Course 2

Testing, Inspection, Walkthrough

How to perform SQ CONTROL?

- Testing
- Inspections
- Walkthroughs
- Reviews

Testing

- Process of gathering information by making observations and comparing them to expectations [Dale Emery]

- Most used quality improvement activity

Testing

- **Black-box**:

- **White-box**:

Testing

- **Black-box**:
  - tests the functionality of an application
  - Tester cannot see the inner code
- **White-box**:
  - tests internal structures of an application
  - Tester knows the inner code

Testing

- Unit testing
- …

Testing

- Unit testing – execution of a complete class/ routine written by a single person, tested in isolation
- Component testing – execution of a class/ package written by several persons, tested in isolation
- Integration testing – combined execution of 2 or more classes/ components/ packages/ subsystems created by teams – continuous process
- Regression testing – repetition of previously executed test cases
- System testing – execution of the final configuration, including integration with other systems

Non functional testing

- Assess system properties (attributes?)
  - Non critical to functionality
  - Target user experience

- Examples ?

Debugging

- $\approx$ testing?

- When you find an error (execute test case) => 2-steps process:

  1. Determine the location and category of error
  2. Fix the error

Testing vs. Debugging

Testing	Debugging
starts with known conditions, uses predefined methods, and has predictable outcomes	starts from possibly unknown initial conditions and its end cannot be predicted
Performed by testing team	Performed by development team
Can be automated	-
Goal: find as many bugs as possible	Goal: find and remove a bug
Find bugs	Find cause of the bug

Testing tools

- Automate testing process
- Tool for generating test cases
- Tool for performing testing: unit, integration, system

Conclusion: Testing $\Rightarrow$ Software Quality

✓ Testing – important part of SQA

× Testing cannot prove erro

× One testing strategy (unit integration) – finds ≤ 50%

× Combination of testing st 60% errors

=> only testing does not improve SQ

Myers classic test:
1 program – 15 errors
Average ≈5 /15
Best ≈ 9/15

Software Inspection

- Reading or visually inspecting the code
- Best industry practice for detecting software defects <u>early</u> and <u>learning</u> about software artifacts
- Include:
  - the structured review process,
  - standard of excellence product checklists,
  - defined roles of participants, and
  - the forms and reports
- Improve quality attributes: reliability, availability and maintainability

Steps in Software Inspection

- Systematic procedure – all life-cycle
- Steps
- System of checklists
- Roles
- Forms and repor

