**Training Material Release Notes**

**Release 7.0**

Release Date: ***4/14/2020*** Current VectorCAST Version: ***2020***

* Updated the Title Graphic to the PCT Standard
* Updated the Corporate Information Slides
* Updated some graphics and fonts for CI compliance
* Screen captures updated
* Updated QA lecture
* Updated System Testing VectorCAST/QA lab
* Updated C++ testing lab
* Updated Debugging stub code lab

**Release 6.4**

Release Date: ***4/19/2019*** Current VC Version: ***VC2019***

* This release is primarily for compatibility with VC2019
* Added a new diagram (slide 20) to represent the build process
* Changed all colors to be VI standard
* Made sure all screen shots were up to date
* Updated the Test Reports Lecture & Lab slides

**Release 6.3**

Release Date: ***11/12/2018*** Current VC Version: ***VC2018sp3***

* Created an “end results” manage project that will be delivered as part of the standard training material. This is so students can have a reference to go to when they get stuck with the lab. Currently, I am making no mention of this in the PPT slides. I just plan on telling the students verbally. Since the temp license for training has multiple seats, the students could have 1 instance of VC opened on TestCampaign, and the other opened on EndResults. I’m going to experiment a little here to see how this plays out in practice.
* Reformatted the slides to be “VI” compatible. This also allows printing of the Notes Pages, which was previously a problem.
* Added a new Advanced C++ lab on a Payroll System Class. Intent of this lab is to add more beef to the C++ content of the training class. Objectives of the lab:
  + Show that a Base Class will get stubbed automatically
  + Show that including the Base Class in the UUT is typically what you want
  + Show a class that points to another class (in this case, the Employee Class)
  + Show how to affect Visibility thru Custom Stubbing
* Improved the lab instructions for all labs
* The Advanced Stubbing lab has been improved.
* Took out Target lab, but kept the lecture slides. The target lab is out of date and is hard to do much with on a host environment anyways.
* Added links to new support email aliases
* Took out references to “classes of interest”. Instead, made mention of Custom Stubbing (slide 152)
* Fixed RGW “settings repository” reference on slide 448 to “RGW Database Location”
* Improved the Dishwasher lab diagram and Added new slide (extra credit)
* Changed the Object Initialization lab code to be a little more realistic by externing a “debug flag”, instead of some random integer.

**Release 6.2**

Release Date: ***9/14/2018***  Current VC Version: ***VC 2018 SP2***

* The name of the “advanced stubbing” lab has been changed to “stubbing fundamentals”. This is because it’s part of the Core class and not the Advanced Class.
* 2 parts of the Stubbing Fundamentals Lab have been changed:
  + The Airport lab has been changed with a “keypad” lab, to highlight the usage of “spanning a range test”.
  + The Function Pointer lab has been changed to a Temperature Gauge, where the user provides a stub of a Gauge that returns the temp.
* A new slide (14) has been added to give an example of how you might do Unit Testing Manually, and how this parallels VC.
* A new slide (17) has been added to give an overview of the separation between the Test Driver & the UUT, and the concept of an “event”.
* All the Speaker Notes have been reviewed and corrected.
* Updated ProbePoint screen shot on pg 464.

**Release 6.1**

Release Date: ***6/15/2018***  Current VC Version: ***VC 2018 SP1***

Not too much in this release, other than the new MC/DC lab. Other than that, mostly minor fixes, typos, and updated screenshots.

* The MC/DC lab had to change because there are no longer options for embedded logical operators. So, that part of the lab was deleted. I also deleted the part of the lab that dealt with “max conditions”. That topic can be addressed during consulting, if need be. I left the code commented out, just in case it is needed.
* As a replacement for what was taken out in the MC/DC lab, a new function was added (“open\_for\_business()”) with a corresponding lab. Please review this lab and provide feedback.
* Fixed spelling error in “menu.csv” (“NO\_DESERT” to “NO\_DESSERT”) . This file is not necessarily used in the training material, but it is a good example that you may want to use on your own for CSV import.
* Fixed a variety of typos etc…, thanks to input from Kurt K.
* Slide was added to introduce the local sales team (slide #3)

**Release 6.0**

Release Date: ***4/13/2018***  Current VC Version: ***VC 2018 SP1***

The primary purpose of this release is two-fold:

* Convert the over the old Vector Software slide template to the new VI template.
* Restructure the material into a Basic (2-day) and an Advanced (3-day) Class.

