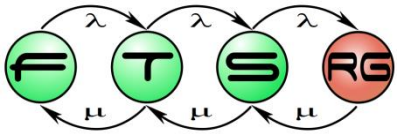
Overview of testing

Zoltan Micskei, Istvan Majzik

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group



Main topics of the course

- Overview (1.5)
  - Introduction, V&V techniques
- Static techniques (1.5)
  - Specification, Verifying source code
- Dynamic techniques: Testing (7)
  - **Testing overview**, Test design techniques
  - Test generation, Automation
- System-level verification (3)
  - Verifying architecture, Dependability analysis
  - Runtime verification

Dynamic techniques

WHAT: code or other executable artefact

HOW: with execution

USING: testing, runtime verification…

Is this really needed / good for me?

> The second priority is to attract/reskill Software Development Engineers for Test skills (SDET). The SDET must have advanced automation skills, white box testing capabilities, development skills, and the ability to build orchestration platforms. They may also be required to possess basic algorithmic application capabilities, and natural language processing skills in the event of it being an AI application.
>
> 2

Source: WQR

RECAP: WHAT IS TESTING?

Definitions, Goals, Approaches

Learning outcomes

- Recall different definitions of testing (K1)
- Explain goals and basic concepts of testing (K2)

Definition of testing (1)

„Testing is an activity performed for evaluating product quality, and for improving it, by identifying defects and problems.”

Source: IEEE, „Software Engineering Body of Knowledge” (SWEBOK) 2004
URL: http://www.computer.org/portal/web/swebok/

Definition of testing (2)

„An activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component.”

Source: IEEE, "Systems and software engineering — Vocabulary," ISO/IEC/IEEE Standard 24765, 2010

Definition of testing (3)

„The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products

- to determine that they satisfy specified requirements,
- to demonstrate that they are fit for purpose and
- to detect defects.

Source: International Software Testing Qualifications Board (ISTQB), URL: http://istqb.org/

Definition of testing (4)

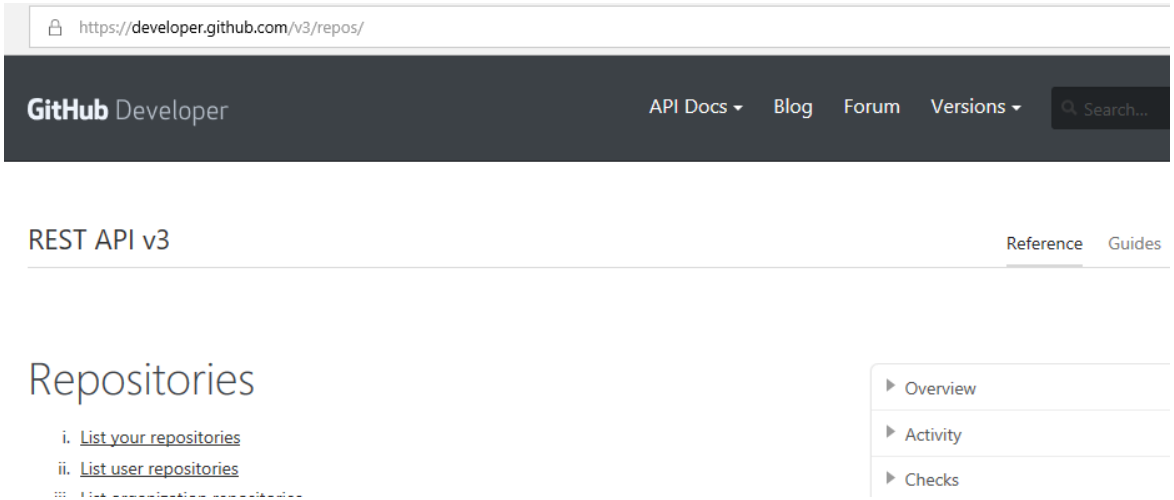
"Testing is the process of evaluating a product by learning about it through exploration and experimentation, which includes: questioning, study, modeling, observation and inference, output checking, etc."

Source: James Bach, Micheal Bolton. Exploratory Testing 3.0

EXERCISE: Testing Mindset

"As an org admin I want to query forks in a GitHub organization to see what external repos we are using."

How do we begin testing this story?



For a more detailed example about a testing thought process see:
http://www.developsense.com/blog/2018/07/exploratory-testing-on-an-api-part-2/

Possible goals of testing

Testing “approaches”

Test-as-information-provider

- Test-last
- Independent test team
- Separate test phase
- Fixed releases

Test-as-quality-accelerant

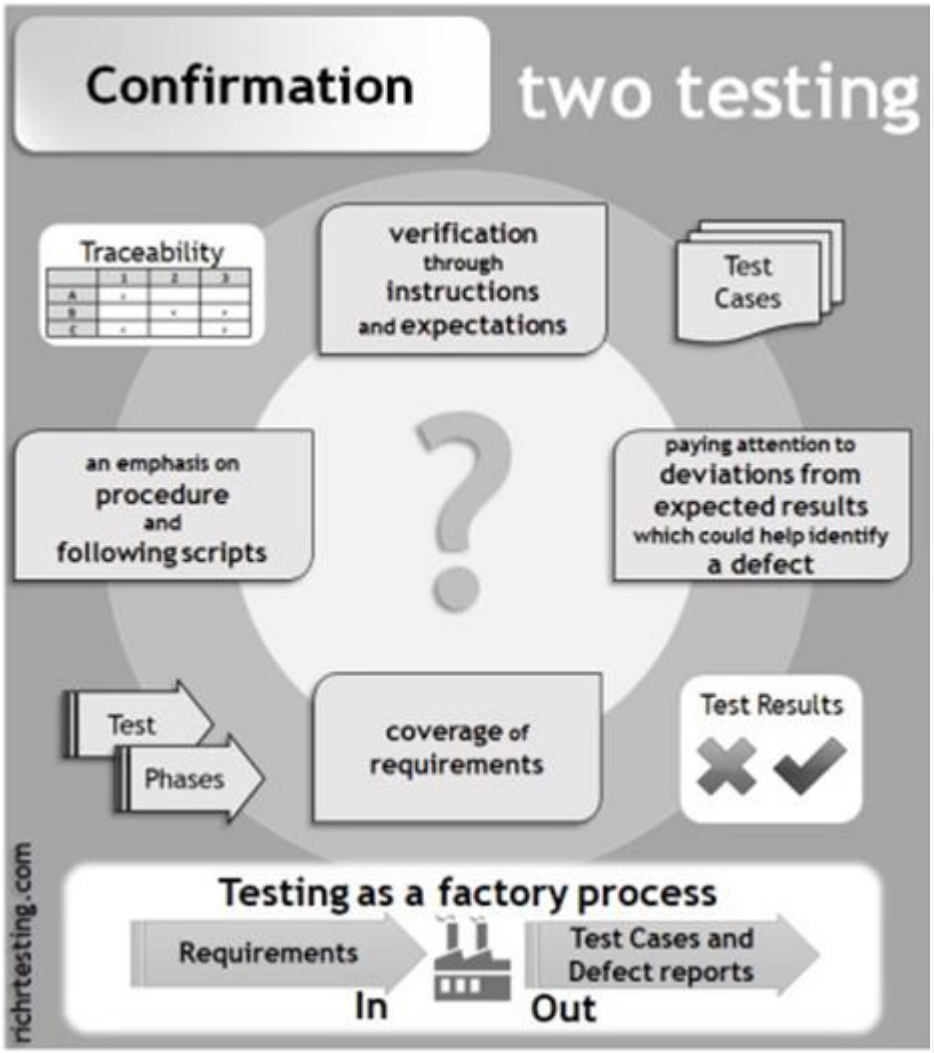
- Test-always
- Testers quality assistants
- Developers write tests
- Release often/always

Source: Alan Page. “Two new… schools”

See also: Modern Testing Principles

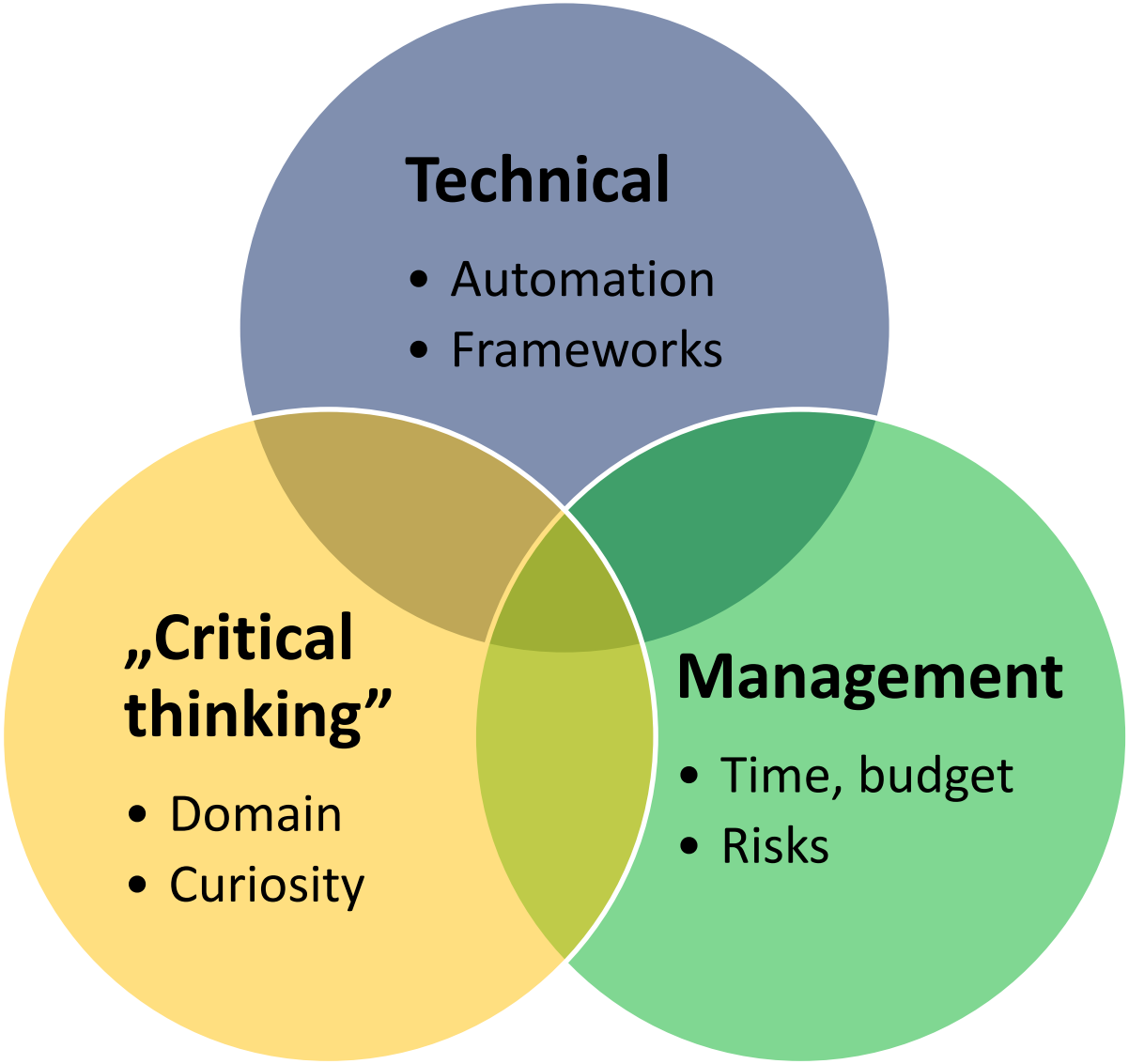
“*Accelerate the Achievement of Shippable Quality*”

