

Carnegie Mellon University

This video and all related information and materials (“materials”) are owned by Carnegie Mellon University. These materials are provided on an “as-is” “as available” basis without any warranties and solely for your personal viewing and use.

You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced websites, and/or for any consequences or the use by you of such materials.

By viewing, downloading, and/or using this video and related materials, you agree that you have read and agree to our terms of use (www.sei.cmu.edu/legal/).

© 2015 Carnegie Mellon University.

Copyright 2015 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

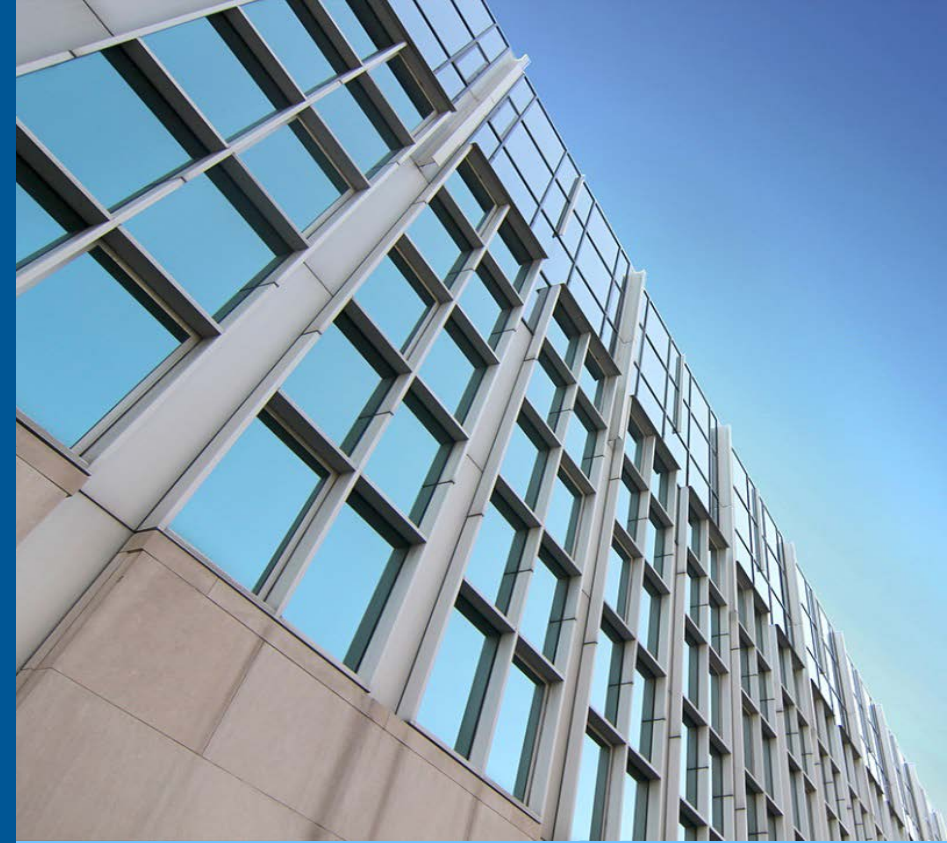
Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0002727

A Taxonomy of Testing Types

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Donald G. Firesmith



Topics

Relevant Testing Challenges

Goals of Presentation

What is Testing?

Presentation Scope

Testing Types

Conclusion

Relevant Testing Challenges

Relevant Testing Challenges

Many testers are only aware of a minority of types of testing, let alone know how to perform them.

Test managers and developers are aware of even fewer testing types.

The test strategies, project test plans, and test sections of system/software development plans tend to identify only a very small number of types of testing (e.g., unit, integration, system, and acceptance testing).

- Not planned → Not performed

Goals of the Presentation

Goals of the Presentation

Make it clear that:

- There are *many* different types of testing.
- Testing is a complex discipline with its own technical jargon.
- There is a lot of overlap between different classes of testing types.
 - Think multiple classification (object-oriented design) or multiple inheritance (object-oriented programming).

Get you to take a look at your:

- Test strategies and test plans and ask yourselves “Are they sufficiently complete?”
- Testers and other testing stakeholders and ask yourselves “Do they need additional training in testing types.”

Polling Question 1

How many different types of testing do you typically use on a project?

- 1-5
- 6-10
- 11-15
- 16+

What is Testing?

What is Testing?

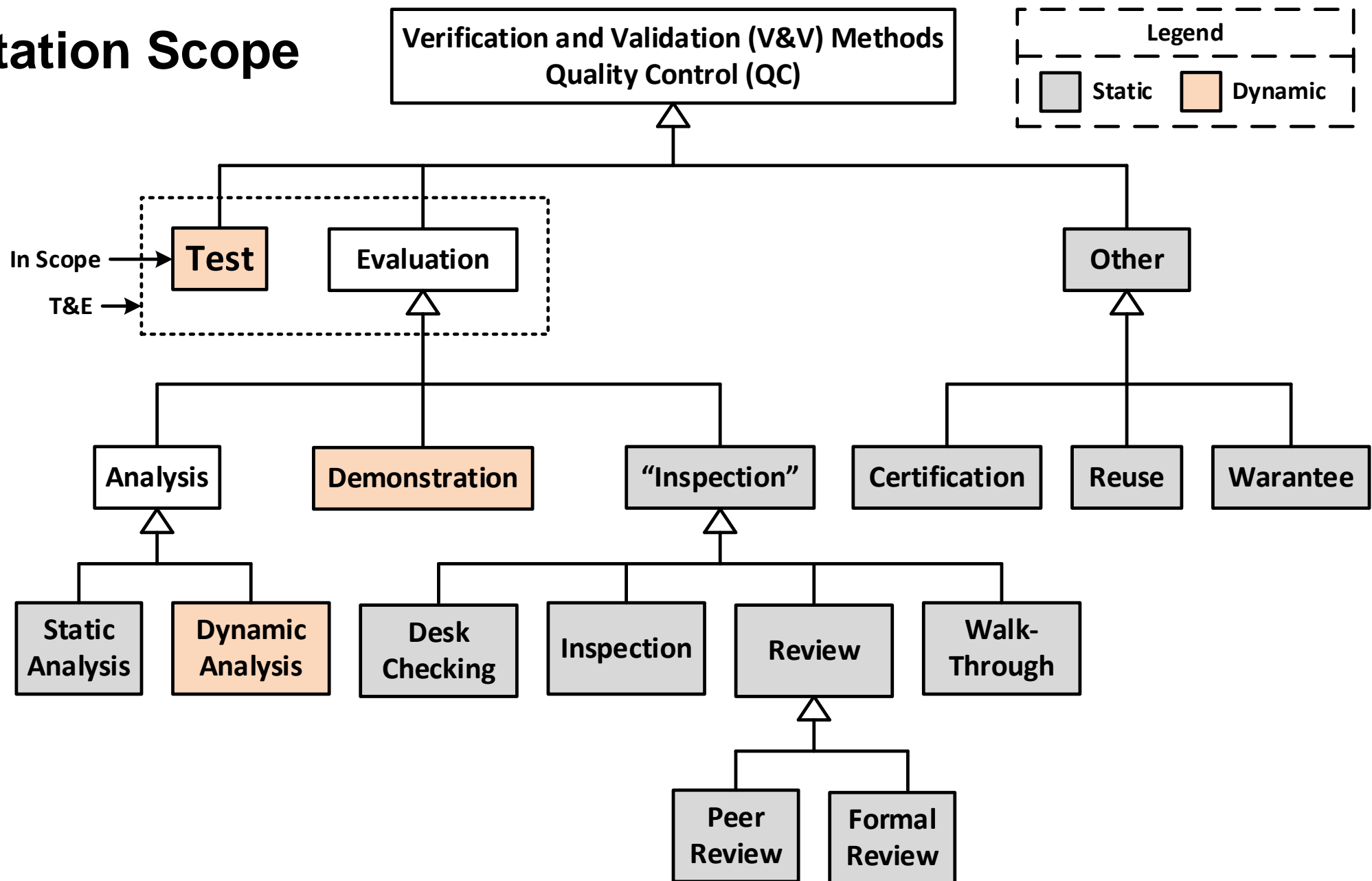
Testing

The **execution** of an Object Under Test (OUT) under specific **preconditions** with specific **stimuli** so that its **actual behavior** can be compared with its **expected or required behavior**

- *Preconditions*: pretest mode, states, stored data, or external conditions
- *Stimuli*:
 - Calls, commands, and messages (control flows)
 - Data inputs (data flows)
 - Trigger events such as state changes and temporal events
- *Actual Behavior*:
 - *During Test*:
 - Calls, commands, and messages (control flows)
 - Data outputs (data flows)
 - *Postconditions*: post-test mode, states, stored data, or external conditions

Presentation Scope

Presentation Scope



The Taxonomy of Testing Types

Types of Testing

A type of testing is:

- A specific way to perform testing
- A class or subclass of testing
- Much narrower in scope than a testing paradigm

There are relationships between the various types of testing.

Most testers know:

- A lot about a few types of testing
- A little about some additional types of testing
- Very little about a sizable number of testing types

Polling Question 2

Have you ever seen a taxonomy of testing types (i.e., a hierarchical categorization of different ways to test)?

Yes

No

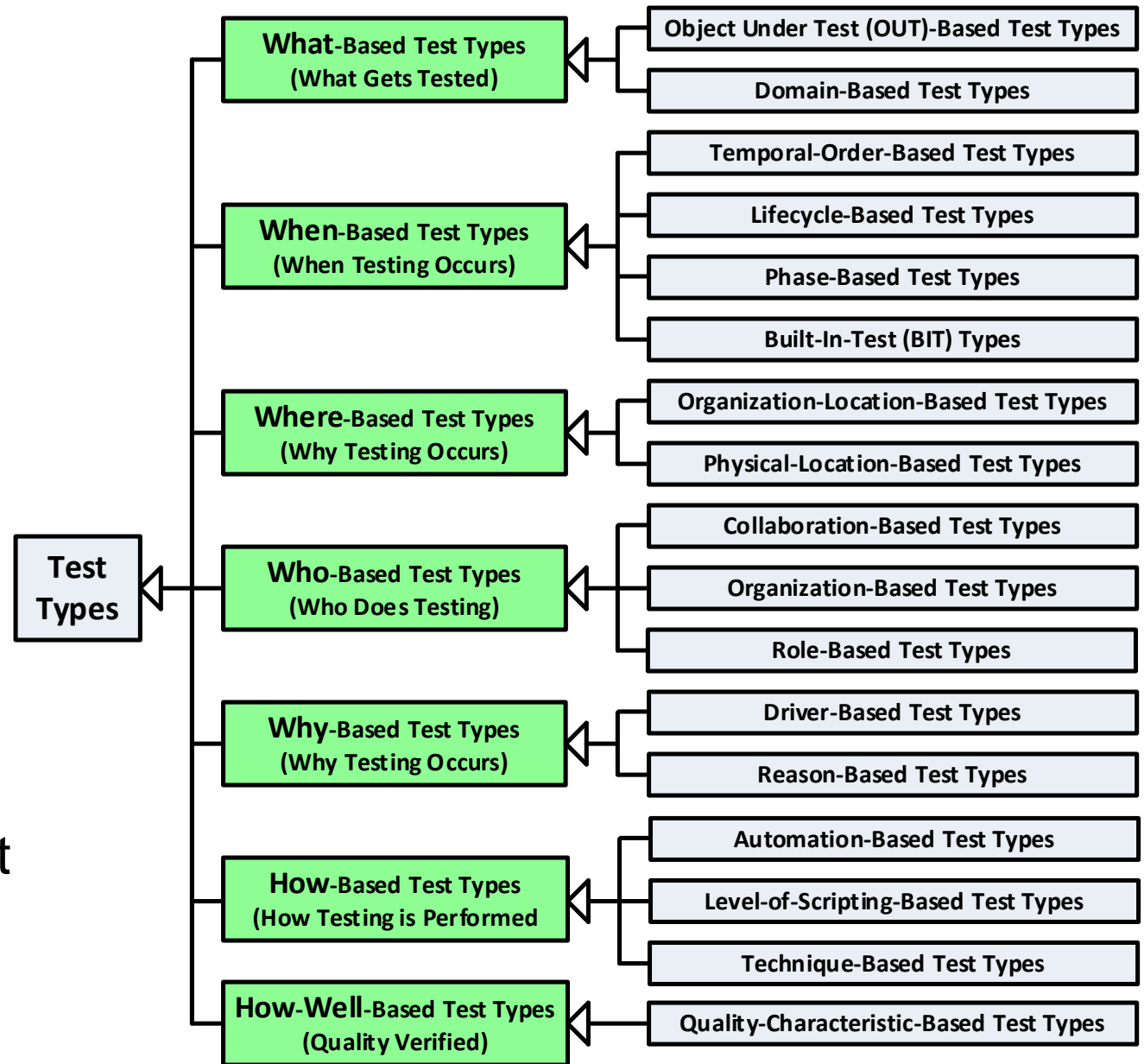
Not Sure

Types of Testing

16 Categories of Testing Types
Answering the 5W+2H Questions:

- What?
- When?
- Where?
- Who?
- Why?
- How?
- How Well?

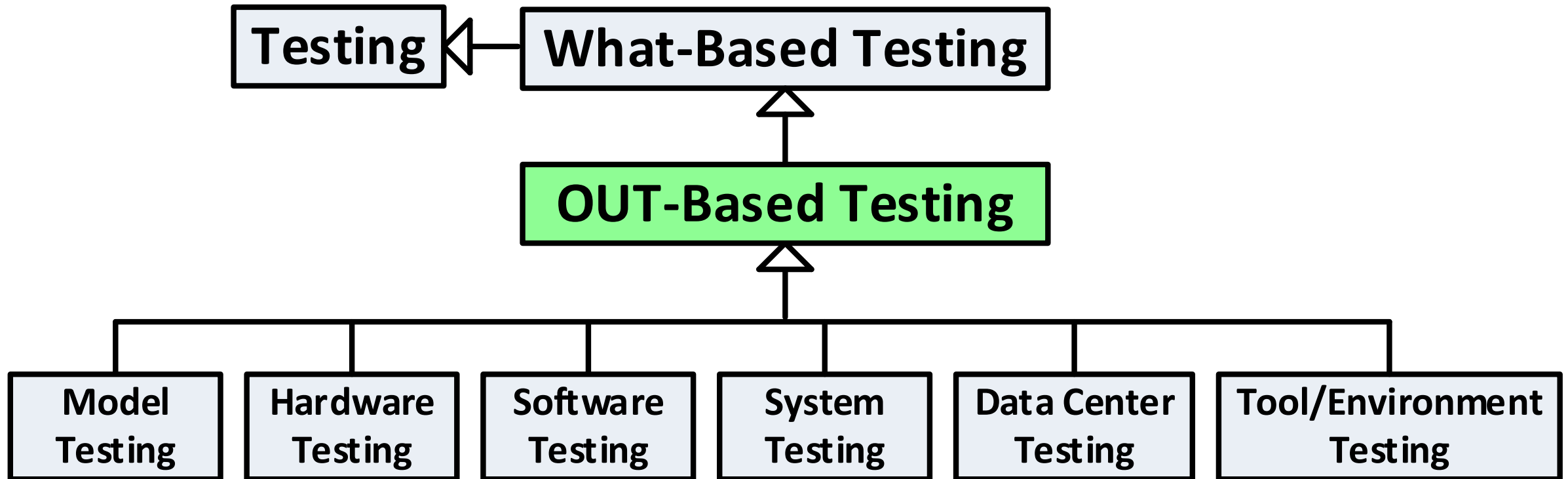
These supertypes are not disjoint
(think multiple inheritance)!



