

Software Review

تهیه کننده: ملیحه هاشمی

AD hoc Reading

- زمانی که روش به خصوصی جهت شناسایی خطاها در artifact های مورد بررسی به مرور کنندگان ارائه نمی شود.
- یک روش بدون ساختار و غیرسیستماتیک می باشد.
- مرور کننده تلاش می کند تا به صورت شهودی براساس مهارت، دانش و تجربه خود، خطاهای بیشتری را شناسایی کند.
- بنابراین میزان موثر بودن این روش کاملاً وابسته به فرد مرور کننده می باشد.
- در حقیقت مدت زمان زیادی طول می کشد تا مرور کنندگان تجربه کافی در این زمینه را پیدا کنند و افرادی که تجربه کمی دارند معمولاً بهره‌وری پایینی در یافتن خطاها دارند. زمانی که هم افراد تجربه کافی در این زمینه را به دست می‌آورند به علت شهودی بودن این تجربه انتقال آن به دیگران دشوار است.
- با این وجود این روش یکی از مرسوم ترین روش‌های مرور می‌باشد.

Checklist-Based Reading

- Fagan در سال ۱۹۷۶ استفاده از چک لیست در حین انجام مرور را پیشنهاد کرده است و از آن زمان تاکنون به کارگیری چک لیست در این زمینه امری متداول بوده و چک لیست های متعددی در این رابطه ارائه شده است.
- چک لیست ها معمولاً براساس تحلیل defect های یافت شده در پروژه های مختلف به دست می آیند.
- در این روش نسبت به روش ad hoc پراکندگی نتایج کمتر است و نتایج کمتر به توانمندی های مرور کنندگان وابسته می باشد.
- نسبت به روش ad-hoc پوشش بیشتری در شناسایی خطاهای کلیدی دارد.
- با توجه به ماهیت چک لیست معمولاً خطاهای پیشین که براساس آن ها چک لیست شکل گرفته قابل شناسایی هستند و ممکن است خطاهای جدید و یا خطاهایی که وابسته به فهم عمیق نرم افزار و artifact مورد بررسی می باشند ناشناخته باقی بمانند.
- یک روش نیمه ساخت یافت به شمار می رود.

Checklist-Based Reading

A SAMPLE FOR REQUIREMENTS REVIEW

1. Requirements specifications shall be testable.
2. Requirements specifications shall not conflict with other requirements specifications.
3. Conditional requirements specifications shall cover all cases.
4. Numerical values in requirements specifications shall include physical units if applicable.

Checklist-Based Reading

A Checklist for Object-Oriented Code Reading

Section	Feature	Checklist item
For each class	Inheritance	Q1. Is all inheritance required by the design implemented in the class? Q2. Is the inheritance appropriate?
	Constructor	Q3. Are all instance variables initialized with meaningful values?
For each method	Data referencing	Q4. Are all parameters used within a method? Q5. Are the correct class constants used? Q6. Are indices of data structures operating within the correct boundaries?
	Object messaging	Q7. Is the correct method being called on the correct object? Q8. Are the correct values passed as parameters in the correct order?
	Method behavior	Q9. Are all assignment and state changes made correctly? Q10. For each return statement, is the value returned and its type correct? Q11. Does the method match the specification?
For each class	Method overriding	Q12. If inherited methods need to behave differently, are they overridden?

Scenario based Reading

➤ در سال ۱۹۹۴ توسط Votta و Porter ارائه گردید.

➤ این روش بعدها به عنوان defect based reading مطرح شد و روش scenario based reading برای گروهی از روش های مرور به کار رفت که از روش های مختلف برای به دست آوردن سناریوهای مرور استفاده می کنند.

➤ به این ترتیب Scenario based Reading شامل روش های گوناگونی است که موارد زیر از مهم ترین آن به شمار می روند:

➤ defect-based Reading

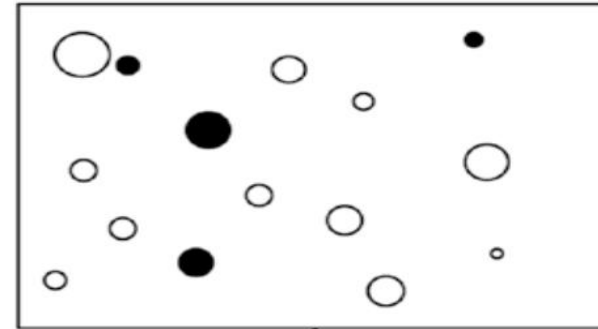
➤ perspective-based reading

➤ در روش مبتنی بر سناریو، سناریوهای عملیاتی طراحی میشوند که با ارائه دستورالعمل ها، سوالات و ... مرورگران را در رابطه با یک حوزه مشخص راهنمایی کنند و از این رو این روش سیستماتیک به شمار می رود. (مسئولیت های مشخص به مرورگران واگذار میشود.)

