Index

Abstraction, 29–31
  dictionary definitions, 29–30
Academia definitions, software, 19, 21–22
Acceptance, customer, 121
Acme Widget Co., 142–44
  test system, 143
  Widgetometer, 143
Actions
  Add History Display, 207
  drive, 188–90
  Get, 82
  Load, 84
  Load All, 85
  new control, 205
  read, 190–93
  Reset, 86–88
  Save, 84
  Save All, 84
  Set, 82
Add History Display action, 207
Advantages, good design leveraging, 13
AI (analogue input), 75
Application, good software, 15
Architecture documents, software, 119–20
ATE (Automatic Test Equipment), 2
Attributes, UI, 167

Bad cohesion, 25
Bad coupling, 22–24
Bad information hiding, 26–27
Big picture, 114–16
Block diagrams, 134–37
  general layout standards, 134
  labeling standards, 135–36
  self-documenting example, 136–37
  wiring standards, 134–35
Book
  not buying, 4
  what it is, 6
Bottom-up design, 40
Build, 180–203
  code and fix, 181
  detail outside code, 193–98
  error handling, 198–99
  hardware, 182–93
  LCOD, 181
  reuse, 200–201
  state machines, 199–200
  style, 201–3

Cases
  Control Display, 178
  Duplicate, 175, 176
Cases for UI controller, additional, 176
Checklists, 126–27
COD (Component Oriented Design), 33
Code
  detail in, 76
  detail outside, 76
  and fix, 157, 181
  reviews, 127–29
  spaghetti, 23
Code, abstraction in, 75–78
  section key files, 77–88
Code, detail outside, 75–78, 193–98
  section key files, 77–88
Cohesion
  bad, 25
  coupling and, 18–21

examples of, 22–29
good, 25–26
strong, 25–26
weak, 25
Complexities, software and its, 16–18
Component mechanisms, 47
Components
abstracting from requirements, 157–65
basic structures of, 55–57
components defined, 35–36
defined, 35–36
DIO, 205
LCOD (LabVIEW Component Oriented Design), 34–36
visual, 206
Construction, software, 120
Control actions, new, 205
Control Display cases, 178
Control Display While Loop, 177
Control >> Drive >> Read Pattern, 45–46
Controllers
additional cases for UI, 176
Program Executive State, 179
Coupling, 18–21, 22–24, 22–29, 24–25
Customers
acceptance of, 121
real, 113

DAQ (data acquisition), 4
Data flow diagrams (DFDs), 121–23
Definitions
software academia, 19, 21–22
strict type, 52–54
Derek's Software Co., 147–51, 151–52
project milestones and results, 151
software, operating system, and hardware, 147–48
stored data format, 150–51
test report format, 149–50
test scenario, 149
Design patterns, 40–43
dictionary definitions, 41–42
Design principles, software, 15–31
Design process, using patterns to help, 165–69
Designs
bottom-up, 40
bottom-up design, 40
design patterns, 40–43
good, 13
LCOD (LabVIEW Component Oriented Design), 36–46
Object Oriented Design (OOD), 37–39
pattern examples, 43–46
planning, 157–80
abstracting components from requirements, 157–65
building prototypes, 169–80
code and fix, 157
using patterns to help design process, 165–69
top-down, 39–40
top-down design, 39–40
Destination, agreeing on, 142–57
Acme Widget Co., 142–44
Derek's Software Co., 147–51, 151–52
example requirements meeting, 144–46
system output, 144
user input, 144
Detail
in code, 76
outside code, 76
Developers, advantages to, 12–13
DFDs (data flow diagrams), 121–23
Diagrams
Data flow, 121–23
homemade, 124–26
LCOD user interface example, 73–74
state transition, 124
Digital Input and Output (DIO), 26
Digital Signal Processing (DSP), 117
DIO component, 205
DIO Component Panel, 28
DIO (Digital Input and Output), 26
Display
hierarchy, 10
New History, 207
DMMs (digital multimeters), 159, 162
Documents, Requirements, 111–16
Documents, software architecture, 119–20
Drive actions, 188–90
DSP (Digital Signal Processing), 117
Duplicate Case, 175

Encapsulation, information hiding and, 21–22
Engineering, software, 105–30
Enumerated types, 48
Enumerated types, 101 things to do with, 48–52
case structure, 50
casting and uncasting, 50
firing sequence of events, 50–51
format into string, 49
scan from string, 49
state machine, 51–52
typecast, 49–50
Error handling, 88–92, 198–99
Config Ini Dialogue Diagram Initialize, 89
Config ini Dialogue Run State, 90
Error States Front Panel, 92
pass through error handling, 90
Example diagram, LCOD user interface, 73–74
Example requirements meeting, 144–46
Examples
pattern, 43–46
self-documenting, 136–37
state machine, 60–64
Executive states, program, 167–68
Executive, update History in test, 208
Expectations, managing, 114