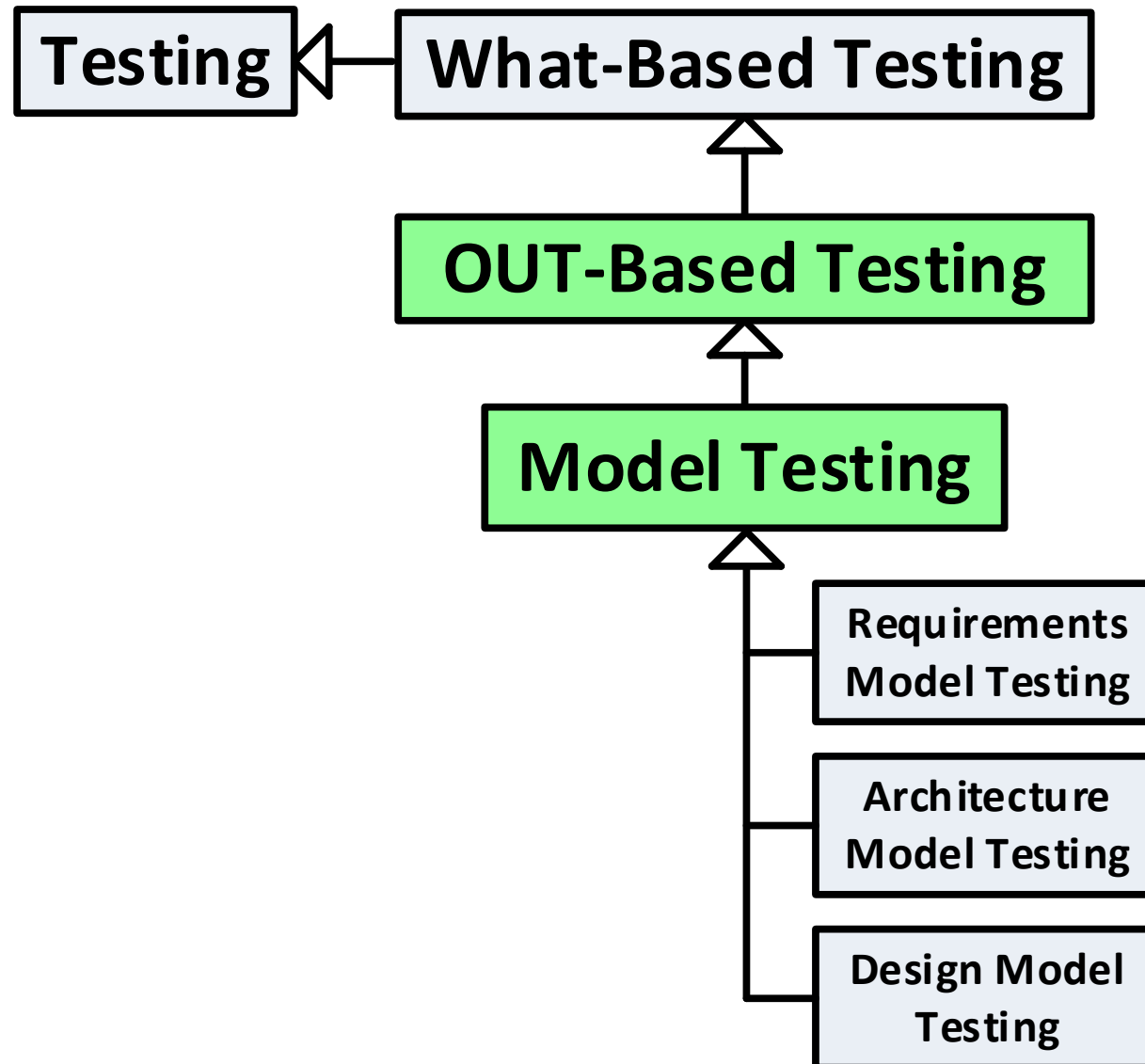
Types of Testing

WHAT is Tested

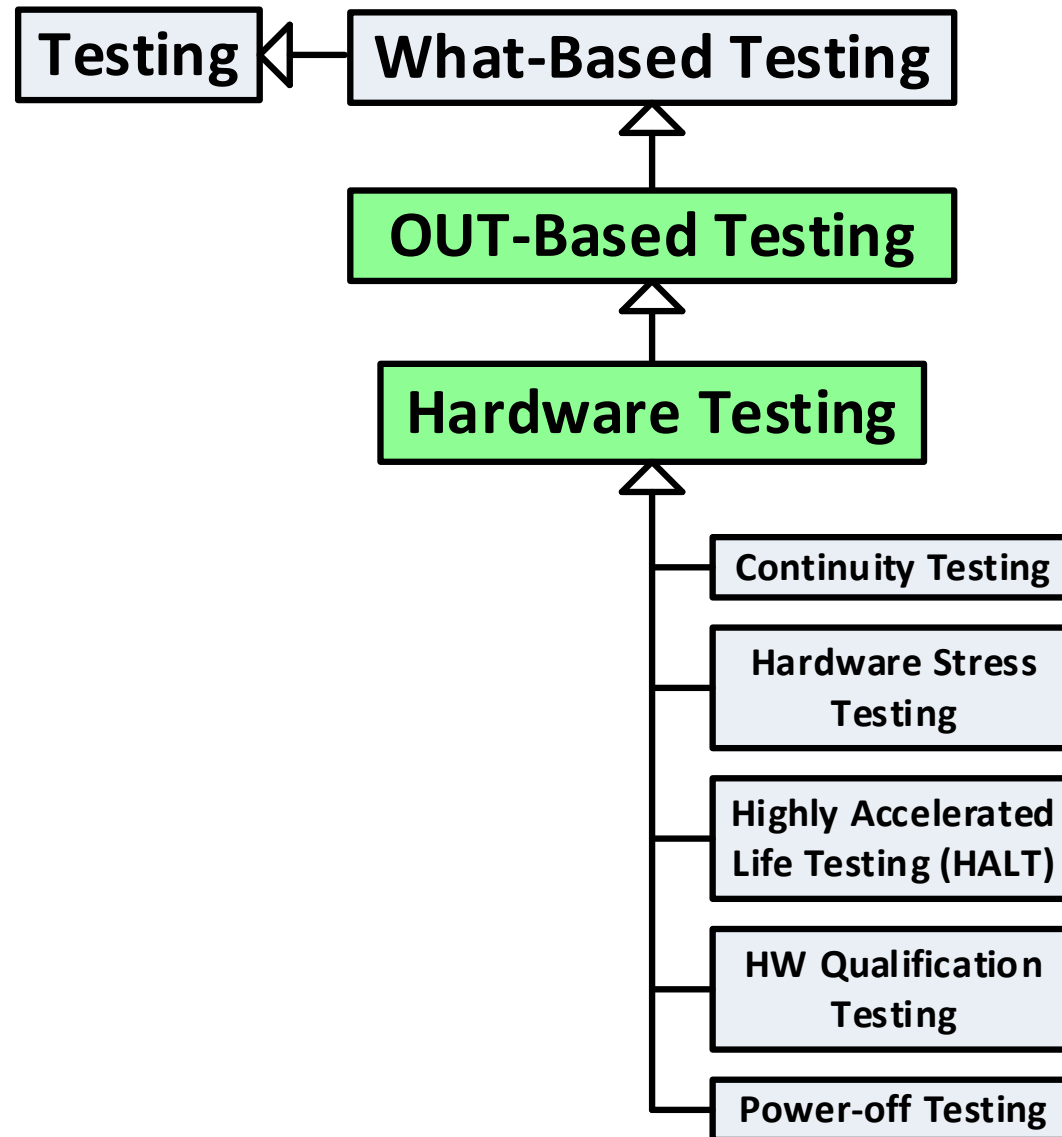
What: by Object Under Test (OUT)



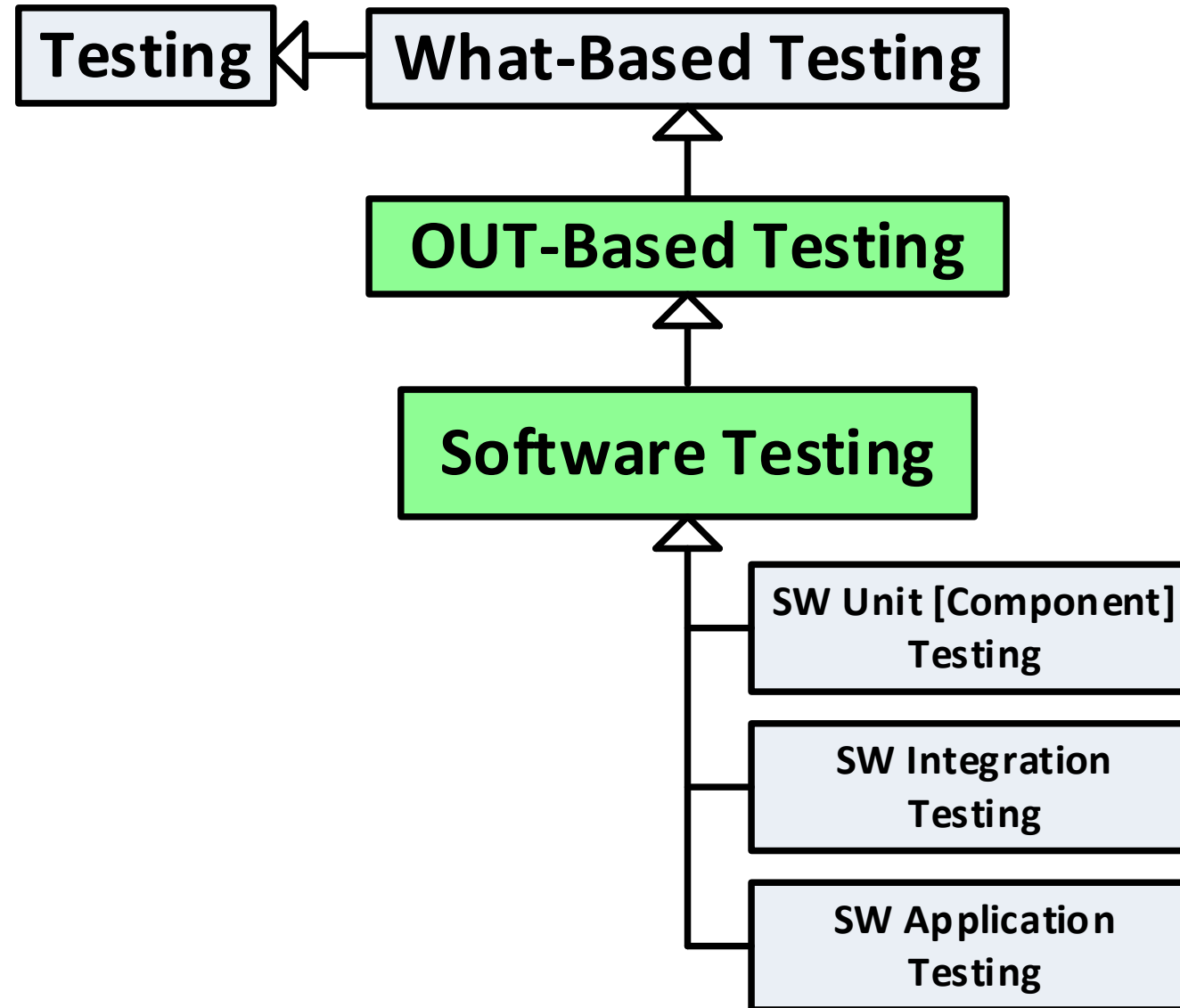
What: by Object Under Test (OUT) – Model Testing



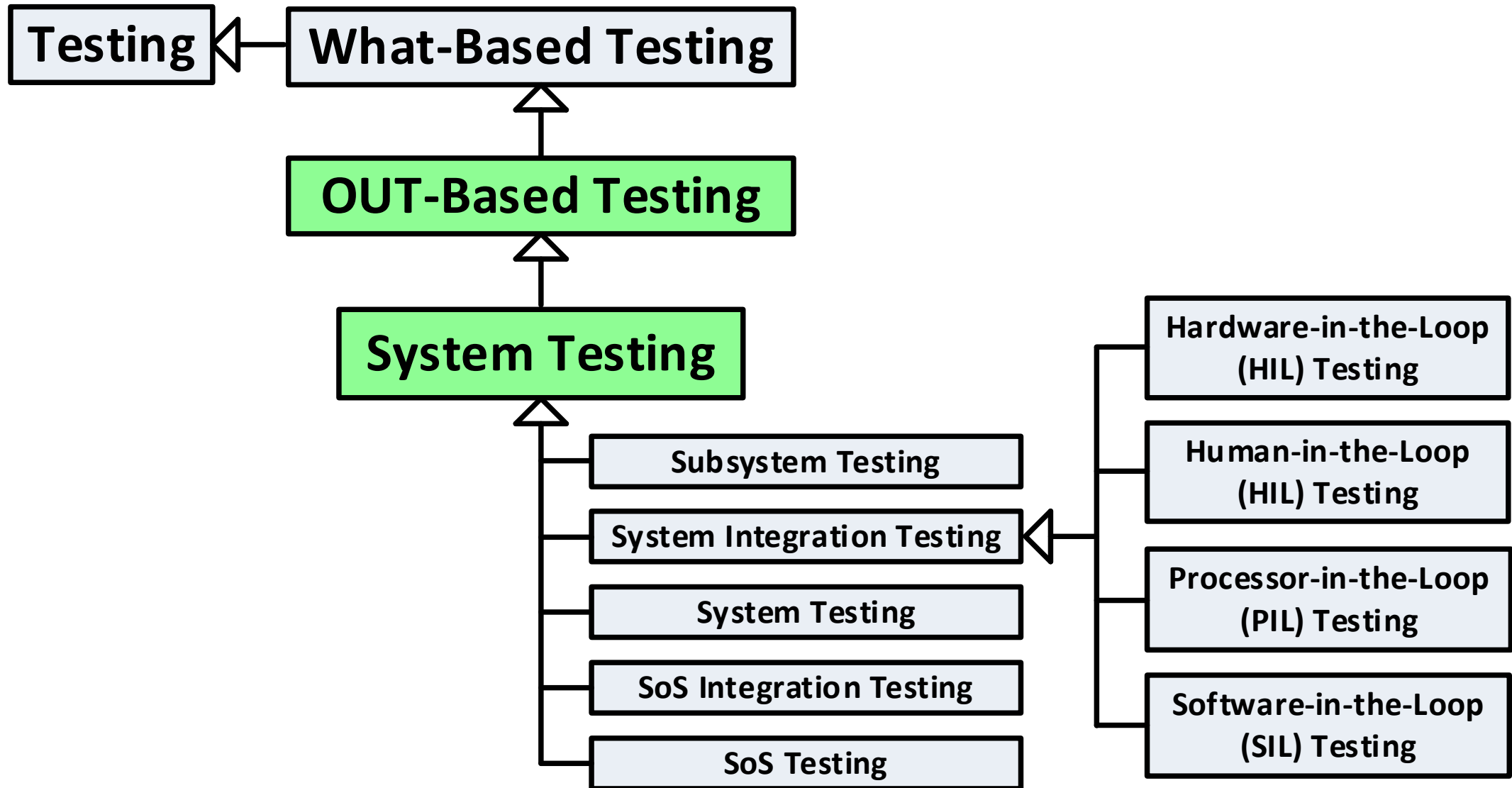
What: by Object Under Test (OUT) – Hardware Testing



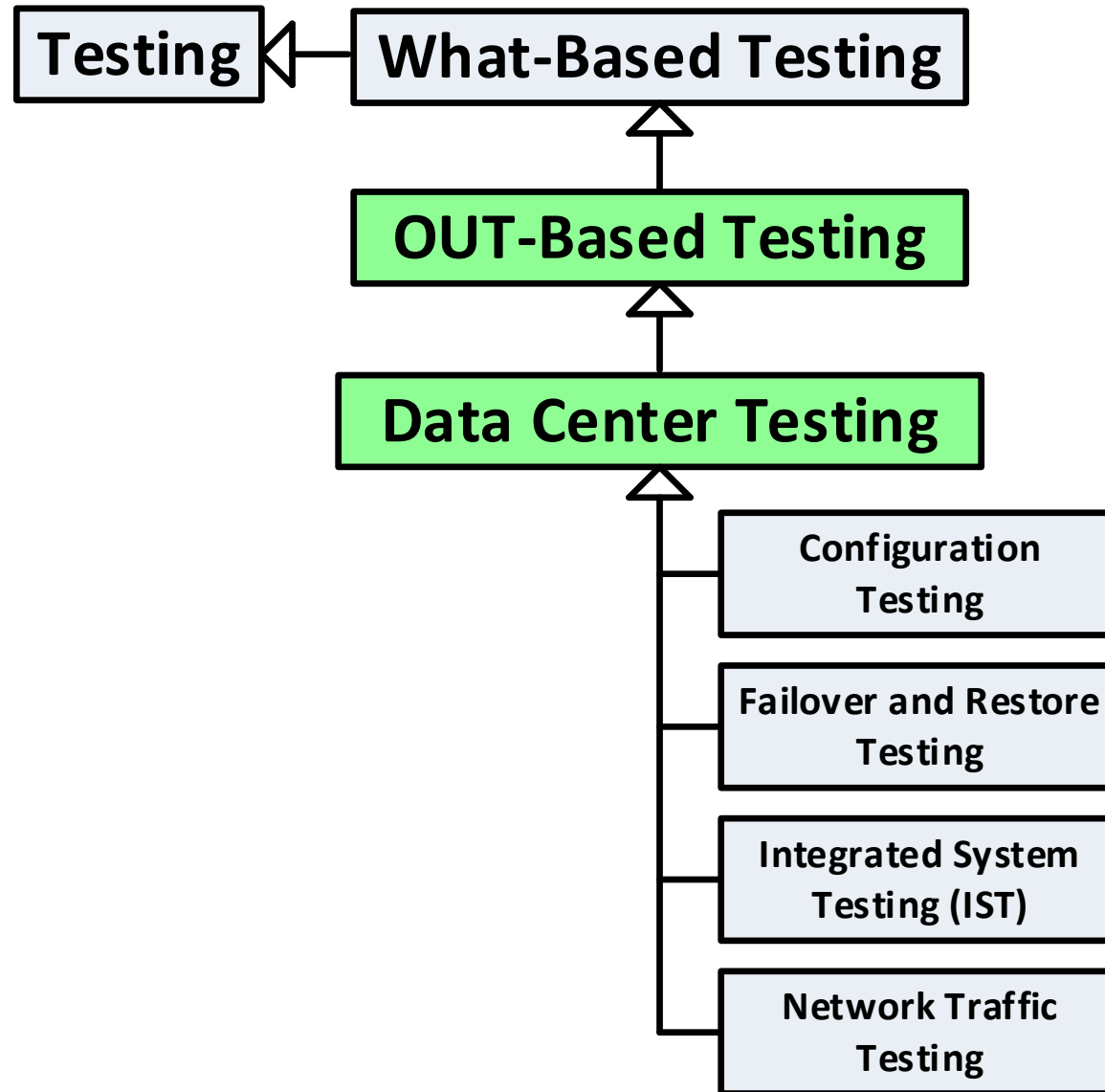
What: by Object Under Test (OUT) – Software Testing



