

# Index

## A

- abstraction (UCM), 46
- access policies, 141
- accountability, 29
- active-development integration branches per release, 215
- activities, 8, 48, 79-81
  - associating changes with, 183
  - change sets, 178
  - delivering together, 189
  - development branches per, 213, 284-285
  - in development workflow in UCM, 182-183
  - organizing and integrating consistent sets of versions, 8-10
  - selecting, 171-172
  - UCM integration of ClearCase and ClearQuest, 269-270
- activity-based configuration management, 44
- adding files, 17
- administration VOBs, 120
- advanced SCM tool support, 43-44
- aging charts (ClearQuest), 301-302
- ALBD servers, 101
- approval process, implementing, 172
- architects
  - ClearCase UCM process, 53
  - defining implementation models, 55-56
    - ClearCase components, 56
    - UML components, 57-58
- architecture
  - ClearCase, 97-98
    - ALBD server, 101
    - client processes, 100
    - example hardware configurations, 102-104
    - license server, 98
    - MVFS, 101-102
    - registry server, 98
    - view server, 98-100
    - VOB server, 98-100
  - software architecture, 55
- architecture-oriented project teams, organizing, 43, 162-163

- artifacts
  - controlling and auditing changes to, 4-5
  - identifying and storing in secure repositories, 4
  - organizing versioned artifacts into versioned components, 5-6
  - supporting changes to, 10-11
- assembly integration, 204-205
- Assembly project (developer builds versus release builds), 245
- assembly project model (UCM integration), 217-219
- associating changes with activities, 183
- Atria Location Broker Daemon. *See* ALBD servers
- attribute types, 92
- attributes, 48, 92
- auditing changes to artifacts, 4-5
- automated delivery, ClearCase MultiSite for, 286
- automating build process, 242-243
  
- B**
- backups, ClearCase MultiSite for, 286
- backward delta, 30
- bandwidth requirements, 107
- base ClearCase. *See* ClearCase
- baseline promotion levels, 127-128
  - Recommended Baseline Promotion Level policy setting, 137
- baseline-naming template, 236
- baselines, 86, 234-236. *See also* rebasing
  - composite baselines, 86-91
  - creating, 176
  - identifying, 134-135
  - at project milestones, 6-7
  - promoting, 239-242
  - recommending, 239-242
  - rejecting, 252
  - timing with build process, 234
  - Web site staging and release process, 250
- basic environment (example hardware configuration), 102
- BAT (business acceptance test), Web site staging and release process, 251
- best practices of SCM, 3
  - controlling and auditing changes to artifacts, 4-5
  - creating baselines as project milestones, 6-7
  - ensuring reproducibility of software builds, 12
  - identifying and storing artifacts in secure repositories, 4
  - integrating, 11-12
  - maintaining workspaces, 10
  - organizing and integrating consistent sets of versions using activities, 8-10
  - organizing versioned artifacts into versioned components, 5-6
  - organizing versioned components and subsystems into new versioned subsystems, 6
  - recording and tracking requests for change, 7
  - supporting concurrent changes to artifacts and components, 10-11
- binary delta file, 85
- branch/LATESTdevelopment integration, 209-212
- branches, 39, 81. *See also* development branches; integration branches
  - by activity, 284-285
  - by project, 281, 283
  - by site, 281, 283
  - by user, 283-284
  - for isolation and integration, 212-216
  - shared source code model, 279
- branching strategy, 213

- bug fixes, 172-173
  - build audit, 94-95
  - build management, 32-33, 42-43, 94-95
    - object sharing, 95
    - parallel and distributed builds, 95
  - build process
    - importance of, 231-232
    - in integration area, 232
      - automating, 242-243
      - baselining software components, 234, 236
      - build stabilization streams, 237-238
      - building software components, 236-237
      - locking integration stream, 232, 234
      - promoting and recommending baselines, 239, 241-242
      - smoke tests, 237
      - unlocking integration stream, 242
    - timing with baselining, 234
  - build stabilization streams, 237-238
  - build-time assembly integration, 204
  - business acceptance test (BAT), Web site staging and release process, 251
- C**
- cache
    - measurement commands, 115
    - performance monitoring and tuning, 115-116
  - canceling
    - delivery, 190
    - rebases, 193
  - cascading branches, 156
  - CCBs (change-control boards), 293
  - CCRC (ClearCase Remote Client), 69, 260, 263
  - change request management. *See* CRM
  - change requests, 290. *See also* CRM
  - change set objects, 80-81
  - change sets, 8, 48, 80-81, 178
    - redoing, 228, 307-310
    - undoing, 228, 307-308, 311
  - change-control boards (CCBs), 293
  - changes
    - to artifacts, controlling and auditing, 4-5
    - associating with activities, 183
    - concurrent changes to the same project files, 35-38
    - conflicting changes, handling, 193-198
    - delivering in UCM, 186-190
  - changing
    - life cycle phases, 20
    - personnel, 20-22
    - processes, 20-21
    - project requirements, 15-16
      - changing life cycle phases, 20
      - changing processes and personnel, 20-22
      - increasing project environment complexity, 18-19
      - increasing software system complexity, 16-18
    - software, 2
  - charts (ClearQuest), 300-302
  - chbl command, triggers on, 239
  - check-out/check-in models, 29
  - checked-out elements, checking in for delivery, 186-187
  - checking in elements for delivery, 186-187
  - checkpointing, 173
    - not available in branch/LATEST development, 211
  - classic make versus clearmake, 96
  - clear-text pools, growth of, 109
  - ClearCase, 34, 44-45, 48
    - architecture, 97-98
      - ALBD server, 101
      - client processes, 100

