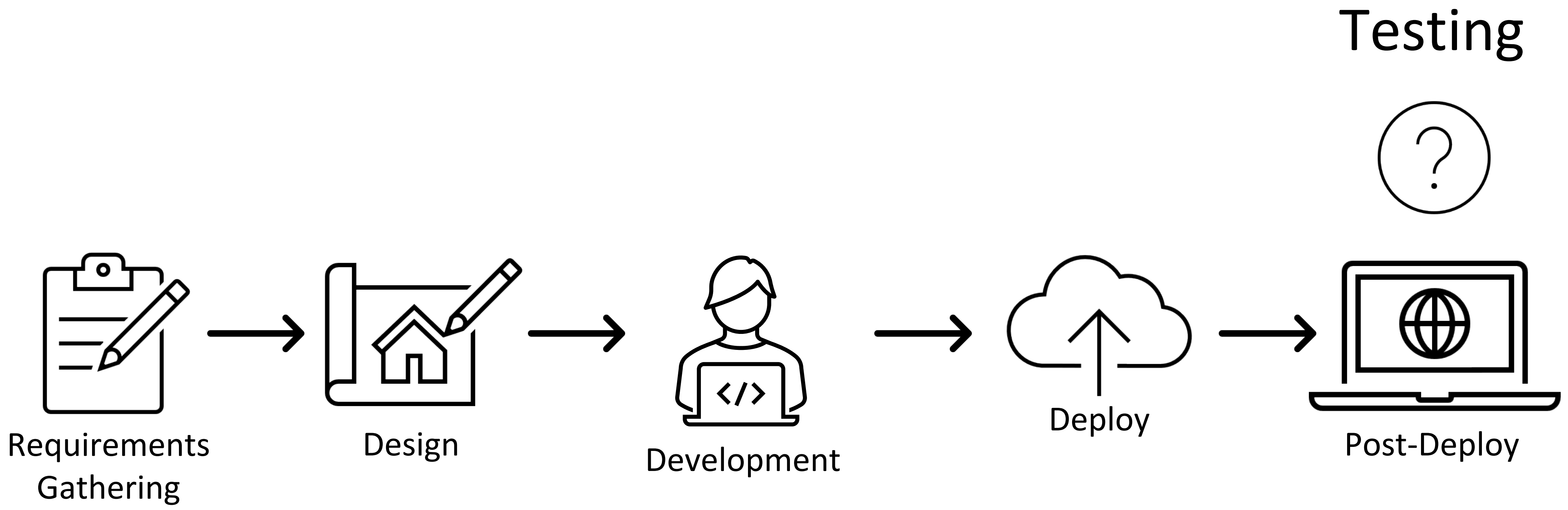
SAND2022-13634C

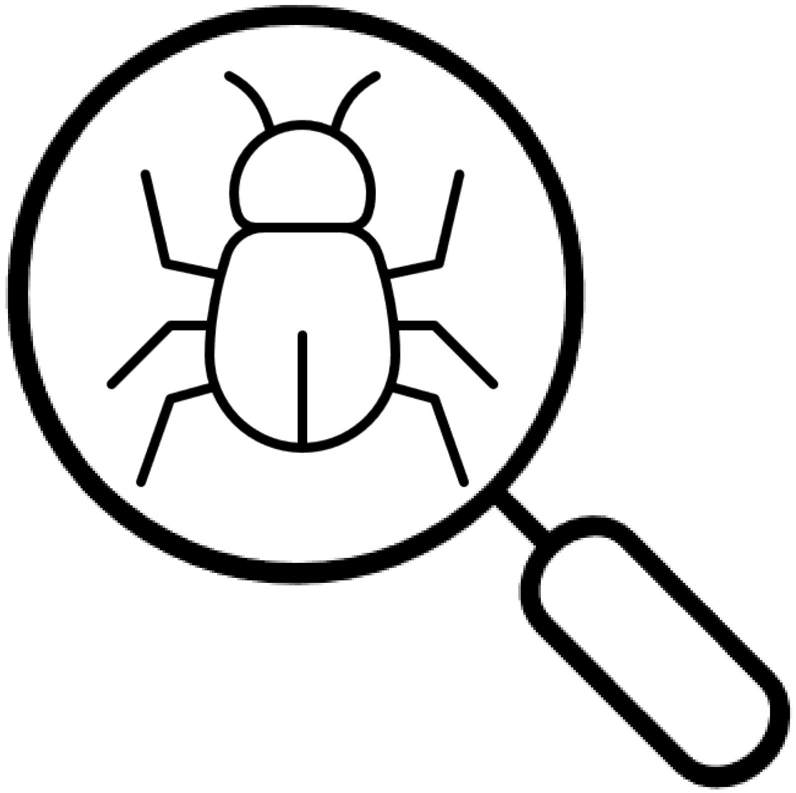
Test Early, Fail Fast….
Shift Testing Left

