

Index

\$()

invoking, *184*

JavaScript event handlers, *182*

\$.ajax, *186*

A

ABC score

definition, *310*

example, *310*

method-level refactoring, *314*,
316

Abstract factory pattern

DIP and OCP, *381*

implementation, *372*

OCP, *371*

Abstraction

clarity improvement, *16*

comments, *308*

leaky, *424*

A/B testing, feature flags, *403*

Acceptance testing

Cucumber, *224*

explicit vs. implicit

requirements, *236*

software, *14*

Accessor method, Ruby objects,

79

Access token, authentication, *143*

Achievable, SMART user stories,
219

Action caching, *408*, *409*, *411*

ActionController, Rails basics,
100

ActionView

link_to, *112*

metaprogramming, *113*

Rails basics, *100*

ActiveModel, validation, *137*

Active monitoring, *406*

ActiveRecord

associations, *148*, *150*

basics, *107*

caching, *408*, *409*

DataMapper comparison, *150*,
151

destroy, *127*

DIP, *379*

group method, *158*

lifecycle hooks, *139*

models, *57*

proxy object, *110*

Rails basics, *100*, *105*, *108*

Active record architectural pattern

RDBMS, *58*

Transform View, *64*

ActiveRelation (ARel)

caching, *410*

composable scopes, *157*

ActiveSupport

decoration, *373*

delegation, *377*

ActiveView::Base, *111*

ActiveX, *417*

Adapter, jQuery, *168*

Adapter pattern, DIP, *379*, *381*

Adaptive maintenance, *299*

Add, Git basics, *447*

Adleman, Leonard, *414*

Adobe ActionScript, *166*

Advice, AOP, *141*

Aggregation, UML, *368*

Agile Alliance, *25*

Agile Development, Virtuous

Triangle of Engineering

SaaS, *33*, *433*

Agile lifecycle

BDD pros and cons, *249*

code reviews, *335*

continuous refactoring, *325*

decision to use, *26*

deployment, *398*

fallacy, *32*

iteration example, *29*, *216*, *436*

legacy code, *298*

maintenance, *320*

overview, *256*, *298*, *362*, *394*

Pivotal Labs, *222*

plan-and-document comparison,
26, *245*, *289*

process, *25*

productivity comparisons, *28*

purpose, *216*

Rails use, *5*

TDD, *25*

testing comparisons, *27*

variants, *25*

Waterfall/Spiral comparison,
436

working code documentation,
299

Agile Manifesto

cost estimation, *222*

development model, *25*

inspiration for, *239*

original proposal, xv

overview, *24*

Agile stakeholders, *216*

AJAX, *see* Asynchronous

JavaScript And XML
(AJAX)

ajax function, *186*

ALGOL, *164*

Algorithm analysis, Turing

Award, *254*

alias_method_chain

decoration, *373*

example, *375*

Allen, Frances, *440*

Alloy, requirements

documentation, *241*

Amaya, *124*

Amazon

Agile lifecycle, *27*

cost of being down, *419*

household name, *4*

page reload issues, *417*

public cloud, *395*

severity 1 bugs, *344*

SOA vs. siloed software, *7*

software education, xiv

Amazon CloudFront, *410*

Amazon Elastic Block Store, *445*

Amazon Elastic Compute Cloud
(EC2)

Bookware VM, *444*

FarmVille, *12*

RightScale, *397*

Amazon Machine Image (AMI),
444

Amazon Web Services

abusive database queries, *411*

APIs, *60*

cloud computing, *12*

AMI, *see* Amazon Machine Image
(AMI)

Amiko, DIP, *378*

AmikoAdapter, DIP, *379*, *381*

And

Cucumber keyword, *224*

imperative vs. declarative
scenarios, *236*

and_return, Red–Green–Refactor
TDD, *264*

Anonymous lambda expression,
blocks, *84*

Antipatterns

Data Clump, *369*

definition, *363*

overview, *362*

AOL, *42*

- AOP, *see* Aspect-oriented programming (AOP)
- Apache web server, 42
- Apdex, *see* Application Performance Index (Apdex)
- API key, TMDb, 257, 273, 274, 276, 277, 279
- APM, *see* Application Performance Monitoring (APM)
- Apple, JavaScript engines, 207
- Apple Newton, 176
- Apple Pages, code editing, 445
- Application architecture
 - ActiveRecord for models, **57**
 - client-server, **42**
 - HTML and CSS, **48**
 - HTTP and URIs, **44**
 - MVC, **54**
 - routes, controllers, REST, **59**
 - Template Views, **62**
 - three-tier and horizontal scaling, **51**
- ApplicationController
 - authentication, 145
 - filters and callbacks, 160
 - OmniAuth gem, 144
 - Rails app example, 111
- Application Performance Index (Apdex), 397
- Application Performance Monitoring (APM), 405
- Application Programming Interface (API)
 - DIP, 379
 - jQuery, 181, 182
 - JSON, **198**
 - RESTful, 59, **256**
 - silo bookstore service, 8
 - SOA bookstore service, 9
 - TMDb, 257
- Application server, three-tier architecture, 51
- Aptana, 445
- Archetypal principals, 415
- ARel, *see* ActiveRelation (ARel)
- ArgumentError, TMDb, 274, 276, 278
- Arguments
 - method calls, 77
 - methods on collections, 86
 - variable length lists, 83
- Ariane 5 rocket explosion, 4
- Arithmetic overflow attack, 421
- Around-filter, caching, 409

- Array, JavaScript, 205
- Array class
 - each method, 88
 - Ruby objects, 73
- Aspect-oriented programming (AOP)
 - COME FROM comparison, 160
 - overview, 141
- Asset pipeline, 173
- assigns(), RSpec, 267
- Associations
 - ActiveRecord vs. DataMapper, 150
 - characteristics, 153
 - error checking, 160
 - example, 147, 148, 150
 - overview, **147**
 - RESTful routes, **154, 157**
- Asynchronous JavaScript And XML (AJAX)
 - coining of term, 168
 - effectiveness, 202
 - example, 187
 - Jasmine specs, 195
 - overview, **185**
 - same-origin policy, 202
 - stubbing, 202
 - testing, **190**
 - XHR, 187
 - XML use, 168
- Atomic migration, 400
- Attributes, RDBMS, 57
- Authentication
 - definition, 142, 395
 - example, 144
- Authorization, 142
- Automatic theorem proving, 287
- Automation
 - autotest, 260
 - FIRST, 258
 - productivity improvement, 17
 - tests with Cucumber, 246
- Automation for repeatability, Bundler, 102
- Autotest
 - automation with, 260
 - usage, 260
- Availability
 - definition, 395
 - quantification, **404**

B

Backfilling, 322

- Background keyword, TMDb example, 235
- Backlog, user stories, 220
- Backtrace, RASP, 117
- Backus, John, 16, 164
- Backus-Naur form, 164
- Backwards traceability, 244
- Bar#foo, 259
- Base URI, **45**
- Basic blocks
 - control flow graph, 310
 - testing approaches, 285
- BasicObject class, Ruby objects, 74
- BDD, *see* Behavior-driven development (BDD)
- BDUF, *see* Big Design Up Front (BDUF)
- Beautiful code
 - definition, 13
 - maintenance costs, 309
 - testing comparisons, 27
- Beck, Kent, 25
- before(:all), dependencies, 291
- before block
 - factory, 270
 - TMDb, 266, 267, 267
- Before-filters
 - caching, 408, 409
 - force_ssl, 415
 - “owning” resources, 156
- before_save
 - ActiveRecord lifecycle hooks, 139
 - example, 140
- Behavioral patterns, GoF, 364
- Behavior diagrams, UML, 367
- Behavior-driven development (BDD)
 - Agile, xv
 - basic ecosystem, 238
 - bug testing, 344
 - Jasmine, 196
 - legacy code, 300
 - overview, **216**
 - pros and cons, **247**
 - TMDb example, 231
 - UI, 227
- belongs_to
 - associations, 149
 - example, 150
- Berners-Lee, Tim, 66
- Bezos, Jeff, 59, 332
- Big Brother Bird, 425
- Big Design Up Front (BDUF), 21

