

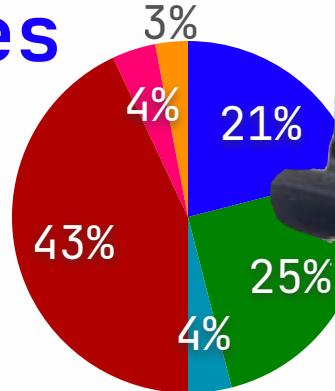
Software Analytics: Metrics

Software Systems – Design – L6T1

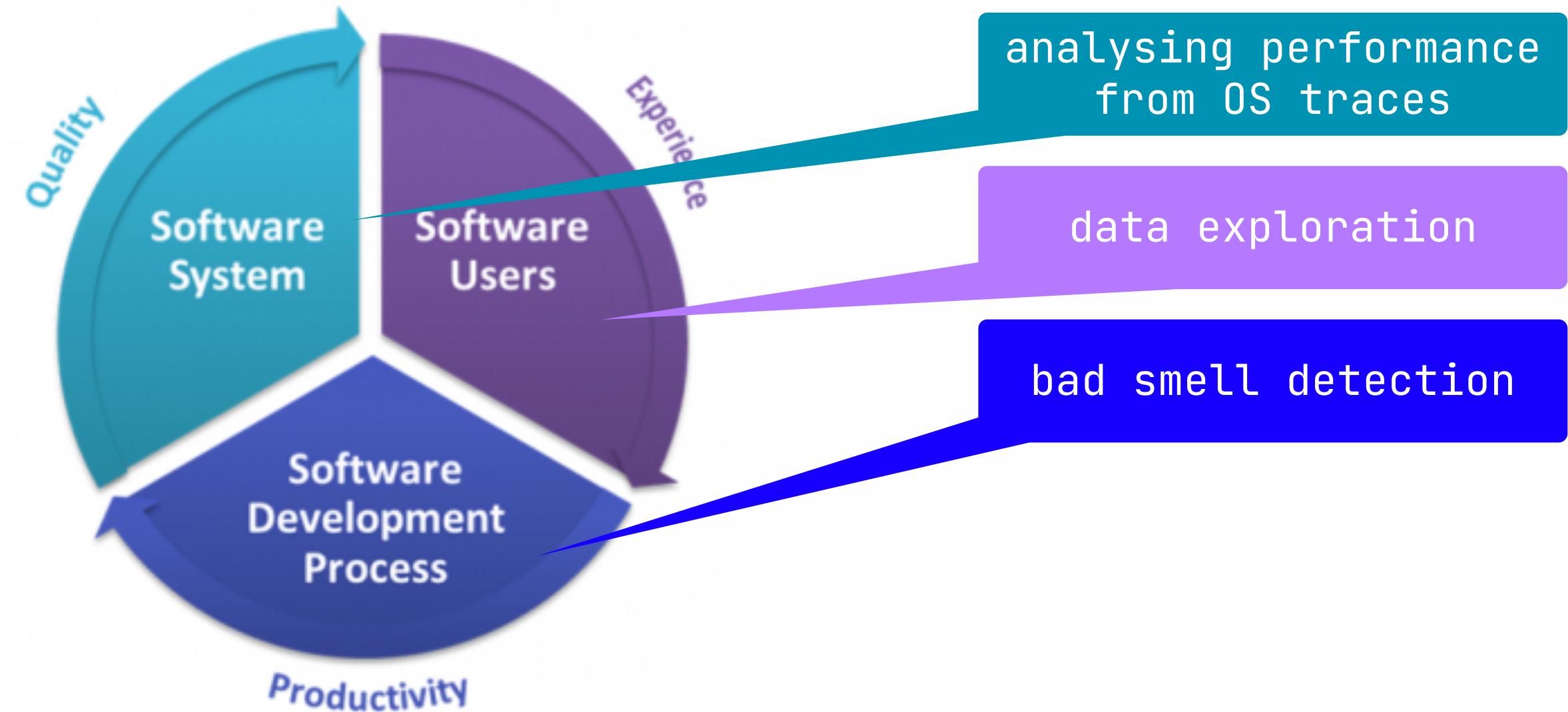
Dr. Vadim Zaytsev aka @grammarware, November 2020

