Index

A

Accounting for change, 180, 224, 245
Accounting for rework, 224, 245–246
Activity Based Costing (ABC), 26
Adaptive behavior, emergence of, 109
Agile management theory and roles, 109, 185
Agile Manifesto principle, 12
Agile methods
  agility, defining, 293
  applicability of, 291
  business benefit of, 155–159
  expedite, as ability to, 293–294
  maturity progression, 297
  problem domain verus process map, 291
  process space, division of, 291–293
  scale *versus* ability to expedite, 294–295
  statistical process control, and, 295–296
  transferable quality improvement, 297–300
Agile software production metrics, 49
Analysis maturity continuum, 280–282
Anticipated ROI, use of, 111
Archetypes, 181
Artisan skills, 298, 299
Attributing value to a release, 152–153
Average Cost per Function (ACPF), 23, 54
Average Investment per Function (AIPF), 75
Average Investment per Function Point (AIPFP), 178
Average Revenue per User per Month (ARPU), 150

B

Batch size, role of, 88–89, 204
Blocked inventory, 68
Bottleneck
  addition of staff to a, 119–120
  failure at unit test, 82
  in Feature Driven Development, 210
  identification of, 79–81
  at system test, 83–84
Brooks' Law, 38, 274–275
Budget buffers, 209
Budget constraints, 41–42, 209
Buffers, role of, 66–68, 98–99, 197–198, 207–209, 217–218

C

Capacity constrained resource (CCR), 34, 146
Cash cost per user per month (CCPU), 150
Chaos theory, 10–11, 44, 278, 280
Chief Programmer Work Packages (CPWP), 191–192, 197, 203–204
Classification phase, 7
Code inspections, effect of, 86
Code reuse, 271
Coding process time, 165–166
Collective ownership approach, 239
"Common cause" variation, 43

Complex Adaptive Systems, 11
Conceptual learning, 297–300
Continuous improvement, role of, 113
Control charts, 5, 278
Control states, 277–280
Convergent *versus* divergent processes, 10
Correlation phase, 7–8
Cost accounting approach, 25–26, 141
Cost benefit of quality, 87
Cost control, 25
Cost efficiency, 25
Cost of change curve, 246–249
Critical chain representations, 70, 196, 208
Critical path, 64–65, 70, 216–217
Cumulative delays, role of, 218
Cumulative flow
  bottleneck, use in identification of, 79–80
  in Extreme Programing (XP), 227
  in Feature Driven Development (FDD), 194
  monitoring, 90–93
  tracking inventory with, 53, 60–61

D

Data management, 184
Defined *versus* empirical processes, 9–10
Delivery date, protection of, 63, 68
Design by feature set (step 4), 183
Design process time, 165
Developer resource constraint, 202
Development manager, 73–74, 77
Divergent *versus* convergent processes, 10
Domain Neutral Component (DNC), 181, 211–212, 279, 283
Drum beat, role of the, 95
Drum-Buffer-Rope, 4, 68, 95

E

Early finish, failure to report, 218
Early start, 65, 66
EBITDA equation, 150–151
Edge of Chaos state, 278, 280
Effect-cause-effect phase, 8
Effort-based metrics, 50
Effort tracking, 56–57
Emergence, 11–12
Empirical *versus* defined processes, 9–10, 277–278
End-to-end traceability, 113
Enterprise Resource Planning (ERP). *See* Manufacturing Resource Planning (MRP)
Epics, 226
Estimated cost of project delivery (D), 23
Evaporating clouds diagram, 272–273
Expected return on investment, calculation of, 24
Expediting requirements, effect of, 99–101
Exploitation considerations and implications, 30–31, 201
Extreme Programming (XP)
  accounting for charge, in, 245
  accounting for rework, 245–246
  collective ownership approach, 239
  continuous integration, use of, 235
  cost of change curve, 246–249
  cumulative flow diagram, 227
  epics, 226
  expediting, and, 294
  financial metrics, 243–249
  goals of, 225
  integration testing, use of, 235–236
  inventory, 225–226, 243
  inventory cap, 229
  inventory tracking, 227–228
  investment, 229, 243–244
  lead time, 228, 284
  net profit equation in, 244
  on-site customer, role of the, 240
  operating expense in, 244
  option theory, 234
  pair programming, use of, 236–237
  pipelining, 229
  planning game, 234
  prioritization of stories, 234
  process step time, 228
  production metrics, 225–231, 288
  production rate, 227
  quality, focus on, 284
  raw material, 225
  refactoring, use of, 230, 239, 245–246
  regular work week, use of, 240
  return on investment in, 245
  risk philosophy, 229
  S-Curve, 248–249
  senior management, for, 230–231
  specialists, elimination of, 240
  stand-up meeting, use of the, 238
  statistical process control, and, 295–296
  story points, 225
  system representation, 226
  tasks, 226
  test driven development (TDD), 238
  testing, 229

