<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-BIT (automatic BIT), 219, 220</td>
</tr>
<tr>
<td>Above-specification test, 33</td>
</tr>
<tr>
<td>Abstract data types, as software design principle, 202</td>
</tr>
<tr>
<td>Abstraction</td>
</tr>
<tr>
<td>in interface design, 198</td>
</tr>
<tr>
<td>as software design principle, 203</td>
</tr>
<tr>
<td>Accelerated life testing (ALT), 169, 174</td>
</tr>
<tr>
<td>Accelerated reliability testing (ART), 169</td>
</tr>
<tr>
<td>Accelerated stresses, 34</td>
</tr>
<tr>
<td>Acceleration factors (α), 31–32, 169–170</td>
</tr>
<tr>
<td>“Acceptable hazard” system, 253</td>
</tr>
<tr>
<td>Accurate performance specifications, writing, 5</td>
</tr>
<tr>
<td>Across-media perceptions, 142</td>
</tr>
<tr>
<td>Actionable cause, 95</td>
</tr>
<tr>
<td>Action results, 85</td>
</tr>
<tr>
<td>Actions, developing, 84–85</td>
</tr>
<tr>
<td>Activation energy, 31</td>
</tr>
<tr>
<td>Actual failure mode, 77</td>
</tr>
<tr>
<td>Additive RPN approach, 82</td>
</tr>
<tr>
<td>Administrative applications, software fault tolerance for, 205</td>
</tr>
<tr>
<td>Adverse effects, mitigating, 38</td>
</tr>
<tr>
<td>Aerospace systems, failure tolerance in, 2</td>
</tr>
<tr>
<td>Aging tests, 33–34</td>
</tr>
<tr>
<td>Air traffic controllers, system rejuvenation for, 205–206</td>
</tr>
<tr>
<td>AIX operating system, 50</td>
</tr>
<tr>
<td>Alaska Highway, 184</td>
</tr>
<tr>
<td>Allocation of functions, 143</td>
</tr>
<tr>
<td>American Society for Quality (ASQ), 184</td>
</tr>
<tr>
<td>Analog circuit redesign, 71</td>
</tr>
<tr>
<td>Application domains, reusing software across, 201</td>
</tr>
<tr>
<td>Architecture, of software systems, 195</td>
</tr>
<tr>
<td>Architecture documentation, 107, 112, 113</td>
</tr>
<tr>
<td>Architecture FMECA, 73</td>
</tr>
<tr>
<td>Architecture-level reliability risk analysis, 108–109</td>
</tr>
<tr>
<td>Arrhenius model, 31</td>
</tr>
<tr>
<td>Assembly errors, 152</td>
</tr>
<tr>
<td>Assembly failure mode, 78</td>
</tr>
<tr>
<td>Assessment leader, in reliability capability assessment, 279</td>
</tr>
<tr>
<td>Assessment updates, 231</td>
</tr>
<tr>
<td>Asynchronous software aging, 46</td>
</tr>
<tr>
<td>Audits, Y2K software, 39</td>
</tr>
<tr>
<td>Austin, Steven S., xiii, 183</td>
</tr>
<tr>
<td>Automatic stress monitoring, sensors for, 220–221</td>
</tr>
<tr>
<td>Basic reliability models, 56, 57</td>
</tr>
<tr>
<td>Basili, Victor, 130</td>
</tr>
<tr>
<td>Bayesian techniques, 20</td>
</tr>
<tr>
<td>Benchmarking, of product defect rate, 126</td>
</tr>
<tr>
<td>Benefits</td>
</tr>
<tr>
<td>of configuration management, 250</td>
</tr>
<tr>
<td>of reliability programs, 227</td>
</tr>
<tr>
<td>of risk assessments, 236–237</td>
</tr>
<tr>
<td>Bernstein, Lawrence, xiii, 193</td>
</tr>
<tr>
<td>BIT capabilities, 218. See also Built-in test (BIT)</td>
</tr>
<tr>
<td>BIT circuitry, 219</td>
</tr>
<tr>
<td>BIT design, 218, 219</td>
</tr>
<tr>
<td>BIT effectiveness, 26, 72, 215</td>
</tr>
<tr>
<td>BIT signals, 218, 219</td>
</tr>
<tr>
<td>types of, 219–220</td>
</tr>
<tr>
<td>Black boxes, in reliability capability assessment, 277</td>
</tr>
<tr>
<td>Block diagrams, 20–23</td>
</tr>
<tr>
<td>functional, 73</td>
</tr>
<tr>
<td>in software design, 196</td>
</tr>
<tr>
<td>Boehm, Barry, 204</td>
</tr>
<tr>
<td>Boolean gates, 29</td>
</tr>
<tr>
<td>Boolean logic, 22</td>
</tr>
</tbody>
</table>
Boolean logic symbols, 96
Bottom-up design, 200–201
Bottom-up failure flow process, 77
Bottom-up system analysis, 74
Boundary levels, in safety design, 254
Box, George E. P., 19, 55, 64
Brainstorming/graphical capture systems, 134
Brainstorming process, 132
Braun-Paine model, 60
“Breadboard” functional checks, 231
Built-in test (BIT), 26, 215. See also BIT entries
Built-in test analyses, 26
Buzan, Tony, 132
California fires, 187
Capability Maturity Model (CMM), 49, 201, 275
Capability maturity model assessment results, 111
Capability Maturity Model Integrated
(CMMI® level 5-rated companies, 125
Capability Maturity Model Integration
(CMMI®, 275
Cascading FMECA, 105–106
Catastrophic severity, 258
Caterpillar, 11
C-BIT (continuous BIT), 219, 220
CCA failure flow diagram, 80. See also Circuit card assembly (CCA)
C compilers, 39
Cerf, Vinton, 204
Challenger catastrophe, 223–224
Change in configuration management, 250
risk handling during, 251
risk management and, 249
Change management, 235
Checklists, 154
new product, 280–281
for reliability, 196, 198
for reliable software design, 211
for reusable software components, 199–200
for trustworthiness, 194–195
Checkpoint technique, 205
Childs, Joseph A., xiii, 15, 87, 223
Chrysler, 2
Chunnel, 188
Circuit card assembly (CCA), 74. See also CCA failure flow diagram
Circuit performance understanding, 160
Clamping diodes, 164–165
C language, system trustworthiness and, 207
“Clean room” software design, 42–43
Closed modules, 199
CMOS devices, parameters derated for, 162–163
Code breakdowns, 42
Code changes, 44, 48
Code modularity, 45
Code reliability, 44–45
Coffin–Manson equation, 170, 173
Cohesive code, as software design principle, 201–202
Cold, designing for, 184–186
Cold Regions Test Center (CRTC), 185, 186
Cold spare, 73, 74
Collaborative tools, 127
Co-located teams, communication interfaces for, 229
Combinatory-explosive analysis, 41
Common-mode failures, treating as critical, 47
Common-mode variation, 127
Communication between customer and designer/supplier, 226–227
with management, 224
during product design, 228–229
Communications-based problems, 43
Compact focus, 127
Comparative safety analysis (CSA), in system safety design, 255
Compatibility issues, 2
Competitiveness, risk and, 249
Complete P-FMECA, 100. See also Process FMECA (P-FMECA)
Complexity, design risk and, 241–242
Complex systems, 2
inherent design reliability of, 62
Component failure mode, 78
Component failure rate models, 54
Component life, 31
Componentry, object-oriented programming vs., 199–200
Components concerns about, 138
failure rates for, 160
reliability prediction of, 165
as software modules, 196, 197
times to failure of, 160
Composite assembly failure rate, 19
Computer-aided software reliability estimation (CASRE) tool suite, 49
Computer communications, 39–40
Computer controllers, services and, 40
Computer system environment, software reliability and, 46
Computer systems, usability of, 141. See also System entries
Condition-based maintenance (CBM), 72, 213, 214, 215
time-based maintenance vs., 216–217
Condition-based maintenance plus (CBM+), 216, 220–221
Configuration management (CM), 249–251 defined, 250
Configuration parameters, 45
Consequences, in risk estimation, 245
Constant failure rate, 19, 20
Constant-failure-rate assumption, 23
Constant temperature cycling test, 170
Context, removal in software design, 203–204
Context diagram, 105
Continued stress analysis, 27
Continuous design reliability improvements, 60
Controls, in product design, 142
Cooper Industries, 2
Correct decisions, accumulating knowledge for, 4
Correction capability, 219
Corrective actions, 99
Corrective maintenance, 213, 214, 216
Cost
of configuration management, 250
of reliability programs, 227
requirements creep and, 238
of risk mitigation, 245
Cost benefits, of reliability programs, 227
Cost of failure, 233
Cost risk, 239
Cox-Lewis (Cozzolino) model, 60
Critical features, monitoring the health of, 12
Critical-function failures, 70
Criticality, of software failure modes, 108–109
Criticality analysis, 67, 68
Critical severity, 258
Critical software FMECA team brainstorming, 115–116
Critical system functions, designing-out failures of, 1–2
Critical test/evaluation mindset, 117
Crosby’s quality maturity grid, 274–275
Cross-functional communications, 127
Cross-functional meetings, 122
Cross-functional teams, 5 inadequate, 238–239
in P-FMECA, 90–91
Crow AMSAA model, 60
Customer loyalty, earning, 233
Customer requirements
developing, 130
of software FMECA, 105, 106
Customers
communicating reliability status to, 18–19
cost of losing, 10
in the development process, 18–19
reliability and, 226–227
role in software design, 196
Cycle time, speeding up, 129. See also Life-cycle entries
Damage modeling, 217, 218
Data analysis tools, 18
Databases, 281
“lessons learned,” 46
software fault tolerance for, 205
Data-driven approaches, 126
Data updates, 68
Date-related problems, exposure areas related to, 39
Date rollover problems, 47
Defect density model, 55