- example hardware configurations, 102-104
- license server, 98
- MVFS, 101-102
- registry server, 98
- view server, 98-100
- VOB server, 98-100
- combining with ClearQuest, 302-304
- components, 56
- composite baseline, 56
- concurrent changes to the same project files, 35
- distributed development support, 259
  - CCRC (ClearCase Remote Client), 263
  - ClearCase MultiSite, 265-266
  - ClearQuest MultiSite, 266-270
  - disconnected use, 262
  - local access, 262
  - remote client access, 260
  - remote terminal or desktop access, 259-260
  - Web access, 260-261
- hardware requirements, 104-105
  - CPUs, 107-108
  - dedicated ClearCase servers, 108
  - disk I/O, 106-107
  - disk space, 108, 110
  - memory, 105-106
  - network bandwidth and reliability, 107
  - user, VOB, view limits, 110
  - VOBserver size, 110-111
- local access support, 281, 283-285
- merge tools, 196, 198
- performance monitoring and tuning, 111, 113
  - lowest level measurements, 114-115
  - middle level measurements, 115-116
  - top level measurements, 116
- producer/consumer model support, 274-275
- shared source code model support, 279
- views, 56, 67-68, 99
  - dynamic versus snapshot, 71, 176
  - dynamic views, 68-71, 101
  - integration view, 181, 189-190
  - limits, 110
  - shared views, 208-209
  - snapshot views, 68-69
  - storage, 99-100
  - view type default policies, 141-142
  - Web views, 69
- VOBs, 65-67
- workspace management, 41
- ClearCase MultiSite, 265-266. *See also*
  - distributed development
    - for automated delivery, 286
    - for backups, 286
    - for interoperability, 286-287
    - using with ClearQuest MultiSite, 268-270
  - ClearCase projects, 59-60, 73, 131. *See also*
    - development workflow in UCM
      - baselines, identifying, 134-135
      - branches by, 281, 283
      - categories of, 131-133
      - components, identifying, 134-135
      - coordinating, 155
        - cooperating projects, 164-167, 169
        - documentation projects, 173-176
        - IS/IT development projects, 169-173
        - large multiproject development efforts, 161-164
        - multiple parallel releases, 155-160
        - small team projects, 173-176
    - core projects, 88
    - creating, 134, 147-149, 152
    - database projects, 88
    - developers, joining and developing, 61
    - GUI projects, 88
    - integration streams, 133

- integrators, 62
- joining, 178
- managing with project manager, 59-60
- parent/child relationships
  - between projects, 78
  - within projects, 78
- project manager, identifying, 134
- project policies, 133, 135
  - access policies, 141
  - component policies, 136-137
  - deliver policies, 137-140
  - view type default policies, 141-142
- scope of work, 133
- selecting location for, 147
- UCM project properties, 142-147
- ClearCase Remote Client (CCRC), 69, 260, 263
- ClearCase UCM (Unified Change Management), 44-45, 53, 177
  - activities, 79-81
  - architects, 53
  - communication, 48
  - composite baselines, 86-91
  - configuration managers, 53
  - developers, 54
  - integrators, 54
  - project managers, 54
  - projects, 73
  - streams, 73
    - extended stream relationships, 79
    - purposes of, 74-76
    - relationships between, 76, 78
- ClearCase UCM Baseline + Change Model, 49-51
  - ClearCase model, 48-49
  - ClearQuest, 46
    - communication, 48
    - support, 302-304
  - components, 86
- development workflow, 177-178
  - activities, 182-183
  - command line, 185-186
  - conflicting changes, handling, 193-198
  - creating development streams, 179-182
  - delivering changes, 186-190
  - integration with developer's IDE, 199-200
  - joining projects, 178
  - modifying files and directories, 183, 185
  - rebasing development stream, 190-193
- integration, 216-217
  - assembly project model, 217-219
  - of ClearCase and ClearQuest, 269-270
  - deliver variations, 225-228
  - hierarchy of streams model, 222-223
  - mixed streams model, 224
  - shared streams model, 222
  - single stream project model, 219
  - traditional parallel project model, 219-222
- labels, 92
- local access support, 281
- producer/consumer model support, 273-274
- redoing change sets, 307-310
- shared source code model support, 278
- undoing change sets, 307-308, 311
- value of, 46
  - abstraction, 46
  - communication, 48
  - control, 47
  - stability, 47
- ClearCase UCM model, 48-49
- ClearCase UCM project properties, 142
  - ClearQuest properties, 144-147
  - general properties, 143
- ClearCase Web Client, 69, 260-261
- ClearCase Web Server, 69