- Bit blit, 16
 - Black-box tests, 14, 285
 - Blocks
 - methods on collections, 86
 - overview, **84**
 - Rails basics, 100, 105
 - Rails reuse, 6
 - XML Builder, 87
 - Booch, Grady, 366
 - Bookstore service
 - siloed software, 8
 - silo vs. SOA, 8
 - SOA, 9
 - Bookware VM
 - Amazon EC2, **444**
 - Rails basics, 100, 102
 - Bot, input checks, 137
 - Bottlenecks, finding, **405**
 - Bottom-up integration, 286
 - Branch coverage (C1), 282
 - Branches
 - changes to master, 354
 - commit example, 340
 - effective use, **339**
 - Git, 341
 - Git commands, 342
 - long-lived, 343
 - mishandling changes, 354
 - version control, 336
 - Brooks, Fred, Jr., 19, 330
 - Buffer overflow attack, 420
 - Bugs
 - caching behavior, 423
 - reporting and fixing, **343**
 - Bundler
 - automation for repeatability, 102
 - Cucumber, 226
 - development vs. production environments, 454
 - gem modification, 129
 - Rails basics, 100
 - But, Cucumber keyword, 224
 - button_to, Rails app example, 128
- C**
- CA, *see* Certificate authority (CA)
 - C0 coverage, 282, 307
 - C1 coverage, 282
 - C2 coverage, 283
 - Cache invalidation, 407
 - Caching
 - bugs leading to, 423
 - code example, 409
 - example, 408
 - object storage, 411
 - rendering and database improvement, **407**
 - CAD, *see* Computer Aided Design (CAD) tools
 - Callbacks
 - authentication, 143
 - event-driven programming, 190
 - JavaScript, **180**
 - pitfalls, 159
 - Call coverage (S1), 282
 - Campfire
 - information sharing, 222
 - pull requests, 335
 - Capability Maturity Model (CMM), 23
 - Capistrano, 398
 - Capybara
 - BDD pros and cons, 247
 - overview, **224**
 - running, **226**
 - testing tools relationship, 248
 - use, 225
 - Cardinality, associations, 147
 - Cartesian product
 - foreign keys, 147
 - through-associations, 152
 - Cascading Style Sheets (CSS)
 - AJAX, 188
 - constructs, 49
 - dynamic HTML, 180
 - editor for, 124
 - flash message, 124
 - Haml constructs, 63
 - introduction, 49
 - Lo-Fi UI, 230
 - overview, **48**
 - Cassandra, DataMapper, 150
 - CDN, *see* Content delivery network (CDN)
 - Cerf, Vinton E. “Vint,” 44
 - Certificate authority (CA), 415
 - Change control board,
 - maintenance requests, 321
 - change event, JavaScript, 183, 185
 - Change management
 - definition, 351
 - plan-and-document, 244, 320
 - Change points, legacy code, 300, 303
 - Change request forms, 320
 - Change requests, maintenance, 320
 - Characterization tests
 - creation, **304**
 - legacy code, 300
 - TimeSetter, 307
 - CI, *see* Continuous integration (CI) testing
 - Clarity via conciseness,
 - productivity improvement, 16
 - Class
 - definition in Ruby, 78
 - Ruby objects, 74, **78**
 - Class class, Ruby objects, 74
 - Class diagram, UML, 367, 368, 371, 377
 - Class inheritance, Ruby objects, 74
 - ClassName, method calls, 76
 - Class–Responsibility–Collaborator (CRC) cards, 304, 305
 - Class variables
 - cohesion, 369
 - Ruby, 79
 - click event, authentication, 180
 - Client pull, server push
 - comparison, 47
 - Client-Server Architecture, **42**
 - design pattern, 43
 - high-level view, 42
 - Client-side JavaScript
 - definition, 166
 - DRY, 167
 - tips for Ruby programmers, **169**
 - Clone, version control, 336
 - Closures
 - Rails reuse, 6
 - Ruby blocks, 84
 - Cloud computing, **10**
 - definition, 12
 - Heroku deployment, **451**
 - performance and scalability, 422
 - CloudFoundry, 395
 - CloudFront, 410
 - CLU language, yield basis, 89, 90
 - Clusters, 11
 - CMM, *see* Capability Maturity Model (CMM)
 - COCOMO, *see* Constructive Cost Model (COCOMO)
 - Codd, Edgar F. “Ted,” 57
 - Code reviews
 - necessity, **335**
 - plan-and-document, 348
 - Code smells
 - avoidance, 324
 - definition, 310

- examples, *311, 326*
- overview, **309**
- Code-to-test ratio
 - definition, 282
 - minimum requirements, 290
- Coding standards, 333
- CoffeeScript, 173, 207, 208, 451
- Cohesion
 - definition, 369
 - example, 369
 - refactoring, 316
- Collections
 - Enumerable, 88
 - has_many, 149
 - methods on, 86
 - operators, 85
 - proxy object, *110*
 - reuse, 17
- COME FROM, *160*
- Comments
 - bad example, *308*
 - level of abstraction, *308*
 - overview, **308**
- Commit
 - branch example, *340*
 - confusion with push, 453
 - Git basics, 446, 447
 - large commits, 453
- Commit squashing, Git, 337
- Common Vulnerabilities and Exposures database, 420
- Communication
 - client pull vs. server push, 47
 - cookies, 46
 - HTTP request, **45**
 - HTTP and URIs, **44**
 - SaaS demands, 10
- Comparison operators, JavaScript, 206
- Compiled languages, continuous integration, 399
- Compilers
 - automation, 17
 - JavaScript, 207
 - Turing Award, *164, 440*
- Complex feature, feature flags, 403
- Complexity, SOLID, 365
- Composition
 - duck typing, 378
 - preference to inheritance, *377*
 - UML, 368
- CompuServe, 42
- Computer Aided Design (CAD)
 - tools, 17
- Conciseness
 - productivity improvement, 16
 - reflection and meta-programming, *113*
- Concurrent Versions System (CVS), 355
- Configuration management
 - IEEE 828-2012, 352
 - varieties, 351
- Configuration setting, Git basics, 446
- Construction, RUP phase, 21
- Constructive confrontation, people management, 348
- Constructive Cost Model (COCOMO), 242
- Constructors, JavaScript, **174**
- Content delivery network (CDN), 410
- Continuous deployment, **398**
- Continuous integration (CI)
 - testing
 - compiled languages, 399
 - definition, 14
 - deployment, **398**
 - rare conditions, 421
- Control flow coverage, 285
- Control flow graph, 310, *310*
- Controllers
 - completion, 263
 - development, 262
 - fat, 129
 - function in MVC, 259
 - index RESTful action, *112*
 - MVC, 54
 - overview, **59**
 - private methods, security issues, 417
 - Rails app example, **111**
 - routes, *104*
 - show action, *114*
 - TMDb example, 267
 - validation interactions, *139*
- convert method
 - characterization tests, 306
 - example, 312
 - refactoring, 314, 318
- Cookies, HTTP, 46
- Corrective maintenance, 298
- Cost estimation
 - Agile lifecycle, **222**
 - change management, 244
 - plan-and-document, 241
- Cost of software design, pitfalls, 32
- CouchDB, DataMapper, *150*
- Coverage concepts, **282**, 282
- CRC cards, *see* Class–Responsibility–Collaborator (CRC) cards
- create
 - associations, 154
 - form submission, **120**
 - Rails app example, 108, *122*
 - replacement for, *138*
- create!, ActiveRecord, 107
- Creational patterns, GoF, 364
- Crisscross merge, Git, 343
- Cross-site request forgery (CSRF), 415
- Cross-site scripting
 - characteristics, 416
 - fuzz testing, 286
- CRUD
 - associations, 155
 - definition, 58
 - edit/update and destroy, **125**
 - models, 107
 - Rails basics, 102
 - routes, *60*
 - views, 111
- Cruft, deployment abuse, 423
- CSRF, *see* Cross-site request forgery (CSRF)
- csrf_meta_tags, 416
- CSS Zen Garden, 48
- Cucumber
 - automation, 17
 - BDD ecosystem, 238
 - BDD pros and cons, 247, 249
 - bug testing, 344
 - characterization tests, 306
 - example, 224, 227
 - integration tests, 259
 - keywords, *224*
 - overview, **224**
 - running, **226**
 - scenario comparisons, 236
 - test-automation, 246
 - testing tools relationship, 248
 - TMDb example, 231
 - UML diagrams, 367
- Curly braces, poetry mode, 76
- Current branch, 340
- Customer collaboration, maintenance, 320
- CVS, *see* Concurrent Versions System (CVS)
- Cyclomatic complexity
 - definition, 310

example, *310*
invention, *310*

D

Dahl, Ole-Johan, *134*

Dapper, *425*

Database index

abusive queries, *413*
example, *412*

Databases

abusive queries, **411**, *412*
ActiveRecord lifecycle hooks,
139
caching, **407**
legacy code exploration, *302*
manual modification, *129*
purpose, *53*
Rails app example, **104**

Database transaction, spec run in,
270

Datacenters, virtual machines, *11*

Data Clump, design smell, *369*

Data consistency, RDBMS
scalability, *64*

Data hiding, Ruby *vs.* Java, *79*

Data integrity, *396*

DataMapper

ActiveRecord comparison, *150*
Google AppEngine, *151*

Data protection, security, **414**

Data race attack, *421*

DDoS, *see* Distributed denial of
service (DDoS)

Debugger, Ruby, *123*

debugger statement, Rails app
example, *119*

Debugging

conventional, TDD comparison,
291

delta debugging, *435*

JavaScript, *172*

Rails app example, **117**

Declarative scenarios

example, *237*
testing, **236**

Declared exceptions, undeclared
comparison, *275*

Decompose Conditional

application, *316*
example, *317*
refactoring, *314*

Decorator pattern

application, *374*
caching, *409*

example, *374*

function, *373*

Defensive programming, basic
philosophy, *404*

Define–use (DU) coverage, *285*

Deflating, *173*

Delegation pattern

duck typing, *378*
example, *373*
preference to inheritance, *377*

Delta debugging, *435*

Denial-of-service attack, *417*, *426*

Dependability, SaaS demands, *11*

Dependencies

development *vs.* production
environments, *454*
replacing with seams, **313**

Dependency Injection Principle
(DIP), **378**

cached object storage, *411*
example, *381*
OCP impact, *381*

Deployment

continuous, **398**
cruft accumulation, *423*
feature flags, **400**
Heroku, **451**
overview, **394**

describe

Jasmine, *192*
TMDb, *266*, *267*, *272*

Deserializing, *57*, *199*

Design patterns

classes, **362**
client-server architecture, *43*
definition, *363*
Demeter Principle, **382**
DIP, **378**
LSP, **376**
OCP, **371**
plan-and-document, **386**
reuse, *17*
SRP, **369**
UML, **366**

Design reviews, plan-and-
document, *348*

Design smells

Data Clump, *369*
definition, *311*

destroy method, Rails app

example, *109*, **125**, *127*

Destructive methods, usage, *86*

Developer keys, RESTful APIs,
257

Development database, legacy
code exploration, *302*

DigiNotar, *426*

Dijkstra, Edsger W., *160*, *287*

DIP, *see* Dependency Injection
Principle (DIP)

Directory structure, Rails app
example, *101*

Disciplined processes, *21*

Distributed denial of service
(DDoS), *426*

DNS, *see* Domain Name System
(DNS)

DNS spoofing, *426*

Documentation

embedded, *301*
working code, *299*

Document Object Model (DOM)
dynamic HTML, *180*

JavaScript, **177**

JavaScript event handlers, *180*

jQuery, *200*

jQuery example, *182*

manipulation with jQuery, *178*,
179

Rotten Potatoes, *177*

use of term, *177*

DOM, *see* Document Object
Model (DOM)