theory of constraints, denial of, 241
throughput, 227, 244
transferable quality improvement, 298–300
unit testing, 238
user stories, assessment of, 233–234

F

Failure
at integration test, 82, 83, 86
at product test, 84, 85
at system test, 87
on system test, 86
at unit test, 81–82
at user acceptance test, 84–85
Failure tolerant organization, role of the, 114
"Fair Process," 32
Feature definition, 184–185
Feature Driven Development (FDD)
accounting for change, 224
accounting for rework, 224
adaptation mechanisms, 189
agile management theory, and, 185
batch size, role of, 204
bottleneck, test, 210
budget buffers and constraints, 209
buffers, role of, 197–198, 207–209, 209, 217–218
build batches, 191–192
build by chief programmer work package (Step 5), 183
Chief Programmer Work Packages (CPWP), 191–192, 197, 203–204
critical chain representation, 196, 208
critical path, maintaining the, 216–217
cumulative delays, role of, 218
cumulative flow diagram, 194
data management (DM), 184
dependencies, role of, 218
design by feature set (step 4), 183
developer resource constraint, 202
domain neutral component, 211–212
early finish, failure to report, 218
estimates *versus* agreed function, 194–195
executive management metrics, 200
exploitation of engineering resources, 201
feature definition, 184–185
feature lifecycle, 193
feature list (Step 2), 182
feature process steps, 196
feature sets, 191–192
feature team, use of, 202–203
file/class access constraint, 202
financial metrics, 221–224
five-point scale, 187–188
formulae, 188–189
inventory, 212–213, 221, 283–284
inventory cap, role of, 209
investment, 221–222
knowledge management system, use of, 199
lead time, 185
level of effort (LOE), estimation of, 186–188
local safety problem, 219
modeling and analysis phase, 210–212
modeling rule of thumb, 186
morning roll call, use of, 213–215
multitasking, role of, 218
operating expense, 186, 222–223
overview feature, 181–182
peer reviews, use of, 210
PERT charts, 195–196
plan by subject area (Step 3), 183
prioritized feature lists, use of, 204–206
problem domain (business logic), 184
process control, 193–194
process steps, 185
process time, 185
production metrics, 182–189, 287–288
project parking lot, 198
quality, focus on, 283–284
queue time, 185
return on investment, 224
safety constraints, 217–218
scheduling subject areas and feature sets, 195–197
scope constraint, 204–205
S-Curve effect, and, 215–216
self-organizing construction within planned assembly, 191
setup time, 185, 203
shape modeling, 182, 216
"student syndrome," 217–218
subject areas, 193
surgical team, use of, 202–203
system representation, 182–183
systems interfaces (SI), 184
ten percent (10%) rule, 219–220
Threshold state, and, 279–280
throughput, 223
time constraints, 207–209, 216–217
time-modified prioritized feature lists, use of, 205–206
transferable quality improvement, 298–300

user interface feature sets, 192
user interface (UI), 184
value-added, 223
variance, focus on, 283
visual control, 199
wait time, 185
workflow, 197–198
Feature lifecycle, 193
Feature sets, 191–192
Feature team, use of, 202–203
Feedback, role of, 11
Feeding buffers, 66
File/class access constraint, 202
Financial metrics
in Extreme Programing (XP), 243–249
in Feature Driven Development (FDD), 221–224
for general business systems, 21
for software development systems, 22
for software services, 149–154
Fire fighting, 298, 299
Five-point scale, 187–188
Flow, identification of, 77–78
Foreseen uncertainty, 43
Framework, development of, 24–25
Functional priority, 40
Function Point (FP), 164, 177, 289