Confirmation and Exploration



Source: Rich Rogers. "Confirmation and Exploration"

Viewpoints in testing



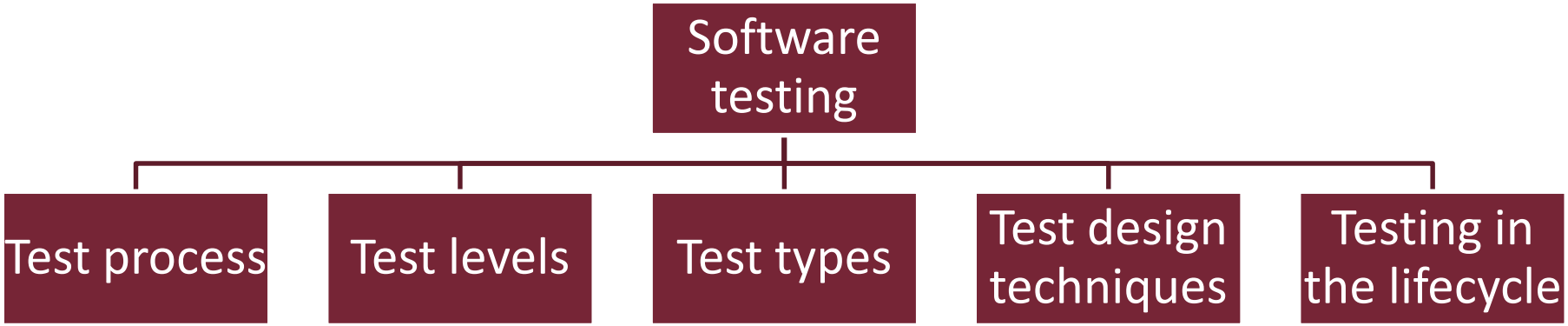
TEST PROCESS

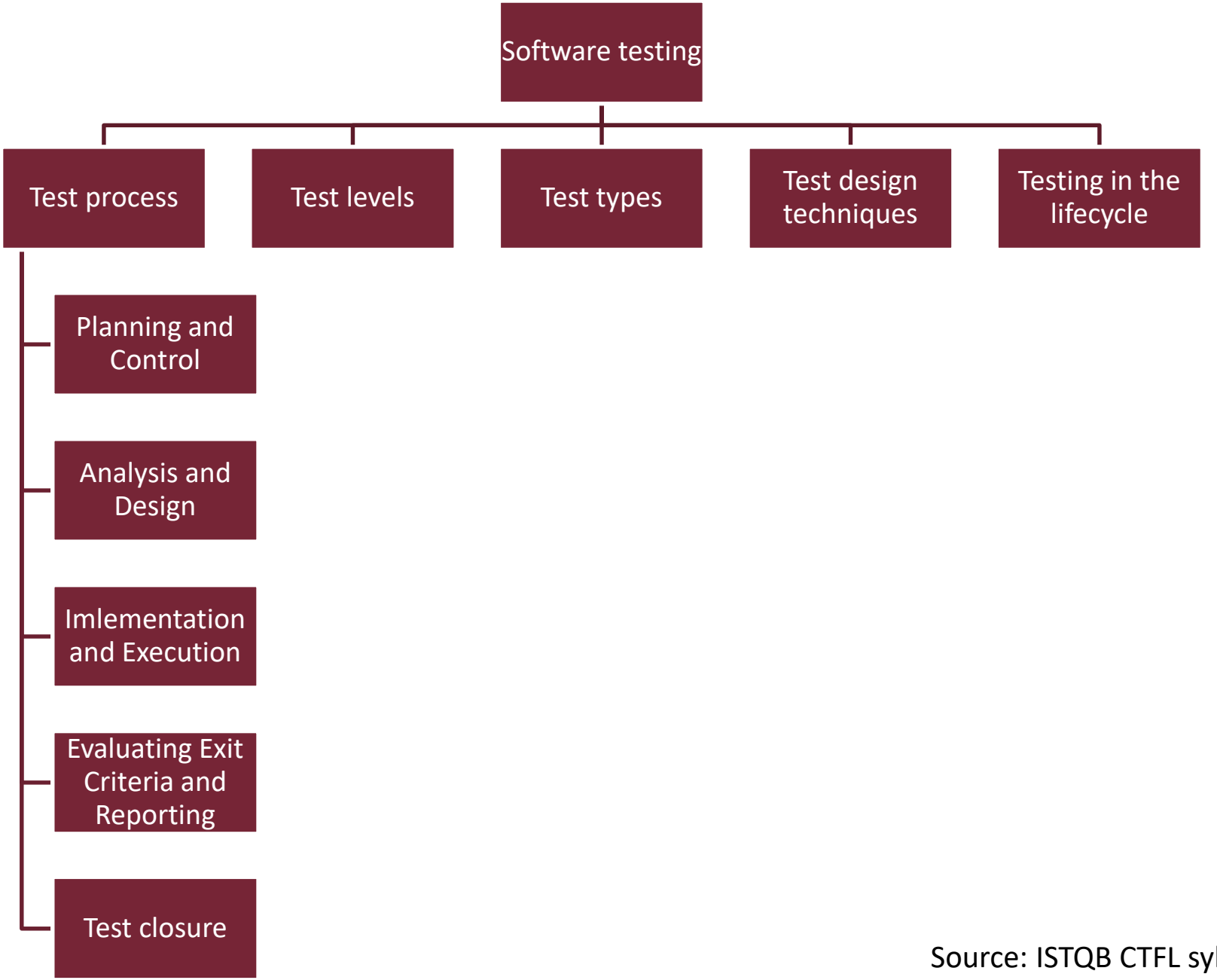
Test activates, Test planning

Learning outcomes

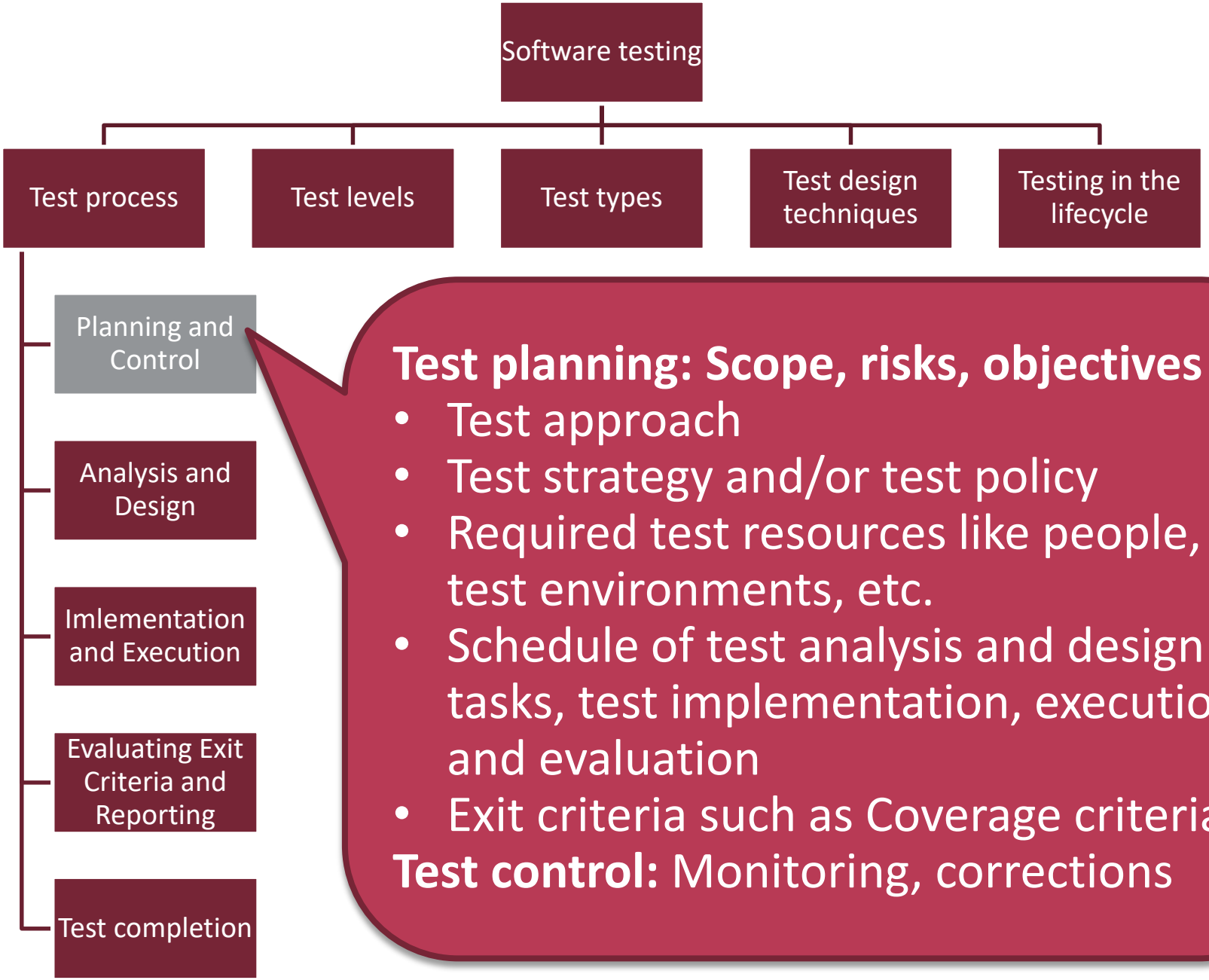
- Explain the activities and tasks in the typical test process (K2)
- Describe methods for test planning (K2)