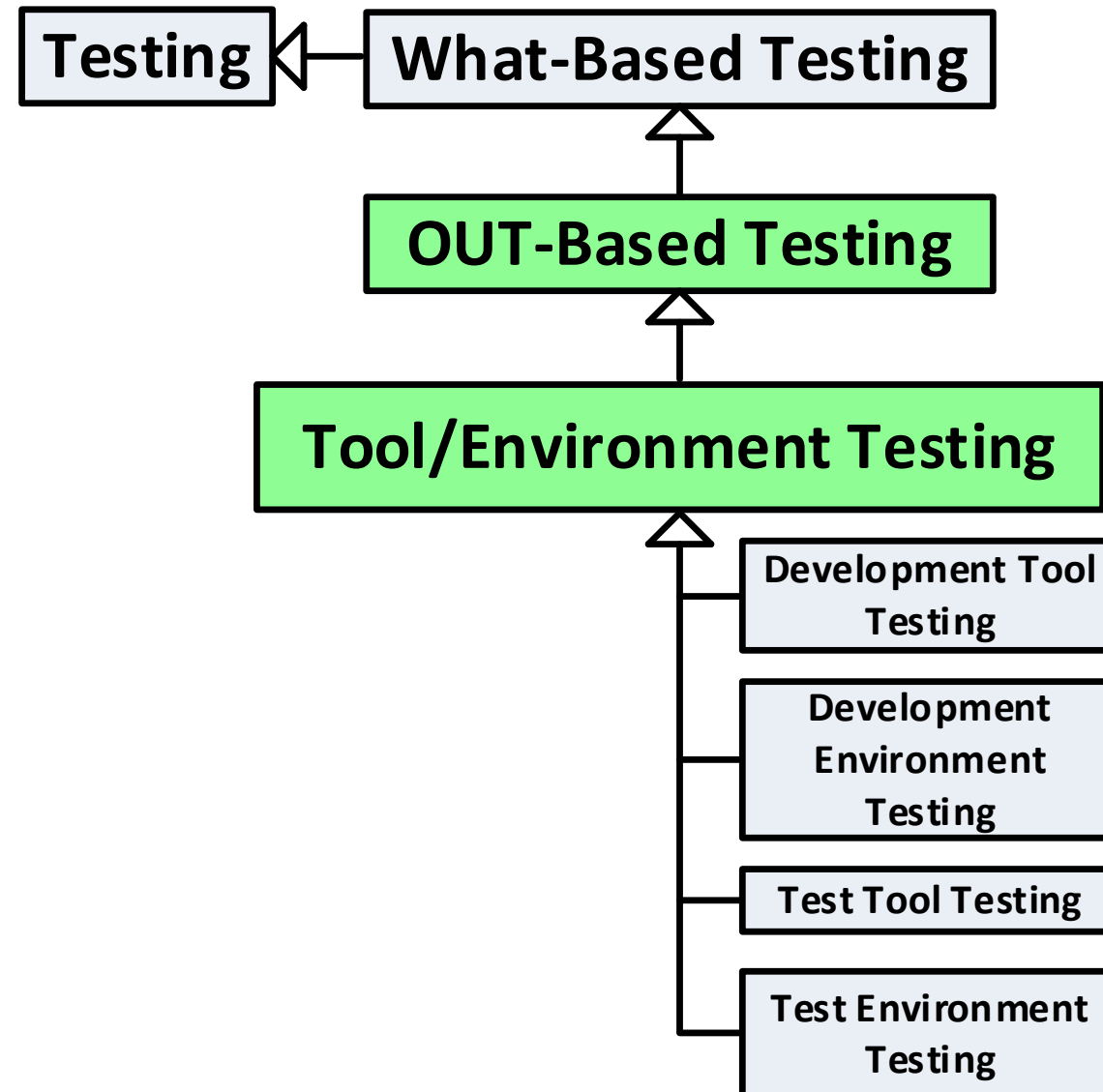
What: by Object Under Test (OUT) – System Testing



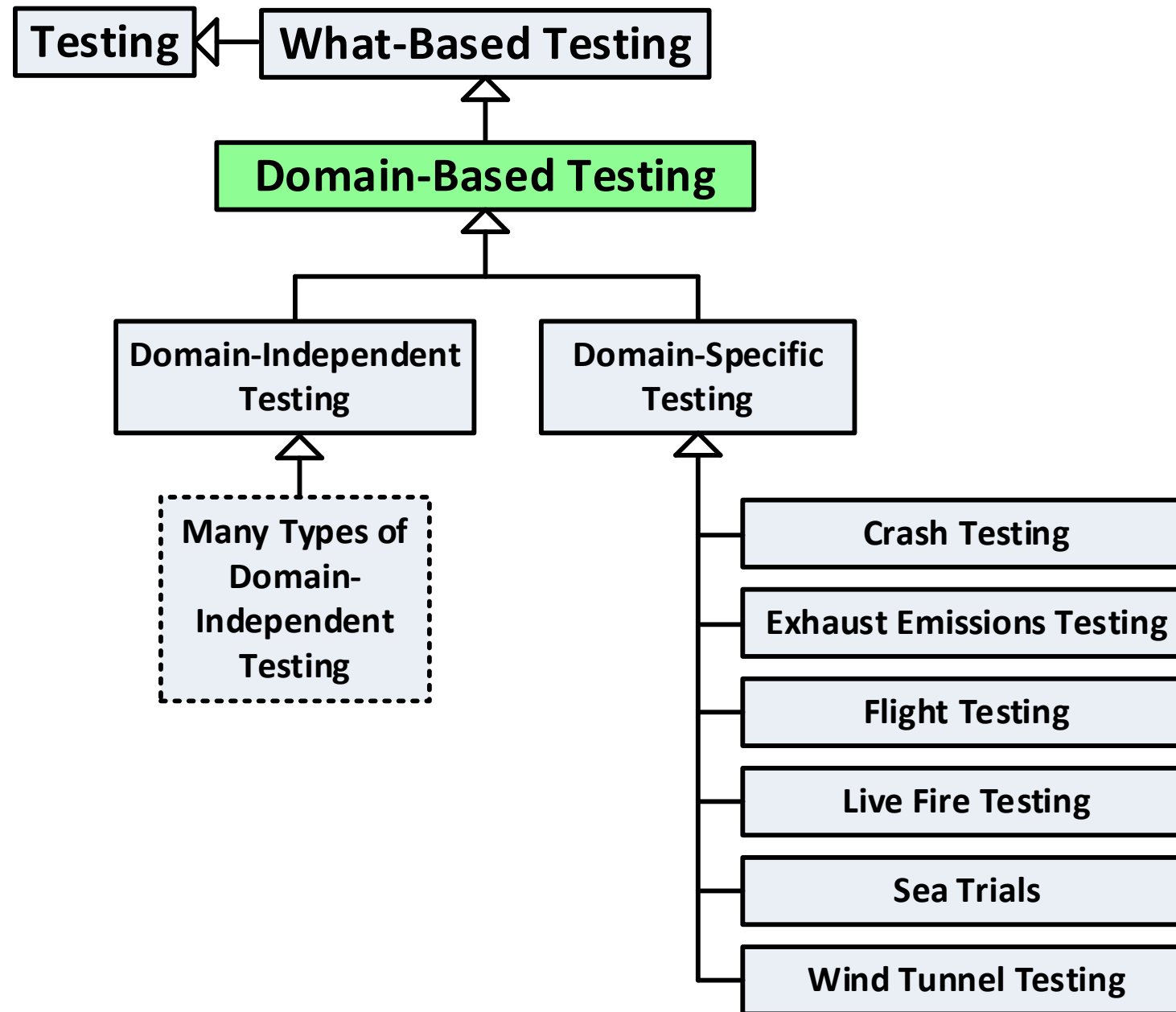
What: by Object Under Test (OUT) – Data Center Testing