Presented by Douglas Stewart

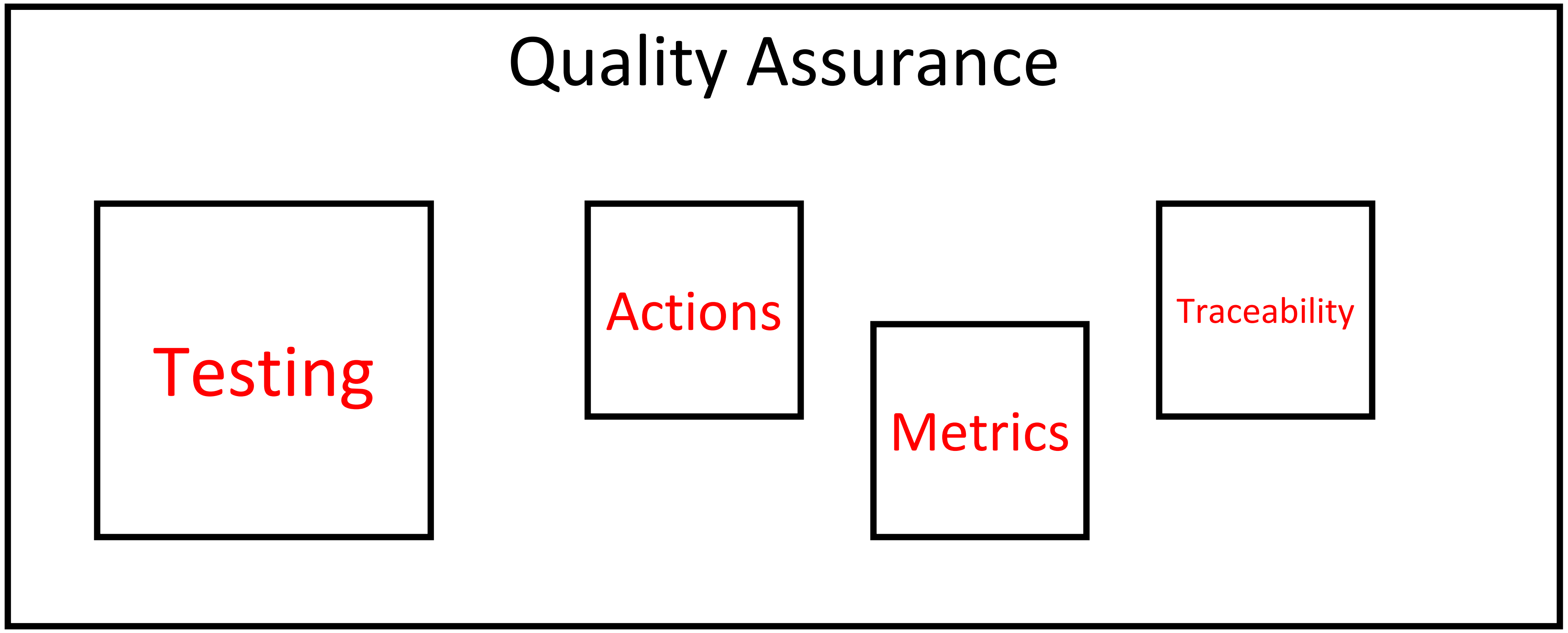
The “Right” Way?