Maintenance Types

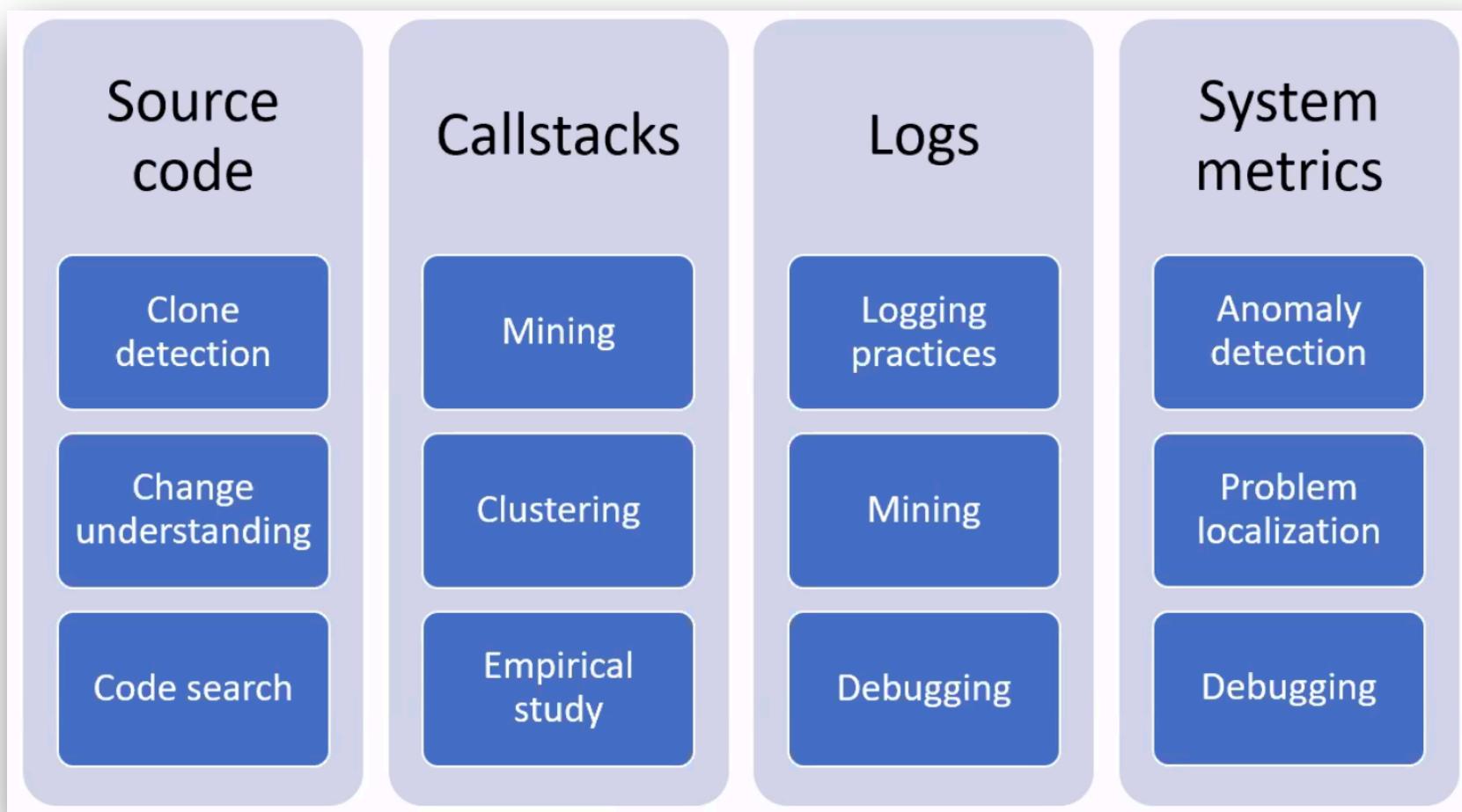
- Corrective
- Adaptive
- Perfective
 - user enhancement
 - efficiency
 - other
- Preventive



Software Analytics



Software Analytics



LOC

- What's a **line of code**?
 - physical (as written)
 - logical (statement?)

```
import java.util.Arrays;
import java.util.List;
List<String> myList = Arrays.asList("a1", "a2", "b1", "c2", "c1");
myList.stream().filter(s -> s.startsWith("c")).map(String::toUpperCase).sorted().forEach(System.out::println);
```