Defect density rate, 49, 50
Defect prevention process, 47
Defect rate, 125
Defects
corrective actions to prevent, 99
minimizing, 129
staff churn and, 207
Defensive/proactive design approach, 44
Defined level, 276–277
Delco Electronics (Delphi), 11
Denson, Bill, 56
Department of Defense (DOD), prognostics and health management capabilities for, 216, 220–221
Derating, 25
of CMOS device parameters, 162–163
described, 165
stress analysis and, 160–161
Derating methodologies, examples of, 162–165
Desert-proofing equipment, 186–187
Desert Storm operation, 186
Design analyses, 230
Communication interfaces for, 229
Design approaches, 44
Design capability, 271
Effect on reliability capability, 271–276
Design change(s)
Forms of, 70
Risk management and, 249
Design change priorities, stack-ranking of, 83–84
Design completeness, testing, 38
Design concepts, choosing, 9
Design consistency, 45
Design constraints, system trustworthiness via, 204–210
Design-corrective actions, assigning priorities for, 69
Design decisions, 88–89
Design engineers
Interest in human factors, 138
Role in decision making, 88
Role in human factors analysis, 144
Design errors, 151, 152
Designers
Checklists for, 154
Role in software design, 196
Design evaluation, 141
Design failure modes, effects, and criticality analysis (D-FMECA), 25–26, 68–71, 72, 87, 229. See also Design FMECA process; Failure modes, effects, and criticality analysis (FMECA) approaches to, 72–74
Benefits of, 89, 90
P-FMECA vs., 72
Purpose of, 68
Reliability programs and, 225
For software product design, 103–104
Design flaws, 69, 139
Finding, 6
Design FMECA process, example of, 74–81
Design for extreme environments, 183–191
Design for reliability (DFR), xv–xvi, 63, 157. See also Designing for reliability; DFR entries
Aim of, 1–2
Integrating with design for safety, 253–266
Paradigms for, 4–13
Stress analysis and, 159–160
Design for safety, 253
Integrating with design for reliability, 253–266
With reliability data, 262–265
Verification of, 261–262
Design for six sigma (DFSS) methods, xv, 121, 126, 129–130. See also DFSS entries; Software design for six sigma (SDFSS) methods
Design guides, 25
Design improvement, using FMECA methods, 86
Design-induced impediments, eliminating, 137
Designing for reliability, issues in, 3–4
Design margin, 3, 157, 161, 162, 170, 178–180, 226
Design of experiments (DOE), 122–123, 127–128
Early experiences with, 121–124
Six sigma process and, 123–124
Design operating point, optimizing, 129
Design parameters
Predicting reliability from, 54
Unspecified, 69
Design principles, 201–204
Design problems, requirements creep and, 238
Design process, for detailed safety design, 261
Design process–induced failure causes, 79
Design process methods, 18
Design process short cycling, 172–173
Design process weaknesses, 69
Design project team, 253
Design-related failures, 64, 71s
Causes of, 69
Design reliability, improving, 75–76
Design reviews, 230
Communication interfaces for, 229
Design review tools, 9–10
Design risk, 235, 241–242
Competitiveness and, 249
Expression of, 242
Design risk management, 241–242
Design rule violations, 78
Design simplification process, system trustworthiness via, 204, 211
Design stability checking, system trustworthiness via, 208, 211
Design tests, 18
Design to unit production costs (DTUPC) budgets, 220–221
Design verification phase, reliability tools in, 28–30
Design weaknesses, 171
identifying, 172
Destruction limits, 170, 177, 180
Detailed design, communication interfaces for, 228
Detailed design phase, reliability tools in, 27–28
Detailed safety design, design and development process for, 261
Detection, improvements related to, 115–116
Detection controls, 84
Detection (DET) factor, 96–97, 98, 99
Detection factor guideline examples, 94
Detection rating, 112–114
assessing, 113
updated, 116
Detection scores, calculating, 113
Deterministic failures, probabilistic failures vs., 159
Deterministic methods, probabilistic methods vs., 159
Development process for detailed safety design, 261
improving, 129
Development process prediction model, 50, 51
Development requirements, 150
DFSS phases, 130. See also Design for six sigma (DFSS) methods
DFSS process, 121
DFSS tools, 129–130, 134
Diagnostic capability, 47
Differing system views, 127
Digital circuit redesign, 71
Digital integrated circuit (IC), 74
Digital logic devices, 164–165
Diode clamping, 164–165
Displays, in product design, 142
“Dissatisfiers,” minimizing, 129
Divide-by-zero situations, 44
Dixon, Jack, xiii, 137, 235
DMADV (define, measure, analyze, design, verify) approach, 130
Documentation in reliability capability assessment, 276, 278
of reliability programs, 229, 231–232
in risk communication, 248
in verifying design for safety, 262
“Domain knowledge.” 43
of software artifacts, 117
Downtime analysis, 6
Duane growth model, 60
Durability analysis, 158
Dynamic modeling methods, 60
“Early life” failures, 76
Early program phase, reliability tools in, 19–27
Edison, Thomas Alva, 4, 185
Effective change management, operational reliability and, 48
Effective improvements, searching for, 99
Effectiveness analysis, 26
Effective software testing, 46
Effects analysis, 39
Eisenhower, Dwight D., 185
Electrical products, prognostics and health management capabilities for, 214–215
Electrical sensors, for automatic stress monitoring, 220
Electrical stress analysis, 157
Electrical stress values, 161
Electrical testing, 281
Electronic components, IEEE 1624–2008 in assessing, 269
Electronic part failures, 64
Electrostatic discharge (ESD) prevention, 30
End-of-life requirements, 226
Energy, use of, 185
Engineering communication interfaces for, 228
design risk and, 242
Engineering changes, 10
avoiding, 8
Engineering requirements, prognostics and health management capabilities and, 221
Engineering risks, 150
Enhanced fault tolerance, 116
Environmental factors, 184–185
Environmental loads, monitoring, 217, 218
Environmentally induced failure causes, 79
Environmental specifications, in requirements generating and analysis, 237
Environmental stresses, 158. See also Extreme environments
of HALT, 175–176
Environmental stressors, designing for, 3
Environmental stress screening (ESS), 34
Environmental testing, 32, 230–231
Equipment, desert-proofing, 186–187
Error containment, 152
Error detection, 72, 219
Error detection and correction (EDAC) capability, 47
Error messages, 153
Error monitoring, intelligent, 153
Error-producing conditions, 151
Errors, 41–42. See also Human error entries
software module behavior during, 195
Error tolerance, design for, 153
Error-tolerant systems, 153
Evaluators, in reliability capability assessment, 276, 279
Event tree analysis, in risk identification, 243, 244
Event trees, 244
Exception handling, 41, 235
Excessive power switching, 164
Execution, of reliability programs, 229, 230–231
Execution domain, software rejuvenation and, 205
Execution domain bounding, system trustworthiness via, 209, 211
Execution-time software reliability models, 48–49
Expected failure time limits, 20
Expected life, 9
Experiment design, early experiences with, 121–124
Extreme environmental conditions, testing for, 188
Extreme environments, 189
design for, 183–191
Extreme heat, 187
Extreme locations, designing for, 185–186
Extremely improbably hazards, 259
Extremely remote hazards, 259
Extremes, withstanding, 190

Fact-based decision-making process, six sigma process as, 128
Factoring, 209
system trustworthiness via, 209–210, 211
Factor interactions, 123
Factor of safety (FOS), 161, 170
Fail safely design, 8
Failure(s), 41, 42. See also System failure entries; Times to failure analyzing, 47
BIT capabilities and, 218–220
in case study, 281
configuration management and, 250
consequences of, 153
cost of, 7
defined, 40
defining in reliability programs, 226–227
due to poorly defined requirements, 237–238
feedback loop for, 28
listing causes of, 75
minimizing via prognostics capability, 213–215
parts-related, 56
path to, 41–42
predicting, 15
probability of, 22, 223–224
risk assessment and, 242–243
sources of, 5
studying and profiling, 47
system management, 43
Failure avoidance, 233
Failure cause codes, 64
Failure cause distribution, 80
Failure cause pareto diagram, 80, 81
Failure causes, 76, 78–80, 81
physical model categories of, 60–62
Failure correction, 232
Failure criticality, calculating, 68–69
Failure data, collecting, 28
Failure distribution, 64
Failure effects, 76
