

CSE 3421

Software Testing

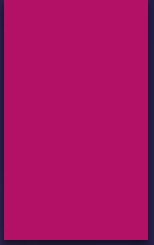
SUMMER 2021

MD. RAFI-UR-RASHID

LECTURER, DEPT. OF CSE, UIU



Why This Lesson?



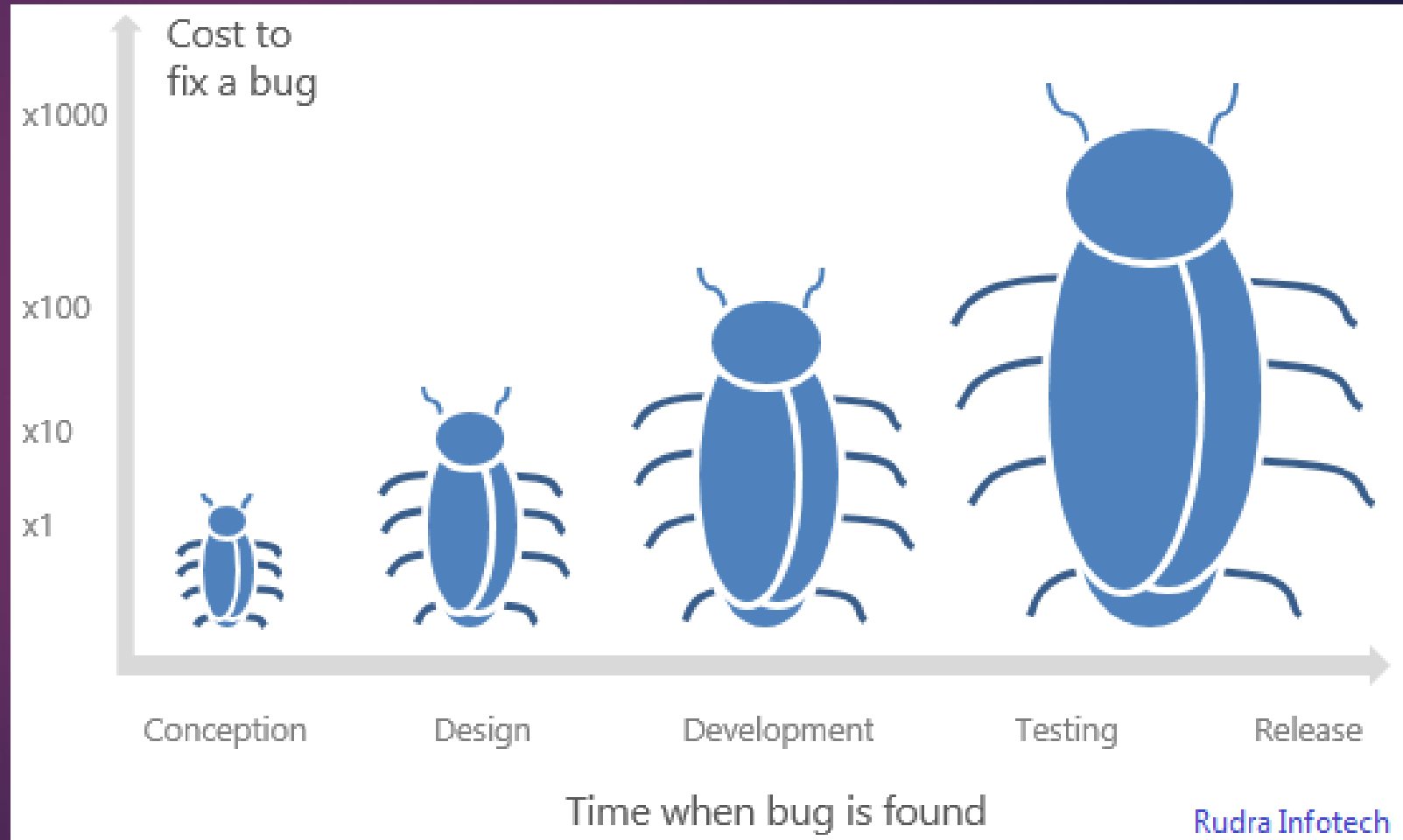
**No customer likes to buy faulty products.
Same goes for the software industry**

What is software testing?