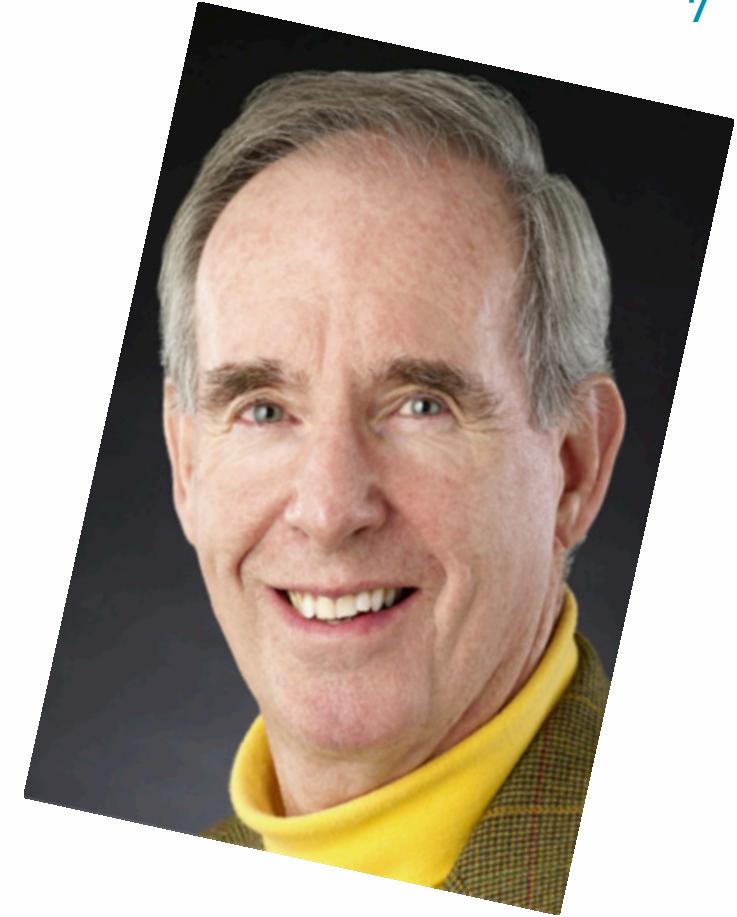
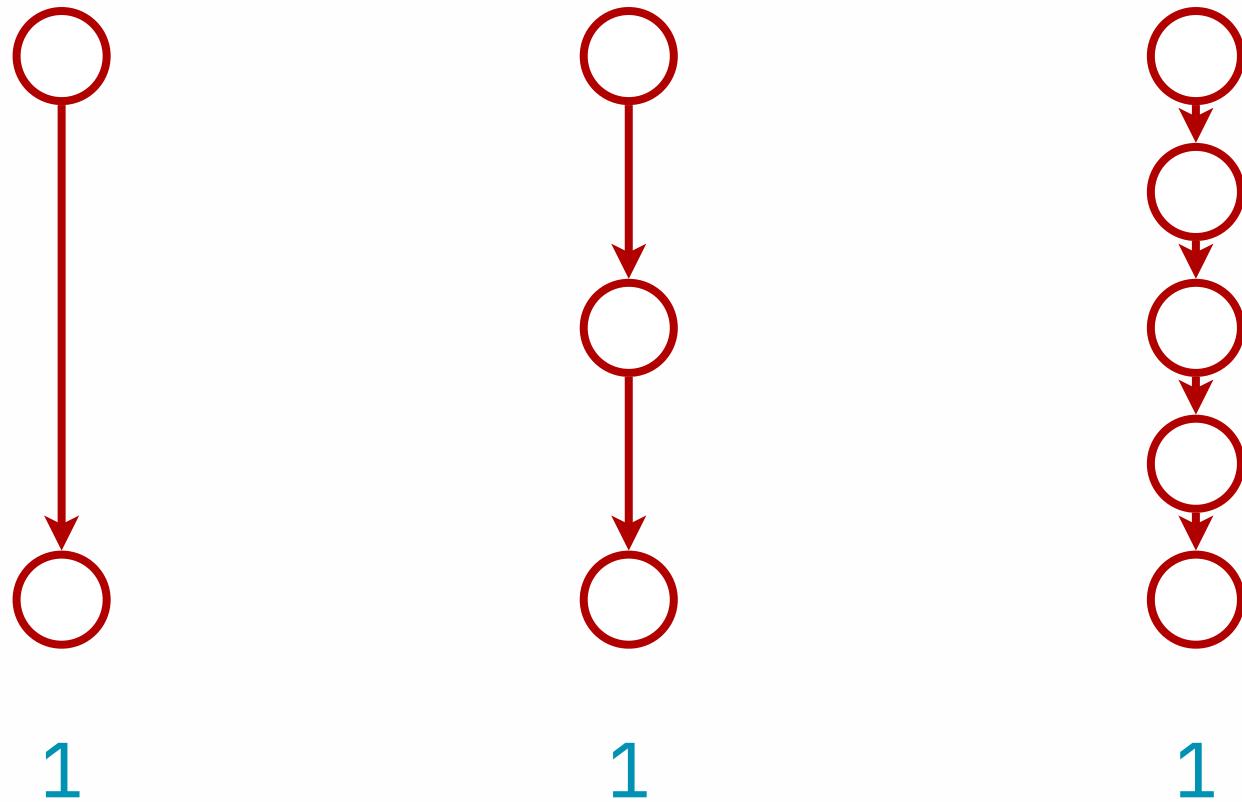
- LOC is an **abstraction** of
 - size
 - effort

CC

- What's complexity, intuitively?
- How hard to write/test/debug/...
- Number of test cases!