Overview of testing concepts





Source: ISTQB CTFL syllabus


Software testing
Test process
Test levels
Test types
Test design techniques
Testing in the lifecycle
Planning and Control
Analysis and Design
Imlementation and Execution
Evaluating Exit Criteria and Reporting
Test completion
Test planning: Scope, risks, objectives
• Test approach
• Test strategy and/or test policy
• Required test resources like people, test environments, etc.
• Schedule of test analysis and design tasks, test implementation, execution and evaluation
• Exit criteria such as Coverage criteria
Test control: Monitoring, corrections

Test strategy

- Guidelines for
  - What methodology?
  - What kinds of tests?
  - What tools?
  - Who will test?
  - Exit criteria?
  - What documentation?
- Possible (simple) example:
  - Test-driven development
  - Module & system
  - JUnit & GUI Tester
  - Developers & test engineers
  - At least 90% statement coverage & cover every use case / requirement
  - Test Report according to IEEE 29119-3

See Annex F of ISO 29119-3:2013 for a more complex example

Test plan

- Mapping test strategy to the actual test project
  - Test objectives
  - Test objects, test environment
  - Resources, roles
  - Schedules

- Defining test phases
  - Length of phase
  - Exit criteria
  - Measuring quality of testing

Test plan outline: ISO 29119-3:2013(E)

a) Context of the testing:
   - i) Project/Test sub-process
   - ii) Test item(s)
   - iii) Test scope
   - iv) Assumptions and constraints
   - v) Stakeholders

b) Testing communication

c) Risk register:
   - i) Product risks
   - ii) Project risks

d) Test strategy:
   - i) Test sub-processes
   - ii) Test deliverables
   - iii) Test design techniques
   - iv) Test completion criteria
   - v) Metrics to be collected
   - vi) Test data requirements
   - vii) Test environment requirements
   - xi) Retesting and regression testing
   - xii) Suspension and resumption criteria
   - xiii) Deviations from the Organizational T

e) Testing activities and estimates

f) Staffing:
   - i) Roles, activities, and responsibilities
   - ii) Hiring needs
   - iii) Training needs

g) Schedule

See Annex F of the standard for examples (agile / traditional)

Risk-based Testing

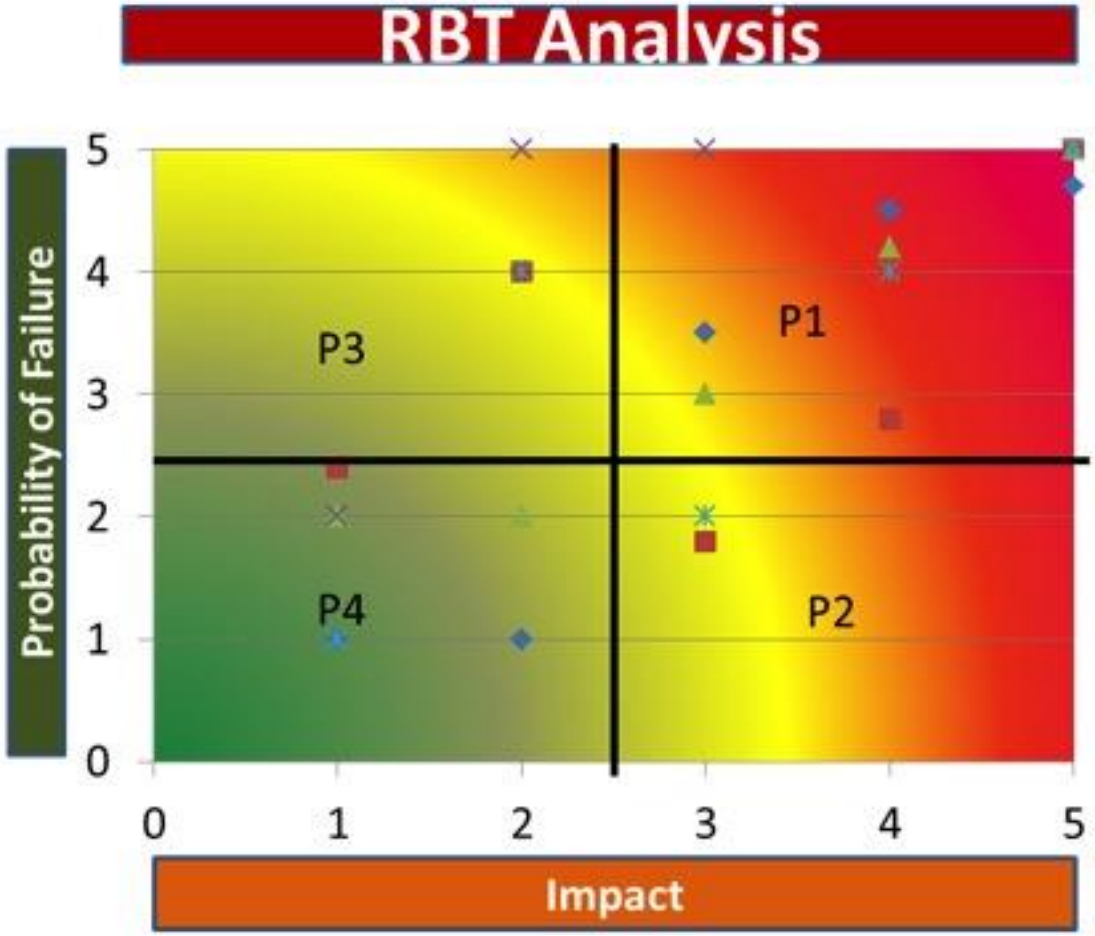
- Level of risk = impact * probability

- P1 – Critical Impact: Must be tested; ideal candidate for automation (candidate for automated smoke testing)

- P2 – High Impact: Should be tested

- P3 – Medium Impact: Can be tested if budget and schedule permits

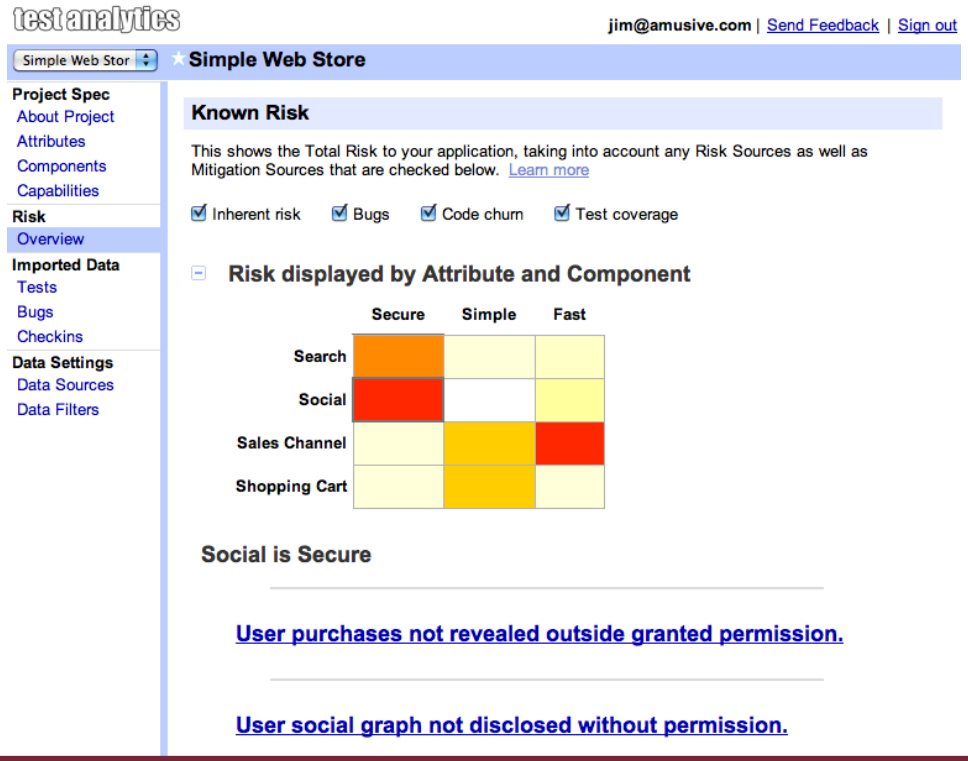
- P4 – Low Impact: May not be tested; no impact on application and no need of automation



