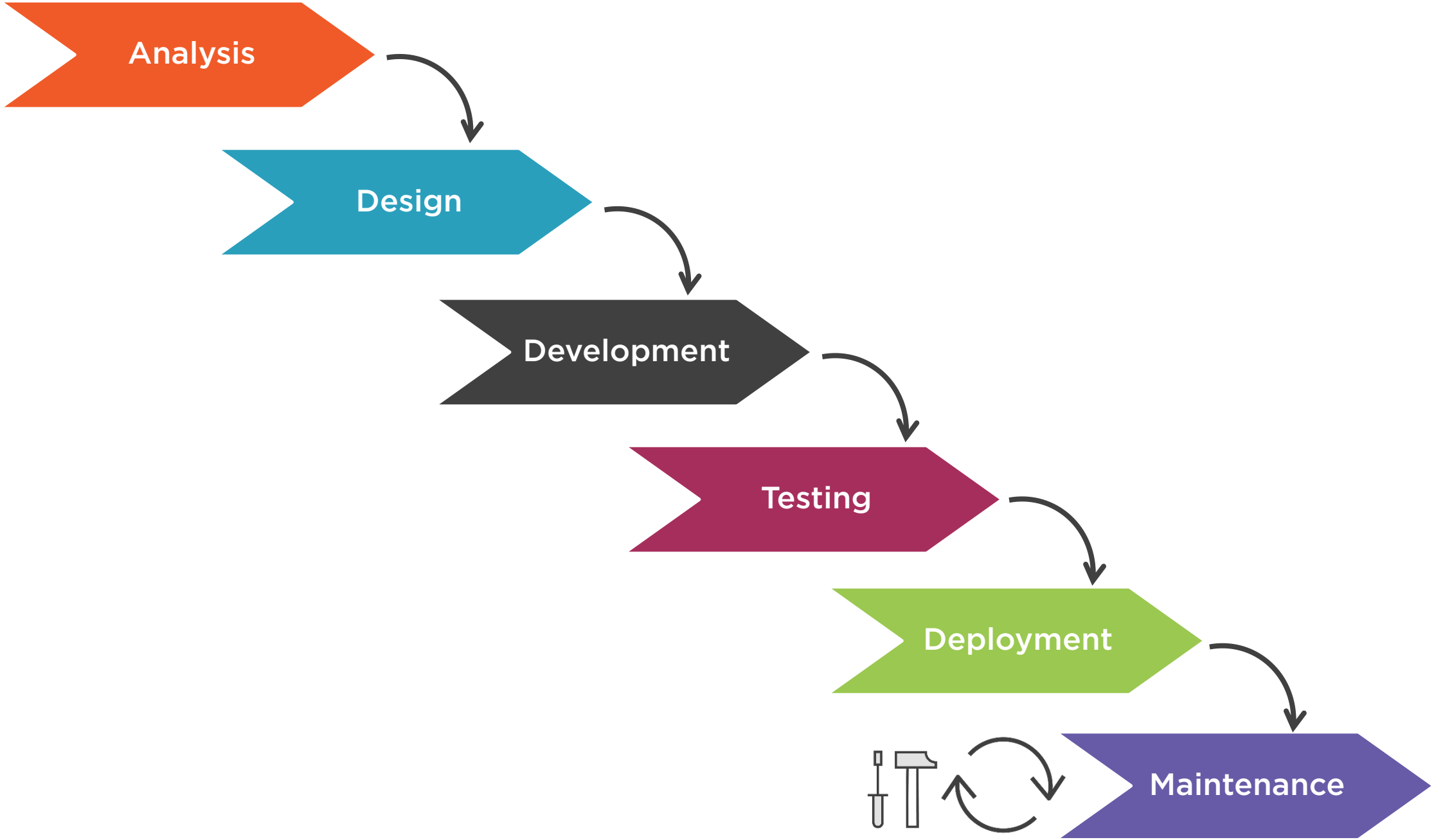
Exploring Maintenance Testing

Yelena Gouralnik
TEST AUTOMATION
ENGINEER

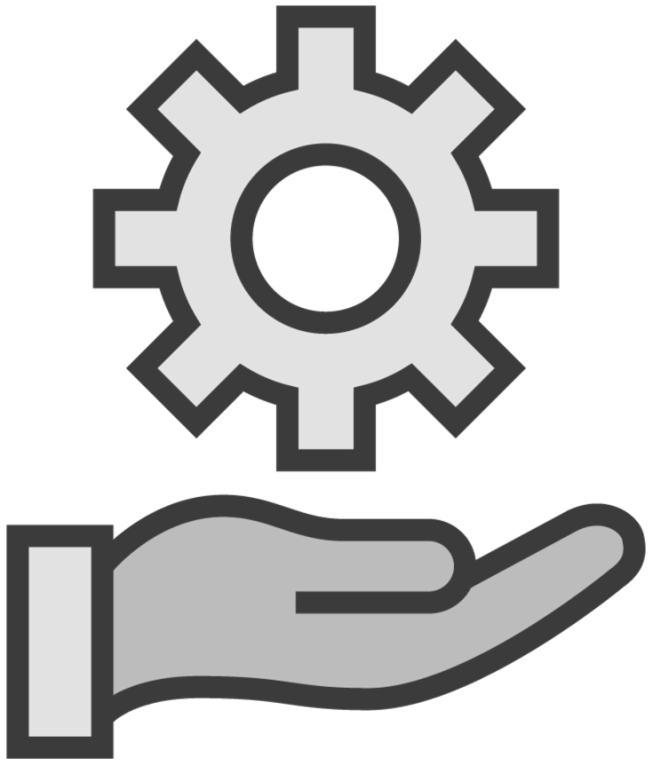

Analysis
Design
Development
Testing
Deployment
Maintenance

Overview

- Kind of maintenance
- Triggers for maintenance testing
- Impact analysis
- Course wrap up

Systems need to be maintained because changes are inevitable.

