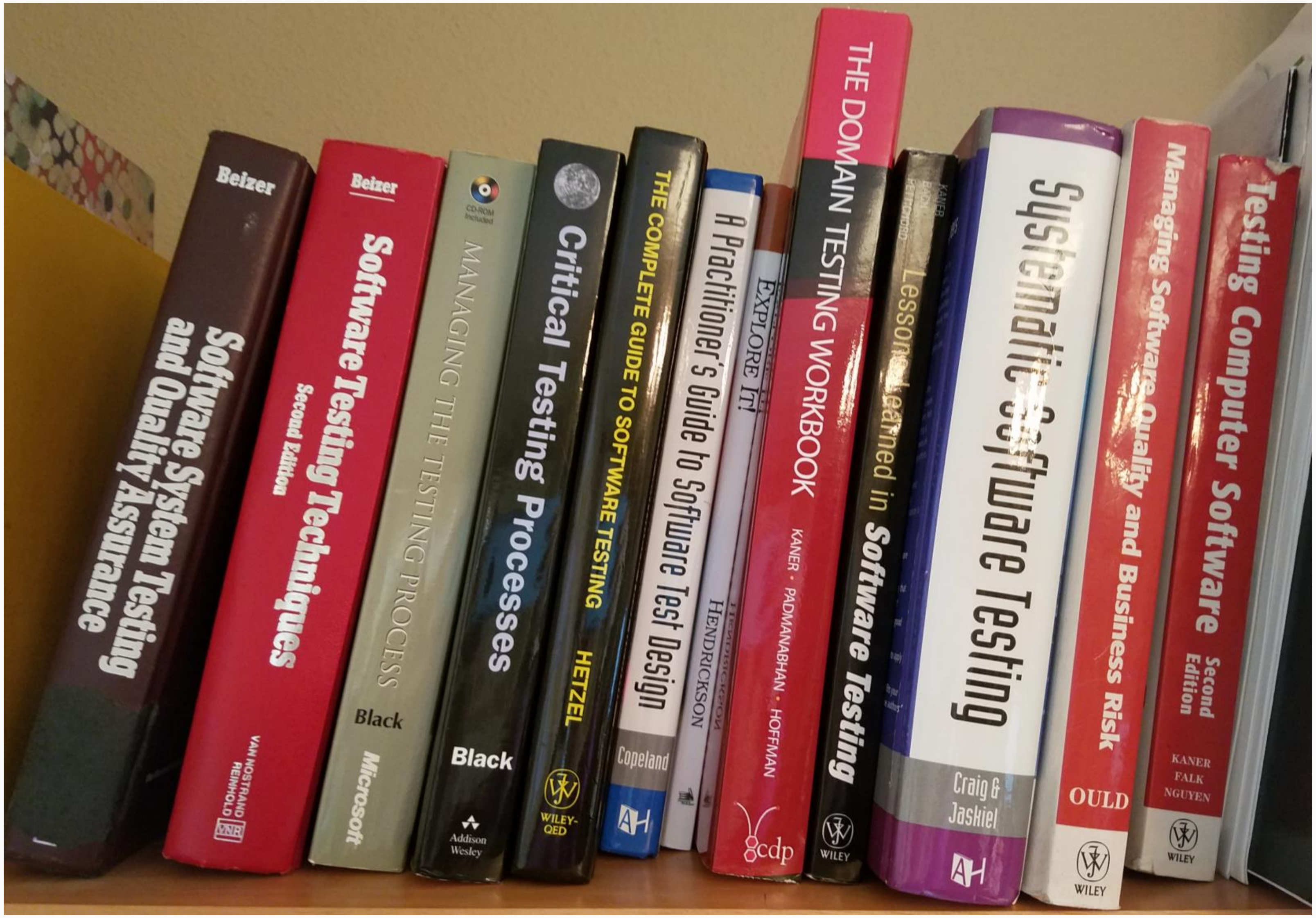
Building In Quality

Ten Years Later


Beizer
Software System Testing and Quality Assurance
Beizer
Software Testing Techniques
Second Edition
VAN NOSTRAND REINHOLD
MANAGING THE TESTING PROCESS
Black
Microsoft
Critical Testing Processes
Black
Addison Wesley
THE COMPLETE GUIDE TO SOFTWARE TESTING
HETZEL
WILEY-QED
A Practitioner's Guide to Software Test Design
Copeland
EXPLORE IT!
HENDRICKSON
THE DOMAIN TESTING WORKBOOK
KANER · PADMANABHAN · HOFFMAN
cdp
Lessons Learned in Software Testing
WILEY
Systematic Software Testing
Craig & Jaskiel
Managing Software Quality and Business Risk
OULD
WILEY
Testing Computer Software
Second Edition
KANER FALK NGUYEN
WILEY

PRINCIPLES OF SOFTWARE ENGINEERING MANAGEMENT
TOM GILB
19246
GLASS
BUILDING QUALITY SOFTWARE
PRENTICE HALL
Managing Software Quality and Business Risk
OULD
WILEY

Some software just has to work

By Lothar Neumann, Gernsbach [1] - Karlsruhe:Bild:Philippsburg2.jpg, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=10060036

NASA photo

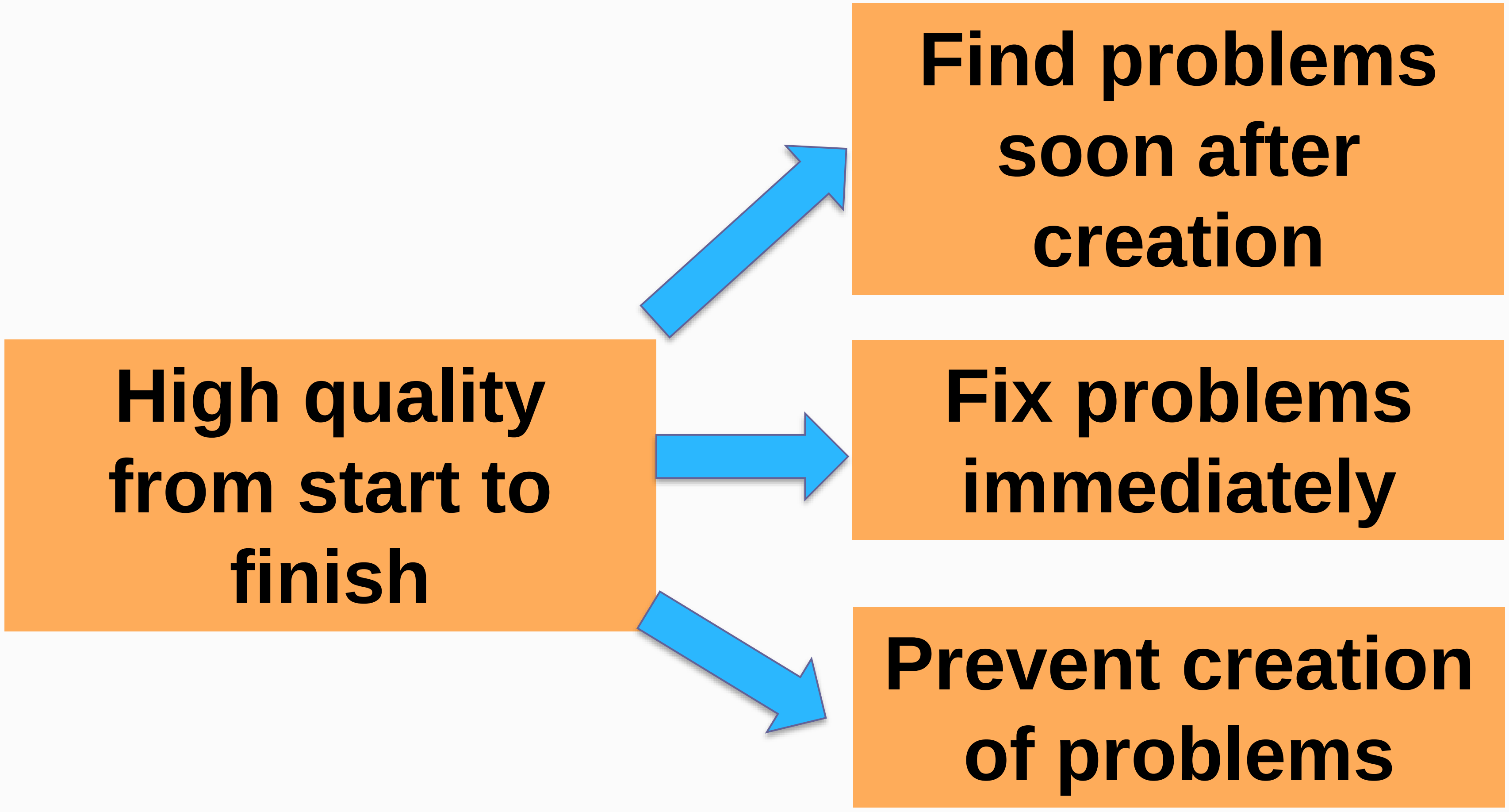
Who Owns Quality?

Fundamental beliefs in high-reliability fields:

- Developers are responsible for quality.
- Top management is responsible for quality (ISO 9000)
- Test group is not responsible for creating quality, simply for assessing it.