Source: Risk-Based Testing: Test Only What Matters

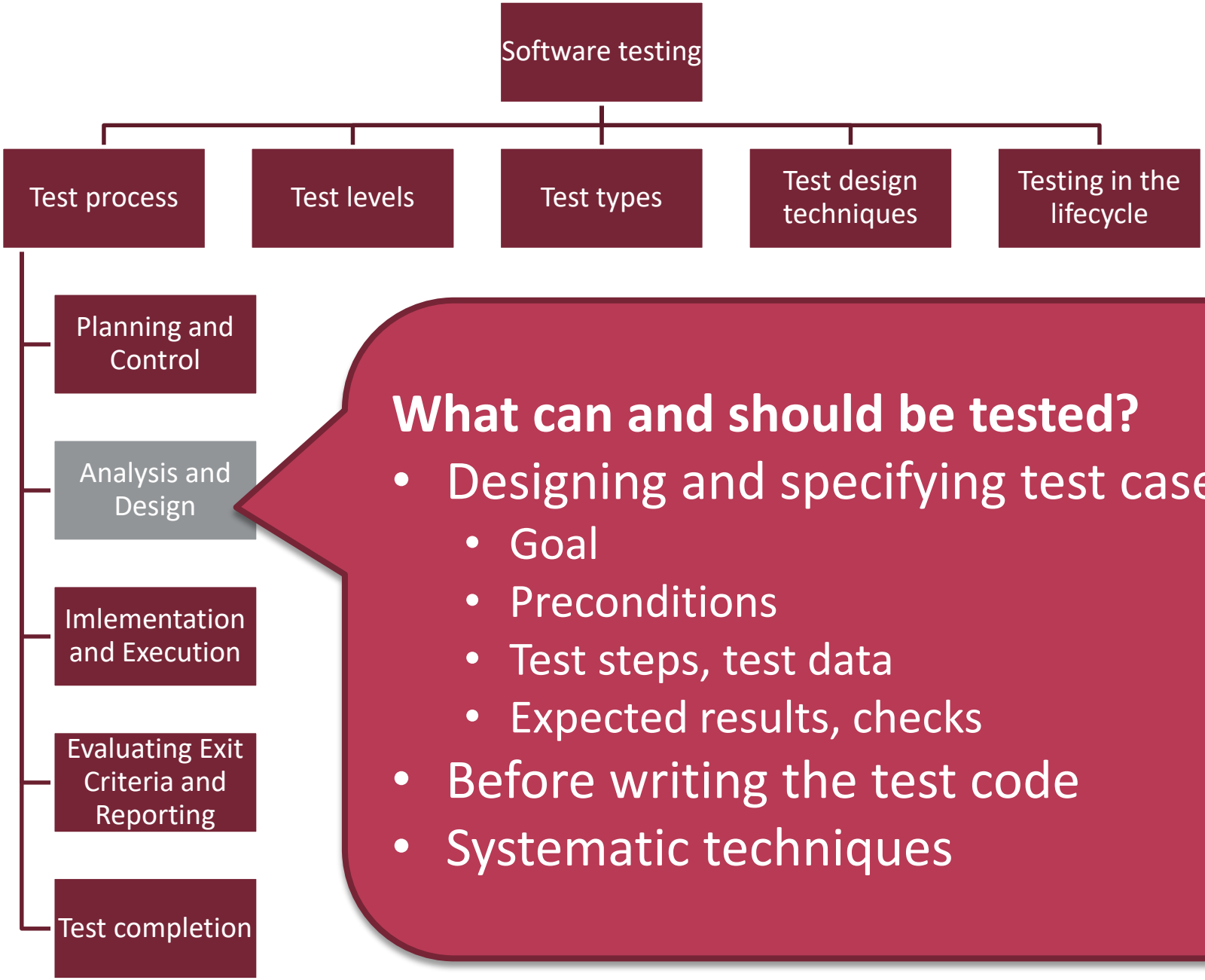
Google “10 minute test plan”

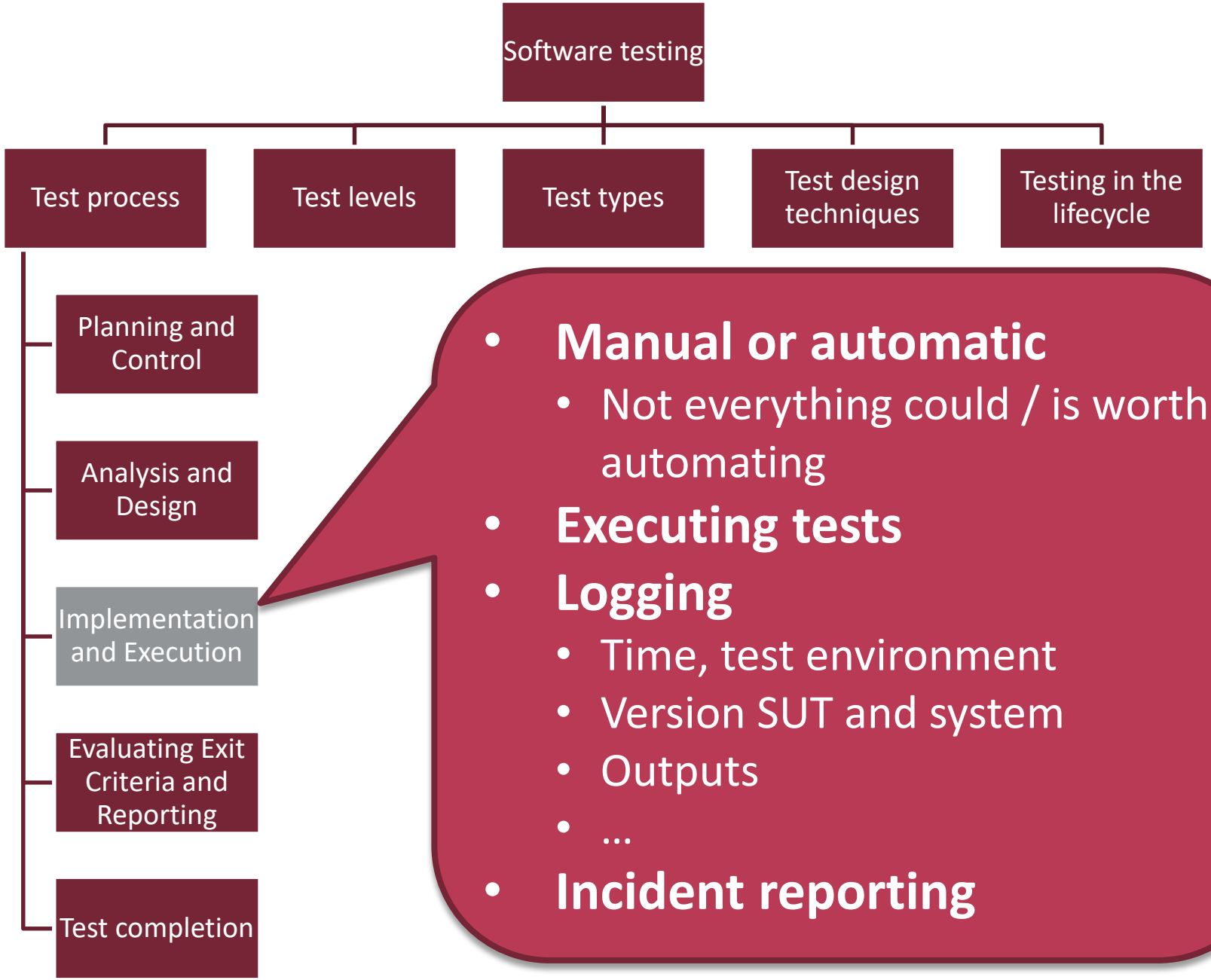
- Why do write a plan that is not used and updated?
- Keep only the most important
  - Attributes, Components, Capabilities (ACC)

