ELLIOTT PRECHTER
EWAVES Flash
Published by Qualitative Analytics

January 2015 Issue

ASSUME NOTHING. TEST EVERYTHING.

January 8th, 1989. British Midland Flight 92—a Boeing 737-400—departs London for Belfast, Northern Ireland. At 28,000 feet an explosion occurs. Vibrations rock the jet and sparks emerge from its side. Smoke enters the cabin. One of the plane's engines is failing.[1]

Experience tells the pilots that the cabin smoke must be from the right engine, because that is where the ventilation system was located in previous 737 models. But their experience led them to make a fatal assumption, because these pilots were flying a new model. They throttle down and shut off the right (*wrong*) engine. Then they increase throttle to left engine, which bursts into flame. With no thrust, Flight 92 is too far from the nearest runway to land. The plane crashes, and 47 of the 118 passengers die.

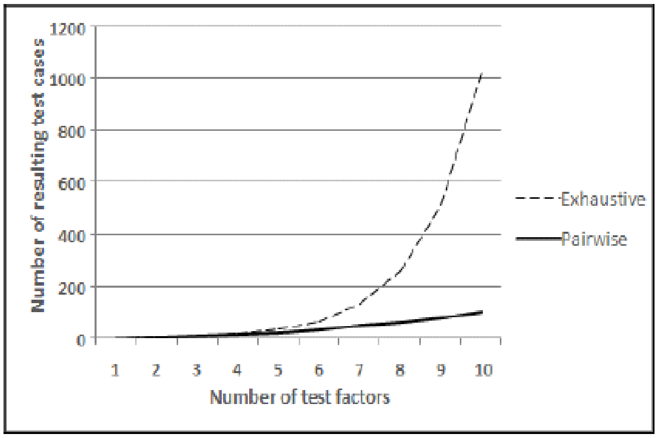
As with all airplane accidents, the crash resulted from a combination of factors, a confluence the aviation industry calls the *error chain*.[2] There were two main issues with Flight 92: First, the plane's engine design had a flaw that caused a mechanical failure (a fact revealed later after similar 737-400 failures). Second, the pilots were not trained for the new model design, which directly led to the fatal disabling of the wrong engine.

Aviation personnel had performed *insufficient testing* on the 737-400 prior to Flight 92, hence the two main failures in the error chain. Boeing engineers did not properly test the engines in flight, and the pilots were not trained (tested) in the new aircraft design. Had either of these crucial tests been performed, it is fair to say those 47 passengers would have survived.

Testing as Philosophy

Testing saves lives. When I say "testing," I don't just mean "try it and see if it works." Testing is a commitment to quality, a best practice, even a philosophy. When you make the product others will depend on, your tests should be highly demanding. Proper testing typically involves four primary considerations:

First, systems must be *testable*. Engineers work to create a "test-driven design." This ensures that testing is integral to design and construction, not an afterthought. Engineers build *testable* components with dedicated mechanisms that allow them to examine a component's internal state for defects. They then test each individual component before assembly with other components. Next they test groups of components, from small to large, to see how they interact. Finally they test the complete system. This bottom-up testing hierarchy can verify the entire system's integrity.



"Pairwise" is a statistical technique to emulate testing all possibilities, but using far fewer total tests.

Second, engineers must test systems under as many combinations of external and internal conditions as is feasible. It may not be possible to test all potential scenarios, yet they can use statistical techniques to simulate completeness.[3]

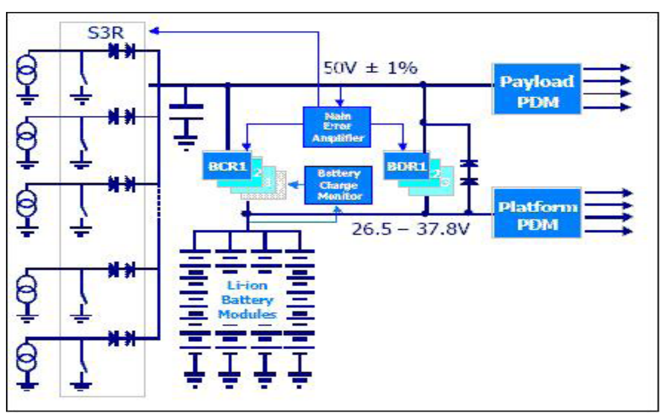
Third, engineers employ redundancy techniques to prevent a *component* failure from becoming a system failure. This is why modern jets are always designed to fly even when a single engine fails. Despite the best design and most rigorous testing, no system is perfect—so checklists, routines, backup plans, and consideration of human factors are also crucial to the safety and integrity of a fully robust system.