➤ به منظور کارایی بیشتر، همپوشانی سناریوها باید به میزان حداقل باشد.

Scenario based Reading

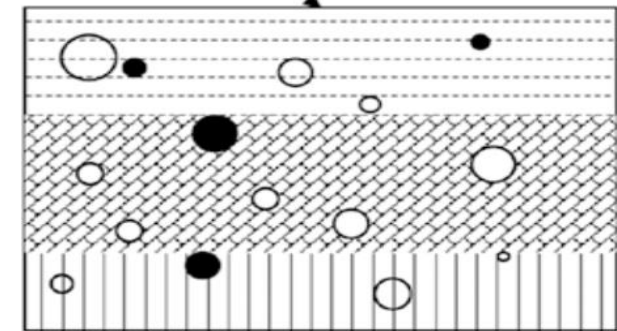
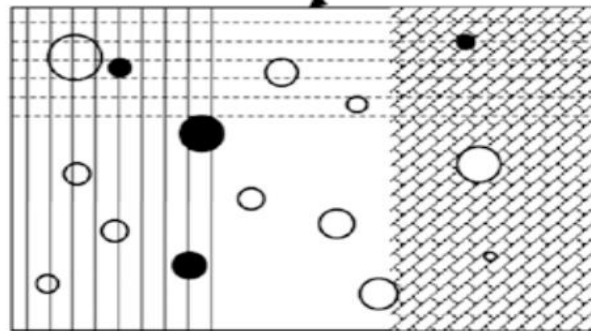
Software Requirement
Specification (SRS)



Unsystematic

Systematic

Lack of coordination
Overlap and gaps



Defect based Reading

◀ اولین روش در بین روش‌های مبتنی بر سناریو است و تمرکز آن بر شناسایی defect های به خصوص است. در این روش هر یک از افراد تیم مرور بر مبنای یک روش قاعده‌مند به جستجوی کلاس به خصوصی از defect ها می‌پردازد به نحوی که در نهایت نسبت به روش ad-hoc و روش مبتنی بر چک‌لیست defect ها به طور موثری شناسایی شوند.

◀ در این روش هر مرورگر طی گام‌های مشخص به شناسایی کلاس‌های به خصوصی از defect ها می‌پردازد.

◀ نقش هر مرورگر مشخص است و دقیقاً تعریف شده است.