•Planning,
•Preparation,

- Moc
- Proc
- Reac
- Revi
- Reco
- Manag
- Consumer

•Inspection Record
•Inspection Reporting Form
•Report Summary Form

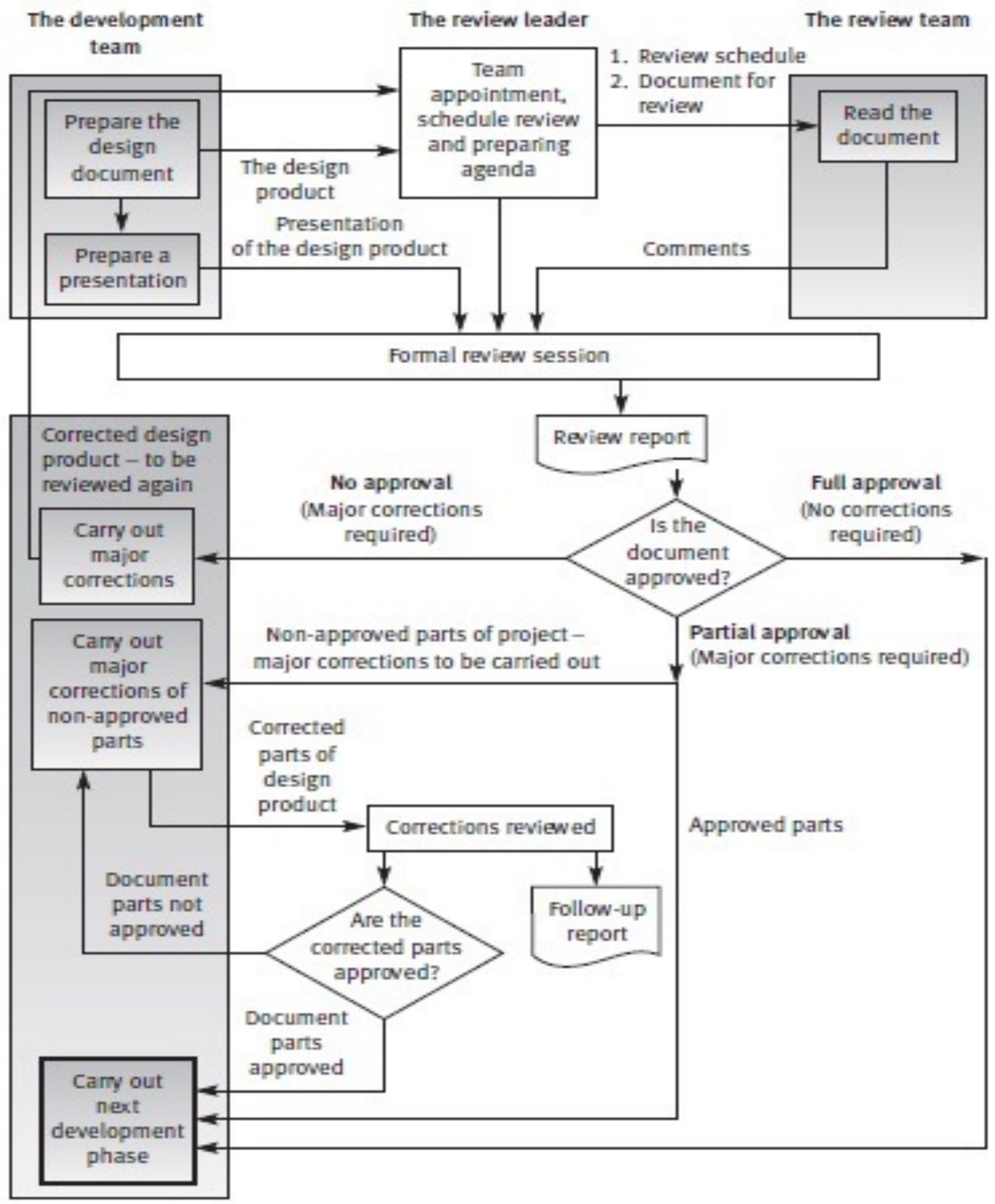
- Checklist
  - For: requirements, architecture, specification, design, code, test procedure
  - Contains: completeness, correctness, style, metrics, rules of constructions, multiple views
- example

Inspection Reporting Form

Issue no.	Line/Page	Checklist	Defect Category	Defect Type	. . .

Report Summary Form

	Major			Minor		
Defect Type	Missing	Wrong	Extra	Missing	Wrong	Extra
Interface						
Logic						
I/O						
…						
Functiona-lity						
Maintai-nability						


The development team
The review leader
The review team
Prepare the design document
Prepare a presentation
The design product
Team appointment, schedule review and preparing agenda
1. Review schedule
2. Document for review
Read the document
Presentation of the design product
Comments
Formal review session
Review report
No approval (Major corrections required)
Is the document approved?
Full approval (No corrections required)
Corrected design product – to be reviewed again
Carry out major corrections
Partial approval (Major corrections required)
Non-approved parts of project – major corrections to be carried out
Carry out major corrections of non-approved parts
Corrected parts of design product
Corrections reviewed
Approved parts
Follow-up report
Document parts not approved
Are the corrected parts approved?
Document parts approved
Carry out next development phase

Review process diagram (from Galin-SQA)

SQA?