Fairy tales, 106
Files, section key, 77–88
Close Config Data diagram, 79
Config Data Read From file after, 79
Config Data Read From file before, 78
Config Data Write To File after, 80
Config Data Write To File before, 80
configuration file functions, 77
new data manager changes, 87
Open Config Data diagram, 78
System Config Data Manager Front Panel, 81

System Config Data Manager Get case, 83
System Config Data Manager Invalid diagram, 82
System Config Data Manager Load All cases, 86
System Config Data Manager Load case, 85
System Config Data Manager Reset cases, 86
System Config Data Manager Save All case, 85
System Config Data Manager Save case, 84
System Config Data Manager Set case, 83
System Config Dialog Front Panel, 88
Fix, code and, 157, 181
Front panels, 137–40
general front panel standards, 137
icon and connectors standards, 139
organization of files, 139–40
private front panel standards, 138–39
public front panel standards, 137–38

General layout standards, 134
General Purpose Interface Bus (GPIB), 5
Get action, 82
Good cohesion, 25–26
Good coupling, 24–25
Good information hiding, 27–29
GPIB (General Purpose Interface Bus), 5
GUIs (Graphical User Interfaces), 43
GUIs (Graphical User Interfaces) design and prototyping, 64–74
LCOD User Interface example diagram, 73–74
stack queue component, 66–70
User Interface Control Wrapper VI, 70–73

Hardware, 182–93
after changes, 204
before changes, 204
control actions, 186–88
drive actions, 188–90
hardware actions, 182–86
read actions, 190–93
Hiding
bad information, 26–27
examples of coupling, cohesion, and information, 22–29
good information, 27–29
information, 21–22
Hierarchy display, 10
History Display action, Add, 207
History Display, New, 207
History in test executive, update, 208
Homemade diagrams, 124–26
Horror stories, 106–7

ICs (Integrated Circuits), 35
Information hiding
bad, 26–27
examples of, 22–29
good, 27–29
Information hiding and encapsulation, 21–22
dictionary definition, 21
software academia definitions, 21–22
Invalid state, 82

Journey, the, 141–209
agreeing on destination, 142–57
build, 180–203
given wrong directions, 203–8
planning your design, 157–80
planning your route, 157–80
requirements, 142–57

Key files, section, 77–88

Labeling standards, 135–36
Laboratory Virtual Instrument Engineering Workbench (LabVIEW), 1
LabVIEW 6i National Instruments, 24
LabVIEW Component Oriented Design (LCOD), 6
LabVIEW (Laboratory Virtual Instrument Engineering Workbench), 1
characteristics of, 9–11
sucks, 2–3
LabVIEW rocks, 7–13
advantages to developers, 12–13
good design leveraging advantages, 13
why it does rock, 7–12
Language, characteristics of software, 8
LCOD complementary techniques, 59–104
abstraction in code, 75–78
checking what comes in and what goes out, 92–98
detail outside code, 75–78
error handling, 88–92
GUI design and prototyping, 64–74
pre- and post conditions, 92–98
reuse, 98–104
state machines, 59–64
UI Controller >> Message Queue Pattern, 64–74
LCOD implementation, 47–57
basic structures of components, 55–57
component mechanisms, 47
message sending, 48–54
persistent local storage, 54–55
LCOD (LabVIEW Component Oriented Design), 6, 33–46, 181
components, 34–36
design, 36–46
LCOD user interface example diagram, 73–74
LEDs (light emitting diodes), 60
Lesson, classic OO, 37–39
Life-cycle models, 107
Life cycles
prototyping, 110
spirals, 108
waterfall, 107
Light emitting diodes (LEDs), 60
Load action, 84
Load All action, 85
Local storage, persistent, 54–55
Loop, Control Display While, 177
Loose coupling, 24–25

Machines
state, 59–64, 199–200
Managing expectations, 114
Meeting, example requirements, 144–46
Merge VIs, 99–103
Message sending, 48–54
101 things to do with enumerated types, 48–52
all about enumerated types, 48
strict type definitions, 52–54
Metrics, 130
Models, life-cycle, 107
Modification, VIs with little or no, 201

New control actions, 205
New History Display, 207

OOA (Object Oriented Analysis), 34
OOD (Object Oriented Design), 37–39
OO lesson, classic, 37–39
OO (Object Oriented), 2
OOP (Object Oriented Programming), 30
Opportunistic reuse, 98

Pattern examples, 43–46
Control >> Drive >> Read Pattern, 45–46
UI Controller >> Message Queue Pattern, 43–44
Patterns, 200
design, 40–43
using to help design process, 165–69
hardware actions, 168–69
hardware attributes, 169
program executive states, 167–68
UI, 165–66
UI attributes, 167
UI states, 166–67
Pictures tell a thousand words, 121–26
diagrams – data flow diagrams (DFDs), 121–23
homemade diagrams, 124–26
state transition diagrams, 124
Planned reuse, 99
Plans, test, 118–19
Post conditions. *See also* Pre- and post conditions, 95–96, 97
Postmortems, time for, 129–30
Pre- and post conditions, 92–98
conclusion, 97–98
Error State cases, 93
Error State diagram, 93
factory precondition, 95
postconditions, 97
pre- and postconditions example before, 94
preconditions, 95–96
Program Executive State Controller, 179
Program executive states, 167–68
Project is dead, 129–30
Project validation, 116–17
Properties, Strings[], 84
Prototypes, building, 169–80
designing User Interface, 170–80
Transfer of User Interface Pattern, 169
Prototyping life cycles, 110

