

Debugging C#

Debugging Fundamentals



Neil Morrissey

Solutions Architect

linkedin.com/in/neilmorrissey | www.neilmorrissey.net

Debugging C#

Version Check



Version Check

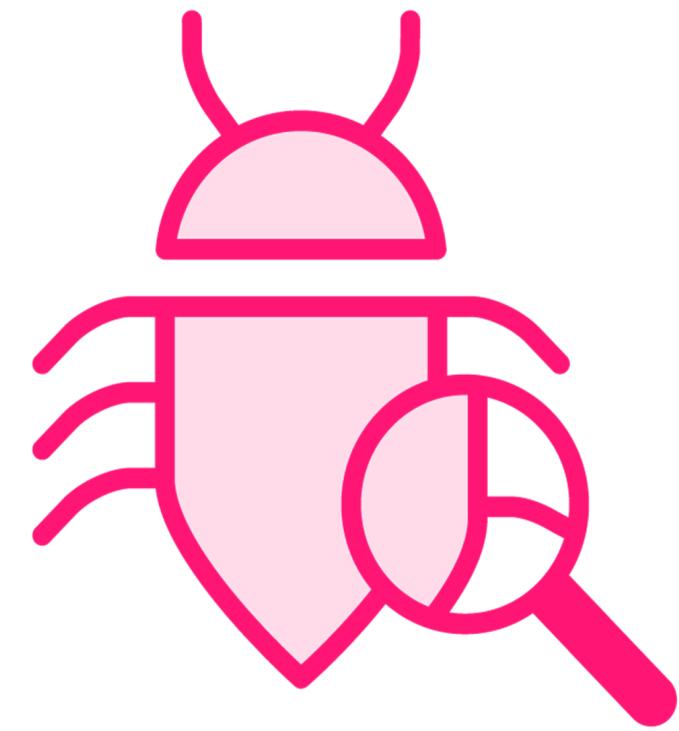


This version was created by using:

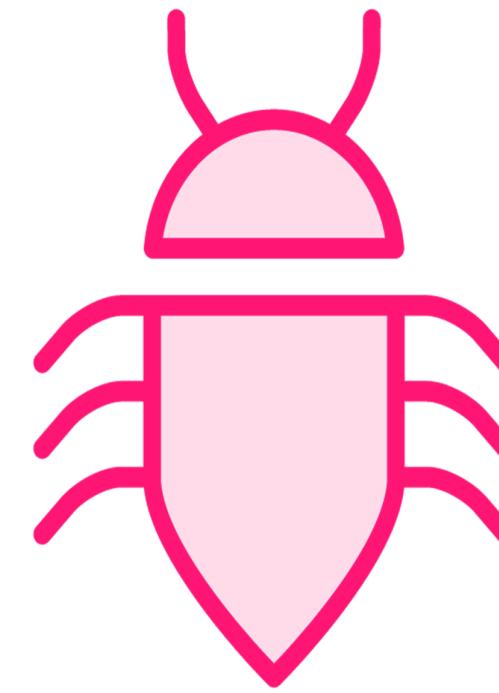
- C#12
- Visual Studio 2022 (Community Edition)
- Visual Studio Code v1.86.0



