Software Testing

Test Prioritization Strategies

by   J. Janvier (Jay/文宇)

Test Prioritization Strategies

Test Prioritization

In most projects budget and/or time are a limiting factor,
so to get the best results Test Prioritization should be applied.

Example situation:

Software Component A	Software Component B

Test run 1:

100 Tests
Failed: 6 Tests

100 Tests
Failed: 37 Tests

You reported the bugs in bug tracking software, and solved.

Test run 2:

100 Tests
Failed: 1 Tests

100 Tests
Failed: 43 Tests

What will you do?

Test Prioritization

Strategies:

- Customer Requirements-Based Prioritization
- Coverage-Based Prioritization
- Cost Effective-Based Prioritization
- History-Based Prioritization
- Risk-Based Prioritization

In most projects several strategies are combined to reach the optimal balance between quality and available resources(budget/time)

Test Prioritization

Customer Requirements-Based Prioritization Strategy

1. Tests are ranked on basis of several factors/formulas:
   - Customer-Assigned priority (CP)
     The customer indicates the importance of each requirement
   - Implementation/Requirements Complexity (RC)
     How complex will the implementation of the requirement be?
   - Requirements Volatility (RV)
     How often is the requirement changed, and therefore must be tested again?
2. Tests are then executed in the ranked order.

Example:

Test Prioritization

Coverage-Based Prioritization Strategy

Tests are prioritized on basis of Coverage Criteria:

Types of Coverages:
- Requirements Coverage,
  - Initial Requirements Coverage
  - Total Requirements Coverage (requirements from all levels)
  - Additional Requirements Coverage (added/changed req.)
- Statement Coverage (testing the paths by calling only main methods)
- Code Coverage,

Based on the budget/time a percentage is set for the selected type(s) of coverage, and testing continues until this percentage is reached.

Example:

Test Prioritization

Cost Effective-Based Prioritization Strategy

Tests are prioritized on basis of costs, like how much it would cost:
- to automate the test?,
- to analyse the results?,
- to setup the test environment (need equipment?),
- to execute the test?, etc.

This strategy is often applied in combination with other strategies.

Sometimes the budget for software testing is divided over the project phases, and therefore Test Case prioritization might be needed for each phase.

Test Prioritization

History-Based Prioritization Strategy

Tests are prioritized on basis of Test execution history.

Example situation:

Software Component A	Software Component B

Test run 1:

100 Test Cases
Failed: 6 Test Cases

100 Test Cases
Failed: 37 Test Cases

You reported the bugs in bug tracking software, and solved.

Test run 2:

100 Test Cases
Failed: 1 Test Cases

100 Test Cases
Failed: 43 Test Cases

What will you do?

Based on metrics you will shift focus to the complex/unstable component;
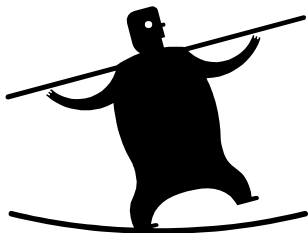And give these tests a higher priority.

Test Prioritization

Risk-Based Prioritization Strategy

Tests are prioritized on basis of:
- What the damage of failure of the requirement will be (impact), and
- The probability that the failure of the requirement/function will occur.

Example Risk Matrix:

Impact		Probability of Failure		
		Low	Medium	High
	High	Req-2 Req-5	Req-1	Req-6
	Medium	Req-8 Req-3	Req-7 Req-11	Req-10
	Low	Req-4	Req-12 Req-14 Req-13	Req-9

Questions?