For the most part, the content has not changed, but there has been a lot of Reformatting, Re-organizing, & Rewording, as follows:

* Changed file from .pptm to .pptx to ensure that no macros are used
* Changed all references of Manage to “VectorCAST Project”
* There are now 4 pages of bubbles (or topics). The 1st 2 pages represent the material to be used in the Basic class. The 3rd page is for the Advanced class, and the final page are extra topics that can be used at anytime, or assigned as homework.
* Renamed Module A to Test Case Building
* Combined Module B&C into Test Case Fundamentals
* Renamed Module D to Compound Testing
* Renamed Module E to “Additional Test Case Options” and has been moved to a Miscellaneous Topic on the last page of bubbles
* Took out the numeric/hex part of the Strings lab. Too much time was being spent on the “string” topic during the lab.
* Added a “pointer” lab in “Test Case Fundamentals” on AddIncludedDesert to emphasize the concept of allocating pointers and accessing the resultant pointer
* Renamed Advanced Stubbing Lab to Stubbing Fundamentals. Content is the same.
* Took out the “Manage” module, primarily because we no longer use the term “manage”. This module needs to be re-worked and the content may be added back later. If this material is needed in the meantime, it can be used from the prior release.
* No changes were made to CSV Import, but any references to Create CSV Template have been removed. Almost no customers use the template option.

**Release 5.0**

Release Date: ***11/10/2017*** Current VC Version: ***6.4x***

* Improved some of the Quiz Questions
* Changed the “Cover” Lecture slides to use “QA” instead of “Cover”
* Added a new lab for QA

Added

**Release 4.2**

Release Date: ***08/09/2017*** Current VC Version: ***6.4u***

* Moved RGW launch bubble from the last of bubbles (pg 29) to the 2nd page (pg 25) because it’s related to Reporting
* Added Probe Points Training
* General Clean up on some of the Notes Pages

**Release 4.1**

Release Date: ***5/1/2017*** Current VC Version: ***6.4Q***

* Added Lecture/Lab for a module entitled “Troubleshooting the Build Process”. This is an optional lecture/lab listed on Additional Topics (pg 29)
* Added Lecture/Lab for a module entitled “Debugging Execution Problems”. This is an optional lecture/lab listed on Additional Topics (pg 29)
* Added a new function in “mcdc\_example.c” to show a trivial example of why MC/DC is important.
* Improved the CSV lab instructions and also took out the 2nd part for Creating a CSV template
* Took out Square.cpp from C++ lab (note that the lab didn’t really change, but we now include only “rectangle” and not “combo”). All of the source code is still the same. Combo & Square are still there if needed.
* In general, added & improved several test questions
* Added the following slides:
  + “Event” Definition (in Module A)
  + Manage Report (in Test Reports Section)
  + MC/DC slide that explains the minimum number of tests required to satisfy MC/DC (N+1)
  + Analytics slide in the Regression Module, and mentioned VCDASH as a cmd line tool
  + CBT slide in Regression Module
* Added Analytics to the Regression Lab
* Added Europe Support email info
* Added slide that talks about the Vector Acquisition. Slide is hidden (4th slide in deck) but is included in case a customer asks.

**Release 4.0**

Release Date: ***11/28/2016*** Current VC Version: ***6.4M***

* Took out all references to the old way of building test cases. Everything now is Manage Based
* Changed 1st lab to use Add\_Tax\_Tip(), instead of Place\_Order()
* Changed the order of the Dishwasher Lab. Pages 2 & 3 are switched.
* Changed order of the Advanced Stubbing lab so that the Object Init lab is now the last one. Easier to skip that way, if you want.
* Update the Cover slides & lab to reflect recent changes
* Added CBA lab to the existing Cover lab, with a new source file (cba\_example.c)
* In the Abstract Class lab, took out the declaration of the “dummy” class in Unit Appendix User Code, but left “#include <rectangle.h>”. There was too much data entry for the “dummy” class.
* In the Template Lab, there used to be 2 different declaration types for the “int” & “float” templates in Appendix User Code. That seemed to confuse a lot of students so it’s now been changed to use the same declaration type for both.
* Changed the wording and added a screen shot on the C++ lab. Content is basically the same.
* Added additional instructions for Communication.c lab (Void Pointer) and added a screen shot. Put the 3 User Code lines into the source code for easy copy/paste.
* Minor Rewording and typos. Added some screenshots to help with some of the labs.
* Updated the “start\_vc.bat” file to hopefully resolve some licensing issues.