Definitions



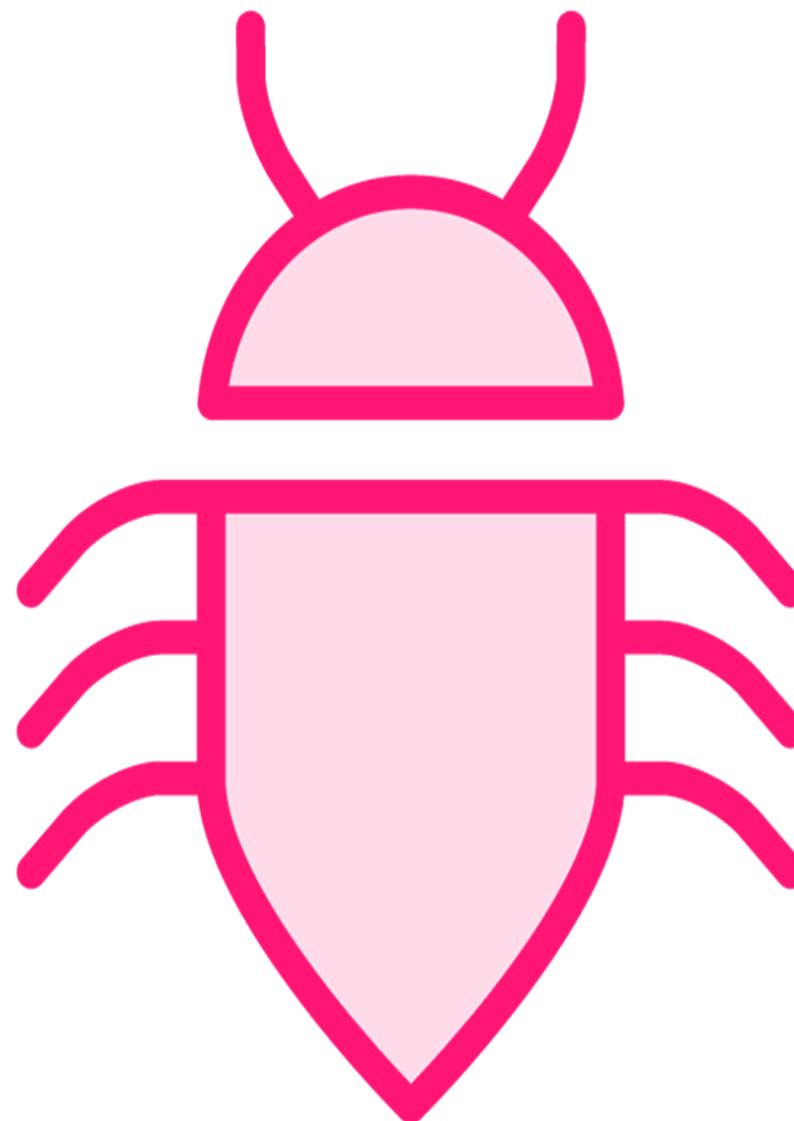
What is Debugging?



What is a Bug?



Software Bugs



Engineering term from the 1800's

1946 Mark II large-scale computer

Error traced to a Moth in a relay

A bug is a defect in code



Definitions

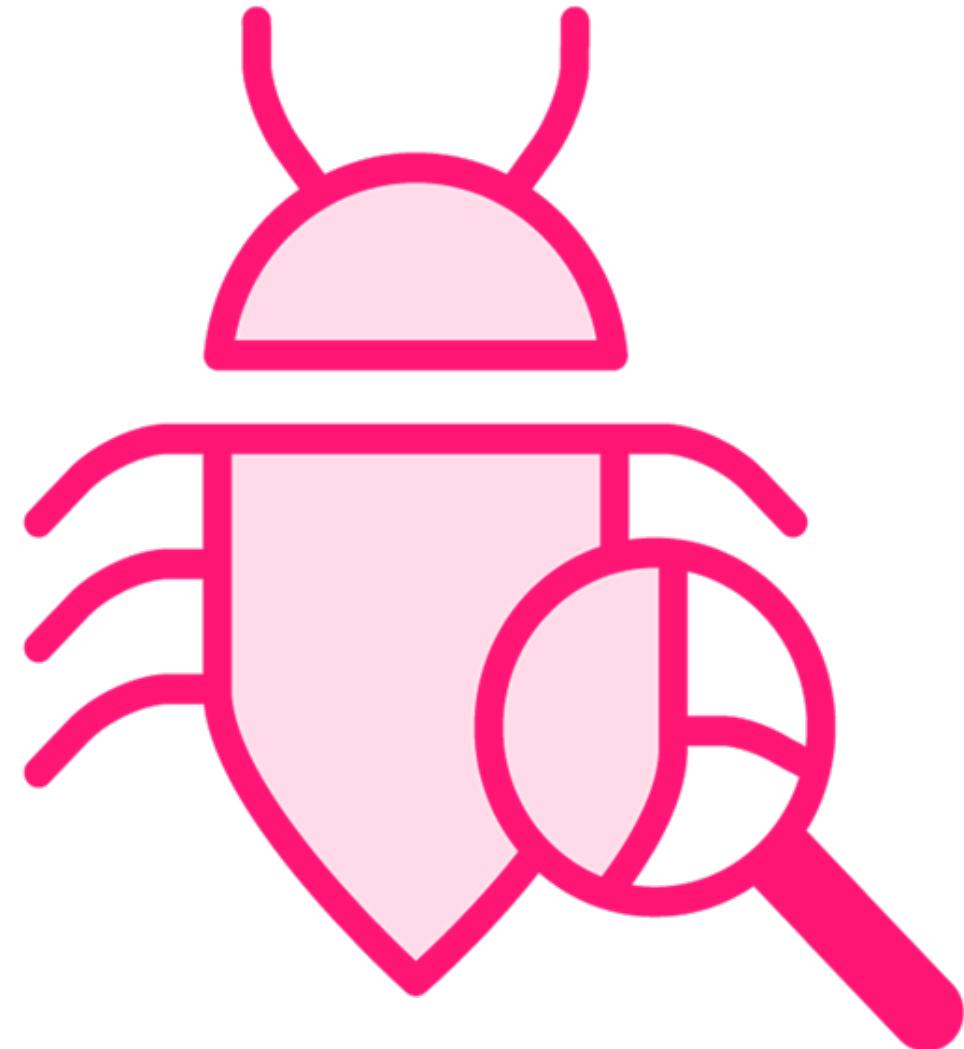
Debugging:

The process of identifying the cause of an error and correcting it

Testing:

The process of detecting errors during development





Debugging during development

Debugging production issues

- Recreating the issue in dev
- Pressure to fix the issue

Not everyone knows how to debug

- Developers vary in time it takes to debug and number of bugs found *

* Code Complete, McConnell, Steve, 2004 - Microsoft Press



Definitions

Debugging:

The process of identifying the cause of an error and correcting it

Testing:

The process of detecting errors during development

Refactoring:

The process of restructuring source code to improve its operation without altering its functionality



Definitions

Debugging:

The process of identifying the cause of an error and correcting it

Testing:

The process of detecting errors during development

Refactoring:

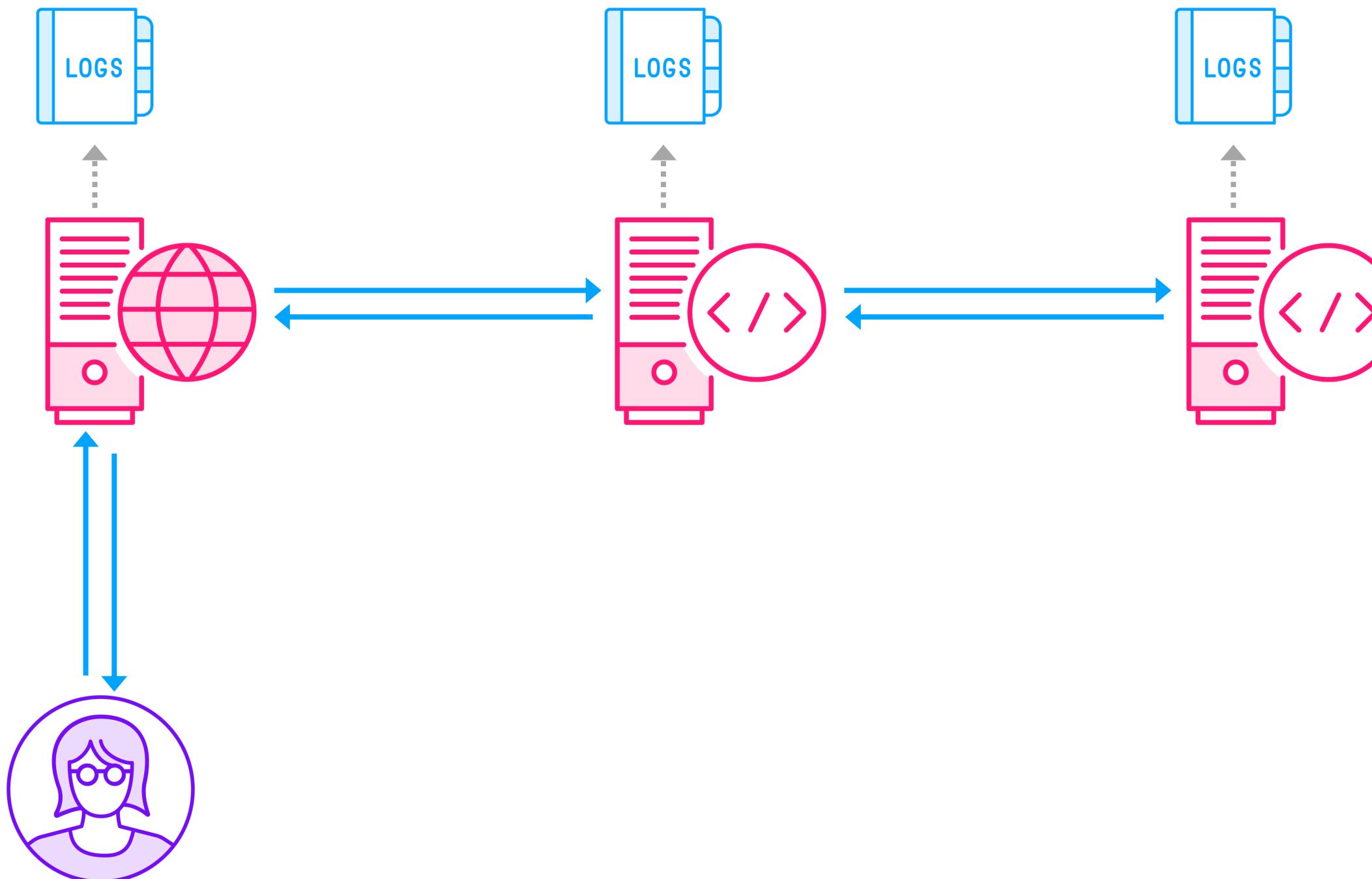
The process of restructuring source code to improve its operation without altering its functionality