listing, 75
mitigating, 75
in P-FMECA, 92–93
reducing the severity of, 67
Failure elimination, stress analysis for, 157–167
Failure factors, reviewing, 99
Failure flow process, bottom-up, 77
Failure-free design, 13
Failure management capability, building, 84
Failure mechanism detection, acceleration of, 169–181
Failure mechanism models, 75
Failure mechanisms, xv, 76
acceleration of, 173–174
knowing, 32
Failure mode(s), 76
actual, 77
analyzing effects of, 106–107
assembly, 78
assessing, 67
causes of, 95–96, 109, 110
component, 78
designing out, 2–3
of hardware D-FMECA, 74
with a high severity number, 82
induced, 76
inherent, 76, 78
listing and numbering, 74
manufacturing process, 34, 78
mitigating the effects of, 114–115
part, 78
performing, 39
in P-FMECA, 92–93
potential, 77, 106
potential causes of, 95
prioritizing, 69
process-induced, 77
reappearance of, 29
severity rating for, 108–109
for software elements, 108
system-level, 78
time-related characteristics of, 76
Failure mode acceleration, 178
Failure mode and effects analysis (FMEA),
2–3, 11, 43–44, 67–68, 127, 134, 185, 258
Failure mode data, 77
Failure mode results, reporting, 100–101
Failure modes and effects analysis, 47
Failure modes, effects, and criticality analysis
(FMECA), 9, 11, 67, 68–69, 84, 85, 86, 87. See also Design failure modes,
effects, and criticality analysis
(D-FMECA); FMECA entries; Process FMECA (P-FMECA)
applied to software development, 103–104,
benefits of, 89
development and applications of, 103–104
goal of, 69
important aspect of, 90
multiple viewpoints on, 89
reasons for performing, 75
in risk identification, 243
in software development phases, 105
software elements for, 106, 107
value of, 68
Failure modes, effects, and criticality analysis
and maintainability analysis
(FMECA-MA), principles of, 71–72
Failure modes, failure mechanisms, and
failure causes list, 158
Failure occurrence rating, 111–112
Failure of commission, 40
Failure of omission, 40–41
Failure points, distribution of, 179
Failure precursors, 217
monitoring and reasoning of, 217–218
Failure prevention, 9
Failure probability, 162
Failure probability distribution function
(PDF), 31
Failure rate(s), 12
calculating, 19
for components, 160
constant, 19, 20
total, 19
Failure rate allocation, 64
Failure rate database, 165
Failure rate models, 54
Failure repair, 232
Failure reporting, 231–232
Failure reporting and corrective action system
(FRACAS), 28, 32, 64, 230, 231–232.
See also FRACAS entries
Failure review board (FRB), 28, 33. See also
FRACAS/FRB process/methodology
Failure scenarios, foreseeing, 4
Failure symptoms, 72
Failure time limits, 20
Failure tolerance, 1–2
Failure tracking (FT) practice, 268, 273–274
Failure tracking system, 281
Failure trends, 28–29
Fallback modes, 2
Fan-out rating, 164
Fast path processing, 209
Fatigue, 214
Fault activation, 50
Fault avoidance and removal procedure, 47
Fault detection, 218–220
Fault detection probability, 26, 72
Fault injection, 47
Fault injection experiments, 106
Fault isolation, 26–27, 218–220
probability of, 72
requirements for, 5–6
Fault latency, 51
Fault management, 84
Faults, 41, 42
identifying hardware, 174–175
Fault tolerance, 70
enhanced, 116
software reliability and, 195, 196
system trustworthiness via, 204–205, 211
trustworthiness and, 193
Fault tolerance capability, 72
Fault-tolerant architecture/design, 57
prognostics capability and, 214
Fault-tolerant design capabilities, 73
Fault tree analysis (FTA), 149
in risk identification, 243, 244
in system safety design, 257
Fault triggers, 41
Federal Aviation Administration (FAA), 205–206
Feedback, 26, 282
Feedback loop
for failures, 28
reliability program as, 232–233
Feynman, Richard, 223–224
Field failures, 281
Field-repairable units (FRUs), 219
50% derating, 25
50% design margin, 178–179
Finite element analysis (FEA), 158–159
Firmware failure, 38–39
Five-corner analysis, 62
500% return on investment, 8
Flexible Computer Interface Form, 198
FMECA analyst, 86. See also Failure modes, effects, and criticality analysis (FMECA)
FMECA engineers, 85
FMECA process, 117, 230. See also Process FMECA (P-FMECA)
FMECA series, 119
FMECA steps, for software development, 106–116
Follow-on work, in verifying design for safety, 261–262
Follow-through, for improvement actions, 100
Follow-up meetings, 85
Follow-up reports, 231–232
“Fool-proofing,” 230
Ford, Henry, 130
Formal inspections, software FMECA vs., 118
Four-corner analysis, 62
Four-corner tests, 170
Four-lives design, 9
Fowler, Martin, 209–210
FRACAS data, 75, 84. See also Failure reporting and corrective action system (FRACAS)
FRACAS/FRB process/methodology, 34. See also Failure review board (FRB) effectiveness of, 32
Frequent hazards, 259
Full requirements development, 46
Functional approach, to D-FMECA, 72–73
Functional block diagram, 73
Functional capability assessments (FCAs), organizational reliability capability and, 268
Functional D-FMECA, 73–74
Functions
allocation of, 143
defining, 13
Future uncertainty, assessing, 62
Generalization, as software design principle, 203
General Motors, 11
Globalization, xvi
Goal, question, and metric (GQM) tool, 127, 131–132. See also GQM measures
Goals
clarifying meaning of, 130
of reliability programs, 224–225
of risk assessment, 243
of system safety, 253–254
“Gold plating,” 204
GQM measures, 130. See also Goal, question, and metric (GQM) tool
Graphical capture/brainstorming systems, 134
Graph theory, 111
Group thinking, 132
HALT analyst, 177. See also Highly accelerated life testing (HALT)
HALT process, key points of, 171–172
HALT step stresses, 175
HALT stresses, 170
HALT test coverage analysis key points of, 174–175
test coverage of, 174
Handling errors, 152
Hard failure condition, 164, 177, 180
Hardware, identifying faults in, 174–175
Hardware approach, to D-FMECA, 73, 74
Hardware design, human-centered, 139
Hardware products, IEEE 1624–2008 in assessing, 269
Harris Corporation, 206
Harry, Mikel, 128
Harsh environments, 183
HAST stresses, 170. See also Highly accelerated stress testing (HAST)
Hatton, Les, 207
Hazard analysis, 148–149, 266
for specific products, 262–265
Hazard consequences, in risk estimation, 245
Hazard levels. See Risk levels
Hazard probability
design risk and, 242
in risk estimation, 244
Hazards ranking, 245–247, 259
risk communication and, 248, 249
in risk estimation, 247, 260
system safety design and, 255–257
Hazard severity
design risk and, 242
in risk assessment code matrix, 260–261
in risk estimation, 244
in system safety design, 257–260
Hazard worksheet definitions, 256
Hazard worksheets, 257–258, 266
Health monitor data, 214
Heat, designing for, 186–191. See also Thermal entries
Heat testing, 187–188
Heuristic risk assessment methodology, 109
Hierarchy system, in design for safety, 254
High-level design documentation, 107, 112, 113
High-level design FMECA, 105. See also Failure modes, effects, and criticality analysis (FMECA)
Highly accelerated life testing (HALT), 33–34, 169–181. See also HALT entries
design margin in, 178–180
environmental stresses of, 175–176
negative aspects of, 172
sample size for, 180
sensitivity to stress in, 176–178
test coverage in, 174–175
time compression in, 173–174
usefulness of, 180
wide acceptance of, 172–173
Highly accelerated stress audit (HASA) profiles, 172
Highly accelerated stress screening (HASS), 34
Highly accelerated stress screen (HASS) profiles, 172
Highly accelerated stress testing (HAST), 169. See also HAST stresses
High reliability achieving, 121–122
advantages of, 282–283
High-reliability lasers, 121–122
High risk, 236, 245, 247, 266
in risk assessment code matrix, 260
High-temperature operating limit, 179
Historical data, root-cause-related, 112–113
Historical failure data, 111
Hobbs, Gregg, 169
Hot spare, 73, 74
Human behavior, factors that influence, 138
Human-centered design, 138–144
considerations in, 139, 140
Human-centered design paradigm, implementing, 141
Human-centered hardware design, 139
Human-centered software design, 139–142
Human error, 30, 138, 150–153
error-tolerant systems and, 153
mitigation of, 152–153
technology and, 151
types of, 151–152
Human error rate prediction technique, 147
Human factors
design engineer’s interest in, 138
in reliable design, 137–155
role in design, 139
Human factors analysis, 144–149
methods of, 144
purpose of, 144
Human factors analysis tools, 145–149
Human factors design checklist, 154