Finally, testing will verify a design's *implementation*, as well as its *legitimacy*. Test outcomes—instead of fallible opinions—are the final arbiter. It works in theory only after it's proven in practice.

Aerospace Does It Best

Errors of the kind that doomed Flight 092 are rare in the aerospace industry, which is widely regarded as the world leader in test standards and practices. A friend of mine who formerly worked as an engineer at NASA and Boeing informed me that both companies often employ *three* levels of redundancy in their designs. Separate teams construct each level of redundancy to reduce the chance of unilateral defects.

"Lockstep" is a common and powerful redundancy scheme. Multiple implementations of a computer chip perform the same operations simultaneously. If all implementations do not produce the same results, a majority vote arbitrates. With this level of fault-tolerance standard in the aerospace industry, it's no wonder that Superman was able to tell Lois Lane "statistically speaking, it's the safest way to travel."



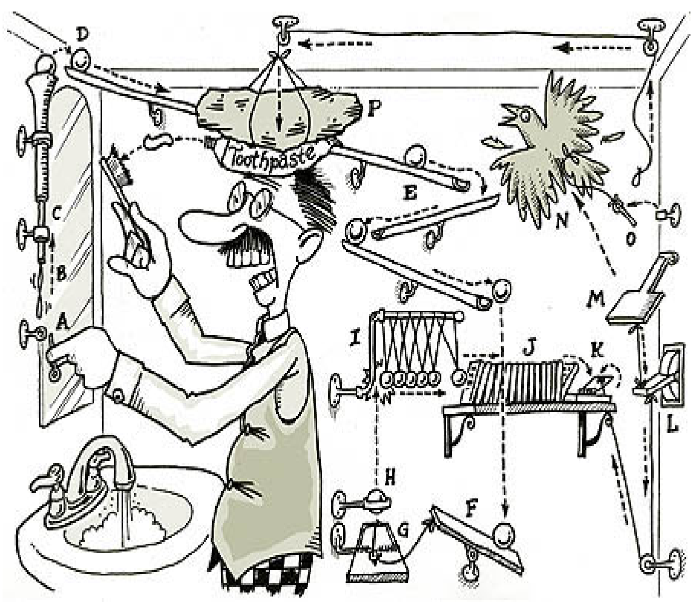
"Voting" mechanism on The GIOVE-A spacecraft. Notice the multiple layers of BCR control chips, each one implemented separately and "voting" to achieve fault-tolerance in their outputs. [4]

The Problem with Software

Software is notoriously error-prone. Current industry estimates are 1-5 bugs for every 100 lines of code.[5] (The first "bug" originated in 1946 when a moth trapped in a relay caused a malfunction in the Computation Laboratory at Harvard University.[6]) Eliminating bugs accounts for most programming work.

Bugs have a huge range of effects—from the annoyance of a non-critical program freezing to the potentially crippling security breach that gives a hacker access to sensitive information. The worst-case scenarios are life threatening. Perhaps the most notorious example is the Therac-25 radiation therapy machine. Between 1985 and 1987, a number of patients died when the Therac-25 erroneously gave them *100 times* the normal dose of radiation. The cause was known as a "race condition," a bug that is difficult to diagnose since it depends on precise timing factors. In this case, the typing speed of the Therac-25 operator was a crucial influence on whether the bug would appear.[7]

At first it may seem odd that software is so error prone, since its foundations are theoretically pristine.[8] Like literature, code does not operate in a physical form, but rather in the world of thought-stuff. Unlike mechanical systems, software is digital and does not depend on temperature, lubrication, metal fatigue, etc.