Diagnostics:

Instrumenting applications to gather data about its internal operation, after it's been released



Diagnostics Instrumentation



Diagnostics



Code you add to reveal internal state of application

Can structure to enable/disable logging

Can structure to log certain types/levels

Can use as input to debugging – to recreate state of application when error occurred



More information on Diagnostics

.NET Diagnostics for Applications: Best Practices

Neil Morrissey



Module Overview

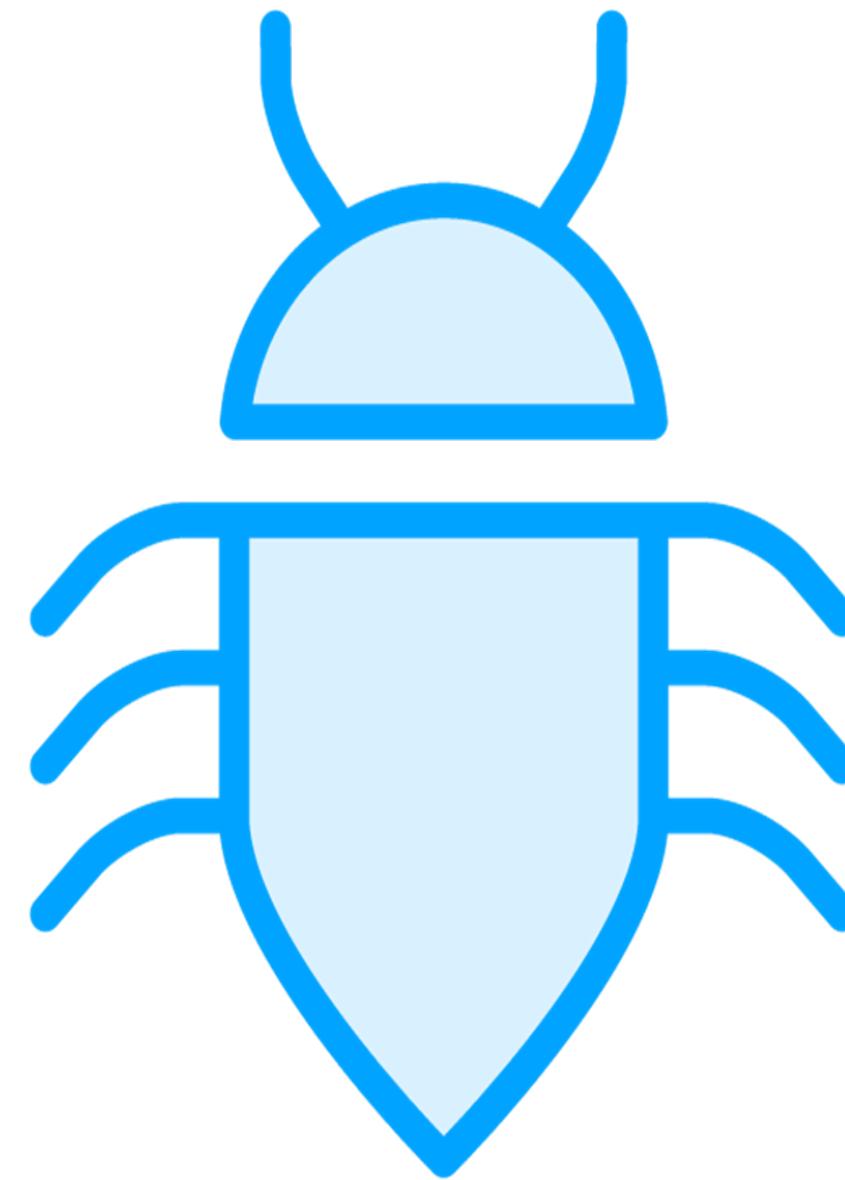


- Debugging definitions
- Categories of defects
- Debugger basics
- Debugging an enterprise application