What: by Object Under Test (OUT) – Tool / Environment Testing



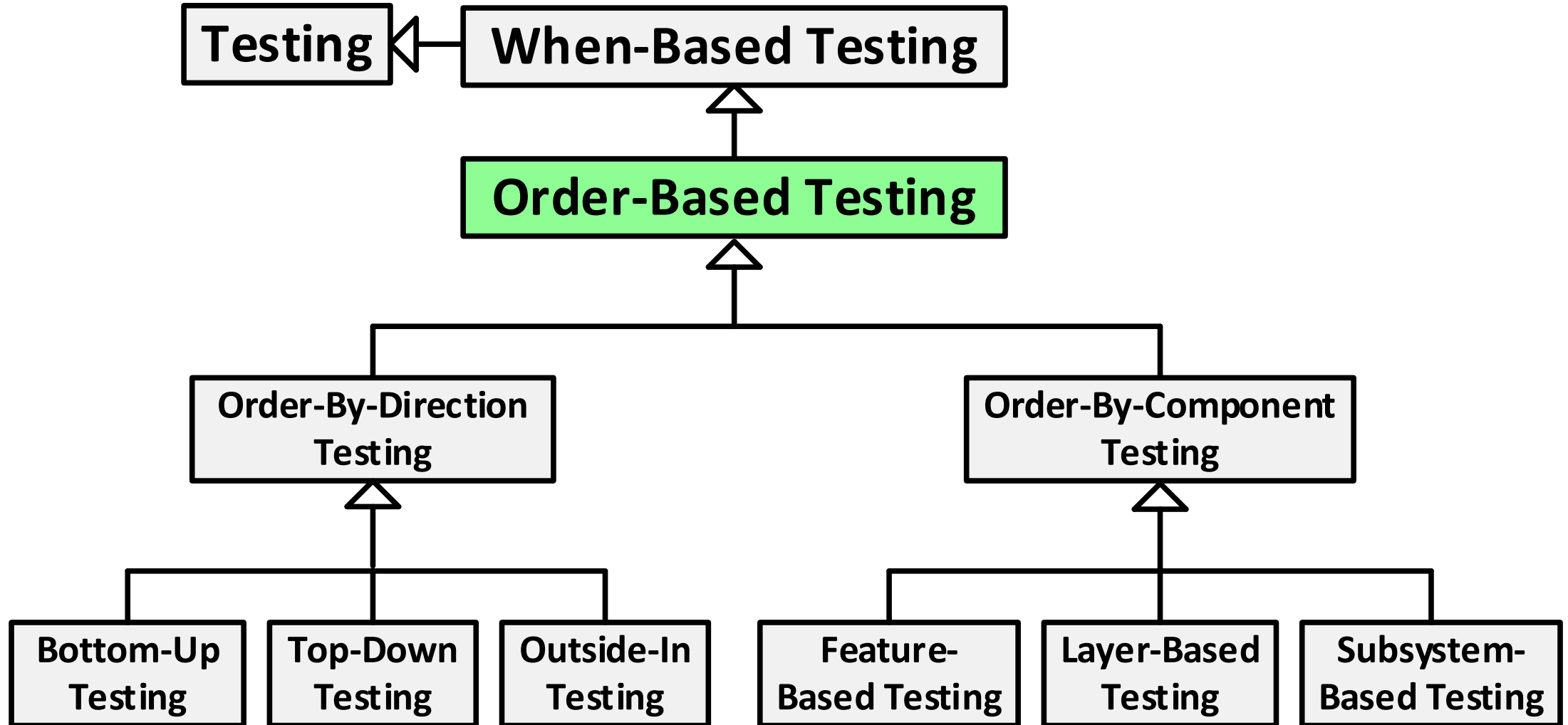
What: by Domain



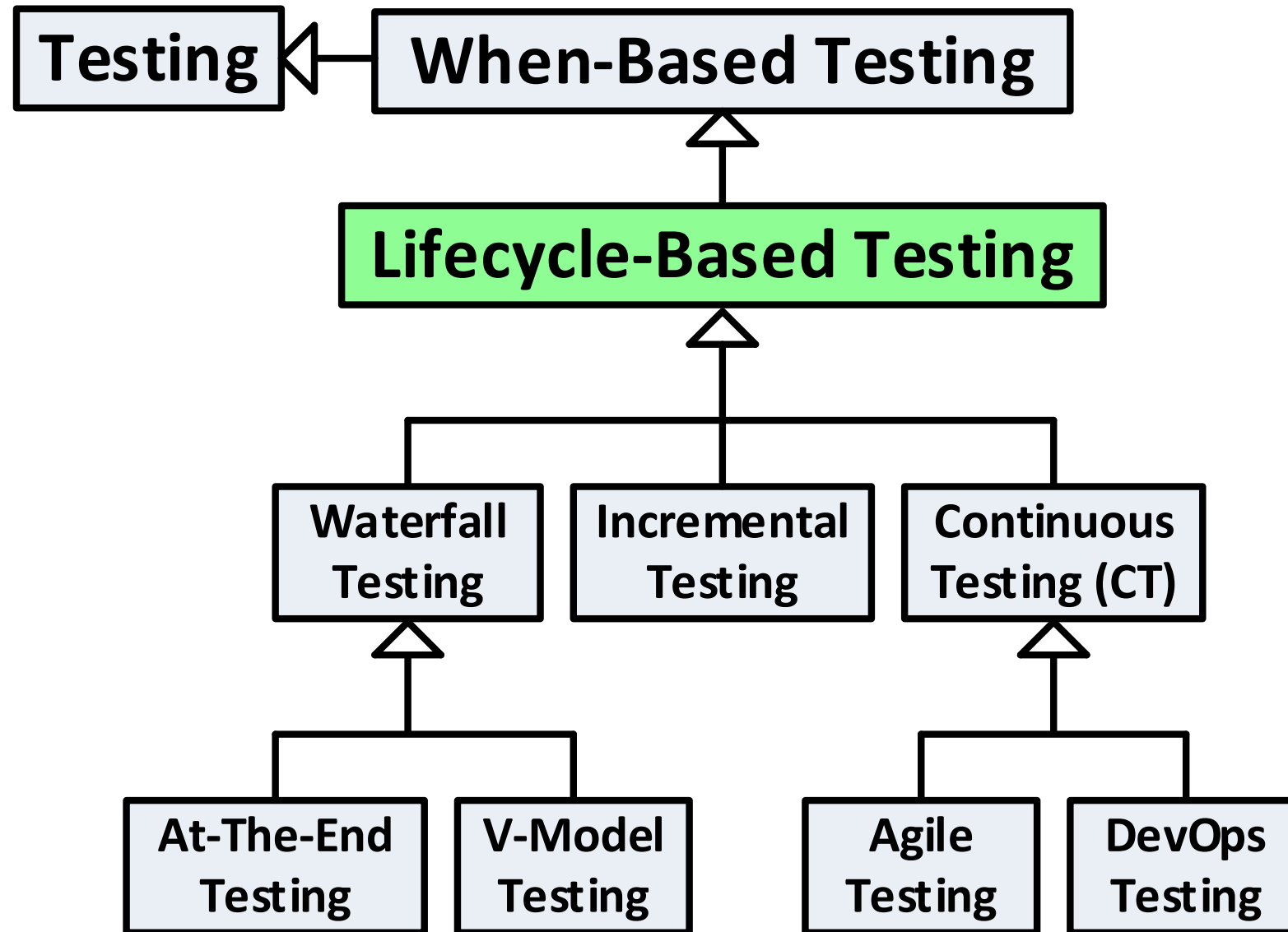
Types of Testing

WHEN Testing Occurs

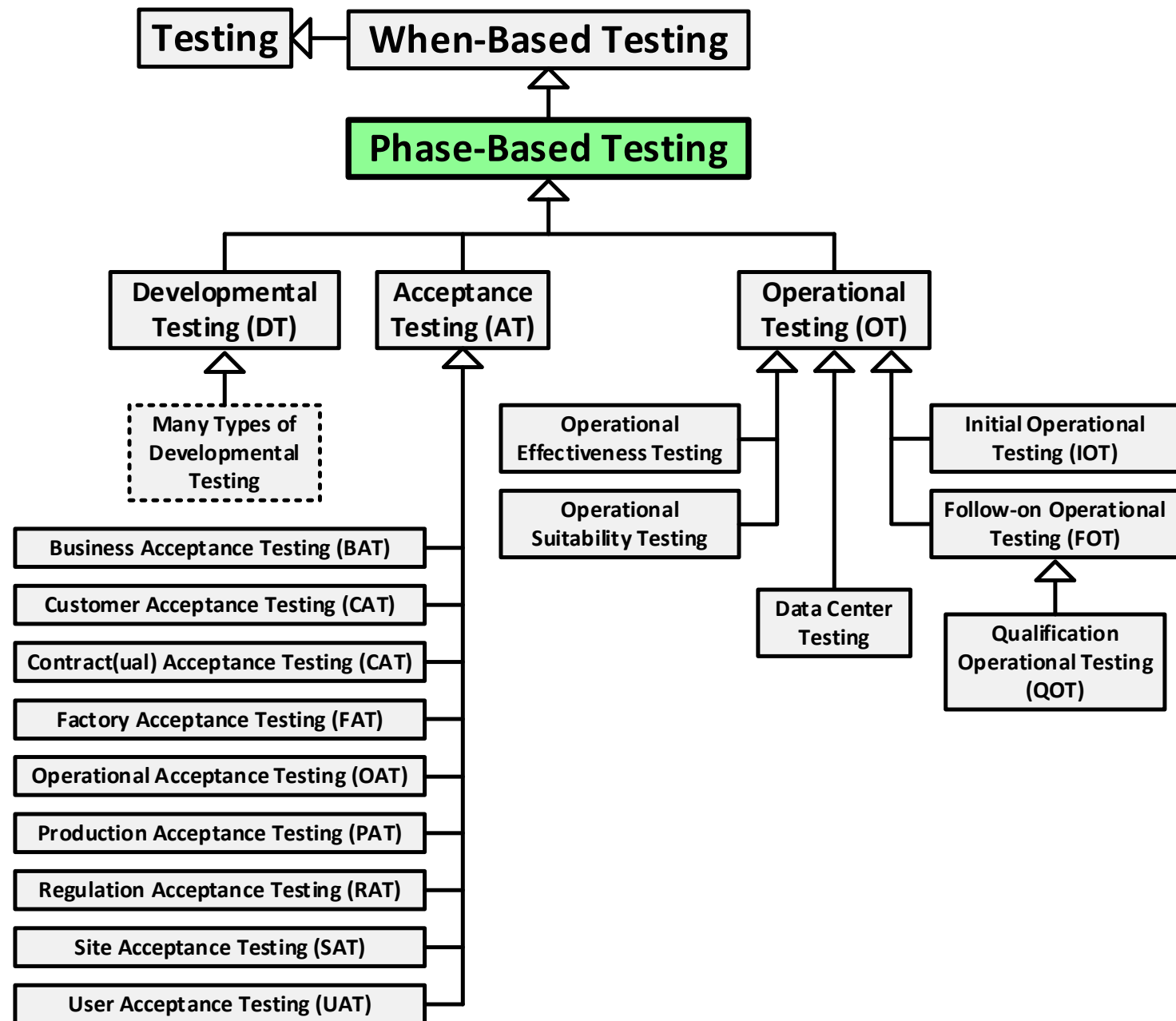
When: by Temporal Order



When: by Lifecycle

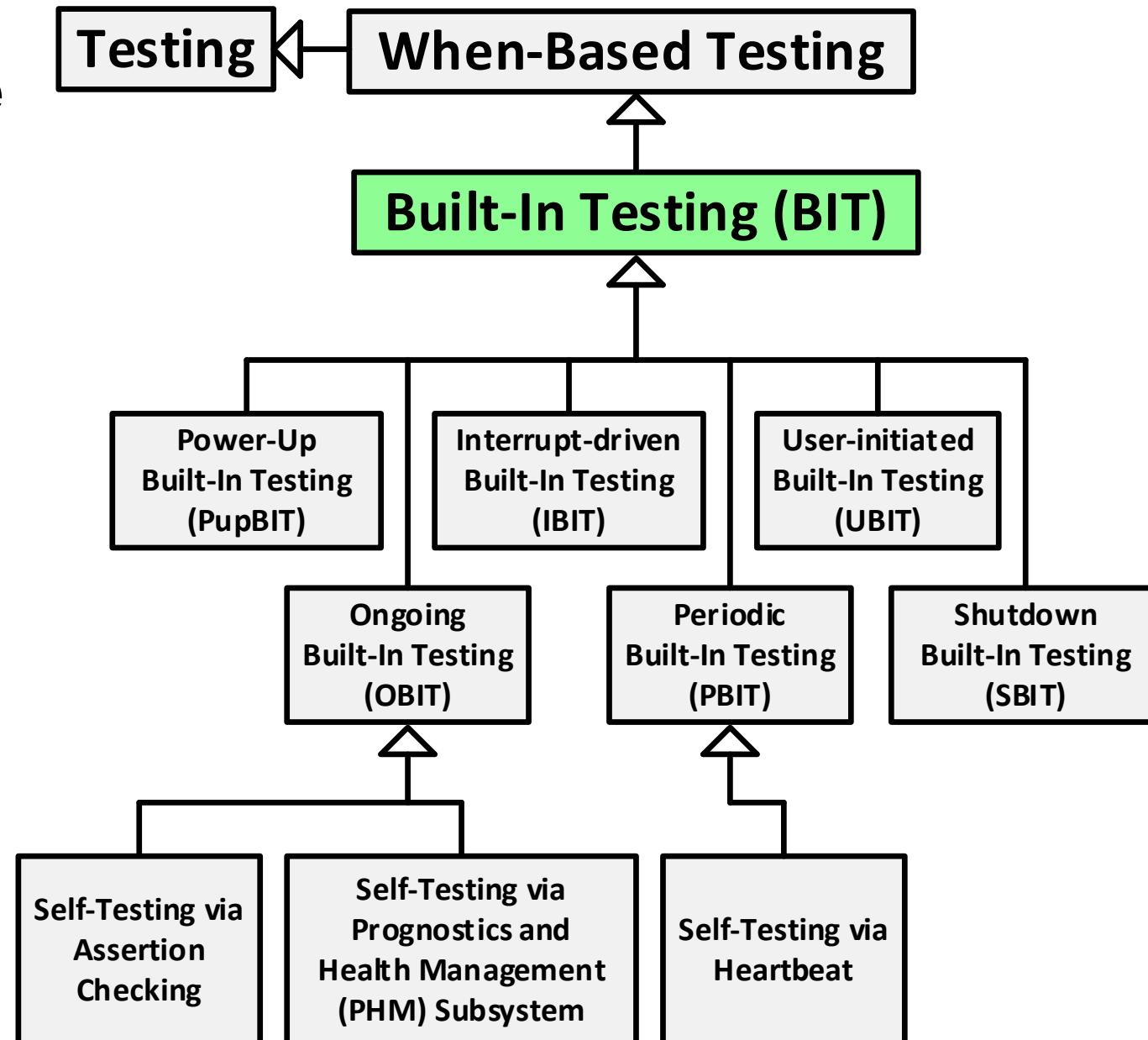


When: by Phase



Beware of
Synonyms and
Almost Synonyms!