The combination of simple primitives can create infinite complexity

Yet errors in code remain prolific, for the same reason typos in books remain even after proofreading: Writing code is a human endeavor. But unlike editing, even "minor" coding errors are magnified by the overwhelming complexity of large software systems. The human brain has strict limits on its ability to effectively integrate many moving parts. Windows 8, for example, consists of some 80 million lines of code. No individual can know the whole project.[9]

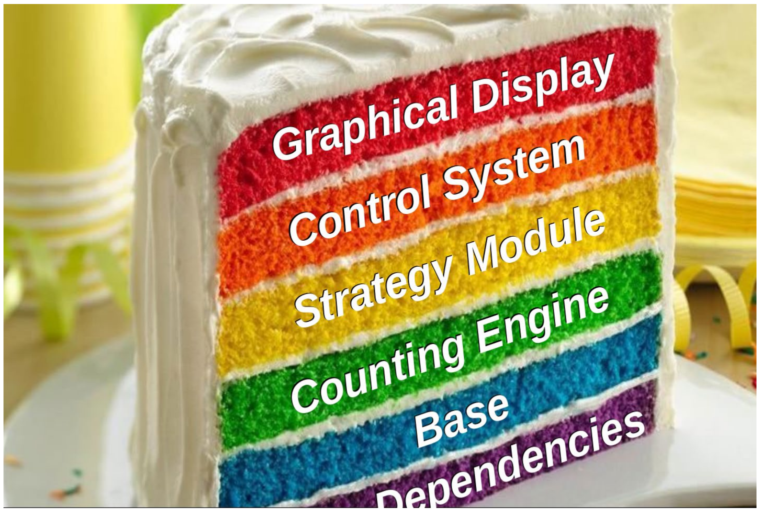
Programmer overconfidence also makes bugs prolific. I have met many maverick programmers who believe their code is "blessed" beyond the need for testing. Without exception, those individuals ranked among the worst developers and their code included the most bugs. Fred Brook's famous book, *The Mythical Man Month*, has a theory on why this is so:

> All programmers are optimists. Perhaps this modern sorcery especially attracts those who believe in happy endings and fairy godmothers. Perhaps it is merely that computers are young, programmers are younger, and the young are always optimists. But however the selection process works, the result is indisputable: "This time it will surely run," or "I just found the last bug."

Testing EWAVES

Much as drawdowns are inevitable to money managers, bugs are a fact of life to programmers. Recognizing and dealing with them is the only path towards success. Therefore, we employ numerous techniques to ensure that EWAVES 2 is as robust as possible.

Layering is a powerful concept in test-driven design. We are developing EWAVES 2 with a layered architecture. The lowest layer is the *base,* which handles the most primitive operations, such as dealing with memory, reading market data, performing mathematical operations, etc. The next layer is the *counting engine*, which uses the Elliott Wave Principle to label wave forms in market data. The *strategy module* rests on top of the counting engine. It is responsible for using the machine's Elliott wave analysis to produce Flash recommendations. The *control system* is next, and is responsible for scheduling each analysis across multiple



CPU cores (see the the April issue of *EWAVES Flash* ewaves.com/1404EWF). Finally, the topmost layer is a *graphical user interface* (GUI), which is used both for human-level interaction with the control system and as a visual tool for navigating through charts and looking at the counts and recommendations. The GUI is a completely separate program, so it can communicate with the control systems of multiple individual computers running EWAVES 2. The ability to command an "army" of EWAVES 2 machines from a single GUI allows us to have supercomputer-like power when it comes to running tests on large amounts of market data.

Layers obey a simple hierarchy. Any changes to a given layer can only affect layers above it. For example, a change to the graphical display will not alter the counting engine or its subcomponents. Our layered design simplifies our ability to think about the system, test each layer thoroughly and independently, and manage the propagation of changes to the system.[10]