Human factors engineering (HFE), 137–138 defined, 137
Human factors requirements, 150
Human factors specialists, role of, 142
Human factors validation, 154
Human–machine interface, 142–143
Human-made molecules, 184
Human performance requirements, 154
Human reliability analysis, 146–147
Humans
cold weather survival and, 185–186
under extreme conditions, 188
Human systems integration, risk-based approach to, 150
Humidity, as a stress condition, 170
Hyundai, 2
I-BIT (initiated BIT), 219, 220
Ideas, associating, 132
IEC 61709 international standard, 165–166
IEEE 1332–1998, reliability objectives in, 272
IEEE 1413.1, 63, 76
IEEE 1624–2008, 278
organizational reliability capability assessment and, 267–268, 269–270
IEEE 1624 scoring guidelines, 276–277
IEEE 1633–2008, 55
IEEE 15288–2008, 270
IEEE Design for Reliability (DFR) Committee, xvii
IEEE Reliability Society future initiatives, 132
IEEE Reliability Society Standards, 225
IEEE Spectrum, 40
Immediate cause, 95
Importance levels, in design for safety, 253–254
Important features, impact of, 13
Improbable hazards, 259
Improved Performance Research Integration Tool (IMPRINT), 145
“Improved training,” 99
Improvement actions, in P-FMECA, 98–100
Improvements
follow-through in, 100
searching for effective, 99
Inadequate cross-functional teams, 238–239
Include files, 202
Incorrectly calibrated instruments, 78
Increased design strength, 178
Increasing reliability/lowering cost paradox, 9–11
Incremental cost reductions (ICRs), 283
Independent facilitator, 116–117
Induced failure mode, 76
Infant crib cage, hazard analysis for, 263
Information, in risk communication, 248
Information hiding
in interface design, 199
as software design principle, 202
Inherent design flaws, 69
Inherent design reliability, of complex systems, 62
Inherent failure causes, 79
Inherent failure mode, 76, 78
Inherent failure rate, 57
Inherent reliability, estimating, 54
Inheritance, as software design principle, 202
Innovation, attention on, 9
Input variables, effects of, 122
Inspection(s)
as corrective action, 99
software FMECA vs., 118
in verifying design for safety, 262
Inspection errors, 152
Installation errors, 152
Instruments, incorrectly calibrated, 78
Integral equations, finite element analysis and, 158
Integrated circuits (ICs), 74, 164
Intelligent error monitoring, 153
Interactive problems, 123
Interconnectedness, 6
Interface design, 199
Interface languages, in interface design, 199
Interfaces
designing, 237
in detailed safety design, 261
information hiding and, 202
during product design, 228–229
reliability programs and, 225
in risk communication, 248
in safety design, 254–255
software modules and, 197–199
software refactoring and, 210
software reliability and, 195, 196
Intermittent operation, 177
Interoperability problems, 43
Iraqi Freedom operation, 186
Ishikawa fishbone charts/diagrams, 28, 30
Isolation analyses, 27
IT (information technology) services, critical, 40
Job safety analysis, 147
Jump table, 197
Juran, Joseph M., 98, 232
Keene, Samuel, xiii, 37, 44, 56, 121
Keene development process prediction model, 50, 51, 111
Kepner–Tregoe analysis, 127
Key process input variables (KPIVs), 122
Key process output variables (KPOVs), 121–122
KSLOC (thousand software/source lines of code), 49, 50
Language feature limitation, system trustworthiness via, 207, 211
Language of managers, 232–233
“Language of money,” 232
Languages, in interface design, 199. See also C language
Laptop computers, 164
Laser performance, variables influencing, 122
Latent defects, 41. See also Fault latency
Latent manufacturing flaws, designing to avoid, 11–12
Lean management, 4–5
Lessons-learned approach, for organizations, 119
“Lessons learned” database, 46
Level F probability, 245
Levels of development, in reliability capability assessments, 276–277
Life, 31
Life-cycle application conditions, 280–281
Life-cycle costs, 6, 282
calculating, 10
measuring reliability by, 7
minimizing, 129
Life-cycle engineering concepts, applications of, xvi
Life cycles, 270
Life-sustaining equipment, 184
Life testing, 31–32
Line replaceable units (LRUs), 219
failures of, 80, 81
Link analysis, 148
Liskov substitution principle, in interface design, 199
Load case conditions, 169
Load cases, 175–176
Location research, 187
Logic, underlying probability calculations, 22
Logical block diagram, in software design, 196. See also Block diagrams
Logistic reliability growth models, 60
Looping test routines, 177
Low-level design documentation, 107, 112, 113
Low-level design FMECA, 105. See also Failure modes, effects, and criticality analysis (FMECA)
Low risk, 236, 245, 247, 266
in risk assessment code matrix, 261
Machine–human interface, 142–143
Mainline code, 42
Maintainability analysis, 72
Maintainer-induced failure causes, 79–80
Maintenance
in detailed safety design, 261
improving design reliability via, 213
Maintenance errors, 152
“Malfunction 54,” 42
Managed level, 276, 277
Management. See also Project management
communicating reliability status to, 18
communication interfaces for, 228, 229
communication with, 224
presenting recommendations to, 99
reliability and, 233
of a reliability program, 224–225
Management support, for P-FMECA, 90
Managers, language of, 232–233
Manufacturing, cost of, 184
Manufacturing defects, uncovering, 34
Manufacturing features, identifying, 11
Manufacturing flaws controlling, 12
designing to avoid, 11–12
Manufacturing-induced component failures, 60
Manufacturing processes, failure modes in, 34
Manufacturing process failure mode, 78
Manufacturing process–induced failure causes, 79
Marginal severity, 258
Markov chain methods, 62
Markov models, 62–64
Martian probe failure, 43
Martin, Greg, 206
Master–master configuration, 70
Master–slave configuration, 70
Materials, 23, 24
Mathematical graph theory, 111
Mature organizations, organizational reliability capability and, 268
Maturity-level metrics, 274–276
McCabe method, 111
McCabe subtree, 109, 110
McWilliams model, 60
Mean management, 4–5
“Means–ends abstraction” view, 105
Mean time between failures (MTBF), 1, 20, 23, 60–61, 170, 171
Mean-time-between-failures models, 54
Mean time to failure (MTTF), 171
Mechanical components, IEEE 1624–2008 in assessing, 269
Mechanical loading problems, finite element analysis and, 158
Mechanical products, prognostics and health management capabilities for, 214–215
Mechanical sensors, for automatic stress monitoring, 220
Mechanical stress analysis, 157, 158
Mechanical stress values, 161
Mechanical testing, 281
Medium risk, 236, 245, 247, 266 in risk assessment code matrix, 261
Memory initialization, system trustworthiness via, 208, 211
Memory leakage, 166
Memory leak detection tools, 208
Message logs, software reliability and, 196
Metrics, for clarifying and answering questions, 131
MIL-HDBK-61, on configuration management, 250
MIL-HDBK-217 military handbook, 54
MIL-HDBK-338 military handbook, 165
Military applications, cold weather and, 185, 186
Military electronic systems, 54
Military reliability tests, 171
MIL-STD-756B, 56
MIL-STD-882, 247
MIL-STD-1829, 25
Mind mapping tool, 132–135
Mind maps, 127, 132, 133 building, 132–134
Miner’s criteria, 173
Miscommunications, 43
Missing functions, identifying, 5, 6
Mission-aborting failures, 63
Mission-critical failure, 57, 216
Mission failures, 53
Mission reliability models, 56–57
Mistake-proofing methods, 115
Mitigation, improvements related to, 114–115
Mitigation cost, in risk estimation, 245
Modeling, in reliability programs, 227–228
Modeling methodology, 53
Modular system architecture, 45
Module interface language, 198
Modules design principles for, 201–204 reuse of, 200–201 in software, 195, 196–200
Module size limitation, system trustworthiness via, 208, 211
Module vs. release matrix, 197
Molecules, human-made, 184
Monitoring in condition-based maintenance, 217 of environmental and usage loads in damage modeling, 218 of failure precursors, 217–218 sensors for automatic stress, 220–221
Monte Carlo analysis, finite element analysis and, 159
Monte Carlo models, 62
Moriarty, Brian, xiii, 253
Motorola, six sigma approach at, 124
Mount Saint Helens eruption, 187
m-out-of-n redundant configuration, 21
“m out of n” system, 70
Multiperspective software FMECA team, 106–107. See also Failure modes, effects, and criticality analysis (FMECA)
Murphy’s law, 243
NASA research, 191
National Aeronautics and Space Administration (NASA), 43, 188, 201