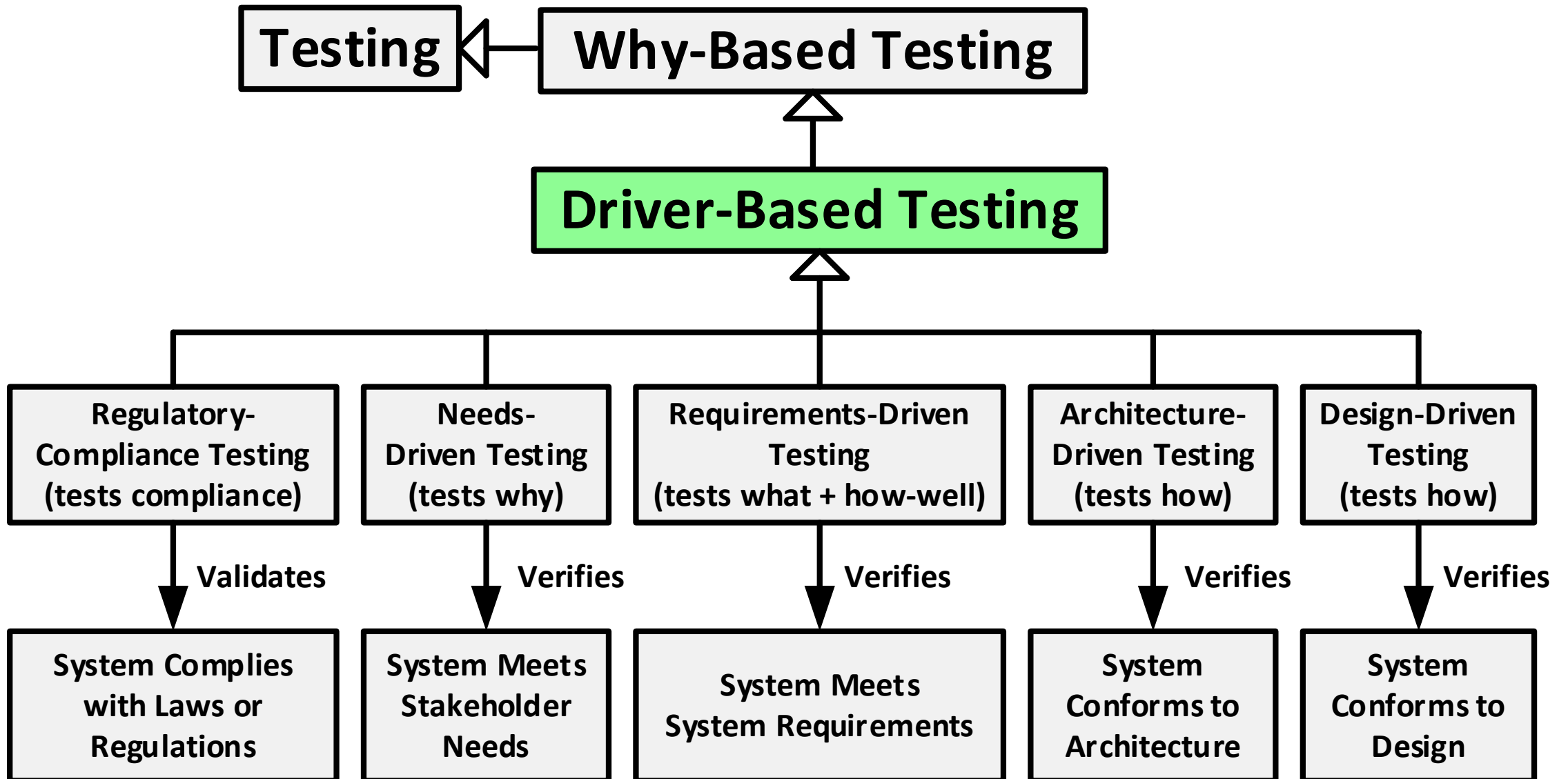
When: by BIT Execution Time



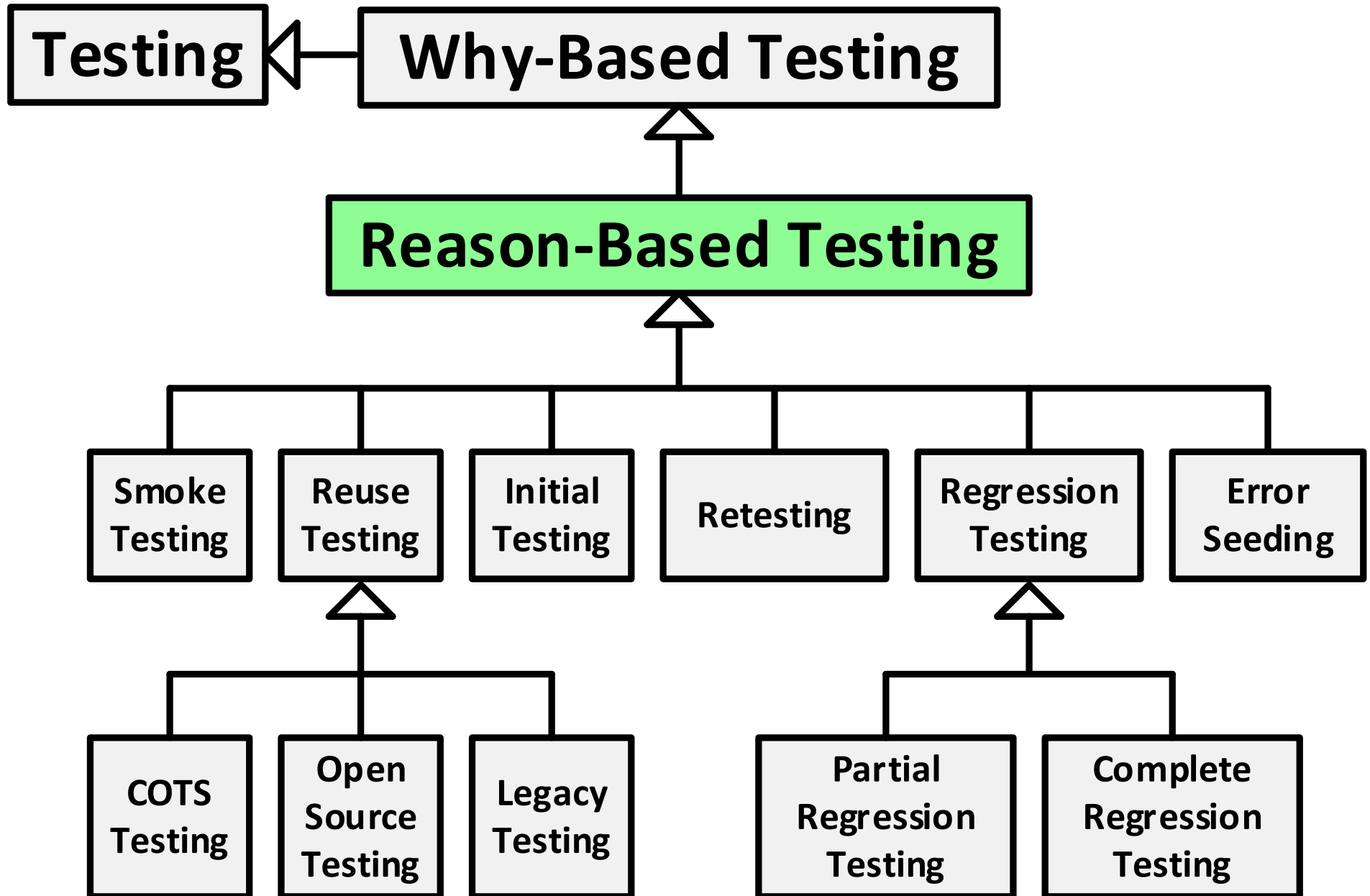
Types of Testing

WHY Testing is Being Performed

Why: by Driver



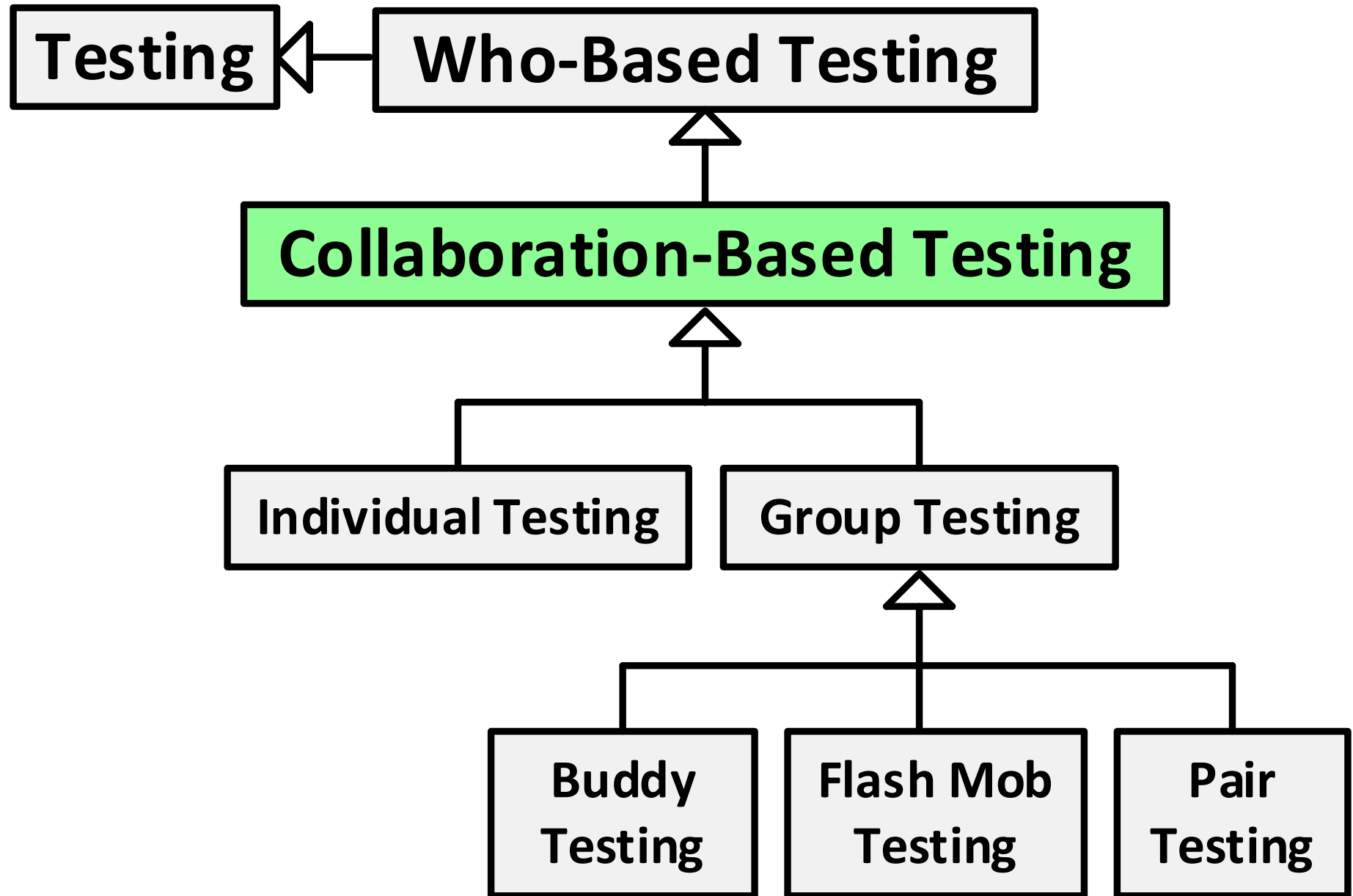
Why: by Reason



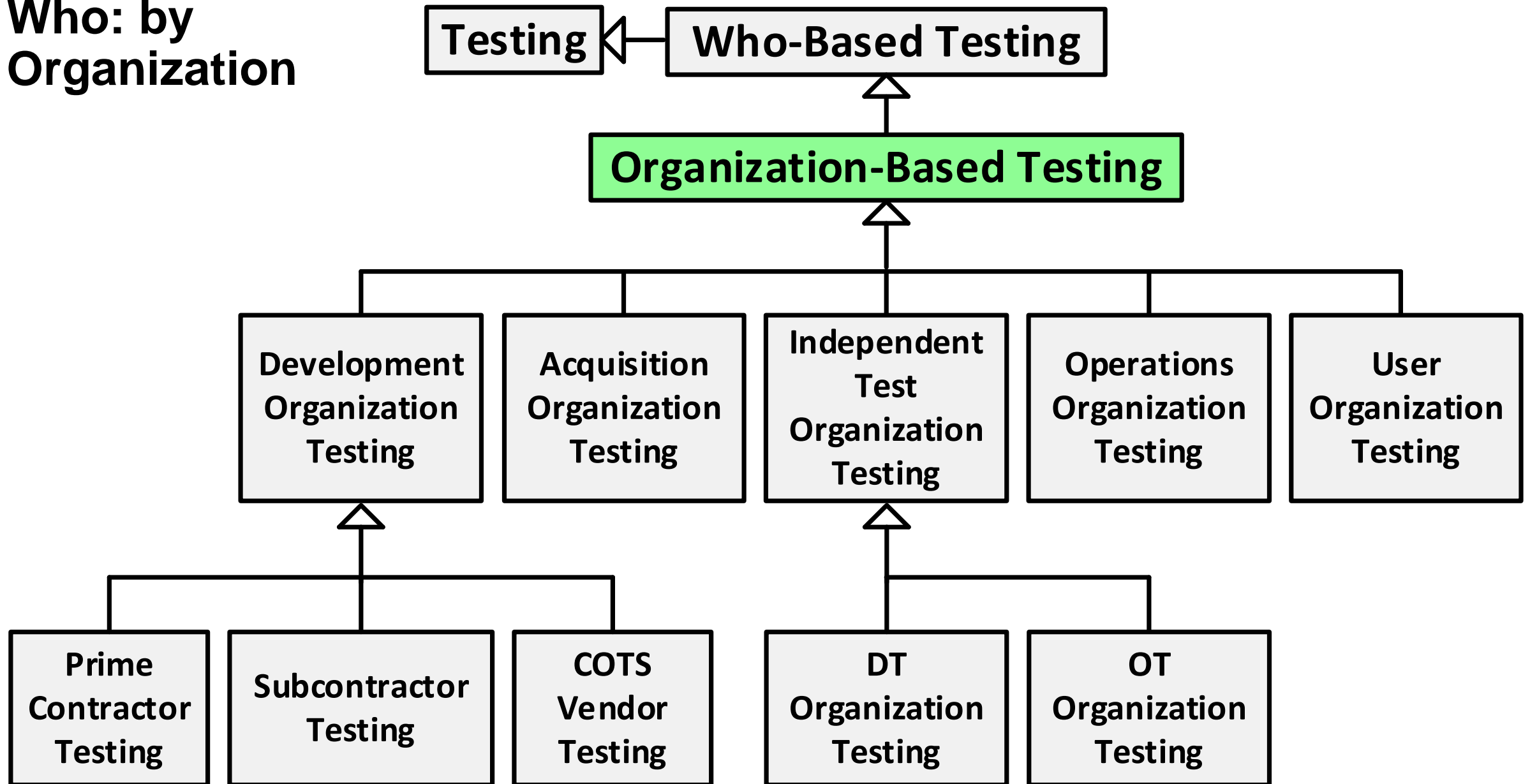
Types of Testing

WHO Performs Testing

Who: by
Collaboration



Who: by Organization

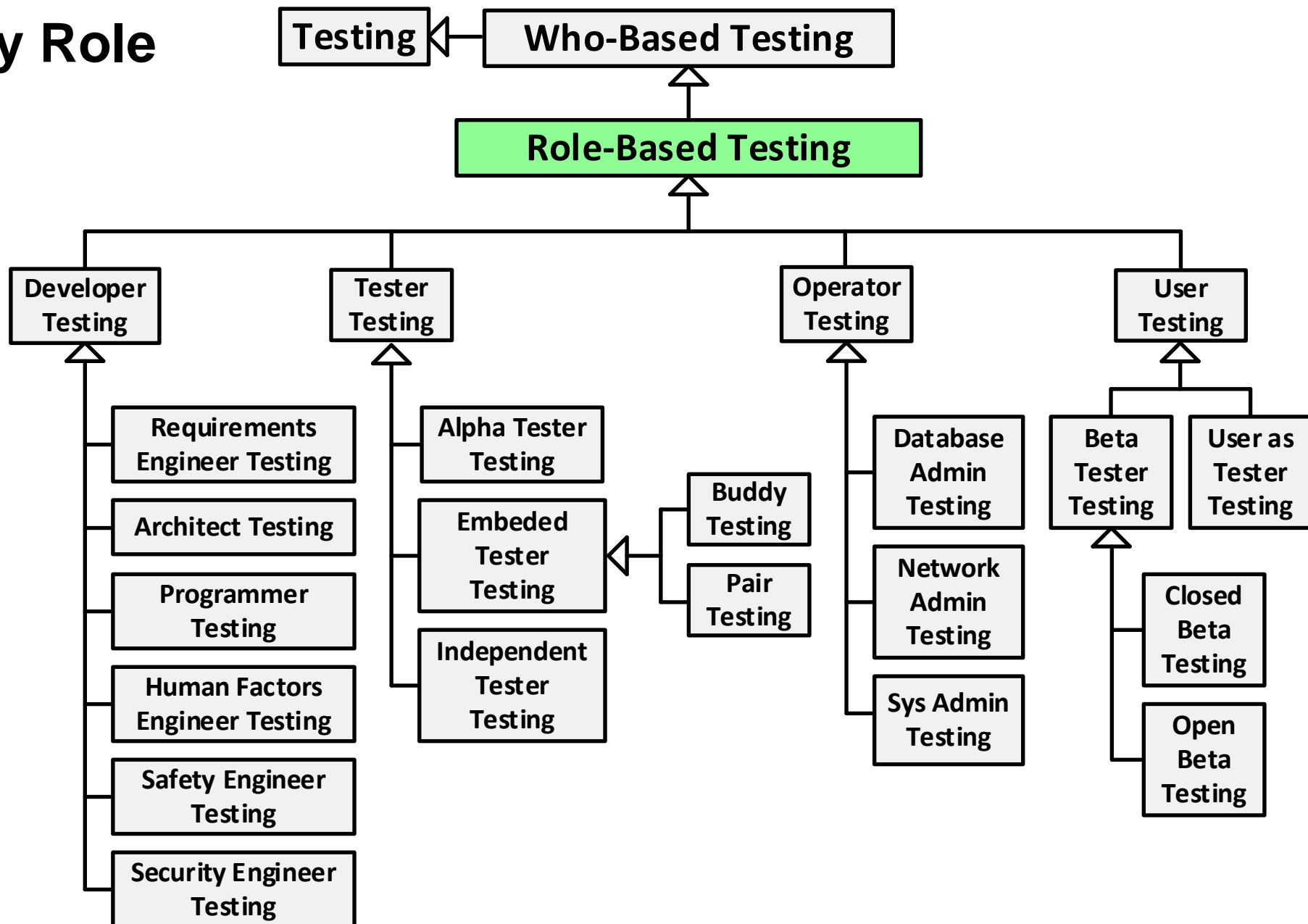


Polling Question 3

Who performs testing on your projects? Check all that apply.

- Project-internal Testers
- Independent Testers
- Developers
- Specialty Engineers (e.g., performance, reliability, safety, security, human factors)
- Quality Engineers
- Others