G

General Accepted Accounting Practices (GAAP), 27
General business systems, 21
Governing rules
adaptive behavior, emergence of, 109
anticipated ROI, use of, 111
continuous improvement, role of, 113
end-to-end traceability, 113
engineers, for, 115
failure tolerant organization, role of the, 114
management roles and rules, 109, 112
maturity increases, as, 111–112
process improvement problem, 110
production metrics, 111, 113
Reinersten's Three Tiers of Control, 109–110
skills gap education, 113
team measurements, 115
"Green" status, 68
Group inventory for convenient testing, 64

I

Ideal State, 278, 279
Idleness, role of, 32–33
Input, value of, 58
Integration testing, 79, 86, 87, 235–236
Intellectual efficiency, 88, 166
Inventory
in Extreme Programing (XP), 225–226, 243
in Feature Driven Development (FDD), 221, 283–284
group inventory for convenient testing, 64
importance of, 27
logical collections of, 63–64
in Scrum, 252–253
in Software Development Lifecycle (SDLC), 164
tracking the flow of, 60–61
in traditional methods, 177
unified development process (UDP), 172
Inventory cap
in Extreme Programing (XP), 229
in Feature Driven Development (FDD), 209
in Scrum, 255
in Software Development Lifecycle (SDLC), 165
in Unified Development Process (UDP), 173
Inventory tracking, 227–228, 254
Investment
in Extreme Programing (XP), 229, 243–244
in Feature Driven Development (FDD), 221–222
reductions in, 134
in Scrum, 255
in Software Development Lifecycle (SDLC), 164
in traditional methods, 177–178
in Unified Development Process (UDP), 172
Investment value, throughput accounting tracking of, 58–59
ISO-9000, 5
Issue log, 68
IT department, agile management in the
budget basis for IT assessment, calculation of, 133
budget basis for measuring the IT organization, 131–132
investment, reductions in, 134
lead time, reduction of, 135
operating expenses, reductions in, 135
throughput, improvements in, 134
true basis, calculation of, 132–133
value-added by a bank lending system, potential definitions for, 132
value-added contribution, corporate IT's, 131–132
Iterative incremental process, 173, 174

J

J-Curve effect, 38, 274–276
JIT. *See* Just-in-Time Inventory
Just Enough Design Initially (JEDI), 6, 283

K

Kanban Approach, 4, 6
Knowledge Management System (KMS), use of, 93, 199
Koskela & Howell's three-dimensional model for project management, 57

L

Labor, implications of adding additional, 274–276
Late start, 65–66
"Late" status, 68
Lead time (LT)
  estimation of, 63
  in Extreme Programing (XP), 228
  in Feature Driven Development (FDD), 185
  reduction of, 135
  in Scrum, 254, 285
  in Software Development Lifecycle (SDLC), 164
  and software production metrics, 53
  in Unified Development Process (UDP), 173
Lean production, 5–6, 284
Learning Organization Maturity Model
  goals of, 105
  Stage 0-Analysis Ability, 105
  Stage 4- Anticipated ROI and the Failure Tolerant Organization, 107
  Stage 1-End-to-End Traceability, 106
  Stage 2-Stabilize System Metrics, 106
  Stage 3- Systems Thinking and a Learning Organization, 106–107
Level-of-effort (LOE) estimate, 50, 186–188
Lifecycle methods, software engineering, 18
Lifetime revenue per subscriber (LRPS), 150
Line of code (LOC), 50
Little's Law, 53
Local safety considerations, 44–46, 219

M

Management accounting for systems
  complex development systems, 18–20
  emergent properties, 14
  operating expenses (OE), 15
  systems thinking, and, 14–15
  throughput accounting, and, 15–17
  value added, and, 16
Management roles and rules, 73–76, 109, 112
Manufacturing Resource Planning (MRP), 95
Marketing Requirement Document (MRD), 171
Maturity progression, 297
Morning roll call, use of, 213–215
Multitasking, role of, 218