Challenger catastrophe and, 223–224 technological advances and, 183–184
National Institute of Standards and Technology (NIST), trustworthiness and, 193–194
Natural gas, exploring for, 189
Navigability, 127
Navigational software, 41
Negligible severity, 258
Neumann, Peter, 12
New design optimization, 129
New Orleans seawall levees, 190
New product checklists, 280–281
New technology, software reuse and, 201
“No doubt” mindset, 2
Nominal fault densities, 49, 50
Nonconstant failure rates, 23
Nondestructive limit, 33
Nonhomogeneous Poisson Process (NHPP) models, 60
Nonperiodic software execution, software rejuvenation and, 205
Non-repairable models, 56
Non-standard environments, 183
Nuclear plants, 188
hazard analysis for primary power supply in, 265

Object-oriented programming componentry vs., 199–200
software reuse and, 200–201
Occasional hazards, 259
Occurrence (OCC) factors, 95–96, 97, 98, 99
Occurrence ratings surrogate measures for establishing, 111–112
updated, 115
Off-nominal inputs, 44
One-factor-at-a-time (OFAT) problem, 123
100% design margin, 8, 9, 178
One-time cost reductions (OCRs), 283
On-site assessment phase, in reliability capability assessment, 276, 278, 279
Open and closed principle, in interface design, 199
Open dialogue sessions, 67
Open modules, 199
Open source development, 200
Operating and support hazard analysis (O&SHA), in system safety design, 257
Operating high-temperature requirement, 179
Operating margin, 178
Operating risks, 150
Operational process capability, 49
Operational reliability, effective change management and, 48
Operational stresses, 158
Operation Desert Storm, 186
Operation Iraqi Freedom, 186
Operator errors, 151
Operator-induced failure causes, 79–80
Optimum module size, 208
Organizational reliability capability, 270–271
Organizational reliability capability assessment (ORCA), 267–284
advantages of high reliability and, 282–283
benefits of, 283–284
case studies of, 279–282
design capability and performability and, 271–276
goals of, 267–269
IEEE 1624 scoring guidelines for, 276–277
organizational reliability capability in, 270–271
process of, 278–282
reliability capability assessment and, 271
SEI CMMI scoring guidelines for, 277
Organizations, lessons-learned approach for, 119
Original equipment manufacturers (OEMs), organizational reliability capability and, 267, 268, 270
Out-of-specification weaknesses, 34
Outsourcing, xvi
Overhead of indirection, 209
Overstress condition, 178, 179
Pair programming, 134
Parallel devices, 21
Parallel RBD, 56, 57. See also Reliability block diagrams (RBDs)
Parallel RBD model, 58
Parametric value, for power dissipation, 163
Pareto chart/diagram, 77, 83–84
failure cause, 80, 81
“Parking lot” practice, 91–92
Parnas, David, 195, 197
Parnas design checklist, 196, 198
Parsing, 47
Part failure modes, 74, 78
Partial differential equations, finite element analysis and, 158
Part, process, or material (PPM) guidelines, 230
Part-related failures, 64
Parts decisions related to, 23–24
reliability of, 57, 58, 59
Parts flow, 24
Parts-related failure, 56
Parts stress analysis method, 160
Part stress failure rate model, 54
Patent defects, 41
Patent failure, 180
Peak loads, design for, 9
Peer reviews, 266
    software FMECA vs., 118
Perceptions, across-media, 142
Perfection tolerance, 190–191
Performability, 271
Performance, achieving world-class, 282
Performance-based logistics (PBL), 216
Performance data, 55, 64
Performance risk, 239, 240
Performance specifications, writing, 5
Performance trending, 217
Periodic software execution, software
    rejuvenation and, 205
Personnel. See also Hiring good people
    selection and training of, 143–144
    system trustworthiness via talented, 207, 211
P-FMECA form, 88. See also Process
    FMECA (P-FMECA)
P-FMECA process, steps in, 88
P-FMECA processes/subprocesses, defining, 92
P-FMECA training, 89
Phone outages, 48
Physical model categories, of failure causes,
    60–62
Physics of failure (PoF) modeling, 20, 54
Planning
    for an extreme environment, 183
    in P-FMECA, 91
    of reliability programs, 229
    software modules and, 197
Plastic
    safety factor of, 190
    testing, 184
Poke-Yoke method, 115
Populations, expanding, 185
Potential failure modes, 77, 106
    identifying, 106
Potential issues, prioritization of, 98
Potential risks, 38
Power dissipation ($P$), 162–163, 164
    parametric value for, 163
Power grid, failure of, 153
Power Law model, 60
Power standby mode, 164
Power supply design, 163–164
Power switching, excessive, 164
Power system failures, 37–38
Precedence, in risk mitigation, 247, 260. See
    also Safety order of precedence
Preconditions, removal of context and,
    203–204
Prediction results, 20
Predictive fault trees, 20–23
Preferred parts list, 24
Preflight checklist, 154
Preliminary hazard analysis (PHA)
    in risk identification, 243
    in system safety design, 255
Prevention, improvements related to, 115
Preventive maintenance, 213, 216
    condition-based vs. time-based, 216–217
Previous systems, in design for safety, 254
Priorities
    in hazard assessment, 258–260
    ranking, 68
Prioritization, of potential issues, 98
PRISM model, 56
Proactive/defensive design approach, 44
Proactive level, 276, 277
Probabilistic failures, deterministic failures
    vs., 159
Probabilistic methods, deterministic methods
    vs., 159
Probabilistic model for reliability, 22
Probabilistic pattern failures, 69, 71
Probability
    design risk and, 242
    of failure, 22
    of fault detection, 26
    in quantitative risk analysis, 236
    risk assessment and, 243
    in risk estimation, 244
    in risk mitigation, 247, 260
Probability calculations, logic underlying, 22
Probability definitions, 259, 260
Probability distribution function (PDF), 31
Probability information, 266
Probability levels, of hazards, 258–260
Probability of detection (PD), 82, 83
Probability of failure, 162
Probability of failure occurrence (PFO), 82
Probable hazards, 259
Problem Reports (PRs), in verifying design
    for safety, 261
Procedures, in risk mitigation, 247, 260
Process capability, six sigma approach for
    measuring, 126
Process defect rate, importance of sigma level
    for, 125, 126
Process flowchart/flow diagram, 91, 278
See also Failure modes, effects, and criticality analysis (FMECA); FMECA process; P-FMECA entries
benefits of, 89–90
cross-functional team in, 90–91
D-FMECA vs., 72
failure modes and effects in, 92–93
improvement actions in, 98–100
key support considerations for, 90–91
management support for, 90
as a motivation tool, 90
performing, 91–98
planning in, 91
principles of, 87–88
process stakeholders and, 89
for software process, 103–104
team preparation in, 91–92
use of, 88–90
Process improvement, six sigma approach for, 124
Process-induced failure modes, 77
Process modeling, 30
Processor assignments, 70
Process robustness, 130
six sigma approach for measuring, 126
Process selection, 23, 24
Process sigma (σ). See also Sigma (σ); Six sigma entries
defect rate as function of, 126
practical implications of, 126
Process stakeholders, P-FMECA and, 89
Procurement assurance, communication interfaces for, 228
Product attributes, in configuration management, 250
Product concept perspectives, 6–7
Product configuration, in configuration management, 250
Product credibility, 190
Product defect rate, benchmarking, 126
Product “delighters,” 129
Product design, 7
consideration for humans in, 138
displays and controls in, 142
feedback in, 232
key interfaces during, 228–229
risk communication in, 248
role of human factors in, 139
Product designers (PDs), organizational reliability capability and, 267, 268, 270
Product design life
role of reliability engineering in, 17–18
tools used in, 17
Product deterioration, prognostics capability and, 214
Product development, xvi
Product enhancements, 129, 189
Product-failure levels, reducing, 124
Product failures, xv
process-related, 88–89
Product feedback, 26
Product FMECA, 105. See also Failure modes, effects, and criticality analysis (FMECA)
Product friendliness/modularity, 6
Product improvement, 144
Production and field support, 32–34
Production engineering, communication interfaces for, 228
Production issues, 230
Production testing, 33–34
Product life, reliability and, 226–229
Product life cycles, 270
reliability tools in, 16–17
Product life tasks, 16, 17
Product maintenance, facilitating, 129