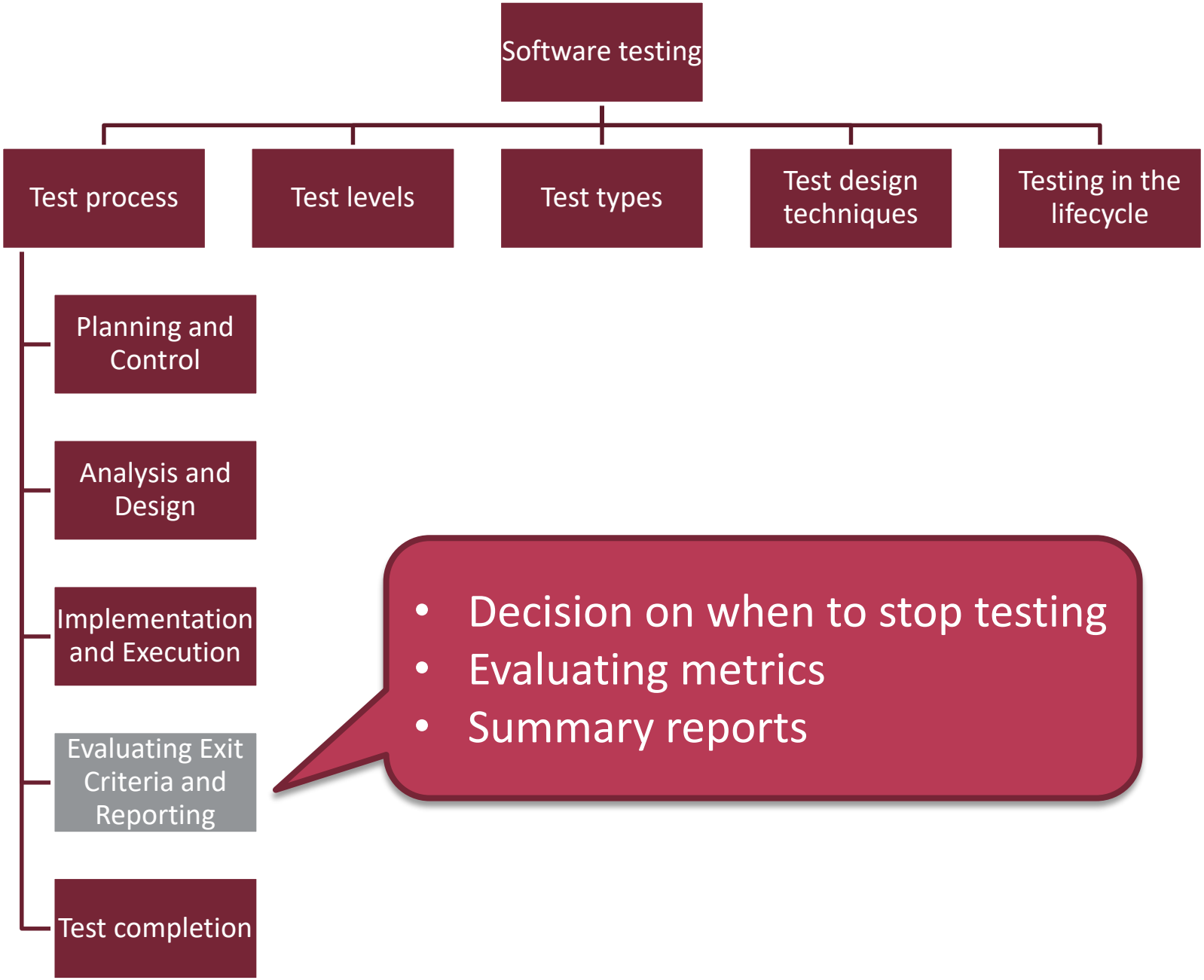


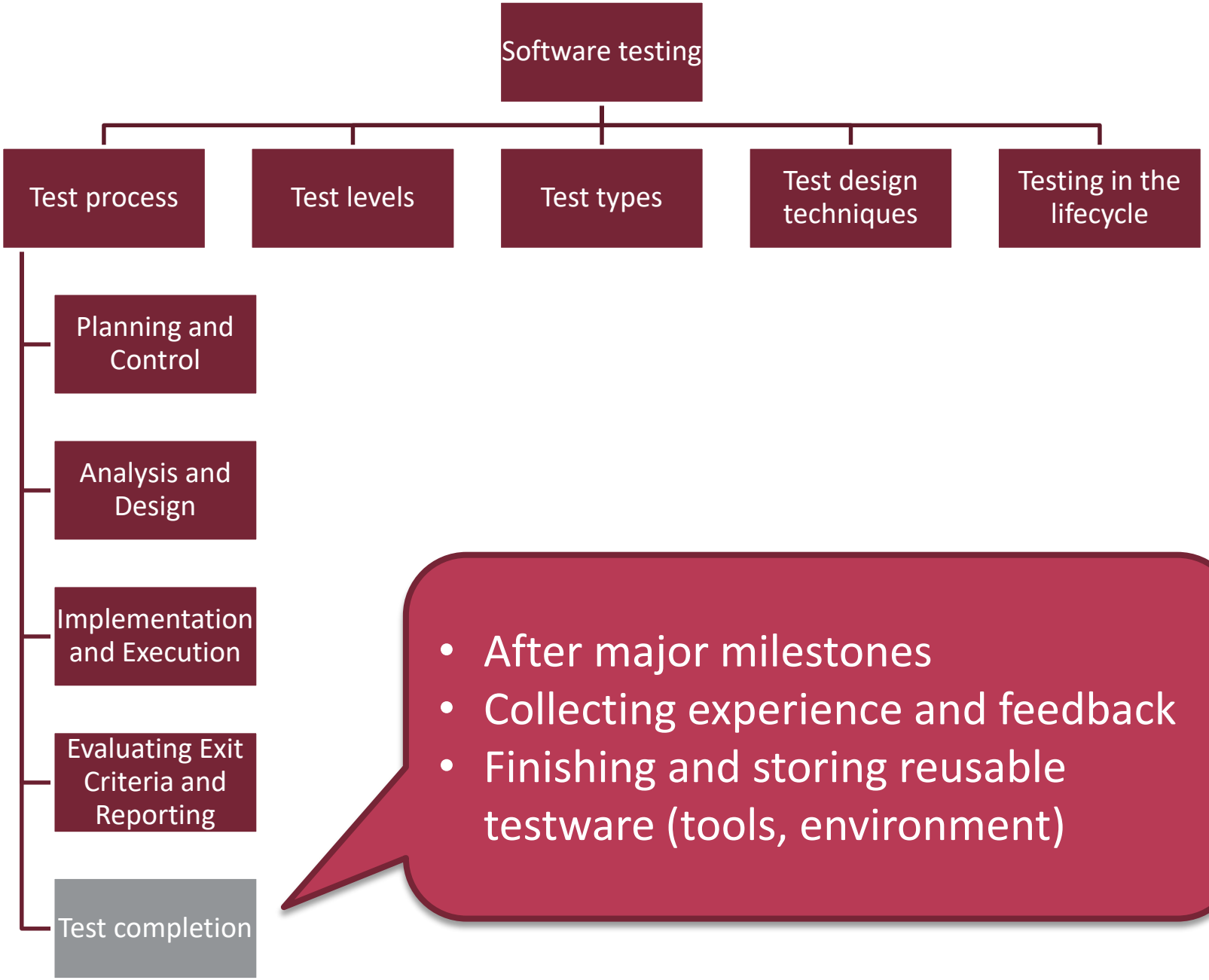
Source:
Google Test Analytics - Now in Open Source

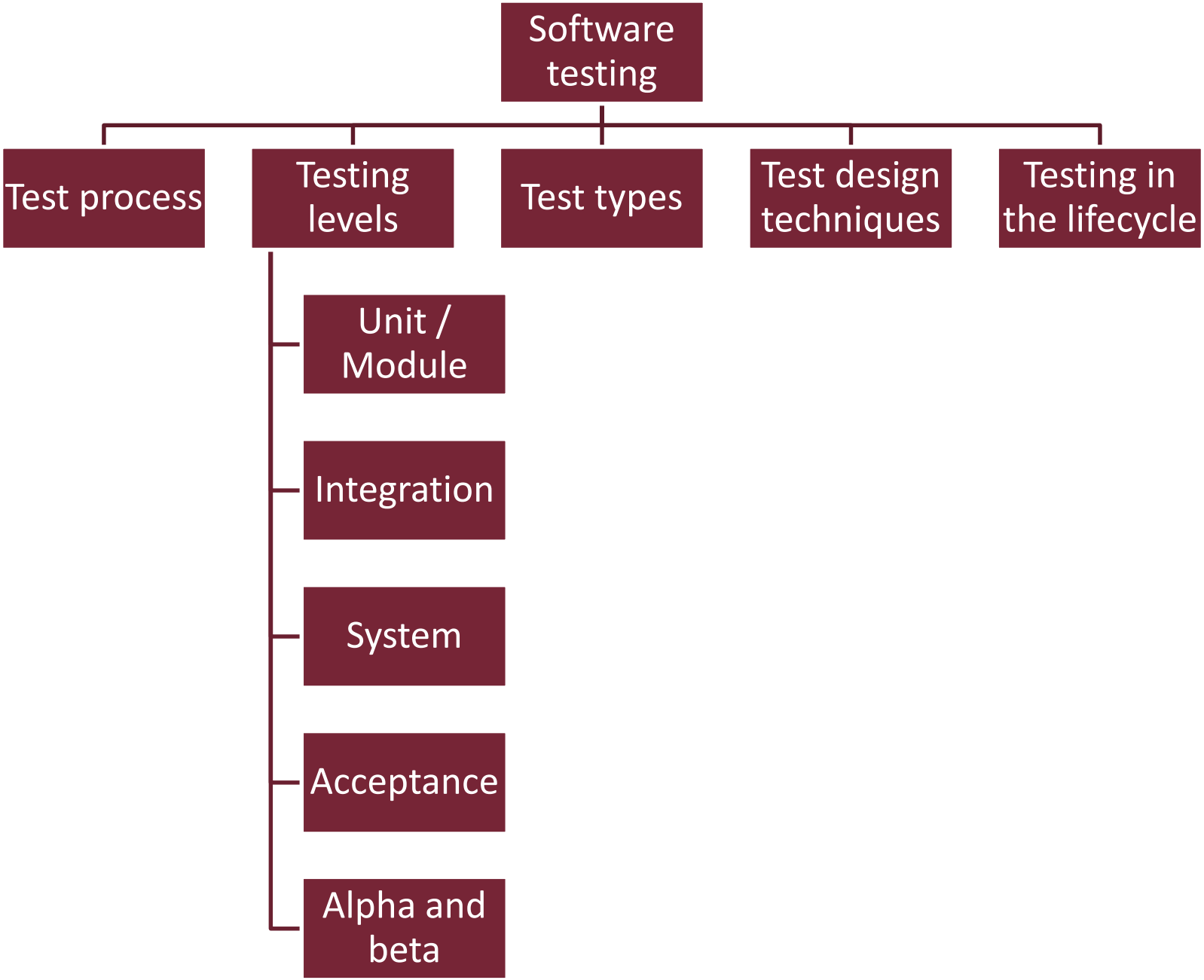
Further questions for test planning: Inquiry Method


Software testing
Test process
Test levels
Test types
Test design techniques
Testing in the lifecycle
Planning and Control
Analysis and Design
Imlementation and Execution
Evaluating Exit Criteria and Reporting
Test completion
What can and should be tested?
• Designing and specifying test cases
• Goal
• Preconditions
• Test steps, test data
• Expected results, checks
• Before writing the test code
• Systematic techniques