- Inspections & walkthroughs – finds 30-70% of
  - logic design errors
  - coding errors
- Inspection - IBM reported an 83% defect detection rate

Inspection vs. Testing

Issues related to non-functional properties: Maintainability, evolvability, reusability

Properties difficult to test: Scalability, efficiency, security, integrity, robustness, reliability, exception handling

Artefacts: requirements, architecture, design documents (cannot "execute" as tests)

Inspection

***Benefits*:**

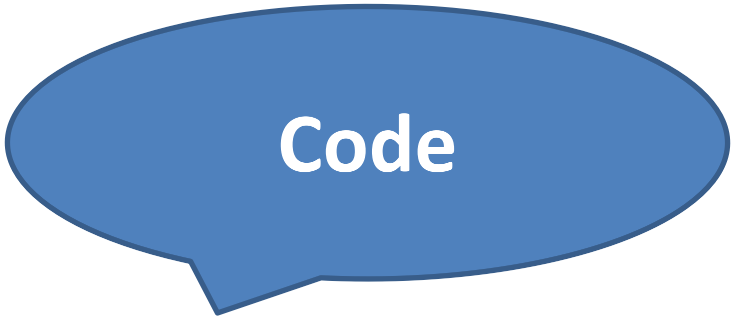
- Knowledge sharing
- Find flaws early
- Better communication: feedback

***Drawbacks*:**

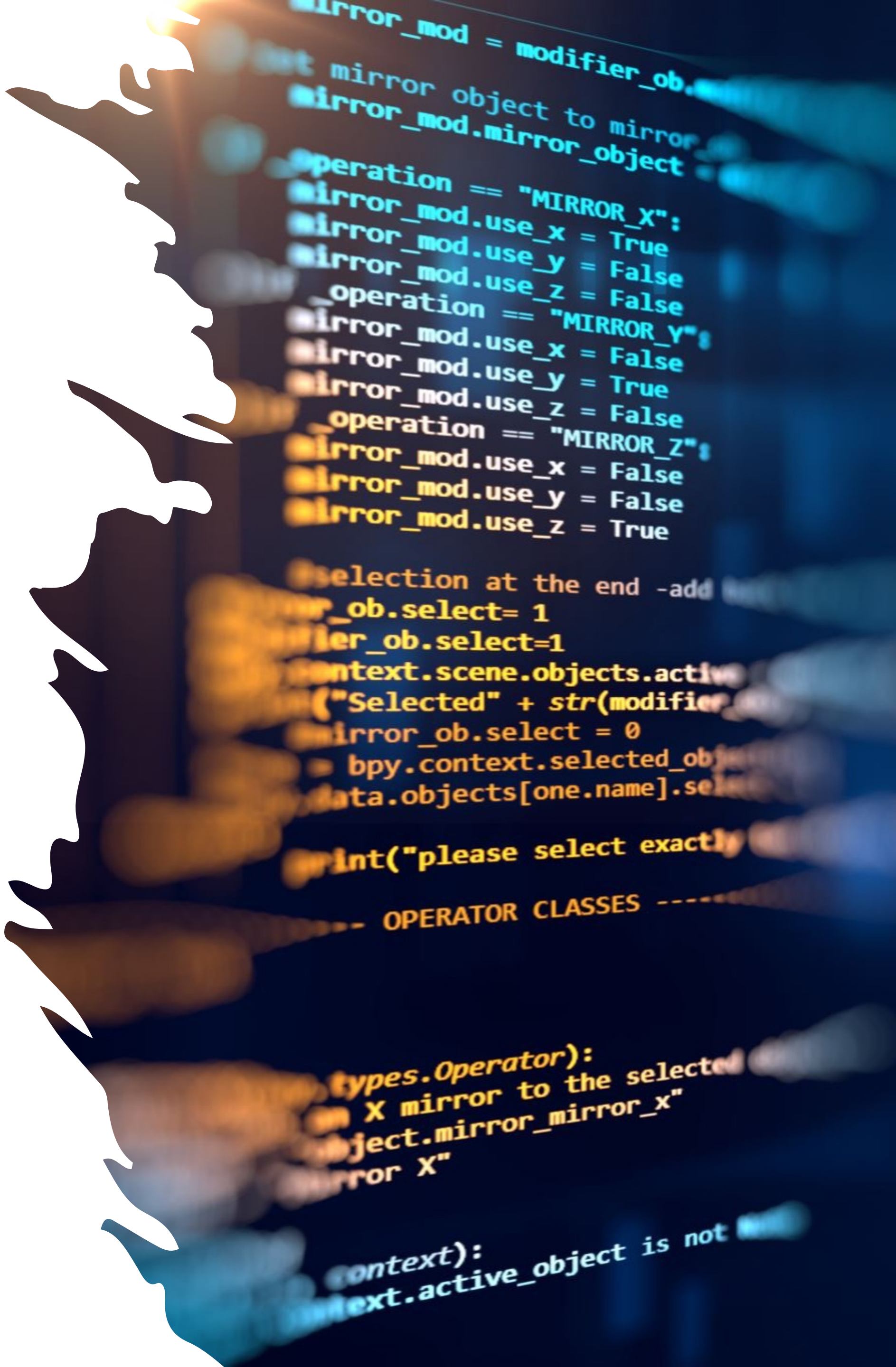
- Why fix? Why walkthrough code? / The reviewer will find it
- Used for HR evaluation