Domain language, Cucumber, *236*

Domain Name System (DNS)

hostname mapping, *44*
root zone, *45*

Domain-specific language (DSL)
domain language difference,
236

RSpec, *259*

Don't Repeat Yourself (DRY)

factory, *270*

JavaScript, *167*

JavaScript custom events, *185*

metaprogramming, *81*

MVC, **136**

Rails app example, *104*

reuse, *17*

Ruby's metaprogramming

features, *70*

search_tmdb, *268*

similar things, *126*

TMDb controller example, *267*

TMDb example, *235*

Double, **261**, *263*

down method, Rails app example,
105

Down-migration, *402*

- Driver, pair programming, 334
- DRY, *see* Don't Repeat Yourself (DRY)
- DSL, *see* Domain-specific language (DSL)
- Duck typing
 - abstract factory pattern, 372
 - composition/delegation, 378
 - each method, 85
 - overview, **88**
 - Time class, 89
- DU coverage, *see* Define–use (DU) coverage
- Dumb fuzzing, 285
- Dynabook, 432
- Dynamic HTML, 180
- Dynamic language features
 - pitfalls, 83
 - readable specs, 271
- Dynamic typing
 - Lisp, 88
 - reuse, 17
 - Ruby objects, 73

E

- each method
 - Array class, 88
 - definition, 84
 - duck typing, 85
 - iterator definition, 89
- Eager load query, 412
- eBay
 - Agile lifecycle, 27
 - software education, xiv
- EC2, *see* Amazon Elastic Compute Cloud (EC2)
- ECMAScript, 166
- Edit
 - partial, 136
 - Rails app example, **125**, 126
- EDSAC, Turing Award, 2
- Eich, Brendan, 166
- Emacs
 - creation, 444
 - mix-ins, 88
 - working with code, 445
- Embedded documentation, RDoc, 301
- Embedded domain-specific language, RSpec, 259
- Encapsulate Field, refactoring, 319
- Encapsulation, Ruby *vs.* Java, 79
- Encryption, data protection, 414

- Engineering disciplines, 21
- Entry/Exit coverage (S1), 282
- Enumerable
 - collection methods, 88
 - Rails app example, 108
 - yield example, 90
- Environments, Rails basics, 104, 106
- Epic, Pivotal Tracker, 221
- Epoch, 82
- eRB renderer, 417
- Error checking, associations, 160
- errors method, validation, 137, 139
- Eudora, 42
- Event-driven programming, 190
- Event handlers, JavaScript, 180
- Events
 - JavaScript, **180**
 - JavaScript custom events, 185
- Examples
 - completion, 263
 - RSpec, 260, 262
 - sample code, 266
 - TMDb controller, 267
- Exceptions
 - declared *vs.* undeclared, 275
 - TMDb, 274, 275, 277, 278
- Expectations, **265**
 - careless use of, 247
- Expiration
 - bugs, 423
 - caching, 407
- EXPLAIN, SQL, 414
- Explicit requirement
 - testing, **236**
 - TMDb, 273
- eXtended HyperText Markup Language (XHTML)
 - HTML 5, 48
 - JavaScript DOM, 177
- eXtensible Markup Language (XML)
 - AJAX, 168
 - HTML 5, 48
 - JavaScript DOM, 177
 - SPAs, 166, 199
 - structured data, 173
 - stubbing, 202
 - Transform View, 64
- External domain-specific language, 259
- External monitoring, 406
- Extract Class, refactoring, 370
- Extract Method

- application, 316
- example, 316, 317
- refactoring, 314, 315

- Extreme Programming (XP)
 - Agile variants, 25
 - pair programming, 333

F

- Facade pattern, DIP, 379, 381
- Facebook
 - Agile lifecycle, 27
 - database use, 53
 - has_and_belongs_to_many, 154
 - hits per day, 422
 - household name, 4
 - OAuth, 142
 - OmniAuth, 145
 - pair programming, 335
 - SMART user stories, 219
 - software education, xiv
 - SSO, 142
- Facebook Connect
 - authentication, 395
 - SSO side effects, 146
- Facebook Platform, SOA, 7
- Factories, **269**, 271
 - fixture comparison, 198
- FactoryGirl, 270, 271
- failure handler function, 202
- fake_results, TMDb, 266
- FakeWeb, 202, 278
- Fallacy, 32
- false, Ruby regular expressions, 72
- FarmVille, growth, 12
- Feature
 - cool, successfulness, 246
 - Cucumber, 224
 - mock-up distinction, 246
- Feature branch, 340
- Feature flags
 - deployment, **400**
 - example, 401
 - uses, 403
- Fibonacci scale, 220
- Fielding, Roy, 59
- Fields, active records, 57
- Filters
 - definition, 139
 - pitfalls, 159
 - scopes, 158
- find-based methods, Rails app
 - example, 108
- find_in_tmdb

- exception, 274
- functional tests, 284
- Internet stub, 276
- Red–Green–Refactor TDD, 263, 267
- stubbing, 202
- TDD, 272
- usage, 267
- Firebug, 174
- Firefox, client-server architecture, 42
- Firewalls, secure platform fallacy, 423
- FIRST, 258**
- First-class objects, JavaScript, 174, 175
- Five nines, 11
- Fixnum class
 - metaprogramming, 81
 - Ruby objects, 73
 - time arithmetic, 82
- Fixtures, **269**
 - example, 194
 - factory comparison, 198
 - HTML example, 195
 - Jasmine-JQuery, 201
 - legacy code exploration, 302
 - YAML files, 271
- Flash
 - CSS, 124
 - Rails app example, **122**
 - XSS security issues, 417
- Flickr, branch management, 340
- Float class, Ruby objects, 73
- force_ssl, 415
- Foreign keys, **147**
 - abusive database queries, 413
 - example, 149
- Fork-and-pull, remote collaboration, 339
- Formal methods
 - definition, 287
 - NASA cost example, 287
 - testing alternatives, 14
- Formal specification languages, requirements documentation, 241
- :format, 62
- Formatter class
 - design patterns, 374
 - example, 372
 - OCP, 371
- Form submission
 - authentication, 146
 - Rails app example, **120, 121**

- Form validation, client-side, testing, 197
- Forward traceability, 244
- Fragment caching, 410
- Framework concepts
 - JavaScript
 - AJAX, **185**
 - client-side, **169**
 - DOM and jQuery, **177**
 - events and callbacks, **180**
 - functions and constructors, **174**
 - overview, **166, 206**
 - single-page apps and JSON APIs, **198**
 - testing, **190**
 - Rails, advanced
 - associations and foreign keys, **147**
 - MVC DRYness, **136**
 - queries with reusable scopes, **157**
 - RESTful routes for
 - associations, **154**
 - SSO and third-party authentication, **142**
 - through-associations, **152**
 - Rails basics (*see also* Ruby on Rails basics)
 - controllers and views, **111**
 - databases and migrations, **104**
 - debugging, **117**
 - edit/update and destroy, **125**
 - form submission, **120**
 - models, **107**
 - overview, **100**
 - redirection and flash, **122**
 - Ruby basics
 - blocks, **84**
 - classes, methods, inheritance, **78**
 - idiomatic language, **93**
 - iterators via yield, **89**
 - metaprogramming, **81**
 - mix-ins and duck typing, **88**
 - objects, **73**
 - operation as method, **75**
 - overview, **70**
 - Free Software Foundation, 444, 445
 - Front Controller
 - MVC, 55
 - web app comparison, 55
 - full_messages, validation, 137
 - Full table scan, 412

- Full URI, **45**
- Function, definition, 256
- Functional programming, Ruby blocks, 85
- Functional requirements, elicitation, 239
- Functional tests
 - bugs, 344
 - characterization tests, 306
 - definition, 283
 - testing method comparison, 284
- Function points, 242
- Functions, JavaScript, **174**
- Function scope, JavaScript, 205
- Fuzz testing
 - definition, 285
 - software, 15
- G**
- Garrett, Jesse James, 168
- Gemfile, 190, 454
- gem mysql, 451
- Gems
 - manual modification, 129
 - Rails basics, 100, 257
 - Ruby, **256**
- Generalize Type, refactoring, 319
- General Markup Language, 48
- Generator, Rails app example, 105
- GenericMailer, DIP, 381
- Geographically distributed programming teams, Agile vs. plan-and-document, 26
- get
 - functional tests, 283
 - TMDb, 265
- GET, POST comparison, 128
- Getter method, Ruby objects, 79
- Git
 - add, commit, index commands, 447
 - basic skills, **446**
 - branch commands, 342
 - branches and rebasing, 343
 - branch use, 341
 - change tracking, 338
 - commit commands, 338
 - common commands, 450
 - hosting services, 447
 - merge commands, 338
 - merging conflicts, 336
- git fetch, 337
- GitFlow, branch management, 340
- GitHub

