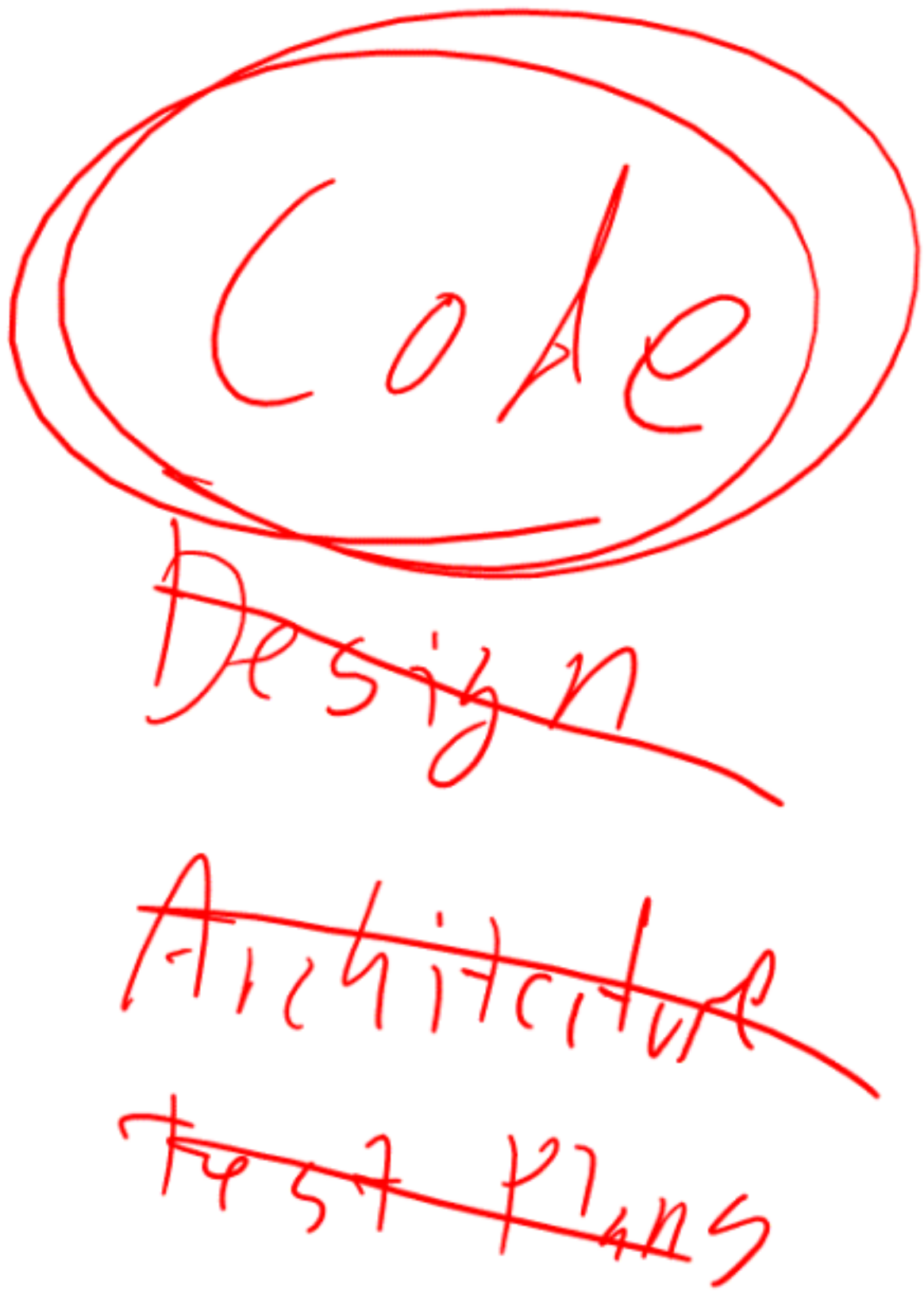
Exam results

Secure Software Development
Code Review With a Tool

Objectives

- List disadvantages of code review
- Explain the concept of static analysis
- Define soundness and completeness
- Explain the risks of false positives and false negatives
- Explain how SA Tools can be integrated into a development cycle
- Justify an architecture for a static analysis system

What artifacts do all software projects have?