Defect Categories





- Functional errors**
- Security defects**
- Syntax errors**
- Logic errors**
- Calculation errors**
- Out-of-bound bugs**
- Compatibility defects**
- System-level integration bugs**
- Performance defects**
- Usability defects**



Functional Errors



Behavior varies from the functional requirements

Often found during testing phase

Examples

- Save button not working
- Form field not being validated
- Multiple records getting created



Security Defects



Open your software to attacks

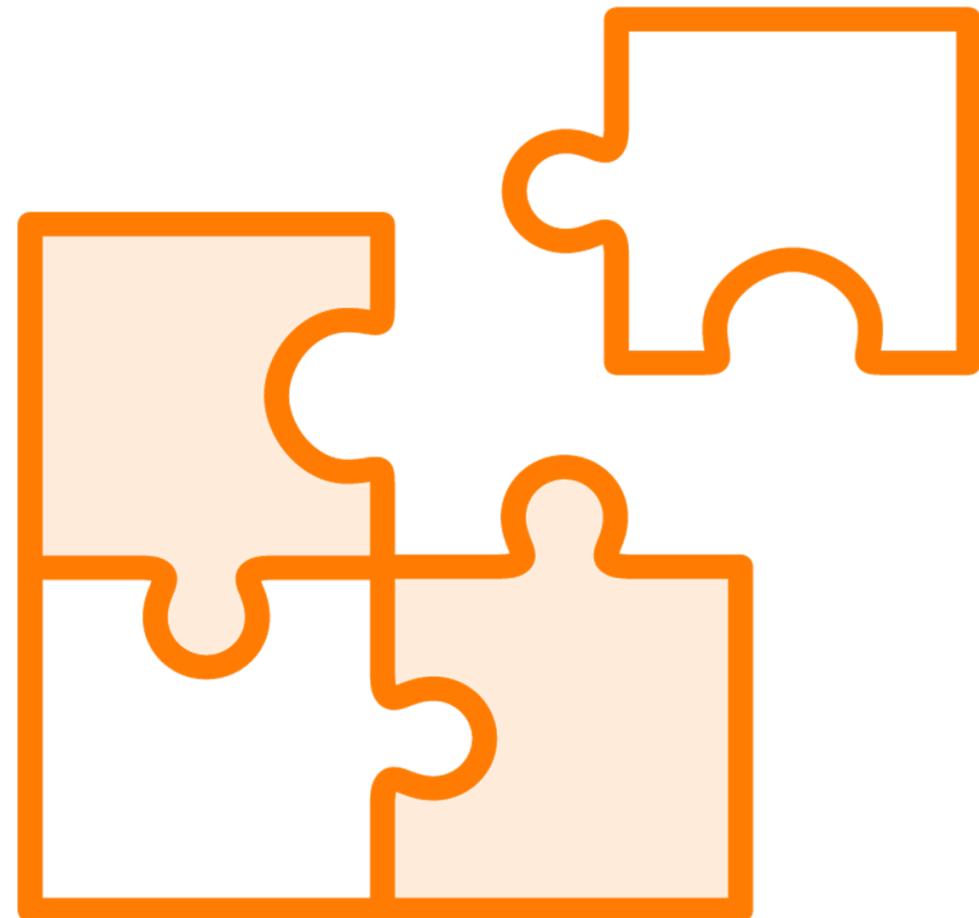
Often discovered during security and penetration testing

Examples

- SQL injections
- Cross-site scripting
- Buffer overflows
- Inferior authentication