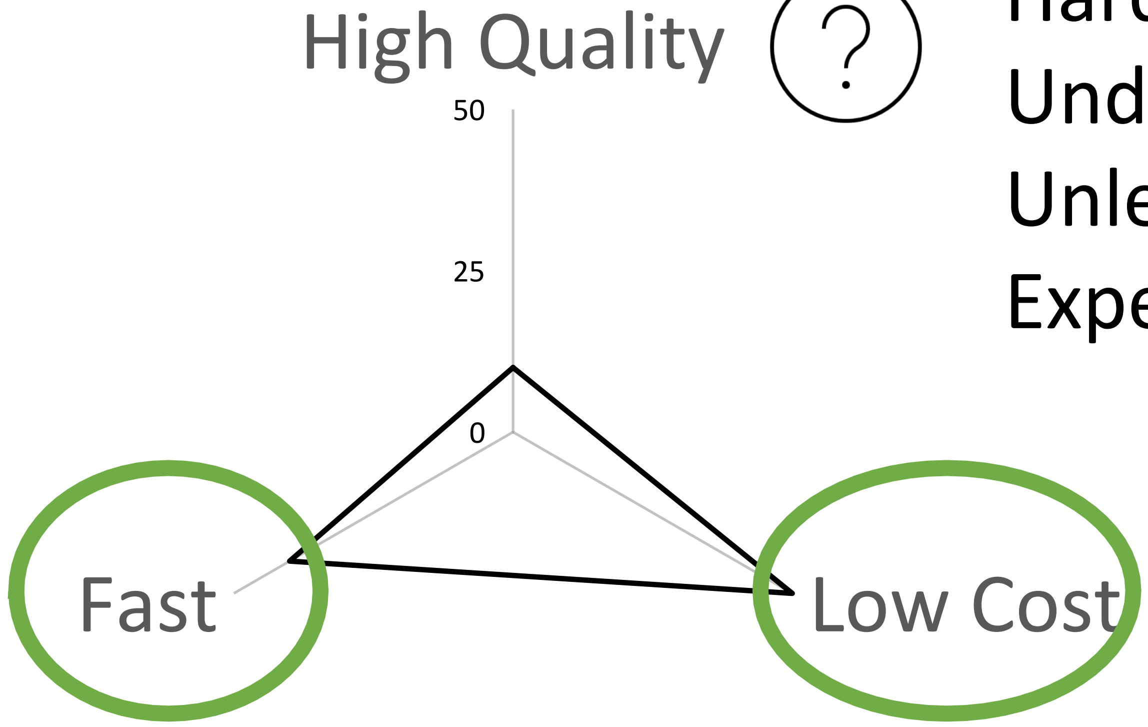
#1 Duty of Testing



Quality Assurance vs. Testing



Pick Any 2



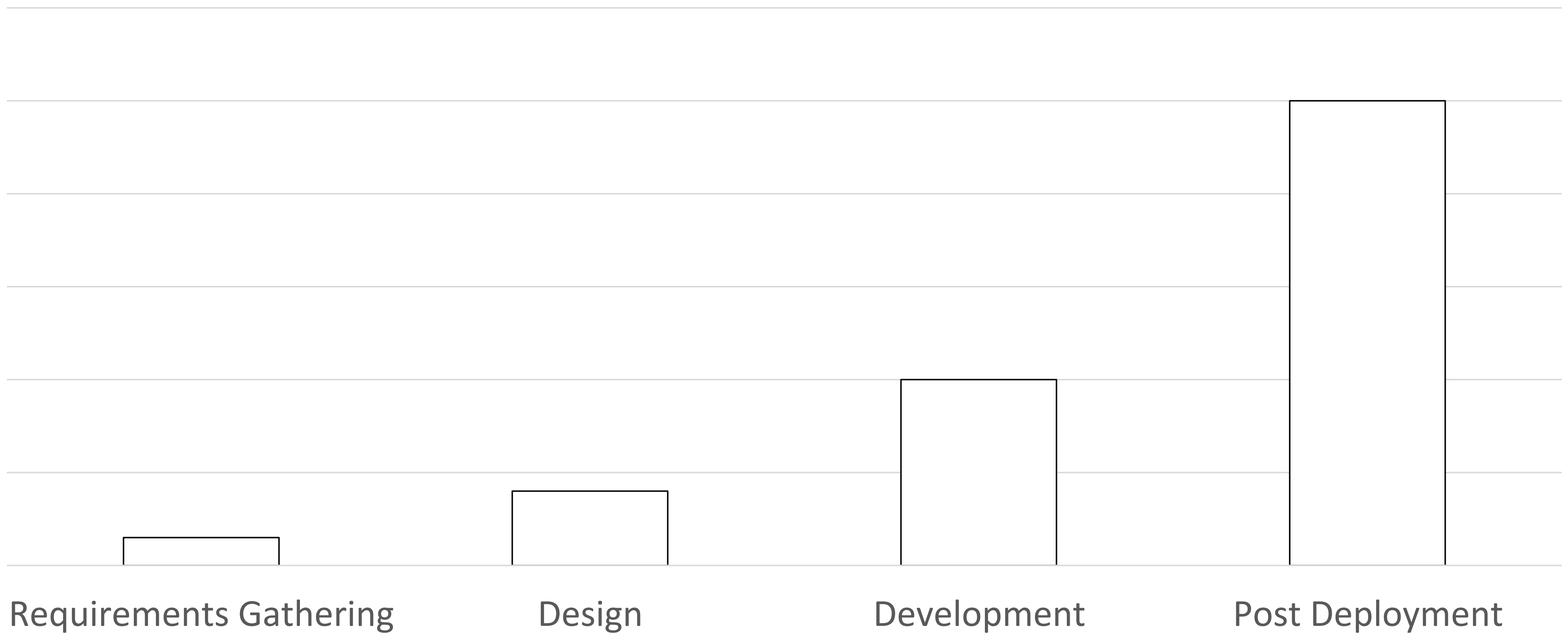
Hard to Understand Unless you are an Expert

Measure = Time

Measure = Money

Easy for Project Management to Understand

Relative “Cost” of Finding Bugs



* For Illustration Purposes Only

MVP != Low Quality

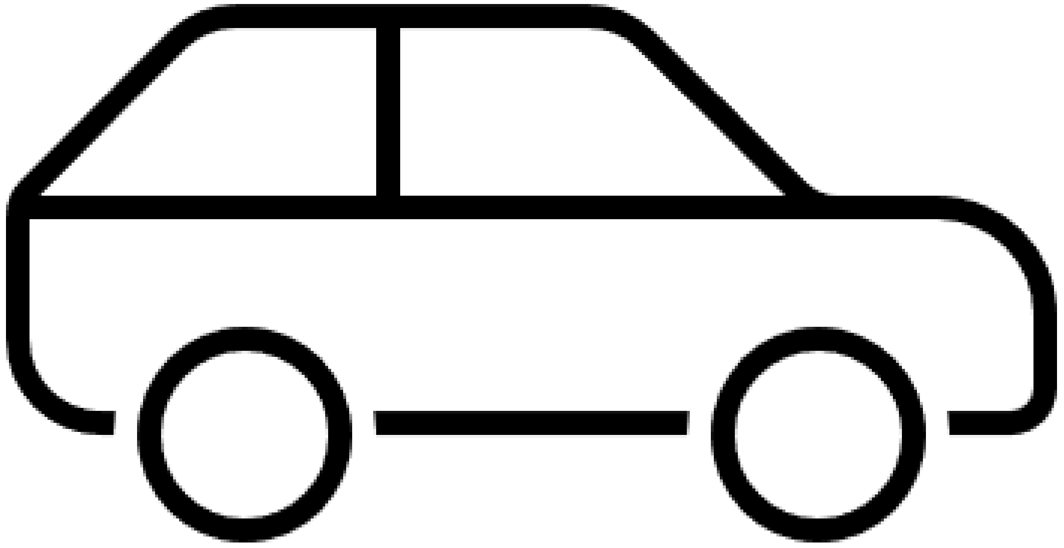
Requirement

“I want a vehicle to transport me from point A to point B”