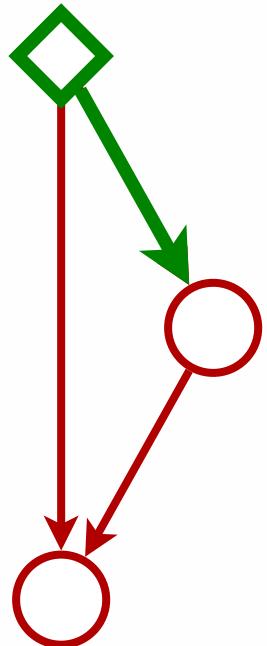


CC: sequence



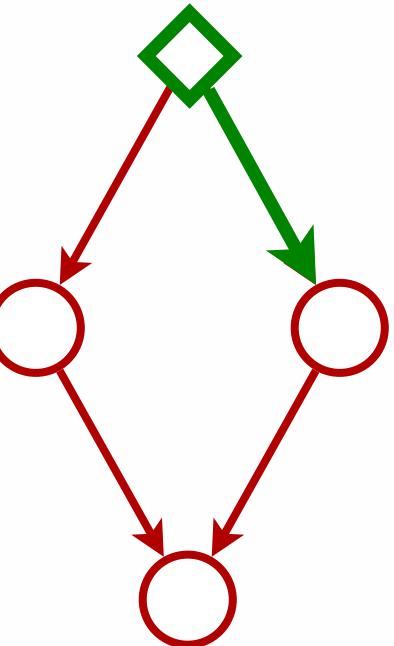
CC: if/else, switch/case

if



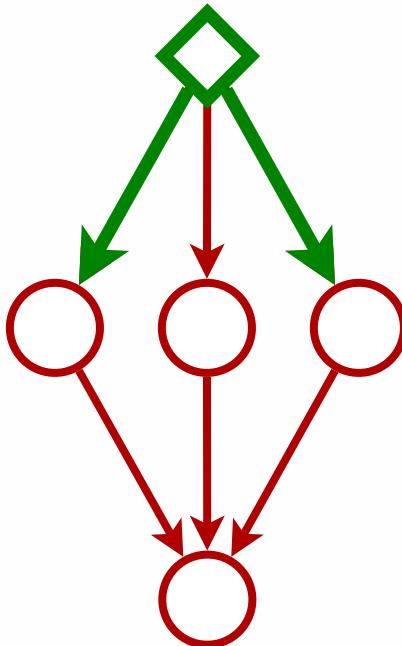
2

if / else

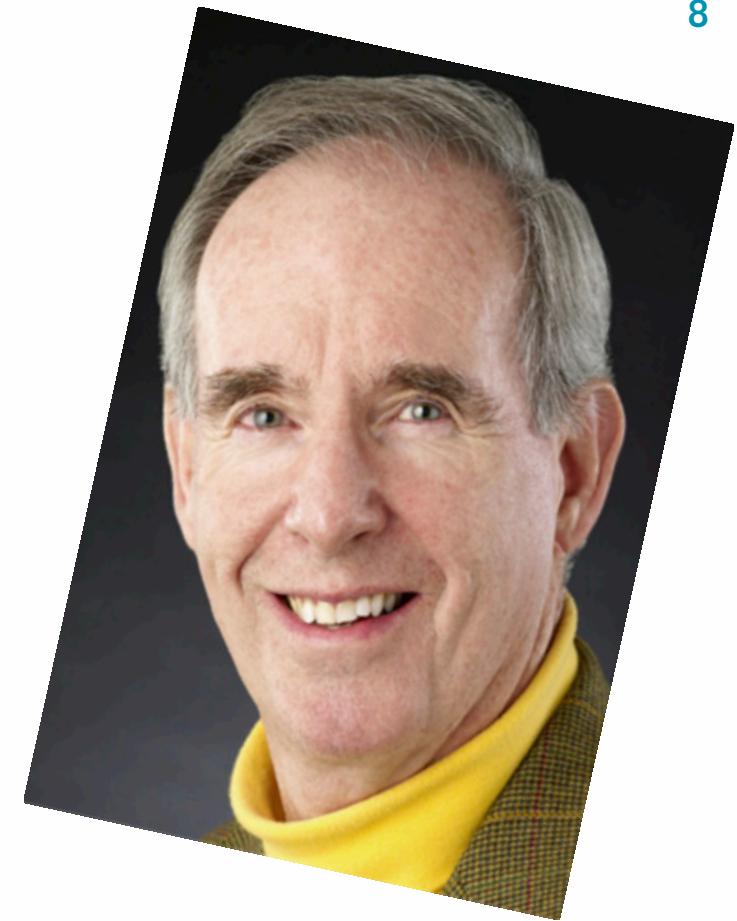


2

switch / case

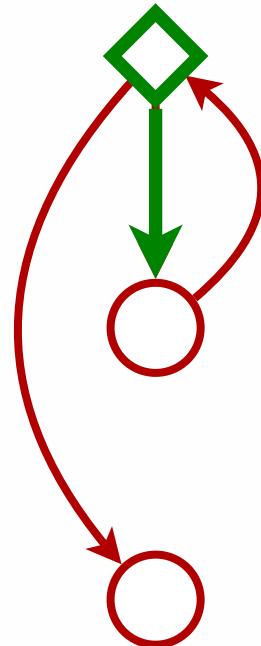


3