- basic skills, **447**
- bookware, 442
- bug testing, 344
- code reviews, 335
- continuous integration, 399
- coursework backup, 445
- information sharing, 222
- remote collaboration, 339
- version control, 336
- git merge, 337
- git pull, 337
- Given, Cucumber keyword, 224
- Glass-box tests, 14, 285
- Globally-unique ID (guid), 143
- Global object, JavaScript, 174
- GNU, 443, **444**
- GoF design patterns, 363, **364**
- Google
 - Agile lifecycle, 27
 - code reviews, 422
 - coding standards, 333
 - Dapper, 425
 - error message searches, 118, 442
 - household name, 4
 - importance of testing, 249
 - JavaScript engines, 207
 - OAuth, 142
 - responsiveness, 397
 - software education, xiv
 - testing variety, 290
- Google+, siloed software, 7
- Google Analytics, 425
- Google AppEngine, 51
 - cloud computing, 12
 - DataMapper, 150, 151
- Google Apps
 - availability, 404
 - OmniAuth, 145
- Google Closure, 207
- Google Docs
 - client-server architecture, 43
 - information sharing, 222
 - JavaScript, 166, 207
- Google Maps
 - AJAX, 168
 - client-side SPAs, 198
 - XmlHttpRequest, 185
- Go To Statement Considered Harmful* (Dijkstra), 160
- Graceful degradation, JavaScript, 166
- Gray, Jim, 68
- Group, Rails basics, 100
- guid, *see* Globally-unique ID (guid)
- H**
- HABTM, *see*
 - has_and_belongs_to_many (HABTM)
- Haml, *see* HTML Abstraction Markup Language (Haml)
- Hardening, continuous integration, 399
- has_and_belongs_to_many (HABTM), 154
- Hash class, Ruby objects, 73
- Hashing, abusive database queries, 413
- has_many
 - example, 150
 - function, 149
 - through-associations, 152
- has_one, function, 149
- HCI, *see* Human Computer Interface (HCI)
- HEAD, web browsers, 60
- Heavyweight processes, 21
- Heroku
 - bookware, 442
 - caching, 411
 - cloud deployment, **451**
 - deployment automation, 398
 - indexing, 413
 - PaaS, 395
 - Progstr-Filer addon, 417
- Hidden form fields,
 - authentication, 146
- Higher-level programming languages, clarity improvement, 16
- Highly Productive Framework and Tools, Virtuous Triangle of Engineering SaaS, 33, 433
- HiJax, 186
- Hoare, Charles Antony Richard, 98
- Hopper, Grace Murray, 4
- Horizontal scaling, **51**, 65
- Hostnames
 - mapping by DNS, 44
 - URI, 45
- HTML, *see* HyperText Markup Language (HTML)
- HTML Abstraction Markup Language (Haml)
 - before-filters for “owning,” 156
- caching, 407
- CoffeeScript, 208
- common constructs, 63
- debugging, 119
- edit vs. new views, 126
- flash message, 124
- Lo-Fi UI, 230
- Rails app example, 102, 113
- sanitization, 112
- template views, 62, 63
- TMDb code, 233
- HTTP, *see* HyperText Transfer Protocol (HTTP)
- httpperf, 406
- HTTP method
 - request, **45**
 - route helpers, 113
 - routes, 60
- HTTP redirect, Rails basics, 103
- HTTP request
 - example, **45**
 - Rack, 373
 - stateless protocol, 46
- HTTPS, *see* Secure HTTP protocol (HTTPS)
- HTTP server, three-tier architecture, 51
- HTTP verb, 59
- Human Computer Interface (HCI), 217
- HyperText Markup Language (HTML)
 - client-side JavaScript, 171
 - delete action, 127
 - development, 66
 - dynamic HTML, 180
 - fixture example, 195
 - Haml constructs, 63
 - HTML 5 client pull vs. server push, 47
 - HTML 5 features, 48
 - introduction, 48
 - Jasmine-JQuery sandbox method, 197
 - JavaScript DOM, 177
 - JSON response, 200
 - overview, **48**
 - Rails basics, 100
 - security vulnerabilities, 417
 - SPAs, 198
 - stubbing, 202
 - sync with app, 204
 - yield, 92
- HyperText Transfer Protocol (HTTP)

- Bookware VM on EC2, 444
- client pull vs. server push, 47
- communication, 44
- development, 66
- request example, **45**
- stateless protocol, 46
- three-tier architecture, 52

Hypervisor, 443

I

- IANA, *see* Internet Assigned Numbers Authority (IANA)
- IDEs, *see* Integrated Development Environments (IDEs)
- IEEE 828-2012, 352
- IEEE 1012-2012, 350
- IEEE 1219-1998, 321
- IEEE 16326-2009, 346
- IEEE/ANSI standard 830/1993, 23
- IEEE Standard 829-2008, 287, 288
- IEEE Standard 830-1998, 239, 240
- Immobility, SOLID, 365
- Imperative scenarios, **236**, 237
- Implicit requirement, **236**
 - discovering and testing, **272**
- Inception, RUP phase, 21
- include
 - confusion with require, 93
 - module contract checking, 88
- Independence
 - factories, 271
 - fixtures, 270
- Index, Git basics, 447
- Index action
 - controller and template, 112
 - Rails app example, 111
 - redirection, 123
- Index controller, template views, 62
- Indexing, Heroku, 413
- Infinitely scalable, 11
- Inheritance
 - composition/delegation preference, 377
 - design patterns, 374
 - Ruby, **78**
- Initialize, classes, 78
- Inspections, plan-and-document, 348
- Instance variables, 78, 369
- Instrumentation, debugging, 119

- Integrated Development Environments (IDEs), 445
- Integration testing
 - bugs, 344
 - characterization tests, 306
 - Cucumber, 224
 - explicit vs. implicit requirements, 236
 - RSpec vs. Cucumber, 259
 - software, 14
 - testing method comparison, 284
 - types, 286
 - unit test comparison, **282**
- Interaction diagrams, UML, 367
- Interface Segregation Principle (ISP), 371
- Internal domain-specific language, RSpec, 259
- Internal monitoring, 405
- Internet Assigned Numbers Authority (IANA), 45
- Internet Explorer 5
 - JavaScript, 168
 - XmlHttpRequest, 185
- Internet Service Provider (ISP), deployment, 394
- Interpolation, template views, 62, 63
- Interpreters, automation, 17
- Interviewing, requirements elicitation, 239
- IP address
 - networking explanation, 45
 - TCP/IP network, 44
- IPv6, networking explanation, 45
- ISO 9001 standard, 420
- ISP, *see* Interface Segregation Principle (ISP); Internet Service Provider (ISP)
- Iteration, Agile lifecycle, 216, 216, 436
- Iterators
 - making with yield, **89**
 - Ruby blocks, 84

J

- Jacobson, Ivar, 366
- Jasmine
 - AJAX testing, 190, 195
 - commonly used features, 193
 - directory creation, 191
 - setting up, 191
 - spies, 194
- Jasmine-JQuery

- commonly used features, 193
- fixtures, 200, 201
- sandbox method, 197

Jasmine-Rails, 192

Java

- continuous integration, 399
- exceptions, 275
- JavaScript relationship, 169

JavaScript

- AJAX, **185**
- basic constructs, 170
- Capybara, 225
- client-side, **169**
- deployment, 399
- DOM and jQuery, **177**
- dynamic HTML, 180
- ECMAScript, 166
- events and callbacks, **180**
- functions and constructors, **174**
- issues with, 205
- jQuery API, 181
- over-reliance on, 204
- overview, **166**, **206**
- partial, 136
- prototype inheritance, 176
- Rails app example, 113
- same-origin policy, 202
- Scheme origins, 166
- silent failures, 203
- site enhancement, 203
- sync with app, 204
- task parallelism, 190
- testing, **190**
- unobtrusive code, 172
- XSS security issues, 417

JavaScript Application Programming Interface (JSAPI), 168, 177, 196

JavaScript engine, 207

JavaScript Object Notation (JSON), 166, **198**

- Jasmine-JQuery, fixtures, 201
- response example, 200
- structured data, 173
- stubbing, 202

Java vs. Ruby

- classes, 79
- coding in Ruby, 92
- Enumerable, 88
- feature translation, 80
- metaprogramming, **81**
- objects, **73**
- Ruby overview, **70**
- syntax errors, 119
- type casting, 75

- yield, 90
- Join, foreign keys, 147
- Join point, AOP, 141
- Joy, Bill, 445
- jQuery
 - Jasmine, 191
- jQuery
 - Adapter, 168
 - API, 181
 - DOM manipulation, 178, 179
 - example, 182, 188
 - JavaScript event handlers, 180, 182
 - JavaScript for Ruby
 - programmers, 171
 - minifying, 173
 - overview, 177
- jQuery()
 - invoking, 184
- jQuery Mobile, 207
- JSAPI, *see* JavaScript Application Programming Interface (JSAPI)
- JSLint, 205
- JSON, *see* JavaScript Object Notation (JSON)

K

- Kahan, William, 360
- Kahn, Bob, 44
- Kay, Alan, 432
- Keypair, 415, 448
- Key Performance Indicators (KPIs), 405
- Knuth, Donald, 254
- KPIs, *see* Key Performance Indicators (KPIs)

L

- Lack of Cohesion of Methods (LCOM), 369, 369
- lambda, TMDb, 274
- Lampson, Butler, 296
- LAMP Stack, 51
- Latency
 - added, effects, 422
 - monitoring, 425
 - overprovisioning, 404
 - responsiveness, 396
- Lazy evaluation
 - caching, 410
 - scopes, 159
- LCOM, *see* Lack of Cohesion of Methods (LCOM)

- Legacy code
 - characteristics, 298
 - definition, 13
 - exploration, 301
 - Microsoft Zune, 306
 - testing comparisons, 27
- Lexical scoping, method calls, 76
- Lightweight process, Agile, 25
- Lines of code (LOC)
 - cost estimation, 241
 - legacy code exploration, 303
 - method length, 312
- link_to, Rails app example, 112, 114
- Linux
 - creation, 444
 - VirtualBox, 443
- Liskov, Barbara, 89, 376, 392
- Liskov Substitution Principle (LSP)
 - example, 374
 - overview, 376
 - UML class diagram, 377
- Lisp
 - dynamic typing, 88
 - inventor, 10
- LiveScript, 166, 205
- Load balancers, 3-tier architecture, 52
- LOC, *see* Lines of code (LOC)
- Lo-Fi user interface
 - sketches and storyboards, 227
 - sketches without storyboards, 246
 - TMDb example, 230
 - TMDb storyboard, 232
- logger.debug, 119
- Logic tier, 3-tier architecture, 51
- Longevity testing, monitoring, 406
- Long method, code smell, 314
- Looking up a method, Ruby objects, 74
- LSP, *see* Liskov Substitution Principle (LSP)

M

- Mac OS X
 - Unix tools, 447
 - VirtualBox, 443
- Magic number, code smells, 316
- MailerMonkey, 379, 381
- Maintainability, categories, 298
- Maintenance manager, 319
- Maintenance phase

- characterization tests, 304
- comments, 308
- IEEE 1219-1998, 321
- legacy code, 298
- legacy code exploration, 301
- metrics, code smells, SOFA, 309
 - plan-and-document, 319
 - refactoring, 313
- Maintenance requests, 320
- Maintenance software engineers, 319
- Makefiles, automation, 17
- Malicious attackers
 - authentication, 146
 - niche apps, 424
- Malicious users, input checks, 137
- Malware, deployment, 399
- Man-in-the-middle attack, 426
- Markup language, 48
- Marshalling, 57, 173
- Martin, Robert C., 365
- Massive Open Online Course (MOOC)
 - book materials, 4, xviii
 - enabling components, 437
- Master-slave, three-tier architecture, 52
- Match-condition, TMDb, 265
- max method, Enumerable, 88
- McCabe, Frank, Sr., 310
- McCarthy, John, 10
- MCDC, *see* Modified Condition/Decision Coverage (MCDC)
- Mean time between repairs (MTTR), 420
- Mean time to failure (MTTF), 420
- Measurable, SMART user stories, 219
- Mechanization of Contract Administration Services (MOCAS), 13
- Merge-commit, 340
- Merge conflicts, version control, 336
- Merging
 - branch management, 342
 - mishandling changes, 354
 - version control, 336
- Metaclass, Ruby objects, 74
- Metaprogramming
 - abstract factory pattern, 372
 - conciseness, 113
 - productivity improvement, 16