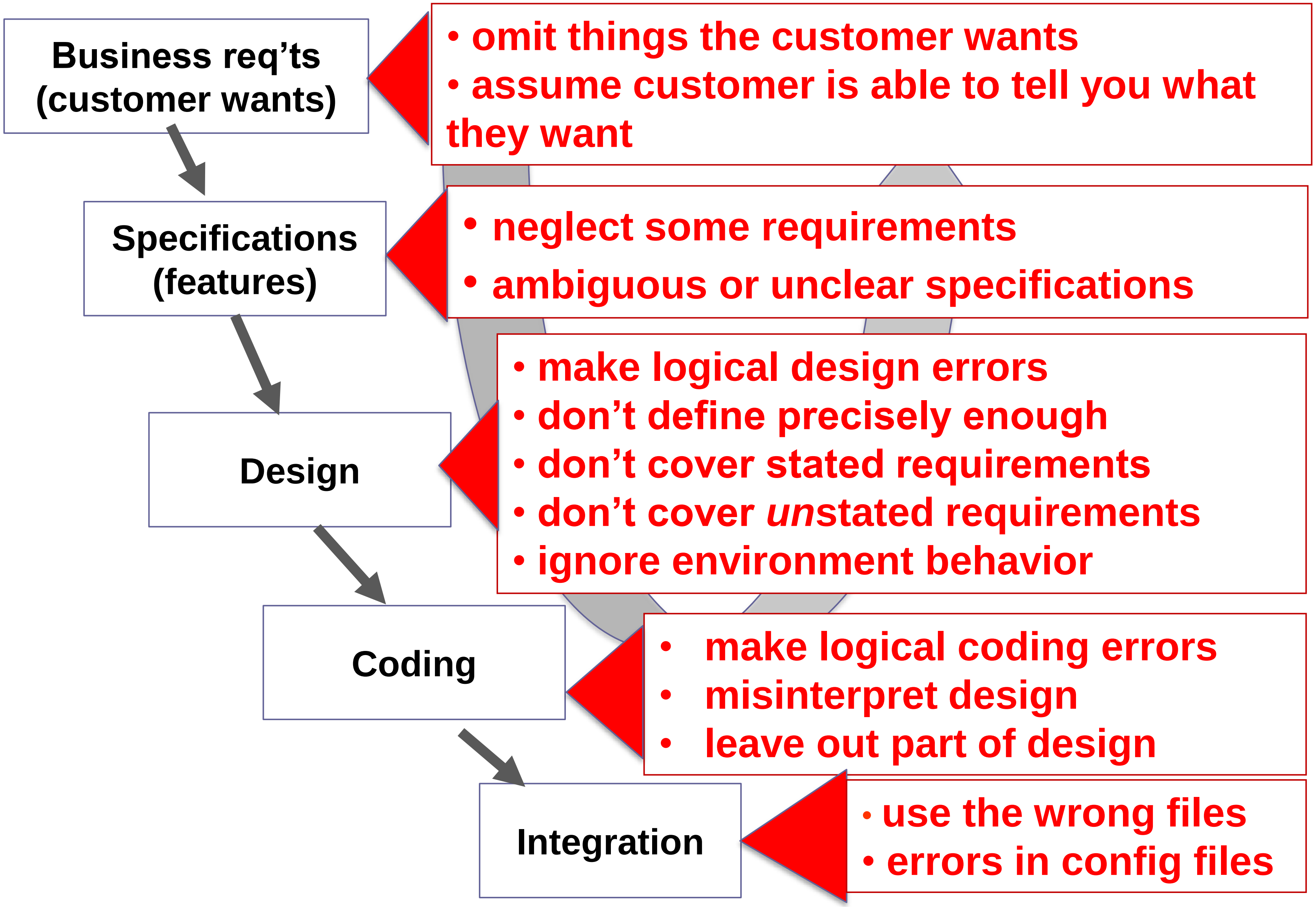
© 2017 Iberle Consulting

Quality Strategies in High-Reliability Software


High quality from start to finish
Find problems soon after creation
Fix problems immediately
Prevent creation of problems

© 2017 Iberle Consulting

How Defects Are Created



Business req'ts (customer wants)
- omit things the customer wants
- assume customer is able to tell you what they want

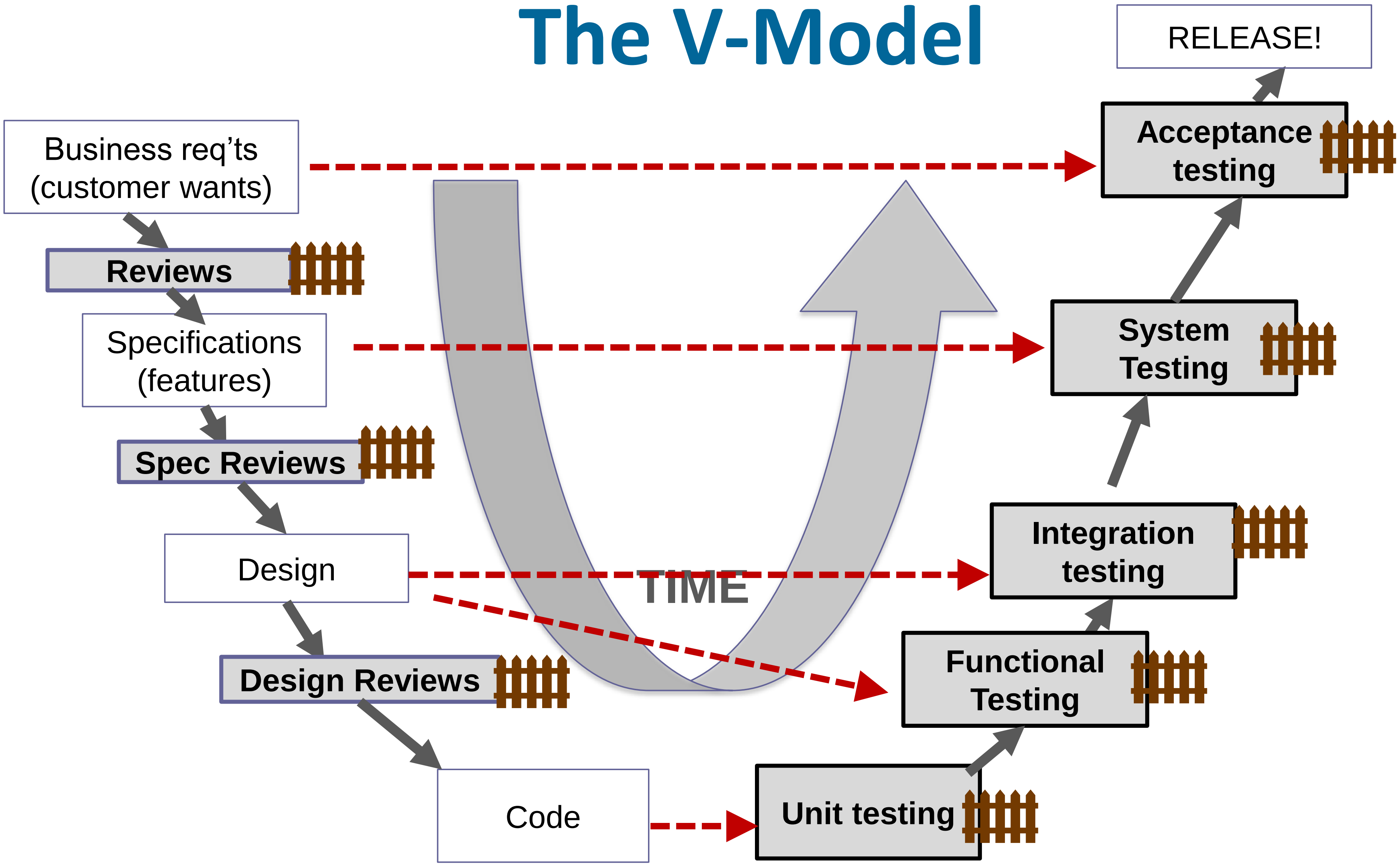
Specifications (features)
- neglect some requirements
- ambiguous or unclear specifications

Design
- make logical design errors
- don't define precisely enough
- don't cover stated requirements
- don't cover *un*stated requirements
- ignore environment behavior

Coding
- make logical coding errors
- misinterpret design
- leave out part of design

Integration
- use the wrong files
- errors in config files

The V-Model


RELEASE!
Business req'ts (customer wants)
Reviews
Specifications (features)
Spec Reviews
Design
Design Reviews
Code
Acceptance testing
System Testing
Integration testing
Functional Testing
Unit testing
TIME

Quality Gate is a Set of Exit Criteria

Typical quality gate in medical products during 1990s:

- ❑ Complies with coding standards
- ❑ Compiles with no flags level 3 or above.
- ❑ Cyclomatic complexity at or below 20
- ❑ Modules with complexity above 10 have been peer-reviewed
- ❑ Unit testing completed with 100% path coverage
- ❑ Passes all unit tests
- ❑ Integration test scripts written

A High-Reliability Definition of Done

Checklist for Technical Completeness for User Stories (Herman 2016)

Design:

- ❑ Design covers everything in the user story and acceptance criteria.
- ❑ Design reviewed by area experts and feedback is incorporated.
- ❑ User story has a link to the design.

Code:

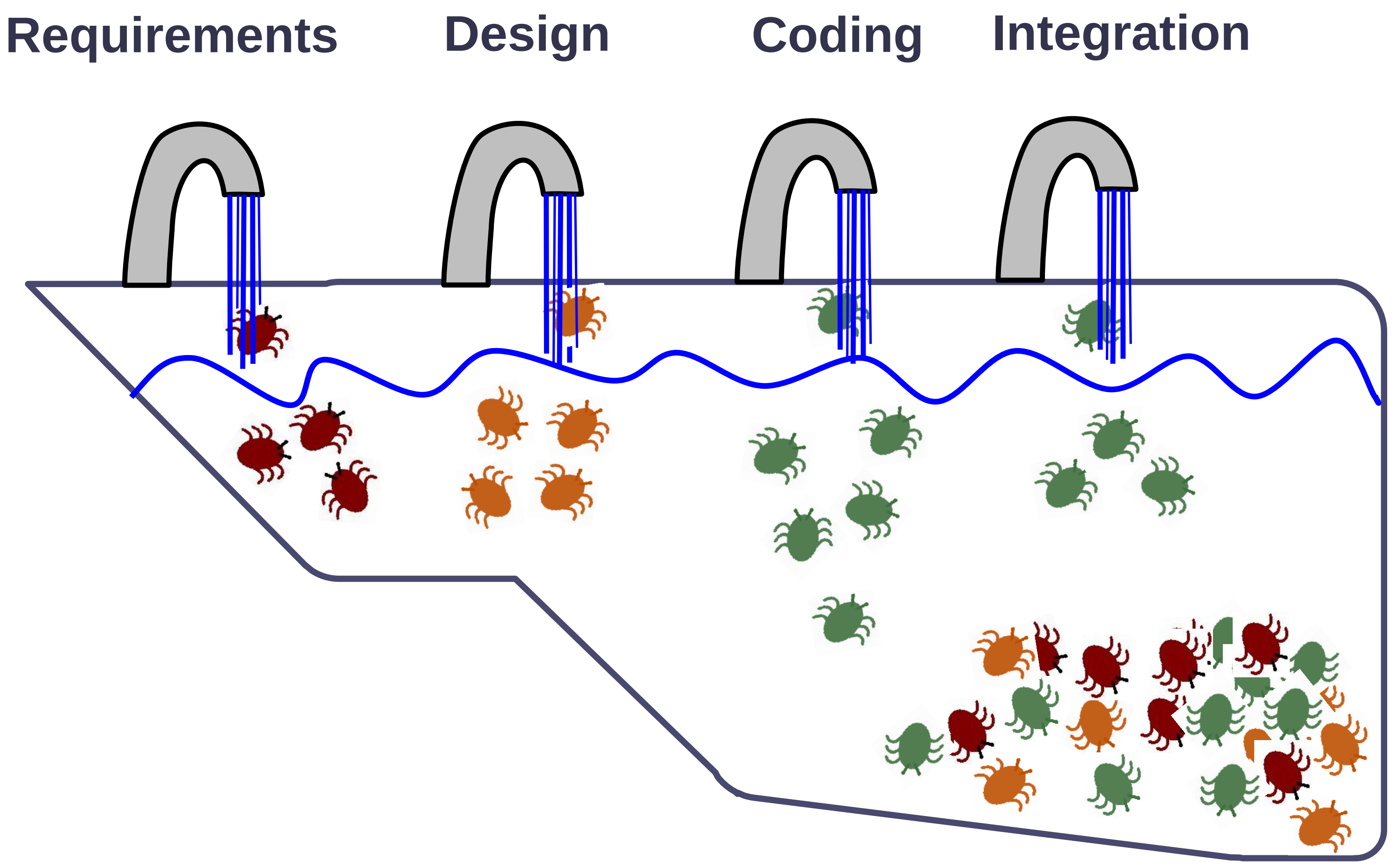
- ❑ Code implements the design.
- ❑ Unit tests cover the design (includes use cases, API contracts).
- ❑ Code compiles and runs, on the build server, without errors, warnings, or unit test failures.
- ❑ Code and unit tests have been peer reviewed and adjustments made per comments.
- ❑ No new defects.