Product manufacturers (PMs), organizational reliability capability and, 267, 268, 270
Product migration, 129
Product modeling, in reliability programs, 227–228
Product producibility, 27
Product reliability, HALT and, 180
Product requirements, developing and understanding, 130
Products
ability to perform, 15
designing reliability into, xv
hazard analysis for specific, 262–265
safety rating of, 190
Product sensitivity, to stress conditions, 176–178
Product understandability, 127
Product variation, minimizing, 129
Prognostics, fault detection and isolation and, 218–220
Prognostics and health management (PHM) capabilities, 72, 213–215
automatic stress monitoring and, 220–221
BIT types and, 219–220
as Department of Defense policy, 216, 220–221
improving reliability via, 213–221
Prognostics and health management (PHM) capabilities (Continued)
as part of condition-based maintenance, 217
Prognostics health monitoring, design for, 12–13
Prognostics-type warnings, 8
“Program ABC,” 215
Program changes, 48
Program life tasks, 16, 17
Program logic, 39
Program manager, risk communication to, 248
Programming consistencies problem, 38
Program requirements development, 39, 43
Program risk, 235, 239–241
competitiveness and, 249
Program risk management process, 240–241
“Program XYZ,” 215
Project management, program risk in, 239–241
Prototype software releases, 46
Prototyping, 145
Purchase specifications, variations in, 122
Purchasing, communication interfaces for, 228
Purchasing group, 281
Qualitative risk analysis, 236
Quality engineering, communication interfaces for, 228
Quality functional development (QFD), 46
Quality functional development tool, 127
Quality improvement, six sigma approach for, 124
Quality inspections, 152
Quality/reliability education, xvi
Quality/reliability engineering, evaluation of, xv
Quality standards, 184
Quantifying risk, 242
Quantitative criticality analysis, 68
Quantitative risk analysis, 236
Quantitative tools, 127
Questioning, six sigma as a backdrop for, 128
Questionnaires, in reliability capability assessment, 276, 278, 279
Questions, requirement-related, 130–131
Race condition, 71
Raheja, Dev, xiii, 1
Random sampling, repeated, 62
RASCI (responsible, accountable, supportive, controlling, and informed) responsibilities, 38
Rated power dissipation, 162
Raytheon, 2
RBD diagrams. See Reliability block diagrams (RBDs)
RBD models, 56–57
Reasoning, of failure precursors, 217–218
Recommendations
presenting to management, 99
reviewing, 98
Recovery code, 205
Redesign, effects of, 122
Redundancy
designing in, 3
providing, 8
Redundancy concept, 73, 74
Redundant devices, 21
Redundant hardware elements, 70
Refactoring, system trustworthiness via, 209–210, 211
Refrigerator, hazard analysis for, 264
Rejuvenation, system trustworthiness via, 205–206, 211
Release planning, software modules and, 197
Reliability, 266. See also High reliability automatic stress monitoring and, 220–221
defined, 2
designing for, xvii
importance of, xv, 184–185
improving via prognostics and health management capabilities, 213–221
management and, 233
measuring by life-cycle costs, 7
Parnas design checklist for, 196, 198
probabilistic model for, 22
as a process, 4
product life and, 226–229
in safety design, 255
science of, 1
software structure influence on, 195–196
specifying and estimating, 18
state of the art of, 2–4
system, 23
techniques to improve, 194
trustworthiness and, 193, 194
in verifying design for safety, 261–262
Reliability allocations, 23, 56
Reliability analysis (RA) practice, 268, 273
Reliability analyst, 37, 38
Reliability assessment, 231, 232. See also Organizational reliability capability assessment (ORCA)

Reliability block diagrams (RBDs), 19, 20–23, 56–57. See also RBD entries system reliability models using, 57–59

Reliability “budget,” 23

Reliability capability, 270
effect of design capability on, 271–276
Reliability capability assessment, 271
Reliability capability maturity, 275–276
Reliability data, design for safety with, 262–265
Reliability design tools, 15–35
caracteristics of, 34–35
Reliability development growth test (RDGT), 171

Reliability engineering, xv, 225
as a discipline, 16–17
role in product design life, 17–18
Reliability engineers
effectiveness of, 15
role of, xv–xvi
Reliability growth, 30
Reliability growth curves, 60
Reliability growth modeling, 32, 60
Reliability improvements, 32–33, 180
Reliability improvements (RI) practice, 268, 274
Reliability life tests, 31
Reliability management, 223–233
importance of, 223–224
Reliability measurements, 160
Reliability methods, automation of, 15
Reliability modeling, xv
value of, 53–54
Reliability models, 20–23, 53–65
goal of, 53
Reliability objectives, 272
Reliability performance, 7, 231
understanding and communicating, 17–19
Reliability practices
assessing key, 267–268, 272–276
Reliability prediction model, 54
Reliability predictions, 19–20
for components, 165
Reliability program plans, 229
Reliability programs
cost of, 227
documentation of, 229, 231–232

execution of, 229, 230–231
as feedback loops, 232–233
goals of, 224–225
management of, 224–225
planning of, 229
Reliability program shortcomings, IEEE 1624–2008 in discovering, 269
Reliability requirements, 226
Reliability requirements (RR) practice, 268, 272–273
Reliability Society future initiatives, 132
Reliability status, communicating to management and customers, 18–19
Reliability tasks, program and product life tasks tied to, 17
Reliability testing, 230–231, 232
Reliability testing (RT) practice, 268, 273
Reliability tests, traditional, 169
Reliability tools
continued use of, 28
in the design verification phase, 28–30
in the detailed design phase, 27–28
in the early program phase, 19–27
in the product life cycle, 16–17
Reliable code, 44–45
Reliable design, human factors in, 137–155
Reliable software, developing, 37–51
Reliable software design checklist, 211
Remote hazards, 259
Removal of context, as software design principle, 203–204
Repairable models, 56
Repeatable level, 276
Repeated random sampling, 62
Reports, in reliability capability assessment, 278, 279
Requirement analysis, 5–7
Requirement-related questions, 130–131
Requirements. See also Customer requirements; Development requirements; Fault isolation requirements; Full requirements development; Human factors requirements; Human performance requirements; Operating high-temperature requirement; Product requirements; Program requirements entries; Reliability requirements; Staff requirements; Stress load requirement; System requirements; User-interface requirements development of, 144
Requirements. See also Customer requirements; Development requirements; Fault isolation requirements; Full requirements development; Human factors requirements; Human performance requirements; Operating high-temperature requirement; Product requirements; Program requirements entries; Reliability requirements; Staff requirements; Stress load requirement; System requirements; User-interface requirements (Continued)

end-of-life, 226
prognostics and health management capabilities and, 221
May 2, 2012
in safety design, 254
Requirements analysis, 237–238
Requirements creep, 238
Requirements development
six sigma approach to, 121–135
system reliability and, 130
Requirements documentation, 107, 112, 113
Requirements FMECA, 73. See also Failure modes, effects, and criticality analysis (FMECA)
Requirements generation, 237–238
need for, 237–238
Requirements management, poor, 43
Requirements omission, 238
Residual risk, risk communication and, 248, 249
Responsible, accountable, supportive, consulted, informed (RASCI) tool, 128
Results, reporting, 100–101
Return on investment (ROI), 10. See also 500% return on investment; 100,000% return on investment
Reuse, of software modules, 200–201, 201–204
Rework, minimizing, 129
Risk(s), 235–236
balancing benefits against, 237
categories of, 235
competitiveness and, 249
configuration management and, 249–251
early detection of, 119
levels of, 236
human factors and, 150
potential, 38
reasons for overlooking, 237–239
requirements creep and, 238
Risk analysis, 54, 240, 241
Risk assessment(s), xvi, 240, 241, 242–243
benefits of, 236–237
in requirements generating and analysis, 237
Risk assessment code (RAC), 244, 245, 260, 266
in verifying design for safety, 262
Risk assessment code matrix, 259
safety assessment using, 260–261
Risk assessment matrix, 245, 246
Risk assessment methodology, heuristic, 109