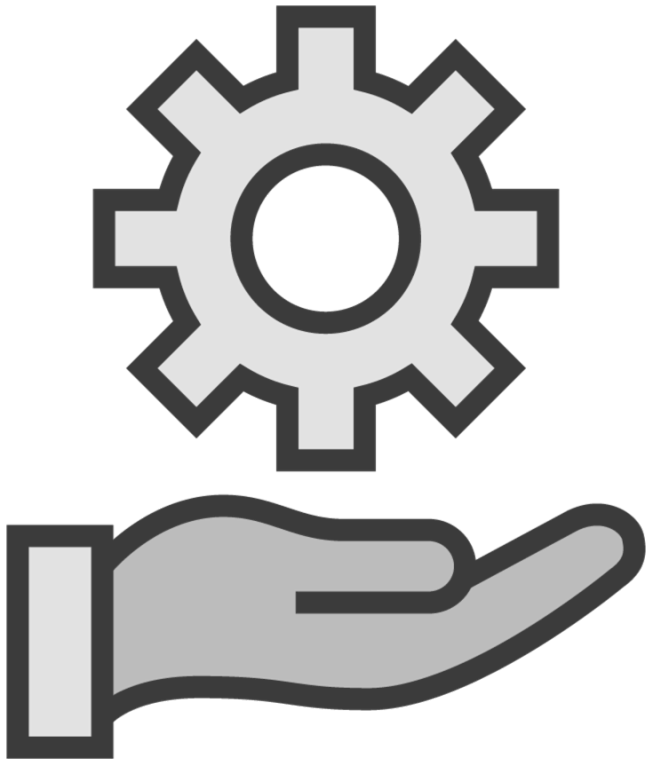
Change Categories



Unplanned

- Bugs or failures in production
- A “hotfix” required

Change Categories



Planned

- Software enhancements
- Operational and environmental upgrades
  - Example: SQL Server migration to a newer version
- Retirement
  - SW not fit for purpose anymore
  - Rewriting the SW using newer tech

