

Index

A

abbreviations, accessibility, 331

acceptance testing, 5, 137, 207

- acceptance, 207
- black box, 207
- Cucumber, 229–235
- customer, 207
- executable specification, 207
- functional requirements, 207
- Given, When, Then syntax, 209
- iteration and, 207
- legacy code, 238–239
- nonfunctional requirements, 207
- system, 207

accessibility, 305. See also disabilities

- bandwidth, 309
- Cynthia Says, 346
- difficulty of, 308
- InFocus, 347
- myths, 306–308
- standards
 - Australia, 336
 - Canada, 336
 - European Union, 336–337
 - Japan, 337
 - Section 508, 335
 - United Kingdom, 337
 - WCAG 1.0, 333–334
 - WCAG 2.0, 334–335
- toolbars
 - Section 508 Toolbar, 348
 - Web Accessibility Toolbar, 347
 - WebAIM WAVE toolbar, 348
- W3C CSS Validation Service, 343–344
- W3C HTML Validation Service, 343
- WebAIM WAVE, 344–346

accessibility testing, 137, 337–339

- ASP.NET MVC applications, 348–352
- Fangs, 342

importance of, 308–309

JAWS (Job Access With Speech), 341–342

Lynx, 340

screen readers, 340

text-based browsers, 339

tools, 339

WebForms, 348–352

when to do, 307

accessible sites

abbreviations, 331

acronyms, 331

alternative text

- context, 322

- effective, 321–323

- when to add, 320–321

CAPTCHA, 332

forms

- labels, 325

- layout, 323–325

- validation, 325

frames, 332

images, 320

interfaces, 331–332

JavaScript and, 329–330

media

- audio descriptions, 328–329

- captioning, 328

- controls, 328

navigation, 332

tables, 325–328

timeouts, 333

user-preferred text size, 331

AccountTest function, 194

acronyms, accessibility, 331

ActiveCache object, 131

ad hoc testing, 137

adaptive keyboards, 317

AddressController, 109

addresses, displaying, 109–114, 131–132

AddToBasket method

AddToBasket method, 95, 99

ADO.NET Data Services, 1, 13

repository, inserting data, 121

agile software development

iterative processes, 9

tests, 7

aging, disabilities and, 309

AJAX, UI testing, 196

AJAX (Asynchronous JavaScript and XML), 1, 191

delay, 244

AJAX Framework, 13

alternative text

accessible sites, 320–321

effective, 321–323

context, 322

ambiguity, 243

Anders, Mark, 14

anti-patterns, mocking, 45–46

ANTS Memory Profiler (RedGate), 264

ANTS Performance Profiler (RedGate), 264

ANTs Profiler (RedGate), 287–292

Apache Jakarta Velocity project, 66

app.config, 130

NHibernate, 120

arthritis, 317

ASP, classic ASP, 14

ASP.NET

caching, 118

controls, 62

membership security provider, 126

security, 118

ASP.NET MVC Framework, 1, 13, 65–66

Accounts folder, 69

ActionResult, 73

applications

accessibility testing, 348–352

structure, 67–70

content directory, 69

controller, 66

creating, 71–76

DB, 70

design pattern, 66

downloading, 67

GetModel method, 75

HelloWorldMVC, 69

helper methods, 83

Home folder, 69

installation, 67

model, 66

Models folder, 69

ProductRepository, creating, 76–79

ProductsModelMenu, 75

scripts directory, 69

Tests.Integration assembly, 70

Tests.Unit assembly, 70

text assemblies, 70

UI, 70

view, 66

ViewData, 73

ViewResult, 73

Views folder, 69

Wrox Pizza Store, 67–116

ASP.NET WebForms, 1, 13, 61

GetData method, 64

view, interface implementation, 63

ASPnetCredentialsService, 126

ICredentialsService, 126

ISecurityContext, 127

ASPnetDBHelper method, 128

aspx file, 62

ASPX tags, 62

assemblies, WatiN, 190

attacking the data, 355

attacking the mechanism, 355

attention, cognitive disabilities, 319

attributes