- clearmake, 94-95
  - parallel and distributed builds, 95
  - versus classic make, 96
- ClearQuest, 52-53, 294, 296-297
  - activities, 81
  - charts, 300-302
  - in deployment stage, 247-248
  - queries, 297-298
  - reports, 299
  - UCM, 46
    - communication, 48
    - support, 302-304
- ClearQuest Designer, 296
- ClearQuest Eclipse Client, 294
- ClearQuest MultiSite, 266-268, 304
  - using with ClearCase MultiSite, 268-270
- ClearQuest properties, 144
  - Link to This ClearQuest User Database policy, 145
  - Perform ClearQuest Action After Changing Activity policy, 147
  - Perform ClearQuest Action After Delivery policy, 145
  - Perform ClearQuest Action Before Changing Activity policy, 147
  - Perform ClearQuest Action Before Delivery policy, 145
  - Perform ClearQuest Action Before Work On policy, 145
  - Project Is ClearQuest-Enabled policy, 144
  - Transfer ClearQuest Mastership After Delivery policy, 146, 270
  - Transfer ClearQuest Mastership Before Delivery policy, 145, 270
  - Transition to Complete Action After Changing Activity policy, 147
  - Transition to Complete After Delivery policy, 146
- client processes, 100
- clients, view storage on, 100
- CMS (configuration management system), 22
- command-line interface
  - creating PVOBs, 118-119
  - creating VOBs and components, 122-123
  - in UCM, 185-186
- commands
  - Deliver, 188-189
  - Rebase, 191
- commercial software, staging and release processes, 248
- communication
  - challenges to distributed development, 257
  - UCM, 48
- completing
  - delivery, 190
  - rebases, 193
- completion stage (CRM), 291, 293
- complex environment (example hardware configuration), 104
- complexity, increasing
  - in project environment, 18-19
  - in software systems, 16-18
- component policies, 136-137
- component-based development, 43, 162-163
- components, 57
  - baseline promotion levels, 127-128
  - ClearCase, 56
  - creating
    - with command-line interface, 122-123
    - with graphical user interface, 123-126
  - identifying, 134-135
  - independent components, coordinating
    - cooperating projects, 164-166
  - releasing, 62
  - shared components, coordinating
    - cooperating projects, 167, 169
  - supporting changes to, 10-11

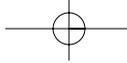
- third-party components, 17
- UCM, 86
  - versioned components, 5-6
- composite baselines, 56, 86-87, 89-91
- compressed file, 85
- compressed text file, 85
- concurrent changes, 18
- configuration control, 4
- configuration identification, 4
- configuration management, 2
- configuration management system (CMS), 22
- configuration managers
  - ClearCase UCM process, 53
  - setting up SCM environment, 58-59
- configuration records, 94-95
- configuration specification, 68
- configuration status accounting, 4
- conflicting changes, handling, 193-196, 198
- control (UCM), 47
- controlling changes to artifacts, 4-5
- cooperating projects, coordinating
  - independent components, 164-166
  - shared components, 167-169
- coordinating ClearCase projects, 155
  - cooperating projects, 164-169
  - documentation projects, 173-176
  - IS/IT development projects, 169-173
  - large multiproject development efforts, 161-164
  - multiple parallel releases, 155-160
  - small team projects, 173-176
- copy-over problem, 28
- core projects, 88
- CPU requirements, 107-108
- CRM (change request management), 289-290
  - ClearQuest, 294, 296-297
    - charts, 300-302
    - queries, 297-298
    - reports, 299
    - UCM support, 302-304

- ClearQuest MultiSite, 304
  - stages of, 290-291
    - completion stage, 293
    - decision stage, 292-293
    - evaluation stage, 292
    - implementation stage, 293
    - submission stage, 291-292
    - verification stage, 293
- csset.pl script
  - integrator role usage, 307
  - interface, 308
  - location of, 307
  - redo operation, 309-311
  - undo operation, 311
  - usefulness of, 308-309

## D

- database projects, 88
- database storage, growth of, 109
- DEC (Digital Equipment Corporation), 22
- decision stage (CRM), 291-293
- dedicated ClearCase servers, 108
- defects, 290
  - completion stage (CRM), 293
  - decision stage (CRM), 292-293
  - evaluation stage (CRM), 292
  - implementation stage (CRM), 293
  - submission stage (CRM), 291-292
  - verification stage (CRM), 293
- defining implementation model, 116-117
- Deliver command, 188-189
- deliver policies, 137
  - all deliveries, 138-139
  - interproject deliveries, 140
  - intraproject deliveries, 139-140
- deliver variations (UCM integration), 225
  - interproject deliveries, 227
  - posted delivery, 227
  - push versus pull, 225

- triggers, 227
- undoing and redoing change sets, 228
- delivering
  - activities together, 189
    - UCM integration of ClearCase and ClearQuest, 270
  - changes
    - ClearCase MultiSite for, 286
    - handling conflicting changes, 193-194
    - in UCM, 186-190
- delta storage, 29-31
- deltas, 30
  - backward delta, 30
  - forward delta, 30
  - inline delta, 31
- deployment process, 243-244
- deployment stage
  - ClearQuest in, 247-248
  - modeling with streams, 245-246
- derived object pools, growth of, 109
- derived objects, 94-95
- developer builds versus release builds, 245
- developer isolation. *See* isolation
- developer streams, traditional parallel project model (UCMintegration), 220-222
- developers
  - ClearCase UCM process, 54
  - development branches per, 213
  - joining projects, 61
- development
  - component-based development, 43
  - geographically distributed development, 43
  - of projects, 61
- development branches, 213
  - per activity, 213
  - per developer, 213
  - per feature, 213
  - per patch, 215
  - per patch bundle, 216
- development streams, 133
  - creating, 179, 181-182
  - rebasing, 190-191, 193
    - before delivery, 188
- development teams. *See* teams
- development technologies, challenges to distributed development, 257-259
- development workflow in UCM, 177-178
  - activities, 182-183
  - changes, delivering, 186-190
  - command line, 185-186
  - conflicting changes, handling, 193-198
  - development streams
    - creating, 179, 181-182
    - rebasing, 190-191, 193
  - files and directories, modifying, 183, 185
  - integration with developer's IDE, 199-200
  - projects, joining, 178
- DevTest state, Web site staging and release process, 250
- Digital Equipment Corporation (DEC), 22
- directories, 85
  - modifying in UCM, 183, 185
  - version control, 17
- directory versioning, 83-84
- disconnected use (ClearCase distributed development support), 262
- disk I/O requirements, 106-107
- disk space requirements, 108, 110
- distributed builds, 95
- distributed development, 19, 43, 255. *See also*
  - ClearCase MultiSite
    - challenges to, 256
    - communication, 257
    - organization, 256
    - technology, 257-259
  - ClearCase support for, 259
  - CCRC(ClearCase Remote Client), 263
  - ClearCase MultiSite, 265-266
  - ClearQuest MultiSite, 266, 268-270