**Release 3.0**

Release Date: 1/8/2016 Current VC Version: 6.4a

* The primary purpose of this release is to use Manage to create new Environments. The Legacy slides are still there, but now you have a choice of what path to go down, either the Legacy method, or Manage Method.
* In general, the content of the training slides has not changed too much. Most all of the existing slides are still the same. In summary, the changes are as follows:
  + Explain that envs can be created in one of 2 ways (slide 36)
  + Added the option on the 1st lab to create an env either with the new method (i.e. Manage), or the legacy way (slide 48)
  + Added a new lab “Building Envs with Manage” (slide 391…)
  + Combined the 2 old Regression Modules into one (slide 24 – see old & new options)
  + Added a new Module for regression testing with Manage (slide 172)
  + Added a new lab for regression testing within Manage (slide)
* Here is a little more detail on the changes:
  + The 1st Module (Environment Building) has been changed to include slides that talk about building Environments with Manage (slides 36 & 37)
  + Prior to this release, there were 2 Regression Modules (on slide 24). The 1st one was high level stuff and the 2nd one had more detail, plus a lab. I have condensed both of those modules into one, which is called Legacy Regression Testing, and is now 2nd of the 2 Modules.
  + The 1st of the 2nd Regression Modules is now for Regression Testing within Manage. Some of these slides are duplicated from the Legacy module, but there are new slides as well.
  + In the new Regression Testing Module, the “Test Script Editing” has been simplified. Basically, it just does a simple Export-Edit-Import process to get the general idea across.
  + The new Regression Testing labs show how to build & execute all Environments within Manage, both inside the GUI and at the Command Line. Also, Incremental Building (CBT) is shown as well.

**Release 2.2**

Release Date: 10/7/2015 Current VC Version: 6.3I

* Changed ctypes.h & manager.c to use new #defs for COST\_OF\_STEAK, etc…
* Changed a few screen shots that were out of date
* Various Typos, etc…
* Changed “use\_function\_ptr” in Advanced Stubbing to return an enum
* Changed the way “newseconds” was declared in vcast\_time.c because it was causing a compiler error on some compilers

**Release 2.1**

Release Date: 9/1/2015 Current VC Version: 6.3g

* Took out hidden page 9
* Deleted black box on page 8
* In the labs for vcast\_time & advanced stubbing, the Part #’s were misleading, so that has been fixed.
* In the lab for vcast\_time, there was references to Step 5, Step 6, etc…, which really meant Steps 5/6 of the New Env Wizard. However, some students were thinking Step 5/6 of the lab instructions. That has been re-worded to be more clear.
* Pages 314 & 348 were affectively being hidden because they were not part of the corresponding “show”. That has been fixed.
* Changed the Advanced Stubbing Lab instructions to take advantage of “myFunction” now being in the pull-down list, instead of having to write User Code. This was on page 430, part 1.
* Added “menu.csv” to the SupportFiles subdirectory. This isn’t referenced in the PPT slides, but it’s nice to have if you ever want to do a CSV demo based on the Restaurant theme, as opposed to “Line.c”.
* Moved prototypes for Get\_Table\_Record & Update\_Table\_Record from manager.c over to a new file database.h, where they more logically belong.
* Checked return code from “log\_tax\_receipts” so it can effectively be stubbed
* Changed “datalink\_transmit.c” so it can be portable across all compilers.
* Added some slides in the Best Practices section that show examples of accessing global data
* Added some slides in the Best Practices section that talk about some of the perils in using pointer variables.
* Cleaned up instructions for the Void Pointer lab (communication.c). Students should now be able to cut and paste the 3 lines that need to be added to User Globals. This hopefully will avoid some of the compile issues that we typically run into on this lab.
* Added a new batch file (start\_vc.bat) which can be used to possibly alleviate some of the hassles of installing the training license. However, this batch file assumes that all students unzip SourceFiles.zip into “c:\VCAST\Training”.