N

Net profit, 21–22, 24, 179, 244
Net profit for services (NPBITDA), calculation of, 152

O

Object Oriented Analysis, 8
Object Oriented Software Engineering (OOSE), 289
Offshore development and process maturity, 121–122
One-dimensional model of project management, 57
On-going investment, role of, 142
On-site customer, role of the, 240
"On Time" status, 68
Operating expense (OE)
  in Extreme Programing (XP), 244
  factors of, 146
  in Feature Driven Development (FDD), 222–223
  importance of, 27
  operating expense for services (OEBIDA), calculation of, 151
  reductions in, 135
  in traditional methods, 178–179
Operational learning, 297–300
Operationally validated theories, 298
Operations review
  attendees, 123
  financial information, presentation of, 124–125
  information, presentation of, 124–128
  minute taking, 128
  production metrics, presentation of, 125–126
  program management metrics, presentation of, 127
  project management metrics, presentation of, 127
  purpose of, 123
  timing, 124
Option theory, 234
Outsourcing decisions, 120–122
Overtime, effectiveness of, 81

P

Pair programming, use of, 236–237
Parallel paths, definition and identification of, 64–65
Peer reviews, use of, 210
People constraint, protecting the, 37–38
Perishable requirements, 32

PERT charts, 64, 69, 195–196
Pipelining, 229, 256
Plan by subject area (Step 3), 183
Planning game, 234
PMI models for project management, 55
Predictions of profitability, 23–24
Prioritized feature lists, use of, 204–206
Problem domain, 184, 291
Process improvement, role of, 110, 138
Process lead time, elements of, 88
Process map, problem domain *versus*, 291
Process maturity, improvements in, 282–283, 285
Product backlog, 251, 259
Production efficiency, 26
Production metrics
  in Extreme Programing (XP), 225–231
  in Feature Driven Development (FDD), 182–189, 287–288
  governing rules, 113
  in Scrum, 251–2562, 288–289
Production quantity, measuring, 52
Production rate (R)
  in Extreme Programing (XP), 227
  governing rules, 111
  representation, 79
  in Scrum, 253
  in Software Development Lifecycle (SDLC), 165
  in Unified Development Process (UDP), 173
Product line strategy, 74
Product management, agile
  cost accounting for Software Product Development, 141
  management accounting, 138
  on-going investment, role of, 142
  operating expense, factors of, 146
  process improvement, role of, 138
  product mix, role of, 142–148
  sales and throughput, calculation of, 137–138
  scope constraint, management of, 143–144
  throughput accounting approach, 138–142
  time-based throughput model, appropriateness of the, 140
  traditional cost accounting approach, 138
Product manager, 74–75
Product mix
  constraints, and, 146–148
  effect on investment, and, 146
  revenue is the goal, when, 144–145
  role of, 142–143, 154
Profitability, predictions of, 24
Profit by service release, calculation of, 153
Program manager, 74
Project buffer, 63, 119–120
Project delivery date, protection of, 63
Project manager's new work, development of, 59–60
Project parking lot, 198
"Project" status, 68
  "late" status, 68
  "on time" status, 68
  "red" status, 68
  "watch" status, 68
  "yellow" status, 68

Q

QA. *See* Quality assurance, importance of
Quality assurance, importance of, 5, 6, 81, 86–88, 277
Queue growth, 79
Queue time, 88, 175, 185

R

Rapid Application Development (RAD)
  inventory cap in, 265
  lead time in, 266
  limitations of, 266–267
  operating expense in, 266
  principles of, 265
Raw material
  in Extreme Programing (XP), 225
  in Scrum, 252
  in Unified Development Process (UDP), 171
Recovery and stretch of software production constraints, 81
"Red" status, 68
Refactoring, 93, 230, 239, 245–246, 257
Regression effects, 85
Reinersten's Three Tiers of Control, 109–110
Release, 260–261
Release backlog, 251, 259
Release manager, 74
Resource constraints, 42–43, 68–70
Return on Investment, 4, 21, 24, 152–153, 179, 224, 245
Rigorous Software Methodologies (RSM), 292, 296, 299
Risk philosophy, 229, 256
ROI. *See* Return on Investment
Roles *versus* functions, agile management, 76