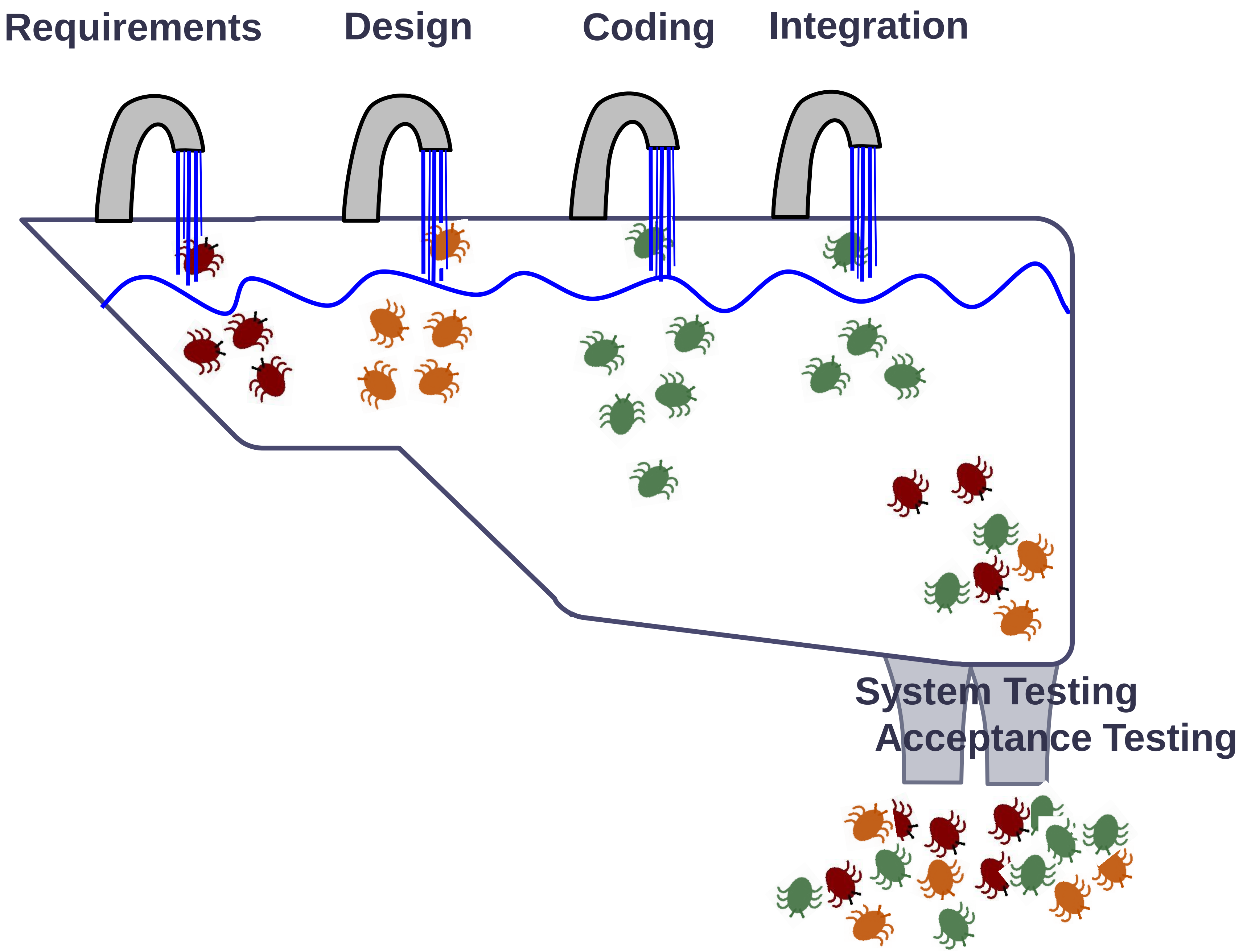
Acceptance Testing

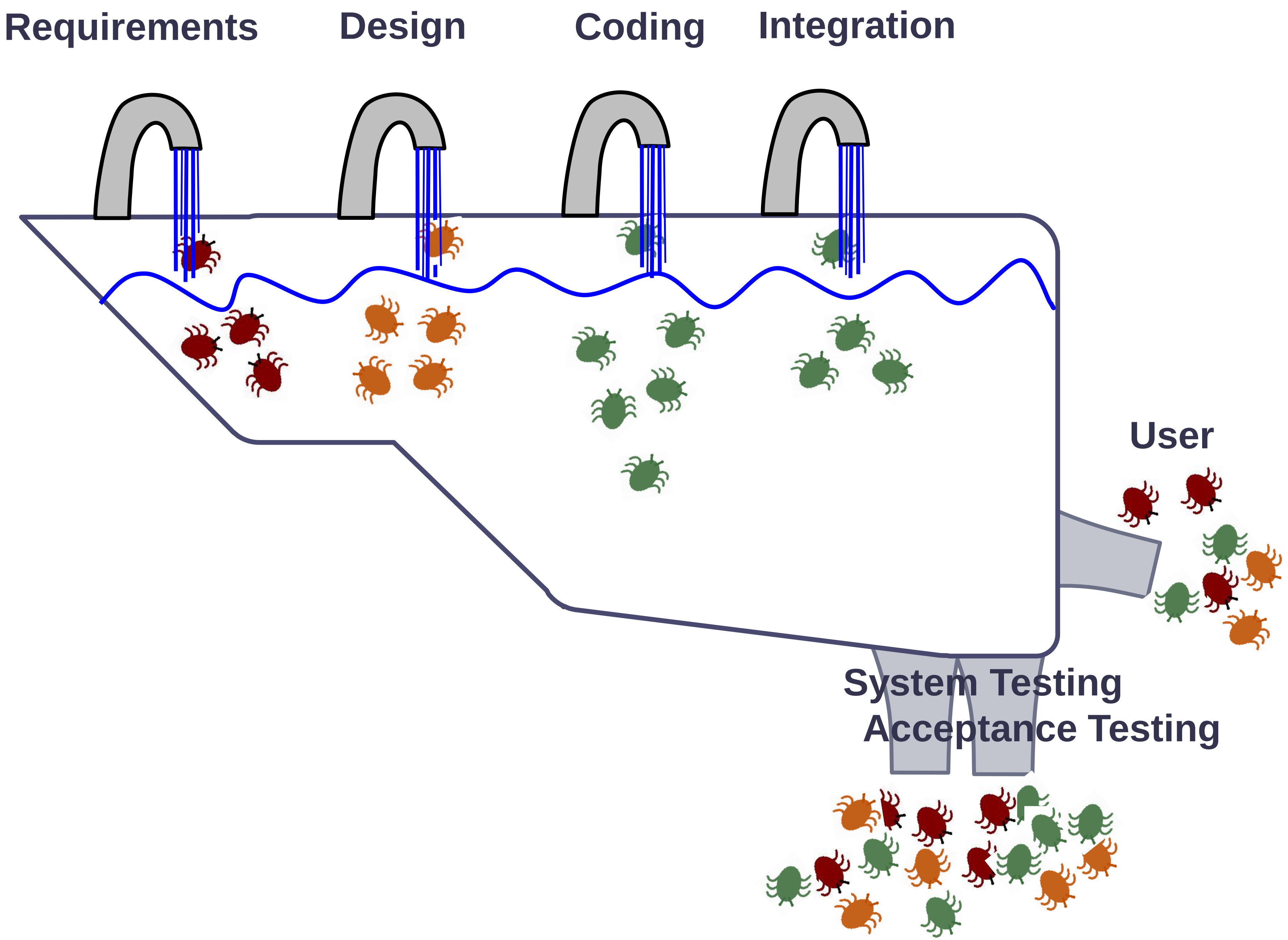
- ❑ Acceptance tests in a form listed below have been written and entered into project management system
  - ❑ May include manual, automated, and unit or integration tests
  - ❑ Verification Procedure and/or SMART for high risk stories, SMART for stories with medium risk
- ❑ Acceptance tests have been reviewed by a developer and feedback has been incorporated.
- ❑ Acceptance tests pass on a branch or main build; unresolved issues found on main build have been logged into defect tracking system.
- ❑ Executed results have been attached to project management system.

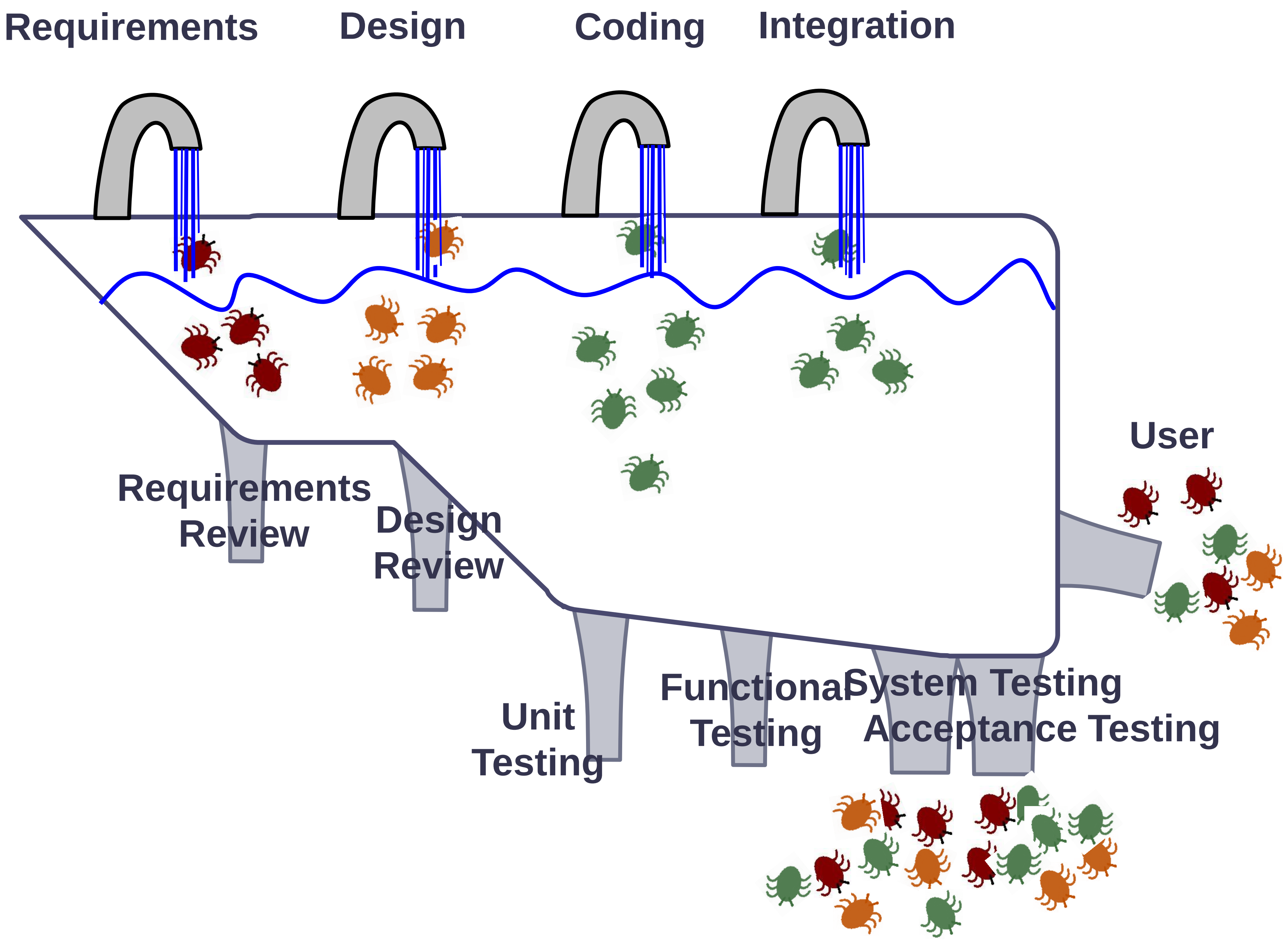
Defect Fixing:

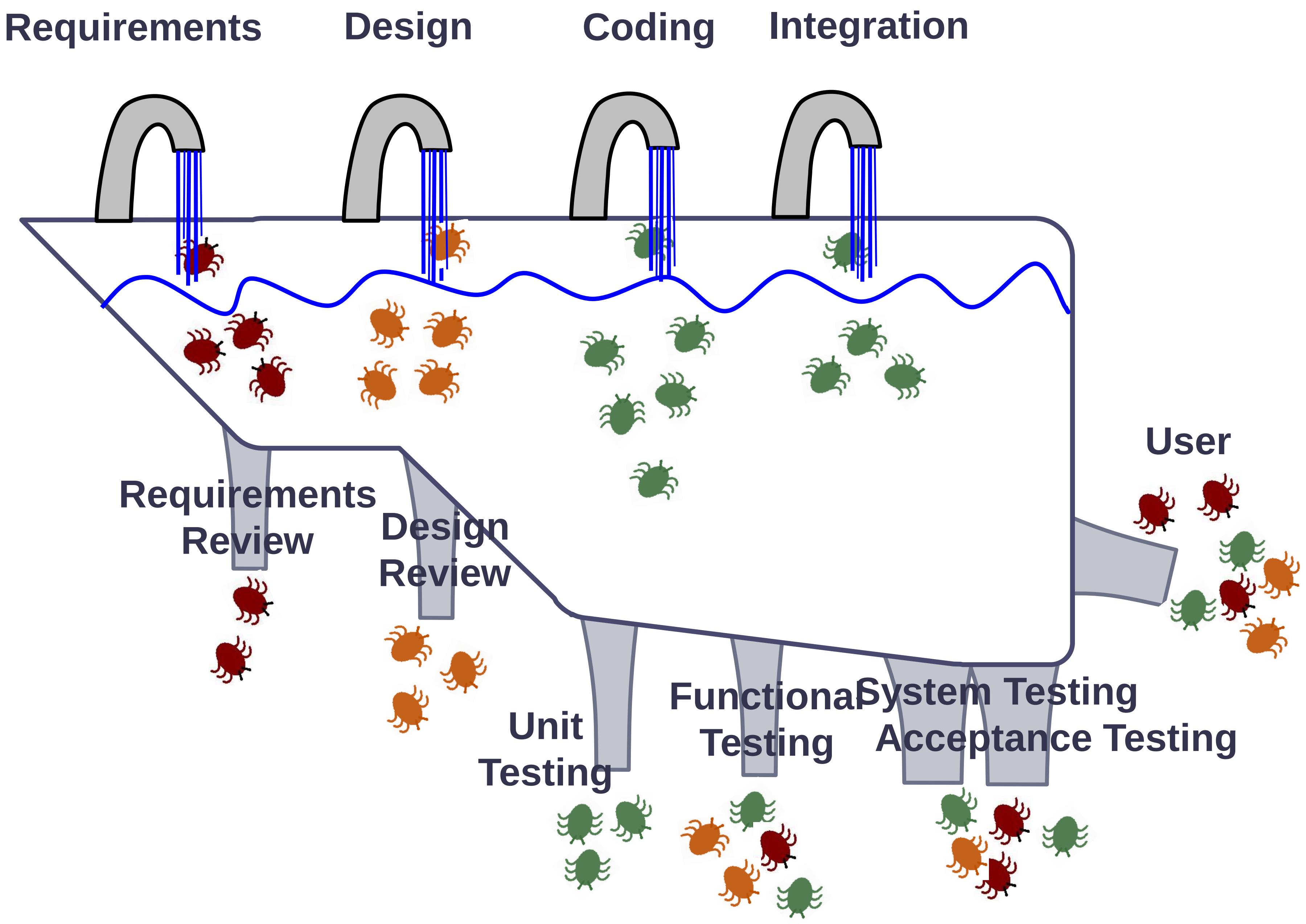
- ❑ Minimal steps to reproduce are documented in the defect description.
- ❑ Root cause analysis is documented in the defect description.
- ❑ Fix approach is documented in the defect description.