Code review

- **Definition**: *an integral process of software development that helps identify bugs and defects before the testing phase*
- Human / automated



- Manual vs. Automated Code review?

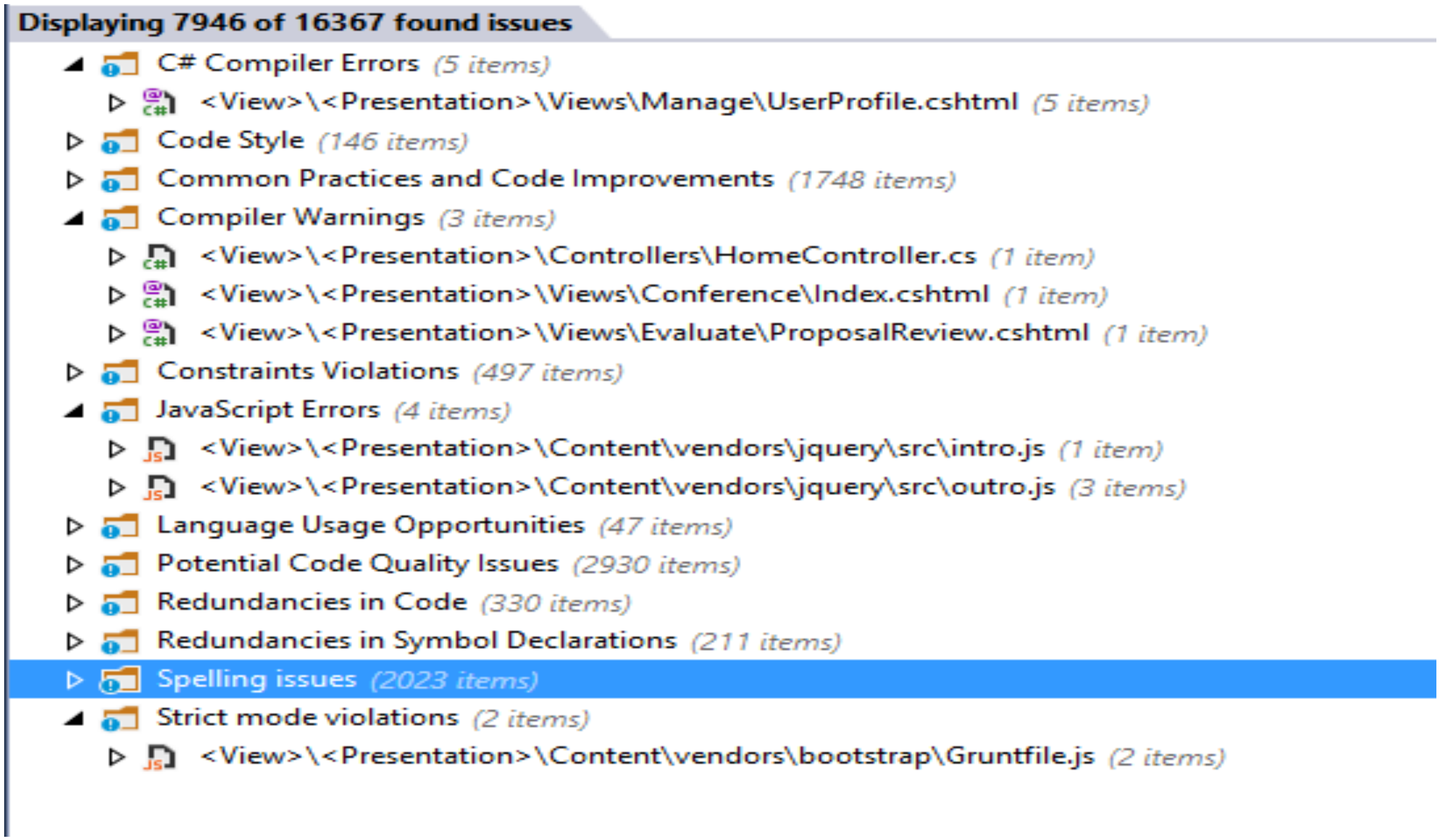


```
irror_mod = modifier_ob.
 mirror object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
 _operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
 _operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
 _ob.select= 1
 er_ob.select=1
 ntext.scene.objects.active
 ("Selected" + str(modifier
 irror_ob.select = 0
 = bpy.context.selected_ob
 ata.objects[one.name].se

 int("please select exactly

 - OPERATOR CLASSES ----

 types.Operator):
 X mirror to the selected
 ject.mirror_mirror_x"
 rror X"

 ontext):
 ext.active_object is not
```