S

Safety constraints, 217–218
Sales and throughput, calculation of, 137–138
Scheduling subject areas and feature sets, 195–197
Scientific development, phases of, 7–9
Scientific management paradigm, 56
Scope constraint, 40–41, 143–144, 204–205
Scrum
- cumulative flow, 253
- engineering practices, 263
- expediting policy, 255, 260, 285, 294
- goal commitment, 261
- goals of, 251
- inventory, 252–254
- inventory cap, 255
- investment, 255
- lead time, 254, 285
- meeting, daily, 261
- pipelining, 256
- process step time, 255
- product backlog, 251, 259
- production metrics, 251–257, 288–289
- production rate, 253
- products, 251
- raw material, 252
- refactoring, 257
- release, 251, 260–261
- release backlog, 251, 259
- review process, 263
- risk philosophy, 256
- Scrum Master, 259
- senior management metrics, 257
- sprint backlog, 251, 259
- sprint planning and project management, 254
- sprints, 251
- statistical process control, and, 295–296
- system representation, 252
- team size and composition, 261–262
- testing, 256
- thirty day sprint, 260
- throughput, 253
- transferable quality improvement, 298–300
- working environment, 262–263

S-Curve, 90–93, 215–216, 248–249
SEI Software Capability Model, 105
Self-organizing construction within planned assembly, 191
Service business economics, 150
Setup time, 88, 175, 185, 203
Shape modeling, 182, 216
Six Sigma, 6
Skills gap education, 113
Software Development Lifecycle (SDLC)
- analysis process time, 165
- coding process time, 165–166
- design process time, 165
- Function Point (FP) metric, 164
- idleness, efficiency, and growing inventory levels, 170
- inventory, 164, 166–167
- inventory cap, 165
- investment, 164
- lead time, 164
- process step time, 165–166
- production rate, 165
- raw material, functional specification for, 163
- slack, lack of, 170
- specialists and high inventory levels, 169–170
- testing process time, 166
- throughput, 164–165
- uncertainty, role of, 168
- variance reduction and waste, 168–169
- waste and long lead times, 167

Software production metrics
- agile software production metrics, 49
- Average Cost Per Function (ACPF), 54
- effort-based metrics, 50
- expressions of inventory, 52
- inventory-based metrics, 49
- lead time (LT), and, 53
- level-of-effort estimate, 50
- measurement of inventory, 51–52
- nonfunctional requirements, 51
- OE per unit, 54
- production quantity, measuring, 52
- selection of, 49
- tracking inventory, 53
- traditional software production metrics, 50

Software Resource Planning
- buffers, role of, 98–99
- drum beat, role of the, 95
- expediting requirements, effect of, 99–101
- goals of, 95
- release of requirements into the systems, planning, 96
- starving a process, effect of, 97–98
- subordination of the CCR, 95
- swamping a process, effects of, 96–97

Theory of Constraints, and, 95
waste, cost and causes of, 101–103
Software services, financial metrics for
attributing value to a release, 152–153
average revenue per user per month (ARPU), 150
cash cost per user per month (CCPU), 150
definition of software service, 149
EBITDA equation, 150–151
lifetime revenue per subscriber (LRPS), 150
net profit for services (NPBITDA), calculation of, 152
operating expense for services (OEBIDA), calculation of, 151
product mix, role of, 154
profit by service release, calculation of, 153
return on investment for services, calculation of, 152
ROI by service release, calculation of, 153
service business economics, 150
throughput for service, calculation of, 150–151
uncertainty, role of, 154
SPC theory. *See* Statistical Process Control theory
Specialists
availability of, 69
elimination of, 240
versus generalists, use of, 272–274
high inventory levels, and, 169–170
Sprint backlog, 251, 259
Sprint planning and project management, 254
Sprints, 251
Staffing decisions
bottleneck, addition of staff to a, 119–120
conventional view of turnover costs, 117
full-time engineer, cost of replacing a, 118
loss of throughput on a constraint, 118–119
offshore development and process maturity, 121–122
outsourcing decisions, 120–122
project buffer, impact on, 119–120
temporary engineer, cost of replacing a, 118–119
throughput accounting view of turnover costs, 117
turnover, role of, 117
Stand-up meeting, use of the, 238
Starving a process, effect of, 97–98
Statistical Process Control Theory, 5, 277, 295–296
Story points, 225
"Student syndrome," 217–218
Subject areas, 193
Subject matter expert (SME), 221
Subordination, 31, 95
Surgical team, use of, 202–203
Swamping a process, effects of, 96–97
System Goal, 20
Systems interfaces (SI), 184
Systems thinking and learning organizations, 11
detail complexity, 15
general systems, 13–17
inherent complexity, 15
System testing, 79, 83–84, 86, 87

