

**INSIDE**

---

**Economics of Code Refactoring**

Refactoring old code is a cost-effective method of reducing long term IT budget dollars

---

**Control/DCD 2.4**

The latest Release of Control/DCD introduced new features such as Enhanced Compile Mode and the Special Narrative

---

**Announcing New Webinars for 2022**

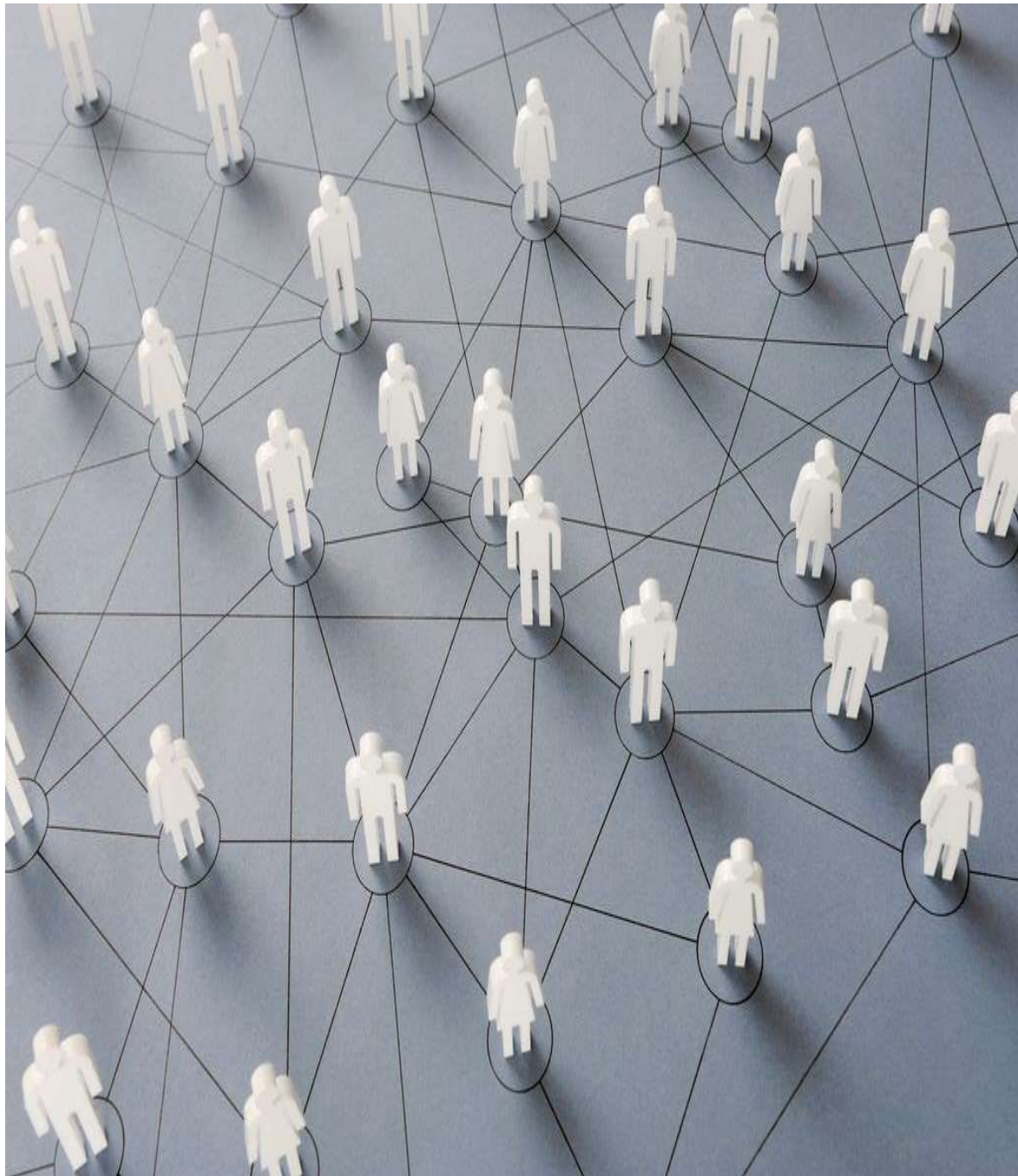
Marble Computer will provide 1-hour webinars on Code Refactoring and enhancements for COBOL Version 4.2.

---

**The Observer**

Thoughts From Marble Computer, Inc. CTO  
Marshal Crawford

---



# The Economics of Code Refactoring

When deciding to refactor or not, you should consider the costs, risks and benefits of doing the refactoring. This will tell you when it's worthwhile to start refactoring, and when it would be wise to stop.

## **Costs:**

Cost of Doing the Refactoring:

This is the cost of changing the code: renaming data fields or changing copybooks; copying code and moving action statements to align with the logical program flow. Removing dead code; extracting common or reusable code to a separate location within the program; or any of the other valid refactoring moves. Typically, this cost is low, as it's not hard to make a few syntactical changes to code. Use of IDE software reduces the time and cost of refactoring code.

