

# Context Factors in Choosing Debugging Strategies

# Programming Strategy

A strategy is a **high-level plan** for accomplishing a task, describing a series of **steps** to take to accomplish a **goal**.

Describe how software developers do similar everyday tasks

E.g., Offering techniques to debug a tricky defect, improve the performance of an application, or design and refactor code.

# A Sample Debugging Strategy

## **STRATEGY LocalizeFault()**

read the names of all of the functions and variables in the program

IF the faulty output is logged to a command line

    SET 'outputLines' TO the line numbers of calls to console logging functions

FOREACH 'line' in 'outputLines'

    IF the program executed 'line'

        Analyze the line to determine its role in the overall behavior of the program

        IF any part of 'line' is inconsistent with its purpose

            RETURN 'line' as the fault location

        IF 'line' executed with an incorrect value

            SET 'wrongValue' TO the incorrect value

            RETURN STRATEGY LocalizeWrongValue('wrongValue')

    IF 'line' does not execute at all

.....

# A Sample Debugging Strategy

**STRATEGY LocalizeFault()**

**read the names of all of the functions and variables in the program**

IF the faulty output is logged to a command line

SET 'outputLines' TO the line numbers of calls to console logging functions

FOREACH 'line' in 'outputLines'

IF the program executed 'line'

Analyze the line to determine its role in the overall behavior of the program

IF any part of 'line' is inconsistent with its purpose

RETURN 'line' as the fault location

IF 'line' executed with an incorrect value

SET 'wrongValue' TO the incorrect value

RETURN STRATEGY LocalizeWrongValue('wrongValue')

IF 'line' does not execute at all

.....

# A Sample Debugging Strategy

## STRATEGY LocalizeFault()

read the names of all of the functions and variables in the program

**IF** the faulty output is logged to a command line

SET 'outputLines' TO the line numbers of calls to console logging functions

FOREACH 'line' in 'outputLines'

IF the program executed 'line'

Analyze the line to determine its role in the overall behavior of the program

IF any part of 'line' is inconsistent with its purpose

RETURN 'line' as the fault location

IF 'line' executed with an incorrect value

SET 'wrongValue' TO the incorrect value

RETURN STRATEGY LocalizeWrongValue('wrongValue')

IF 'line' does not execute at all

.....

# A Sample Debugging Strategy

## STRATEGY LocalizeFault()

read the names of all of the functions and variables in the program

IF the faulty output is logged to a command line

SET 'outputLines' TO the line numbers of calls to console logging functions

FOREACH 'line' in 'outputLines'

IF the program executed 'line'

Analyze the line to determine its role in the overall behavior of the program

IF any part of 'line' is inconsistent with its purpose

RETURN 'line' as the fault location

IF 'line' executed with an incorrect value

SET 'wrongValue' TO the incorrect value

RETURN STRATEGY LocalizeWrongValue('wrongValue')

IF 'line' does not execute at all

.....

# Debugging Strategies

This list is curated from books and literature review:

[Debugging Strategies.pdf](#)

# Context Factors

Specific **elements and circumstances** surrounding a particular situation or issue

These factors influence the **choice** of a debugging strategy.



# Context Factors Examples

## DEFECT CHARACTERISTICS

If the bug is happening every time you run the program, use strategy X

If the problem is shown through an error message -> Strategy ??

When facing a bug in production

## CODEBASE CHARACTERISTICS

When working on an unfamiliar codebase

## TOOL CHARACTERISTICS

When debugger tool is available

## ORGANIZATION CHARACTERISTICS

When working in a collaborative environment and you found a defect -> strategy C

If there is an Agile development culture -> Strategy A

## INDIVIDUAL CHARACTERISTICS

If you have experience using a tool for debugging

When you are familiarity with the programming environment and language.

Can you think of any context factor that affected your choice of debugging strategy recently?

# Phase 1

Let's Start the first phase of the study