Software testing
Test process
Test levels
Test types
Test design techniques
Testing in the lifecycle
Planning and Control
Analysis and Design
Implementation and Execution
Evaluating Exit Criteria and Reporting
Test completion
• Manual or automatic
• Not everything could / is worth automating
• Executing tests
• Logging
• Time, test environment
• Version SUT and system
• Outputs
• …
• Incident reporting


Software testing
Test process
Test levels
Test types
Test design techniques
Testing in the lifecycle
Planning and Control
Analysis and Design
Implementation and Execution
Evaluating Exit Criteria and Reporting
Test completion
• Decision on when to stop testing
• Evaluating metrics
• Summary reports


Software testing
Test process
Test levels
Test types
Test design techniques
Testing in the lifecycle
Planning and Control
Analysis and Design
Implementation and Execution
Evaluating Exit Criteria and Reporting
Test completion
• After major milestones
• Collecting experience and feedback
• Finishing and storing reusable testware (tools, environment)

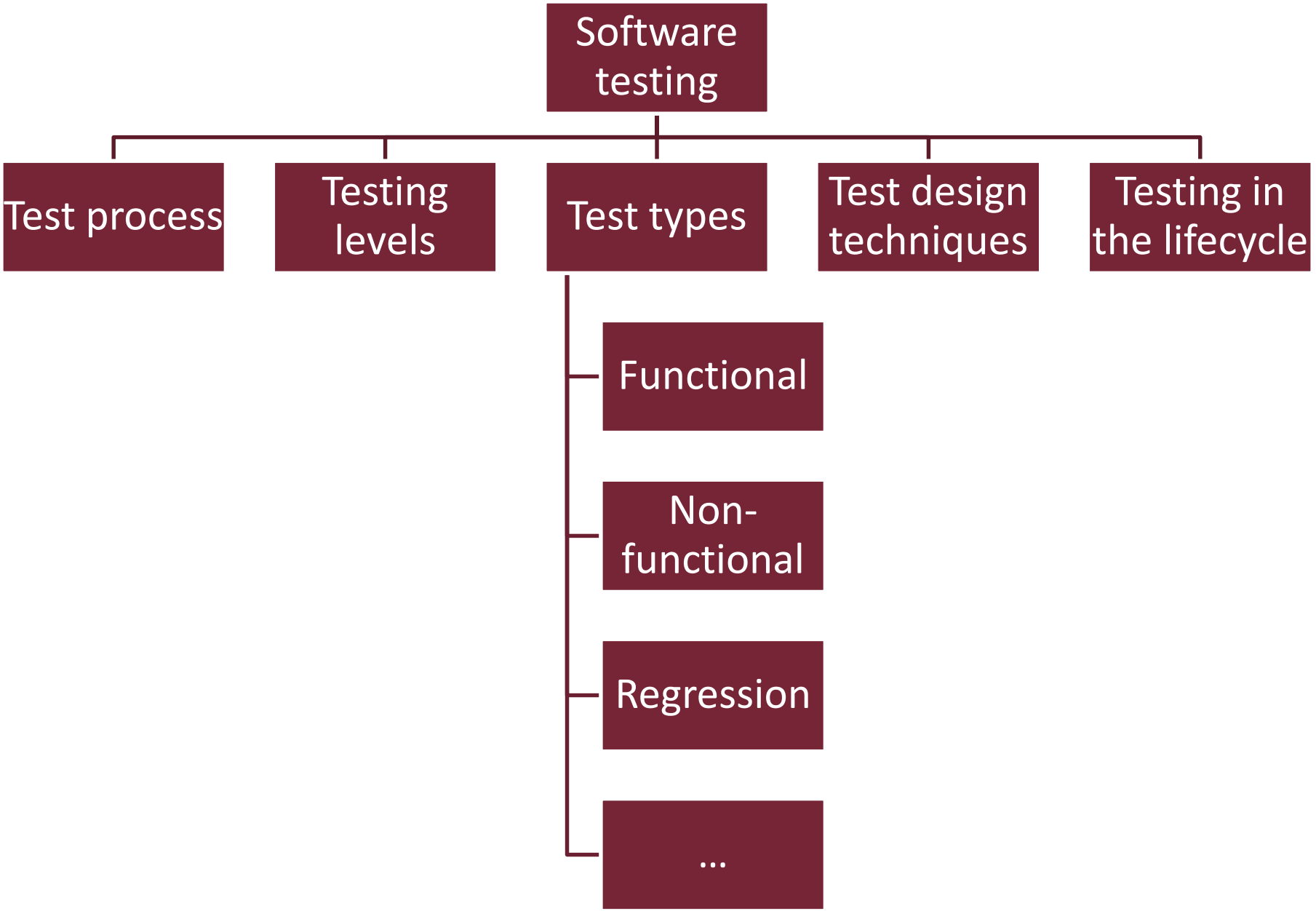

Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
Unit / Module
Integration
System
Acceptance
Alpha and beta

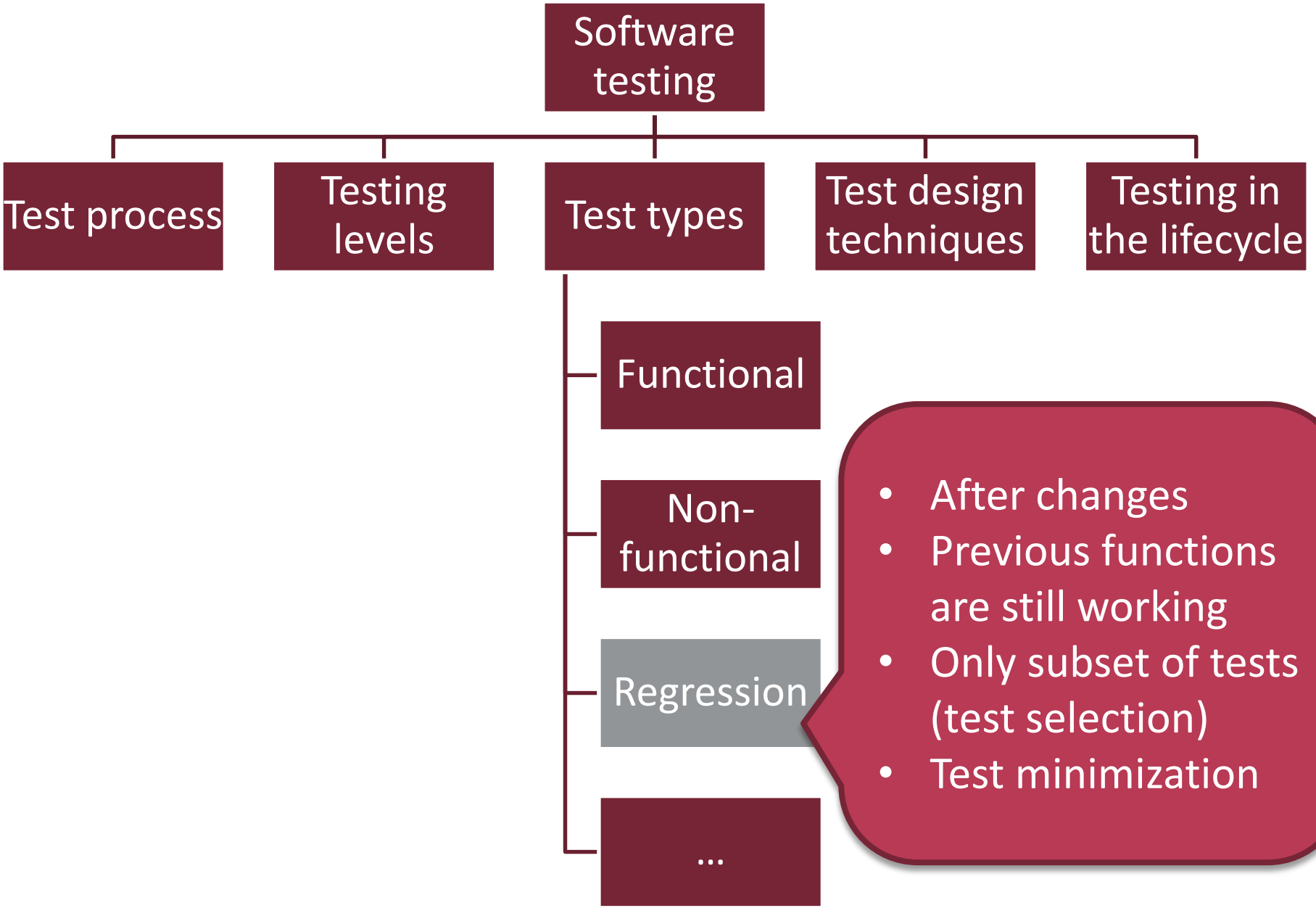
Characteristics of tests in different levels