- disconnected use, 262
    - local access, 262
    - remote client access, 260
    - remote terminal or desktop access, 259-260
    - Web access, 260-261
  - producer/consumer model, 270-273, 275
    - ClearCase support, 274-275
    - UCM support, 273-274
  - remote team members model, 280-281, 286
    - ClearCase local access support, 281, 283-285
    - UCM local access support, 281
  - shared source code model, 275-278, 280
    - ClearCase support, 279
    - UCM support, 278
  - distribution charts (ClearQuest), 300
  - documentation, VOB server size requirements, 110-111
  - documentation projects, coordinating, 173-176
  - dynamic views, 68-71, 101
    - versus snapshot views, 71-72, 176
- E**
- Eclipse, ClearQuest Eclipse Client, 294
  - element types, 84-85
  - elements, 56, 81
  - embedded systems, staging and release processes, 248-249
  - emergency bug fixes, 172-173
  - enforcement, 20
  - enhancement requests, 290
    - completion stage (CRM), 293
    - decision stage (CRM), 292-293
    - evaluation stage (CRM), 292
    - implementation stage (CRM), 293
    - submission stage (CRM), 291-292
    - verification stage (CRM), 293
  - evaluation environment (example hardware configuration), 102
  - evaluation stage (CRM), 291-292
  - evolution of SCM tools, 22-23
    - activity-based configuration management, 44
    - advanced SCM tool support, 43-44
    - build management, 32-33, 42-43
    - component-based development, 43
    - concurrent changes to the same project files, 35-38
    - delta storage, 30-31
    - geographically distributed development, 43
    - modern SCM tool support, 34
    - parallel development support, 38, 40-41
    - project team categories, 23-25
    - release management, 42-43
    - release management/product maintenance, 33-34
    - repositories and check-out/check-in models, 29
    - workspace management, 31-32, 41
  - extensive project teams, 25
  - extensive team integration, 206, 208
  - extreme projects (project category), 133
- F**
- feature integration, 206
  - feature-oriented project teams, organizing, 163-164
  - features, development branches per, 213
  - Feiler, Peter, 8
  - file copies, supporting early SCM, 25-26
  - file systems, MVFS (multiversion file system), 101-102
  - files, 84
    - adding, 17
    - concurrent changes to the same project files, 35-38
    - modifying in UCM, 183, 185

follow-on projects, 155-156  
  creating, 157  
forward delta, 30  
foundation baselines, 134  
full baselines, 236

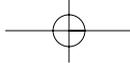
## G-H

geographically distributed development. *See*  
  distributed development  
graphical user interface  
  creating PVOBs, 119  
  creating VOBs and components, 123-126  
GUI projects, 88  
hardware configuration management, 2  
hardware configurations, examples, 102-104  
hardware devices, traceability of software  
  versions, 249  
hardware requirements, 104-105  
  CPUs, 107-108  
  dedicated ClearCase servers, 108  
  disk I/O, 106-107  
  disk space, 108, 110  
  memory, 105-106  
  network bandwidth and reliability, 107  
  user, VOB, view limits, 110  
  VOB server size, 110-111  
hierarchy of streams model (UCMintegration),  
  222-223  
html (Hypertext Markup Language), element  
  types, 85  
hyperlinks, 92

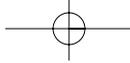
## I

I/O requirements, 106-107  
IDE, integration with, 199-200  
identifying artifacts in secure repositories, 4

implementation models  
  architects, 55-56  
  ClearCase components, 56  
  UML components, 57-58  
  defining, 116-117  
implementation stage (CRM), 291, 293  
implementation technologies, challenges to  
  distributed development, 257-258  
implementation view, 56  
importing source code for creating VOBs,  
  126-127  
increasing  
  complexity  
    project environment, 18-19  
    software systems, 16-18  
  number and frequency of product  
    releases, 19  
  team size, 18  
incremental baselines, 236  
independent components, coordinating  
  cooperating projects, 164-166  
individual integration, 205  
individual projects, 24, 132  
inline delta, 31  
integrated system vendor (ISV), 15  
integration, 203-204. *See also* conflicting  
  changes, handling  
    assembly integration, 204-205  
    best practices for SCM, 11-12  
    consistent sets of versions using activities,  
      8-10  
    with developer's IDE, 199-200  
  and isolation, 208  
    branch/LATEST development, 209-212  
    branches, 212-216  
    shared views, 208-209  
    UCM, 216-228  
  merge integration, 204