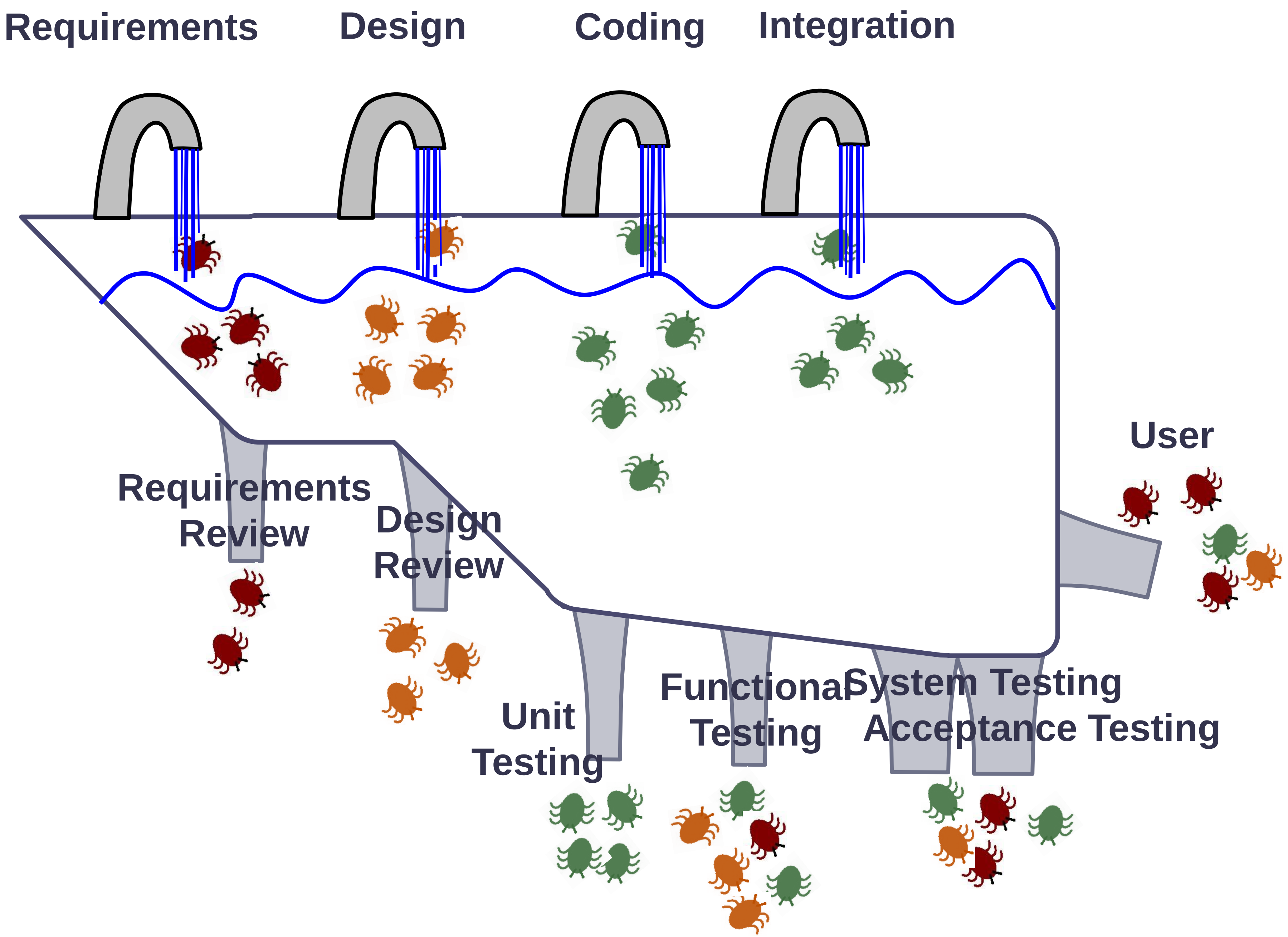

Requirements
Design
Coding
Integration

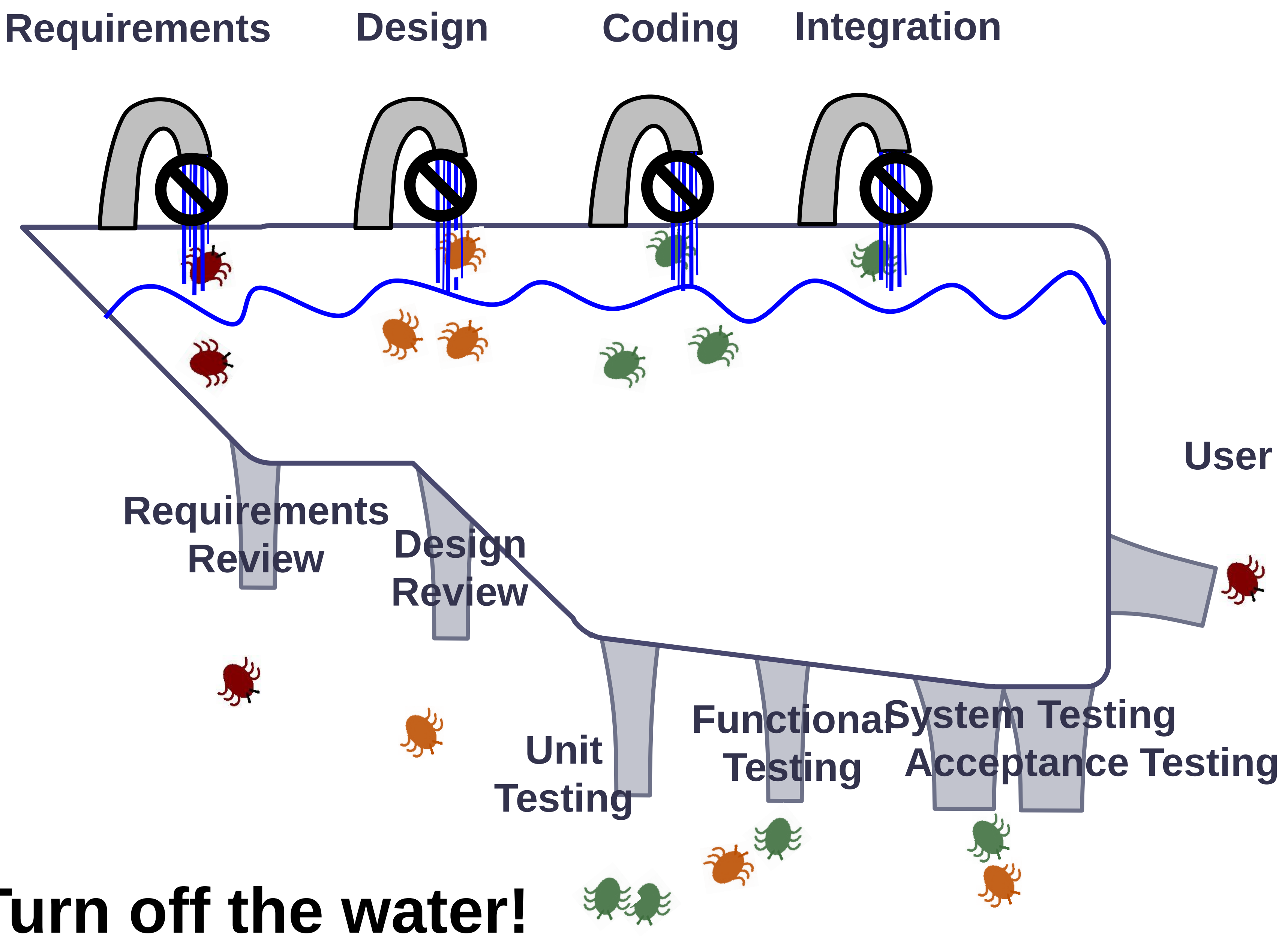

Requirements
Design
Coding
Integration
System Testing
Acceptance Testing


Requirements
Design
Coding
Integration
User
System Testing
Acceptance Testing


Requirements
Design
Coding
Integration
Requirements Review
Design Review
Unit Testing
Functional Testing
System Testing
Acceptance Testing
User


Requirements
Design
Coding
Integration
Requirements Review
Design Review
Unit Testing
Functional Testing
System Testing
Acceptance Testing
User


Requirements
Design
Coding
Integration
User
Requirements Review
Design Review
Unit Testing
Functional Testing
System Testing
Acceptance Testing



Turn off the water!

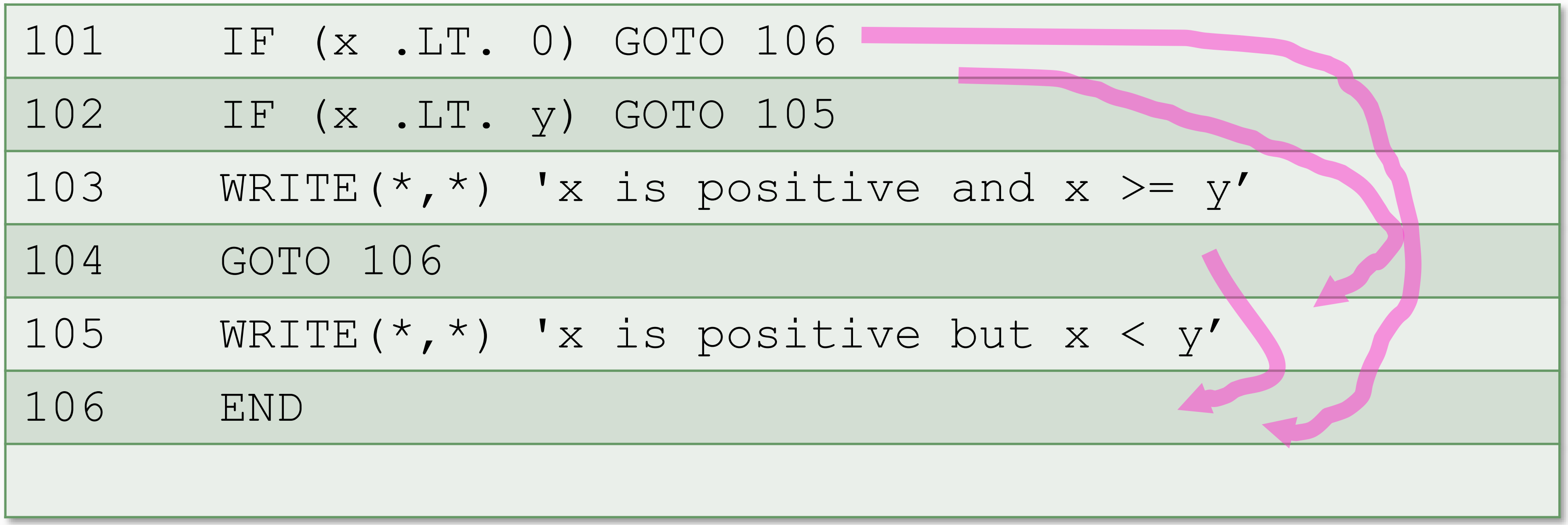
Strategies for Preventing Defects

1) Make the structure & logic visible
2) Mistake-proof with tools
3) Maintain intellectual control
4) Know your domain
5) Design away the opportunity for error

Make the Logic Visible

```
106     END
105     WRITE(*,*) 'x is positive but x < y’
104     GOTO 105
103     WRITE(*,*) 'x is positive and x >= y’
102     IF (X .LT. Y) GOTO 105
101     IF (X .LT. 0) GOTO 106
```

Better Visibility



```
101      IF (x .LT. 0) GOTO 106
102      IF (x .LT. y) GOTO 105
103      WRITE(*,*) 'x is positive and x >= y’
104      GOTO 106
105      WRITE(*,*) 'x is positive but x < y’
106      END
```

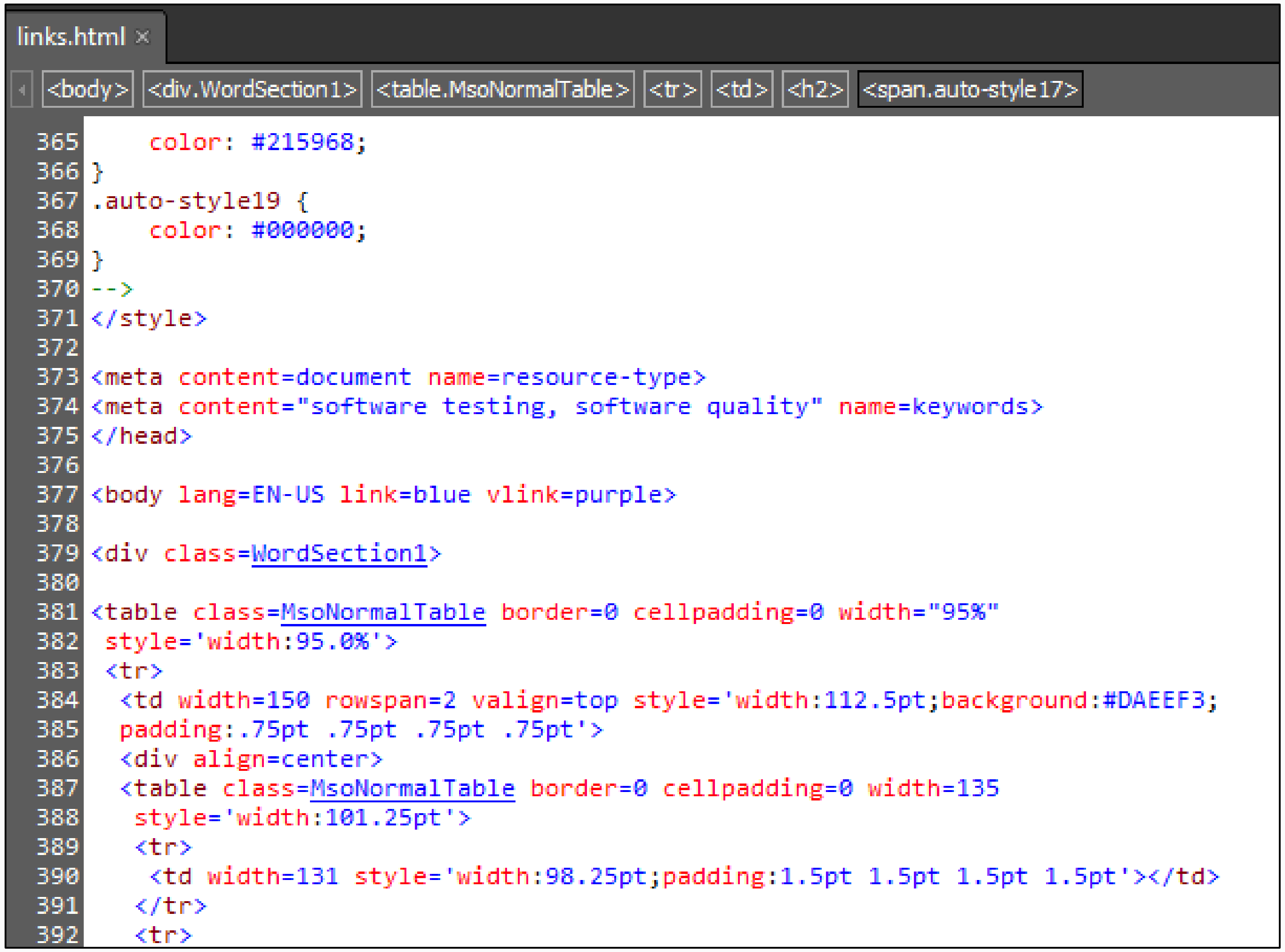
Mistake-Proof with Tools

```
IF (x .GT. 0) THEN
IF (x .GE. y) THEN
WRITE(*,*) 'x is positive and x >= y’
ELSE
WRITE(*,*) 'x is positive but x < y’
END IF
END IF
```

More Visibility

```
IF (x .GT. 0) THEN
    IF (x .GE. y) THEN
        WRITE(*,*) 'x is positive and x >= y’
    ELSE
        WRITE(*,*) 'x is positive but x < y’
    END IF
END IF
```

Today's Tools Rock!



```
    color: #215968;
}
.auto-style19 {
    color: #000000;
}
-->
</style>

<meta content=document name=resource-type>
<meta content="software testing, software quality" name=keywords>
</head>

<body lang=EN-US link=blue vlink=purple>

<div class=WordSection1>

<table class=MsoNormalTable border=0 cellpadding=0 width="95%"
 style='width:95.0%'>
 <tr>
  <td width=150 rowspan=2 valign=top style='width:112.5pt;background:#DAEEF3;
  padding:.75pt .75pt .75pt .75pt'>
  <div align=center>
  <table class=MsoNormalTable border=0 cellpadding=0 width=135
   style='width:101.25pt'>
   <tr>
    <td width=131 style='width:98.25pt;padding:1.5pt 1.5pt 1.5pt 1.5pt'></td>
   </tr>
   <tr>
```