- programming as, **81**
- Rails reuse, 6
- route helpers, *113*
- scopes, *158*
- XML Builder, 87
- Method
 - collections, 86
 - definition, 59, 256
 - definition with self, 80
 - length, *312*
 - naming issues, 93
 - RSpec listing, *280, 281*
 - Ruby objects, **74, 78**
- Method call
 - number of arguments, 77
 - operation as, **75**
- Method chaining, Ruby objects, 74
- Method coverage (S0), 282
- method_missing
 - metaprogramming, 81
 - Rails app example, 108
 - XML Builder, 87
- Method stub, Red–Green–Refactor TDD, 263
- Metrics
 - common metrics, *311*
 - definition, 310
 - monitoring, 425
 - overview, **309**
 - plan-and-document, 348
 - rigid adherence, 324
- Microsoft
 - Agile lifecycle, 27
 - fuzz testing, 285
 - JavaScript engines, 207
 - OAuth, *142*
 - pair programming, 334
 - public cloud, 395
 - software education, xiv
 - Windows 95 release, 19
 - XmlHttpRequest, 185
- Microsoft Azure, 12, 395, 411
- Microsoft Internet Information Server, 42
- Microsoft JScript, *166*
- Microsoft Office 365, 5, 432
- Microsoft Office Excel 2013, 437
- Microsoft Word, code editing, 445
- Microsoft Zune, *306, 307*
- Middleware, *51*
- Migrations
 - atomic, 400
 - database modifications, 129
 - example, *401*
 - foreign keys, *149*
 - Rails app example, **104, 105, 106**
 - seeding comparison, *109*
- Minifying, *173*
- min method, Enumerable, 88
- Mix-ins, **88**
 - Rails reuse, 6
 - reuse, 17
- MOCAS, *see* Mechanization of Contract Administration Services (MOCAS)
- Mock
 - definition, 263
 - Movie object, 269
- Mocks, **265**
- Mock trainwreck, 291
- Mock-ups, completed feature distinction, 246
- Model checking, 287
- Models
 - ActiveRecords, **57, 107**
 - MVC, 54, *108*
- Model–View–Controller (MVC)
 - app requests, 56
 - controller function, 259
 - DRY, **136**
 - implementation detail exposure, 157
 - JavaScript, 166
 - model behaviors, *108*
 - overview, **54**
 - Rails basics, 100, *101*
 - template views, 63
 - web app comparison, 55
- Modified Condition/Decision Coverage (MCDC), 283
- Module
 - definition, 88
 - Rails basics, 100
- Module testing, 14
- Mojito, 207
- MongoDB, DataMapper, *150*
- Monitoring
 - approaches, 425
 - overview, **405**
 - plan-and-document, 242
 - types, *406*
- MOOC, *see* Massive Open Online Course (MOOC)
- Moore's Law, productivity improvement, 15
- Movie class
 - blocks example, 84
 - mock usage, 269
 - models, 107
 - Ruby objects, 78
- MoviesController class
 - Rails basics, 103, 111
 - Red–Green–Refactor TDD, 262
 - RSpec, 260, *260*
- MSN, 42
- MTTF, *see* Mean time to failure (MTTF)
- MTTR, *see* Mean time between repairs (MTTR)
- Multi-homing, networking explanation, 45
- Multiplicity, UML, 368
- Mutation testing
 - definition, 285
 - software, 15
- MVC, *see* Model–View–Controller (MVC)
- MySQL
 - cloud deployment, *451*
 - EXPLAIN, *414*
- N**
- Naming
 - bugs, 423
 - caching, 407
- NAT, *see* Network address translation (NAT)
- Naur, Peter, *164*
- Navigator, pair programming, 334
- Nebraska Student Information System, 419
- Negative expectations, careless use, 247
- Nested resources, 160
- NetBeans, 445
- Netflix, OAuth, *142*
- Network address translation (NAT), VM networking, *444*
- Networking
 - early work, *44*
 - explanation, 45
- Network interface device, 45
- Network protocol, 44
- new
 - associations, 154
 - form submission, **120**
 - JavaScript, *174, 176*
 - partial, *136*
- New Relic, 405, 425
- NewtonScript, *176*
- New York Times
 - Facebook Connect, *146*

- Facebook Platform, 7
- nil
 - Red–Green–Refactor TDD, 264
 - Ruby regular expressions, 72
- NilClass, debugging, 118
- Node.js, 166, 207
- Non-functional requirements
 - abusive database queries, **411**
 - availability quantification, **404**
 - caching, **407**
 - continuous integration and deployment, **398**
 - deployment, **394**
 - elicitation, 239
 - monitoring and bottlenecks, **405**
 - overview, **424**
 - plan-and-document, **419**
 - releases and feature flags, **400**
 - responsiveness quantification, **396**
 - security, **414**
- NoSQL storage systems,
 - DataMapper, 150
- n + 1 queries, 412
- Null Object pattern, 380, 380
- Nygaard, Kristen, 134

O

- OAuth, 142
- Object class, Ruby, 79
- Object literals, client-side
 - JavaScript, 169, 171
- Object-orientation (OO)
 - Ruby objects, 74
 - seams, 265
 - Turing Award, 134
- Object-oriented programming
 - reuse, 17
 - Turing Award, 432
- Objects, Ruby, **73**
- Observer, pair programming, 334
- OCF, *see* Open/Closed Principle (OCP)
- OmniAuth
 - authentication, 143
 - example, 144
 - Strategy pattern, 373
 - Twitter, 145
- OO, *see* Object-orientation (OO)
- OpenCL, JavaScript, 207
- Open/Closed Principle (OCP), **371, 381**
- OpenID, 396
- Open source software, Linux and GNU, 444
- Operation, method call, **75**
- Origin
 - sync issues, 354
 - version control, 336
- Overloading, jQuery, 178
- “Owning”
 - associations, 155
 - before-filters, 156

P

- PaaS, *see* Platform as a Service (PaaS)
- Page caching, 408, 411
- Page Controller
 - MVC, 54
 - web app comparison, 55
- Pair programming, **333**
 - Agile, xv
 - example, 334
 - member dynamics, 353
- Palm webOS, 207
- params[]
 - associations, 155
 - authentication, 145
 - create action, 122
 - form submission, 121
 - Rails app example, 104
 - Red–Green–Refactor TDD, 262
- Parentheses
 - number of arguments, 77
 - poetry mode, 76
 - Ruby regular expressions, 72
- Partial
 - DRY, 136
 - example, 136, 137
 - JavaScript, 136
 - new/edit templates, 136
- Partial URI, **45**
- Pastebin, 50, 102, 217
- Patches, legacy code, 299
- Path coverage (C2), 283
- Patterns, *see also* Design patterns
 - active record architecture, 58
 - GoF, 364
 - Transform View, 64
 - types, 366
- Payment Card Industry, 417
- PayPal, 417
- PdfFormatter, 373, 374
- Peer-to-peer architecture, 43
- Penetration tests, 421
- People management, plan-and-document, 348
- Per-computer performance, 65
- Perfective maintenance, 298
- Performance
 - apps in development, 422
 - cloud computing scalability, 422
 - database, caching, **407**
 - fast computer fallacies, 422
 - optimization with measurement, 422
 - overview, **424**
 - plan-and-document, 419
 - SOA servers, 423
- Perlis, Alan, 164
- Persistence tier, 3-tier architecture, 51
- PERT, *see* Program Evaluation and Review Technique (PERT)
- PhoneGap, 207
- PHP
 - DataMapper, 151
 - template views, 63
- Pitfall, 32
- Pivotal Labs
 - Agile cost estimation, 222
 - Jasmine, 190
 - pair programming, 334
 - software consultancy, 222
- Pivotal Tracker
 - automation, 17
 - bookware, 442
 - bugs, 344
 - UI, 221
 - user stories, **220**
- Plan-and-Documents
 - Agile comparison, 25, 26, 245
 - Agile testing comparison, 289
 - BDD pros and cons, 249
 - CMM, 23
 - decision to use, 26
 - early version, 18
 - formal specification languages, 241
 - function points, 242
 - IEEE Standard 829-2008, 288
 - IEEE Standard 830-1998, 240
 - lifecycle goals, 21
 - maintenance phase, **319**
 - maintenance plan example, 321
 - major tasks, 238
 - non-functional requirements, **419**
 - overview, **18, 243**

- productivity comparisons, 28
- project management, **345**, 355
 - RUP, 21, 22
 - software metrics, 310
 - Spiral lifecycle, 19, 20
 - testing comparisons, 27
 - testing overview, **286**
 - user stories, **238**
 - Waterfall lifecycle, 19
- Plan-driven processes, 21
- Platform as a Service (PaaS)
 - data encryption, 414
 - deployment, 395
 - monitoring, 405
 - rolling reboot, 404
- Poetry mode, 76
- Pointcut, AOP, 141
- Points
 - BDD pros and cons, 249
 - plan-and-document, 238
 - user stories, **220**
- Port numbers
 - IANA assignment, 45
 - TCP/IP, 45
 - URI, 45
- Positive expectations, careless use, 247
- POSIX, 207
- post
 - functional tests, 283
 - TMDb, 265
- POST, GET comparison, 128
- PostgreSQL, 413, 414, 451
- Post-receive URI repo, GitHub, 399
- Preflight checking, feature flags, 403
- Presentation tier, 51
- Preventive maintenance, 299
- Principal
 - authentication, 142
 - public-key cryptography, 415
- Principle of fail-safe defaults, 414
- Principle of least privilege
 - authentication, 146
 - security, 414
- Principle of psychological acceptability, 414
- printf debugging, 119
- Privacy, 395
- Probing, 406
- Production clients, 42
- Production database, legacy code exploration, 302
- Productivity