- scenarios
    - extensive team integration, 206, 208
    - individual integration, 205
    - major team integration, 205-206
    - small and modest team integration, 205
  - integration area, build process in, 232
    - automating, 242-243
    - baselining software components, 234, 236
    - build stabilization streams, 237-238
    - building software components, 236-237
    - locking integration stream, 232, 234
    - promoting and recommending baselines, 239, 241-242
    - smoke tests, 237
    - unlocking integration stream, 242
  - integration branches, 213
    - per patch bundle, 216
    - per promotion level, 214
    - per release, 214
    - per system variant, 216
  - integration streams, 133
    - locking, 232, 234
    - unlocking, 242
  - integration view, 181
    - testing delivery in, 189-190
  - integrator role, cset.pl script usage, 307
  - integrators, 203
    - ClearCase UCM process, 54
    - projects, 62
    - releasing components, 62
    - releasing systems and subsystems, 63
    - system integration, 62
  - interface, cset.pl script, 308
  - internal software components, staging and release processes, 252-253
  - interoperability, ClearCase MultiSite for, 286-287
  - interproject deliveries, 227
    - deliver policies, 140
    - intraproject deliveries, deliver policies, 139-140
  - IS/IT development projects, coordinating, 169-173
  - isolation and integration, 208
    - branch/LATEST development, 209-212
    - branches, 212-216
    - shared views, 208-209
    - UCM, 216-217, 219-225, 227-228
  - ISV (integrated system vendor), 15
  - iteration planning, 164, 166, 168
  - iterative performance tuning process, 113
- J-K-L**
- joining projects, 178
  - label types, 92
  - labels, 92
    - relating version sets, 236
  - large multiproject development efforts, coordinating, 161-164
  - Last Released state, Web site staging and release process, 252
  - latency, 114
  - license servers, 98
  - life cycle phases, changing, 20
  - Link to This ClearQuest User Database policy, 145
  - local access (ClearCase distributed development support), 262
    - ClearCase support, 281, 283-285
    - UCM support, 281
  - locking integration stream, 232, 234
  - logical design, defining implementation model, 116-117
  - lowest level measurements (performance monitoring and tuning), 114-115

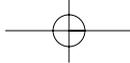


## M

- mainline integration branches, 215
- mainline projects, 156-157
- maintaining workspaces, 10
- maintenance projects, 157
  - emergency bug fixes, 172-173
- major project teams, 24, 132
- major team integration, 205-206
- Make Baseline operation, 234
- management
  - activity-based configuration management, 44
  - build management, 32-33, 42-43
  - release management, 33-34, 42-43
  - UCM. *See* UCM
  - workspace management, 31, 41
- managers
  - configuration managers
    - ClearCase, 53
    - setting up SCM environments, 58-59
  - project managers
    - ClearCase, 54
    - managing projects, 59-60
- managing types, 93-94
- mapping, defining implementation model, 117
- mastership, 265, 268
- memory
  - effect on performance, 114
  - requirements, 105-106
- merge integration, 204
- merge tools (ClearCase), 196, 198
- merges. *See* conflicting changes, handling; integration
- metadata, 66-67, 91
- Microsoft Word, element types, 85
- middle level measurements (performance monitoring and tuning), 115-116
- mixed streams model (UCMintegration), 224
- modeling deployment stage with streams, 245-246
- modest environment (example hardware configuration), 103
- modest project teams, 24, 132
- modest team integration, 205
- Modifiable Components policy setting, 136
- modifying files and directories in UCM, 183, 185
- monitoring performance, 111, 113
  - lowest level measurements, 114-115
  - middle level measurements, 115-116
  - top level measurements, 116
- multiple parallel releases, coordinating, 155-160
- multiple PVOBs, creating, 120-121
- multiproject development efforts, coordinating, 161-164
- MultiSite. *See* ClearCase MultiSite
- MVFS (multiversion file system), 69-71, 101-102

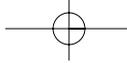
## N-O

- naming with baseline-naming template, 236
- network, effect on performance, 114
- network bandwidth and reliability requirements, 107
- nightly build process. *See* build process
- object sharing, 95
- organization, challenges to distributed development, 256
- organizing. *See also* coordinating ClearCase projects
  - consistent sets of versions using activities, 8-10
  - projects, 147
  - versioned artifacts into versioned components, 5-6
  - versioned components and subsystems into versioned subsystems, 6