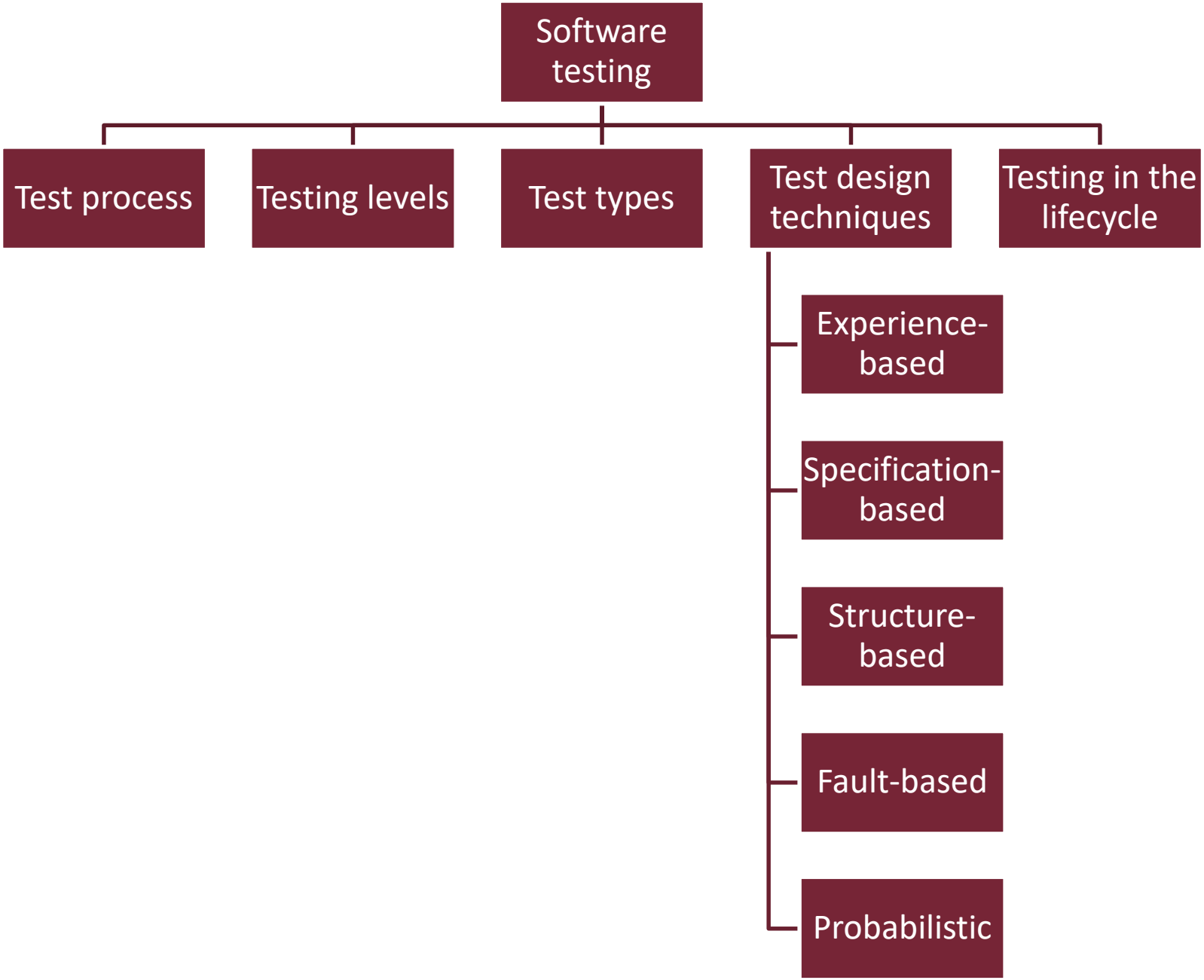
Recommendations from *How Google Tests Software*:

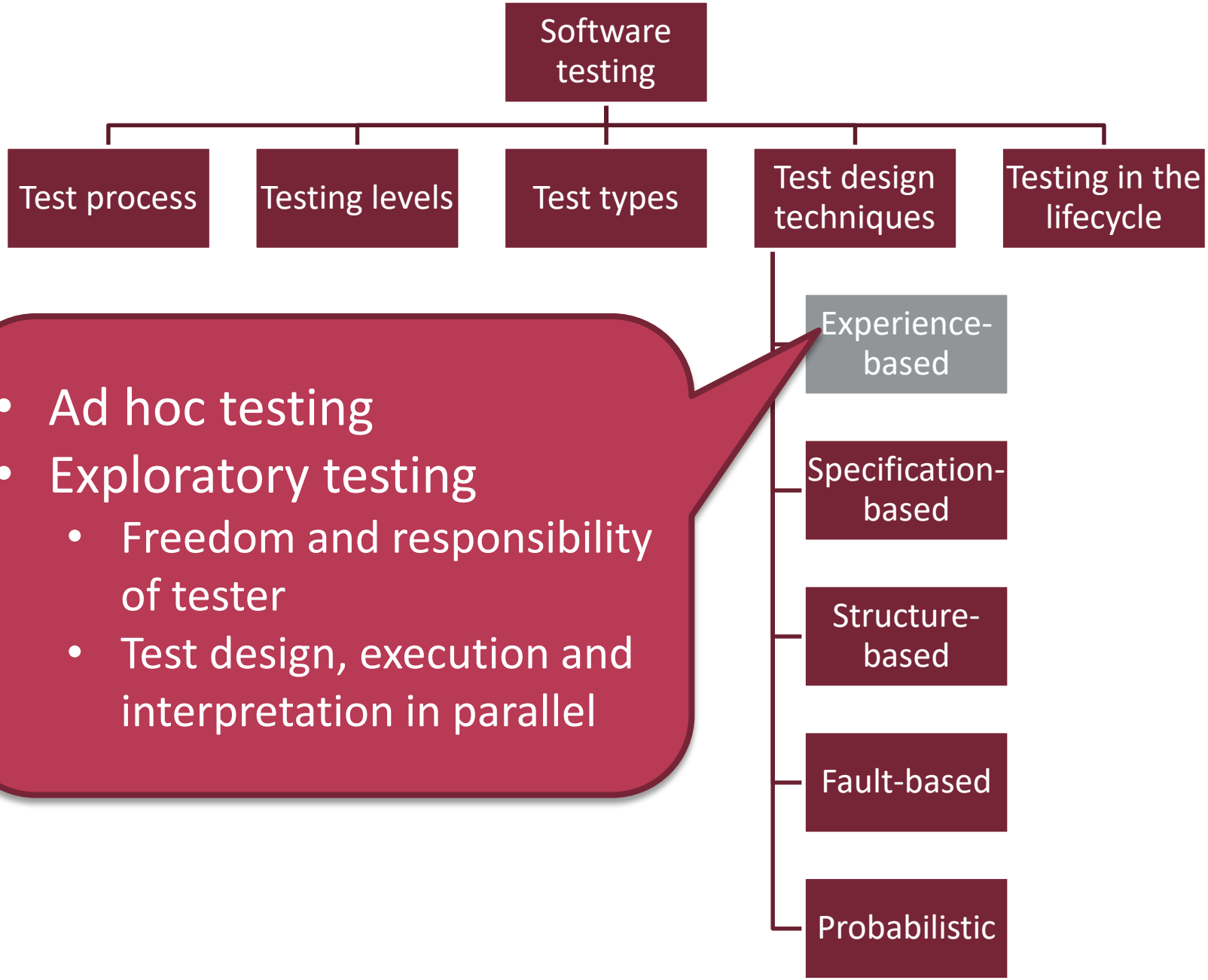
	Small	Medium	Large
Execution time	< 100 ms	< 1 sec	As fast as poss.
Time limit (kill)	1 minute	5 minutes	1 hour

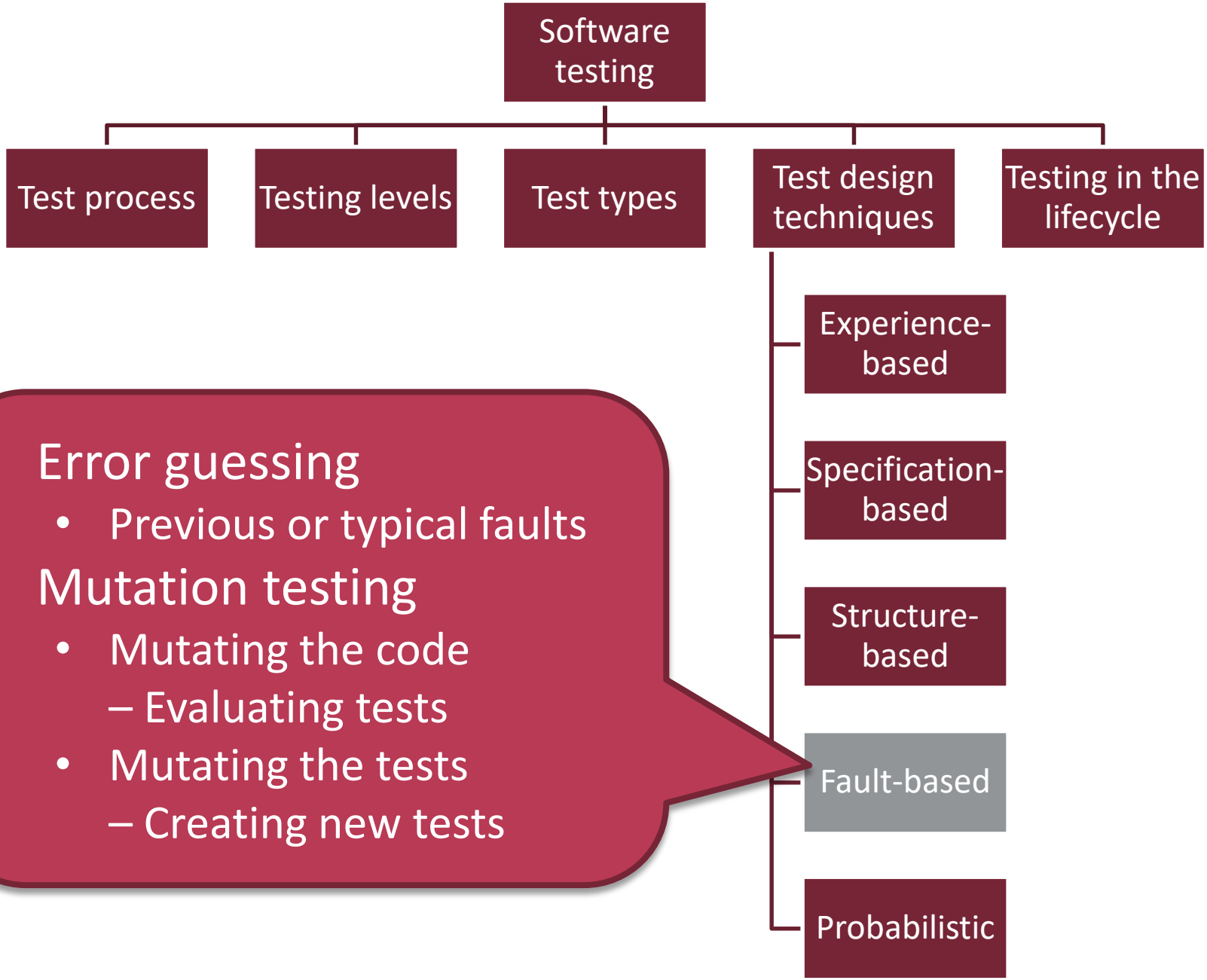
Resource	Small	Medium	Large
Network (socket)	Mocked	only localhost	Yes
Database	Mocked	Yes	Yes
File access	Mocked	Yes	Yes
System call	No	Not recommended	Yes
Multiple threads	Not recommended	Yes	Yes
Sleep	No	Yes	Yes
System properties	No	Yes	Yes