Software testing is the process of analyzing a software item to detect the differences between existing and expected conditions (i.e., bugs)

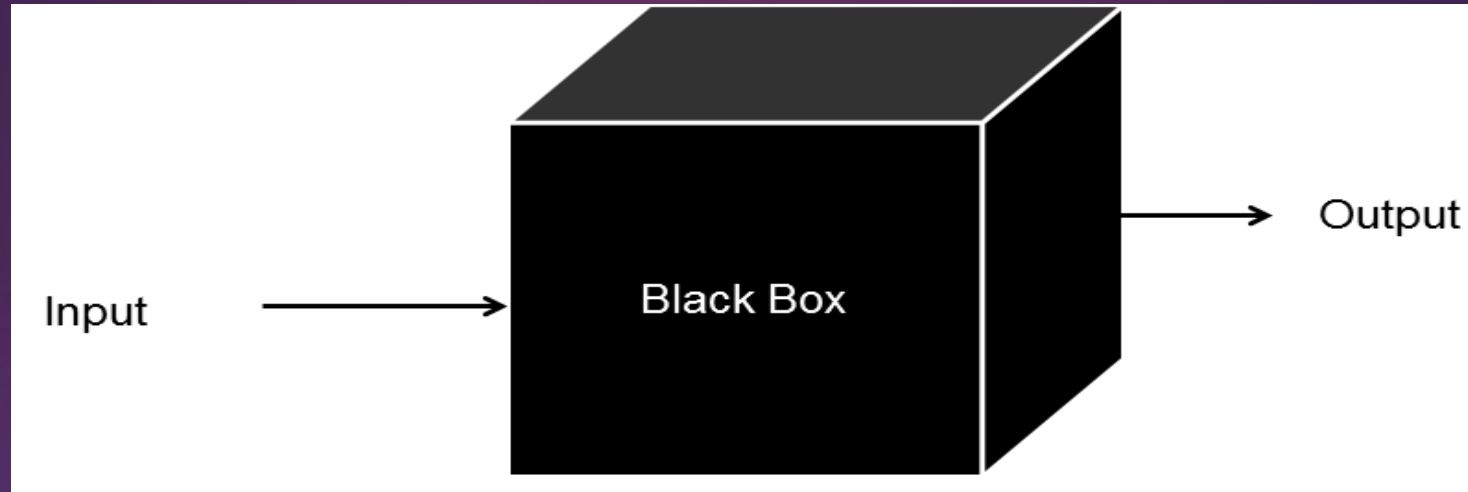
When should we test software?

Answer: Throughout the whole development process



Testing Models

Black box testing



Ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.

Black box testing

- ▶ Interface visible, internals unknown
- ▶ From the outside, you are testing its functionality against the specs
- ▶ For software this is testing the interface
 - ▶ What is input to the system?
 - ▶ What you can do from the outside to change the system?
 - ▶ What is output from the system?
- ▶ Tests the functionality of the system by observing its external behavior