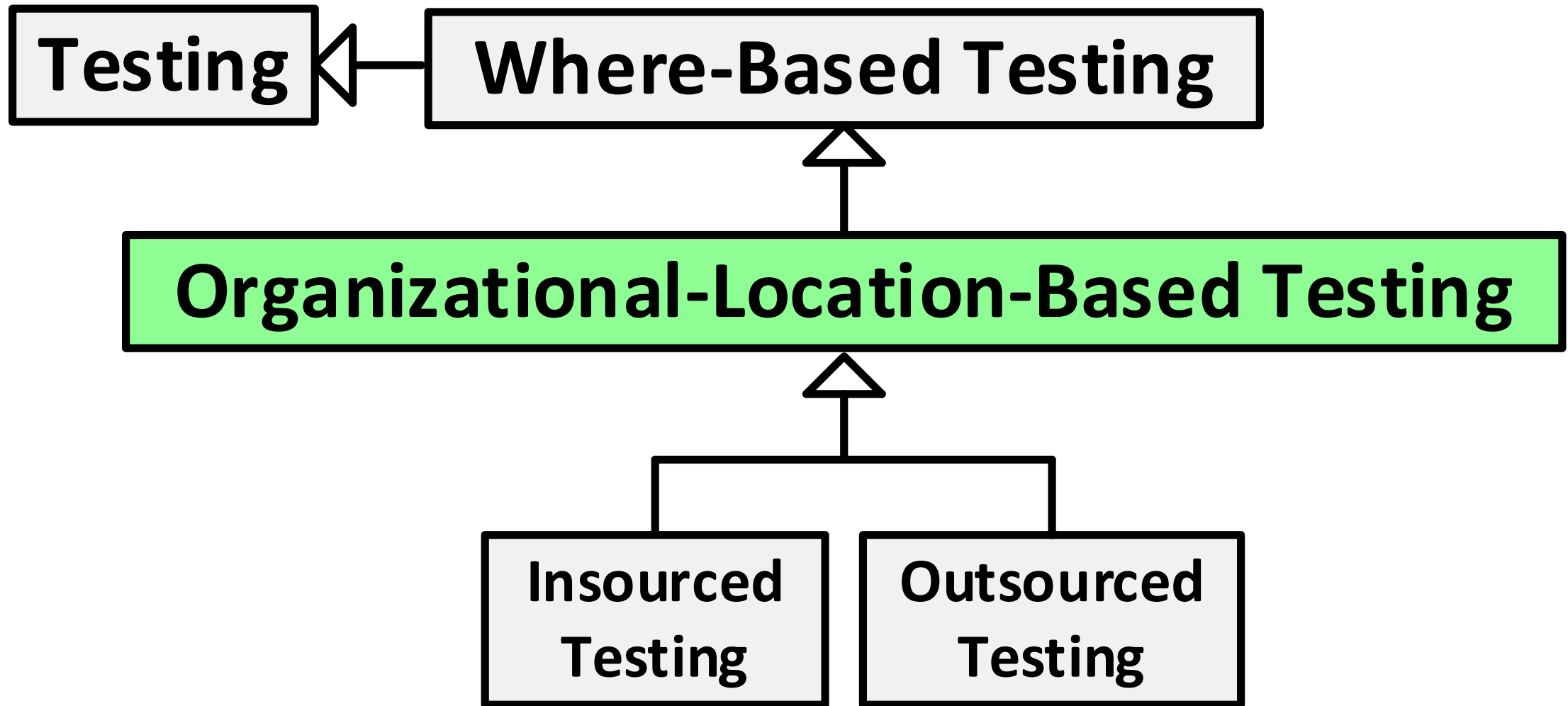
Who: by Role



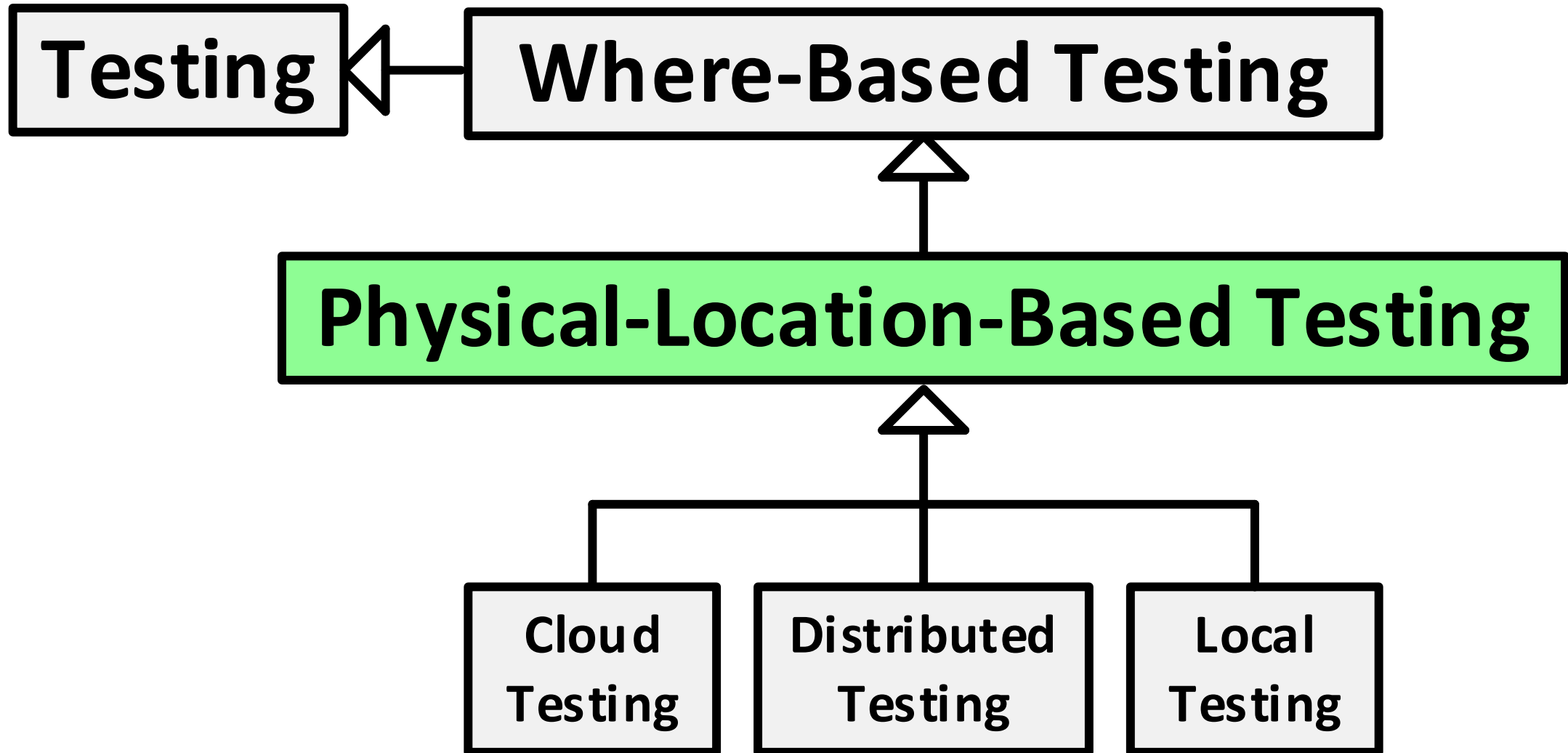
Types of Testing

WHERE Testing is Performed

Where: by Organizational Location



Where: by Physical Location

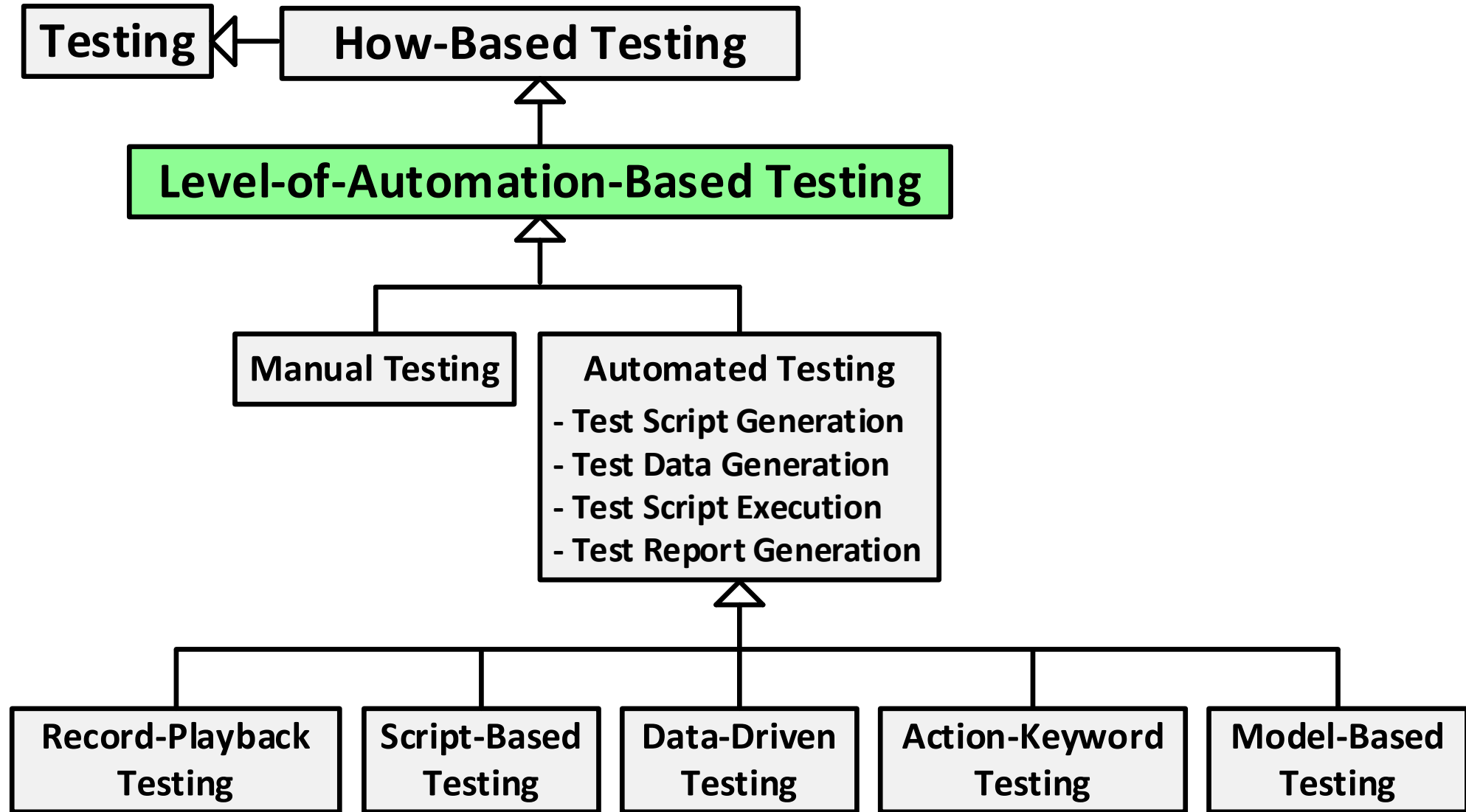


Types of Testing

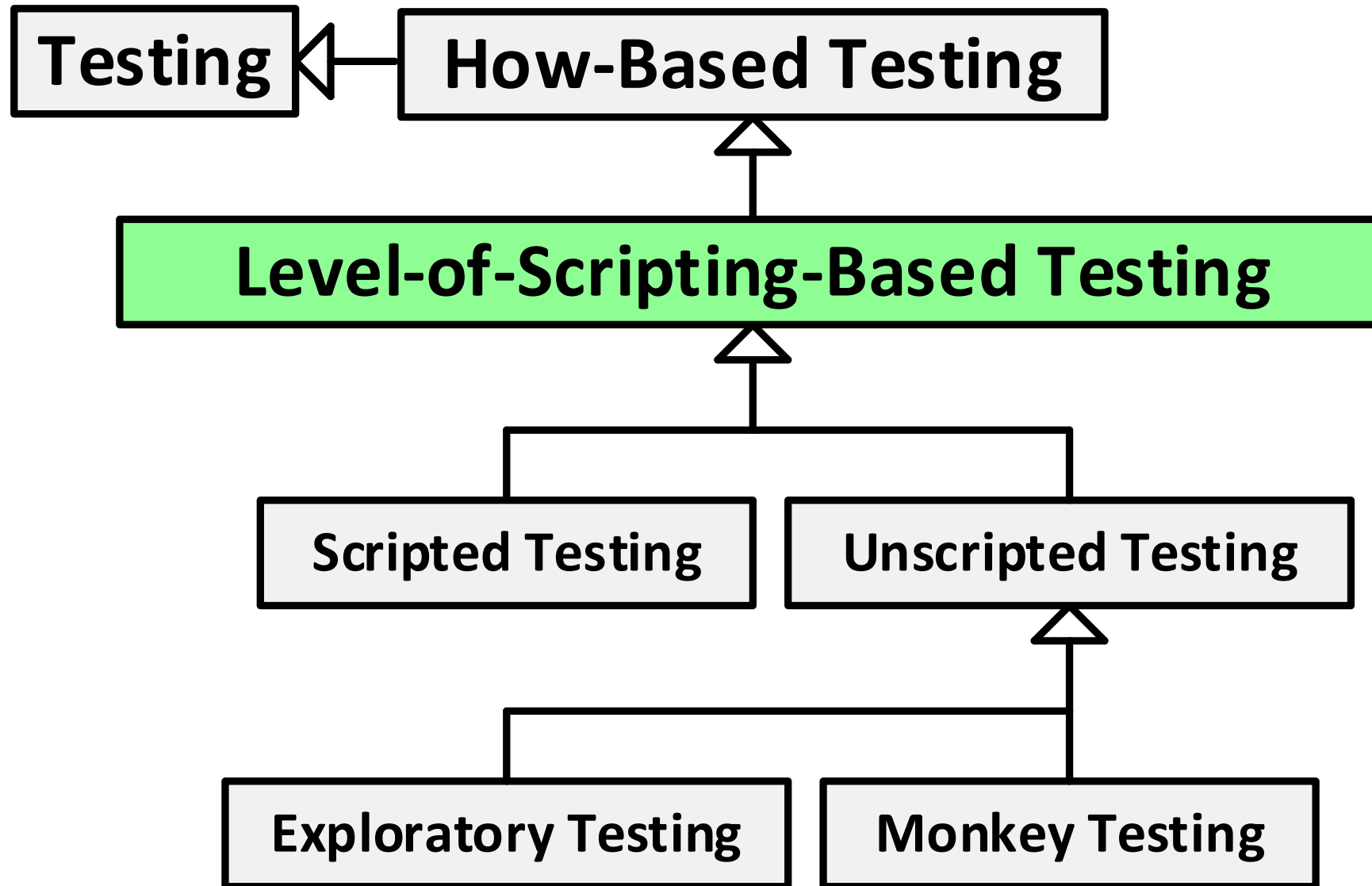
HOW Testing is Performed



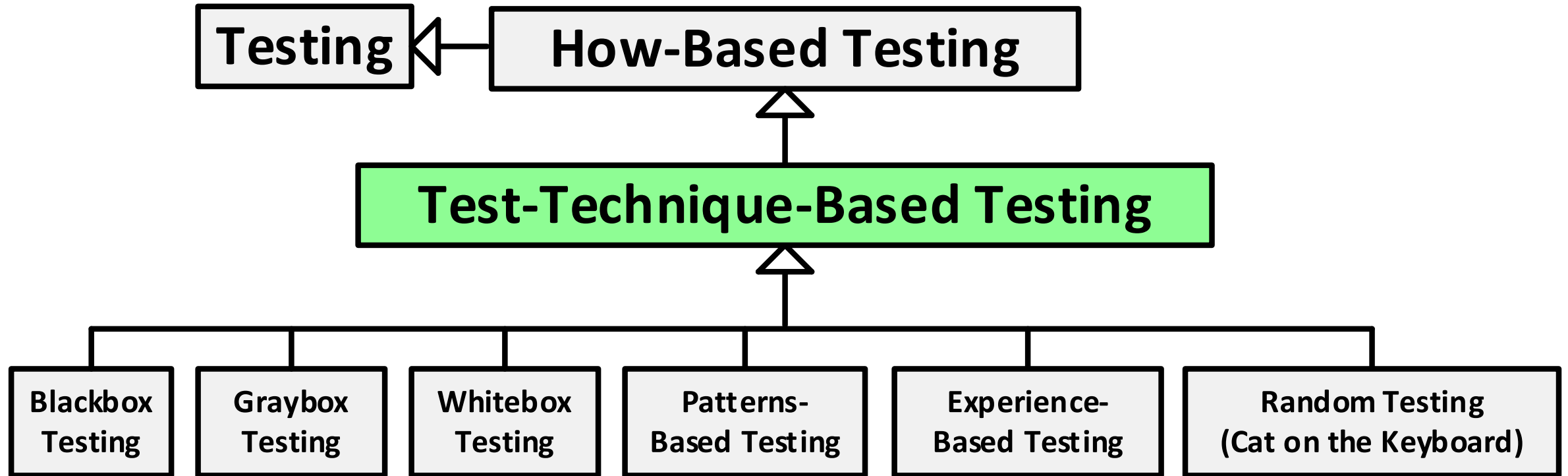
How: by Level of Automation



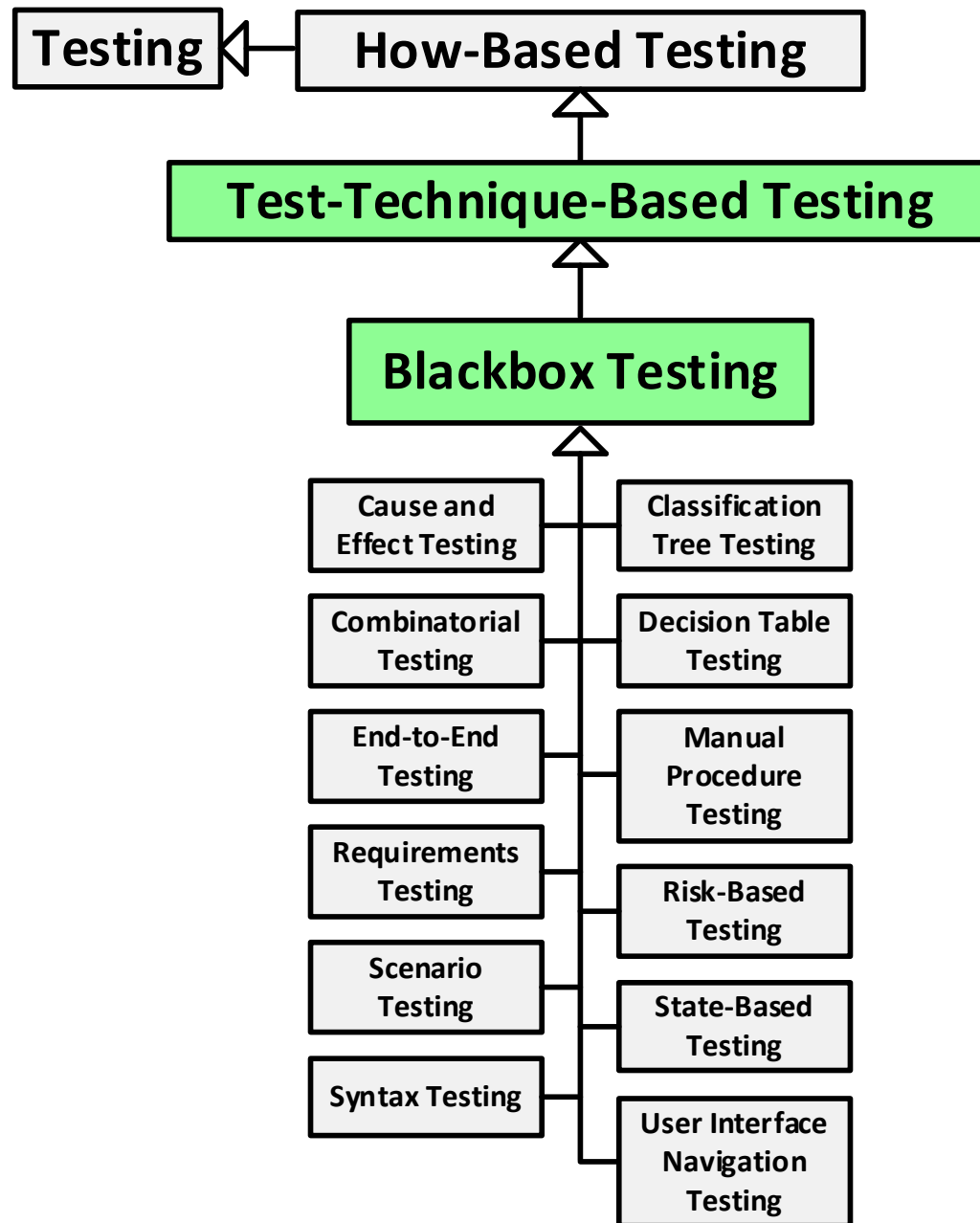
How: by Level of Scripting



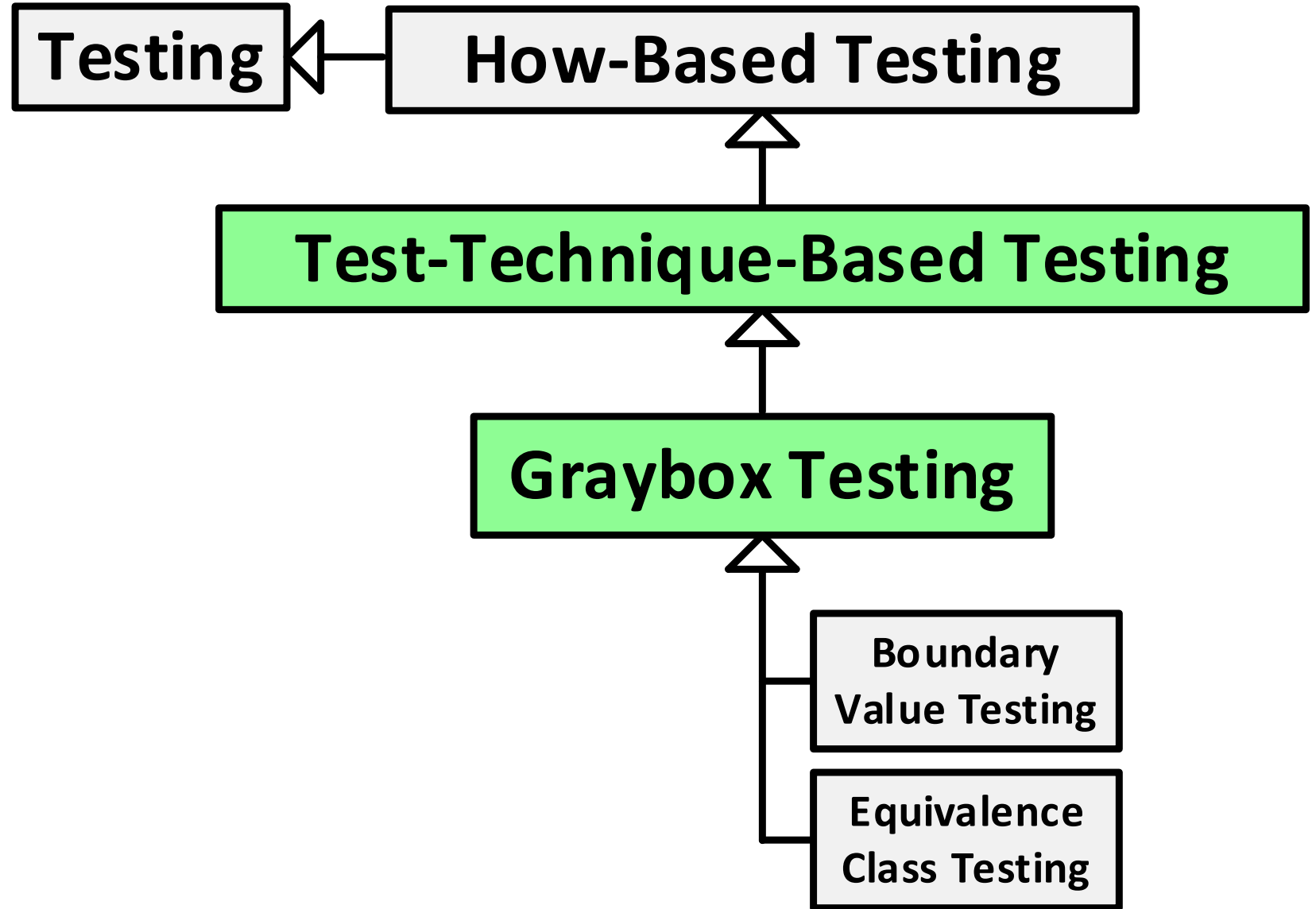
How: by Technique



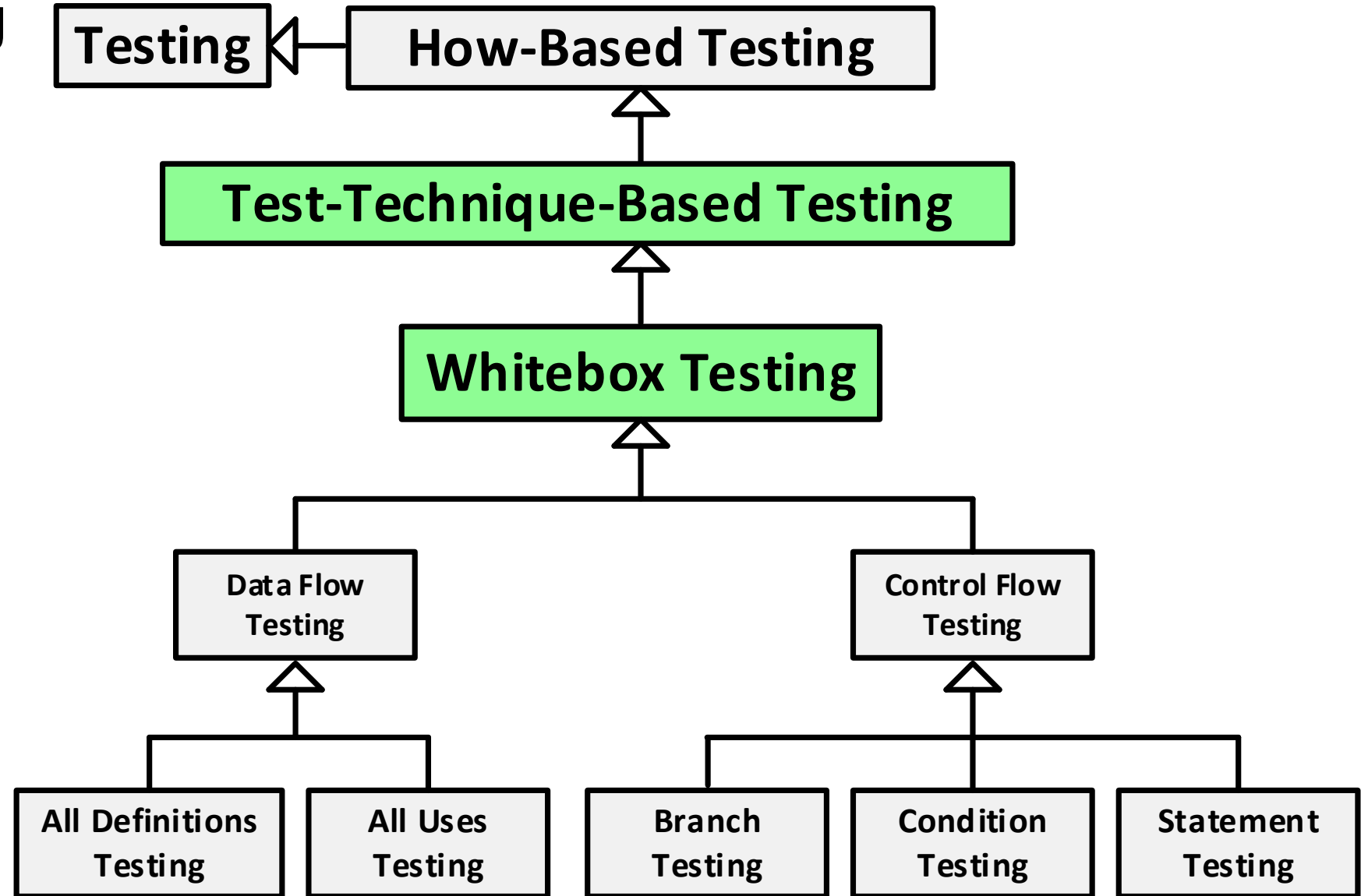