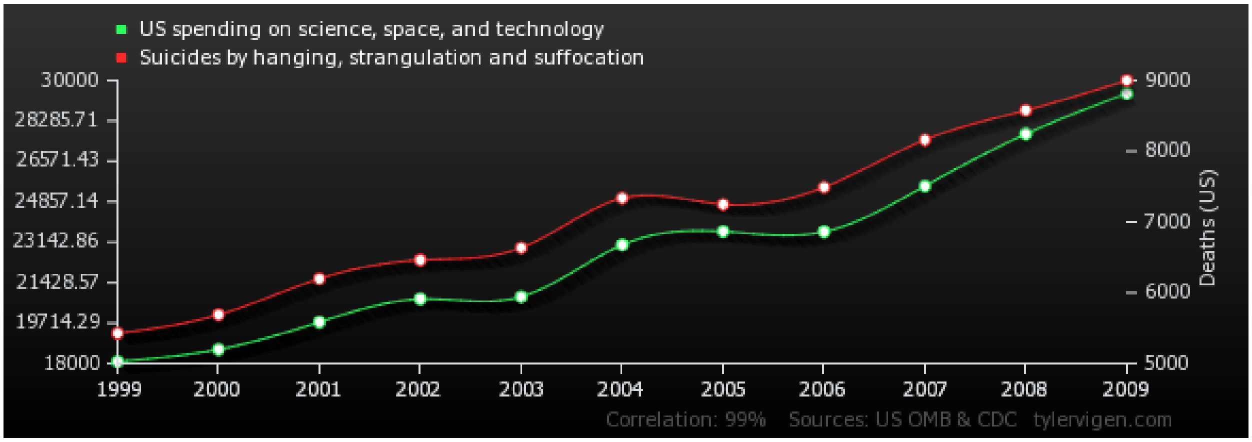
Another technique we use heavily in EWAVES 2 is *Don't Repeat Yourself* (DRY), which states that knowledge should be expressed only once in a given project.[11] The opposite of DRY is *We Enjoy Typing* (WET). A WET project may require a writer to keep separate documentation updated and in-sync with the rest of the project. EWAVES 2 is a DRY project, so we document *within* the project as we write and test the code. This allows us to produce project documentation on the fly, with graphical charts from the tests. This procedure automatically keeps our documentation synchronized with the rest of the system. The DRY principle leads to clean and robust designs, and may have been first described by theologian Thomas Aquinas in the 13th century: "If a thing can be done adequately by means of one, it is superfluous to do it by means of several; for we observe that nature does not employ two instruments if one suffices."[12]

Most important, we maintain a suite of tests for every component and layer of the system. Each time we modify EWAVES 2, we run the full suite of tests to ensure that nothing breaks. Our software development team must write some of the tests, but increasingly our Elliott wave analysts can build tests visually by using our specialized EWAVES 2 test tool. This tool allows us to point-and-click to create wave patterns and associated metadata. In turn, these patterns serve as tests for the counting engine.

For example, one of our analysts plotted an impulse wave where wave 2 goes below wave 1. He then saved the chart as a test, including information stating that the labeling scheme is invalid. This test, generated entirely visually, is then automatically converted into code that tests the EWAVES 2 counting engine to ensure that it does not violate this basic rule of the Elliott Wave Principle. At present we have several hundred tests. We expect to carry out at least 3,000-5,000 tests by the time we release EWAVES 2.

Test before You Trade

Too many financial programmers are lax about testing. They accommodate a natural bias toward easy answers—and easy money. The most flagrant violation of proper testing practices is *curve-fitting*. This practice involves finding a mathematical function or model that fits past data, and leads many quantitative analysts to a false sense of having discovered something useful. They may feel secure because of their backtest results, but curve-fit results rarely maintain into the future. As a concrete example, the graphic on the following page demonstrates how powerful mathematical correlations in data may appear if enough data are examined. Yet it should be obvious from the lack of a causal hypothesis that this correlation will likely fail to provide any future predictive value. Finding these kinds of spurious artifacts is the common result of quantitative curve-fitting techniques. Especially when it comes to financial markets, they simply *do not work.*[13]



Compounding the problem is that many incorrect ideas *sound* logical. An especially common trading myth—"you can't go broke taking a profit"—results from what David Aronson, author of *Evidence Based Technical Analysis*, called "the crime of small numbers." Although this myth seems logical after a winning trade turns sour, a strategist can't universalize a principle from a sample size of one. Hindsight bias clouds this statistical truth.

Sometimes it is best to "take a small profit," but at other times it's better to "wait for a big gain, and take small hits in the meantime." It all depends on what your method says to do; but only if the theory behind the methodology is sound and tests confirm it.

The opposite of the crime of small numbers can be a problem in the wrong hands. Too many samples allows cherry picking to find those that fit expectations.

As an example, a friend once recommended a quantitative breakout strategy to me. A breakout happens when prices move outside of a range, triggering a buy signal. He showed me numerous examples of its efficacy.