- Agile vs. plan-and-document, 28 improvement, **15**
- Product owner, Scrum, 333
- Profiling, 405
- Program Evaluation and Review Technique (PERT), 242
- Programming languages
 - productivity improvement, 16
 - refactoring, 319
 - SaaS frameworks, 5
 - seams, 265
 - style sheets, 333
 - Turing Award, 98, 164, 214, 254, 287, 392
- Programming frameworks, languages for, 5
- Program synthesis, 435
- Progstr-Filer, 417
- ProjectLocker
 - basic skills, **447**
 - bookware, 442
 - coursework backup, 445
 - version control, 336
- Project management
 - bugs, **343**
 - code reviews, **335**
 - Git branches, **339**
 - IEEE 16326-2009, 346
 - overview, **354**
 - pair programming, **333**
 - plan-and-document, **345**
 - two-pizza team and Scrum, **332**
 - version control, **336**
- Promiscuous pairing, 334
- Properties, client-side JavaScript, 169
- protect_from_forgery, 416
- Prototype inheritance, JavaScript, 174, 176
- Prototyping, Spiral lifecycle, 19, 20
- Provenance, authentication, 142
- Proxy object, Rails app example, 110
- Proxy pattern, DIP, 380
- Public cloud services, *see* Utility computing
- Public-key cryptography
 - basic concept, 415
 - networking explanation, 45
 - Turing Award, 414
- Public repos, remote
 - collaboration, 339
- Pull changes, version control, 336
- Pull requests, GitHub, 335

- Push changes
 - confusion with commit, 453
 - git hosting services, 447
 - version control, 336

Q

- QA, *see* Quality assurance (QA)
- Quality, definition, 13
- Quality assurance (QA)
 - bugs, 343
 - overview, 258
 - testing overview, **13**
- Queries, composition with reusable scopes, **157**
- Query plan, EXPLAIN, 414
- quirksmode.org, 168

R

- Rack application server
 - continuous integration, 399
 - Decorator, 373
 - Rails basics, 51, 100
- RailsCasts, 238
- Rails concepts, *see also* Ruby on Rails basics
 - associations and foreign keys, **147**
 - caching, 407
 - controllers and views, **111**
 - databases and migrations, **104**
 - debugging, **117**
 - edit/update and destroy, **125**
 - form submission, **120**
 - models, **107**
 - MVC DRYness, **136**
 - overview, **100**
 - queries with reusable scopes, **157**
 - redirection and flash, **122**
 - RESTful routes for associations, **154**
 - security defenses, 418
 - SSO and third-party authentication, **142**
 - through-associations, **152**
- raise params.inspect, debugging, 119
- rake cucumber, 231
- rake jasmine, 192
- RASP (Read, Ask, Search, Post), debugging, 117
- Rational Unified Process (RUP)
 - design/code reviews, 348
 - overview, 22

- plan-and-document, 21
- tasks, 239
- RCS, *see* Revision Control System (RCS)
- RDBMS, *see* Relational database management system (RDBMS)
- RDoc, 301
- Rebasing
 - definition, 343
 - Git, 337
- Receiver
 - methods on collections, 86
 - Ruby objects, 74
- Rectangle
 - duck typing, 378
 - example, 376
 - LSP, 376
- Red–Green–Refactor
 - final task, 267
 - overview, **258**
 - TDD, 261
- Redirection
 - example, 140
 - Rails app example, **122**
- Red–Yellow–Green analysis,
 - Cucumber user stories, 226
- Reengineering, 322
- Refactoring
 - change points, 300
 - clean code example, 318
 - code smells, SOFA, 311
 - continuous, **325**
 - definition, 299, 314
 - examples, 315, 326
 - Extract Class, 370
 - language choice, 319
 - method-level, **313**
 - resisting enhancements, 324
 - TMDb API key, 279
- Reflection
 - conciseness, 113
 - productivity improvement, 16
 - Ruby objects, 74
- regexps, *see* Regular expressions (regexps)
- Regression testing
 - maintenance, 320
 - purpose, 14
- Regular expressions (regexps)
 - Cucumber, 224
 - Ruby overview, 70, 72
- Relational algebra formalism, 57
- Relational database, storage
 - structure, 53
- Relational database management system (RDBMS)
 - active record architectural pattern, 58
 - design, 57
 - example table, 57
 - Rails app example, 104
 - scalability, 64
- Release branch
 - definition, 340
 - example, 341
- Release management
 - definition, 351, 395
 - plan-and-document, 419
- Relevant, SMART user stories, 219
- Reliability
 - overview, **424**
 - plan-and-document, 420
- Reloading, Rails app example, 103
- Remote, git hosting services, 447
- Remote collaboration, Git, 339
- Remote Performance Monitoring (RPM), 405
- Rendering, caching, **407**
- Repetition, SOLID, 365
- Replication, database scaling, 426
- Reply, client-server architecture, 42
- Repositories
 - adding files, 453
 - Git basics, 446
 - remote collaboration, 339
 - version control, 336
- Representational State Transfer (REST)
 - AJAX, 186
 - Amiko, 379
 - API, **256**
 - associations, **154**, 157
 - caching, 408, 423
 - controllers, 111, 112, 114
 - CRUD, 102, 103, 125
 - CSRF, 416
 - form submission, 120, 121
 - Index action, 123
 - JSON APIs, 199
 - metaprogramming, 113
 - overview, **59**
 - same-origin policy, 202
 - session[] hash overstuffing, 130
 - SOA, 61
 - SOAP/WS-* comparison, 62
 - SRP, 369
 - template views, 63
- TMDb, 278
- TMDb API, 257
- Request, *see also* HTTP request
 - AJAX, 186
 - client-server architecture, 42
- Request-reply protocol, HTTP, 47
- Request tracing, 425
- require, confusion with include, 93
- Requirements basics, *see also* Non-functional requirements
 - Agile cost estimation, **222**
 - BDD and user stories, **216**
 - Cucumber and Capybara, **224**, **226**
 - Lo-Fi interface sketches and storyboards, **227**
 - plan-and-document, **238**
 - points, velocity, Pivotal Tracker, **220**
 - Rotten Potatoes enhancement, **230**
 - scenario comparisons, **236**
 - SMART user stories, **219**
- Requirements creep, 244
- Requirements traceability, 244
- require 'ruby-debug,' 119
- Resource
 - URI, 45
 - Web app as, 59
- response, controller specs and refactoring, 265
- Responsiveness
 - definition, 395
 - Google, 397
 - quantification, **396**
- REST, *see* Representational State Transfer (REST)
- RESTful APIs
 - developer keys, 257
- Reuse
 - productivity improvement, 16
 - Rails, 6
- ReviewsController, associations, 154
- Revision Control System (RCS), 355
- RightScale, 397
- Risk analysis, plan-and-document, 244
- Risk management, plan-and-document, 244
- Ritchie, Dennis, 40
- Rivest, Ronald, 414
- Rolling reboot, 404
- Root class, Ruby objects, 74

- Root zone, 45
 - Rotten Potatoes
 - AJAX Jasmine specs, 195
 - associations, 147
 - BDD, 217
 - client-server architecture, 42
 - commit example, 340
 - Cucumber scenario, 224, 224
 - DIP, 378
 - DOM, 177
 - enhancement, **230**
 - form submission, 121
 - JavaScript features, 167
 - jQuery example, 182
 - Lo-Fi UI, 228
 - migrations, 400
 - movie creation, 60
 - Rails basics, 100
 - records, 57
 - release branch example, 341
 - routes, 60
 - RSpec, 259
 - scenario comparisons, 236
 - storyboard, 228
 - third-party authentication, 142
 - tiers, 51
 - views, 54
 - Routes
 - associations, **154**
 - examples, 60
 - :format, 62
 - helpers via metaprogramming, 113
 - nested, 155
 - non-resource-based, 103
 - overview, **59**
 - Rails basics, 102
 - wildcard tokens, 104
 - RPM, *see* Remote Performance Monitoring (RPM)
 - RSA algorithm, 414, 415
 - RSpec
 - automation, 17
 - bug testing, 344
 - example, 260
 - example development, 262
 - function, 270
 - integration tests, 259
 - JavaScript and AJAX testing, 190
 - method listing, 280, 281
 - Red–Green–Refactor TDD, 264
 - response, 265
 - setting up, 191
 - testing tools relationship, 248
 - ruby-debug, 260
 - ruby-debug19, 102
 - Rubygems, 100, **256**
 - RubyMine, 445
 - Ruby on Rails basics, *see also*
 - Rails concepts
 - advantages, 5
 - AOP support, 141
 - blocks, **84**
 - classes, methods, inheritance, **78**
 - debugger, 123
 - dynamic language features, 271
 - FIRST, TDD, Red–Green–Refactor, **258**
 - GoF design patterns, 363
 - IDEs, 445
 - idiomatic language, **93**
 - iterators via yield, **89**
 - Java programmers, 92
 - JavaScript, **169**, 170
 - metaprogramming, **81**, 372
 - mix-ins and duck typing, **88**
 - objects, **73**
 - operation as method, **75**
 - origins, 66
 - overview, **70**
 - scalability, 64
 - Smalltalk origins, 432
 - Rumbaugh, James, 366
 - RuntimeError, TMDb, 273, 276, 278
 - RUP, *see* Rational Unified Process (RUP)
- ## S
- S0 coverage, 282
 - S1 coverage, 282
 - SaaS, *see* Software as a Service (SaaS)
 - SaaS on Cloud Computing, Virtuous Triangle of Engineering SaaS, 33, 433
 - Salesforce, 399
 - Same-origin policy, 202
 - SAMOSAS, team management, 347
 - sandbox method, Jasmine, 197
 - Sandwich integration, 286
 - Sanitization, Haml, 112
 - Sass, 451
 - save
 - associations, 153
 - Rails app example, 108
 - validation, 138
 - save!
 - associations, 153
 - Rails app example, 109
 - validation, 138
 - Scalability
 - cloud computing performance, 422
 - definition, 395
 - Rails, 64
 - RDBMS, 64
 - SaaS demands, 11
 - SCCS, *see* Source Code Control System (SCCS)
 - Scenario outlines, 238
 - Scenarios
 - Cucumber, 224, 224
 - requirements elicitation, 239
 - Scheduling
 - change management, 244
 - plan-and-document, 242
 - software projects, 245
 - Scheme
 - JavaScript origins, 166
 - URI, 45
 - SCM, *see* Software configuration management (SCM)
 - Scopes
 - filter criteria, 158
 - query composition, **157**
 - Scoping, Pivotal Labs Agile cost estimation, 223
 - Scratch branch, legacy code exploration, 301
 - Scripting languages, productivity improvement, 16
 - Scrum, **332**, xv
 - ScrumMaster, 333
 - Seam, **261**
 - definition, 263
 - dependency replacement, **313**
 - language variations, 265
 - search_terms, Red–Green–Refactor TDD, 262
 - search_tmdb
 - code sample, 268
 - controller specs and refactoring, 265
 - Red–Green–Refactor TDD, 261
 - specs, 268
 - subject code, 263
 - TDD, 259
 - Secret, encryption, 415
 - Secret key, git hosting services, 447
 - Secure HTTP protocol (HTTPS)