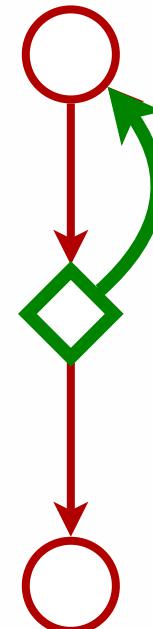
CC: while, for

while



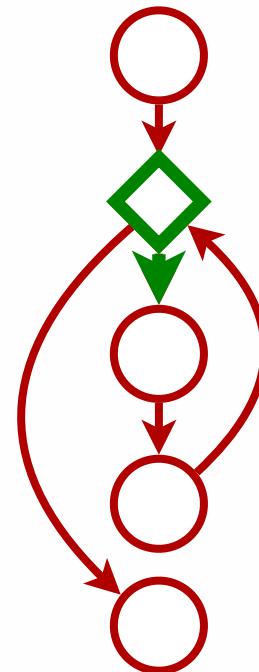
2

do / while

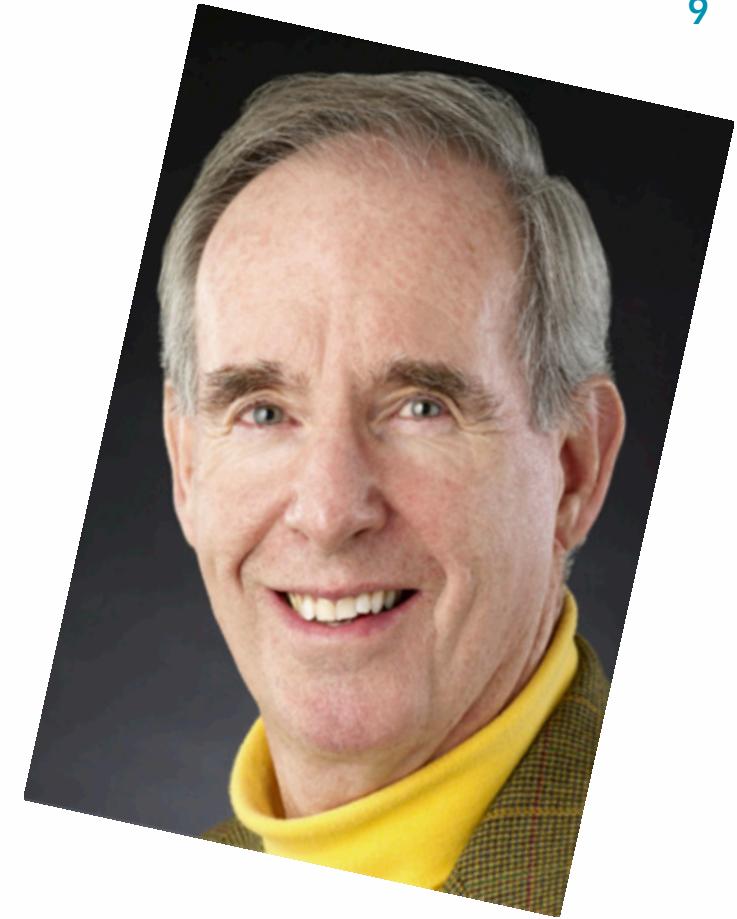


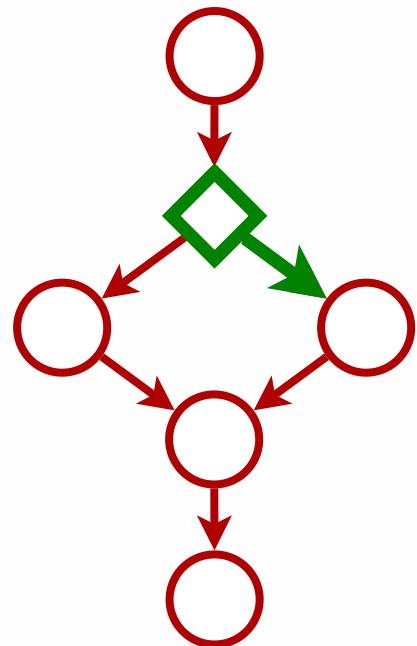
2

for

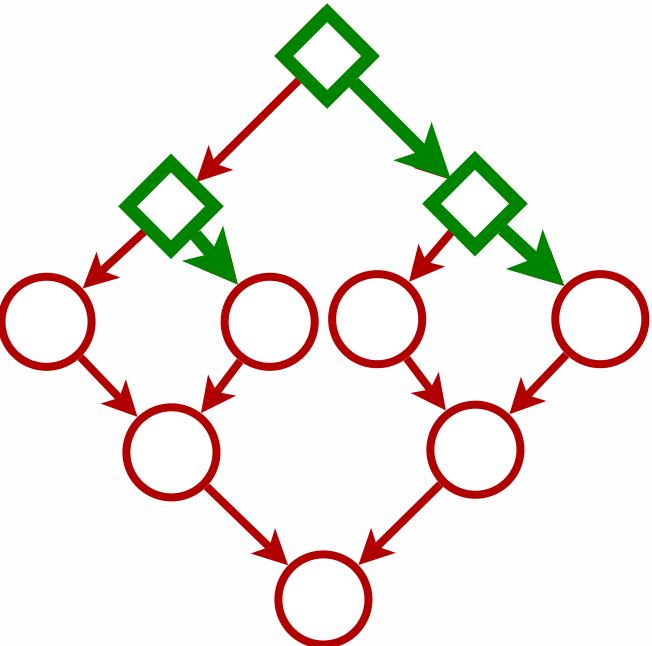


2

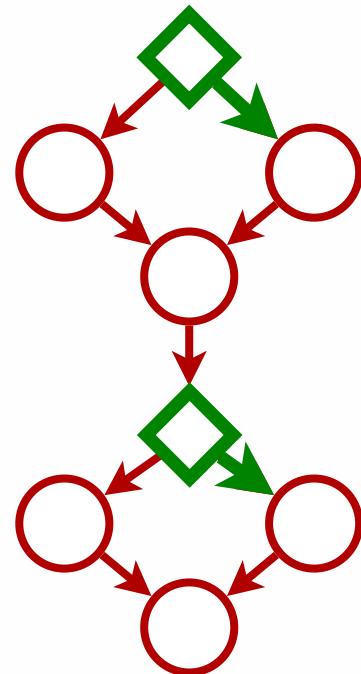


CC