Risk avoidance, 69
Risk-based approach, to human systems integration, 150
Risk communication, 248–249
Risk documentation, 240, 241
Risk estimation, 244–245
Risk evaluation, 245–247
Risk handling, 240, 241
during change, 251
Risk identification, 240, 241, 243–244
Risk levels, 247, 266
in risk assessment code matrix, 260–261
Risk management, 235, 240
in the change process, 249
importance of, 236–237
Risk management culture, 239
Risk Management Guide for DoD Acquisition, 240
Risk mitigation, 152, 247, 260
techniques for, 69
Risk monitoring, 240, 241
Risk planning, 240–241
Risk priority number (RPN), 68–69, 82–85, 97–98
calculating, 114
revising, 85
updating calculation of, 116
Risk profiles, 116
Risk reduction, 150
Root cause, 95
Root-cause analysis, 28–30
Root-cause failure analysis, 64, 231
Root-cause failure pattern, 180
Root-cause resolution, 180
Root causes, 29
addressing, 99
escaping detection, 113–114
historical data related to, 112–113
probability of occurrence of, 111–112
RPN calculation, 114. See also Risk priority number (RPN) updated, 116
RPN limit, 69
RPN pareto chart, 84
RPN prioritization, 98
RPN ranking, 83–84
RPN rating system, value of, 111
RPN scores, 85, 111, 114
RPN table, 83
RPN threshold, 114, 116
RPN values, 98, 111
Run time checks, software module behavior during, 195

Safer C: Developing Software for High-Integrity and Safety-Critical Systems (Hatton), 207
Safety, 266
Safety analysis, requirements creep and, 238
Safety analysis techniques, 255–260
Safety assessment, using risk assessment code matrix, 260–261
Safety control requirement testing, in verifying design for safety, 262
Safety-critical components, 9
Safety design
design and development process for, 261 reliability in, 255 start of, 254–255
Safety designer, in system safety design, 257–258, 266
Safety devices, in risk mitigation, 247, 260
Safety factors, 161 designing in, 189–190 of plastic, 190
Safety order of precedence, 258, 260
Safety rating, of products, 190
Safety team, 253
Sample analysis, 281
Sample size, 180
Sample testing, 184
S-BIT (startup BIT), 219
Schedule risk, 239–240
Scheer, Wilbert E., 189
Schottky TTL, 164
Screen tests, 18
SEI capability level, 50. See also Software Engineering Institute (SEI)
SEI CMMI model, 278. See also Capability Maturity Model Integration (CMMI®)
SEI CMMI scoring guidelines, 277
Self-assessment phase, in reliability capability assessment, 276, 279
Sensitivity, to stresses, 176–178
Sensors, for automatic stress monitoring, 220–221
Separation of concerns, as software design principle, 203
Series–parallel RBD, 56, 57, 59. See also Reliability block diagrams (RBDs)
Series RBD, 56, 57
Series RBD model, 57–58
Serious risk, 245, 247
Services, computer controllers and, 40
Severity factor guideline example, 93
Severity (SEV) factors, 92–93, 95, 96, 97, 98, 99. See also Hazard severity
Severity definitions, 258, 260
Severity of effect (SE), 82
Severity rating for software failure modes, 108–109 updated, 115
“Shall not” specifications, 6
Shocks, designing for, 3
Short cycling, 172–173
Sigma (σ), 124. See also Six sigma entries
Similarity analysis, 60
Similarity modeling method, 55
Simple designs, 129
Simplicity, trustworthiness and, 193
Simplification, system trustworthiness via, 204, 211
Simulation techniques, xv
Single-point failures avoiding, 70–71 correcting high-risk, 69 eliminating, 67, 70–71
Singpurwalla model, 60
Situational awareness, 127, 129
Situational awareness design features, 85
Six sigma analysis tools, 134–135
Six sigma (6σ) approach. See also Software design for six sigma (SDFSS) methods foundations of, 124–126 for measuring process capability and process robustness, 126 to requirements development, 121–135
Six sigma black belt, 134–135
Six sigma process, 121
DOE and, 123–124 meaning of, 124–126 successes of, 134
Six sigma process capability, 124, 125
Six sigma tools, 38, 121, 126–128
Six sigma three-pronged initiative, 126–128
Small code changes, 48
Small interface languages, in interface design, 199
Sneak failures, 12
Soak-in period, 208
Soak time, 174
Soft failure, 164
Software (SW)
  aging conditions in, 45–46
  bounding the execution domain of, 209, 211
  correct vs. reliable, 195
  developing reliable, 37–51
  evaluating under stress conditions, 38
  factoring and refactoring of, 209–210, 211
  modules in, 195, 196–200
  trustworthiness of, 193–195
Software approach, to D-FMECA, 73
Software architect, role in software design, 196
Software artifacts
  authors of, 118–119
  domain knowledge of, 117
Software changes, system trustworthiness and, 204
Software code, importance of sigma level for, 125
Software componentry, object-oriented programming vs., 199–200
Software configuration management, software reuse and, 201
Software corrosion, 3–4
Software defects
  damage from, 12–13
  staff churn and, 207
Software design, 42–43
  human-centered, 139–142
Software design checklist, 211
Software design for six sigma (SDFSS) methods, 103–104. See also Design for six sigma (DFSS) methods; Six sigma (6σ) approach
Software design metrics, 166–167
Software design principles, 201–204
Software detection activities/methods, 113
Software development process, 49
Software documentation, 45
Software element failure modes, effects of, 109
Software element identification, 106, 107
Software elements failure modes for, 108
  identifying, 117–118
Software Engineering Institute (SEI), 49. See also SEI entries
Software engineers, trustworthiness and, 194
Software execution, trustworthiness of, 194
Software failure(s), 38–39, 40, 41, 42
Software failure modes, severity rating for, 108–109
Software failure rates, 49
Software fault tolerance, system trustworthiness via, 204–205, 211
Software fault-tolerant library, 205, 211
Software FMECA. See also Failure modes, effects, and criticality analysis (FMECA) applications of, 119–120
  briefing the results from, 118
  customer requirements of, 105, 106
  lessons learned from, 117–119
  potential of, 120
  proper scoping of, 119
  roles and responsibilities with, 116–117
  vs. peer reviews and formal inspections, 118
Software FMECA process facilitator, 116–117
Software FMECA results, 118
Software FMECA RPN scores, 114
Software FMECA team(s), 106–107, 111
  structural approach for, 113
Software FMECA team brainstorming, 115–116
Software FMECA templates, 119
  populating, 117–118
Software FMECA workshops, facilitating, 118
Software-Implemented Fault Tolerance (SwiFT), 205
Software inspections, software FMECA vs., 118
Software-intensive products, 141
Software life-cycle models, 104
Software maintainability, 44
Software modules
  design principles for, 201–204
  reuse of, 200–201
Software outputs, effects on, 112
Software problems, 37
  biggest cause of, 43
Software process experts, 116
Software products, IEEE 1624–2008 in assessing, 269
Software projects, failure of, 130
Software rejuvenation, 46
  system trustworthiness via, 205–206, 211
Software reliability, 37
  computer system environment and, 46
definitions and concepts in, 40–44
design considerations concerning, 44–47
  forecasting, 49
Software reliability analysis, 55
Software reliability models, execution-time, 48–49
Software reliability prediction tools, 49–51
Software revision, 200–201
Software safety analysis, 104
Software sneak failure, 12
Software stress analysis/testing, 166–167
Software stresses, evaluating, 166
Software stress mitigation activities, 167
Software structural reinforcement, 167
Software structure, influence on reliability, 195–196
Software system call tree, 109, 110
Software system development, trustworthiness and, 193
Software systems, architecture of, 195. See also System entries
Software testing, 118
  effective, 46
Software test personnel, 118
Software test routines, 177
Software understandability, 44
Software V model, 104
Solely Reactive level, 276
Sony, 2
Source code, 107, 112, 113
testing, 48
Space exploration, 191
Space shuttle program, failure analysis in, 47
Spare capability, 73–74
“Special cause” effects, 127
Specification limits, 178
Specification reviews, 230
Specifications, 2. See also System specifications
  accuracy and completeness of, 6
developing, 5
  missing functions in, 5
oversights and omissions in, 4
Spiral development model, 44
Spiral system development, 85
Stability, checking system/software design, 208, 211
Stack-ranking, of design change priorities, 83–84
Staff churn, software defects and, 207, 211.