MVP != Low Quality

MVP != Low Quality

MVP != Low Quality

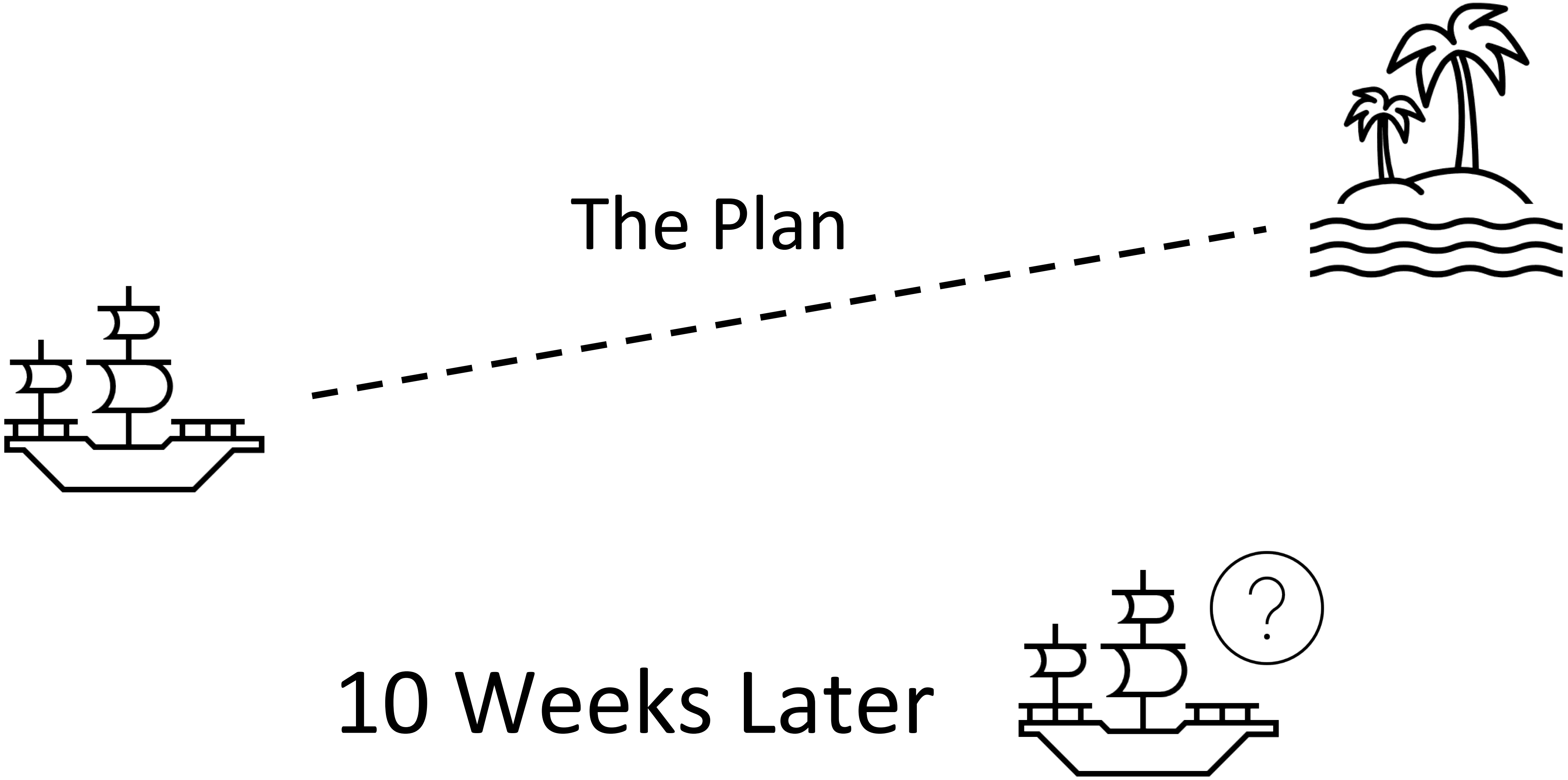


Can You Afford Testing on the Right?