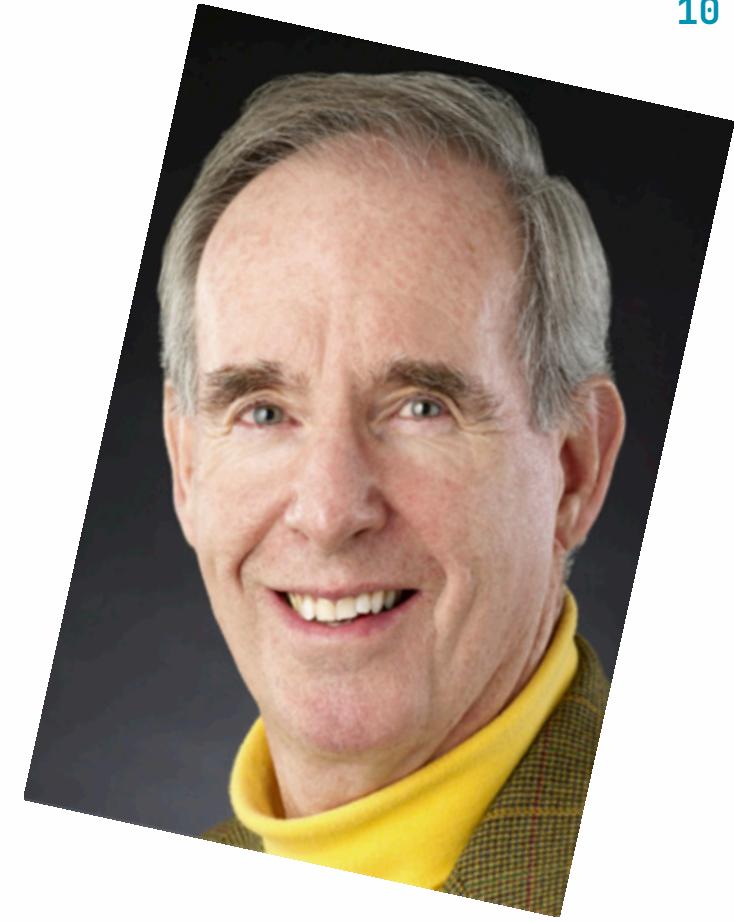
2



4



3



Software Metrics

- Code coverage
- Coupling
- Cohesion
- Defect density
- Function points
- Maintainability index
- Max inheritance depth
- MIPS
- Number of classes
- ... of methods per class
- Rate of delivery
- Satisfaction level
- Time to fix bugs
- . . .