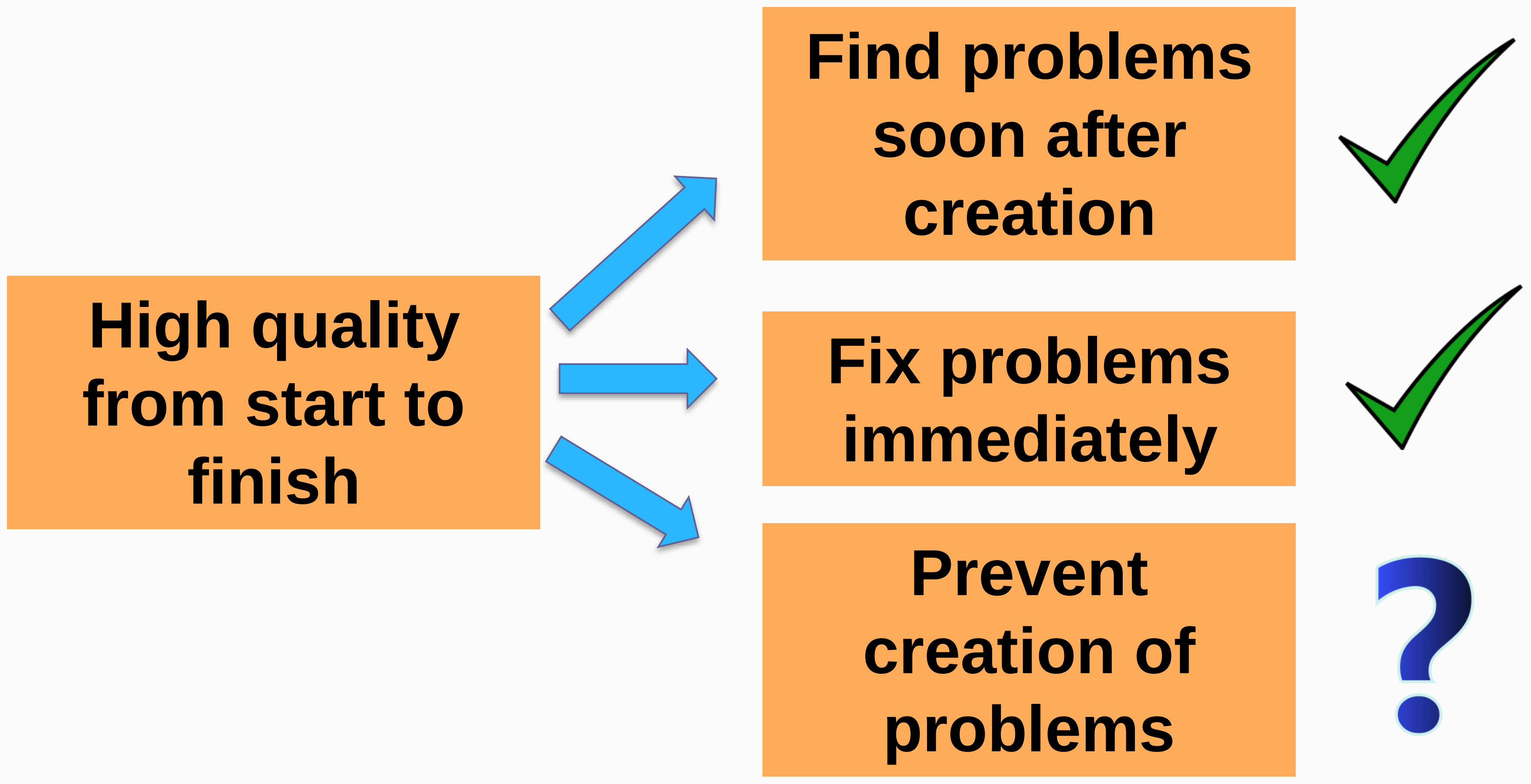
Maintain Intellectual Control

No more than seven things at once!

Abstractions which simplify:

- Classes and objects
- Design notation – UML, DFDs
- Design patterns

Test Driven Development


High quality from start to finish
Find problems soon after creation
Fix problems immediately
Prevent creation of problems
?

Know Your Domain

Coworkers

Classes

Books

Webinars

Conferences

Design Away Opportunity for Error

User interface field must be long enough for longest value in corresponding database field.

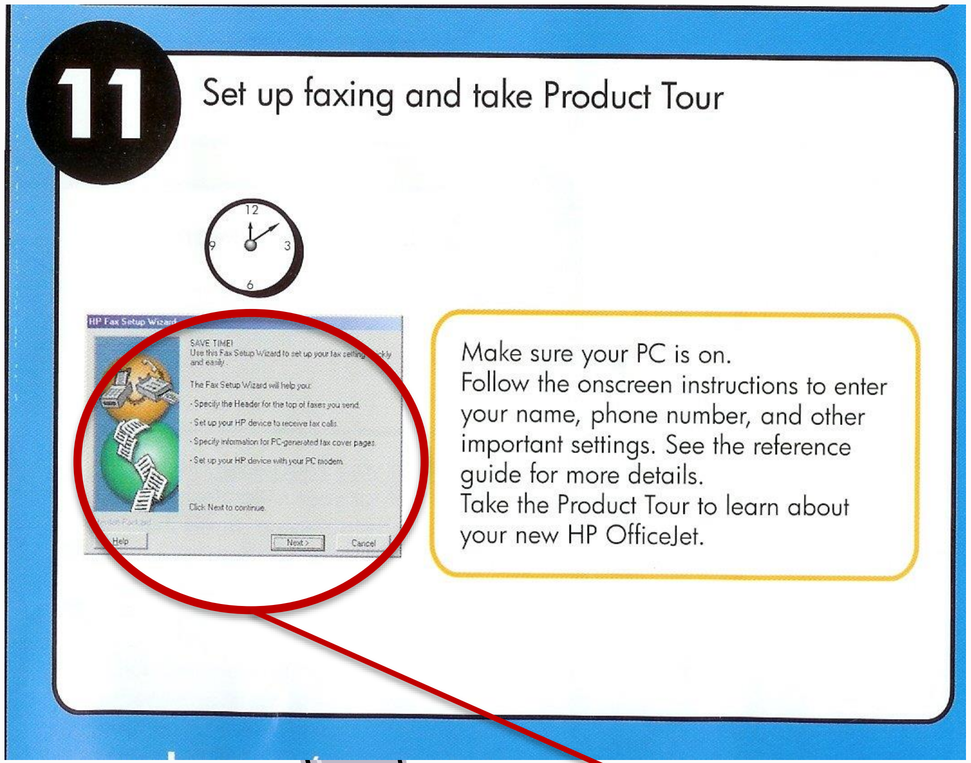
A) Code UI field length to match database field length and write test for longest possible value.

B) Test tool runs through UI code, checking UI field lengths against database field lengths.

C) User interface reads field length from database and creates UI field to match.

Credit: James Shore, *Art of Agile Development*, “No Bugs”

Design Away Opportunity for Error



11 Set up faxing and take Product Tour

HP Fax Setup Wizard

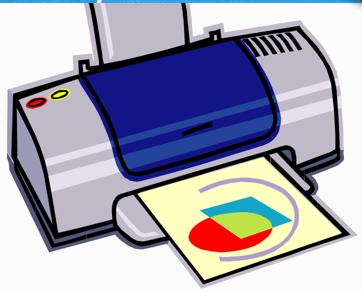
SAVE TIME!
Use this Fax Setup Wizard to set up your fax setting quickly and easily.

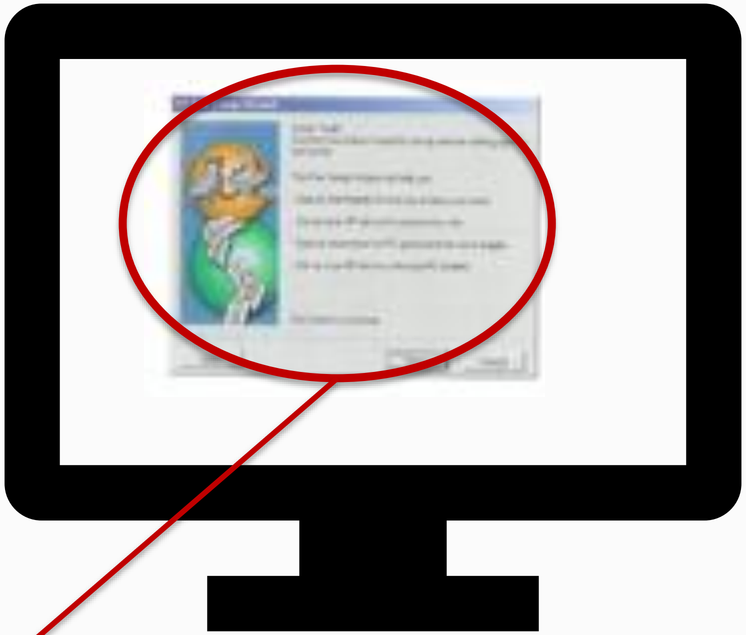
The Fax Setup Wizard will help you:

- Specify the Header for the top of faxes you send.
- Set up your HP device to receive fax calls.
- Specify information for PC-generated fax cover pages.
- Set up your HP device with your PC modem.

Click Next to continue.

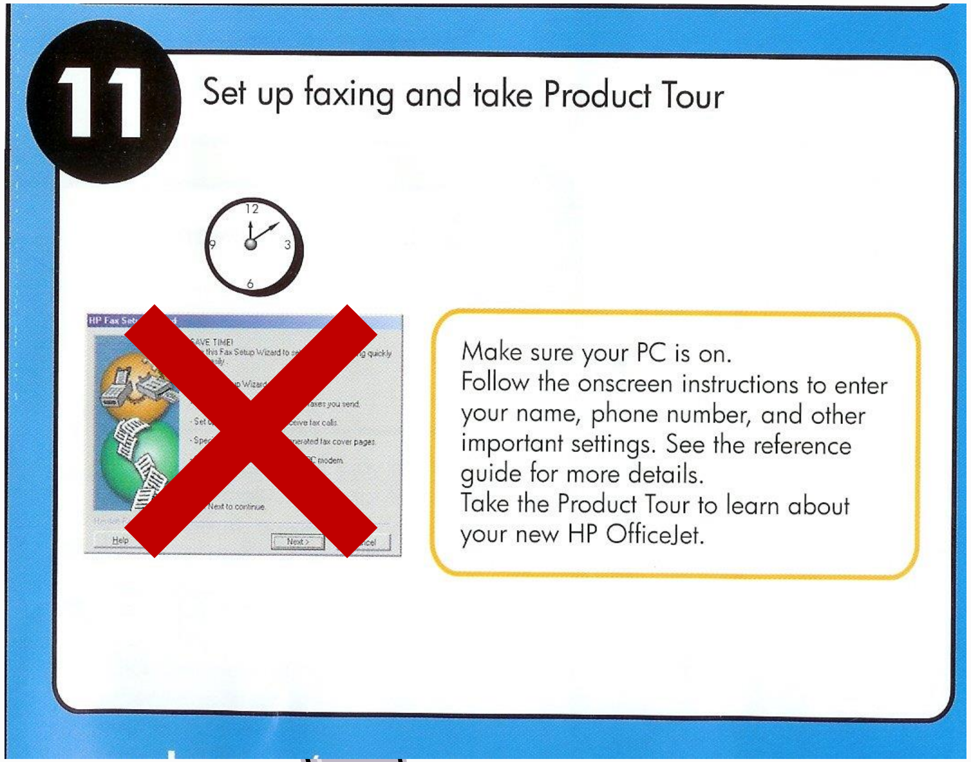
Make sure your PC is on.
Follow the onscreen instructions to enter your name, phone number, and other important settings. See the reference guide for more details.
Take the Product Tour to learn about your new HP OfficeJet.





