

SOFTWARE TESTING

CO3015 / CO5252

CH2. SOFTWARE TECHNICAL REVIEW

*types.Operator):
X mirror to the selected
object.mirror_mirror_x"
for X"*

Content

- ▶ Software Technical Review
- ▶ Inspections and Walkthroughs
- ▶ An Error Checklist for Inspections
- ▶ Java Code Review Checklist Samples

Technical Review?

https://en.wikipedia.org/wiki/Software_technical_review

- ▶ A software technical review is a form of peer review in which "a team of qualified personnel ... examines the suitability of the software product for its intended use and identifies discrepancies from specifications and standards. Technical reviews may also provide recommendations of alternatives and examination of various alternatives" (IEEE Std. 1028-1997, IEEE Standard for Software Reviews, clause 3.7).[1]
- ▶ "Software product" normally refers to some kind of technical document. This might be a software design document or program source code, but use cases, business process definitions, test case specifications, and a variety of other technical documentation, may also be subject to technical review.
- ▶ Technical review differs from software walkthroughs in its specific focus on the technical quality of the product reviewed. It differs from software inspection in its ability to suggest direct alterations to the product reviewed, and its lack of a direct focus on training and process improvement.

Technical Review?

<http://tryqa.com/what-is-technical-review-in-software-testing/>

- ▶ The goals of the technical review are:
 - ▶ To ensure that at an early stage the technical concepts are used correctly
 - ▶ To assess the value of technical concepts and alternatives in the product
 - ▶ To have consistency in the use and representation of technical concepts
 - ▶ To inform participants about the technical content of the document

Technical Review?

<https://www.softwaretestinggenius.com/understanding-software-technical-reviews-strs/>

- ▶ A review process can be defined as a critical evaluation of an object. It includes techniques such as walkthroughs, inspections and audits. Most of these approaches involve a group meeting to assess a work product.

- ▶ Software technical reviews can be used to examine all the products of the software evolution process. In particular, they are especially applicable and necessary for those products not yet in machine-processable form, such as requirements or specifications written in natural language.

Inspections and Walkthroughs

Chapter 22

Software Technical Reviews



An Error Checklist for (Code) Inspections

- ▶ Data Reference Errors
- ▶ Data Declaration Errors
- ▶ Computation Errors
- ▶ Comparison Errors
- ▶ Control-Flow Errors
- ▶ Interface Errors
- ▶ Input/Output Errors
- ▶ Other Checks

An Error Checklist for Inspections

Data Reference Errors

- ▶ 1. Unset variable used?
- ▶ 2. Subscripts within bounds?
- ▶ 3. Non-integer subscripts?
- ▶ 4. Dangling references?
- ▶ 5. Correct attributes when aliasing?
- ▶ 6. Record and structure attributes match?
- ▶ 7. Computing addresses of bit strings? Passing bit-string arguments?
- ▶ 8. Based storage attributes correct?
- ▶ 9. Structure definitions match across procedures?
- ▶ 10. Off-by-one errors in indexing or subscripting operations?
- ▶ 11. Inheritance requirements met?

An Error Checklist for Inspections

Data Declaration Errors

- ▶ 1. All variables declared?
- ▶ 2. Default attributes understood?
- ▶ 3. Arrays and strings initialized properly?
- ▶ 4. Correct lengths, types, and storage classes assigned?
- ▶ 5. Initialization consistent with storage class?
- ▶ 6. Any variables with similar names?

An Error Checklist for Inspections

Computation Errors

- ▶ 1. Computations on nonarithmetic variables?
- ▶ 2. Mixed-mode computations?
- ▶ 3. Computations on variables of different lengths?
- ▶ 4. Target size less than size of assigned value?
- ▶ 5. Intermediate result overflow or underflow?
- ▶ 6. Division by zero?
- ▶ 7. Base-2 inaccuracies?
- ▶ 8. Variable's value outside of meaningful range?
- ▶ 9. Operator precedence understood?
- ▶ 10. Integer divisions correct?

An Error Checklist for Inspections

Comparison Errors

- ▶ 1. Comparisons between inconsistent variables?
- ▶ 2. Mixed-mode comparisons?
- ▶ 3. Comparison relationships correct?
- ▶ 4. Boolean expressions correct?
- ▶ 5. Comparison and Boolean expressions mixed?
- ▶ 6. Comparisons of base-2 fractional values?
- ▶ 7. Operator precedence understood?
- ▶ 8. Compiler evaluation of Boolean expressions understood?

An Error Checklist for Inspections

Control-Flow Errors

- ▶ 1. Multiway branches exceeded?
- ▶ 2. Will each loop terminate?
- ▶ 3. Will program terminate?
- ▶ 4. Any loop bypasses because of entry conditions?
- ▶ 5. Possible loop fall-throughs correct?
- ▶ 6. Off-by-one iteration errors?
- ▶ 7. DO/END statements match?
- ▶ 8. Any non-exhaustive decisions?
- ▶ 9. Any textual or grammatical errors in output information?