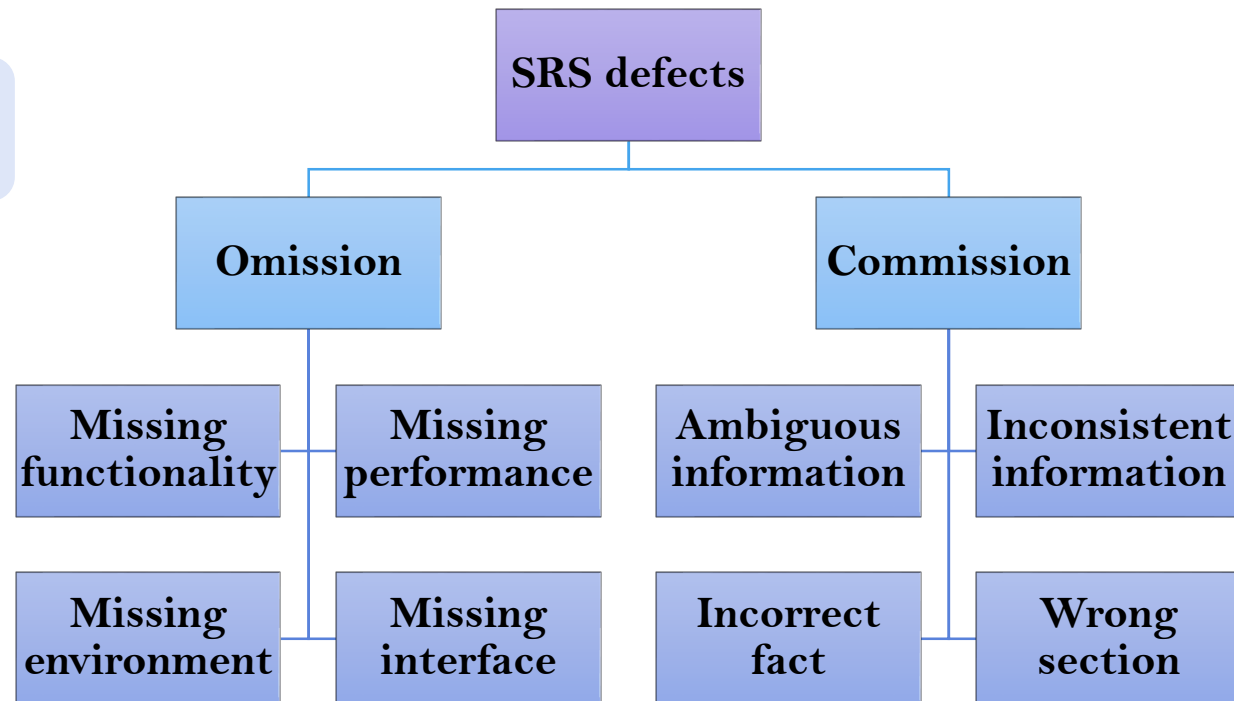
Defect based Reading

defect های موجود در SRS در دو دسته کلی تقسیم بندی می شوند:

➤ Omission : زمانی اتفاق می افتد که بخشی از اطلاعات مورد نیاز در SRS موجود نمی باشد.

➤ Commission : زمانی اتفاق می افتد که اطلاعات ناصحیح، افزونه، مبهم یا متناقض در SRS موجود باشد.

Defect Taxonomy



Defect based Reading

- ◀ Missing functionality: اطلاعاتی که رفتار عملیاتی سیستم را توصیف می‌کنند در SRS موجود نمی‌باشد.
- ◀ Missing performance: اطلاعاتی که توصیف کننده‌ی کارایی موردنظر سیستم هستند در SRS نیامده و یا به نحوی بیان شده است که برای تست پذیرش مورد قبول نیست.
- ◀ Missing environment: اطلاعاتی که توصیف کننده‌ی سخت‌افزار، نرم‌افزار، پایگاه‌داده و پرنسل محیطی هستند که در آن سیستم اجرا خواهد شد در SRS موجود نمی‌باشد.
- ◀ Missing interface: اطلاعاتی که توصیف کننده‌ی نحوه‌ی تعامل سیستم با اشیای خارج از محدوده‌ی آن هستند در SRS موجود نمی‌باشد.

Defect based Reading

- ◀ **Ambiguous information:** واژگان، عبارات و یا جملاتی که برای فهم عملکرد سیستم الزامی هستند تعریف نشده اند یا به طریقی تعریف شده اند که چندین برداشت می توان از آن کرد.
- ◀ **Inconsistent information:** اطلاعاتی که در بخش های مختلف SRS آمده یکدیگر را نقض می کنند یا بیان کننده ی اقداماتی هستند که هم زمان نمی توانند درست بوده و یا با هم انجام شوند.
- ◀ **Incorrect fact:** توصیفی که در SRS آمده است نمی تواند در زمینه موردنظر صحیح باشد.
- ◀ **Wrong section:** SRS به درستی سازماندهی نشده است و یا اطلاعات در بخش های نادرست قرار گرفته اند.

Defect based Reading

Incorrect Functionality Scenario

- 1. For each functional requirement, identify all input and output data objects.**
 - a) Are all values written to each output data object consistent with its intended function ?
 - b) Identify at least one function that uses each output data object.
- 2. For each functional requirement, identify all specified system events.**
 - a) Is the specification of these events consistent with their intended interpretation?
- 3. develop an invariant for each system mode or state, i.e., under what conditions must the system exit or remain in a given mode.**
 - a) Can the system's initial conditions fail to satisfy the initial mode's invariant?
 - b) Identify a sequence of events that allows the system to enter a mode without satisfying the mode's invariant.
 - c) Identify a sequence of events that allows the system to enter a mode but never leave.

Perspective Based Reading

این روش در سال ۱۹۹۶ توسط Basili و همکارانش ارائه گردید.

در این روش هر مرورگر، artifact مورد بررسی را با سناریوهایی که بر مبنای دیدگاه به خصوصی به دست آمده مرور می کند و در اثر تجمیع سناریوهای دیدگاه های متفاوت میتواند این artifact را به میزان مناسبی پوشش داد.

