

# INDEX

## A

- Acceptance testing, 128, 144
  - with Extreme Programming, 179, 185
- Application programming interface (API), 177
- Application server, 196
- Architecture testing, 204
- Automated debugging tools, 158, 160
- Automated testing
- Automated test tools, 120

## B

- Backtrack debugging, 168
- Basic e-commerce architecture, 194
- Beck, Kent, 188
- Big-bang testing, 105
- Black-box testing, 9–11, 44, 205. *See also* Equivalence partitioning
  - methodologies of, 52
  - vs. white-box testing, 114
- BONUS module, 94, 101
  - boundary value analysis of, 102
- Boolean logic network, 85

- Bottom-up testing, 116–119
  - disadvantages, 117
  - vs. top-down testing, 109
- Boundary conditions, 59
- Boundary-value analysis, 44, 59, 196
  - compared to equivalence class testing, 59
  - input boundaries for, 102
  - program example, 60, 61
  - weakness of, 65
- Branch coverage, 45
- Branch decision testing, exceptions to successful testing, 46
- Branching statement, Java code sample, 47, 48
- Browser compatibility testing, 198, 204
- Brute force debugging, 158
  - problems with, 160
- Business layer, 196, 201
  - business layer testing, 199, 205–208
- Business tier, 196

## C

- C++ compiler testing, 10
- Case stories, 179

Cause-effect graph:  
   converting to decision table, 81  
   definition, 66  
   symbols, 68  
   used to derive test cases, 71

Cause-effect graphing, 44,  
   65–88, 149  
   described, 85  
   for program specification  
     analysis, 86

Check4Prime.java, 186, 213

Check4PrimeTest.java, 186, 216

Checklists, 27–38  
   comparison errors, 31  
   computation errors, 30  
   control flow errors, 32  
   data-declaration errors, 29  
   data reference errors, 27  
   input/output errors, 35  
   inspection error, 36  
   interface errors, 34

Client-server applications, 193

Client-server architecture,  
   194–196, 201

Code inspection, 22, 24–26  
   benefits, 26  
   checklist for, 27  
   process, 26  
   time required, 26

Code walkthrough, 38

Comparison errors, 31

Compatibility/configuration/  
   conversion testing, 138

Compiler testing, 10

Computation errors, 30

Condition coverage, 44, 46, 97

Configuration testing, 138

Content testing, 204

Continuous testing, 179

Control-flow errors, 32

Control-flow graph, 11

## D

Data access layer, 196, 201  
   data access layer testing, 199,  
   208–212

Data checking, 23

Data-declaration errors, 29

Data-driven testing, 9

Data integrity, 210

Data reference errors, 27

Data-sensitivity errors, 13

Data tier, 196

Data validation, 207

Debugging, 157–175  
   by backtracking, 168–169  
   brute force method, 158–160  
   by changing the program, 159  
   cost of, 23  
   by deduction, 164–168  
   error analysis and, 173  
   by induction, 160–164  
   principles of, 170–173  
   procedures for, 147  
   by testing, 169

Decision/condition-coverage  
   testing, 49, 99  
   and/or masking, 49

Decision coverage, 44, 45

Decision coverage testing, defi-  
   nition, 47

- Decision table, 79
  - from cause-effect graph, 81
- Deductive debugging, 164
  - test cases, 166
- Desk checking, 40
- Development vs. testing, 127
- Documentation testing, 142
- Driver module, 106
- E**
- E-commerce architecture, 194–196
- Economics of program testing, 9
- End-user test, 144
- Equivalence classes:
  - compared to boundary-value analysis, 59
  - identifying, 53, 54
  - program example, 56
  - splitting, 55
- Equivalence partitioning, 44, 52
  - steps to complete, 53
  - weakness of, 65
- Eratosthenes, sieve of, 190
- Erroneous input checks, 55
- Error analysis debugging, 173
- Error guessing, 44, 88
  - example, 89
- Error-locating debugging
  - principles, 170
- Error-repairing debugging
  - principles, 171
- Errors:
  - checklists for, 27–35
  - comparison, 31
  - computation, 30
  - control-flow, 32
  - data-declaration, 29
  - data reference, 27
  - input/output, 35
  - interface, 34
- Errors found/errors remaining
  - relationship, 20
- Exhaustive input testing, 9, 10, 13
- Exhaustive path testing, 11, 13
- External specification, 124, 125
- Extreme Programming (XP), 177
  - acceptance testing with, 178–183, 185
  - basics of, 178
  - case stories with, 179
  - continuous testing with, 179
- Extreme Testing with, 183
  - practices, 179–181
  - project example, 182
  - refactoring with, 180
  - strengths and weaknesses, 182
  - unit testing with, 179, 183–185
- Extreme Testing (XT), 177, 178, 183–190
  - applied, 186
  - concepts of, 183
  - program sample, 213
  - test case design, 186
  - test driver for, 189
  - test harness with, 190
- Extreme unit testing, 183