Code review vs. inspection

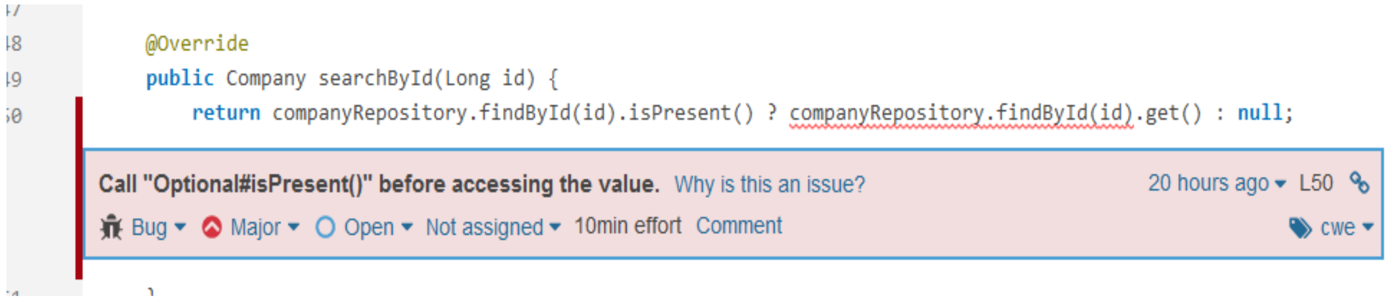
- No difference – some authors
- Inspection: issues not detected by code review
- Automated code review: no human feedback
- Inspection can use code review

ReSharper



```
Displaying 7946 of 16367 found issues
◢ C# Compiler Errors (5 items)
    ▷ <View>\<Presentation>\Views\Manage\UserProfile.cshtml (5 items)
▷ Code Style (146 items)
▷ Common Practices and Code Improvements (1748 items)
◢ Compiler Warnings (3 items)
    ▷ <View>\<Presentation>\Controllers\HomeController.cs (1 item)
    ▷ <View>\<Presentation>\Views\Conference\Index.cshtml (1 item)
    ▷ <View>\<Presentation>\Views\Evaluate\ProposalReview.cshtml (1 item)
▷ Constraints Violations (497 items)
◢ JavaScript Errors (4 items)
    ▷ <View>\<Presentation>\Content\vendors\jquery\src\intro.js (1 item)
    ▷ <View>\<Presentation>\Content\vendors\jquery\src\outro.js (3 items)
▷ Language Usage Opportunities (47 items)
▷ Potential Code Quality Issues (2930 items)
▷ Redundancies in Code (330 items)
▷ Redundancies in Symbol Declarations (211 items)
▷ Spelling issues (2023 items)
◢ Strict mode violations (2 items)
    ▷ <View>\<Presentation>\Content\vendors\bootstrap\Gruntfile.js (2 items)
```