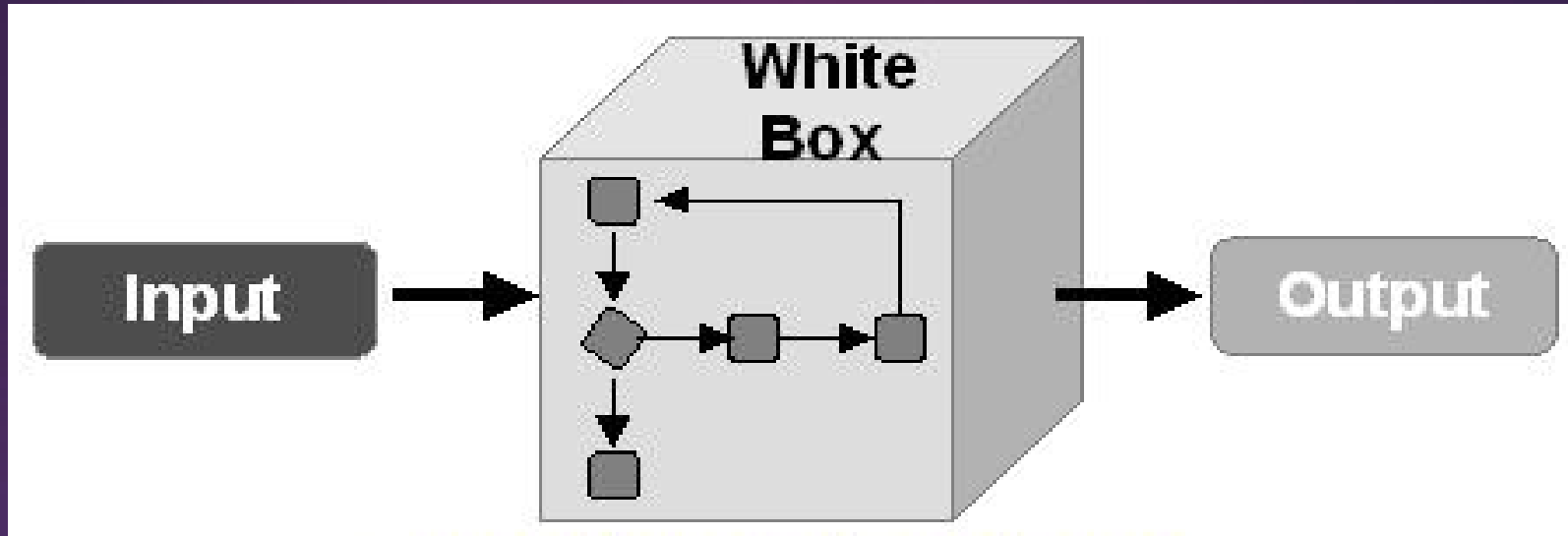
Advantages

- ▶ Robust with respect to changes in implementation
 - Test data need not be changed when code is changed
- ▶ Allows for independent testers
 - Testers need not be familiar with code
- ▶ Less time-consuming

Disadvantages

- ▶ It will miss bugs in the implementation that are not covered by the specification
- ▶ Does not allow performance optimization and alternate algorithms

White box testing



takes into account the internal mechanisms of a system or component.

White box testing

- ▶ Given knowledge of the internal workings, you thoroughly test what is happening on the inside
- ▶ Close examination of procedural level of detail
- ▶ Logical paths through code are tested
 - Conditionals
 - Loops
 - Branches
- ▶ Status is examined in terms of expected behaviors

Advantages

- ▶ Allows thorough testing of the software
- ▶ Helps finding hidden bugs & errors
- ▶ Yields useful test cases.
- ▶ Enables code optimization and experiment with alternate algorithms.

Disadvantages

- ▶ Impossible to thoroughly exercise all paths
 - Exhaustive testing grows without bound
- ▶ Requires Code Access
- ▶ Skilled tester required
 - Requires high level knowledge of internals of the software under test.
- ▶ More time-consuming

Functional testing

- ▶ **FUNCTIONAL TESTING** is a type of software testing whereby the system is tested against the functional requirements specifications (FRS).
- ▶ Functions (or features) are tested by feeding them input and examining the output.
- ▶ This type of testing is not concerned with how processing occurs, but rather, with the results of processing.
- ▶ During functional testing, Black Box Testing technique is used in which the internal logic of the system being tested is not known to the tester.