TestClass, 20

TestFixture, 17

TestMethod, 20

audio descriptions of images, 328–329

Audit (Web Events), 250

auditory disabilities, 316

Australia's accessibility standards, 336

authentication

ASP.NET membership security provider, 126

broken, 381–382

manual testing, 249

automation. See also UI (user interface) testing

FitNesse, 211

what to automate, 200–201

B

base events (Web Events), 250
Basel, 353
BaseUITest class, 179, 182
BDD (Behavior Driven Development), 4, 46
 And, 48
 Cucumber, 229
 Given, 48
 Ruby, 47
 TDD and, 46
 Then, 48
 When, 48
Beck, Kent, 2
black box test, 5, 207
 test monkey, 166
black hat hacker, 354
Black Marble, 220
blindness/low vision, 312–315
 cataracts, 312
 diabetic retinopathy, 313
 glaucoma, 313
 keratoconus, 313
 macular degeneration, 315
 retinitis pigmentosa, 315
blocks, Ruby, 231
boiler plate code, 29
boundary conditions, test suites, 52
Brail, 66
Braille display, 312
broken authentication, 381–382
broken windows syndrome, 39
Brooks, Frederick, 5, 141
browser
 driving, 143–148
 impersonating, 142
BrowserShots, 251
Brup, 392
brute force attacks, 355
bug hunts, 246
bugs, 1
 cost of, 11
 regressions, 38
Build method, 181
business scenarios, automation, 200

C

C#, NUnit, 2
cache control, 155
Cache object, 131
caching, 131
 ActiveCache object, 131
 InMemoryCacheService, 131
Camtasia, 246
Canada's accessibility standards, 336
capacity planning, 277–278
 bandwidth, 278
 failure prediction, 278
 goals, 278
capacity testing, 137
CAPTCHA (Completely Automated Public Turing Test to Tell Computers and humans Apart), 332
captioning, accessible sites, 328
CartEmpty control, output, 104
Cassini testing pattern, 167–170
cataracts, 312
categories, 85
 adding, 124–125
 ProductCategory, 88
cerebral palsy, 317
chaining method calls, 181
Chelimsky, David, 3
classes
 move to class refactoring, 55
 NHibernateRepositoryTests, 121
 ProductRepositoryTests, 124
 responsibility, 23
classic ASP, 14
closed (Open Closed Principle), 25
cloud computing, performance testing, 279–280
CMS (Content Management System), 307
code, smells, 37
code-behind files, SOLID and, 62
code coverage, 19
code profiling, 278–279
code reviews, 37
cognitive disabilities, 318–320
 attention, 319
 linguistic comprehension, 319
 memory, 319

cognitive disabilities (*continued*)

- problem solving, 319
- readability tests, 319–320
- reading comprehension, 319
- verbal comprehension, 319
- color keyed information, 310**
- ColumnFixture, 215**
- Compuware Dev Partner, 264**
- ConfigurationManager, 130**
- constructors, overloading, dependencies and, 35**
- continuous integration, 20**
- controls, accessible sites, 328**
- cookies**
 - manipulation, 354
 - positioning, 354
- coupling, 140**
- CreateShowLink, 90**
- Credentials service, UserID and, 110**
- cross-browser testing, 251–255**
 - automation, 198–200
- Cruise Control, 20**
- cryptography, Insecure Cryptographic Storage, 382–383**
- CSRF (cross-site request forgery), 377**
- CSS (cascading style sheets), W3C CSS Validation Service, 343–344**
- Cucumber, 47, 229–235**
 - Watir, 234, 235–238
- Cunningham, Ward, 3, 211**
- CurrentCategory property, verification, 94**
- customer test, 5**
- Cynthia Says, 346**

D

- data, attacking, 355**
- Data Dude, 123, 257**
- debugging**
 - origins of term, 2
 - vs. testing, 2
- DefaultNHibernateConfig object, 78, 123**
 - GetTestDatabase method and, 123
- delay, 243–244**
- dependencies**
 - constructor overloading, 35
 - new, 35

