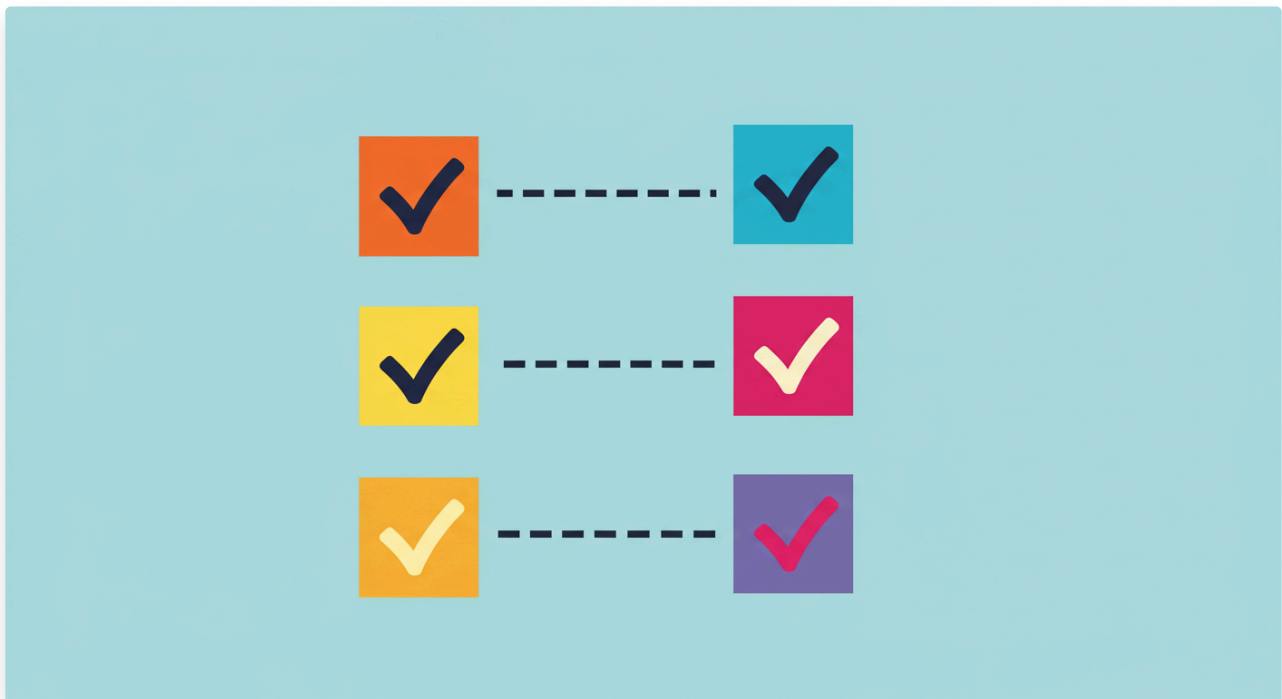


Evaluating Program Designs

Textbook

Evaluating Program Designs



When looking at software, we mainly consider two things: readability (how easy its code is for people to understand) and usability (how easy and helpful the program is for those who use it).

Readability

Readable code is super important because it helps maintain software, allows programmers to work together easily, and makes fixing bugs quicker. To check if code is readable, look for:

- **Clear comments:** Are there notes explaining what the code does?
- **Meaningful names:** Are variables and functions named in a way that makes sense (e.g., `calculateTotal` instead of `ct`)?
- **Consistent formatting:** Is the code neatly organized with proper indentation and spaces?
- **Modularity:** Is the code broken down into smaller, specific functions or sections?
- **Simplicity:** Is the code straightforward, or does it seem overly complicated?

You can check readability in various programs: open-source apps (like a free calculator), code from classmates in a group project, or even a company's software to see how easy it would be for a new developer to understand.

Usability

Usability focuses on the **user experience (UX)**. It's about making sure a program is easy to learn, efficient, and enjoyable to use. Good usability means more people will use the program, complete tasks quickly, and be happy with it. To evaluate usability, think about:

- **Learnability:** How easy is it for new users to figure out?
- **Efficiency:** How quickly can tasks be completed?
- **Memorability:** How easy is it to remember how to use the program after not using it for a while?
- **Error prevention and recovery:** How well does the program stop errors from happening, and how easy is it to fix things if an error occurs?
- **User satisfaction:** Do users enjoy using the program overall?