SonarQube

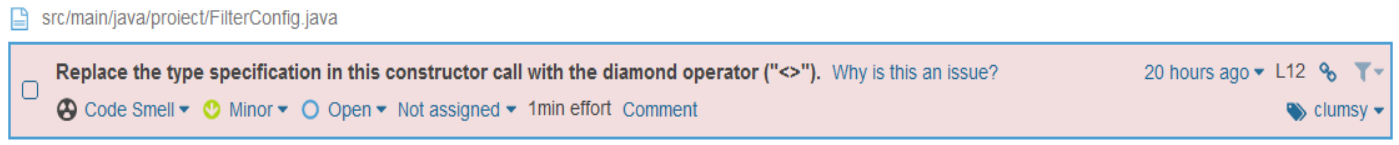


```
    @Override
    public Company searchById(Long id) {
        return companyRepository.findById(id).isPresent() ? companyRepository.findById(id).get() : null;
```

Call "Optional#isPresent()" before accessing the value. Why is this an issue? 20 hours ago L50

Bug Major Open Not assigned 10min effort Comment cwe

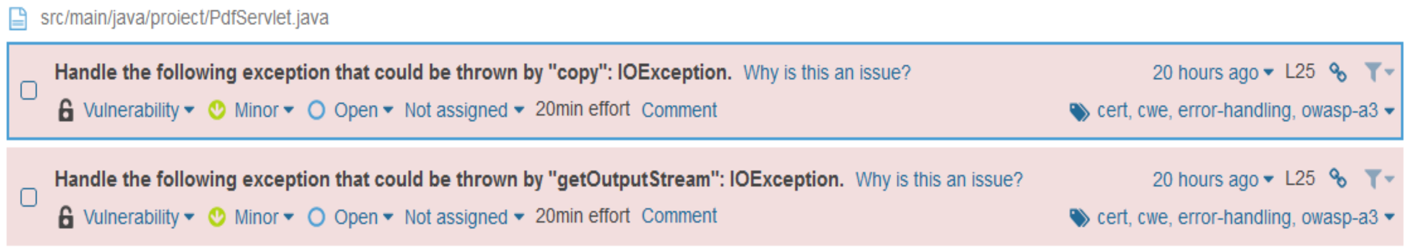
```
    }
```



src/main/java/proiect/FilterConfig.java

Replace the type specification in this constructor call with the diamond operator ("<>"). Why is this an issue? 20 hours ago L12

Code Smell Minor Open Not assigned 1min effort Comment clumsy



src/main/java/proiect/PdfServlet.java

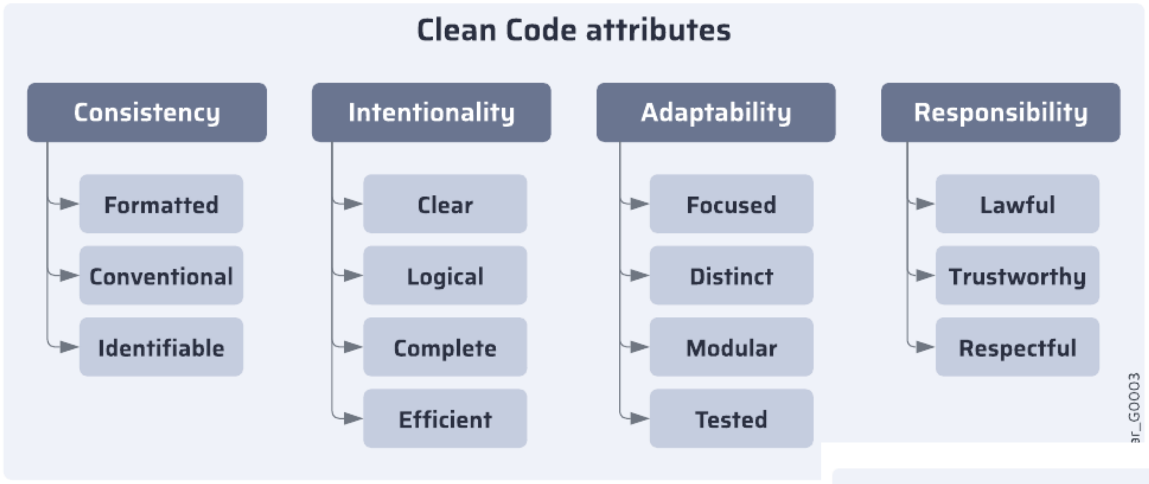
Handle the following exception that could be thrown by "copy": IOException. Why is this an issue? 20 hours ago L25