What my friend didn't realize is that a breakout of some kind can nearly always be found in retrospect for a winning trade. By cherry picking, he ignored the fact that many breakouts lead to immediate reversals and therefore losing trades. The real question is: Do the winners outpace the losers over time? After testing his system I found the answer was no.

Of course, even when test results look good in sample, quantitative systems always eventually burn out on out-of-sample data, as discussed in the first issue of *EWAVES Flash*. But *qualitative* analysis—based on EWP—is robust. As the previous issue of *EWAVES Flash* discussed, we performed extensive testing of the new EWAVES 1.1 before putting it into action. 1.1 produced similar results both in *and* out of sample, a key requirement for us to release it. (After all, the future is all out of sample.) To view these test results, please visit ewaves.com and select the July 2014 issue of *EWAVES Flash*, or go directly to ewaves.com/1407EWF.

Bitcoin

For those of you who subscribe to Elliott Wave International's *The Elliott Wave Theorist*, you may have read some of my guest articles on the Bitcoin cryptocurrency. The technology has revolutionary implications for low-overhead banking, protecting savings from inflation, making micropayments, transferring money internationally, and even practical crypto-anarchism.

Regardless of these features, Bitcoin has also been the focus of extreme financial speculation. The intense crowd emotions coupling its rise and fall form potent Elliott wave patterns, demonstrating the efficacy of EWP so clearly that I decided to reprint my 2013 forecast and the ensuing price action. As far as I can tell, there is no methodology other than EWP than has this kind of potential to reveal the financial future, which is why we are so hard at work on building EWAVES 2.

(And yes, for those of you who are asking, when EWAVES 2 launches we will consider tracking cryptocurrencies.)

Bitcoin Bubble or Bitcoin Breakthrough? How about both?
by Elliott Prechter (Excerpt from The December 2013 *Elliott Wave Theorist*)

EWT discussed Bitcoin for the first time in August 2010, when the currency traded at six cents. As far as we know, EWI was the first financial publisher to discuss it. Bitcoin was unknown to the general public and off investors' radar. Even the earliest adopters did not take it as seriously as they should have. The most notable example of this is the man who paid 10,000 BTC for a pizza. This pizza purchase is now famous (https://bitcointalk.org/index.php?topic=137.0), and many continue to track its price in USD terms via the "Bitcoin Pizza Index," which recently hit an all-time high of over $12 million.

Fast forward to today, and the currency is regularly featured in financial news and social media. Bitcoin Magazine has become popular, Congress is holding hearings on the currency, Germany has defined its role in finance, China is ruling on its legality, and the business world is adopting it. The most prominent business to embrace Bitcoin is Virgin Galactic, one of the many creations of billionaire Richard Branson (http://www.cnbc.com/id/101220710).

EWT readers were prepared for all this. When Bitcoin was still in the shadows, the August 2012 issue said,

> Presuming bitcoin succeeds as the world's best currency—and I believe it will—it should rise many more multiples in value over the years. —*EWT*, August 2012

The big question on the minds of investors is not what Bitcoin *has* achieved, but should they buy Bitcoins *now*? It's amusing that so many people ignored Bitcoin upon hearing about it in 2010, but now that its price has gone up 20,000 times, they want to invest. Notwithstanding the currency's potential, this shift in attitude is a signal saying now is not the time to buy. Let's look at four areas of evidence:

1) Optimism is off the charts. Past issues of *The Elliott Wave Financial Forecast* discussed people selling their homes and borrowing money to invest in Bitcoins. That was near the peak of wave ③. Now the desire to buy has grown even more extreme. Bloggers are calling for Bitcoin to reach $1 million…*soon*. One young investor borrowed a million dollars from his father and without his knowledge invested it in Bitcoin (https://bitcointalk.org/index.php?topic=359228.0). The other day I walked into a convenience store wearing a Bitcoin T-shirt, and the owner asked me if he should invest now. I felt like I was living in 1929.