- Bookware VM on EC2, 444
- networking explanation, 45
- secure platform fallacy, 423
- Secure Sockets Layer (SSL)
 - data encryption, 415
 - decorators, 375
- Security
 - customer data protection, **414**
 - overview, **424**
 - plan-and-document, 420
 - Rails defenses, 418
 - secure platform fallacy, 423
- Seeding, 109
- Selector notations, CSS, 48, 49
- Self
 - class method definition, 80
 - future-proofing example, 81
 - JavaScript prototype
 - inheritance, 176
 - time arithmetic, 82
- Sencha Touch, 207
- send_email, 375, 379
- Send method, 75
- Separator, Ruby blocks, 84
- Serializing, 57, 173, 199
- Server certificate, authentication, 142
- Server push, client pull
 - comparison, 47
- Server-side applications,
 - JavaScript, 166, 167
- Service level agreement (SLA), 397
- Service level objective (SLO), 397, 397
- Service Oriented Architecture (SOA)
 - associations, 157
 - basic concept, 7
 - bookstore service, 8, 9
 - continuous integration, 399
 - designing for, **130**
 - Facebook Platform, 7
 - :format in routes, 62
 - REST, 59, 61
 - server performance, 423
 - SPAs, 166
 - Transform View, 64
- Session
 - cookies, 46
 - RESTful association routes, 157
- session[]
 - associations, 154
 - authentication, 145
 - characteristics, 125
 - overstuffing, 129
- SessionsController, OmniAuth
 - gem, 144
- Severity 1 bugs, Amazon, 344
- SGML, *see* Standard Generalized Markup Language (SGML)
- SHA-1 algorithm, 446
- Shamir, Adi, 414
- Sharding
 - database scaling, 426
 - RDBMS scalability, 64
- Shared-nothing architecture, three-tier, 52
- Shared-repository, version control, 336
- Short-lived code, 13
- Shotgun problem solving, 118
- Shotgun surgery, 372
- should, RSpec, 270
- should_not, TMDb, 265
- should_receive
 - Red-Green-Refactor TDD, 263
 - TMDb, 266
- show
 - AJAX, 186
 - controller method, 114
 - Rails app example, 111
- Show, caching, 409
- Siloed software
 - bookstore service, 8, 8
 - SOA comparison, 7
- SimpleCov, C0 coverage check, 283
- Simple Object Access Protocol (SOAP), REST/WS-*
 - comparison, 62
- Simula, 74, 134
- Sinatra, DataMapper, 151
- Single-page applications (SPAs), 166, **198**, 202
- Single Responsibility Principle (SRP), **369**
- Single sign-on (SSO)
 - Facebook, 142
 - side effects, 146
 - third-party authentication, **142**, 142
- Sketches
 - Lo-Fi UI, **227**
 - storyboard necessity, 246
- SLA, *see* Service level agreement (SLA)
- Slashdot, hits per day, 422
- SLO, *see* Service level objective (SLO)
- Slots, client-side JavaScript, 169
- Smalltalk, 74, 432
- Smart fuzzing, 285
- SMART user stories, **219**
- SOA, *see* Service Oriented Architecture (SOA)
- SOAP, *see* Simple Object Access Protocol (SOAP)
- SOFA
 - definition, 311
 - method length, 312
 - overview, **309**
- Software as a Service (SaaS)
 - basic concept, **4**
 - beautiful *vs.* legacy code, **12**
 - cloud computing, **10**
 - frameworks and languages, 5
 - important structures, 43
 - IT demands, 10
 - plan-and-document, **18**
 - productivity, **15**
 - SOA, **7**
 - software engineering *vs.* programming, **33**
 - testing, **13**
- Software configuration
 - management (SCM), 446
- Software Craftsmanship
 - movement, 365
- Software development processes
 - Agile Manifesto, **24**
 - plan-and-document, **18**
- Software education, weaknesses, xiv
- Software engineering, coining of term, 18
- Software evolution, causes, 12
- Software projects
 - budget surveys, 24
 - management basics, 355
 - scheduling issues, 245
- Software rejuvenation, 404
- Software Requirements
 - Specification (SRS)
 - IEEE Standard 830-1998, 240
 - plan-and-document, 239
 - plan-and-document and testing, 286
- Software Wall of Shame, 24
- SOLID
 - acronym variations, 363
 - basic concept, **362**
 - definition, 365
 - design guidelines, 365
 - GoF design patterns, 364

- ISP, 371
- OCP, 371
- SRP, 369
- sort method, Enumerable, 88
- Source code control, 446
- Source Code Control System (SCCS), 355
- Spaceship operator, 88
- SPAs, Single-page applications (SPAs), 166
- Specific, SMART user stories, 219
- Spiral lifecycle
 - Agile lifecycle, 436
 - design/code reviews, 348
 - overview, 20
 - plan-and-document, 19
 - productivity comparisons, 28
 - RUP, 21
 - tasks, 239
 - testing comparisons, 27
- Sprint, Scrum, 332
- spyOn, 192, 194
- SQL, *see* Structured Query Language (SQL)
- SQL injection
 - characteristics, 416
 - code example, 416
 - fuzz testing, 286
 - niche apps, 424
 - Rails app example, 108
- SQLite3, Rails basics, 100, 104
- Square
 - duck typing, 378
 - LSP, 376
- SRP, *see* Single Responsibility Principle (SRP)
- SRS, *see* Software Requirements Specification (SRS)
- ssh-keygen, 415
- SSL, *see* Secure Sockets Layer (SSL)
- SSL certificate, 415
- SSO, *see* Single sign-on (SSO)
- Staging site, 400
- Stallman, Richard, 444, 445
- Standard Generalized Markup Language (SGML), 48
- Stateless protocol
 - HTTP, 46
 - three-tier architecture, 52
- Statement coverage (C0)
 - check with SimpleCov, 283
 - definition, 282
- Static variable, Java *vs.* Ruby, 79
- Step definitions

- Cucumber, 224
- Cucumber and Capybara, 226
- domain language, 236
- Storyboards
 - Lo-Fi UI, **227**
 - Rotten Potatoes, 228, 231
 - sketches without, 246
 - TMDb UI, 232
- Strategy, OmniAuth gem, 143
- Strategy pattern
 - OCP, 371
 - OmniAuth, 373
- Stress, continuous integration, 399
- Stress testing, monitoring, 406
- String
 - JavaScript, 206
 - symbol comparison, 103
 - symbol interchangeability, 93
- String class
 - Ruby objects, 73
 - type casting, 75
- String literals, JavaScript, 206
- Stripe, 417
- Structural patterns, GoF, 364
- Structure diagrams, UML, 367
- Structured processes, 21
- Structured Query Language (SQL)
 - associations, 148
 - EXPLAIN, 414
- Structured storage, 57
- stub
 - example, 194
 - Jasmine, 192
 - TMDb, **265**
- Stubbing
 - Internet, AJAX, 202
 - Internet for TMDb, **276**
 - testing issues, 290
- Style sheets, programming
 - languages, 333
- Subject code, Red–Green–Refactor TDD, 263
- submit, JavaScript, 183
- Subversion, 336, 355, 446
- success handler function, 202
- Symantec, 417
- Symbol
 - Ruby overview, 70
 - string comparison, 103
 - string interchangeability, 93
- Symmetric-key cryptography, 415
- Syntactic sugar, method calls, 75, **76**
- Syntax errors, debugging, 119

- Synthesis, productivity
 - improvement, 16
- System building, 351
- System testing, software, 14

T

- Tables, RDBMS, 57, 57
- Table scan, databases, 412
- Task-parallel programming, event-driven comparison, 190
- TCP/IP, *see* Transmission Control Protocol/Internet Protocol (TCP/IP)
- TDD, *see* Test-driven development (TDD)
- Teams
 - dividing workload, 354
 - overview, **354**
 - plan-and-document, 347
 - Scrum, 333
- Teardown,RSpec, 264
- Template Method pattern
 - delegation, 373
 - OCP, 371
- Template View
 - alternatives, 64
 - index RESTful action, 112
 - MVC, 54
 - overview, **62**
 - web app comparison, 55
- Test coverage
 - completeness fallacies, 290
 - definition, 14
 - full-coverage validity, 290
- Test-driven development (TDD)
 - Agile, 25, xv
 - bugs, 344
 - continuous refactoring, 325
 - conventional debugging
 - comparison, **291**
 - JavaScript, 172
 - JavaScript and AJAX testing, 190
 - legacy code, 300
 - overview, 256, **258**
 - Red–Green–Refactor, 261, 261
 - TMDb, 231, 234, **272**
 - user stories, 217
- Testing
 - Agile *vs.* plan-and-document, 27
 - approaches and terminology, **284**
 - before/after coding, 291
 - bugs, 344

- client-side form validation, *197*
- exhaustive, *14*
- Google, *249*
- JavaScript and AJAX, **190**
- method comparison, *284*
- minimum requirements, *290*
- over-stubbing issues, *290*
- pending specs, *308*
- plan-and-document perspective, **286**
- plan-and-document vs. Agile, *289*
- QA, **13**
- reliance on type, *290*
- security, *421*
- tools relationship, *248*
- verification/validation
 - approaches, *14*
- Then, Cucumber keyword, *224*
- The Open Movie Database (TMDb)
 - add movie example, *233*
 - API, **256**
 - code example, *234*
 - DRY code, *235*
 - expectations, mocks, stubs, setup, teardown, **265**
 - FIRST, TDD, RSpec, **258**
 - fixtures and factories, **269**
 - Haml code, *233*
 - implicit requirements, **272**
 - stubbing the Internet, **276**
 - TDD cycle, **261**
 - UI storyboard, *232*
 - user stories/Lo-Fi UI example, *230*
- Therac-25, *4*
- Third-party authentication, SSO, **142, 142**
- 37 signals, *66*
- this keyword
 - incorrect use, *204*
 - JavaScript, *174, 182*
- Thompson, Ken, *40*
- Three-tier architecture
 - caching, *407*
 - cloud computing performance, *422*
 - overview, **51, 52, 53**
- Through-associations, **152, 152**
- Throughput, responsiveness, *396*
- Tiger team, *421*
- Timeboxed, SMART user stories, *219*
- Time class