An Error Checklist for Inspections

Interface Errors

- ▶ 1. Number of input parameters equal to number of arguments?
- ▶ 2. Parameter and argument attributes match
- ▶ 3. Parameter and argument units system match?
- ▶ 4. Number of arguments transmitted to called modules equal to number of parameters?
- ▶ 5. Attributes of arguments transmitted to called modules equal to attributes of parameters?
- ▶ 6. Units system of arguments transmitted to called modules equal to units system of parameters?
- ▶ 7. Number, attributes, and order of arguments to built-in functions correct?
- ▶ 8. Any references to parameters not associated with current point of entry?
- ▶ 9. Input-only arguments altered?
- ▶ 10. Global variable definitions consistent across modules?
- ▶ 11. Constants passed as arguments?

An Error Checklist for Inspections

Input/Output Errors

- ▶ 1. File attributes correct?
- ▶ 2. OPEN statements correct?
- ▶ 3. Format specification matches I/O statement?
- ▶ 4. Buffer size matches record size?
- ▶ 5. Files opened before use?
- ▶ 6. Files closed after use?
- ▶ 7. End-of-file conditions handled?
- ▶ 8. I/O errors handled?

An Error Checklist for Inspections

Other Checks

- ▶ 1. Any unreferenced variables in cross-reference listing?
- ▶ 2. Attribute list what was expected?
- ▶ 3. Any warning or informational messages?
- ▶ 4. Input checked for validity?
- ▶ 5. Missing function?

TABLE 3.1 Inspection Error Checklist Summary, Part I

Data Reference	Computation
1. Unset variable used?	1. Computations on nonarithmetic variables?
2. Subscripts within bounds?	2. Mixed-mode computations?
3. Noninteger subscripts?	3. Computations on variables of different lengths?
4. Dangling references?	4. Target size less than size of assigned value?
5. Correct attributes when aliasing?	5. Intermediate result overflow or underflow?
6. Record and structure attributes match?	6. Division by zero?
7. Computing addresses of bit strings? Passing bit-string arguments?	7. Base-2 inaccuracies?
8. Based storage attributes correct?	8. Variable's value outside of meaningful range?
9. Structure definitions match across procedures?	9. Operator precedence understood?
10. Off-by-one errors in indexing or subscripting operations?	10. Integer divisions correct?
11. Inheritance requirements met?	

Part I: Data Inspections

Data Declaration	Comparison
1. All variables declared?	1. Comparisons between inconsistent variables?
2. Default attributes understood?	2. Mixed-mode comparisons?
3. Arrays and strings initialized properly?	3. Comparison relationships correct?
4. Correct lengths, types, and storage classes assigned?	4. Boolean expressions correct?
5. Initialization consistent with storage class?	5. Comparison and Boolean expressions mixed?
6. Any variables with similar names?	6. Comparisons of base-2 fractional values?
	7. Operator precedence understood?
	8. Compiler evaluation of Boolean expressions understood?

TABLE 3.2 Inspection Error Checklist Summary, Part II

Control Flow	Input/Output
1. Multiway branches exceeded?	1. File attributes correct?
2. Will each loop terminate?	2. OPEN statements correct?
3. Will program terminate?	3. Format specification matches I/O statement?
4. Any loop bypasses because of entry conditions?	4. Buffer size matches record size?
5. Possible loop fall-throughs correct?	5. Files opened before use?
6. Off-by-one iteration errors?	6. Files closed after use?
7. DO/END statements match?	7. End-of-file conditions handled?
8. Any nonexhaustive decisions?	8. I/O errors handled?
9. Any textual or grammatical errors in output information?	

- ▶ Input/Output Errors
- ▶ Other Checks

Inspections

Interfaces	Other Checks
1. Number of input parameters equal to number of arguments? 2. Parameter and argument attributes match? 3. Parameter and argument units system match? 4. Number of arguments transmitted to called modules equal to number of parameters? 5. Attributes of arguments transmitted to called modules equal to attributes of parameters? 6. Units system of arguments transmitted to called modules equal to units system of parameters? 7. Number, attributes, and order of arguments to built-in functions correct? 8. Any references to parameters not associated with current point of entry? 9. Input-only arguments altered? 10. Global variable definitions consistent across modules? 11. Constants passed as arguments?	1. Any unreferenced variables in cross-reference listing? 2. Attribute list what was expected? 3. Any warning or informational messages? 4. Input checked for validity? 5. Missing function?

An Error Checklist for Inspections

- ▶ What else?
 - ▶ Comments?
 - ▶ Programming styles/standards?
 - ▶ ...?

Java Code Review Checklist Samples

- ▶ <http://users.csc.calpoly.edu/~jdalbey/301/Forms/CodeReviewChecklistJava.pdf>
- ▶ <https://www.scribd.com/document/253415083/Java-CheckList>
- ▶ <https://teaching.csse.uwa.edu.au/units/CITS5501//workshops/workshop06/checklist.pdf>
- ▶ https://courses.cs.washington.edu/courses/cse403/12wi/sections/12wi_code_review_checklist.pdf

Make your own one!?

Summary

- ▶ Software Technical Review
 - ▶ examines the suitability of the software product for its intended use and identifies discrepancies from specifications and standards
- ▶ Inspections and Walkthroughs
 - ▶ Walkthrough: more informal
 - ▶ Inspection: more formal
- ▶ An Error Checklist for Inspections
 - ▶ Data Reference / Declaration Errors
 - ▶ Computation / Comparison Errors
 - ▶ Control-Flow / Interface Errors
 - ▶ Input/Output Errors and Other Checks
- ▶ Java Code Review Checklist Samples