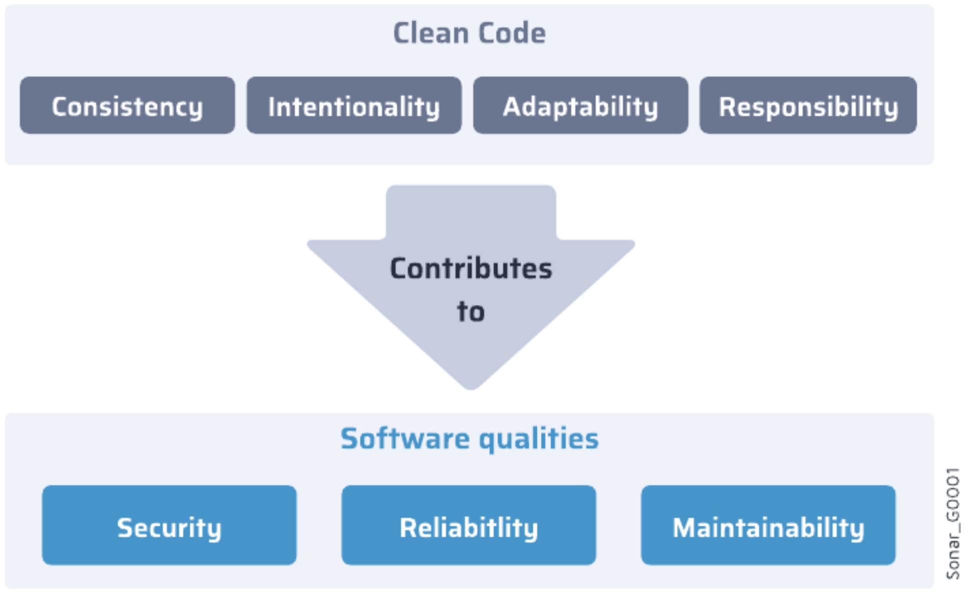
Vulnerability Minor Open Not assigned 20min effort Comment cert, cwe, error-handling, owasp-a3

Handle the following exception that could be thrown by "getOutputStream": IOException. Why is this an issue? 20 hours ago L25

Vulnerability Minor Open Not assigned 20min effort Comment cert, cwe, error-handling, owasp-a3