- dependency, 32**
- deployment testing, 137**
- design**
 - changes, 53–59
 - TDD and, 36
- deuteranomaly color blindness, 310**
- developer testing, 4, 16**
- developers, testing and, 9–10, 15–16**
- diabetic retinopathy, 313**
- dictionary attacks, 355**
- disabilities. See also accessibility**
 - aging and, 309
 - auditory, 316
 - Census Bureau, 309
 - cognitive, 318–320
 - attention, 319
 - linguistic comprehension, 319
 - memory, 319
 - problem solving, 319
 - readability tests, 319–320
 - reading comprehension, 319
 - verbal comprehension, 319
 - motor, 317–318
 - adaptive keyboards, 317
 - arthritis, 317
 - cerebral palsy, 317
 - head wand, 318
 - mouth sticks, 318
 - muscular dystrophy, 317
 - oversized trackballs, 318
 - Parkinson's disease, 317
 - spinal cord injuries, 317
 - photosensitive epilepsy, 315
 - visual
 - blindness/low vision, 312–315
 - color blindness, 310–311
- DisplayCart, output, 104**
- displaying products, on web page, 71**
- Django, 65**
- DLR (Dynamic Language Runtime), 47**
- documentation**
 - manual testing, 244
 - vulnerability assessment, 388–391
- DoFixture, 215**
 - story-based model, 216
- DoS (Denial of Service) attack, 355**

driving browser, 143–148

screen capturing, 146–147

Visual Studio web tests, 147–148

DRY (don't repeat yourself), 149**dummy objects, definition, 4****E****edge cases, 248–249****email servers, 132–133****EmailService object, 132****encoding, 155****epilepsy, photosensitive, 315****error handling, improper, 377–380****error messages, 377–378**

manual testing, 245

errors

WatiN, 190–191

Web Events, 250

European Union's accessibility standards, 336–337**events, unsolicited, 140–141****executable specification, 207****ExecuteDbCommand method, 129****expense, 5–6****exploratory testing, 137, 245–246****expressions, lambda, 43****extension methods, 185****external services, 247–248****Extract method, 54****F****fail, 4****Failure to Restrict URL Access, 385–386****failures, random, 141****fake objects, definition, 4****Fangs, 342****faults, simulating, 44****Fennell, Richard, 220****Fiddler, 292–293****FireBug, 255****Firebug, 293–294****firefighting defects, 265****Firefox, Selenium IDE, 159****FIT (Framework for Integrated Testing), 3, 211****FitNesse, 3, 211**

ColumnFixture, 215, 228

DoFixture, 215, 228

SubWikis, 217

WatiN, 224–229

wiki, 211

parts, 227

FitServer, 213**Flesch-Kincaid Grade Level test, 320****Flesch Reading Ease test, 319****fluent interface, 181****Fluent NHibernate, 77, 118**

mappings and, 118

FormCollection, 105**Fowler, Martin, 53, 72****fragile tests, 37–38, 139**

layout changes, 140

frames, accessible, 332**functional requirements, 207****functional testing, 4, 136–138****functions**

AccountTest, 194

GetLorumIpsum, 171

GetSpecialCharacters, 172

UnitTestsToRun, 194

G**Gallio, 21****Gamma, Erich, 2****Garbage Collector, 274–275****GenerateSchemaCreationScript method, 123****GetAll method, 122****GetCategoriesFromProductList method, 88****GetControllerInstance method, 81****GetData method, 64****GetItems method, 186–187****GetLorumIpsum function, 171****GetModel method, 75****GetPassword method, 197****GetPrice() method, 85****GetProductsByCategory method, 91, 124****GetSpecialCharacters function, 172****GetTestDatabase method, 123****GetUser method, 127****Given, When, Then syntax, 209, 230**

glaucoma, 313
GO statements, 129
graphical user interface software testing, 137
GUI tests, 4
Gunning Fog Index, 320
Guthrie, Scott, 14

H

Haack, Phil, 133
hackers, 356

- black hat, 354
- Script Kiddies, 356–357
- white hat, 354

head wand, 318
Heartbeat (Web Events), 250
HelloWorldMVC, 69
helper methods, ASP.NET MVC Framework, 83
hi-jacking, 355
HIPPA, 353
history of testing tools, 2–3
Hopper, Admiral Grace Murray, 2
HTML