این روش ابتدا در رابطه با مستند نیازمندی به کار رفته و سپس در مورد طراحی، کد و واسط کاربری (خصوصا usability) مورد استفاده قرار گرفته است.

Perspective Based Reading

◀ قالب سناریو در این روش به صورت زیر است:

◀ مقدمه: در این بخش مشخص می شود که کدام ذی نفع و چه دیدگاهی در رابطه با artifact مورد نظر مطرح است.

◀ دستورالعمل: در این بخش راهنمایی هایی در مورد نحوه استخراج اطلاعات در مورد artifact مورد نظر ارائه میشود که باعث می شود مرورگر تنها بر روی اطلاعات مورد نیاز تمرکز کند.

◀ سوالات: مرورگر باید در حین دنبال کردن دستورالعمل ارائه شده به سوالات مطرح شده پاسخ دهد.

Perspective Based Reading

Document identification ◀

- software requirements specification
- software architectural or design description
- graphical user interface
- source code files
- a test plan including detailed test

Stakeholder identification ◀ در رابطه با هر artifact شناسایی شده باید ذی نفعان کلیدی آن شناسایی شوند.

Perspective understanding ◀ در رابطه با هر ذی نفع کلیدی شناسایی شده باید دیدگاه او از طریق مشخص نمودن اطلاعاتی و عملیاتی که برای این ذی نفع مهم است تعیین گردد.

Introduction development ◀

Instruction development ◀

Questions development ◀

Scenario review, testing, and updating ◀

Perspective Based Reading: Requirement specification

❖ ذی نفعان کلیدی مستند نیازمندی:

❖ کاربران نهایی

❖ آزمون گر (tester)

❖ طراح

برای هر یک از این
perspective ها باید
سناریوهایی طراحی شود.

Perspectives



End user

نیازمندی ها باید صحیح و کامل باشد و کارکردهای مورد نیاز را در برگیرد.



Tester

نیازمندی ها باید قابل تست و بدون ابهام باشند تا براساس آن ها بتوان test case های مورد نیاز را تولید کرد.



Designer

نیازمندی ها باید دارای جزئیات کافی با دقت مورد نظر باشند تا طراح بتواند مولفه های اصلی نرم افزار را شناسایی و طراحی کند.

Perspective Based Reading: Requirement specification

REQUIREMENT READING SCENARIO FOR TESTERS

Introduction: perspective-based reading is built around the concept that customers of a document are at the best position to judge if the document meets their needs and quality expectation. different customer reads the document and detects defects from their respective perspective. **This reading scenario assumes a tester's perspective. From the tester's perspective, requirements must be testable and unambiguous.**

Instructions: For each requirement or functional specification, create a test case to verify that the implementation satisfies the requirement. Compose test cases into a test suite. While composing the test suite, answer questions below

Questions:

1. do you have all information to identify the item being tested and to identify your test criteria? Can you create reasonable test cases for each item based on the criteria?
2. is there another requirement for which you would create a similar test case but would get a contradictory result?
3. Can you be sure the test generated will output the correct value in the correct units?
4. are there other interpretations of the requirement that the implementer might make due to the particular description of the requirement? Will this impact the test case and its oracle?
5. does the requirement make sense based on your domain knowledge or from what is specified in the general description?

Perspective Based Reading: Design Specification

❖ ذی نفعان کلیدی طراحی نرم افزار عبارتند از:

Requirement engineer ❖

Designer ❖

Implementer ❖

Tester ❖

Maintainer ❖

	Requirements Engineer	Designer	Implementer	Tester	Maintainer
Operation schema	x	x		x	
Class diagram	x	x	x		
Collaboration diagram		x	x	x	x
Operation pseudocode			x	x	x
Data dictionary	x				

Perspective Based Reading: Design Specification

DESIGN READING SCENARIO FOR DESIGNERS

Introduction

Assume a designer's perspective when reading the design artifacts. A designer concerns the correctness and completeness of design diagrams with respect to analysis diagrams.

Instructions

Get familiar with the purpose of the system and the requirements (use cases). Locate the class diagrams, compare them with analysis diagrams, and make sure classes, attributes, methods, associations, constraints, and other abstract concepts are consistent. As design diagrams are a refinement of analysis diagrams, design diagrams sometimes contain additional elements. Locate collaboration diagrams and compare them to the corresponding operation schema, and make sure the messages, parameters and their types, and the resulting behaviors are consistent.