Explore

Projects Issues

Filters

Quality Gate

Passed	31k
Warning	26
Failed	7.5k

Reliability (Bug)

A	22k
B	1.3k
C	7.4k
D	1.9k
E	5.4k

Security (Vulnerabilities)

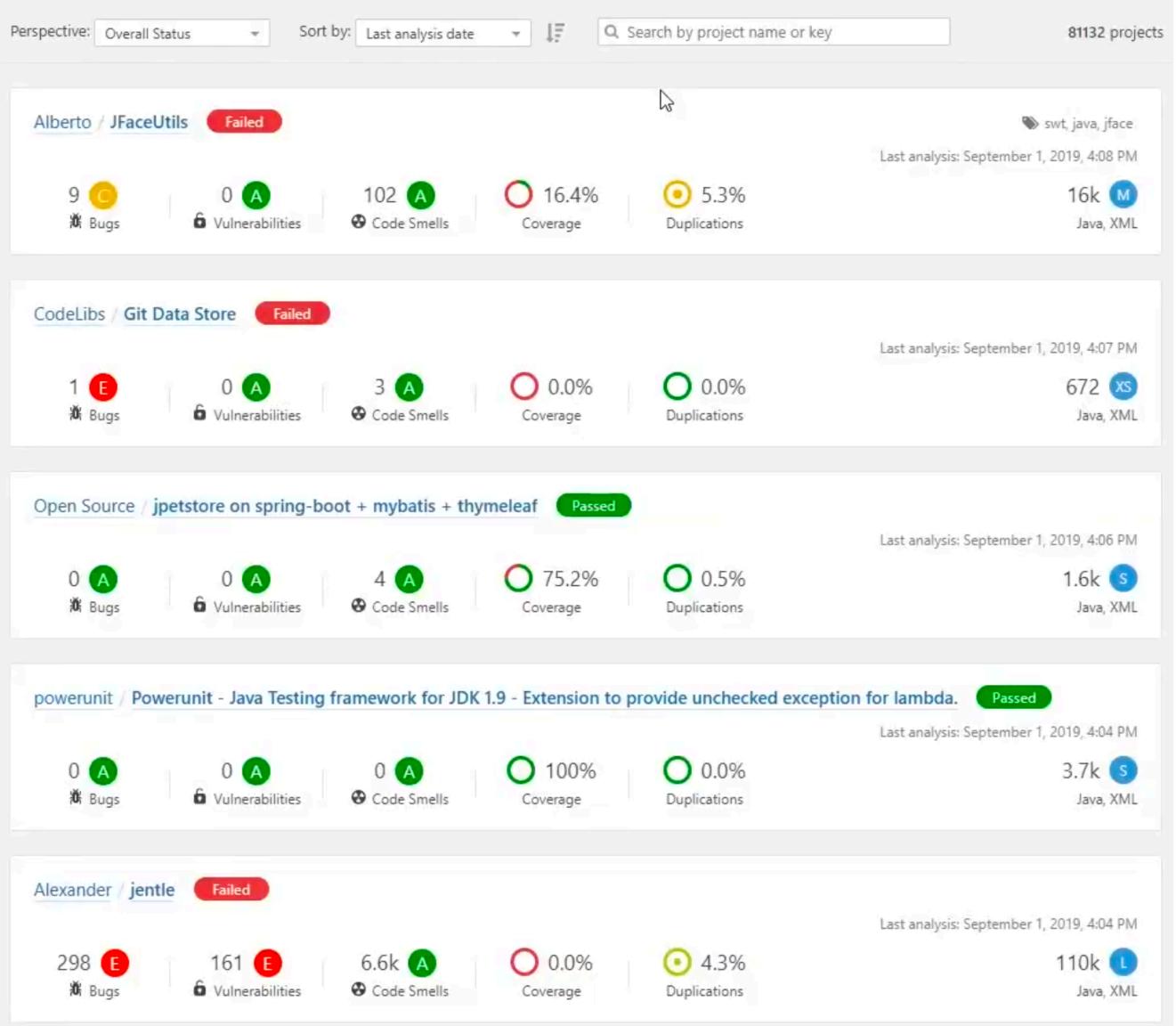
A	26k
B	6k
C	103
D	2.5k
E	3.9k

Maintainability (Code Smells)