Non-Functional/ Technical testing

- ▶ **Technical TESTING** is a type of software to check non-functional aspects (performance, usability, reliability, security etc.) of a software.
- ▶ It is explicitly designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing
- ▶ It basically uses white-box testing, sometimes black box



Types of Testing

Unit testing

- ▶ Testing of individual software or hardware units or groups of related units
- ▶ Done by programmer(s)
- ▶ Generally white box
- ▶ Verify that code does what it is intended to do at a very low structural level
- ▶ Automation desirable for repeatability
- ▶ **Component Testing:** Almost identical to unit testing, except at a slightly larger scale



Integration testing

- ▶ Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them
- ▶ Done by programmer as they integrate their code into code base
- ▶ Verifies that units work together when they are integrated into a larger code base
 - ▶ Just because the components work individually, that does not mean that they all work together when integrated
- ▶ Generally white box, maybe some black box
- ▶ Automation desirable for repeatability

System Testing

- ▶ **System testing** is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.
- ▶ System testing is performed on the entire system in the context of a **Functional** Requirement Specification(s) (FRS) and/or a **Non-functional/Technical** Requirement Specification that in combination generate Software Requirement Specification (SRS).
- ▶ So, it covers non-functional requirements as well.

Load, and Stress Testing

- ▶ **Load testing** is a kind of Performance Testing which determines a system's performance under real-life load conditions. **Load testing** usually identifies the maximum operating capacity of an application.
- ▶ **Load** can be the **expected concurrent number of users** on the application performing a specific number of transactions within the set duration. This test will give out the response times of all the important business critical transactions.
- ▶ **Stress testing** *implies* evaluating a system beyond the limits of its specification. In **Stress testing**, we measure the breakpoint of a system.

Difference between load and stress test

- ▶ The goal of a load test is to prove that a system can handle the expected volume with minimal to acceptable performance degradation.
- ▶ A stress test is a test designed to increase the number of simultaneous requests on a system past a point where performance is degraded, possibly even to the point of complete failure.
- ▶ **Difference in short:** If you are testing normal, expected load, this is *load testing*. But when you want to determine how the system behaves under extreme load (DoS) and when it breaks, this is *stress testing*.

Security testing

- ▶ **SECURITY TESTING** is a type of software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders.
 - ▶ **Vulnerability Scanning:** This is done through automated software to scan a system against known vulnerability signatures.
 - ▶ **Ethical hacking:** It's hacking an Organization Software systems. Unlike malicious hackers ,who steal for their own gains , the intent is to expose security flaws in the system.
 - ▶ **Penetration testing:** This kind of testing simulates an attack from a malicious hacker. This testing involves analysis of a particular system to check for potential vulnerabilities to an external hacking attempt.