- parsing, 142–143
- W3C HTML Validation Service, 343

HtmlHelper, 83
HTTP

- editing, 392
- expected status code, 155
- proxying, 392

HttpContext, ISecurityContext, 127
Huggins, Jason, 3
Hyper-V, 258

I

IAddressRepository, mocking, 113
ICacheService, 131
ICredentialsService, 126

- implementing, 100

IDbConfigure, 78
IEnumerable, 194
IIS (Internet Information Service), restart

- application pool, 247

IIS log parser, 282–285
IL (intermediate language), 45

images, accessible sites, 320
impersonating browser, 142
improper error handling, 377–380
Index() method, 71
Index page, 88
Index_Returns_All_Products_In_Database, 71
infinite input sequence, 141
InFocus, 347
information leakage, 377–380
injection flaws, 369
InMemoryCacheService, 131
INotificationService interface, 132
input sequence, 141
Insecure Communications, 383–385
Insecure Cryptographic Storage, 382–383
Insecure Direct Object Reference, 375–376
integration tests, 4

- speed, 117
- unit tests comparison, 117–132

interfaces

- accessible sites, 331–332
- boilerplate code, 29
- cohesion, 29
- fluent, 181
- implementing, ASP.NET WebForms, 63
- INotificationService, 132
- IRepository, 72

inversion, 32
IoC (Inversion of Control), 35

- integration with StructureMap, 79–81

IProductRepository

- ProductsMenuController, 79
- stubs, 72

IRepository interface, 72

- inheritance, 72

IronRuby, 47
ISecurityContext, 127

- ASPnetCredentialsService, 127
- HttpContext, 127

IShoppingCartRepository, 125

- requesting shopping cart, 96

isolated tests, 16
iteration, acceptance tests, 207
iterative processes, 8–9

J**Japan's accessibility standards, 337****JavaScript**

- accessible sites and, 329–330
- disabling, 255
- manual testing, 255–256
- TDD methods, 192
- UI testing, 191–198

JavaScriptTestRunner, 3**JAWS (Job Access With Speech), 341–342****jQuery, 192, 255****jQuery, hyperlinks and, 108****jQuery Framework, 2****jQuery JavaScript library, 192****jQuery project, 2****JUnit, 2****K****keratoconus, 313****Key Performance Metrics, 271****keyboards, adaptive, 317****L****lambda expressions, 43****legacy code, 6**

- acceptance testing and, 238–239
- unit tests, 115–116

legal blindness, 312**linguistic comprehension, 319****LINQ to Objects, 92****Liskov Substitution Principle, 27–29****load injectors, 268****Load Runner (HP), 264****load test, 5****load test architecture, performance testing, 268–269****load testing, 137**

- Fiddler, 292–293
- Firebug, 293–294
- IIS log parser, 282–285
- Microsoft Network Monitor, 287
- NetStat, 285
- Performance Monitor, 286–287
- RedGate ANTs Profiler, 287–292
- tools, WCAT, 280–282

VSTS, 295–302

YSlow, 294–295

load testing patterns

- code profiling, 278–279
- log replaying, 279

log replaying, 279**Log4Net, 250****LogOn method, 189****Lorem Ipsum, 7****L0phtCrack, 393****Lynx, 340****M****macular degeneration, 315****malicious file execution, 373–374****managers, testing and, 10–11****manual testing, 210**

- authentication, 249
- cases, 256–259
- cross-browser testing, 251–255
- documentation, 244
- edge cases, 248–249
- error messages, 245
- exploratory testing, 245–246
- external services, 247–248
- importance of, 241–242
- JavaScript, 255–256
- network testing, 248
- part automation, 242
- unit tests and, 7
- usability, 243–244
- WebEvents, 250

mappings

- Fluent NHibernate, 118
- Session object, 119
- testing, SQLite database, 120

Martin, Robert (Uncle Bob), 23**MbUnit, 20****mechanism, attacking, 355****media, accessibility**

- audio descriptions, 328–329
- captioning, 328
- controls, 328

MembershipUser, 127

- StubMembershipUser, 127

memory, cognitive disabilities, 319

memory performance counters, 272–273

method calls, chaining, 181

methods

- AddToBasket, 95
- ASPnetDBHelper, 128
- Build, 181
- ExecuteSqlCommand, 129
- extension methods, 185
- GenerateSchemaCreationScript, 123
- GetAll, 122
- GetCategoriesFromProductList, 88
- GetControllerInstance, 81
- GetData, 64
- GetItems, 186–187
- GetModel, 75
- GetPassword, 197
- GetPrice(), 85
- GetProductsByCategory, 91, 124
- GetTestDatabase, 123
- GetUser, 127
- Index(), 71
- LogOn, 189
- productListed, 225
- pull-up, 55–58
- RenderPartial, 84
- SaveOrUpdate, 122
- SetControllerFactory, 81
- ShouldContain, 185
- Show, 91
- Strengths, 216
- Test, 17
- transport methods, 155
- WithDiv, 93