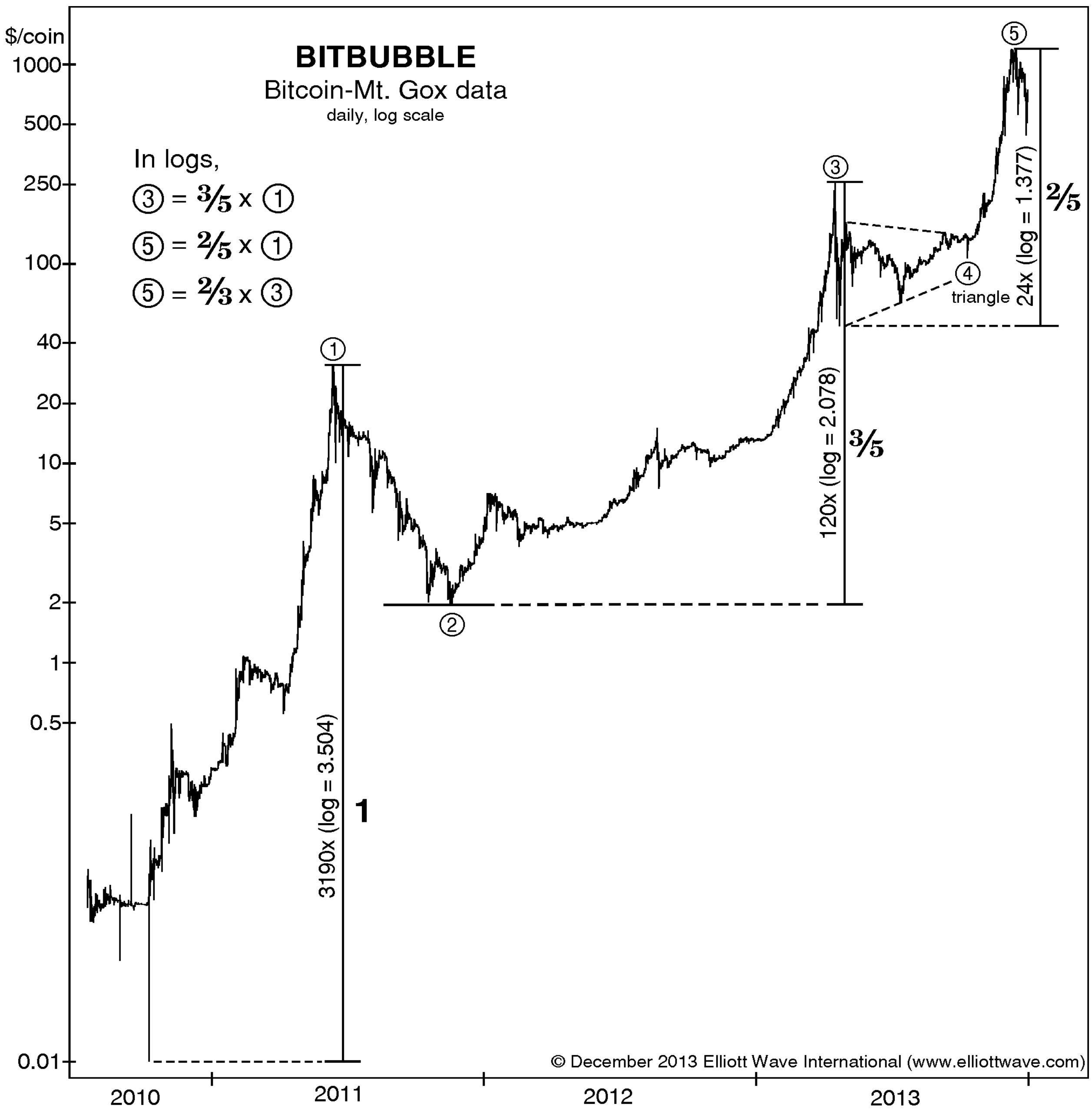
2) Investors have recently been rushing to buy a rash of 95 (at last count; see https://bitcointalk.org/index.php?topic=134179.0) new clones of Bitcoin that have recently emerged: Litecoin, Namecoin, Zerocoin, BBQCoin, PPcoin, PrimeCoin, NovaCoin, FeatherCoin, TerraCoin, Devcoin, Megacoin, Mincoin, DigitalCoin, Anoncoin, Worldcoin, Freicoin, IxCoin… and more. (That they are clones is obvious from the lack of imagination in naming.) This rush of clones is reminiscent of the South Sea bubble of 1720 and the dot-com mania of 1999, when shares of zero-profit, copycat companies (and even fake ones) sold like hotcakes. Virtually every week now, the Bitcoin code is forked into a new coin that investors bid up. It's as if buyers feel the world will run out of cryptocurrency, which in fact is infinitely and freely duplicable.