A	37k
B	697
C	180
D	22
E	12

Coverage

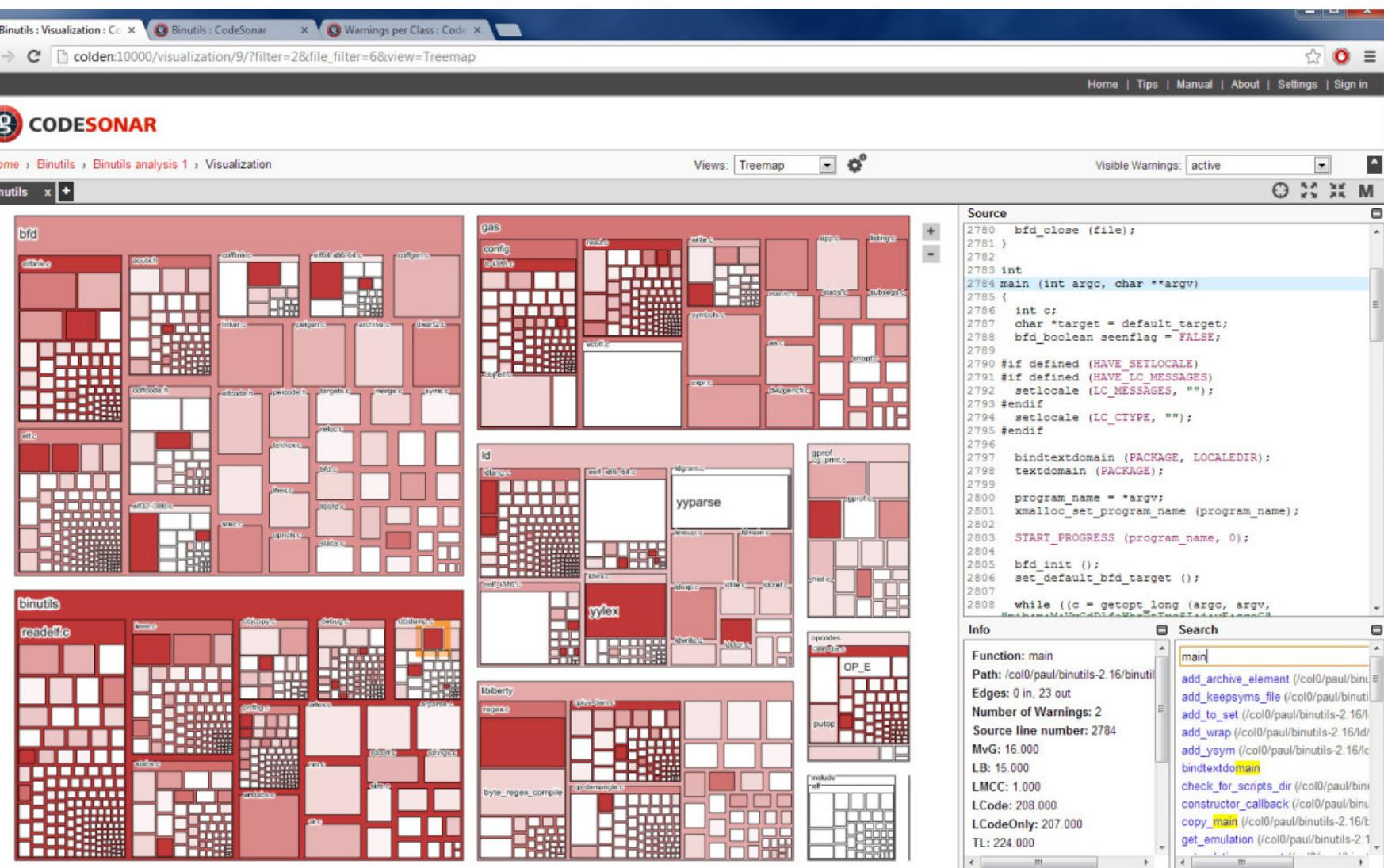
≥ 80%	3.9k
70% - 80%	1.2k
50% - 70%	1.8k
30% - 50%	1.5k
< 30%	25k
No data	47k



SonarQube

sonarqube

CodeSonar



John McHale, CodeSonar Visualization gets new enhancements..., 2013

Conclusion

- Software **analytics**
 - using collected data to understand software
- Software **metrics**
 - lines of code, cyclomatic complexity, etc
- Future / reality
 - from static metrics to clickstreams
 - rely on **advanced tools**

Topics/slides Disclaimer

- Good ✓

- watch before Q&A
- embrace reality
- try out at labs
- ask for feedback
- apply to project
- dig deeper
- recall from slides

- Bad ✗

- slides over videos
- assumptions
- blanks
- timing

