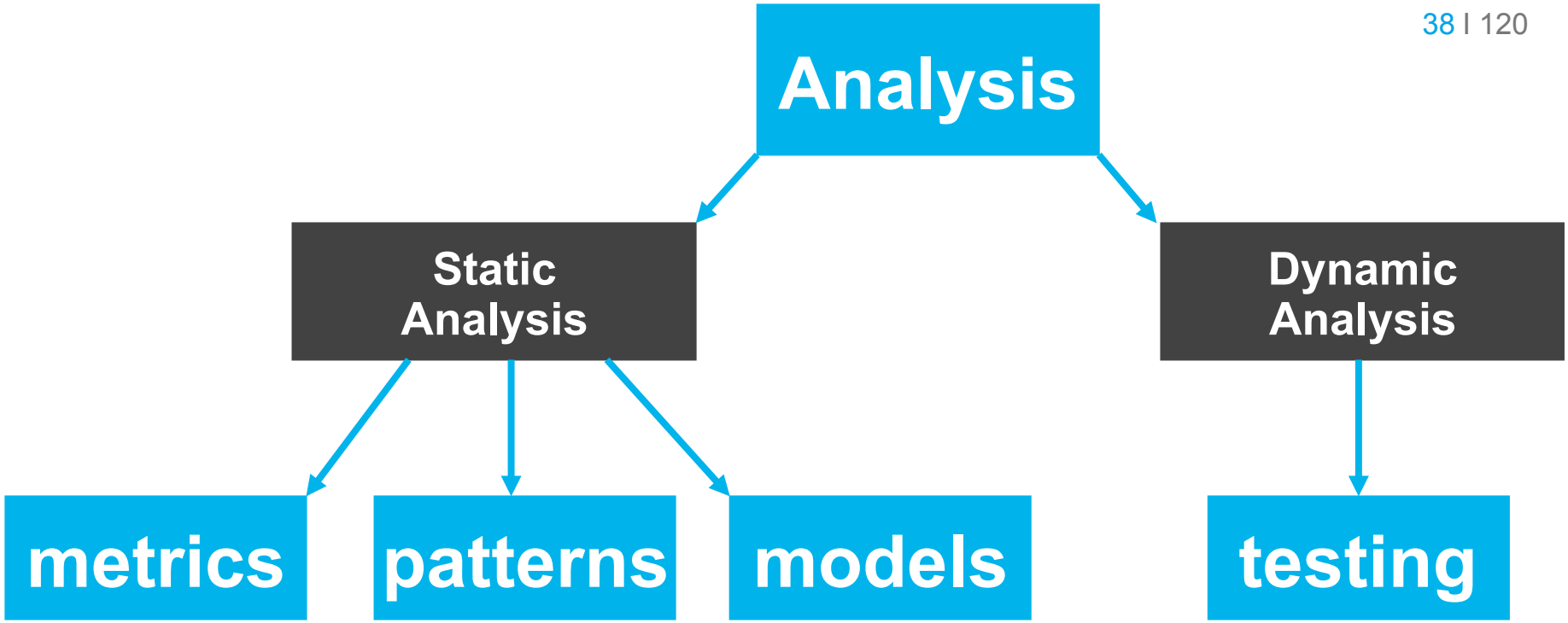
Structure of the lecture

Software Improvement Group



PATTERNS

Patterns

Coding style and coding standards

- E.g. layout, identifiers, method length, …

Secure coding guidelines

- E.g. SQL injection, stack trace visibility

Bug patterns

- E.g. null pointer dereferencing, bounds checking

Code smells

- E.g. “god class”, “greedy class”, ..

Patterns
Style and standards

Checking coding style and coding standards

- Layout rules (boring)
- Identifier conventions
- Length of methods
- Depth of conditionals

Aim

- Consistency across different developers
- Ensure maintainability

Tools

- E.g. CheckStyle, PMD, …
- Integrated into IDE, into nightly build
- Can be customized

Checking secure coding guidelines

- SQL injection attack
- Storing and sending passwords
- Stack-trace leaking
- Cross-site scripting

Aim

- Ensure security
- Security = Confidentiality + Integrity + Availability

Tools

- E.g. Fortify, Coverity

Detecting bug patterns

- Null-dereferencing
- Lack of array bounds checking
- Buffer overflow

Aim

- Correctness
- Compensate for weak type checks

Tools:

- e.g. FindBugs
- Esp. for C, C++

Run PMD / Checkstyle / FindBugs

- E.g. on a project of your own
- E.g. on some (easy-to-compile) open source project

Inspect results

- False or true positives?