Microsoft Application Center Test, 3

Microsoft Network Monitor, 287

MITM (man in the middle) attack, 356

mock objects

- definition, 4
- IAddressRepository, 113

mocks, 40–41

- anti-patterns, 45–46
- Rhino Mocks, 43
- stubs comparison, 41–43
- TDD, 46
- TypeMock, 44–45

monochromacy color blindness, 310

MonoRail, 65

motor disabilities, 317–318

- adaptive keyboards, 317
- arthritis, 317
- cerebral palsy, 317
- head wand, 318
- mouth sticks, 318
- muscular dystrophy, 317
- oversized trackballs, 318
- Parkinson's disease, 317
- spinal cord injuries, 317

mouth sticks, 318

move to class refactoring, 55

MSGBuild, 21

MSTest, 20

muscular dystrophy, 317

MVP (Model-View-Presenter), 62–65

myths

- accessibility, 306–308
- agile software development, 7
- expense, 5–6
- legacy code, 6
- test creation, 6
 - timing, 8

N

N-Stealth, 392

NAnt, 21

navigation, accessibility, 332

Nessus, 392

.NET CLR Exceptions, 276

.NET CLR loading, 276

.NET CLR LocksAndThreads, 276

.NET CLR memory, 274–275

.NET framework, 20

NetStat, 285

network testing, 248

NewAddress control, 110

Newkirk, James, 20

NHibernate, 76, 118

- app.config, 120
- DefaultNHibernateConfig object, 123
- enabling, 118
- GenerateSchemaCreationScript method, 123
- mapping, 77
 - schema and, 119
- SQL Server, 120

SQLite, 120
 verification, 126

NHibernateFixtureBase, 119

NHibernateModel, 119–120

NHibernateRepository, 121

ProductRepository, 121

NHibernateRepositoryTests class, 121

Nikto/Wikto, 392

nonfunctional requirements, 207

nUnit, 2

NVelocity, 66

O

Object Mothers, 89

open (Open Closed Principle), 25

ORM (object relational mapping), 76

overloading constructors, dependencies, 35

oversized trackballs, 318

OWASP (Open Web Application Security Project), 354

tools, 361

Top 10, 361

- broken authentication, 381–382
- CSRF (cross-site request forgery), 377
- Failure to Restrict URL Access, 385–386
- improper error handling, 377–380
- information leakage, 377–380
- injection flaws, 369–373
- Insecure Communications, 383–385
- Insecure Cryptographic Storage, 382–383
- Insecure Direct Object Reference, 375–376
- malicious file execution, 373–374
- Session Management, 381–382
- XSS (cross-site scripting), 362–369

P

pair programming, 36

Parkinson's disease, 317

Paros, 392

parsing, HTML, 142–143

part automation, 242

pass, definition, 4

password crackers, 393

penetration testing, 356

Performance Monitor, 286–287

performance testing, 5, 137

- application under test, 269
- ASP.NET Performance Counters, 274
- availability, 262, 266
- baseline, 262–263
- capacity planning, 277–278
- client questions, 267
- cloud computing, 279–280
- Compuware Dev Partner, 264
- concurrency, 266
- disk performance counters, 273
- environment, 268–269
 - design, 270–271
- HP Load Runner, 264
- IIS performance counters, 273
- importance of, 265–266
- in-house testing, 271
- Key Performance Metrics, 271
- KPIs (Key Performance Indicators), 277
- load controllers, 269
- load injectors, 268
 - hardware, 271
- load test architecture, 268–269
- load testing
 - Fiddler, 292–293
 - Firebug, 293–294
 - IIS log parser, 282–285
 - Microsoft Network Monitor, 287
 - NetStat, 285
 - Performance Monitor, 286–287
 - RedGate ANTs Profiler, 287–292
 - VSTS, 295–302
 - YSlow, 294–295
- load testing patterns
 - code profiling, 278–279
 - log replaying, 279
- load testing tools, WCAT, 280–282
- .NET performance counters, 274–276
- objective setting, 267
- outsourcing, 271
- overview, 261–264
- performance analysis, 276–277
 - checklist, 277
- performance monitoring
 - memory performance counters, 272–273
 - real-time analysis, 272
- performance times, 263

performance testing (*continued*)

