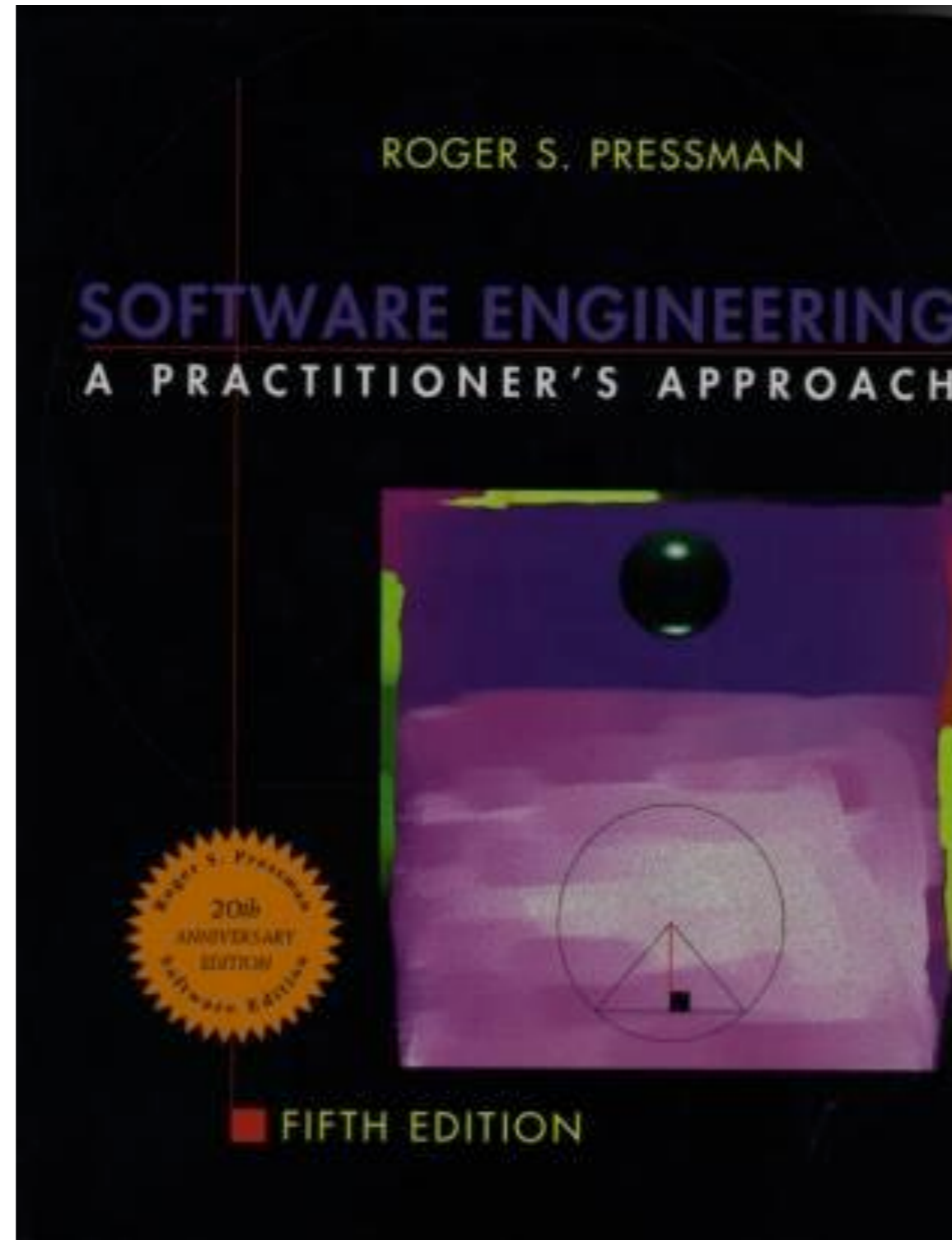


Software Testing

2013

Book



Outline

- Software testing fundamental
- Test case design
- WhiteBox testing
- Basis path testing
- Control structure testing
- BlackBox testing

Testing fundamental

- To ensure that software is right
- Finding errors
- Good test case = high chance to find uncovered errors
- A successful test = found uncovered error

Testing principle

- Tests based on customer requirements
- Planned long enough
- Progress from small to large
- Exhaustive testing is not possible
- Better to hire independent third-party

Testability

- means how easily software can be tested
- Should keep it in mind
- There are some guide line to achieve the software testability

Operability

- Few bugs
- No bugs blocked the execution of tests
- In development stages

Observability

- What you see is what you test
- System states & variable are observable
- All factors affecting the output
- incorrect output is easily identified
- Automatically detect & report internal errors
- Source code is accessible

Controllability

- The better control the better automated & optimized test
- All possible output can be generated through some combination of input
- All code is executed through
- Direct control hardware & software

Decomposition

- Control scope of testing
- Independent modules
- quickly isolate problems

Simplicity

- Quickly to test
- Functional simplicity
- Structure simplicity
- Code simplicity

Stability

- The fewer changes, the fewer disruption
- Changes are infrequent
- Changes are controlled
- Software recovery

Understandability

- More information, smarter test
- Design & components well understood
- Changes to design are communicated
- Technical document is
 - instantly accessible
 - well organized
 - specific, detailed, and accurate

Good test

- High prob. of finding an error
- Not redundant
- Neither too simple nor too complex