Cost of Testing the Change:

Any change you make to the code may introduce bugs, and so must be thoroughly tested. With a complete set of automated regression tests, this cost will be very low because you're leveraging the testing investment made when the tests were written. However, without automated testing, the testing costs can be significant.

Cost of Updating Tests and Documentation:

Refactoring may change the current set of testing protocols, or even introduce new requirements. The development of a new test bed and user classes may take development time. This can be a non-trivial cost to projects that maintain a comprehensive set test beds.

## **Risks:**

Risk of Introducing Bugs (not caught by testing):

If regression testing falls short of perfection, there's a risk that bugs introduced during the change may be released into production, causing loss of business data. The political repercussions of such a slip can be severe. However, if your test beds are good this risk is minor.

## **In Summary:**

Cost and Risk of Refactoring tend to be low, as most IT organizations have good regression test software. Also, the "size" of refactoring projects tends to be smaller, as the code tends to maintain good conceptual structure, even in the face of change.

## Benefits:

Adding new features no longer corrupts the system's structure:

Optimizing the maintenance process to minimize the visible changes to the source code must, as with any optimization, compromise something else: In programming, what gets compromised is the readability, maintainability, and logical structure of the system being maintained. So, over time, a system maintained without refactoring will become unmaintainable, and must be scrapped and rewritten from scratch. Refactoring avoids the costs of working with unmaintainable code, and the eventual cost of replacing the system.

Improves programmer's understanding of the system:

Refactoring produces shorter simpler methods. It's less work to understand the smaller amount of code that needs to be changed to implement any given function.

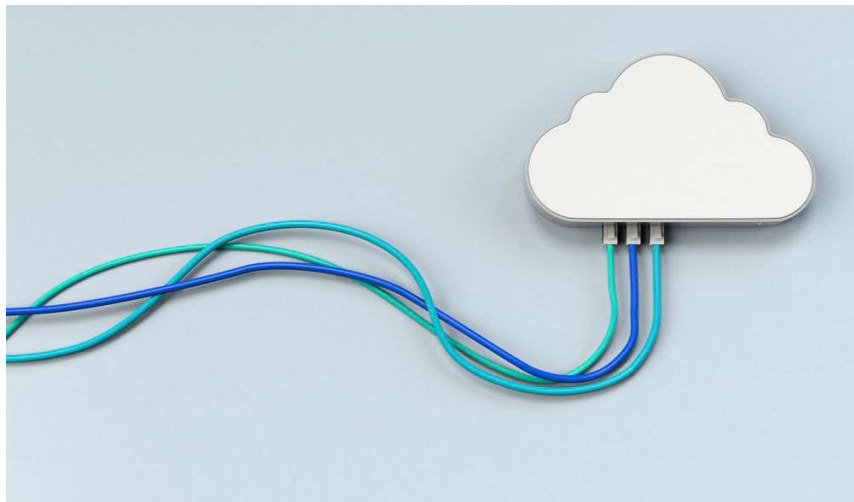
Refactored code is easier to test if that was one of the goals of the refactoring.

Well Factored Code is easier to maintain!

It's

- easier to add new functionality to
- easier to test
- easier to find and fix bugs in

And *easier* translates directly into *faster* ... fewer man hours, costs less. Spending a little overhead refactoring all the time means an overall success rate increase.



## New Webinars For 2022

**Marble Computer will be hosting 3 Webinars for Q1 2022. We will be focusing the webinars on Code Refactoring and COBOL Release 4.2 as that compiler goes End of Service on April 30, 2022.**

Tentative dates and times are as follows:

Thursday January 27, 2022, 11:00 AM PST - Code Refactoring

Thursday February 24, 2022, 11:00 AM PST - Code Refactoring

Thursday March 24, 2022, 11:00 AM PDT - The fate of COBOL Version 2.4. and the use of Enhanced Compile mode

## Control/DCD Release 2.4

The latest release of Control/DCD has added new features that are monumental design changes for current and future customers.

Enhanced Compile Mode - adds a hierarchal flowchart, new COBOL Analytics and a "Special Narrative" for data field tracking.

Independent Mode - produces the Control/DCD reports and analytics without a COBOL Compiler.

A complete reformatting of significant reports to fit within the 80-column limits of ISPF.

Summary reporting on programs that may need modifications for migration to COBOL compiler 6.2 or 6.3. This includes Invalid Data errors, Call Parameter arguments, missing initialized values, Packed and Binary Data issues.

Please contact sales at 800-252-1400 for more information.



## The Observer

Marble Computer is celebrating 39 years of commitment to the Data Correlation and Documentation software product referred to as DCD. During the past 39 years we've modified the original code several times with new releases for DCD II, DCD III, DCD IV, and Control/DCD. Control/DCD Release 2.4 represents a 2-year undertaking. We've modified the look and feel of the product. We've added several new features that support Code Refactoring. Control/DCD's output now fits on a single ISPF page. We've integrated our analytics and narrative within the standard COBOL compiler listing. Control/DCD Release 2.4 represents a major investment for our company, our customers, and future clients.