Logic Errors



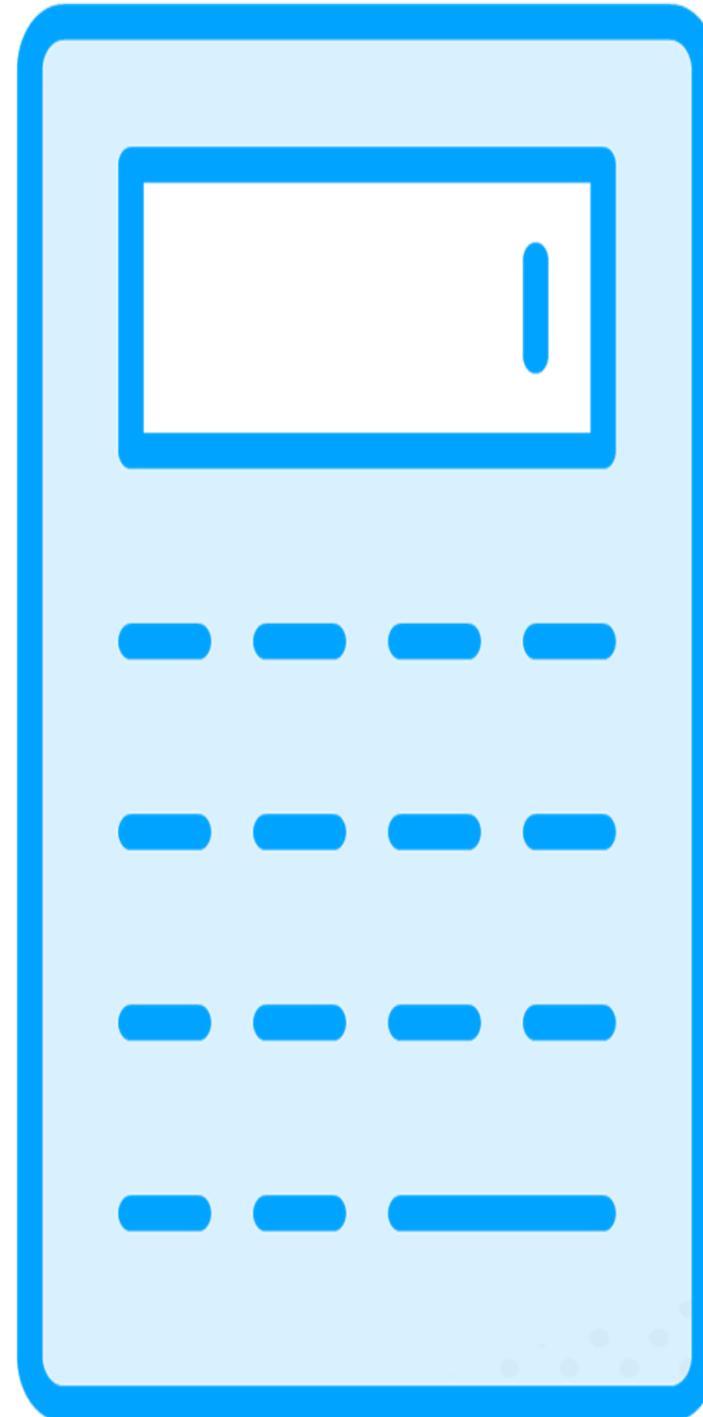
Coding errors that result in wrong output

Examples

- Endless loops
- Object not initialized
- Counter not reset
- Control flow issue