- arithmetic operations, *81, 82*
- duck typing, *89*
- Rails app example, *108*
- timeout handler function, *202*
- Timeouts, performance protection, *424*
- TimeSetter, *307, 316*
- TLS, *see* Transport Layer Security (TLS)
- TMDb, *see* The Open Movie Database (TMDb)
- Tom Knight and the Lisp Machine, *118*
- Top-down integration, *286*
- Torvalds, Linus, *444, 446*
- Tracked files, Git basics, *446*
- Transform View, template view
 - alternatives, *64*
- Transition, RUP phase, *21*
- Transmission Control Protocol/Internet Protocol (TCP/IP)
 - communication, *44*
 - early work, *44*
 - port numbers, *45*
- Transport Layer Security (TLS), *415*
- trigger, JavaScript custom events, *185*
- TurboTax Online, *5, 432*
- Turing Award
 - Adleman, Leonard, *414*
 - Allen, Frances, *440*
 - Backus, John, *16*
 - Brooks, Fred, Jr., *19, 330*
 - Cerf, Vinton E. "Vint," *44*
 - Codd, Edgar F. "Ted," *57*
 - Dijkstra, Edsger W., *287*
 - Gray, Jim, *68*
 - Hoare, Charles Antony Richard, *98*
 - Kahan, William, *360*
 - Kahn, Bob, *44*
 - Kay, Alan, *432*
 - Knuth, Donald, *254*
 - Lampson, Butler, *296*
 - Liskov, Barbara, *392*
 - McCarthy, John, *10*
 - Nygaard, Kristen, *134*
 - Perlman, Alan, *164*
 - Ritchie, Dennis, *40*
 - Rivest, Ronald, *414*
 - Shamir, Adi, *414*
 - Thompson, Ken, *40*
 - Wilkes, Maurice, Sir, *2*

- Wirth, Niklaus, *214*
- Twitter
 - authentication, *142, 144*
 - Big Brother Bird, *425*
 - OAuth, *142*
 - OmniAuth, *145*
 - pair programming, *335*
 - Rails applications, *6*
- Two-pizza teams, **332, 336**
- Type casting, method calls, *75*

U

- UI, *see* User interface (UI)
- UML, *see* Unified Modeling Language (UML)
- UMPLE, *368*
- Undeclared exceptions, *275*
- Undefined method, Ruby objects, *74*
- Undisciplined process, Agile as, *25*
- Unified Modeling Language (UML)
 - ActiveRecord vs. ActiveRecord, *151*
 - associations, *147*
 - class diagram, *367, 371, 377*
 - diagram descriptions, *367*
 - legacy code exploration, *302, 303*
 - overview, **366**
 - usage decisions, *368*
 - use case diagram, *218*
- Uniform Resource Identifier (URI)
 - AJAX XHR, *187, 188*
 - before-filters, *156*
 - CSRF, *416*
 - definition, *45*
 - FakeWeb, *278*
 - fuzz testing, *286*
 - httpperf, *406*
 - HTTP request, **45, 59**
 - nested routes, *155*
 - params hash, *104*
 - post-receive, GitHub, *399*
 - Rails app example, *114, 126*
 - Rails basics, *100, 102*
 - route helpers, *113*
 - routes, *60*
 - TMDb API key, *257*
 - URL comparison, *45*
- Uniform Resource Locators (URL)

- AJAX XHR, *187*
- Jasmine, *196*
- URI comparison, *45*
- Unit testing
 - bugs, *344*
 - characterization tests, *306*
 - Cucumber, *226*
 - integration test comparison, **282**
 - software, *14*
 - testing method comparison, *284*
- Universal clients, web browser, *42*
- Unix
 - coursework skills, *445*
 - fuzz testing, *285*
 - inventors, *40*
 - Linux origins, *443*
 - time representation, *82*
 - tools on OS X/Windows, *447*
- Unmarshalling, *57*
- Unobtrusiveness, JavaScript, *167*
- update
 - JavaScript custom events, *185*
 - Rails app example, **125**, *126*
 - replacement for, *138*
- update_attributes
 - Rails app example, *109*
 - validation, *138*
- update_attributes!, validation, *138*
- URI, *see* Uniform Resource Identifier (URI)
- URL, *see* Uniform Resource Locators (URL)
- Use-case analysis, *218*
- Use case diagram
 - definition, *218*
 - UML, *367*
- Use cases, requirements
 - elicitation, *239*
- User interface (UI), Lo-Fi UI, **227**
- User stories
 - Agile, *25*
 - BDD pros and cons, *249*
 - case analysis, *218*
 - Cucumber and Capybara, **224**, **226**
 - overview, **216**
 - plan-and-document, **238**
 - points, velocity, Pivotal Tracker, **220**
 - scenario comparisons, *236*
 - SMART user stories, **219**
 - TMDb example, *230*
- US National Vulnerabilities Database, *426*
- Utility computing, *11*

V

- Validation
 - action substitution, *138*
 - ActiveModel, predefined, *137*
 - associations, *153*
 - BDD, *217*
 - controller/view interactions, *139*
 - IEEE 1012-2012, *350*
 - plan-and-document, *349*
 - software quality, *14*
- var, JavaScript, *205*
- Variable naming
 - guidelines, *312*
 - issues, *93*
- VBScript, *417*
- VCSs, *see* Version control systems (VCSs)
- Velocity
 - Agile, *25*
 - automation, *17*
 - BDD pros and cons, *249*
 - plan-and-document, *238*
 - user stories, **220**
- Verification
 - BDD, *217*
 - IEEE 1012-2012, *350*
 - plan-and-document, *349*
 - software quality, *14*
- VeriSign, *415*
- Version compatibility, continuous integration, *399*
- Version control
 - Git basics, **446**
 - history, **354**
 - merge conflicts, **336**
- Version control systems (VCSs)
 - automation, *17*
 - early VCSs, *336*
 - Git basics, *446*
- Version management, *351*
- vi, *445*
- Views
 - example development, *262*
 - fat, *129*
 - MVC, *54*
 - Rails app example, **111**
 - validation interactions, *139*
- vim, *445*
- VirtualBox, **443**
- Virtual Case File, *4*
- Virtual machines (VM)
 - bookware image, *442*, **443**
 - client-server architecture, *42*

- datacenters, *11*
- Virtual Private Server (VPS), *394*
- Virtuous Triangle of Engineering SaaS, *33*, *433*
- Viscosity, SOLID, *365*
- VM, *see* Virtual machines (VM)
- VM networking
 - NAT, *444*
 - resetting, *453*
- Voucher class, CRC card, *305*
- VPS, *see* Virtual Private Server (VPS)

W

- W3C, *see* World Wide Web Consortium (W3C)
- Warehouse Scale Computers, *11*
- Waterfall lifecycle
 - Agile lifecycle, *436*
 - design/code reviews, *348*
 - plan-and-document, *19*
 - productivity comparisons, *28*
 - RUP, *21*
 - Spiral lifecycle, *20*
 - tasks, *239*
 - testing comparisons, *27*
- Web application frameworks
 - comparison, *55*
 - databases, *53*
 - Fielding's work, *59*
 - three-tier architecture, *51*
- Web browsers
 - AJAX effectiveness, *202*
 - client-server architecture, *42*
 - communication, *44*
 - continuous integration, *399*
 - cookies, *46*
 - Cucumber and Capybara, *226*
 - HEAD, *60*
 - JASPI, *168*
 - JavaScript event handlers, *181*
 - Rails basics, *100*
 - REST, *59*
 - Scheme, *166*
- WebCL, JavaScript, *207*
- Webdriver
 - Capybara, *225*
 - testing tools relationship, *248*
- WEBrick, *42*
- Rails app example, *104*
- three-tier architecture, *51*
- Web servers
 - communication, *44*
 - networking explanation, *45*

- three-tier architecture, 51
- WebSockets, client pull *vs.* server push, 47
- WebWorkers, 190
- When
 - Cucumber keyword, 224
 - imperative *vs.* declarative scenarios, 236
- where method, Rails app example, 108
- White-box tests
 - definition, 285
 - software, 14
- Whole-system tests, testing
 - method comparison, 284
- Wildcards
 - regular expressions, 70
 - routes, 104
- Wilkes, Maurice, Sir, 2
- window, JavaScript, 177, 181
- Windows
 - Unix tools on, 447
 - VirtualBox, 443
- Windows 95, release celebration, 19
- Wirth, Niklaus, 214
- Word processor, code editing, 445
- Workflows, RUP, 21
- Working software
 - fix *vs.* redesign, 324
 - maintenance, 320
- World Wide Web Consortium (W3C), HTTP/HTML development, 66

X

- XHR, *see* XMLHttpRequest (XHR)
- XHTML, *see* eXtended HyperText Markup Language (XHTML)
- XMing, 444
- XML, *see* eXtensible Markup Language (XML)
- XML Builder, blocks and metaprogramming, 87
- XmlHttpRequest (XHR), 185, 187
- XP, *see* Extreme Programming (XP)
- XQuartz, 444
- X server, 444
- XSS, security issues, 417
- X Window System, 444

Y

- Yahoo
 - Mojito, 207
 - web portal, 42
- YAML, *see* Yet Another Markup Language (YAML)
- Yegge, Steve, 7
- Yet Another Markup Language (YAML)
 - fixtures, 270, 271
 - structured data, 173
- Yield
 - around-filters, 140
 - example, 90
 - iterators from, **89**
 - Rails reuse, 6
 - Ruby *vs.* operating systems, 90
- Y2K bug, 258

Z

- Z, requirements documentation, 241