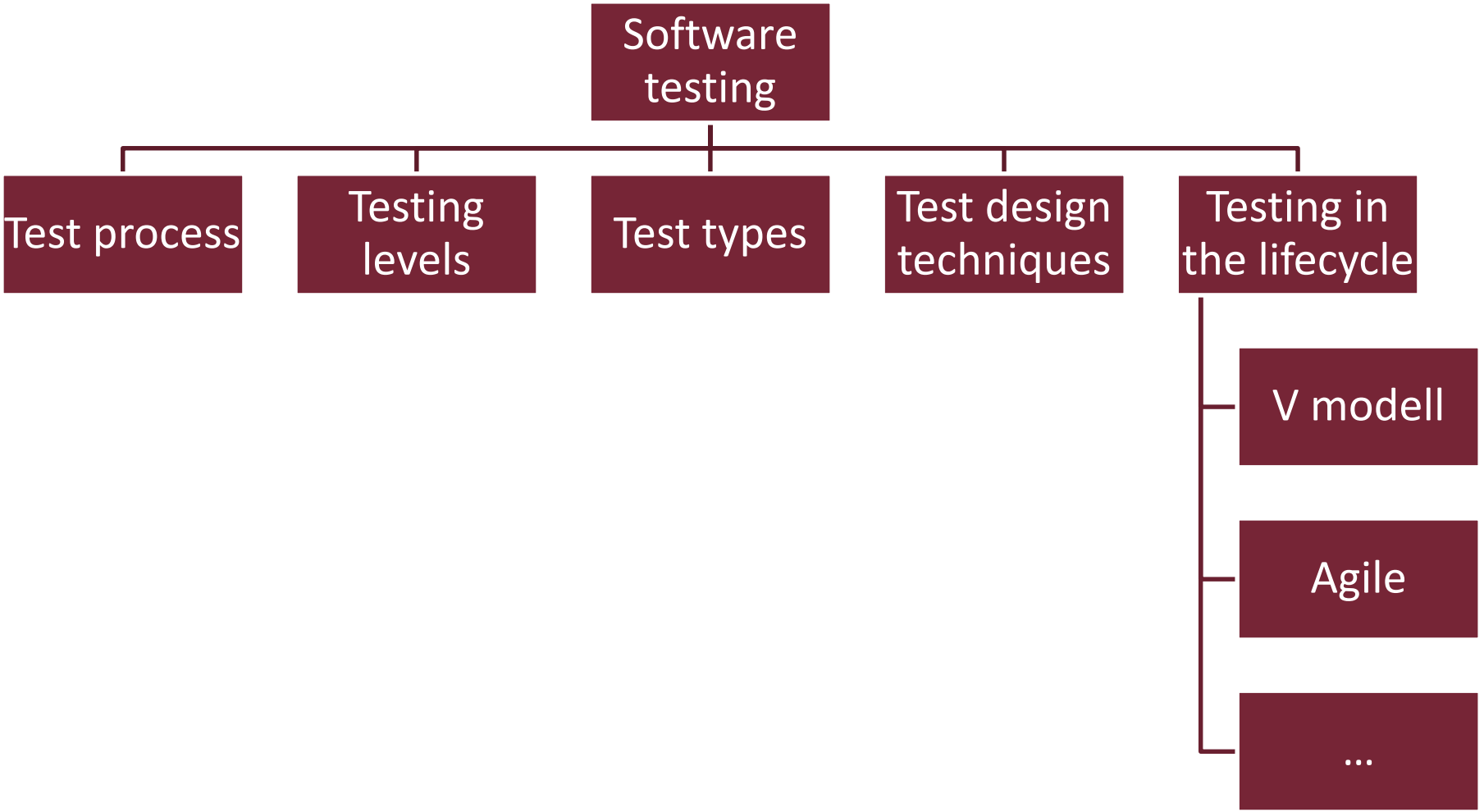

Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
Functional
Non-functional
Regression
...

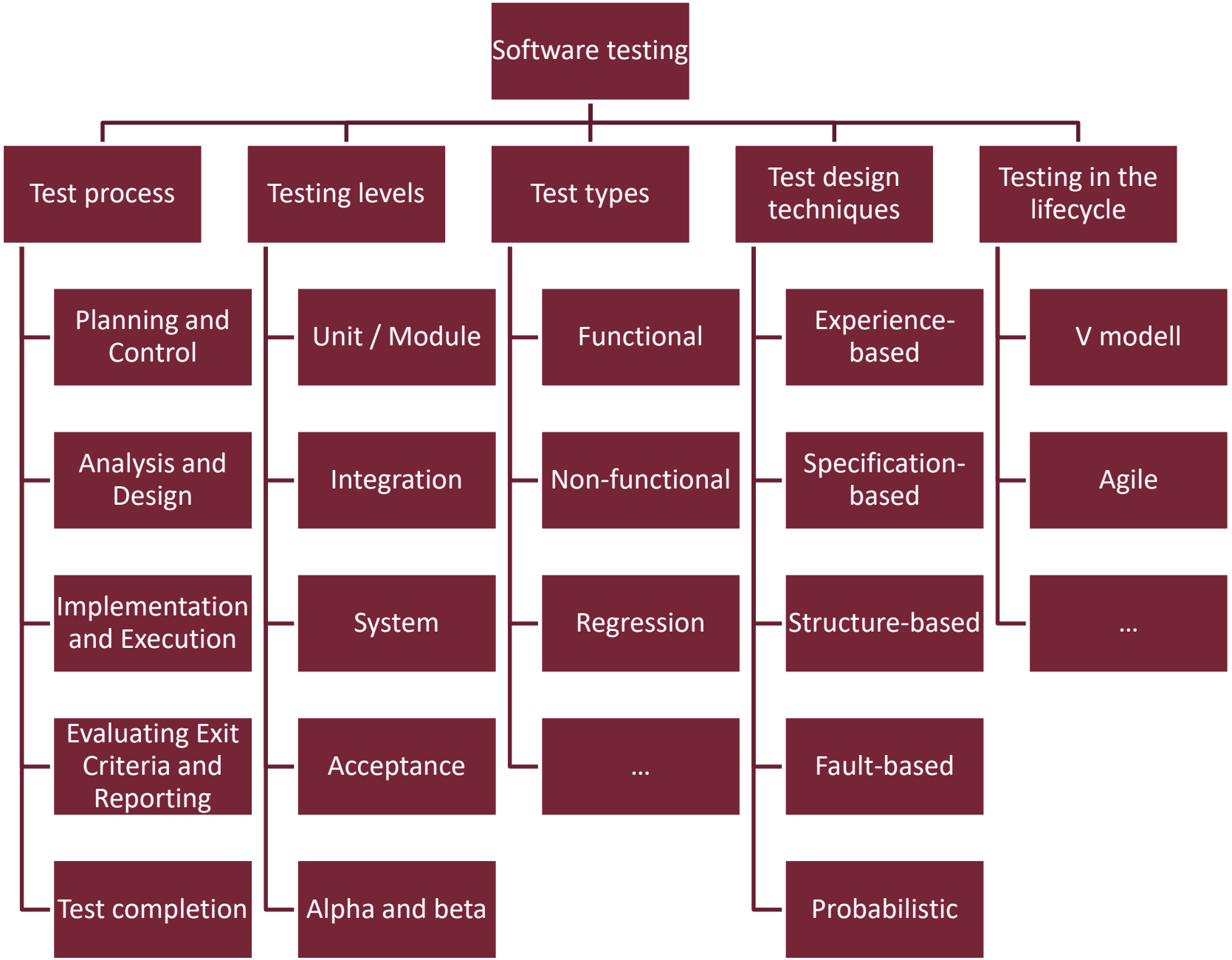

Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
Functional
Non-functional
Regression
...
After changes
Previous functions are still working
Only subset of tests (test selection)
Test minimization


Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
Experience-based
Specification-based
Structure-based
Fault-based
Probabilistic


Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
Experience-based
Specification-based
Structure-based
Fault-based
Probabilistic
• Ad hoc testing
• Exploratory testing
• Freedom and responsibility of tester
• Test design, execution and interpretation in parallel


Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
Experience-based
Specification-based
Structure-based
Fault-based
Probabilistic
• Error guessing
• Previous or typical faults
• Mutation testing
• Mutating the code
– Evaluating tests
• Mutating the tests
– Creating new tests


Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
V modell
Agile
...


Software testing
Test process
Testing levels
Test types
Test design techniques
Testing in the lifecycle
Planning and Control
Analysis and Design
Implementation and Execution
Evaluating Exit Criteria and Reporting
Test completion
Unit / Module
Integration
System
Acceptance
Alpha and beta
Functional
Non-functional
Regression
...
Experience-based
Specification-based
Structure-based
Fault-based
Probabilistic
V modell
Agile
...