Code Review

- Effective methodology for detecting bugs

⇒ Read the code

What is wrong with code reviews

- Takes a lot of time
  - ⇒ manpower intensive
- May not find everything
  - ⇒ Humans miss things
- Very tedious

How do we catch glaring coding mistakes?

```
public static int doSomething()
{
    int x = 0;
    int y = 1;
    int z = x + y
    return z
}
```

;

;

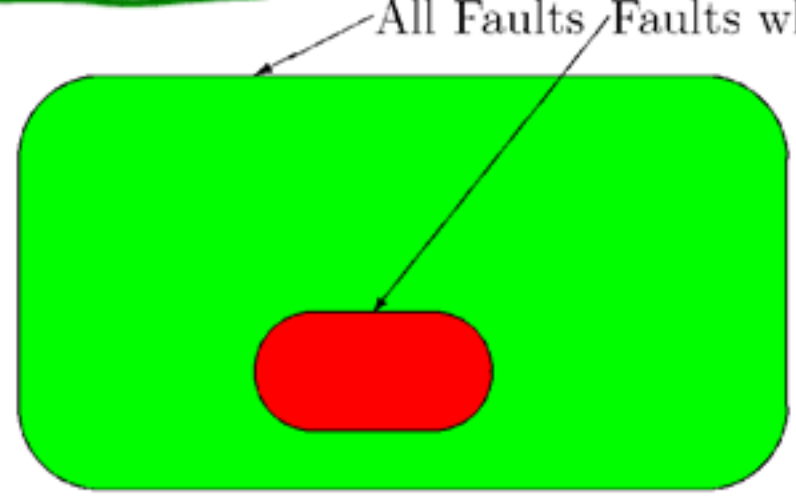
10-15 s

Tool => almost instantly

Parser to detect this error.

Static Analysis Overview

- Similar to a spell checker or grammar checker.
- Search through code to detect bug patterns
  - error prone coding practices that arise from the use of erroneous design patterns, misunderstanding of language semantics, or simple and common mistakes.
- Static Analysis tools detect faults
  - Not all faults will fail
    - 90% of downtime comes from 10% of the faults



- Can detect many different classifications of software faults
  - Coding standards violations
  - Buffer overflows (Viega et all)
  - Security vulnerabilities (Livshits and Lam)
  - Memory leaks (Rai)
  - Timing anomalies (race conditions, deadlocks, and livelocks) (Artho)

```
int array[10];
    ⋮
{
    array[10] = 0;
```

Assuming object is out of bounds

Issue: Soundness

- Many tools cut corners
  - Pointer analysis is hard
  - Two choices: leave some bugs behind or get swamped by false positives
  - Delicate balance
  - Still very good at catching the low hanging fruit and finding dangerous constructs

Issue: False Positives and False Negatives

- False positives are costly

⇒ mistakes flagged that are not issues.

- False negatives

Mistakes that are missed by the tool.

Static Analysis Problems

- Static analysis tools aim for good, not perfect
  - False Positive rate can be very high
  - Greater than 50% for certain tools

- Tools can be influenced by programmer style.
  - Tools may need to be "tuned" based on constructs used

Configuration

What are we looking for in an SA tool?