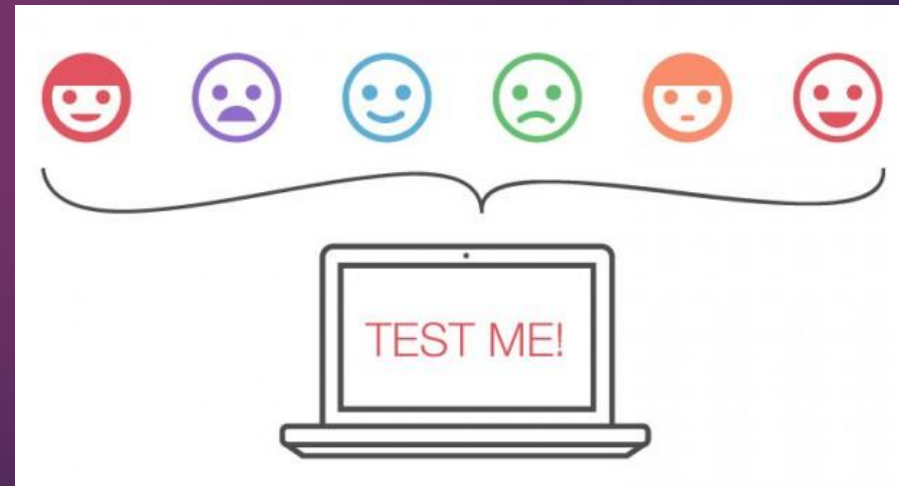
Regression testing

- ▶ Regression testing is selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements
- ▶ Subset of the original set of test cases.
- ▶ Core group of tests re-run often after any significant changes
 - ▶ Choose a representative sample of tests that exercise all the existing functionalities
 - ▶ Chose additional test cases that are most likely to be affected by the change
- ▶ **Smoke test:** a subset of the regression test cases that establish that the system is stable and all major functionality is present and works under “normal” conditions



Beta testing

- ▶ Organization can offer an advance partial or full version of a software package free to one or more potential users.
- ▶ Users use the software as they wish, with the understanding that they will report any errors revealed during usage back to the organization.
- ▶ Advantages
 - ▶ Identification of unexpected errors
 - ▶ Low costs
 - ▶ Wider population / environment
- ▶ Disadvantages
 - ▶ Lack of systematic testing
 - ▶ Low quality error reports