Quote/project validation, 116–17

Read actions, 190–93
Real customers, 113
Requirements Document, 111–16
big picture, 114–16
constraints, 113
managing expectations, 114
real customers, 113
Requirements, the journey, 142–57
Reset action, 86–88
Reuse, 98–104, 200–201
of components, 43
cycle of, 43
of designs, 42
merge VIs, 99–103
opportunistic, 98
opportunistic reuse, 98
patterns, 200
planned reuse, 99
planning, 99
VI templates, 104
Reviews, code, 127–29
Routes, planning, 157–80
abstracting components from requirements, 157–65
building prototypes, 169–80
code and fix, 157
using patterns to help design process, 165–69

Save action, 84
Save All action, 84
Section key files, 77–88
Close Config Data diagram, 79
Config Data Read From file after, 79
Config Data Read From file before, 78
Config Data Write To File after, 80
Config Data Write To File before, 80
configuration file functions, 77
new data manager changes, 87
Open Config Data diagram, 78
System Config Data Manager Front Panel, 81
System Config Data Manager Get case, 83
System Config Data Manager Invalid diagram, 82
System Config Data Manager Load All cases, 86
System Config Data Manager Load case, 85
System Config Data Manager Reset cases, 86
System Config Data Manager Save All case, 85
System Config Data Manager Save case, 84
System Config Data Manager Set case, 83
System Config Dialog Front Panel, 88
Self-documenting example, 136–37
Sending, message, 48–54
Set action, 82
Soap box, 4–6
Software
academia definitions, 19, 21–22
architecture documents, 119–20
is not easy, 3
and its complexities, 16–18
weapons against complexity, 17
Software application, good, 15

Software Co., Derek's, 147–51, 151–52
Software construction, 120
Software design principles, 15–31
abstraction, 29–31
coupling and cohesion, 18–21
dictionary definitions, 18
software academia definitions, 19
examples of coupling, cohesion, and information hiding, 22–29
information hiding and encapsulation, 21–22
software and its complexities, 16–18
Software engineering
defined, 105
essentials, 105–30
checklists, 126–27
code reviews, 127–29
fairy tales, 106
horror stories, 106–7
metrics, 130
pictures tell a thousand words, 121–26
project is dead, 129–30
quote/project validation, 116–17
Requirements Documents, 111–16
software architecture documents, 119–20
software construction, 120
target specification, 117–18
test – customer acceptance, 121
test plans, 118–19
time for postmortems, 129–30
usual suspects, 107–11
Software language, characteristics of, 8
Spaghetti code, 23
Specification, 111
Specification, target, 117–18
Spiral life cycles, 108
Stack queue component diagram, 71
Standards
general layout, 134
labeling, 135–36
needing, 131–34
wiring, 134–35
State machines, 59–64, 199–200
definition, 59
State machines example, washing machine, 60–64
States
Invalid, 82
program executive, 167–68
UI, 166–67
State transition diagrams, 124
Stock queue component, 66–70
Storage, persistent local, 54–55
Stories, horror, 106–7
Strings[] property, 84
Strong cohesion, 25–26
Style, 201–3
Style, it's all about, 131–40
block diagrams, 134–37
front panel, 137–40
needing standards, 131–34

Target specification, 117–18
TCP/IP (Transmission Control Protocol/Internet Protocol), 50
Templates, VI, 104
Test executive, update History in, 208
Test plans, 118–19
Tight coupling, 22–24
Top-down design, 39–40
Transmission Control Protocol/Internet Protocol (TCP/IP), 50
Type definitions, strict, 52–54
bad things, 53
good things, 53–54
Types, enumerated, 48

UI, 165–66
UI attributes, 167
UI control diagram, 72
UI controller, additional cases for, 176
UI controller component
diagrams, 175
panel and connector, 174
UI Controller >> Message Queue Pattern, 43–44, 64–74
UI queue component
diagram, 174
panel and connector, 173
UI states, 166–67
UMLs (United Modeling Languages), 121
User Interface Control Wrapper VI, 70–73
User Interface, designing, 170–80
User Interface basic actions, 171
User interface example diagram, LCOD, 73–74

Validation, quote/project, 116–17
VB (Visual Basic), 9
Visual component, 206
VIs with little or no modification, 201
VI templates, 104
VI (Virtual Instrument), 9

Waterfall life cycles, 107
Weak cohesion, 25
Widgetometer, 143
Wiring standards, 134–35