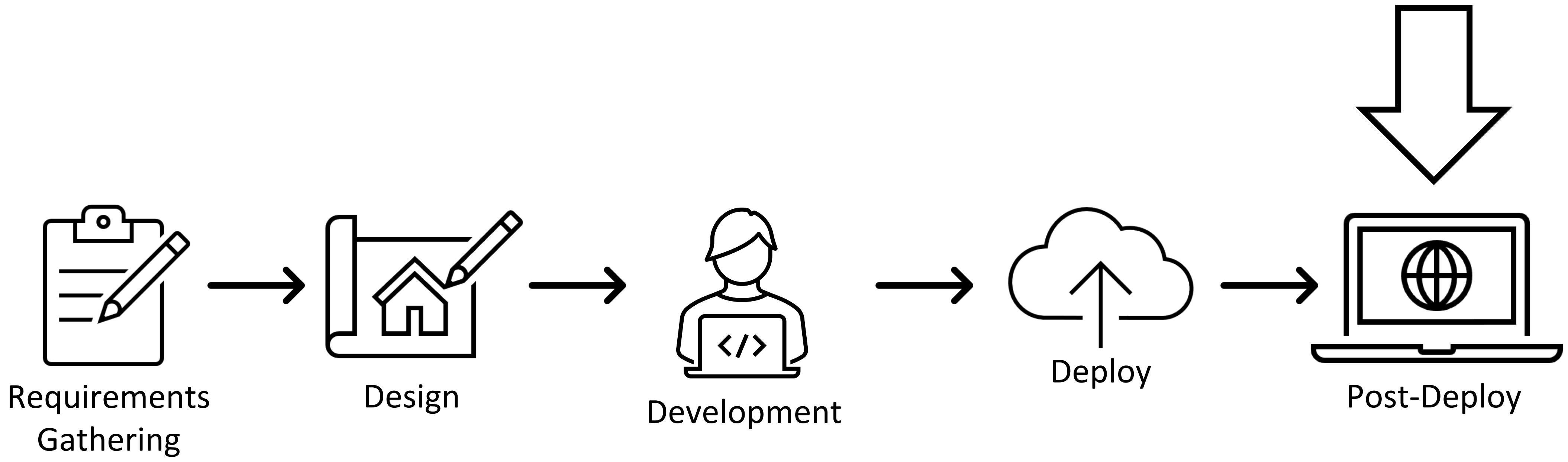
- High Technical Debt
- Difficult/Impossible to Fix Certain Bugs
- Poor Customer Confidence in Product
- Testing “Later” May Never Come
- Frustration of Poor Quality Environment Leads to Development Fatigue

- Perceived Quick to Market (Production)
- Let Your Users Act as Testers
- Low Upfront Costs