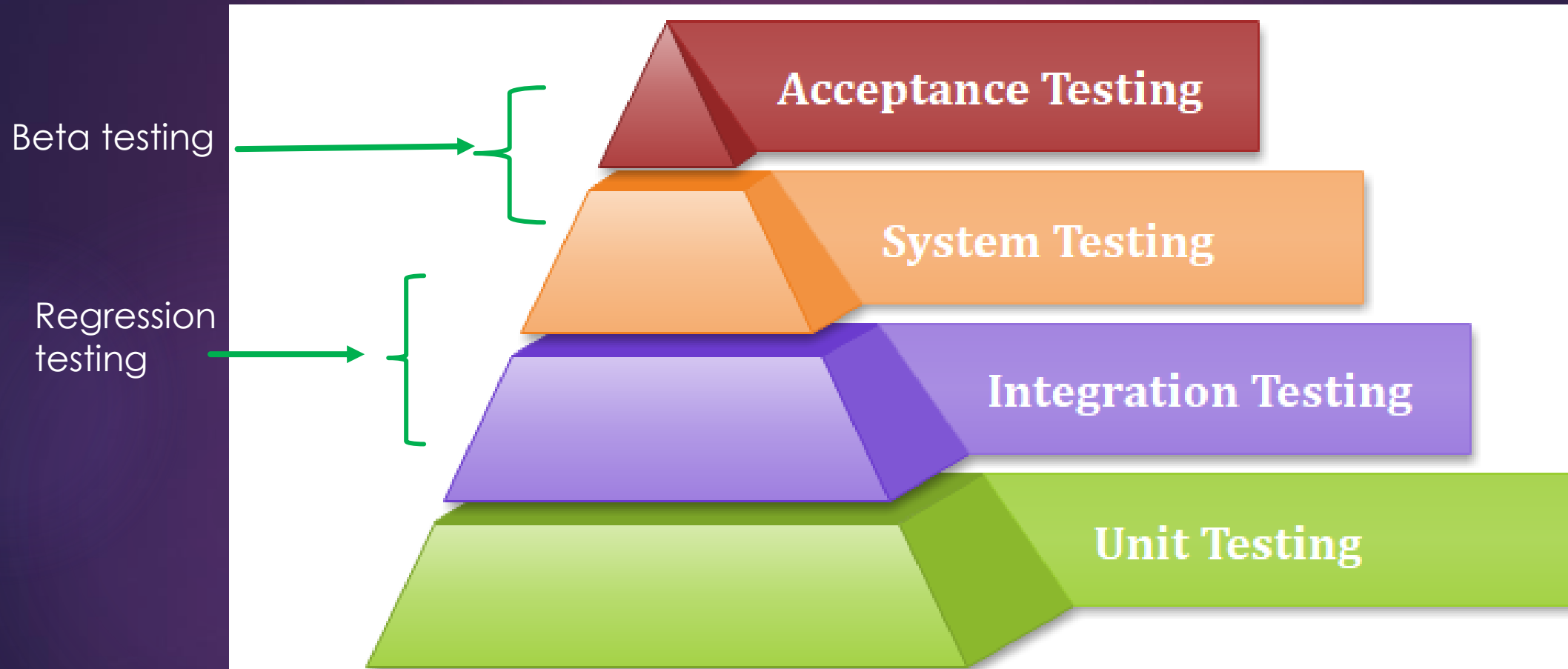


Acceptance testing

- ▶ Formal testing conducted to determine whether or not a system satisfies its acceptance criteria (the criteria the system must satisfy to be accepted by a customer) and to enable the customer to determine whether or not to accept the system
- ▶ Generally done by customer/customer representative in their environment through the GUI
- ▶ Definitely black box



Testing Hierarchy



Static & Dynamic testing

- ▶ Static testing refers to testing something that's not running.
- ▶ It is sort of examining and reviewing the work. E.g. code review, software inspection are some static testing methods.
- ▶ Dynamic Testing involves working with the software, giving input values and checking if the output is as expected.
- ▶ Unit Tests, Integration Tests, System Tests and Acceptance Tests are few of the Dynamic Testing methodologies.

Code smell

- ▶ Code smell implies code components that are **absolute violations of the fundamentals** of developing software that decrease the quality of code.
- ▶ Has negative impact on readability and maintainability.
- ▶ Code smells are usually not bugs; they are not technically incorrect and do not prevent the program from functioning.
- ▶ Instead, they indicate weaknesses in design that may slow down development or increase the risk of bugs or failures in the future.
- ▶ For instance Long function, code duplication, mysterious names etc.
- ▶ Tools such as *CodeGrip*, *Checkstyle*, *PMD*, *FindBugs*, and *SonarQube* can automatically identify code smells.



Code Review

Code review

- ▶ **Code review** is the process of analyzing code written by another developer on the project to judge whether it is of sufficient quality to be integrated into the main project codebase.
- ▶ The formal variant of peer code review, which is better known as software inspection or Fagan-inspection, has been an effective quality improvement practice for a long time.
- ▶ Even with the benefits offered by software inspections, their relatively high cost and formal requirements have reduced the prevalence with which software teams adopt them.
- ▶ Hence, lightweight, informal, and tool-based code review practices are being popular.