**F**

- Facility testing, 133
- Fault tolerance and recoverability, 211
- Flow-control graph, 12
- FORTTRAN DISPLAY command:
  - objectives for, 132
  - test cases for, 133
- Function test, purpose of, 128, 130
- Function testing, 129–130
  - completion criteria for, 149, 151

**G**

- Gamma, Erich, 188
- Graphical user interface (GUI), 1
- Guidelines for program testing, 15

**H**

- Higher-order testing, 123–155
  - defined, 123
- Human factors, 135
- Human testing, 21
  - procedures of, 142
  - See also* Code inspection, Desk checking, Peer ratings, Walkthrough

**I**

- Identifying test cases, 55
- Incremental testing: 105–109
  - compared to nonincremental testing, 107
  - See also* Bottom-up testing, Top-down testing

- Independent test agency, 155
- Induction debugging, 160
  - example, 163
  - steps to follow, 161
- Inductive assertions, 140
- Input checks, erroneous, 55
- Input/output-driven testing, 9
- Input/output errors, 35
- Inspection error checklist, 36
- Inspections, 21
  - checklist for, 27–40
  - and walkthroughs, 22
- Inspection team, 24
  - duties, 25
  - members, 25
- Installability testing, 139
- Installation testing, 128, 144
  - test cases for, 145
- Integration testing, 105
- Interface errors, 34
- Internet applications:
  - architecture testing with, 204
  - Business layer testing, 205–208
  - challenges, 197
  - client-server architecture, 194–196
  - content testing with, 204
  - data integrity with, 210
  - Data layer testing with, 208
  - data validation of, 207
  - fault tolerance with, 211
  - performance goals for, 198
  - performance testing of, 206
  - Presentation layer testing, 202
  - recoverability with, 211
  - response time testing with, 209

- stress testing with, 207
- testing strategies for, 200
- transactional testing of, 208

Internet application testing,  
193–212

- challenges with, 196

Isosceles triangle, 2

## J

Java code sample, branching  
statement, 47, 48

Java program sample, 186, 213,  
216

JUnit, 188

## L

Logic coverage testing, 44. *See*  
also White-box testing

Logic-driven testing, 11

Loop testing, 33

## M

Mean time between failures  
(MTBF), 140, 200

- maximizing, 211

Mean time to recovery  
(MTTR), 141, 142, 200

Memory dump, 159

Missing path errors, 13

Module testing, 91–121, 151

- with automated tools, 120
- completion criteria for, 148
- performing the test, 120
- purpose of, 128,
- test case design, 92

MTBF. *See* Mean time between  
failures (MTBF)

MTEST, 61, 63

- error guessing with, 89
- output boundaries for, 64
- test cases for, 63

MTTR. *See* Mean time to  
recovery

Multicondition coverage crite-  
rion, 101

Multiple-condition coverage,  
44

Multiple condition testing,  
49–50

## N

Network connectivity, 200

Non-computer-based testing,  
21

Nonincremental testing, 105,  
106

## O

Off-by-one errors, 33

## P

Path testing, 45

Peer ratings, 40

Performance testing, 137, 206

PL/1, 92, 101

Presentation layer, 196, 201

- presentation layer testing, 199,  
202–205

Presentation tier, 196

Preventing software errors,  
125

Prime numbers:  
calculation, 190  
list, 221

Print statement debugging, 159

Procedure testing, 142

Program:

- control-flow graph, 11, 12
- debugging, 16
- inspections, 21
- reviews, 21
- walkthroughs, 21

Program errors:

- checklist for inspections, 27
- comparison, 31
- computation, 30
- control-flow, 32
- data-declaration, 29
- data-sensitivity, 13
- determining number of, 149
- input/output, 35
- interface, 34
- missing path, 13
- probability, 19

Program testing:

- as destructive process, 8
- assumption, 6
- black-box, 9
- definition, 5, 6
- economics of, 9
- example, 56
- guidelines, 14, 15
- human factor considerations, 135
- principles of, 14
- strategies, 9
- success, 7
- white-box, 11
- who should test, 16

## R

Random-input testing, 43

Rapid application development, 177

RDBMS. *See* Relational database management system (RDBMS)

Recovery testing, 141

Refactoring, 180

Regression testing, 18, 147

Relational database, 196

Relational database management system (RDBMS), 196

Reliability testing, 139

Response-time testing, 209

## S

Scalene triangle, 2

Security testing, 137

Serviceability testing, 142

Sieve of Eratosthenes, 190

Software development, vs. testing, 127

Software development cycle, 123, 124

Software documentation, 125

Software errors:

- causes, 124, 125
- preventing, 125

Software objectives, external specification, 124

Software prediction, 140

Software proving, 140

Software reliability engineering (SRE), 140

Software requirements, 125

- Software testing:  
 vs. development, 127  
 human factor consideration,  
 135  
 principles of, 14–20  
 summarized, 15
- Software test plan, 146
- SRE. *See* Software reliability engineering (SRE)
- Statement coverage, 44
- Storage dump debugging, 158
- Storage testing, 138
- Stress testing, 134, 207
- Stub module, 106, 110
- System testing, 130–144, 151  
 fifteen categories of, 133–142  
 performing, 143  
 purpose of, 128  
 stress testing, 134, 207
- T**
- Test-based debugging, 169
- Test case design, 43–90. *See also*  
 Black-box testing, White-  
 box testing  
 with Extreme Testing,  
 186–189  
 information required, 92  
 module testing, 92  
 multiple-condition criteria, 51  
 properties of, 52  
 strategy, 90  
 subset definition, 52
- Test case program, 2
- Test cases, 17  
 deductive debugging, 166  
 identifying, 55  
 steps to derive, 66  
 user documentation for, 131
- Test completion criteria, 148–155
- Test driver and application, 189
- Test harness, 190
- Testing. *See also specific test types*  
 of browser compatibility, 198  
 vs. development, 127  
 of Internet applications, 193  
 of large programs, 91  
 psychology of, 5  
 successful vs. unsuccessful, 7
- Testing strategies, for Internet applications, 200
- Test management, 146
- Test plan, components of,  
 146–147
- Test planning and control, 145
- Three-tier architecture, 196. *See also* Business layer, Data access layer, Presentation layer
- Top-down testing, 110–116  
 advantages and disadvantages,  
 114–115  
 vs. bottom-up testing, 109  
 guidelines, 113
- Tracking procedures, 147
- Transactional testing, 208
- Twelve practices of Extreme Programming, 180–181
- U**
- Unit testing, 91, 183–187  
 test case design, 92

Unit testing (*Continued*)  
  with JUnit, 188  
  with Extreme Programming,  
    179, 183  
Usability testing, 135  
User documentation, forming  
  test cases with, 131  
User environment testing, 205  
User stories, 179

## V

Volume testing, 133

## W

Walkthroughs, 22–24, 38  
  team members, 39  
Web-based applications  
  configuration testing with, 138

  security testing with, 137  
  stress testing with, 134  
Web browsers:  
  compatibility among,  
    198  
  content delivery, 195  
Web server, 196  
White-box testing, 11–13, 44  
  logic coverage testing, 44  
  module test, 92, 204  
  module testing, 92  
  vs. black-box testing,  
    114

## X

XP. *See* Extreme Programming  
  (XP)  
XT. *See* Extreme Testing (XT)