**P**

- parallel builds, 95
- parallel development support, 18, 38-41, 193
- parallel project baseline graph, 159, 161
- parallel project model (UCMintegration), 219-222
- parallel releases, coordinating multiple
  - parallel releases, 155-157, 159-160
- parent/child streams model (UCM integration), 77, 222-223
- partial change deliveries, 138
- PAT (production acceptance test), Web site staging and release process, 251
- patch bundles, development and integration branches per, 216
- patches, development branches per, 215
- peer streams, 76
- Perform ClearQuest Action After Changing Activity policy, 147
- Perform ClearQuest Action After Delivery policy, 145
- Perform ClearQuest Action Before Changing Activity policy, 147
- Perform ClearQuest Action Before Delivery policy, 145
- Perform ClearQuest Action Before Work On policy, 145
- performance monitoring and tuning, 111, 113
  - lowest level measurements, 114-115
  - middle level measurements, 115-116
  - top level measurements, 116
- personnel, changing, 20-22
- physical implementation models. *See* implementation models
- platform interoperability, ClearCase MultiSite for, 286-287
- platforms, supporting multiple platforms, 18
- policies. *See* project policies
- posted delivery, 227
- processes
  - changing, 20-21
  - SCM processes, 12
- PROD state, Web site staging and release process, 251
- producer/consumer model (distributed development), 270-273, 275
  - ClearCase support, 274-275
  - UCM support, 273-274
- product maintenance, 33-34
- product releases, increasing number and frequency, 19
- production acceptance test (PAT), Web site staging and release process, 251
- productivity with branch/LATEST development, 212
- Project Creation Wizard, 73, 147-149, 152
- Project Explorer, 73
- project files, concurrent changes to, 35-38
- Project Is ClearQuest-Enabled policy, 144
- Project Management Institute, 73
- project managers
  - ClearCase UCM process, 54
  - identifying, 134
  - managing projects, 59-60
- project milestones, creating baselines, 6-7
- project policies, 133, 135
  - access policies, 141
  - component policies, 136-137
  - deliver policies, 137
    - all deliveries, 138-139
    - interproject deliveries, 140
    - intraproject deliveries, 139-140
  - view type default policies, 141-142
- project requirements, dealing with changes, 15-16
  - changing life cycle phases, 20
  - changing processes and personnel, 20-22

- increasing project environment complexity, 18-19
    - increasing software system complexity, 16-18
  - project team categories, 23-25
  - project VOBs. *See* PVOBs
  - project-integration streams. *See* integration streams
  - projects (ClearCase), 73, 131. *See also*
    - development workflow in UCM
      - baselines, identifying, 134-135
      - branches by, 281, 283
      - categories of, 131-133
      - components, identifying, 134-135
      - coordinating, 155
        - cooperating projects, 164-167, 169
        - documentation projects, 173-176
        - IS/IT development projects, 169-173
        - large multiproject development efforts, 161-164
        - multiple parallel releases, 155-160
        - small team projects, 173-176
    - core projects, 88
    - creating, 134, 147-149, 152
    - database projects, 88
    - developers, joining and developing, 61
    - GUI projects, 88
    - integration streams, 133
    - integrators, 62
    - joining, 178
    - managing with project manager, 59-60
    - parent/child relationships
      - between projects, 78
      - within projects, 78
    - project manager, identifying, 134
    - project policies, 133, 135
      - access policies, 141
      - component policies, 136-137
      - deliver policies, 137-140
      - view type default policies, 141-142
    - scope of work, 133
    - selecting location for, 147
    - UCM project properties, 142-147
  - promotion, 62
    - of baselines, 239, 241-242
  - promotion levels
    - baseline promotion levels, 127-128
    - integration branches per, 214
    - Recommended Baseline Promotion Level policy setting, 137
  - promotion-level authorization, 239
  - properties. *See* UCM project properties
  - public VOBs, 125
  - pull deliver versus push deliver, 225
  - push deliver versus pull deliver, 225
  - PVOBs (project VOBs), 67, 118
    - creating
      - with command-line interface, 118-119
      - with graphical user interface, 119
      - multiple PVOBs, 120-121
- Q-R**
- queries (ClearQuest), 297-298
  - Rational Unified Process
    - creating baselines at project milestones, 6
    - defining implementation model, purpose of, 116
  - RCS (revision control system), 22, 66
  - Read-only Components policy setting, 136
  - Rebase command, 191
  - rebasing, 61, 139. *See also* baselines
    - development streams, 190-191, 193
      - before delivery, 188
      - handling conflicting changes, 195-196
  - Recommended Baseline Promotion Level policy setting, 137
  - recommending baselines, 239, 241-242
  - recording requests for change, 7

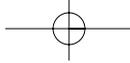


- redoing change sets, 228, 307-310
  - registry servers, 98
  - rejecting baselines, 252
  - relating version sets, 236
  - relationships between streams, 76
    - extended stream relationships, 79
    - parent/child stream relationships between projects, 78
    - parent/child stream relationships within projects, 78
  - release
    - active-development integration branches per, 215
    - integration branches per, 214
  - release builds versus developer builds, 245
  - release management, 33-34, 42-43
  - release process, 243-244
    - commercial software, 248
    - embedded systems, 248-249
    - internal software components, 252-253
    - Web sites, 249-251
  - Released state, Web site staging and release process, 252
  - releasing
    - components, 62
    - systems and subsystems, 63
  - reliability requirements, 107
  - remote client access (ClearCase distributed development support), 260
  - remote team members model (distributed development), 280-281, 286
    - ClearCase local access support, 281-285
    - UCM local access support, 281
  - remote terminal or desktop access (ClearCase distributed development support), 259-260
  - replicas, 265
  - replication, user databases and schema databases, 268
  - reports (ClearQuest), 299
  - repositories, 29
    - identifying and storing artifacts in secure repositories, 4
    - VOBs. *See* VOBs
  - reproducibility of software builds, 12
  - requests for change, recording and tracking, 7
  - requirements. *See* hardware requirements
  - reserved check-out, 35
  - revision control system (RCS), 22, 66
  - Rose models, element types, 85
  - runtime assembly integration, 204
- S**
- SCCS (source code control system), 22
  - schema databases, 267
    - replication, 268
  - SCM (software configuration management), 1-3
    - best practices, 3
    - controlling and auditing changes to artifacts, 4-5
    - creating baselines as project milestones, 6-7
    - ensuring reproducibility of software builds, 12
    - identifying and storing artifacts in secure repositories, 4
    - integrating, 11-12
    - maintaining workspaces, 10
    - organizing and integrating consistent sets of versions using activities, 8-10
    - organizing versioned artifacts into versioned components, 5-6
    - organizing versioned components and subsystems into new versioned subsystems, 6
    - recording and tracking requests for change, 7
    - supporting concurrent changes to artifacts and components, 10-11