Code Review Efforts

32



Avg. 6 hours / week



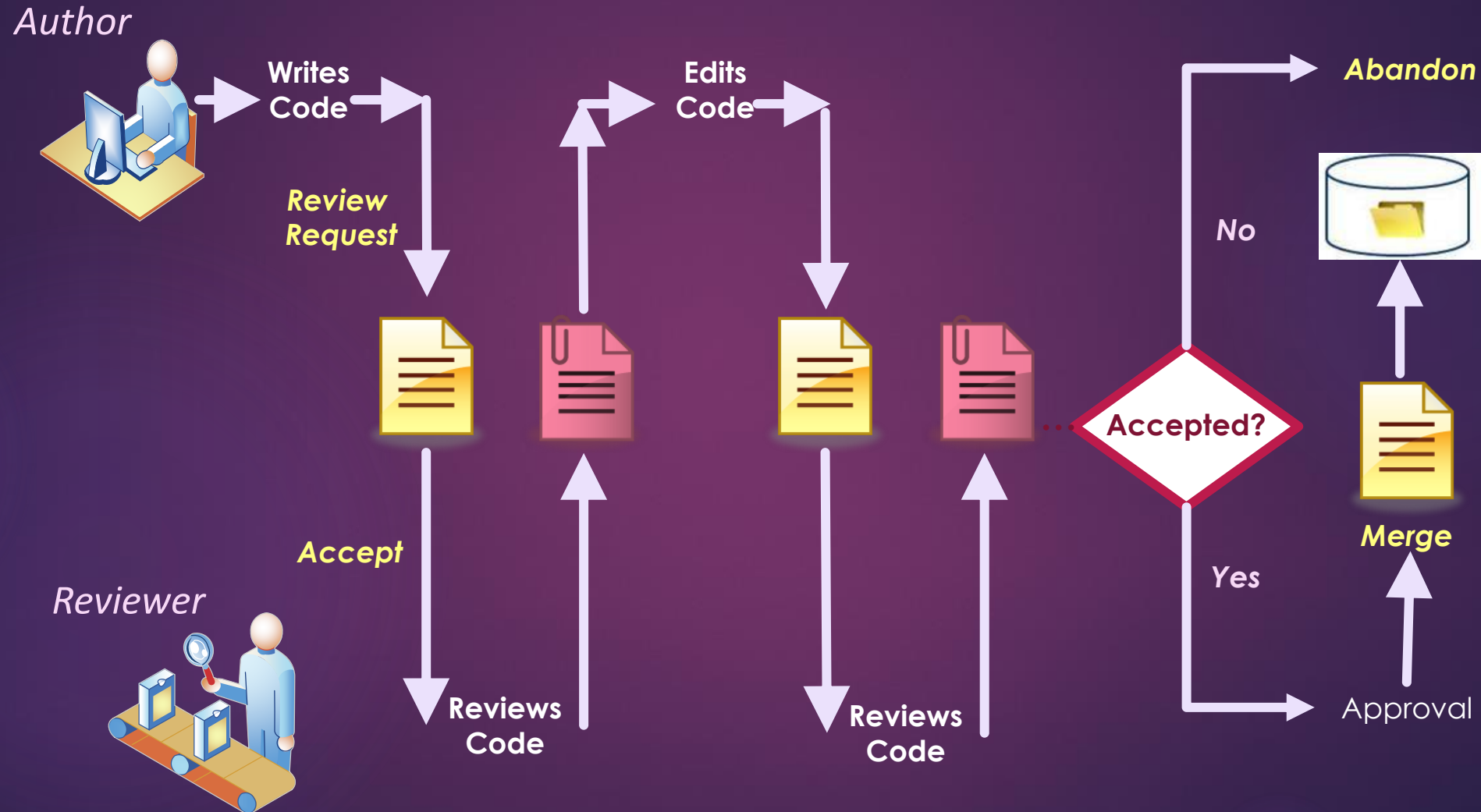
10%~15% of developer times



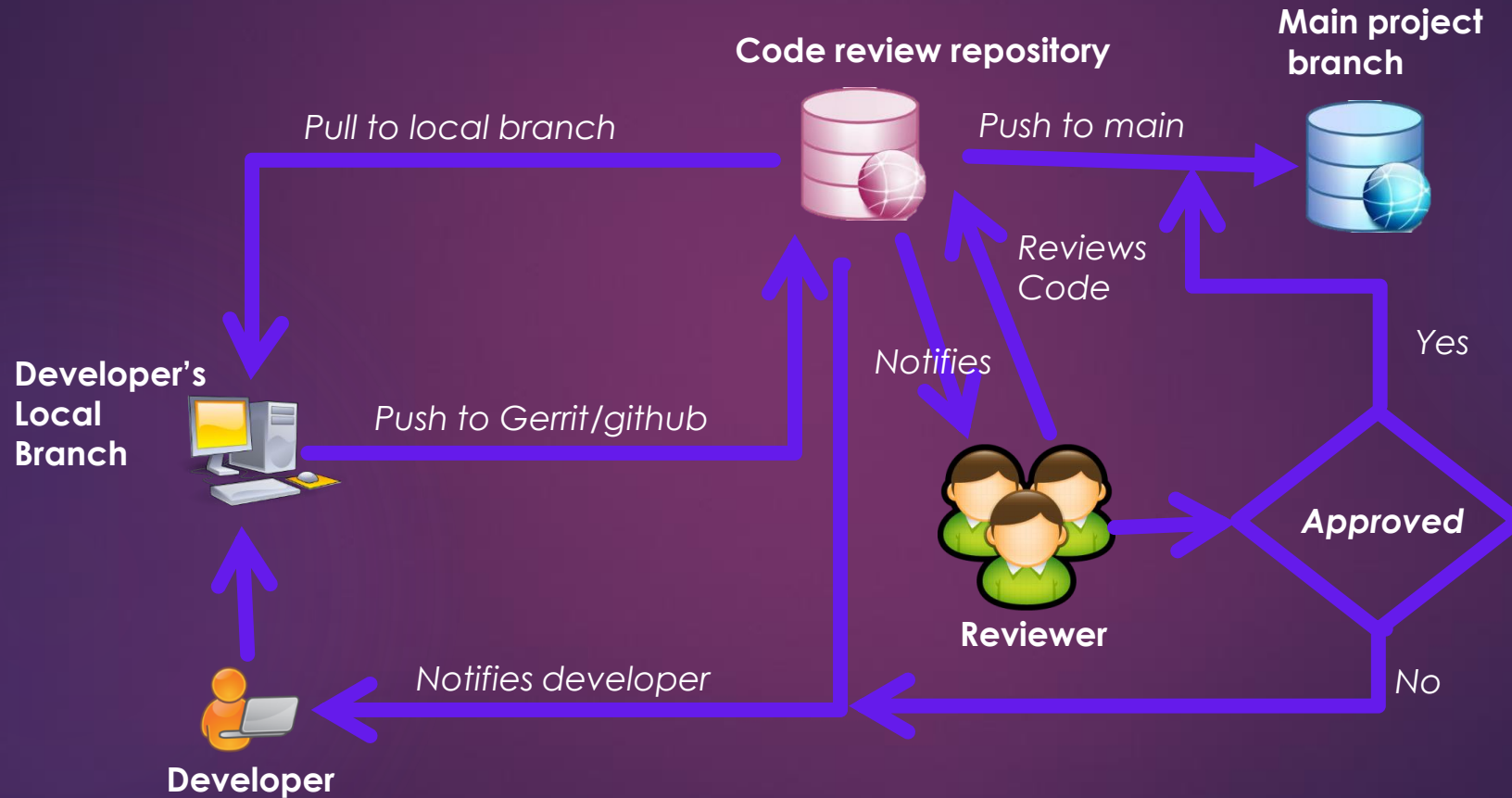
Microsoft : 50,000 developers X 6 hours/ week X \$50 /hour

= \$15,000,000 / Week

Code Review Workflow



Simplified code review workflow



Code Review Tools



Gerrit: <https://code.google.com/p/gerrit/>



Review Board: <https://www.reviewboard.org/>



Phabricator: <https://phabricator.org/>



Crucible: <https://www.atlassian.com/software/crucible>

Issues identified during code reviews

✓ Coding style

✓ Redundant checks

✓ Project design violation

✓ Misunderstood requirements

✓ Critical security defects

✓ Malicious code

✓ Inadequate input validation

✓ Unsafe methods

✓ Lack of exception handling

Practical Impacts

- ▶ Peer review can put strain on interpersonal team relationships.
- ▶ It's difficult to have every piece of work critiqued by peers and to have management evaluating and measuring defect density in your code.
- ▶ It's extremely important that managers create a culture of collaboration and learning in peer review.
- ▶ While it's easy to see defects as purely negative, each bug is actually an opportunity for the team to improve code quality.

Other Benefits of Code review

38

- ▶ Project awareness
- ▶ Peer impression formation
- ▶ Consistent coding style
- ▶ Knowledge dissemination



Thank You