- RedGate ANTS Memory Profiler, 264
- RedGate ANTS Performance Profiler, 264
- requirements, 266–277
- response time, 262, 267
- Root Cause Analysis, 277
- scalability, 262, 266
- SSL overhead, 271
- throughput, 262
- tools, choosing, 264
- utilization, 262
- VSTS (Visual Studio Team Foundation Test Suite), 264
- WCAT (Web Capacity Analysis Tool), 264
- PersistenceSpecification object, 118**
- Pex, 249**
- photosensitive epilepsy, 315**
- Playback model (RhinoMocks), 44**
- principles before guidelines, 305**
- problem-solving, cognitive disabilities, 319**
- ProductBuilder, 181**
- ProductCategory, 88**
- production bugs, 38–39**
- productListed method, 225**
- ProductRepository, 121**
 - creating, 76–79
 - extending, 124
 - implementation, 78
 - inheritance, 122
 - NHibernateRepository, 122
- ProductRepositoryTests class, 124**
- products, displaying, 118–123**
- ProductsMenu, creating, 81–87**
- ProductsMenuModel, 88**
- ProductsMenuPage model, 186**
- ProductsModelMenu, 75**
- programming, pair programming, 36**
- properties, CurrentCategory, 94**
- protanomaly color blindness, 310**
- Prototype, 192**
- pull-up methods, 55–58**

Q

- qUnit, 2, 192**
 - HTML files, 194

R

- RainborCrack, 393**
- rainbow tables, 355**
- random failures, 141**
- random testing, UI testing, 170**
- readability of tests, 17**
- readability tests, cognitive disabilities, 319–320**
- reading comprehension, 319**
- Record model (RhinoMocks), 44**
- record results, 155**
- recorded tests, 149**
 - WatiN recorder, 149–158
- recovery testing, 137**
- Red, Green, Refactor, 21–22, 58**
- Red Gate SQL Data Generator, 257**
- redirects, 155**
- RedirectToRouteResult, 99**
- refactoring, 22, 53**
 - Extract method, 54
 - move to class, 55
 - pull-up methods, 55–58
 - TestDriven Refactoring, 58
- Regex expressions, 231**
- regression testing, 5, 137**
- regressions, 38**
- Rehabilitation Act of 1973, 335**
- RenderPartial method, 84**
- Repositories**
 - implementation, 118
 - inserting data, 121
 - testing, 120
- requests, parse dependent, 155**
- response time goal, 155**
- responsibility, classes, 23**
- results, recording, 155**
- retinitis pigmentosa, 315**
- Rhino Mocks, 43**
 - stub object creation, 72
- RouteSetup object, 94**
- RPS (Requests Per Second), 5, 137**
- RSpec, 47**
 - Cucumber, 229
- Ruby**
 - BDD, 47
 - blocks, 231

Cucumber, 229–235
 methods, brackets, 231
 step_definitions folder, 231

Ruby on Rails, 65

RubyGems, 234

runners, 21

S

sanity testing, 137

SaveOrUpdate method, 122

scalability testing, 137

screen capturing, 146–147

screen readers, 312

accessibility testing, 340

Script Kiddies, 356–357

scrum methodology, 207

Section 508, 335

Section 508 Toolbar, 348

security

attacking the data, 355
 attacking the mechanism, 355
 black hat hacker, 354
 brute force attacks, 355
 cookie manipulation, 354
 cookie positioning, 354
 dictionary attacks, 355
 DoS (Denial of Service) attack, 355
 MITM (man in the middle) attack, 356
 penetration testing, 356
 rainbow tables, 355
 session hi-jacking, 355
 spoofing, 356
 white hat hacker, 354

security scanners, 392

security testing, 137, 353

compliance, 359–360
 consultants, 358–359
 experience level, 358
 importance of, 356–358
 knowledge transfer, 358
 liability and, 357
 OWASP (Open Web Application Security Project),
 360–386
 policy laws, 359–360
 tools, 357, 392–393

