

## Homework 2

### ECE 590: Towards More Reliable Software

Jeff Fan || zf70@duke.edu  
February 12, 2024

#### Question 1:

a. Describe the faults classification in the papers (Drawing a graph may make it clearer).

**Bohrbugs:** These are straightforward, solid faults that manifest consistently under a well-defined set of conditions. They are typically easier to detect and fix.

**Mandelbugs:** These faults are characterized by complexity and obscurity, making their behavior appear chaotic or non-deterministic. They are influenced by interactions within the software system or its environment and may have a delayed failure occurrence.

**Heisenbugs:** These are a subtype of Mandelbugs and refer to faults that change behavior or disappear when an attempt is made to probe or isolate them. Heisenbugs are known for their elusive nature; they often change behavior or disappear under investigation, like when using a debugger. This behavior can be influenced by changes in timing, memory state, or interactions with other processes.

**Aging-Related Bugs:** These are also a subtype of Mandelbugs. They lead to the accumulation of errors over time in the application or its environment, resulting in increased failure rates or degraded performance. Aging-Related Bugs typically emerge over time due to the degradation of system state, resource leaks, data corruption, or accumulation of errors. These bugs become more prevalent or noticeable as the system continues to operate.

In some cases, an Aging-Related Bug might exhibit characteristics of a Heisenbug. For example, a memory leak (an Aging-Related Bug) might only cause noticeable issues under certain conditions, such as when the system is under heavy load or has been running for an extended period.

And I draw a graph to illustrate their relationship.

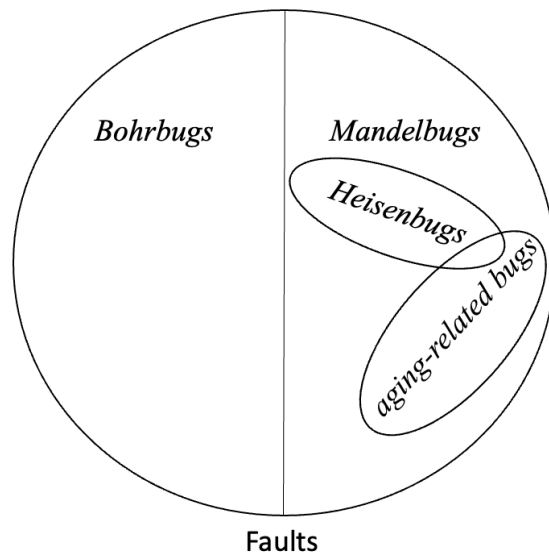


Fig. 1. The relationship of faults.

**b. Explain different ways to deal with different types of bugs and to recover from the failures caused by the bugs.**

**Bohrbugs:** Bohrbugs are straightforward and consistently reproducible, making them relatively easier to manage. The primary strategy is to remove them, which involves detection, diagnosis, and correction during the development and testing phases. Systematic testing, like unit and system tests, is crucial here. Early detection and removal are cost-effective and prevent these predictable bugs from affecting the operational phase.

**Mandelbugs:** Mandelbugs are complex and may not manifest consistently, making them challenging to handle. 'Retry' and 'Replicate' are effective. 'Retry' involves re-executing the failed action, which might succeed due to the non-deterministic nature of these bugs. 'Replication' uses redundant resources or systems to take over if a failure occurs. These methods are used because Mandelbugs' behavior is often unpredictable, and traditional debugging methods might not be feasible.

**Heisenbugs:** Heisenbugs change behavior when investigated, making them elusive. Similar to Mandelbugs, strategies like retrying actions or replicating processes are useful. It's like trying to locate a noise in your car that stops whenever you take it to the mechanic. Keeping a log of when it happens could help identify patterns. Additionally, detailed logging and monitoring can help capture their

state and behavior before they alter. These bugs require such approaches because traditional debugging might cause them to disappear or behave differently.

**Aging-Related Bugs:** These bugs emerge over time, leading to gradual system degradation. The key strategy is 'Rejuvenation', which includes proactive measures like restarting systems or components before failures occur. Think of an old computer that slows down over time. Regularly rebooting it (rejuvenation) can help maintain its performance. This can be done based on monitoring system performance and identifying signs of aging. This approach is effective because it prevents the accumulation of errors or resource depletion that typically cause aging-related issues.