Calculation Errors

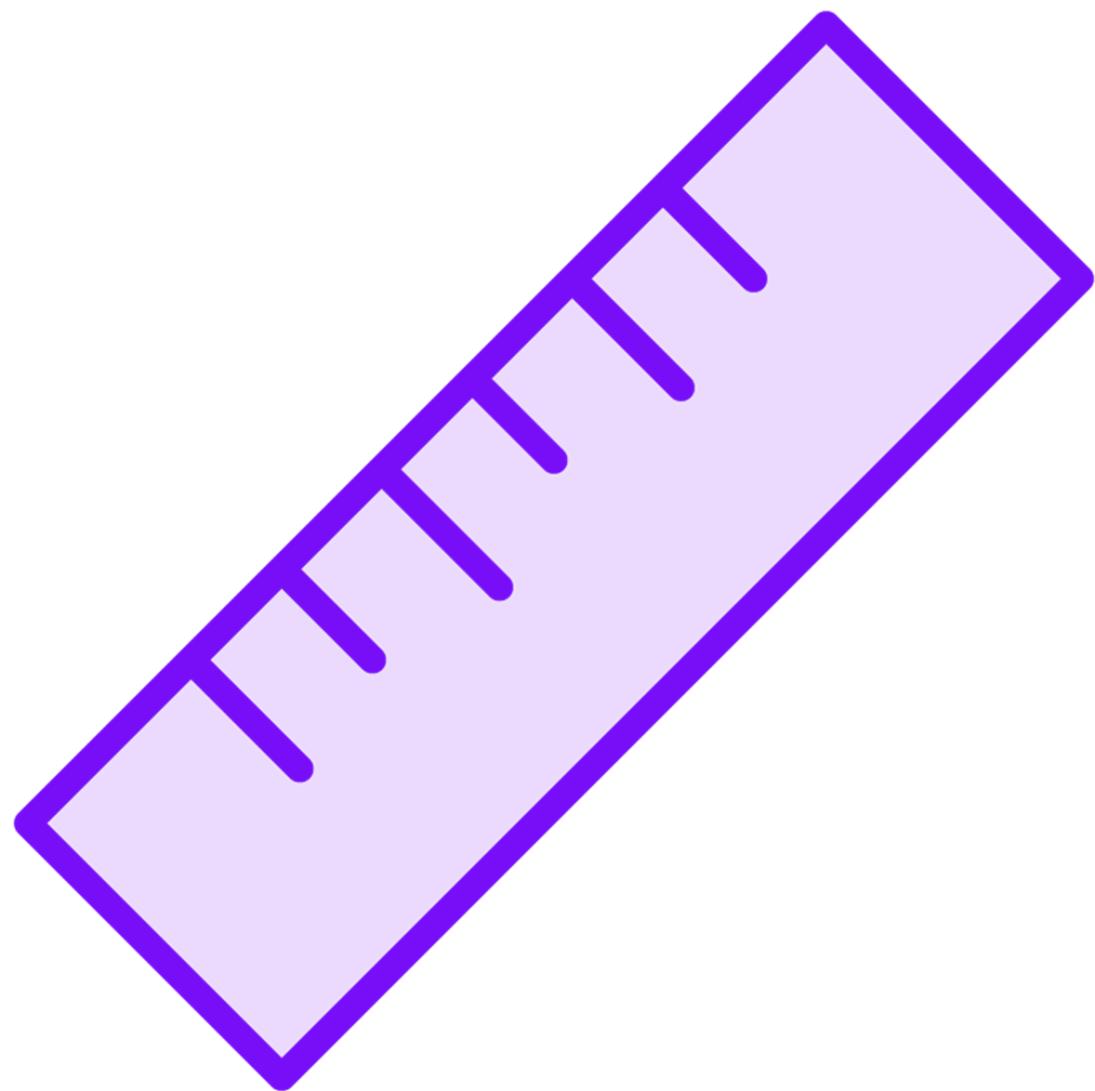


Examples

- Incorrect formula produces wrong result
- Could be costly when dealing with money
- Different units used by different parts of app