You can evaluate usability by looking at popular apps (like Instagram or Google Maps), programs made by your friends, or new tools you find online.

Efficiency and Effectiveness of Digital Tools

It's important to understand how well digital tools perform in real-world situations.

Efficiency measures how well a tool gets a job done with the least amount of wasted time, effort, or resources. It helps you accomplish tasks quickly. To evaluate efficiency, consider:

- **Time savings:** Does the tool save you time?
- **Resource consumption:** Does it use a lot of battery or data?
- **Automation:** Can it do things automatically for you?
- **Processing speed:** How fast does it perform tasks?

For example, a word processor with auto-save and spell check is **efficient** for writing a paper compared to doing it by hand. A search engine is super efficient for finding information. And collaborative document software is efficient for group projects because multiple people can work on it at the same time.

Effectiveness

Effectiveness measures how well a tool helps you achieve your goals or its intended purpose. It's about getting the *right* things done. To evaluate effectiveness, ask if the tool:

- Successfully helps complete the task.
- Produces accurate and reliable results.
- Improves the quality of your work.
- Actually solves the problem it's supposed to.

For example, a language learning app is **effective** for learning a new language. A budgeting app is effective for managing your money. Advanced presentation software is effective for creating professional presentations. These tools directly help you reach specific goals.

Key Takeaway: The best digital tools are **both efficient** (fast and easy to use) and **highly effective** (successfully achieve their purpose) for their specific tasks.

Exercise: Program Evaluation Challenge

This exercise will help you practice evaluating a real program's readability and usability.

Task: Choose one of the following programs (or a similar one approved by your instructor) to evaluate:

- **A simple open-source game:** (e.g., a basic Tic-Tac-Toe game, a simple platformer, or a classic arcade game clone available on GitHub or a similar platform).
- **A basic utility program:** (e.g., a simple calculator, a unit converter, or a to-do list application, preferably one with publicly available code).
- **A new web-based tool or app you haven't used before:** (e.g., a new online image editor, a collaborative brainstorming tool, or a simple data visualization website).

Instructions:

1. **Access the Program:** Get access to the program. If it's code, make sure you can see the source code. If it's a web tool or app, just open it in your browser.

2. **Evaluate Readability (if applicable - for code-based programs):** Spend 15-20 minutes looking at the program's code. Think about:

- Are there clear and helpful comments, or are they missing/confusing?
- Are variable and function names descriptive and easy to understand?
- Is the formatting consistent (indented, well-organized)?
- Is the code modular (broken into logical, smaller pieces like functions or classes)?
- Does any part of the code seem unnecessarily complicated?

After your examination, write a short paragraph (5-7 sentences) summarizing your findings on the code's readability. Provide specific examples from the code to support your points.

3. **Evaluate Usability:** Spend 15-20 minutes actively using the program as if you were a new user, trying to complete a few common tasks. Evaluate:

- **Learnability:** How easy was it to figure out how to use the program without instructions?
- **Efficiency:** How quickly could you complete a task once you understood it, and did it require too many steps?
- **Error handling:** Did you run into any errors? If so, was the error message clear, and was it easy to fix the problem?
- **Satisfaction:** Did you enjoy using the program? Was the interface pleasant and easy to navigate?

After your evaluation, write a short paragraph (5-7 sentences) summarizing your findings on the program's usability. Provide specific examples of what worked well or what was challenging.

4. **Overall Critique:** Write a concluding paragraph (3-5 sentences) that gives an overall review of the program. What are its biggest **strengths and weaknesses** in terms of both readability (if you checked it) and usability? If you could give one piece of advice to the developer, what would be your top suggestion for improvement?

Critical Thinking Questions

1. When a software development team works on a major update, why would prioritizing the ease with which **new programmers can understand the existing code** be just as important as how smoothly the program runs for the end-user? Think about the long-term health of the software.
2. Think about two different apps you use regularly: one you find incredibly helpful but sometimes frustrating to use, and another that is a joy to use but doesn't fully meet your needs. Which app do you consider **more successful overall**, and what does that tell you about the balance between a tool's ability to achieve its purpose (effectiveness) and its ease of use (usability)?
3. Imagine you're designing a new feature for a popular social media platform. What specific design choices related to **user experience (UX)** would you make to help users **avoid common mistakes**, and how would you ensure that if an error *does* occur, the user can easily fix it without becoming frustrated? Give concrete examples.