**Release 2.0**

Release Date: 6/1/2015. Current VC Version: 6.3D

Although the material has been upgraded for 6.3, the material could still be used for any customer that needs to use 6.2x as well.

* We are now using release numbers to identify the Training Version and have arbitrarily selected 2.0 as the version number.
* The Cover lab has been updated to show integration between Cover & UT. Note that this is the lab for the Cover tool, not Coverage in UT.
* The source code for the Advanced User Code Lab and Best Practices has been changed. The new names of the modules start with “data\_link\_”. The old names started with either L1 or L2, to represent Layer1 or Layer2, but that kind of got confusing. Now the code refers to the “packet”, “byte” & “hardware” layers. There are 5 files that are used in these labs: datalink\_receive.c, datalink\_transmit.c, datalink\_hdwr\_rx.h, datalink\_hdwr\_tx.h, & datalink\_types.h.
* Resolved all items in FB 41873

**03-31-15, 6.3B**

* Change Advanced Stubbing code slightly on the “use\_const\_global” function. The old code tried to show that you could get 100% statement coverage, but not 100% branch coverage. Most students got lost on this point, though. So instead, I’ve changed the lab to use Branch Coverage Only and then show that you can only get coverage on 1 branch at a time when you are using Global Constants.
* Add the following additional modules. The intent of adding these modules is 2-fold: 1) If you have an advanced class and you breeze thru the material, you now have additional material, and 2) Some customers don’t allow you into their work-environment on the 3rd day of class, so you can now use this additional material to stretch the class out.
  + Cover, Lecture & Lab
  + Manage, Lecture & Lab
  + Requirements Gateway, Lecture & Lab
  + Advanced User Code, Lab Only
  + Best Practices of Testing, Lecture & Lab

**01-26-15, 6.2l**

* Modified “include paths” lab
  + Changed directory name from “includepath” to “SearchPath” to eliminate confusion over subdirectory called “include”
  + Added an external reference to “invalid\_time” to show how the stubbing of that function can change when you move from a Search directory to TypeHandled or LibraryInclude
* Took out “return” statement in “Utilities.c” because it was interfering with the code coverage when using the “cover default case stmnt” option.
* Resolved all items in FB 38717

**01-5-15, 6.2l**

* This should be the last major release before 6.3. Most of the changes below were minor changes that have been collected over the last year. Almost all screen shots that were out of date have been updated. Here are the specifics:
* Have taken out most (if not all) of the timer widgets. This was used to time the labs, but there were a few problems. For one, it could get in the way of the text on the slide, but also, you had to have the proper add-ins in Power Point to make it work and not everyone has those add-ins.
* Pg 339 – Enter invalid value for Table, not Seat. Invalid Seat can cause an exception, but Table does not (the Invalid Seat example that causes an exception is covered later in the Coverage lab).
* Screen shot on bullet 2, pg 72 was out of date
* Update manager/database with new code changes. PlaceOrder now returns the check total directly. Also, a new function called AddTaxAndTip() has been added as a simpler example for newbies.
* Pg 344, make it clear of the flow from pt1 to pt2
* Pg 388, take out “user code tag” referenced, which was misleading
* Pg 403, select MCDC + statement…
* MCDC lab, part 2, pg 404, needs to be re-worked. The last part of this section was worded poorly and I think missed the intent of a “Large Conditional”. The lab has been updated.
* For Abstract Class lab, and for combo/shape, make it clear to add additional paths & to select “make these default for wizard”.
* Add page #s to pp 375…
* 2 new source code modules have been added to the Source Code. One is called “testability.c” which can be used to show how to make testing easier by separating control & data logic. Also, a moduled called “l2\_dev\_driver.c” has been added to show how to make testing of low level hardware easier. Currently, no labs reference these modules, but they can be useful to show examples of “best testing practices”. In 6.3, we will probably add a lab that uses these modules.
* Cleaned up the use\_function\_pointer( ) function in the Advanced Stubbing lab to make it a little easier to follow.
* Add more detailed instructions on how to use SBF
* Added instructions on how to add include path for “date” lab (advanced Code Coverage). And same for combo/shape in C++ lab
* Polished up lab instructions for Code Coverage Options , Part 2, 1. "Get code coverage for function: TRACE"
* All of the references to "delete all existing test cases" have been taken out. This is most unpleasant to students because they'd prefer to keep the existing test cases as reference for later. Also it is in most cases not really necessary. I always tell students they don't have to and it doesn't cause trouble.
* Change name of float\_example function to not be same as module, just to make it a little clearer.
* On 1st c++ lab (shape/combo), needed to add a note to add include path
* **Display global data options was referred to as “check” global data options on the slide 89**
* **Added “manager\_driver.c” so you can build a Cover Example for Cover training. Also added a build batch file to build the executable (bld\_cover.bat).**
* The following screen shots have been updated:
  + Slide 21
  + Slide 41
  + Done! Slide 43
  + Done! Slide 52
  + Done! Slide 53
  + Done! Slide 56
  + Slide 69
  + Slide 72
  + Slide 97
  + Slide 299
  + Slide 136