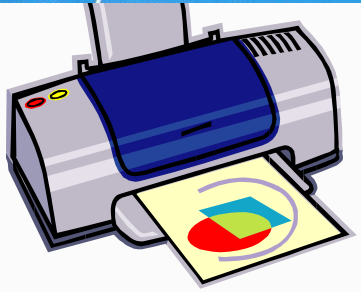
Defect: They don't match

Design Away Opportunity for Error

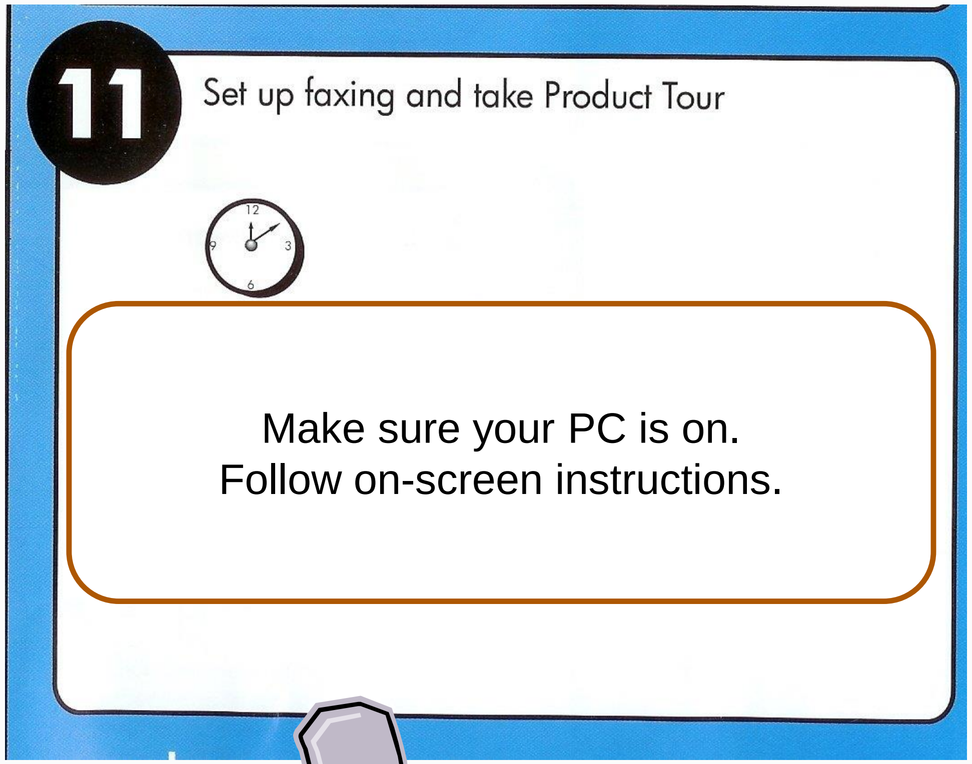


11 Set up faxing and take Product Tour

Make sure your PC is on.
Follow the onscreen instructions to enter your name, phone number, and other important settings. See the reference guide for more details.
Take the Product Tour to learn about your new HP OfficeJet.

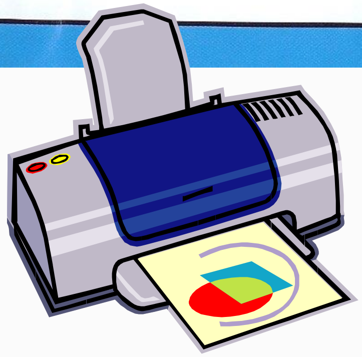


Design Away Opportunity for Error

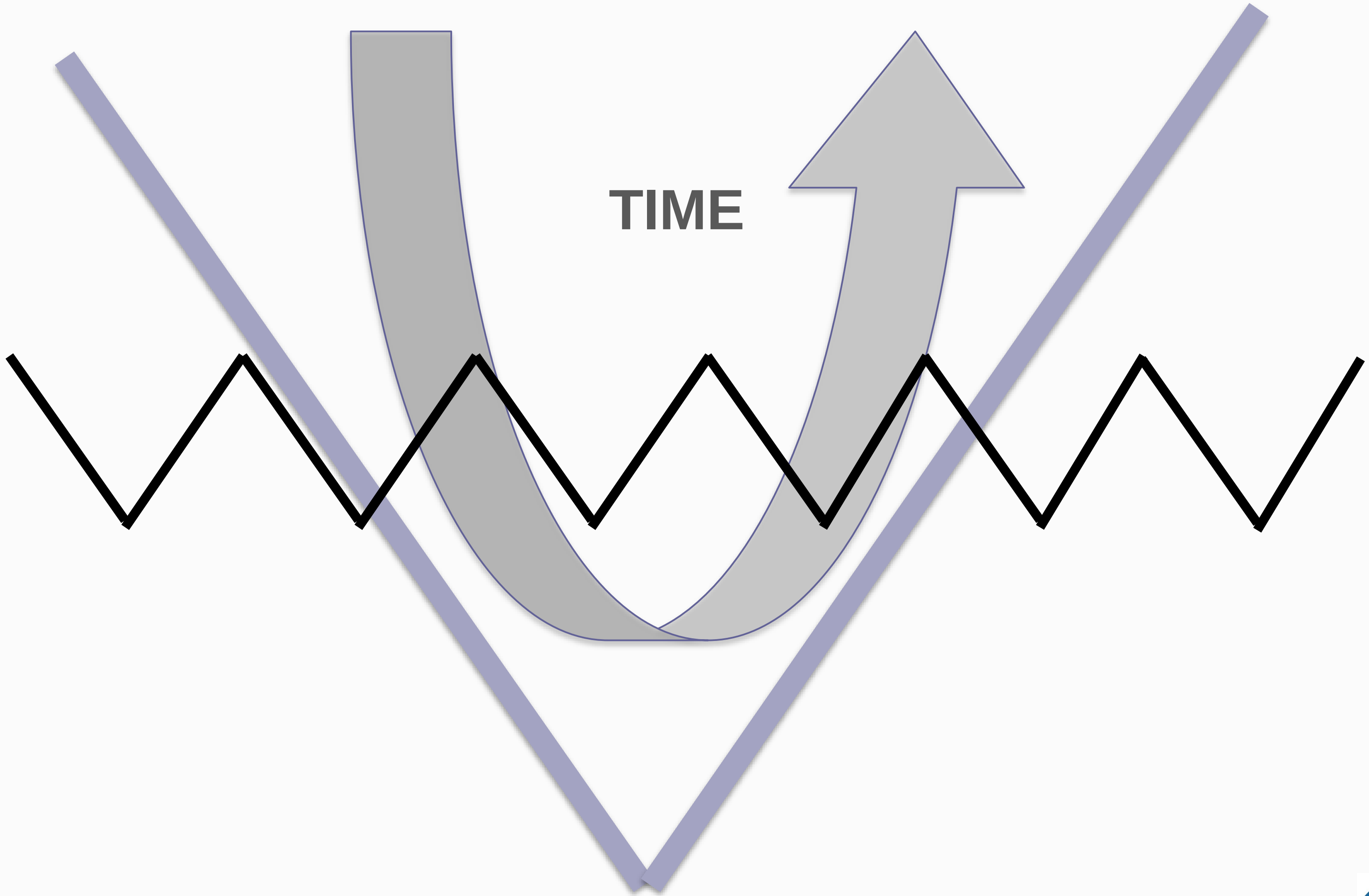


11 Set up faxing and take Product Tour

Make sure your PC is on.
Follow on-screen instructions.

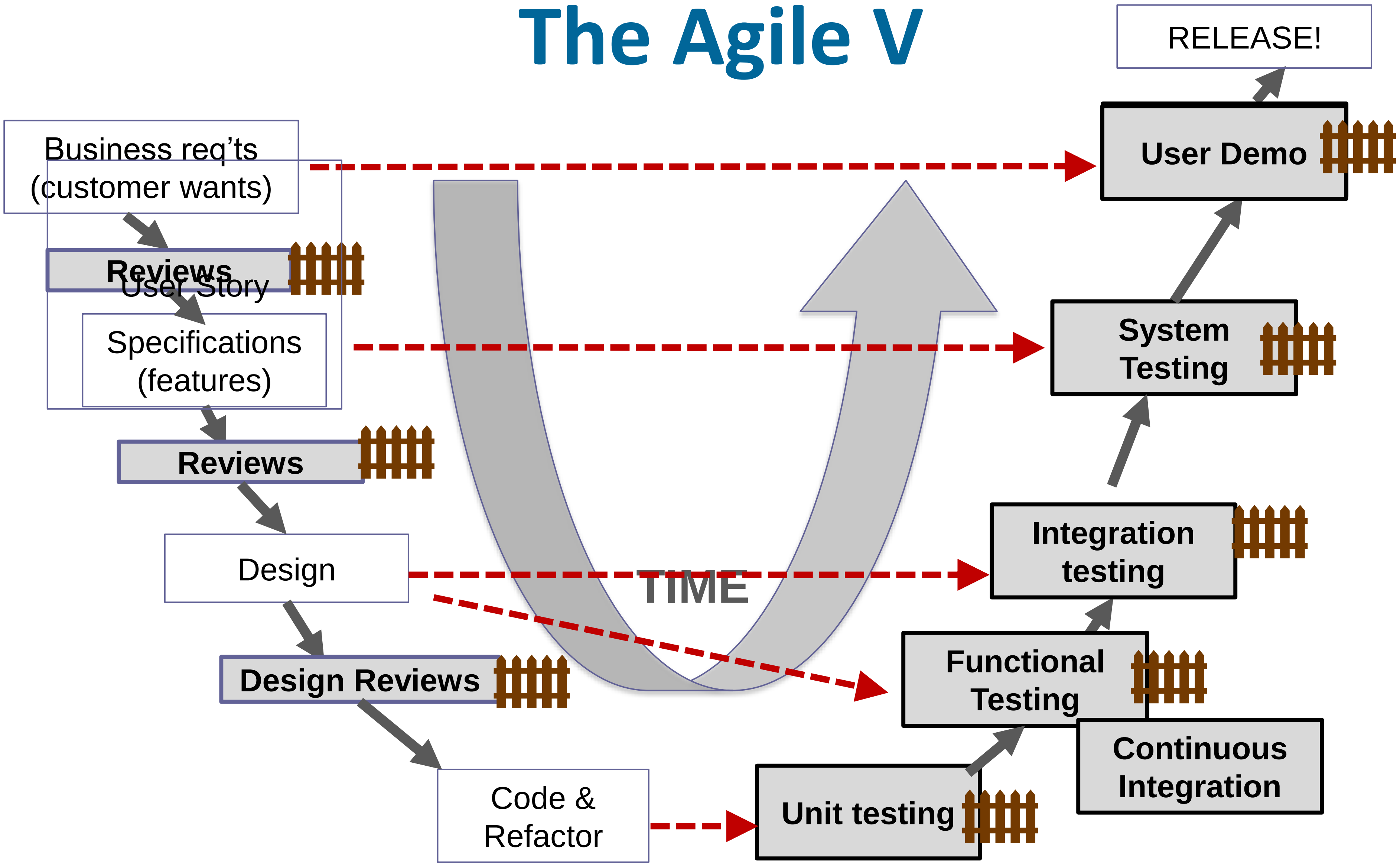


The Agile V

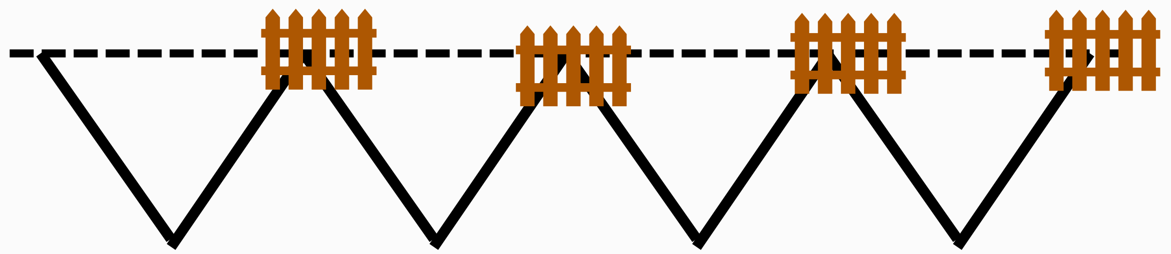


IBERLE
CONSULTING GROUP, INC.