Questions (5)

1. A new programmer joins a team working on an existing calculator application. They need to add a new function for square roots. According to the passage, which aspect of the existing code would be most important for this new programmer to evaluate?

MULTIPLE CHOICE

Choose the correct answer:

- A. Its readability
- B. Its user satisfaction rating
- C. Its processing speed
- D. Its color scheme

2. A user is trying out a new online image editor. They find it difficult to locate the 'crop' tool, and the icons don't clearly indicate their functions. Which aspect of usability is primarily being challenged in this scenario?

MULTIPLE CHOICE

Choose the correct answer:

- A. Efficiency
- B. Memorability
- C. Error prevention
- D. Learnability

3. You are using a new online unit converter tool. You notice that it takes many clicks to perform a simple conversion, even after you know how to use it. Which aspect of the tool's efficiency is being negatively impacted?

MULTIPLE CHOICE

Choose the correct answer:

- A. Time savings
- B. Resource consumption
- C. Automation capabilities
- D. Processing speed

4. A language learning app helps users successfully achieve fluency in a new language. According to the passage, this app would be considered highly effective. What specifically defines a tool's effectiveness?

MULTIPLE CHOICE

Choose the correct answer:

- A. Its speed of task completion.
- B. How well it performs a task with minimal waste.
- C. How well it achieves its intended purpose or helps accomplish goals.
- D. Its ability to be used without any instructions.

5. A software development team is working on a major update. Why would prioritizing the ease with which new programmers can understand the existing code be just as important as how smoothly the program runs for the end-user?

MULTIPLE CHOICE

Choose the correct answer:

- A. User satisfaction is solely dependent on code readability.
- B. Only new programmers need to understand the code, not existing team members.
- C. Readable code ensures efficient maintenance, collaboration, and bug fixing for future development.
- D. Smooth program operation for the end-user is not a critical concern.

Answer Keys & Solutions

Questions

1. A new programmer joins a team working on an existing calculator application. They need to add a new function for square roots. According to the passage, which aspect of the existing code would be most important for this new programmer to evaluate?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---------------------------------|-------------|
| A. Its readability | ✓ Correct |
| B. Its user satisfaction rating | ✗ Incorrect |
| C. Its processing speed | ✗ Incorrect |
| D. Its color scheme | ✗ Incorrect |

Explanation:

Think about what helps a human understand written code.

2. A user is trying out a new online image editor. They find it difficult to locate the 'crop' tool, and the icons don't clearly indicate their functions. Which aspect of usability is primarily being challenged in this scenario?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---------------------|-------------|
| A. Efficiency | ✗ Incorrect |
| B. Memorability | ✗ Incorrect |
| C. Error prevention | ✗ Incorrect |
| D. Learnability | ✓ Correct |

Explanation:

Consider how easy it is for a new user to figure out how to use the program.

3. You are using a new online unit converter tool. You notice that it takes many clicks to perform a simple conversion, even after you know how to use it. Which aspect of the tool's efficiency is being negatively impacted?

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------------------------|-------------|
| A. Time savings | ✓ Correct |
| B. Resource consumption | ✗ Incorrect |
| C. Automation capabilities | ✗ Incorrect |
| D. Processing speed | ✗ Incorrect |

Explanation:

Consider how quickly a task can be completed with minimal effort.

4. A language learning app helps users successfully achieve fluency in a new language. According to the passage, this app would be considered highly effective. What specifically defines a tool's effectiveness?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---|-------------|
| A. Its speed of task completion. | ✗ Incorrect |
| B. How well it performs a task with minimal waste. | ✗ Incorrect |
| C. How well it achieves its intended purpose or helps accomplish goals. | ✓ Correct |
| D. Its ability to be used without any instructions. | ✗ Incorrect |

Explanation:

Focus on whether the tool actually helps you achieve your desired outcome.

5. A software development team is working on a major update. Why would prioritizing the ease with which new programmers can understand the existing code be just as important as how smoothly the program runs for the end-user?

MULTIPLE CHOICE

Correct Answer:

A. User satisfaction is solely dependent on code readability.

✗ Incorrect

B. Only new programmers need to understand the code, not existing team members.

✗ Incorrect

C. Readable code ensures efficient maintenance, collaboration, and bug fixing for future development.

✓ Correct

D. Smooth program operation for the end-user is not a critical concern.

✗ Incorrect

Explanation:

Think about the long-term implications of code quality for a development team.