T

Task planning, 56–57
Tasks, 226
Taylorism, 169
Taylor, Frederick Winslow, 169
scientific management, 56
Team measurements, 115
Team size and composition, 261–262
Ten percent (10%) rule, 219–220
Test driven development (TDD), 238
Theoretical comparison, 6–7
Theory of Constraints, 3–4, 6, 11, 29–34, 95, 241
Theory of Scientific Management, 25
Thirty day sprint, 260
Three-dimensional model for project management, 57
Threshold state, 278, 279–280
Throughput
in Extreme Programing (XP), 227, 244
in Feature Driven Development (FDD), 223
importance of, 27
improvements in, 134
increasing, 34
in Scrum, 253
in Software Development Lifecycle (SDLC), 164–165
in traditional methods, 179
in Unified Development Process (UDP), 173
Throughput accounting, 19–20, 21, 25–26, 117, 139–140, 141–142
Tick-IT, 5
Time-based throughput model, appropriateness of the, 140
Time constraints, 38–40, 207–209, 216–217
Time-modified prioritized feature lists, use of, 205–206
Total Quality Management, 5, 6
Toyota Production System, 4, 6
TQM. *See* Total Quality Management

Tracking metrics, agile project, 67–68
Traditional cost accounting approach, 138
Traditional metrics *versus* agile principles, 271, 289
Traditional project management, 55–56
Traditional software production metrics, 50
Transferable quality improvement, 297–300, 298–300
Transformation, stages of, 18–19
True basis, calculation of, 132–133
Trust, goal of establishing, 41
Turnover, role of, 117

U

Uncertainty
  aggregation of sequential and parallel processes, 46–47
  budget constraint, protecting the, 41–42
  chaos, 44
  classification of, 43–44
  foreseen uncertainty, 43
  local safety considerations, 44–46
  people constraint, protecting the, 37–38
  principle of, 11, 37
  resource constraints, protecting the, 42–43
  role of, 154, 168
  scope constraint, protecting the, 40–41
  time constraint, protecting the, 38–40
  unforseen uncertainty, 44
  variation, 43
Unforseen uncertainty, 44
Unified Development Process (UDP)
  agility, lack of, 175–176
  artifacts, 174
  documentation, 174
  inventory, 172
  inventory cap, 173
  investment phase, 172
  iterative incremental process, 173, 174
  lead time, 173
  process step time, 175
  process time, 175
  production rate, 173
  project management, 175
  queue time, 175
  raw material, 171
  setup time, 175
  throughput, 173
  use cases, 172
  vision document, 171
  wait time, 175
Unit testing, 82, 238
Unvalidated theories, 298, 299
Use cases, 172, 289–290
User interface feature sets, 192
User interface (UI), 184
User stories, assessment of, 233–234

V

Value-added
  by a bank lending system, potential definitions for, 132
  contribution to corporate IT's, 131–132
  cost accounting tracking of, 57–58
  in Feature Driven Development (FDD), 223
Value chain, software production in the, 25
Value efficiency, 26
Variance, 43, 168–169, 283
Vision document, 171
Visual control, 93–94, 199

W

Wait time, 89, 175, 185
Waste, 101–103, 167
"Watch" status, 68
"Waterfall" model for software production, 56, 161, 166–168. *See also* Software Development Lifecycle (SDLC)
Wheeler's four states of control, 278, 295–296
Working capital requirements, determination of, 23–24
Working code (Q), 23
Work-in-process (WIP) inventory, 65, 166

Y

"Yellow" status, 68