

# Shift Left Testing

Shift Left Testing means starting the testing process early in the Software Development Life Cycle (SDLC), instead of testing only after development is completed. By moving testing "to the left" on the timeline—toward requirements and design—defects are caught sooner, reducing time, effort, and cost.

In short: *Test early, fix early.*

## Why People Get Confused

Many assume testing begins only after development is finished. Shift Left Testing changes this mindset by involving testers from the very beginning—during requirement gathering, design discussions, and early builds.

## Simple Testing Example

### Traditional Testing (No Shift Left)

- Developer completes the entire feature.
- Tester starts testing at the end.
- Bugs are found late.
- Fixing becomes time-consuming and expensive.

### Shift Left Testing

- Testers verify requirements before development starts.
- Testers review design for possible issues.
- Test early builds or components.
- Bugs are caught and fixed early.

**Example:** Before coding a login feature, testers validate:

- Username format
- Password rules

- Error messages
- Flow diagrams

Finding issues here avoids major rework later.

## Real-Time Example

### Example: Flight Ticket Booking Website

#### Without Shift Left:

- Full booking module is developed.
- Testers find multiple bugs afterward.
- Fixing delays release.

#### With Shift Left:

- Testers validate requirement: "User must select a future date."
- During design phase, they question edge cases like: "What if user picks a past date?"
- During development, testers test small modules (calendar, dropdown, search).

**Result:** Faster delivery, fewer bugs, higher quality.

## Building a House

Checking the blueprint early helps catch mistakes before construction begins. If errors are found after the house is built, fixing becomes costly.

Shift Left Testing works the same way—catch errors early to avoid bigger problems later.

## One-Line Interview Summary

**"Shift Left Testing means testing early in SDLC to detect defects sooner and reduce cost."**

---

# Advantages of Shift Left Testing

## 1. Early Defect Detection

You find bugs at the requirement or design stage itself.

## 2. Saves Cost

Fixing a bug early is much cheaper than fixing it after development.

## 3. Saves Time

No last-minute bug rush. Project delivery becomes faster.

## 4. Reduces Rework

Because issues are caught early, the development team avoids rewriting big parts of the code.

## 5. Improves Product Quality

The final product becomes more stable with fewer defects.

## 6. Better Collaboration

Testers, developers, and business teams work together from the start.

## 7. Faster Releases

Since testing begins early, release cycles become quicker (very useful for Agile).

## 8. Helps in Requirement Clarity

Testers review requirements early, so missing or unclear requirements get fixed before development.

---

# Disadvantages of Shift Left Testing

## 1. Requires More Planning

Teams must plan the testing strategy early, which takes effort.

## 2. Involves Many Stakeholders

Developers, testers, and business teams need to collaborate from day one.

### **3. Testers Need Extra Skills**

Testers should understand requirements, design, and architecture—not just test execution.

### **4. Early Testing Effort May Look High**

Teams may feel they are spending more time upfront (but it saves more time later).

### **5. Frequent Changes**

If requirements change often, testers must update test cases and review documents repeatedly.

### **6. Not Suitable for All Projects**

Very small or short-term projects may not benefit much from early testing.

---

**Prepared by Betappa Bharath**