The Agile V

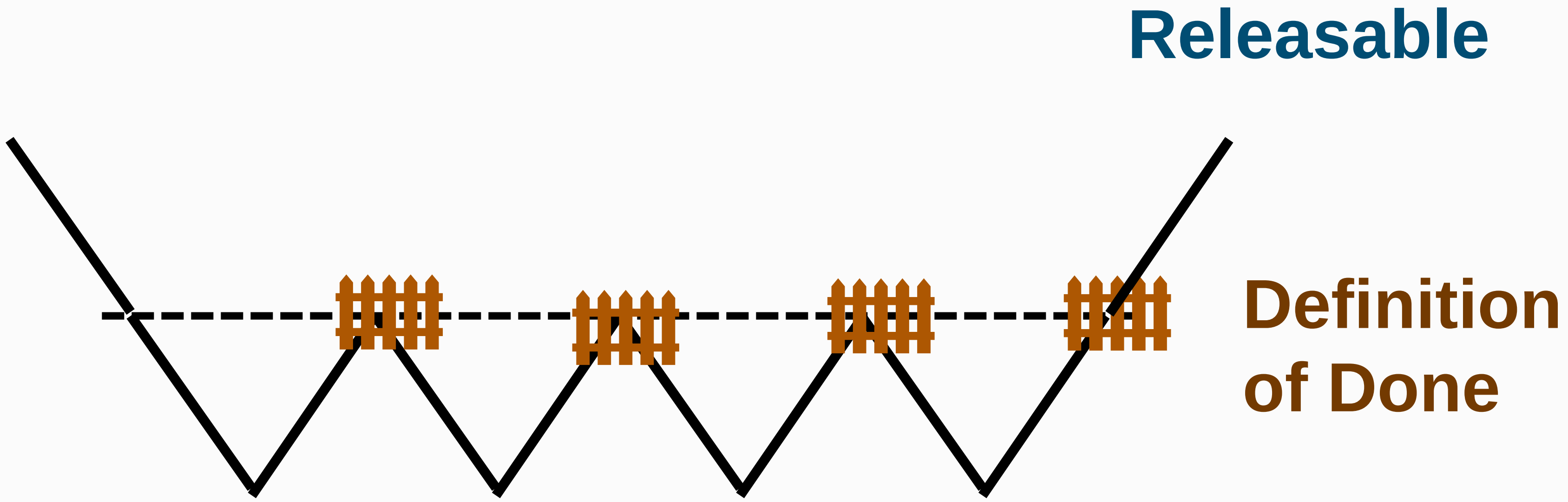

RELEASE!
Business req'ts (customer wants)
User Demo
Reviews
User Story
Specifications (features)
System Testing
Reviews
Integration testing
Design
TIME
Design Reviews
Functional Testing
Continuous Integration
Code & Refactor
Unit testing

Agile Quality Gates



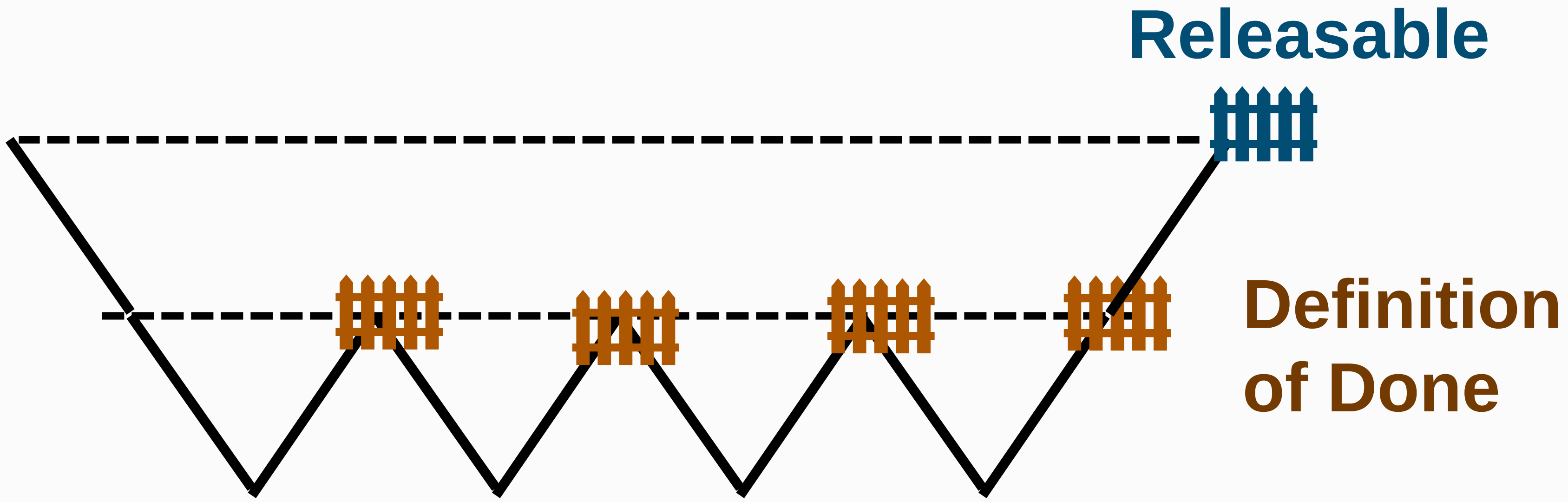
Definition of Done

Agile Quality Gates

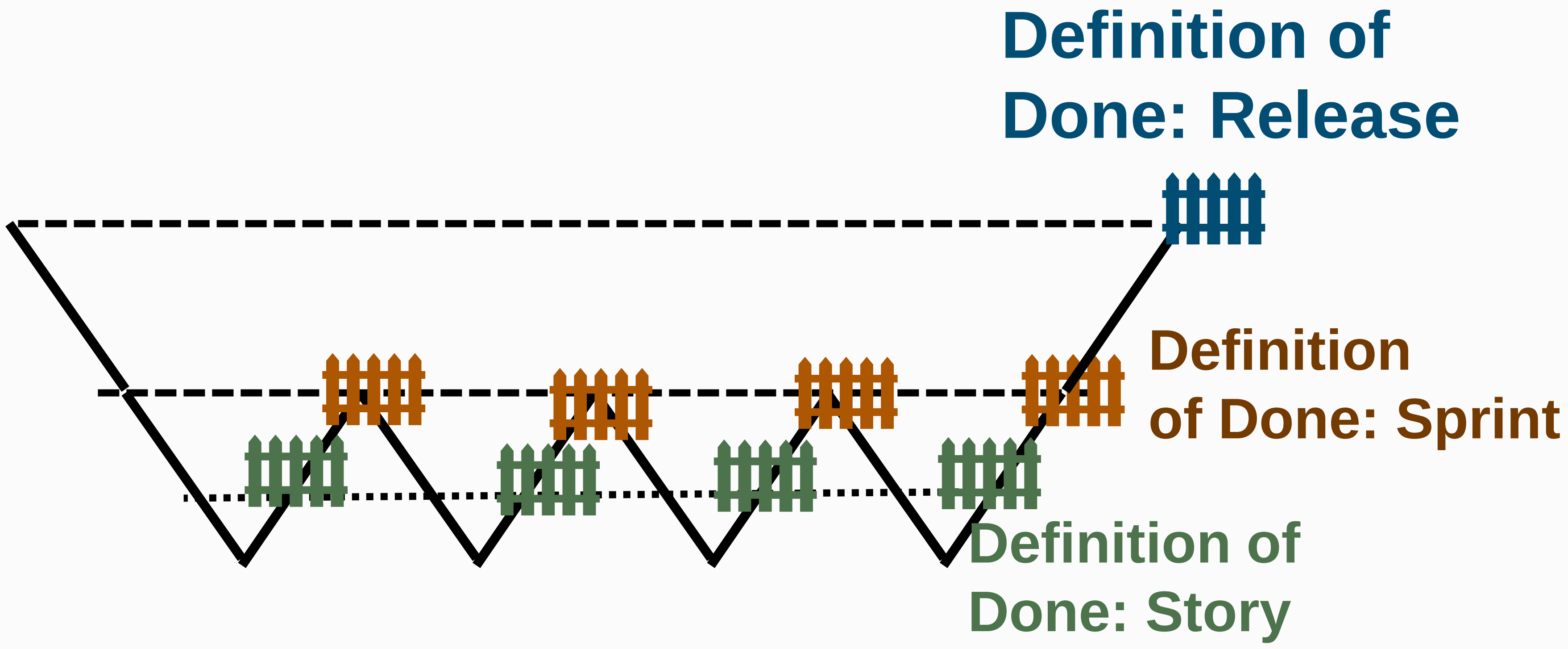


IBERLE
CONSULTING GROUP, INC.

Agile Quality Gates



Agile Quality Gates


Definition of Done: Release
Definition of Done: Sprint
Definition of Done: Story

How Agile Affects Software Quality

The Good

- Fast feedback
- Focus on user and customer
- TDD
- ATDD
- Design patterns
- Pair programming
- Continuous integration

The Problematic

- Demolishing too many quality gates
- Ignoring non-functional requirements
- Expecting user demos to find design problems

Building In Quality – Then and Now

- Quality is <u>built in </u>during development
  - Find problems early and fix right away
  - An ounce of prevention is worth a pound of cure
- Some prevention strategies
  - Make the structure & logic visible
  - Mistake-proof with tools
  - Maintain intellectual control
  - Know your domain
  - Design away the opportunity for error
- There's more in my paper: see PNSQC proceedings or
  www.kiberle.com

Q & A