Testing Provides Feedback to Drive Agile

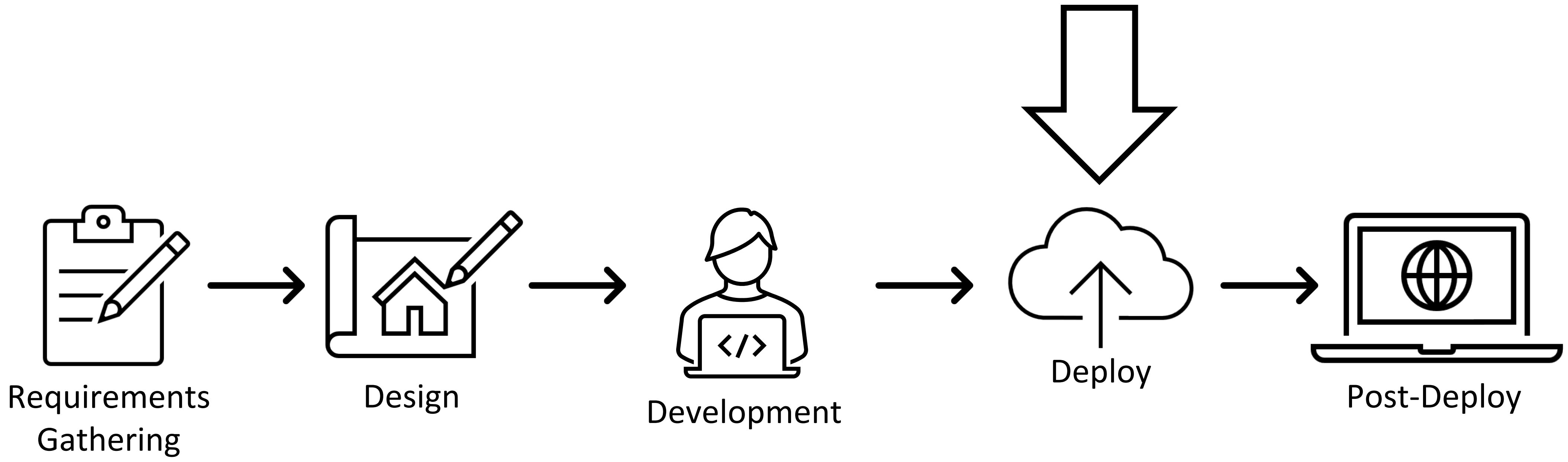


Moving Left



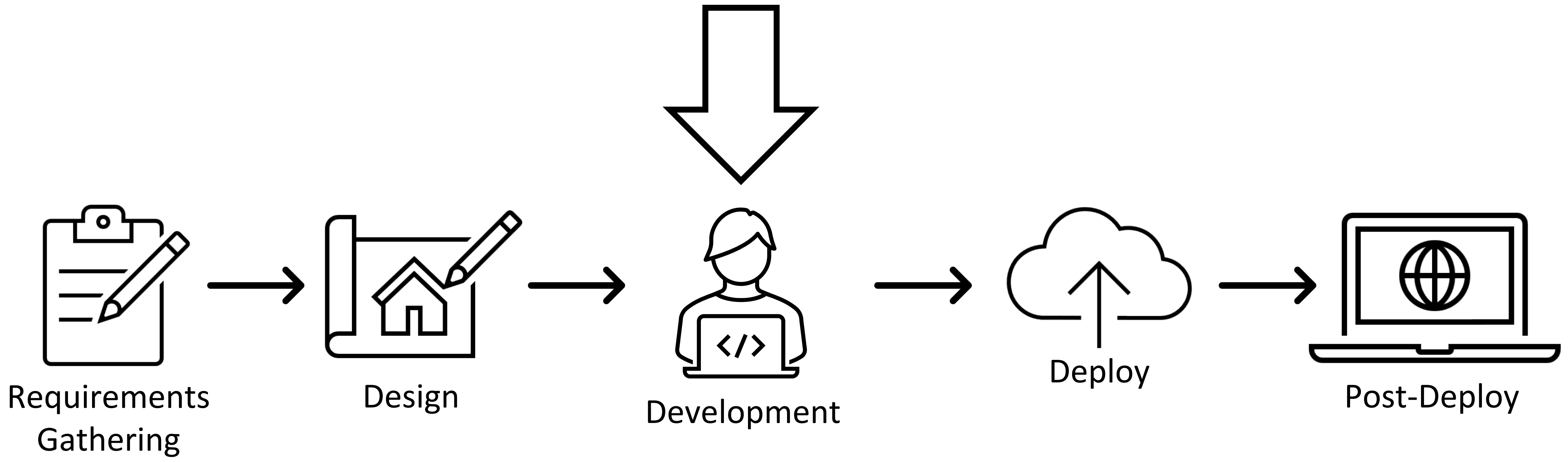
Environment Testing / Functional Regression Testing / UAT