Impact Analysis

Evaluates the changes made to identify consequences and potential side-effects

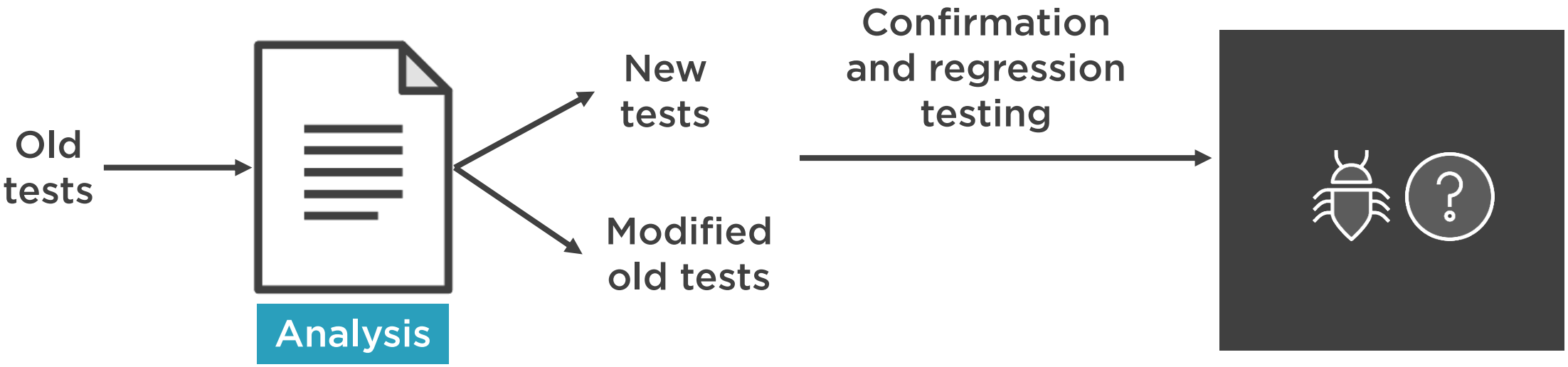
Impact Analysis

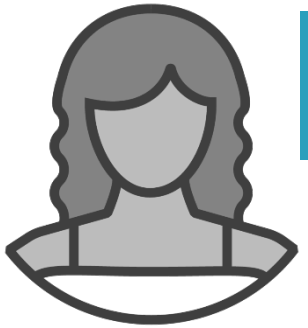
What has changed?

Where in the system?

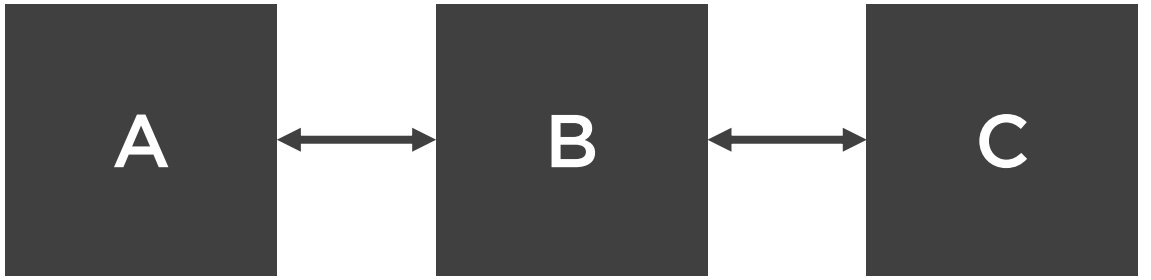
Which parts are definitely affected?

Which parts are likely to be affected?


Old
tests
Analysis
New
tests
Modified
old tests
Confirmation
and regression
testing



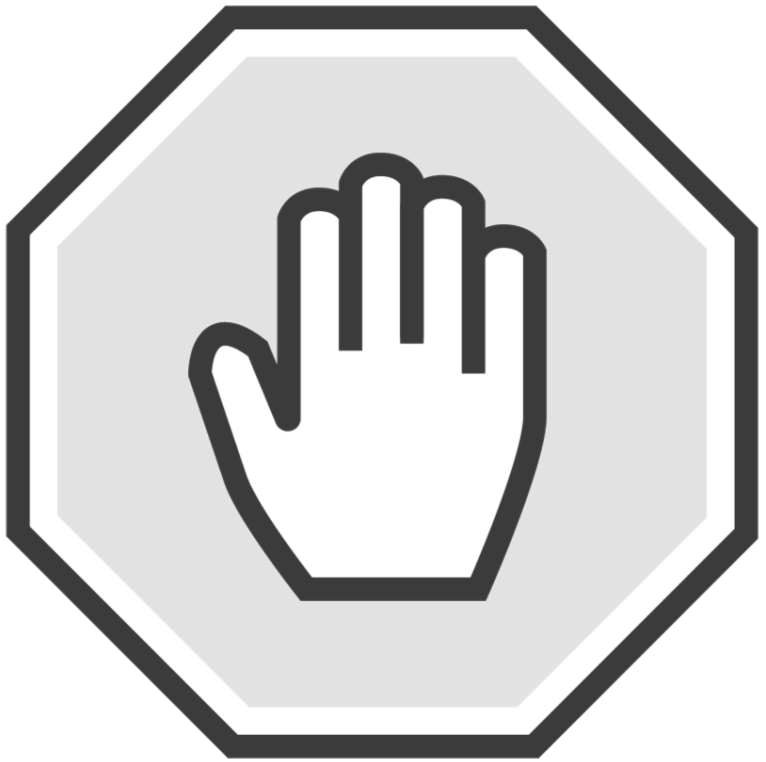
I need to test this, but I only have 2 days...



Prioritize:

1) Test Module A (the new feature)
2) In the past 2 months:
   - Module B had 10 bugs ← Focus here!
   - Module C had 2 bugs
3) Test Module C if you have spare time

Impact Analysis Challenges



- Outdated specifications
- Lack of knowledge
- Outdated test cases with no traceability
- Weak or non-existent tool support

Summary

SDLC: Software Development Lifecycle

Multiple models exist - sequential and iterative

Systems are layered

Levels: unit, integration, system, acceptance

Types: functional, non-functional, black-box, white-box, etc.

Testing never stops - maintenance testing is a must