WASC (Web Application Security Consortium), 360

Selenium, 3, 158

element locators, 162–163
 Selenium client libraries, 161
 Selenium Grid, 164–166
 Selenium IDE, 159
 Selenium RC, 159–163
 Selenium server, 160–161

servers, email servers, 132–133

session hi-jacking, 355

Session Management, 381–382

Session object, mappings, 119

session rules, automation, 200

session timeouts, 247

SetControllerFactory method, 81

shopping cart, adding items, 95–108, 125–131

ShoppingCart object, 98

ShoppingCartController, 95

ShoppingCartLine, output, 104

ShoppingCartPage code, 186

ShouldContain method, 185

Show method, 91

Silverlight, 1, 13

Silverium, 204

slow tests, 39–40

smells. See code smells; test smells

SMO (Server Management Objects), 129

smoke testing, 137

SMTP server, 133

fake, 133

software bug, 1

SOLID design principles, 23–36

code-behind files, 62

D (Dependency Inversion Principle), 31–36

I (Interface Segregation Principle), 29

L (Liskov Substitution Principle), 27–29

O (Open Closed Principle), 24–27

S (Single Responsibility Principle), 23–24

SOX, 353

Spark, 66

special characters, UI testing, 171–173

spinal cord injuries, 317

spoofing, 356

SQL Compare, 123, 129

SQL Data Compare, 129

SQL Server Management Objects, 129

SQLite, 120, 133
SRP (Single Responsibility Principle), 25
SSLDigger, 393
Stockdale, Mike, 3
Strengths method, 216
stress test, 5
stress testing, 138
StructureMap
 initialization, 80
 as IoC framework, 80
 IoC integration, 79–81
StructureMapControllerFactory, creation, 81
stub, definition, 4
StubMembershipUser, 127
 MembershipUser, 127
stubs, 40–41
 mocks comparison, 41–43
sUnit, 2–3
system test, 5

T

tables, accessible sites, 325–328
TAD (Test After Development), 21
TamperData, 376
TDD (Test Driven Development), 4, 7, 11, 21, 46, 207
 design and, 36
 JavaScript, 192
Team City, 20
TED (Test Eventually Development), 21
test data builders, 180–184
test double, definition, 4
test fixture, definition, 4
Test method, 17
test methods, public, 17
test monkey, 166–167
test smells, 37
 development and, 40
 fragile tests, 37–38
 production bugs, 38–39
 slow tests, 39–40
test suites
 boundary conditions, 52
 streamlined, 210

TestableHttpContext, 128
TestClass attribute, 20
TestDriven Refactoring, 58
TestFixture attribute, 17
 BaseUITest class, 179
testing
 automated, 11–13
 developer testing, 16
 developers and, 9–10, 15–16
 managers and, 10–11
 manual testing, 210
 unit testing, 16–20
 vs. debugging, 2
testing tools, history, 2–3
TestMethod attribute, 20
tests
 agile software development, 7
 confidence in, 19
 creation, 6
 timing of, 7–8
 definition, 3
 fragile, 37–38
 legacy code, 6
 readability, 17
 recording as manual, 246
 slow tests, 39–40
 structure, 18
text size, accessibility and, 331
think time, 155
Threat Classification project (WASC), 360
Throw extension method, 44
timeouts, 155
 accessible sites, 333
 session timeouts, 247
timing problems, 140
tinnitus, 316
toolbars
 Section 508 Toolbar, 348
 Web Accessibility Toolbar, 347
 WebAIM WAVE toolbar, 348
transport methods, 155
TTFB (Time To First Byte), 5, 137, 262
TTLB (Time to Last Byte), 262
TypeMock, 44–45

U**UI (user interface), 4, 81**

- automation
 - Playback, 148
 - Record, 148
- elements, hard-to-test, 140
- as object model, 174–185