**01-29-14**

* We are now going to distribute the PPT Slides in PDF version, as opposed to the Flash Executable
* Changed page numbers in Master Layout so that the show up on all pages
* Changed the Lab for Module C to no longer do Cut & Paste (this is done in Module B). Also, used the Add\_Party\_To\_Wait\_List to show the differences between String, Char, & Numeric modes (instead of PlaceOrder).
* Changed the Advanced Stubbing lab to reflect the new way we handle Library Stubs
* Added Basis Path Slide & lab to the Code Coverage section
* Beefed up the code for the Environment User Code (Communication.c). Students will now have to fill in the structures pointed to by the void parameter.
* Took out the extra credit lab of the Include Paths sections. There are a few problems with this lab, so I've take it out. In order to get it to compile, I had to add the include path ../mingw/lib/gcc/mingw32/4.5.2/include. Also, it's not clear why you have to change it from UUT to SBF. Also, the instructions are confusing with 2 different references to "include paths" which mean 2 different things. The basic concept of this extra credit lab was done in the previous exercise, so I don't know that it's necessary. If we do want to resurrect, the slides can be found in the previous slide deck (dated 10/30/13) – ccp

**10-30-13**

* Miscellaneous updates, typos, etc…

**08-26-13**

* Miscellaneous updates, typos, etc…

**07-08-13**

1. Abstract Class module has been re-implemented to reflect the changes in 6.1. The user no longer has to create a concrete class in order to test an Abstract Class. The tool will now do that automatically.
2. Absolute page numbers are now displayed on all pages, however, it is obscured on some pages due to the bottom banner. The reason for adding this is so that the students will now exactly what page to go to in their copy of the material.
3. The 1st 5 sections have been changed from “parts 1-5” to “modules A-E”
4. All labs have been labeled accordingly
5. Added floatAbsValue routine in float\_example.c. This is not used in the lab, but it might be an interesting way to show the difference between MIN & MAX. It’s a little more interesting than the existing function (floatExample), which just returns the input param.
6. io\_wrapper.c has been changed to always return error codes when fopen() fails
7. io\_wrapper.c has also been changed to hide the FILE pointer inside the FILE\_TYPE structure. It is now declared as an unsigned int. This is for the purposes of eliminating confusion between the 2 different file types.
8. Modified control\_flow.c to be a little more interesting. It now has some real shapes to operate on (BOX, CONE, CYLINDER). Also, drag & drop works a little differently now, so the instructions have changed.
9. compound\_and\_init.c has been changed to dishwasher\_fsm.c. Slides for lab have also been changed. The main intent behind this new exercise is to give the student a chance to solve a problem, instead of just following instructions. Having a real-world examples should help with that goal.
10. Some of the Quiz questions have been improved:
    1. Test Case Bldg Mod B
       * Old question was regarding how to set the base of a number. New question is about Floating Point Tolerance
    2. Code Coverage Basics
       * Old question what the specific McCabe # was for some example. New question asks for the defn of McCabe Complexity
    3. Testing C++ Classes
       * Old question was about which class was highlited in yellow. It has been changed to a question about what happens when a class is not specified.

**10-17-12**

* This is the original version of the reorganization effort.