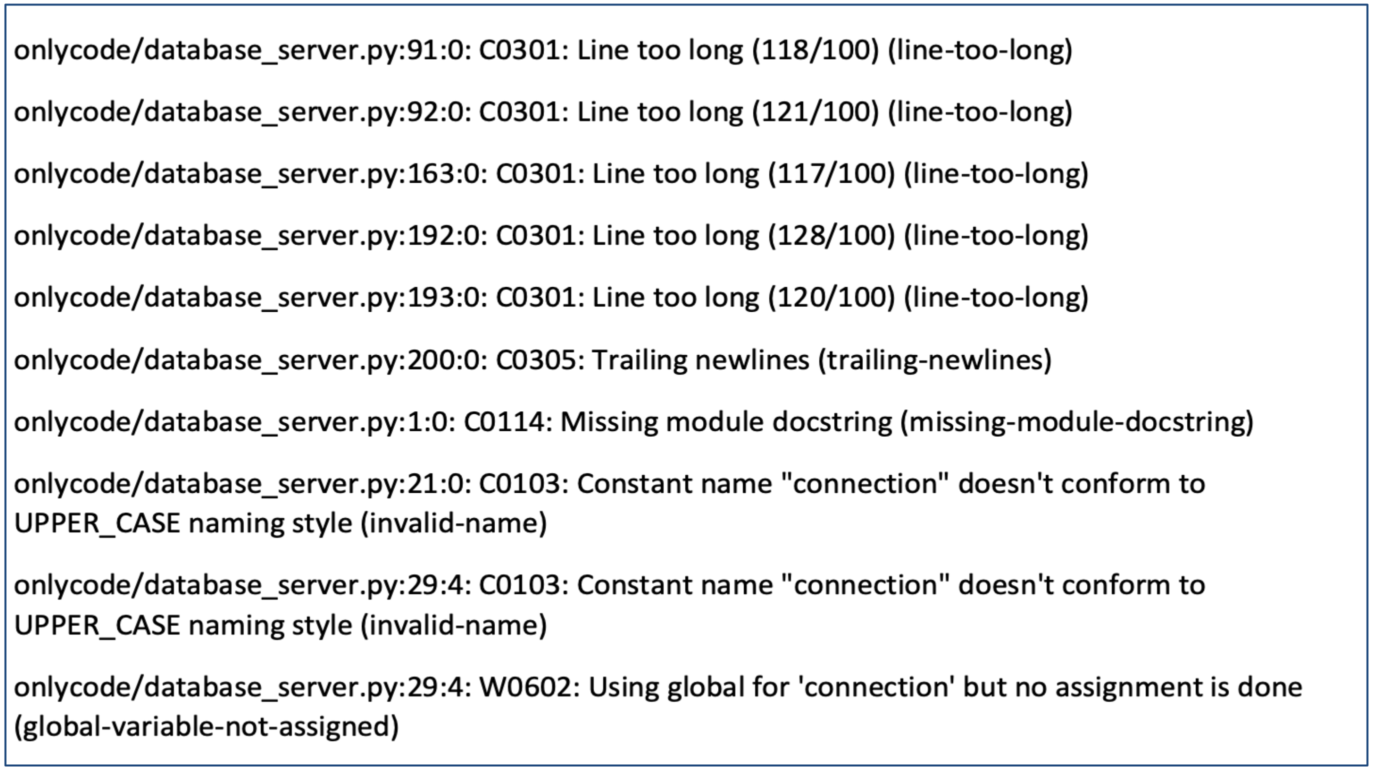
SonarQube


Clean Code attributes
Consistency
Formatted
Conventional
Identifiable
Intentionality
Clear
Logical
Complete
Efficient
Adaptability
Focused
Distinct
Modular
Tested
Responsibility
Lawful
Trustworthy
Respectful
ar_G0003


Clean Code
Consistency
Intentionality
Adaptability
Responsibility
Contributes to
Software qualities
Security
Reliabitlity
Maintainability
Sonar_G0001

Lint(er) tool

- Finds errors, bugs, conventions
- First lint(er) – Unix utility tool for C
- Pylint



```
onlycode/database_server.py:91:0: C0301: Line too long (118/100) (line-too-long)
onlycode/database_server.py:92:0: C0301: Line too long (121/100) (line-too-long)
onlycode/database_server.py:163:0: C0301: Line too long (117/100) (line-too-long)
onlycode/database_server.py:192:0: C0301: Line too long (128/100) (line-too-long)
onlycode/database_server.py:193:0: C0301: Line too long (120/100) (line-too-long)
onlycode/database_server.py:200:0: C0305: Trailing newlines (trailing-newlines)
onlycode/database_server.py:1:0: C0114: Missing module docstring (missing-module-docstring)
onlycode/database_server.py:21:0: C0103: Constant name "connection" doesn't conform to
UPPER_CASE naming style (invalid-name)
onlycode/database_server.py:29:4: C0103: Constant name "connection" doesn't conform to
UPPER_CASE naming style (invalid-name)
onlycode/database_server.py:29:4: W0602: Using global for 'connection' but no assignment is done
(global-variable-not-assigned)
```

3rd generation code review/ static analysis tools

- Analysis performed on AST (Abstract Syntax Tree)
- Detect vulnerabilities
- Rules associated with different quality factors

Case study

Code review tool:

- Resharper
- PMD
- SonarQube
- Pylint

Answer following questions:

- ”best” (good) practices implemented?
- Which SQ factors are investigated?