Perspective Based Reading: Design Specification

DESIGN READING SCENARIO FOR DESIGNERS

Questions While following the instruction, answer questions:

1. Is there anything in analysis documents but not in design documents?
2. Are the messages appropriate and consistent from the collaboration diagrams and operation schema?
3. Are the system start-up conditions clear and correct?
4. Are there corresponding sent messages in operation schema and collaboration diagrams?
5. Is the order of messages correct in the collaboration diagram?
6. For every change of attributes, objects, states, or links in the operation schema, is there a corresponding sent message in the collaboration diagram?
7. Are there any discrepancies of functionality described in the operation schema and in the collaboration diagram?

Perspective Based Reading: Source code

CODE READING SCENARIO FOR CODE ANALYSTS

Introduction: assume the code analyst reading perspective, who has to ensure the right functionalities are implemented in code.

Instructions: Locate the functions implemented in the code. use the stepwise abstraction reading technique if it is a structured code; use the abstraction-driven and use-case-driven reading techniques if it is object-oriented code. Start with the least-dependent code and read bottom-up to form a high-level abstraction of the coded function. For each function, check if the abstraction you derived matches the specification; if there is a deviation, decide if it is a defect and log your findings.

Questions: While following the instruction above, answer questions:

1. does the operation implemented in code match the one described in the specification?
2. is there any operation described in the specification, but not implemented in the code?
3. are data (constants, variables, etc) and their type used correctly?
4. is the calculation performed correctly?
5. is usage of the interface between different modules semantically correct?

Perspective-Based Usability Reading

EXPERT USE READING SCENARIOS

Introduction: assume you are an expert user. Your goal is to fill out the web form and submit it, which can be decomposed into multiple sub-goals in order. For each sub-goal, go through the following stages and check the questions. if your answer to a question is no, raise a usability issue.

Reading Instruction 1: Scan through instruction, objects, and actions in the user interface.

Question set 1:

1. is the text easy to read?
2. is the information organized such that the most important information can be read first?
3. are the more frequently selected items arranged on top in the list control?
4. is redundant information avoided or minimized?

Perspective-Based Usability Reading

EXPERT USE READING SCENARIOS

Introduction: assume you are an expert user. Your goal is to fill out the web form and submit it, which can be decomposed into multiple sub-goals in order. For each sub-goal, go through the following stages and check the questions. if your answer to a question is no, raise a usability issue.

Reading Instruction 2: execute the actions for achieving the sub-goal, using shortcuts whenever available. For each action, check the following questions.

Question set 1:

1. are common shortcuts available, e.g., to allow user to step to the next text field via keyboard?
2. are possible default values provided?
3. does the system compute and remember information for the user whenever possible?
4. Can the user make a selection by clicking on a larger area associated with the object to be selected?
5. are unproductive activities avoided or minimized, e.g., navigation, mouse movement, hand movement, eye movement, etc.?
6. are stressful actions avoided or minimized, e.g., clicking mouse multiple times in a short period of time, clicking on a small object to make a selection, etc.?
7. is the text easy to read?

Perspective-Based Usability Reading

EXPERT USE READING SCENARIOS

Introduction: assume you are an expert user. Your goal is to fill out the web form and submit it, which can be decomposed into multiple sub-goals in order. For each sub-goal, go through the following stages and check the questions. if your answer to a question is no, raise a usability issue.

Reading Instruction 3: Wait for system to respond, if necessary.

Question set 3:

1. does each user action immediately generate perceivable results in the user interface?

General question: in addition to the specific questions in question set 3, consider this high-level question: Can the web form be redesigned to reduce the user's unproductive activities?

Traceability based Reading

Sequence Diagram vs Class Diagrams ◀

State Diagrams vs Class Diagrams ◀

Sequence Diagrams vs State Diagrams ◀

Sequence Diagrams vs Use Cases ◀

State diagram vs Class Diagram ◀

Stepwise Abstraction

- یک روش bottom-up که در آن یک توضیح انتزاعی برای یک بخش از کد به دست می‌آید و روند آن به صورت زیر می‌باشد:
- کد مورد بررسی خط به خط خوانده شده و به قسمت‌های مختلف تقسیم می‌شود.
- برای هر قسمت یک توضیح abstract نوشته می‌شود.
- این توضیحات براساس ارتباط بین قسمت‌های مختلف کد به یکدیگر مرتبط می‌شوند تا در نهایت یک تصویر کلی از عملکرد کد به دست آید.
- این تصویر کلی با توصیف نیازمندی‌ها مقایسه می‌گردد.