Moving Left



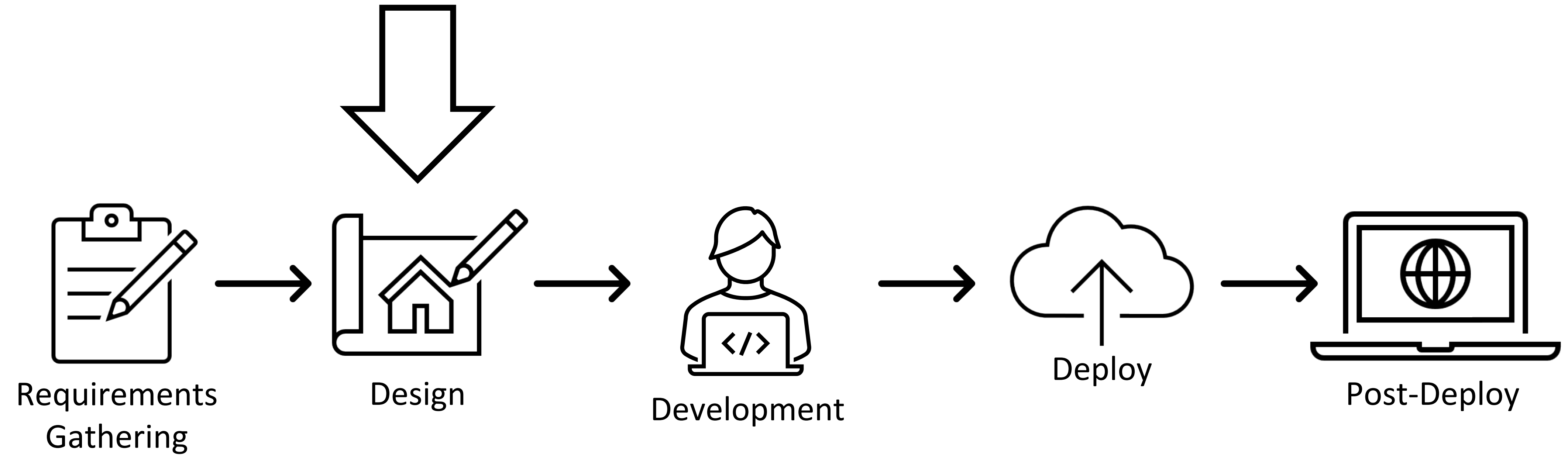
Continuous Deployment

Moving Left



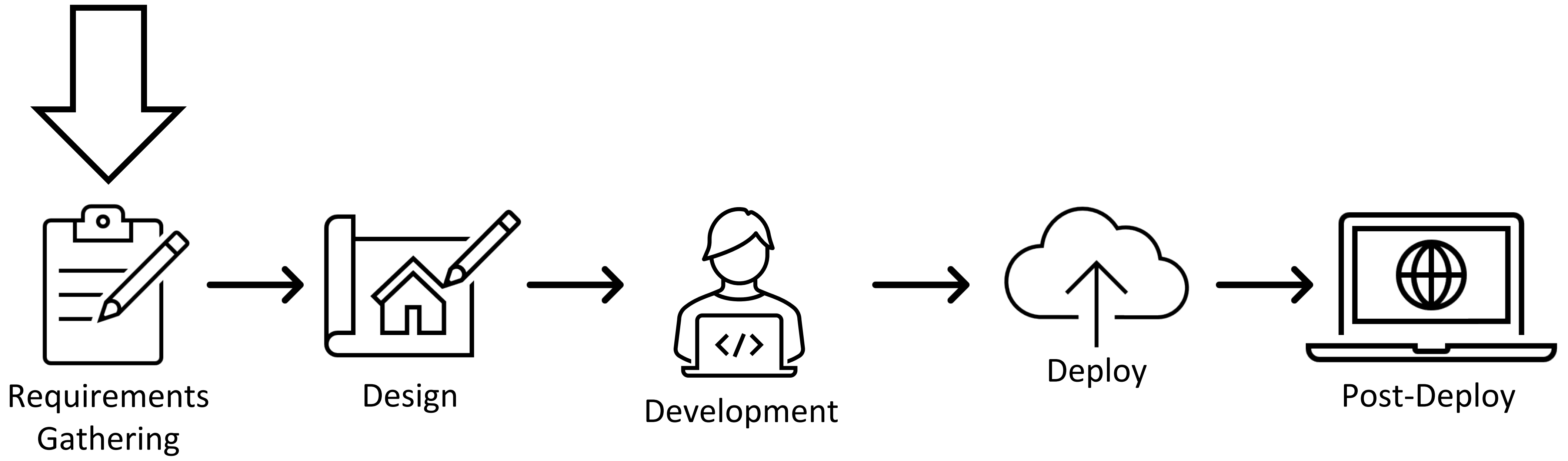
TDD / Unit Testing / Exploratory Testing / Spot-Check Functional Regression / Peer Review

