
Index

A

Abstract data type, 79
ACME, 74
Adaptive
 maintenance category, 5

B

Bag of tasks, 283
Behavior-driven development, 258
Bonferroni's principle, 301

C

Cause/effect diagram, 61
Code of ethics, 78
Cohesion, 218
Composition over inheritance, 142
Consumer-driven contracts, 83
Context, 10
Correction
 maintenance category, 5
Corrective
 maintenance category, 5
Correlation vs. causation, 301
Cross-origin resource sharing, 166
Cyclomatic complexity, 19, 33

D

Davis's principles, 39
Defensive programming, 49, 91
De Morgan's law, 55
Dependency injection, 61
Deviation principle, 26
Don't repeat yourself, 120

E

Encapsulation, 79
Enhancement
 maintenance category, 5
Epistemic complexity, 9
Essence, 4
E type system, 6
Exploratory test, 211

F

Factory method, 49, 121
Fail-fast, 91
Failures, 5
Faults, 5
Feasibility study, 166
Fishbone, *see* Cause/effect diagram
Framework, 54

G

God class, 248

H

Half sync-half async pattern, 283
Heisenbug, 100

I

Information hiding, 79
Infrastructure as code, 324
Intentional programming, 36
Inversion of control, 60
Is-a relationship, 79
Ishikawa, *see* Cause/effect diagram

K

Kano model, 163

L

Layered approach, 41

Learning path, 10

Liskov substitution principle, 79

M

Maintainability, 4

Maintenance activity, 5

Maintenance phase, 5

Model-view-controller pattern, 219

N

N-version programming, 250

O

Optimistic locking, 97

P

Parkinson's law, 7

Perfective

 maintenance category, 5

Pessimistic locking, 97

Preventive

 maintenance category, 5

Primal-dual method, 261

P type system, 6

Q

Quality, 243

 assurance, 12

 attributes, 11

 control, 12

R

Read-time convenience, 53, 80

Reuse, 54

S

Safety argument, 32

Separation of concerns, 62, 218

Service table, 167

Single level of abstraction, 34

Smoke testing, 74

Software architecture, 40

Software engineering method and theory
(SEMAT), 4

Strategy design pattern, 144

Stratified (layered) design, 35

S type system, 6

Sustainability, 247

System requirements, 159

T

Technical debt, 6

Template design pattern, 144

Test-driven development (TDD), 41

Total cost of ownership, 6

U

UML use case diagram, 164

Usability, 219

User requirements, 159

V

Validation, 32

Verification, 32

W

Warning suppression, 80

Write-time convenience, 80

See also Read-time convenience