UI (user interface) testing

- AJAX, 196
- automation, 135–136
 - importance of, 138–139
 - problems with, 139–141
- driving browser, 143–148
- focus, 174
- impersonating the browser, 142
- JavaScript, 191–198
- listing products, 176–180
- naming tests, 176
- parsing HTML, 142–143
- patterns
 - Cassini, 167–170
 - test monkey, 166–167
- randomness, 170
- recorded tests, 149
- Selenium suite, 158–166
- special characters, 171–173
- test data builders, 180–184
- tools, 141–148
 - WCAT (Web Capacity Analysis Tool), 142
- verification, 184–185
- Wrox Pizza, 175–176

UltraEdit, 250**Uncle Bob, 23****unit testing, 16–20**

- aim of, 17
- as fad, 8
- ASP.NET WebForms, 62–66
- automatic, 50
- definition, 4
- documentation, 49
- failures, 52
- focus, 51
- inverse relationship of behaviors, 51
- isolation, 50
- manual testing and, 7
- name, 49

- ordering, 53
- repeatability, 51
- size, 49
- speed, 49

unit testing framework, 17, 20–22**United Kingdom's accessibility standards, 337****UnitTestsToRun function, 194****unsolicited events, 140–141****usability**

- ambiguity and, 243
- delay, 243–244
- manual testing and, 243–244

usability testing, 138**user interface code, automating, 7****user story, 207**

- DoFixture, 216

V**ValidateRequest, 249****validating forms, 325****validation**

- W3C CSS Validation Service, 343–344
- W3C HTML Validation Service, 343

verbal comprehension, 319**verification, 184–185****ViewData property, 73****visual disabilities**

- blindness/low vision, 312–315
 - Braille display, 312
 - cataracts, 312
 - diabetic retinopathy, 313
 - glaucoma, 313
 - keratoconus, 313
 - macular degeneration, 315
 - retinitis pigmentosa, 315
 - screen readers, 312
- color blindness, 310–311

Visual Studio, web tests, 147–148**Visual Studio 2005 Test Edition, 3****Visual Studio 2008 Professional, 3**

- Web Templates, 67

Visual Studio for Database Professionals (Data Dude), 123**VMWare, 258****volume testing, 138**

VSTS (Visual Studio Team System Test), 151, 158, 264, 295

executing load test, 300–302
maintaining load test, 300
scenarios, 297–300
wizard, 296

vulnerability assessment, 386

discovery, 386–388
documentation, 388–391
verification, 388

W

WASC (Web Application Security Consortium), 360

WatiN, 3, 144–146, 185–190

API, 144
assemblies, missing, 190
driving Internet Explorer, 144
errors, 190–191
FitNesse, 224–229
HTML elements, 145
test recorder, 149

- comments, 154–155
- context parameters, 156
- navigation requests, 152
- request properties, 155–156
- validation, 150
- validation rules, 153–154
- Visual Studio Web Testing, 150–152
- VSTS, 158
- web test editor, 152–153
- web tests, running, 156–158

WatiNBrowserFactory, 226

Watir (Web Application testing in Ruby), 3

Cucumber, 234, 235–238

WAVE, 344–346

toolbar, 348

W3C HTML Validation Service, 343

WCAG 1.0, 333–334

WCAG 2.0, 334–335

WCAG (Web Content Accessibility Guidelines), 333

WCAT (Web Capacity Analysis Tool), 3, 142

load testing, 280–282
performance testing, 264

Web Accessibility Toolbar, 347

Web Application Security Statistics, 360

Web Application Stress Tool, 3

Web Events, 250

Audit, 250
Base Events, 250
Errors, 250
Heartbeat, 250

Web Security Threat Classification, 360

Web Services, 1, 13

WebAIM WAVE, 344–346

WebAIM WAVE toolbar, 348

WebForms, 201–203

accessibility testing, 348–352
labels, 325
layout, 323–325
legacy code, 115–116
validation, 325

WebInspect, 392

WebProxy Editor, 392

WebScarab, 362, 392

white box test, 5

Pex, 249

white hat hacker, 354

Wilson, Brad, 20

WithDiv method, 93

workflows, automation, 200

WroxPizza.Core, 69

WroxPizza.Web, 69

X

Xenocode, 254

XP (extreme programming), 207

XSS (cross site scripting) attacks, 249

OWASP, 362–369

xUnit, 4

xUnit.NET framework, 20

Y

YSlow, 294–295