- components, 5
- environments, setting up configuration manager, 58-59
- file copies, 25-26
- processes, 12
- tools, evolution of, 12, 22-23
  - activity-based configuration management, 44
  - advanced SCM tool support, 43-44
  - build management, 32-33, 42-43
  - component-based development, 43
  - concurrent changes to the same project files, 35-38
  - delta storage, 30-31
  - geographically distributed development, 43
  - modern SCM tool support, 34
  - parallel development support, 38, 40-41
  - project team categories, 23-25
  - release management, 42-43
  - release management/product maintenance, 33-34
  - repositories and check-out/check-in models, 29
  - workspace management, 31-32, 41
- scope of work (ClearCase projects), 133
- SEI-CMM (Software Engineering Institute's Capability Maturity Model), 21
- selecting activities, 171-172
- servers
  - ALBD servers, 101
  - license servers, 98
  - registry servers, 98
  - view servers, 98-100
    - combining with VOB servers, 99
    - as separate from VOB servers, 99-100
  - VOB servers, 98-100
    - combining with view servers, 99
    - CPU requirements, 107-108
    - dedicated ClearCase server requirements, 108
    - disk I/O requirements, 106-107
    - disk space requirements, 108, 110
    - hardware requirements, 104-111
    - memory requirements, 105-106
    - network bandwidth and reliability requirements, 107
    - as separate from view servers, 99-100
    - sizing requirements, 110-111
    - user, VOB, view limits requirements, 110
- sets of versions, organizing and integrating using activities, 8-10
- shared components, coordinating cooperating projects, 167, 169
- shared copy approach, 25-28
- shared source code model (distributed development), 275-278, 280
  - ClearCase support, 279
  - UCM support, 278
- shared streams model (UCMintegration), 222
- shared views, 208-209
- sharing objects, 95
- single stream project model (UCM integration), 219
- single-stream projects, creating, 175
- sites, branches by, 281, 283
- small project teams, 24, 132
  - coordinating, 173-176
- small team integration, 75, 205
- smoke tests, 237
- snapshot views, 68-69
  - versus dynamic views, 71-72, 176
- software, changing, 2
- software architecture, 55
- software configuration management. *See* SCM
- Software Engineering Institute's Capability Maturity Model (SEI-CMM), 21

- software integration. *See* integration
  - software reuse, 17
  - software versions, traceability, 249
  - source code
    - importing for creating VOBs, 126-127
    - VOB server size requirements, 110-111
  - source code control system (SCCS), 22
  - source pools, growth of, 109
  - stability (UCM), 47
  - staging component, 244-245
  - staging process, 243-244
    - commercial software, 248
    - embedded systems, 248-249
    - internal software components, 252-253
    - staging component, 244-245
    - Web sites, 249-251
  - state types, 146
  - storing artifacts in secure repositories, 4
  - streams, 73
    - development streams, 133
      - creating, 179, 181-182
      - rebasing, 190-191, 193
      - rebasing before delivery, 188
      - traditional parallel project model (UCM integration), 220-222
    - hierarchy of streams model (UCM integration), 222-223
    - integration streams, 133
    - mixed streams model (UCM integration), 224
    - modeling deployment stage, 245-246
    - parent/child streams, 77
    - peer streams, 76
    - purposes of, 74-76
    - relationships between, 76
      - extended stream relationships, 79
      - parent/child stream relationships between projects, 78
      - parent/child stream relationships within projects, 78
    - shared streams model (UCM integration), 222
    - single stream project model (UCM integration), 219
  - submission stage (CRM), 291-292
  - subsystems
    - integration, 206
    - organizing into versioned subsystems, 6
    - releasing, 63
  - system integration, 62, 206
  - system requirements. *See* hardware requirements
  - system utilities, measuring resource usage, 114
  - system variants, integration branches per, 216
  - systems, releasing, 63
  - SysTest state, Web site staging and release process, 250
- T**
- tasks. *See* activities
  - team isolation. *See* isolation
  - teams
    - extensive project teams, 25
    - geographically distributed development teams, 19
    - increasing size of, 18
    - major project teams, 24
    - modest project teams, 24
    - project team categories, 23-25
    - small project teams, 24
  - technology, challenges to distributed development, 257
    - development technologies, 258-259
    - implementation technologies, 257-258
  - templates, baseline-naming template, 236
  - testing
    - delivery in integration view, 189-190
    - rebases, 193