How: by Technique - Blackbox Testing



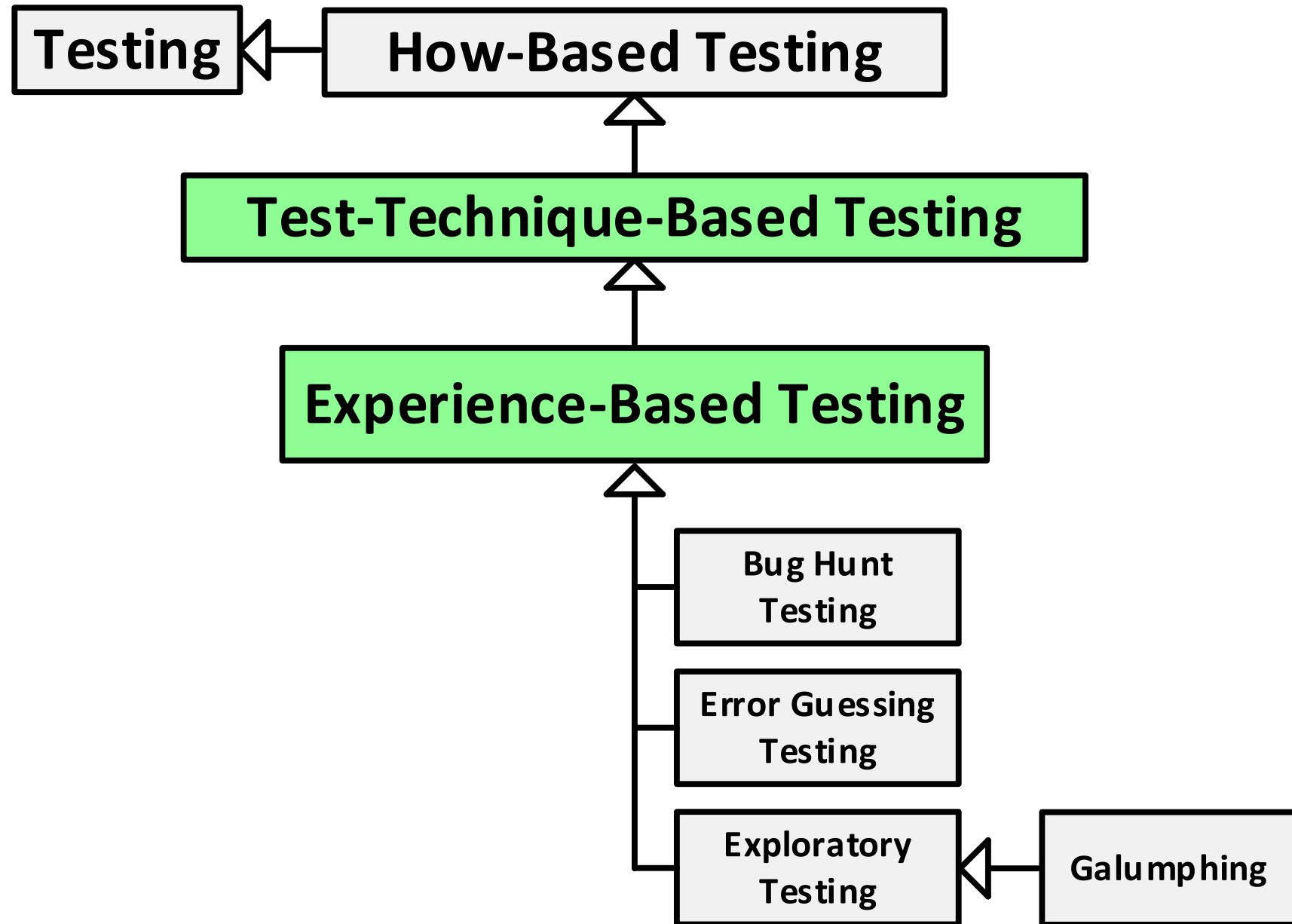
How: by Technique - Graybox Testing



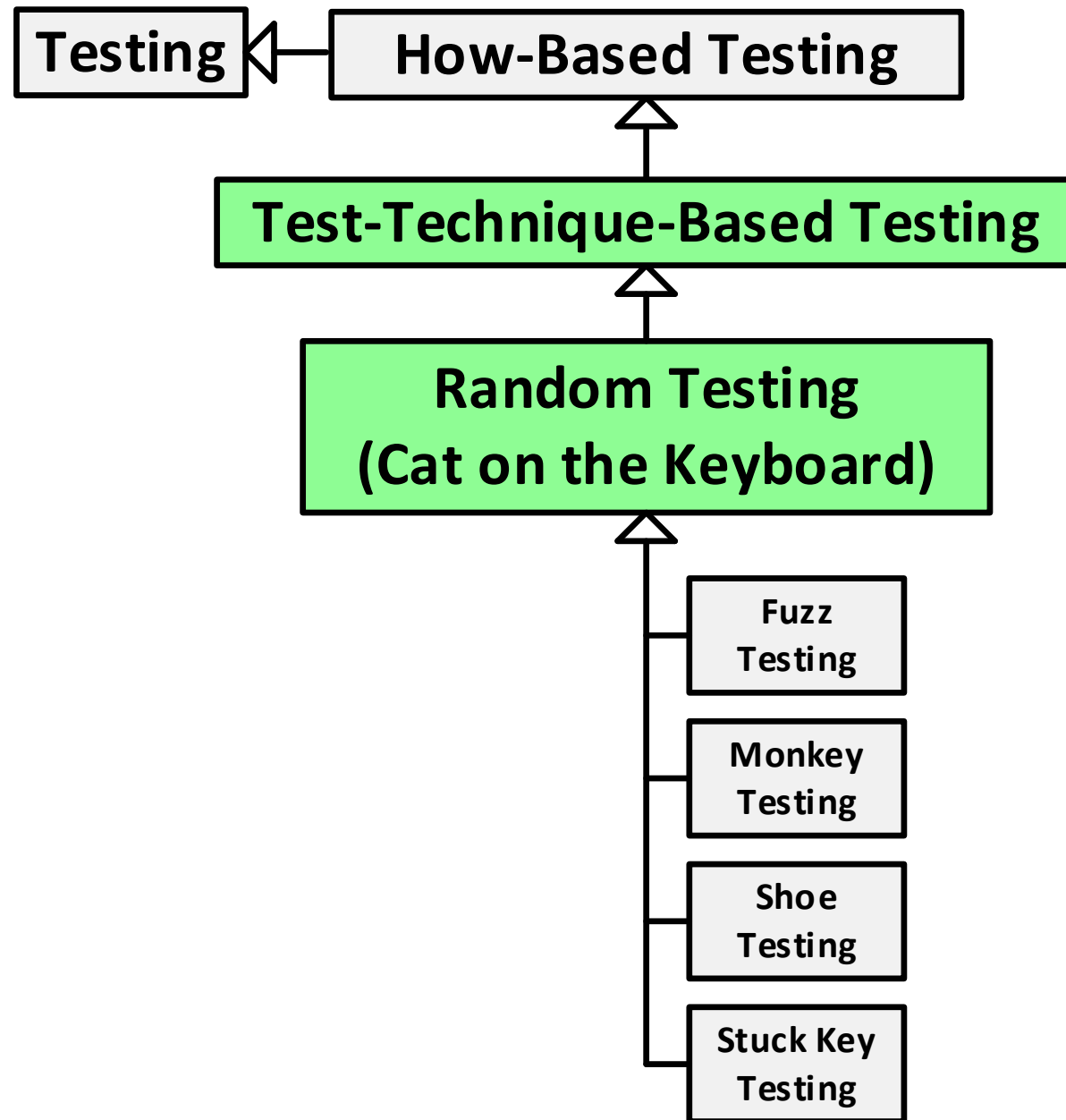
How: by Technique - Whitebox Testing



How: by Technique - Experience- Based Testing



How: by Technique - Random Testing



Types of Testing

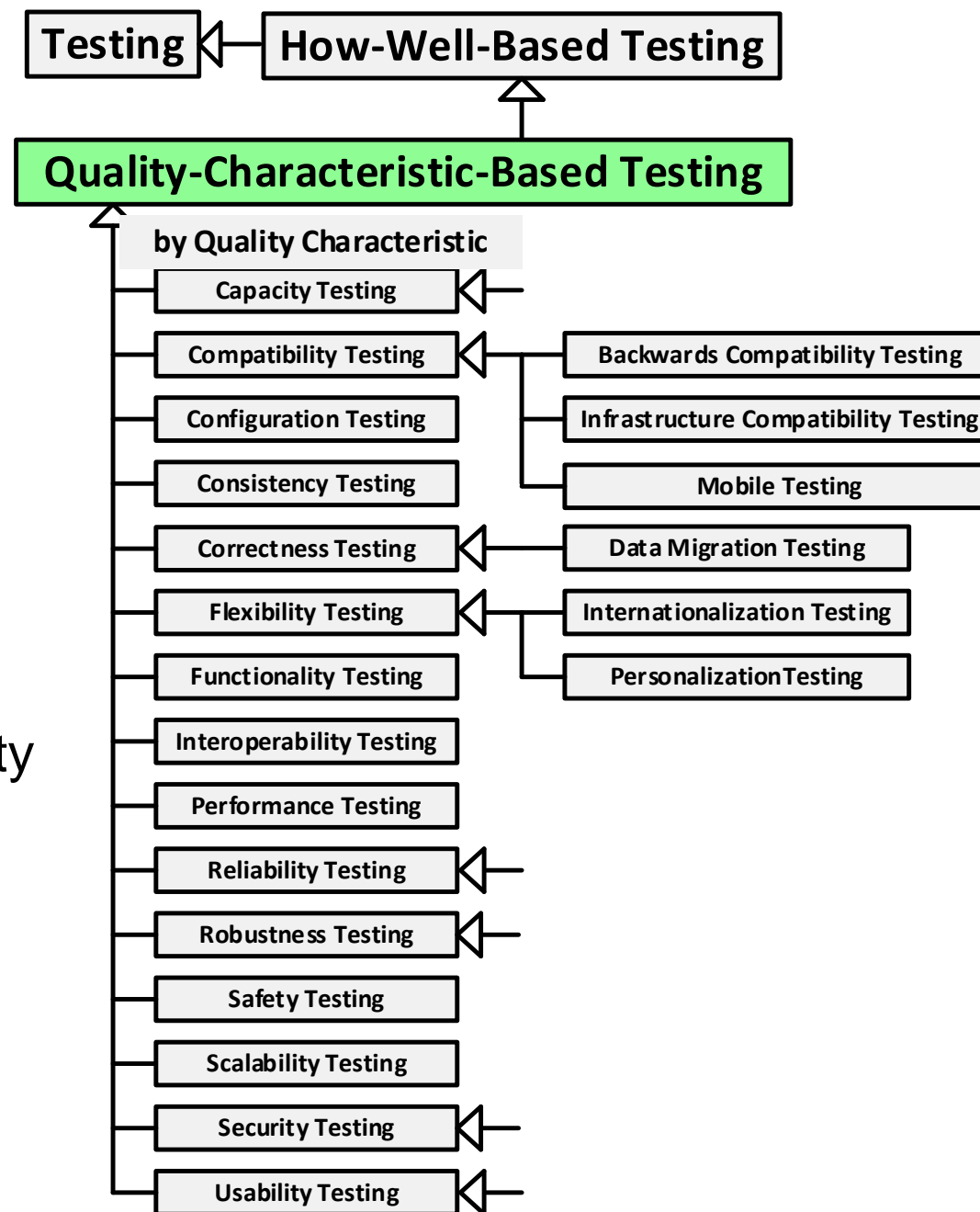
HOW WELL Object Under Test Functions



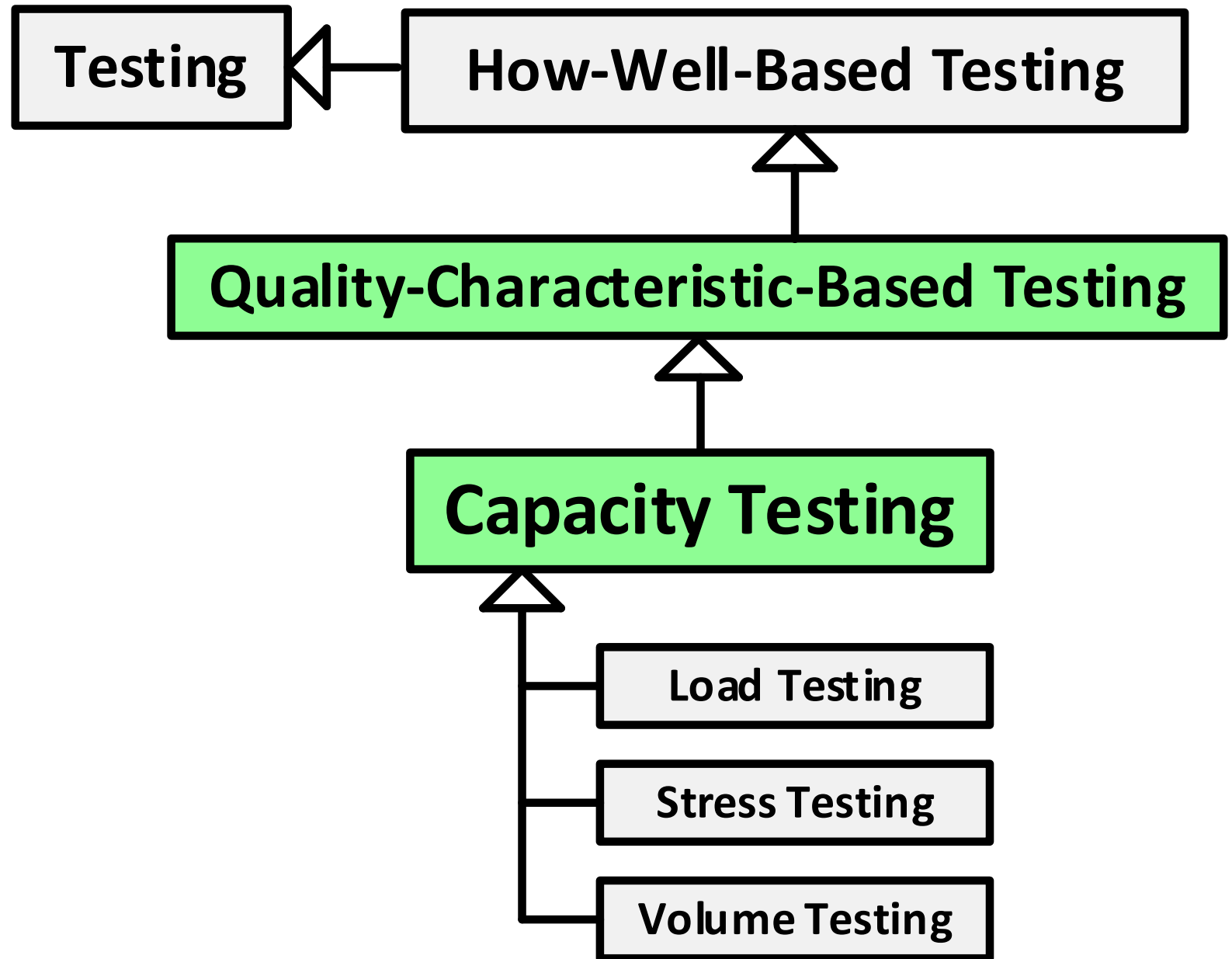
How Well: by Quality Characteristic

Based on the associated
quality characteristic
and its associated
quality attributes:

- Uncover related defects
- Determine level of quality

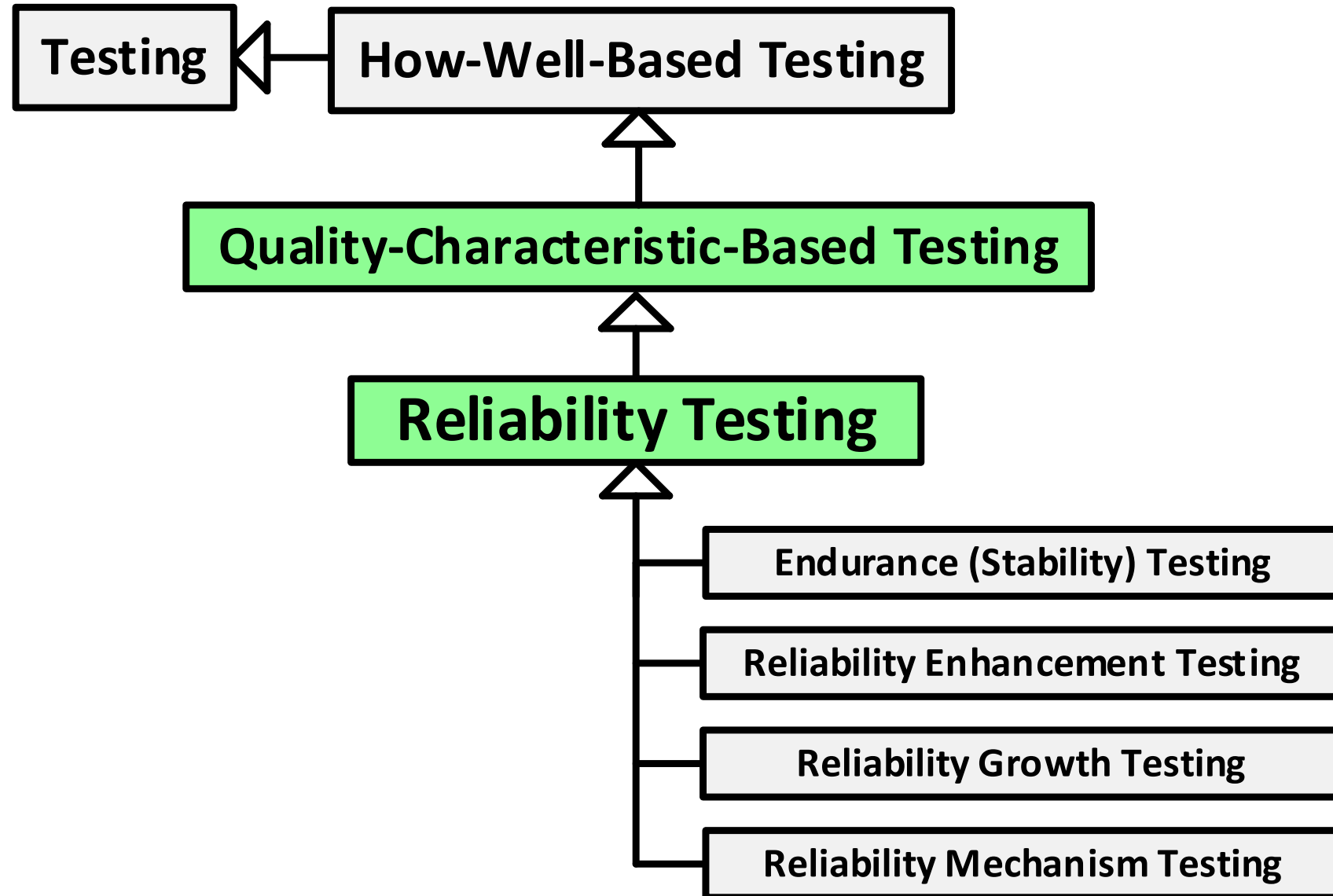


How Well: by Quality – Capacity Testing

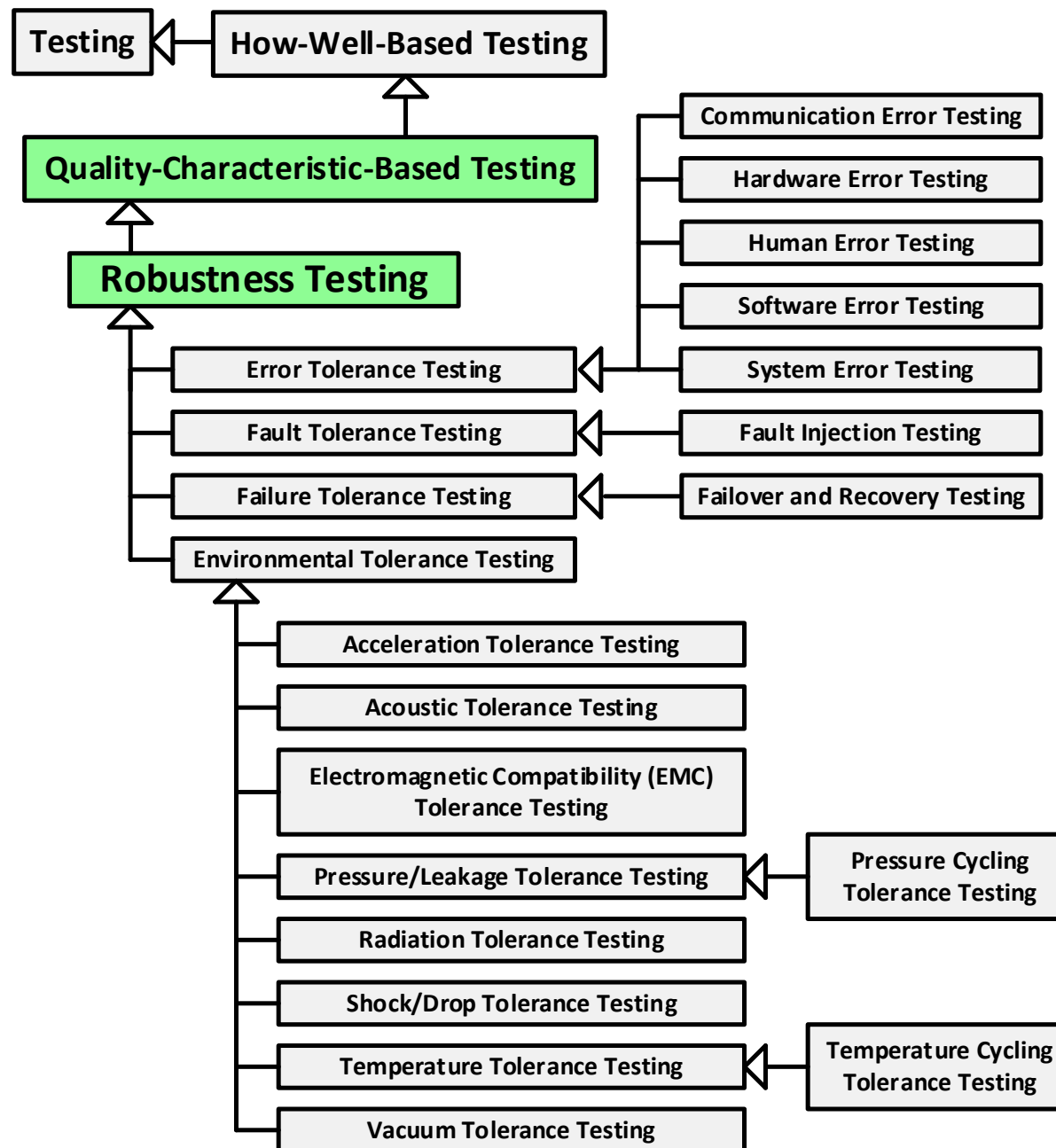


How Well: by Quality

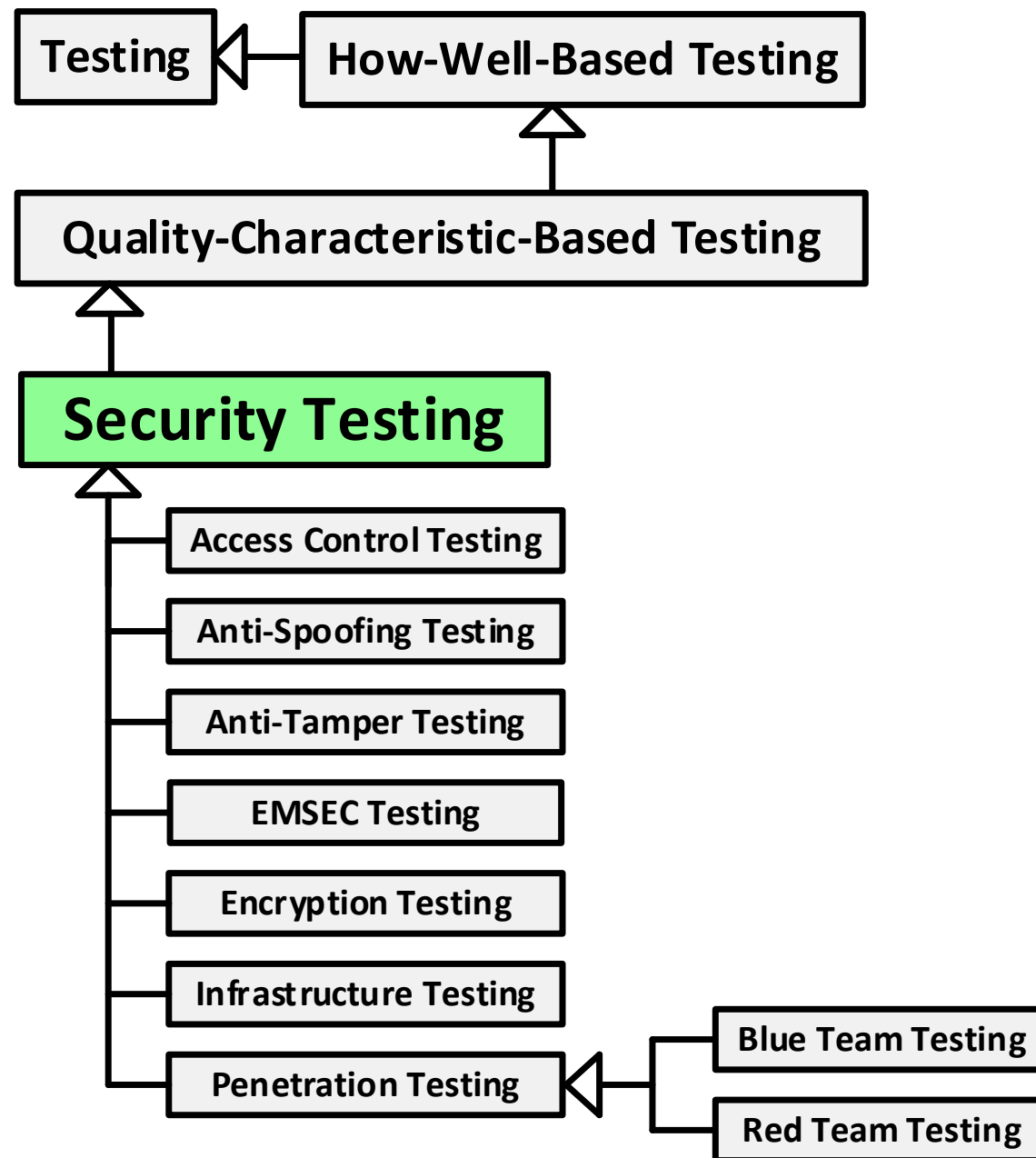
– Reliability Testing



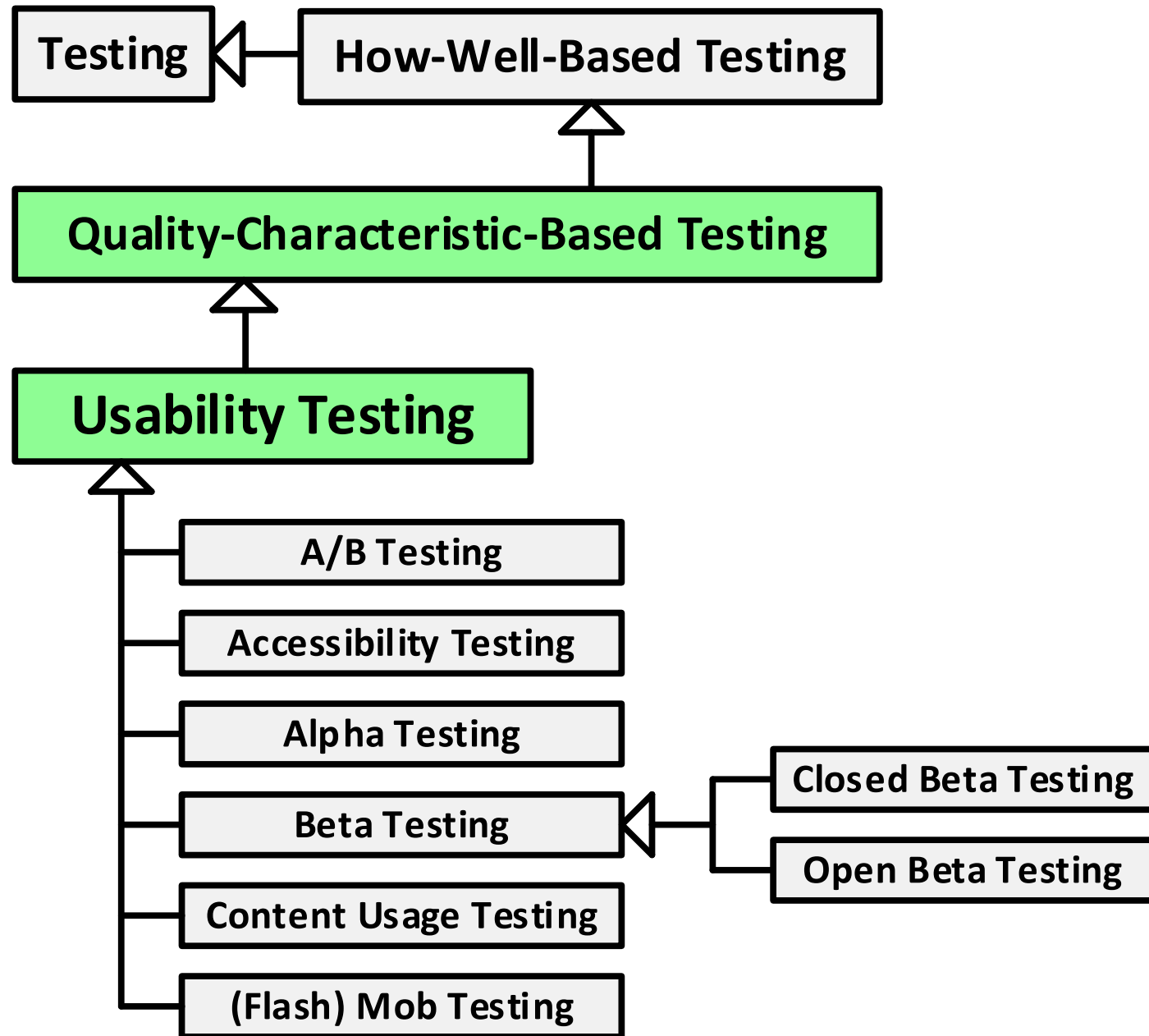
How Well: by Quality – Robustness Testing



How Well: by Quality – Security Testing



How Well: by Quality – Usability Testing



Conclusion

Conclusion

Most systems require quite a few different types of testing.

Most testers are not aware of the majority of the different types of testing.

- If you are not aware that it exists, then you don't know whether you need it.

These types of testing can be organized into a taxonomy by the 5W + 2H questions.

This taxonomy has several uses:

- Ensure the test strategy is sufficiently complete with no important type of testing overlooked.
- Organize testing types to make them and their relationships more understandable.
- Augment test training materials.
- Help categorize and understand limitations of testing tools.

Q&A



Software Solutions Conference 2015

November 16-18, 2015 • Crystal City, VA