  See also Hiring good people; Personnel
Staff requirements, 143
Stakeholders, in cross-functional teams, 238–239
Stand-alone FMECA, 106. See also Failure modes, effects, and criticality analysis (FMECA)
Standard deviation, 124, 125
Standardized parts, software modules as, 200–201
States, of system PHM, 218
Statistical (stochastic) models, 55
Statistical tests, 127–128
Statistics for Experimenters: Design, Innovation, and Discovery (Box, J. S. Hunter & W. G. Hunter), 55
Step duration, 176–177
Step stress conditions, 175–176
Step stresses, 169, 175–176
Step-stress test, 33
Stochastic (statistical) models, 55
Stoddard, Robert W., xiii, 103
Strength curves, stress curves vs., 161–166
Stress analysis/analyses, 25
  continued, 27
derating and, 160–161
design for reliability and, 159–160
  examples of, 162–165
failure elimination and, 157–167
  principles of, 157
types of, 157
  updating, 27
Stress conditions, 178
Stress curves, strength curves vs., 161–166
Stress duration, 158
Stresses
  sensitivity to, 176–178
time compression and, 173–174
Stress load requirement, 163
Stress locations, 158
Stress model, 31
Stress monitoring, sensors for automatic, 220–221
Stress point modules, 166–167
Stress-point resolution system, 167
Stress ratio, 25
Stress-strength overlap, 178, 179
Stress–strength relationship modeling, 20
Strong cohesion, as software design principle, 201–202
Strong preconditions, removal of context and, 203–204
Structural integrity, structural reinforcement and, 167
Structural load testing, 185
Structural reinforcement, structural integrity and, 167
Structure, 195
influence on reliability, 195–196
Subsystem failure mode, 77–78
Subsystem hazard analysis (SSHA), in system safety design, 257
Success, of reliability programs, 225
Supplier issues, 230
Supplier part variation, 122
Suppliers, 281, 282
in reliability capability assessment, 279
reliability objectives and, 272
Supplier selection, IEEE 1624–2008 in, 269–270
Supplier tracking and rating system (STARS), 282
Supply-chain management (SM) practice, 268, 273
Supply chain reliability, 282
Supply chains, 271
organizational reliability capability and, 267, 269
Sustainable design, 7
Swanson, Bill, 2
Switching power supply design, 163–164
Synchronous software aging, 46
System analysis, bottom-up, 74
System applications, prognostics and health management capabilities for, 214–215
Systematic failures, 71
System combined reliability, 59
System component failure, 25
System design, communication interfaces for, 228
System-design architecture, changes related to, 70
System design FMEA, 73. See also Failure modes, effects, and criticality analysis (FMECA)
System development, spiral, 85
System D-FMECA, 74. See also Design failure modes, effects, and criticality analysis (D-FMECA)
System failure, 21
System failure history, 48–49
System failure modes, 77
identification of, 68
System failure process flow, top-down, 77–78
System FMECA, 105. See also Failure modes, effects, and criticality analysis (FMECA)
System hazard analysis (SHA), in system safety design, 257
System hazards, risk communication and, 248
System health deterioration, prognostics capability and, 214
System health management, 213
System health monitoring, 213
System instabilities, 208
System integrators (SIs), organizational reliability capability and, 267, 268, 269, 270
System-level failure mode, 78
System maintenance, software refactoring and, 210, 211
System management failures, 43
System management problems, 43, 85
System modeling, 56–57
System of systems integration (SOSI), 215
System quality/reliability, importance of, xv
System reliability ($R_{sys}$), 21, 23, 57, 58, 59
factors in, 46–47
requirements development and, 130
System reliability formula, 21
System reliability models, 56
using RBDs, 57–59
System requirements
as a factor in failure, 130
generating and analyzing, 237–238
prognostics and health management capabilities and, 221
Systems, error-tolerant, 153
System safety, goals of, 253–254
System safety assessment report (SSAR), in system safety design, 257
System safety design
reliability in, 255
start of, 254–255
Systems maintenance, 143–144
System sparing, 73
System specifications
software reliability and, 195
writing, 3, 4
System synchronization, 45

Task analysis, 146
Tasks, delineating, 143
Team culture, for risk management, 239
Team preparation, in P-FMECA, 91–92
Technical decisions, design risk and, 242
Technical risk, 240
Technique lessons learned, 4
Technological advances, 183–184
software reuse as limiting, 201
Technology, human error and, 151
Temperature-cycle load case, 176
Temperature-cycle step stress, 176
Tennessee–Tombigbee human-made waterway, 188
Testability analysis, 26
Testability engineer, 26
Test Analyze and Fix (TAAF) program, 60
Test analyze and fix (TAAF) tests, 171
Test coverage, 174–175
Test coverage analysis, 175. See also HALT test coverage analysis
Test data analysis, 31–34
Test engineering, communication interfaces for, 228
Test engineers, 117
Testing, system trustworthiness via, 204
Testing guidelines, 280
Testing programs, in verifying design for safety, 261
Test meters, 177
Test organization, risk communication in, 248–249
Test profiles, 175
Test reporting, 231
Test routines, 177
Test stimulus, 177
Thermal performance, design margin for, 178–179. See also Heat entries
Thermal stress analysis, 157
Thermal stress problems, finite element analysis and, 158
Thermal stress values, 161
Thermal testing, 33
Thousand software/source lines of code (KSLOC), 49, 50
Three sigma (3σ) process capability, 124, 125
3-state Markov model, 63
Tiered FMECA, 105. See also Failure modes, effects, and criticality analysis (FMECA)
Time-based maintenance (TBM), condition-based maintenance vs., 216–217
Time compression, 173–174, 178
Time perception, 142
Time Series Analysis: Forecasting and Control (Box, Jenkins & Reinsel), 55
Times to failure, 160
Timing capability, 47
Timing failures, 41
Tolerance factors, system safety design and, 255
Top-down design, 201
Top-down functional D-FMECA, 73–74. See also Design failure modes, effects, and criticality analysis (D-FMECA)
Top-down system failure process flow, 77–78
Top-level FMECA, 73. See also Failure modes, effects, and criticality analysis (FMECA)
Total failure rate, 19
Total product failure rate, 60
Total system reliability ($R_{sys}$), 57, 58, 59
Toyota, 2, 9
Trade-offs, between human and machine, 143
Traditional reliability tests, 169
Training, in risk mitigation, 247, 260
Training and development (TD) practice, 268, 273
Transactions, potential risks in, 38
Transistor–transistor logic (TTL) devices, 164–165
Trustworthiness, 210, 211
defined, 193–194
design for, 193–212
via design constraints, 204–210
Trustworthiness checklist, 194–195
Trustworthy design, software modules and, 197–198
Twain. Mark, 130
Twice-the-life design, 8
Two-stage reliability test approach, 171
Uncertainty in quantitative risk analysis, 236
risk assessment and, 243
Underlying causes, preventing, 115
Undetectable problems, 175
Unexpected failures, extent of, 12
Uninspected workmanship, 78
Uninterrupted primary power supply (UPS), hazard analysis for, 265
Unit under test (UUT), 33
Unreliability, of software, 195
Unspecified design parameters, 69
Unusual events, tracking, 13
Upper spec limit (USL), 124, 125
Usability, 141
Usage loads, monitoring, 217, 218
U.S. Army	designing for heat and, 186–187
technological advances and, 183–184
User evaluation, 141
User-interface requirements, 141
User-oriented products, 141
Users
checklists for, 154
error messages for, 153
in detailed safety design, 261

Variables, influencing laser performance, 122
Variation, managing, 129
Vendor lists, 281
Venn diagram, 272, 274, 275
Verification and validation (VV) practice, 268, 274
Verification methods, 96–97
Vibration testing, 33
Voice Switching Communications System (VSCS), 205–206
Vulnerability, designing to reduce, 11

Warm spare, 73, 74
Warning devices, in risk mitigation, 247, 260
Warranty, 226
Warranty costs, high reliability as mitigating, 282–283
Warranty problems, 6
Watchdog daemon, 205
Waterfall life-cycle model, 104
Weak coupling, as software design principle, 202
Weak preconditions, removal of context and, 203
Wearout failure mechanisms, 180. See also Wearout mechanisms
Wearout failures, 76
Wearout mechanisms, 178, 179
finite element analysis and, 158, 159
Weather, withstanding changes in, 184
“What-if?” analysis, 2
“What if?” studies, 100
WinNT, system upgrades to, 205–206
Work breakdown structure, 227
Workload, 143
World-class performance, achieving, 282
Worst-case applications, system safety design and, 255

Y2K bug, 39
Y2K software audits, 39
Yuhas, C. M., xiii, 193

Zero design-related single-point failures, 69
Zero failures, 1
aiming for, 2, 5
Zero investment, 10, 11
Z-value, 126