- text file, 85
  - third-party components, 17
  - tools
    - ClearCase. *See* ClearCase
    - SCM tools, 12
  - top level measurements (performance monitoring and tuning), 116
  - traceability, 38
    - between hardware devices and software versions, 249
  - tracking requests for change, 7
  - traditional parallel project model (UCM integration), 219-222
  - Transfer ClearQuest Mastership After Delivery policy, 146, 270
  - Transfer ClearQuest Mastership Before Delivery policy, 145, 270
  - Transition to Complete Action After Changing Activity policy, 147
  - Transition to Complete After Delivery policy, 146
  - transparency, 101
  - trend charts (ClearQuest), 301
  - triggers, 93
    - on chbl command, 239
    - on delivery, 227
  - trivial merge, 193
  - tuning performance, 111, 113
    - lowest level measurements, 114-115
    - middle level measurements, 115-116
    - top level measurements, 116
  - Tykal, Jim, 90
  - type managers, 86, 93-94
- U**
- UCM (Unified Change Management), 44-45, 53, 177
    - activities, 79-81
    - architects, 53
    - communication, 48
    - composite baselines, 86-91
    - configuration managers, 53
    - developers, 54
    - integrators, 54
    - project managers, 54
    - projects, 73
    - streams, 73
      - extended stream relationships, 79
      - purposes of, 74-76
      - relationships between, 76, 78
  - UCM Baseline + Change Model, 49-51
    - ClearCase model, 48-49
    - ClearQuest, 46
      - communication, 48
      - support, 302-304
  - components, 86
  - development workflow, 177-178
    - activities, 182-183
    - command line, 185-186
    - conflicting changes, handling, 193-198
    - creating development streams, 179-182
    - delivering changes, 186-190
    - integration with developer's IDE, 199-200
    - joining projects, 178
    - modifying files and directories, 183, 185
    - rebasing development stream, 190-193
  - integration, 216-217
    - assembly project model, 217-219
    - of ClearCase and ClearQuest, 269-270
    - deliver variations, 225-228
    - hierarchy of streams model, 222-223
    - mixed streams model, 224
    - shared streams model, 222
    - single stream project model, 219
    - traditional parallel project model, 219-222
  - labels, 92



- local access support, 281
  - producer/consumer model support, 273-274
  - redoing change sets, 307-310
  - shared source code model support, 278
  - undoing change sets, 307-308, 311
  - value of, 46
    - abstraction, 46
    - communication, 48
    - control, 47
    - stability, 47
  - UCM model, 48-49
  - UCM project properties, 142
    - ClearQuest properties, 144-147
    - general properties, 143
  - UML (Unified Modeling Language), 57
    - components, implementation models, 57-58
  - undoing change sets, 228, 307-308, 311
  - Unified Change Management. *See* UCM
  - The Unified Modeling Language User Guide*, 57
  - Unified Modeling Language. *See* UML
  - UNIX MVFS, 70-71
  - unlocking integration stream, 242
  - unreserved check-out, 35
  - Update, 68
  - user databases, 267
    - replication, 268
  - user interface for ClearQuest, 294
  - users
    - branches by, 283-284
    - limits, 110
- V**
- value of UCM, 46
    - abstraction, 46
    - communication, 48
    - control, 47
    - stability, 47
  - variants. *See* system variants
  - verification stage (CRM), 291, 293
  - version sets, relating, 236
  - version skew, 212
  - version trees, 81-82
  - versioned artifacts, organizing into versioned components, 5-6
  - versioned components
    - organizing subsystems into versioned subsystems, 6
    - organizing versioned artifacts into, 5-6
  - versioned object bases. *See* VOBs
  - versioned subsystems, organizing, 6
  - versioning
    - directories, 17, 83-84
    - MVFS, 101
  - view servers, 98-100
    - combining with VOB servers, 99
    - as separate from VOB servers, 99-100
  - view type default policies, 141-142
  - views, 56, 67-68, 99
    - dynamic versus snapshot, 71, 176
    - dynamic views, 68-71, 101
    - integration view, 181
      - testing delivery in, 189-190
    - limits, 110
    - shared views, 208-209
    - snapshot views, 68-69
    - storage, 99
      - on clients, 100
      - separate from VOBs, 99-100
      - with VOBs, 99
    - view type default policies, 141-142
    - Web views, 69
  - VOB servers, 98-100
    - combining with view servers, 99
    - hardware requirements, 104-105
      - CPUs, 107-108
    - dedicated ClearCase servers, 108

- disk I/O, 106-107
  - disk space, 108, 110
  - memory, 105-106
  - network bandwidth and reliability, 107
  - sizing, 110-111
  - user, VOB, view limits, 110
  - as separate from view servers, 99-100
  - VOBs (versioned object bases), 65-67, 98, 265
    - creating, 117-118
      - administration VOBs, 120
      - with command-line interface, 122-123
      - with graphical user interface, 123-126
      - importing existing source code, 126-127
      - multiple PVOBs, 120-121
      - PVOBs with command-line interface, 118-119
      - PVOBs with graphical user interface, 119
    - limits, 110
    - public, 125
    - storage, 99
      - separate from views, 99-100
      - on servers, 100
    - with views, 99
- W-Z**
- Web access (ClearCase distributed development support), 260-261
  - Web sites
    - commercial software available from, 248
    - staging and release processes, 249-251
  - Web views, 69
  - Windows
    - directory versioning, 83-84
    - dynamic views, 71
  - workflow. *See* development workflow in UCM
  - workspace management, 31-32, 41
  - workspaces, 61. *See also* views
    - maintaining, 10
  - XML (Extensible Markup Language), element types, 85