Out-of-bound Bugs



Values entered that are outside of expected range

Data type entered that wasn't expected



Compatibility Defects

Operating Systems

Hardware

Browsers

Mobile Platforms

Bandwidth

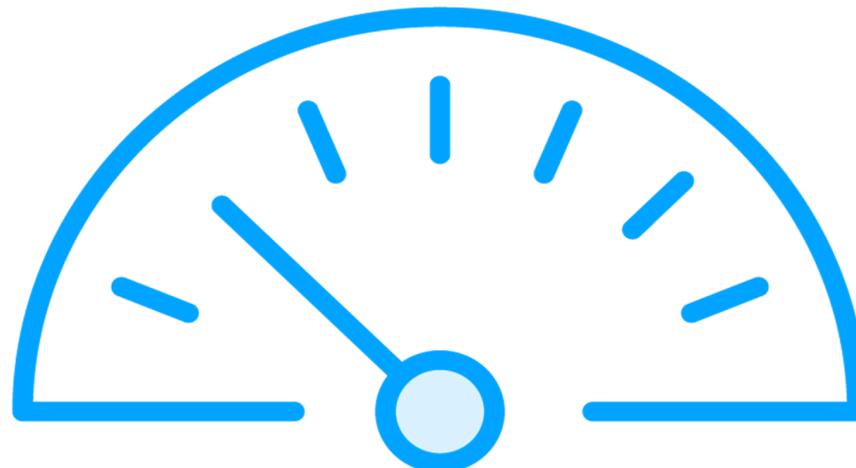
Operating Speeds



System-level Integration Bugs



Performance Defects



Speed

Stability

Response time

Resource consumption

Often found during performance testing

Compare actual values to non-functional requirements



Usability Defects

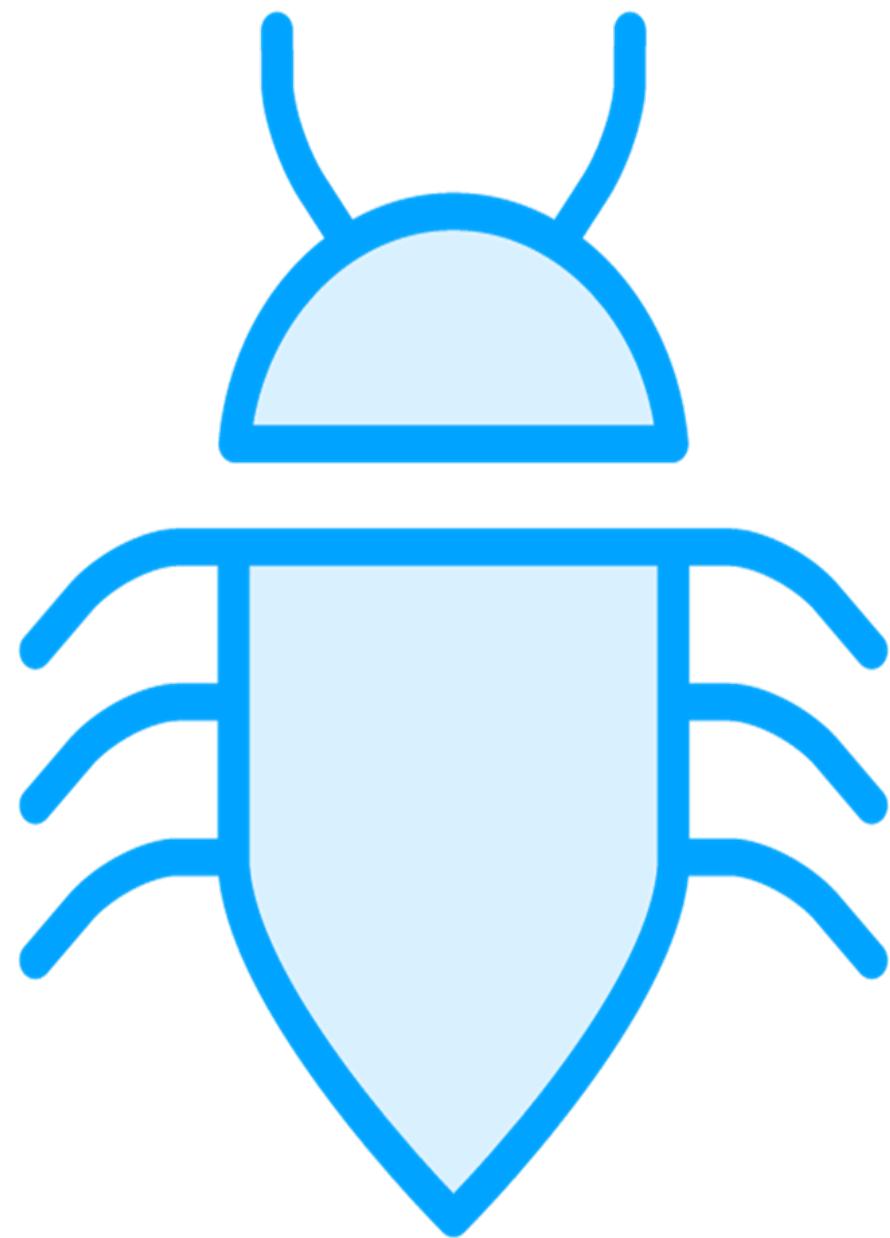


Prevent using software to its fullest capability

Examples

- Complicated user interface
- Violating accessibility standards (WCAG)
- Not providing feedback to user
- Ambiguous icons





- Functional errors**
- Security defects**
- Syntax errors**
- Logic errors**
- Calculation errors**
- Out-of-bound bugs**
- Compatibility defects**
- System-level integration bugs**
- Performance defects**
- Usability defects**





Benefits of Becoming a Better Debugger



Debugging as an Opportunity

- To learn how the program works**
- To learn about the kinds of mistakes YOU make**
- To learn about how YOU fix defects**





Don't Panic!



**Solving Problems
Yourself is Incredibly
Rewarding**



Module Summary



Understanding debugging

Categories of defects

Debugging basics in IDE

Debugging an enterprise application



Up Next:

Up Next:

Debugging Data and Inputs