Moving Left



Requirement-Based Testing Refinement / Design with Testing In-Mind

Moving Left



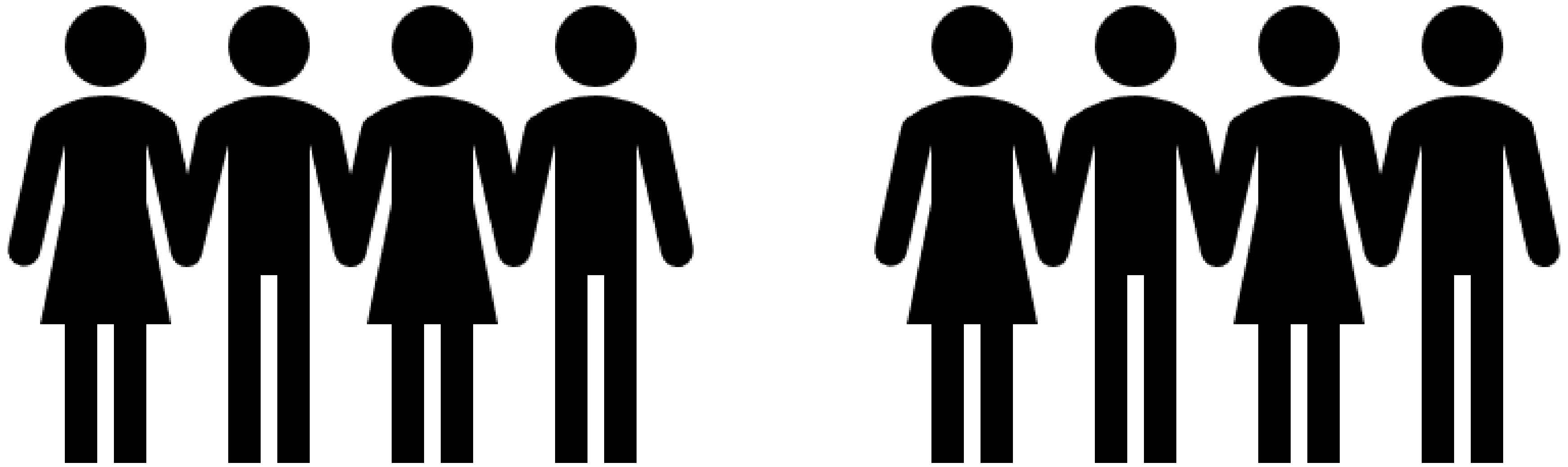
Requirement-Based Test Development

Stop Testing in Production

Test
In
Production

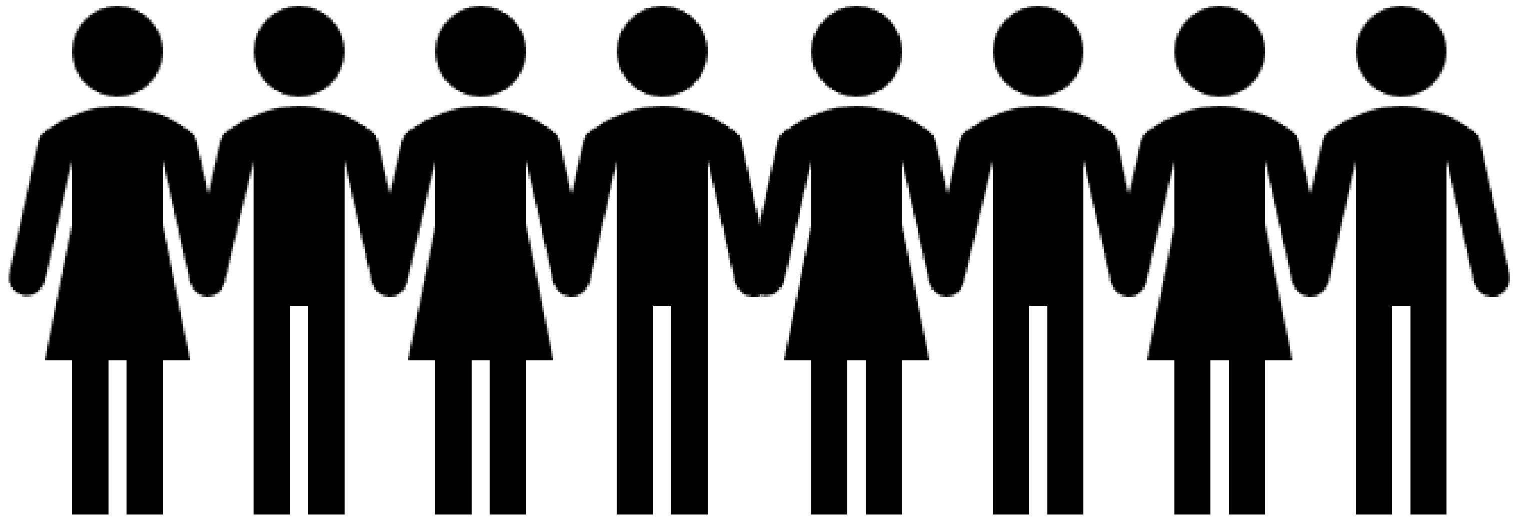
- Smoke Testing
- Synthetic Transactions
- Health Monitoring

Avoid “Us vs. Them”



QA Development

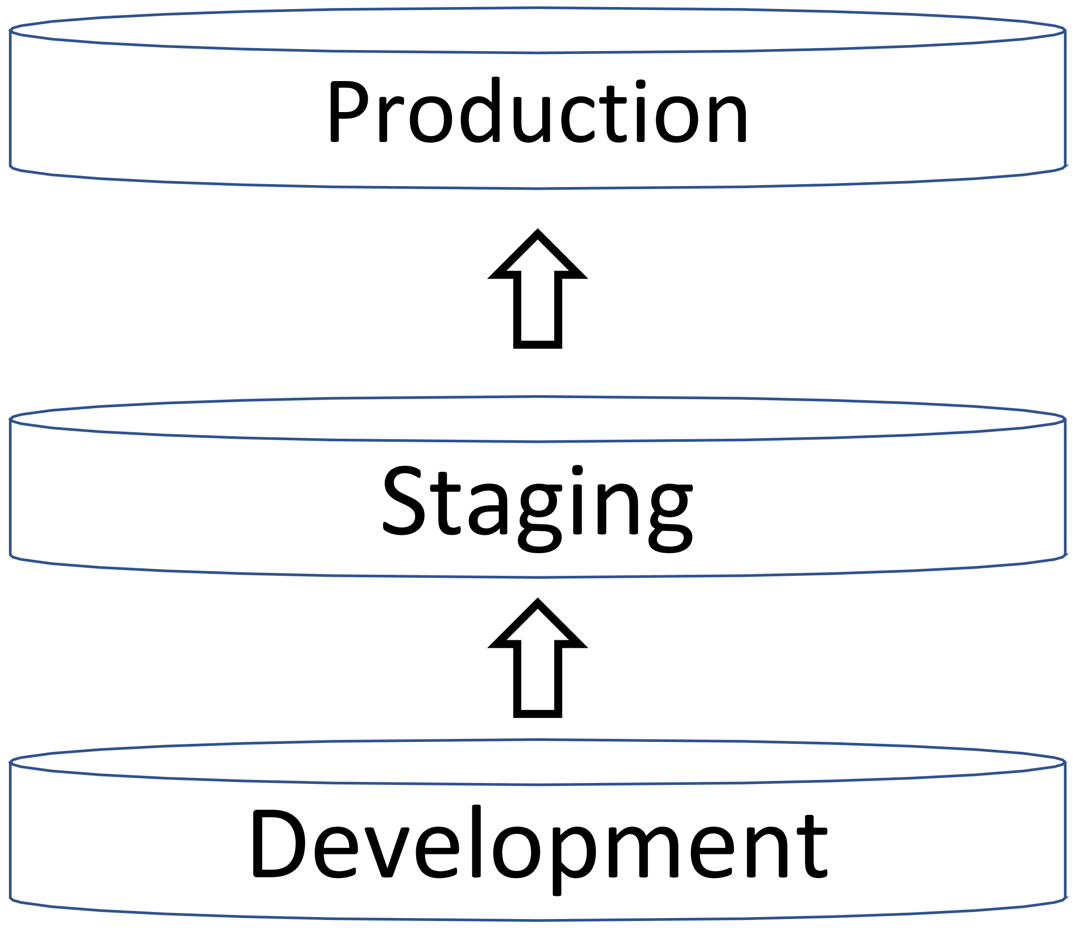
Avoid "Us vs. Them"



The Team

Use Test Management Software

Use Environment Phases



Make Automated Functional Testing Easier

Avoid Obligating Testing User Interface for Functionality

Make Automated Functional Testing Easier

Have Functionality Driven by APIs