- Be designed for security
- Support multiple tiers
- Be extensible
- Be useful for developers and security analysis as well
- Support existing development processes
- Make sense to stakeholders

HP Translates code

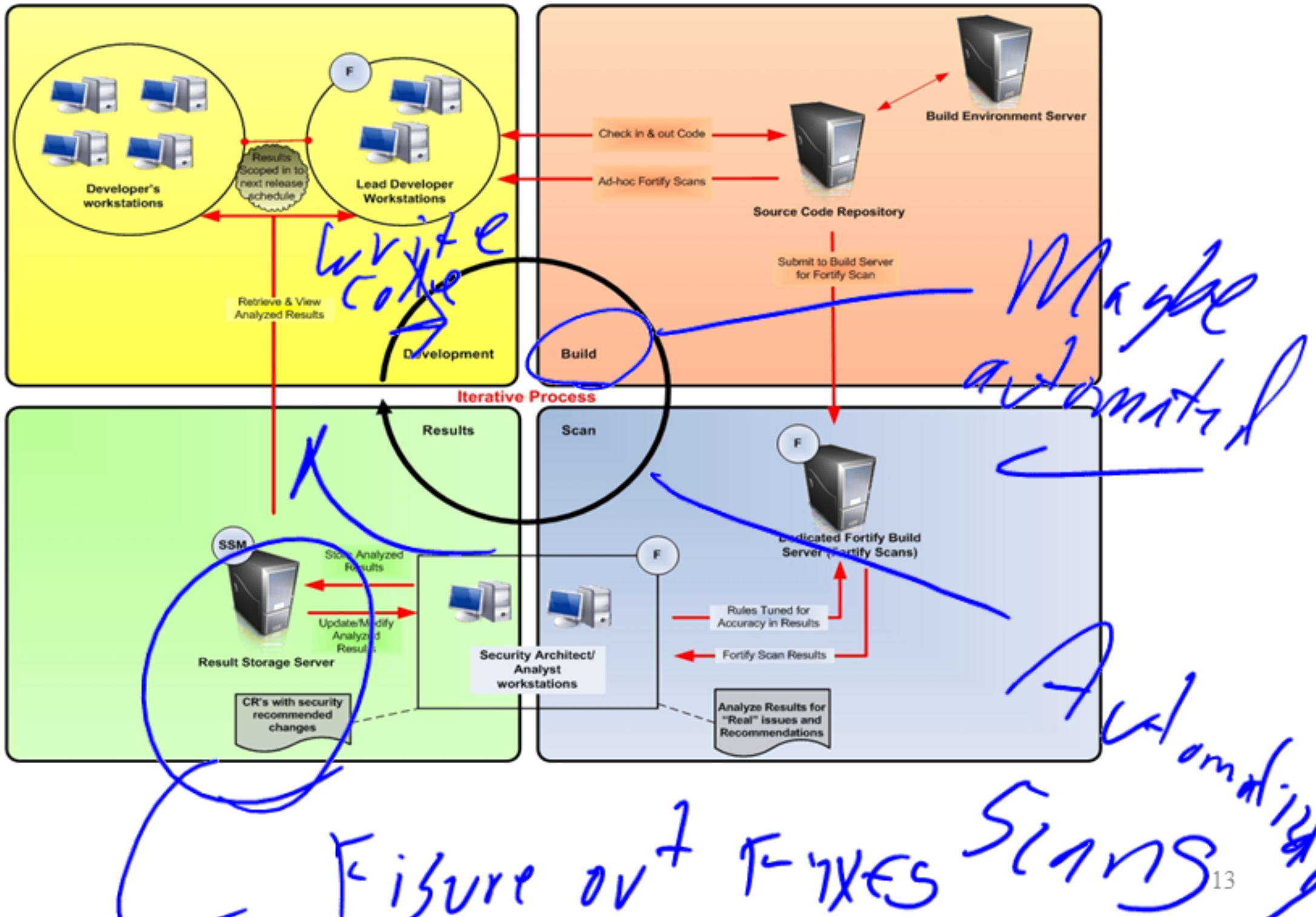
Language agnostic output

Define your own.

Definitions for bugs

Tool Integration into your build cycle

Example of Bureau Model Implementation



Cigital Process