3) The Elliott wave pattern from Bitcoin's inception shows five waves up. The December 11 *Short Term Update* noted that a major top was potentially in place:

> The peak [in Bitcoin] came 10 days after U.S. officials, ranging from an assistant attorney general with the Department of Justice to Fed Chairman Ben Bernanke, "spoke approvingly of the potential of virtual currencies." So, here again, the government is getting on board *at the very tail end of a long rise*.

Since we posted that comment, Bitcoin has fallen an additional 40%, bringing it down nearly 60% from its all-time high.

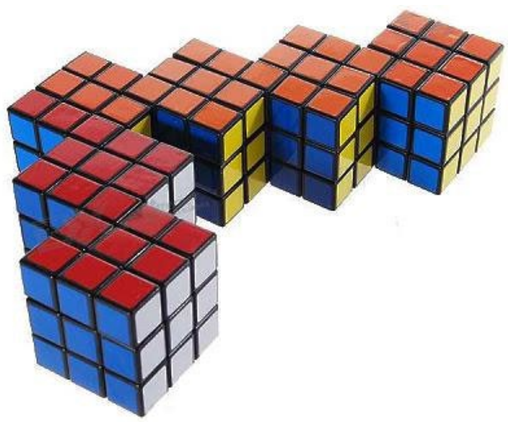
Will this prove to be just another brief, sharp correction or something larger? Take a look at the completed impulse pattern shown below. The structure begins very near the inception of the currency three-plus years ago, when it was selling for a penny. Notice that wave 4 is a triangle (see text, p.49), which typically comes in the fourth-wave position. Wave 5, a thrust, carried to the all-time high of $1242 on November 29. The reversal from that point should mark the start of the largest bear market to date in the currency.



The chart is in log scale to show the returns one would have achieved in each impulse leg of the pattern. Wave ① achieved a stunning 3190x gain. Wave ③ achieved 59.3% (a Fibonacci **3/5**) of the gain of wave ①. Wave ⑤ (measured from the low of wave ④) achieved 39.3% (a Fibonacci **2/5**) of the gain of wave ① and 66.3% (a Fibonacci **2/3**) of the gain of wave ③. Therefore, while each upward move has been large, each successive wave has been *decelerating* in log terms relative to past waves, in each case by a Fibonacci multiple. Also notice that Bitcoin trades more like a commodity than a stock, with its blow-off tops and extended fifth waves. Most of the gain since early 2012 has been within (5) of ③ and the final wave ⑤, all of which is probable retracement territory.

4) Most people involved in this mania seem oblivious to Bitcoin's fundamentals. In my experience, raising these issues publicly earns scorn for spreading "FUD." But there is a good reason—now widely ignored—that Bitcoin is *beta software*. Our August 2010 piece explained how Bitcoin operates, but it's worth revisiting some details to understand just how out-of-touch investor expectations are with the reality of Bitcoin technology. Specifically, let's examine the limitations of Bitcoin's *blockchain*.

The blockchain is the heart of Bitcoin. In its simplest form, the blockchain is a public ledger of all transactions that happen in the Bitcoin network. Each block is composed of individual records that track the ownership of each coin. The transactions "fit" together cryptographically. A block is created about once every 10 minutes by the network. Each block is then cryptographically linked to the previous blocks in the chain, forming a history of all transactions that—to Bitcoin's credit—cannot be forged. To the extent that Bitcoin currency is real, it could be said that the blockchain *is* the Bitcoin currency.



Yet the core problem with the blockchain is that it grows over time and must be shared by every full Bitcoin node. Today it is nearing 13 GB in size. Now, 13 GB doesn't sound too large, but at the current rates of exponential growth the blockchain is projected to become over a terabyte in size in just three years. What's more, the amount of accompanying data required to handle just a fraction of Visa-level traffic would overwhelm even the fastest Internet connections. This technical hurdle makes the "Bitcoin is going to a million" commentary seem premature.

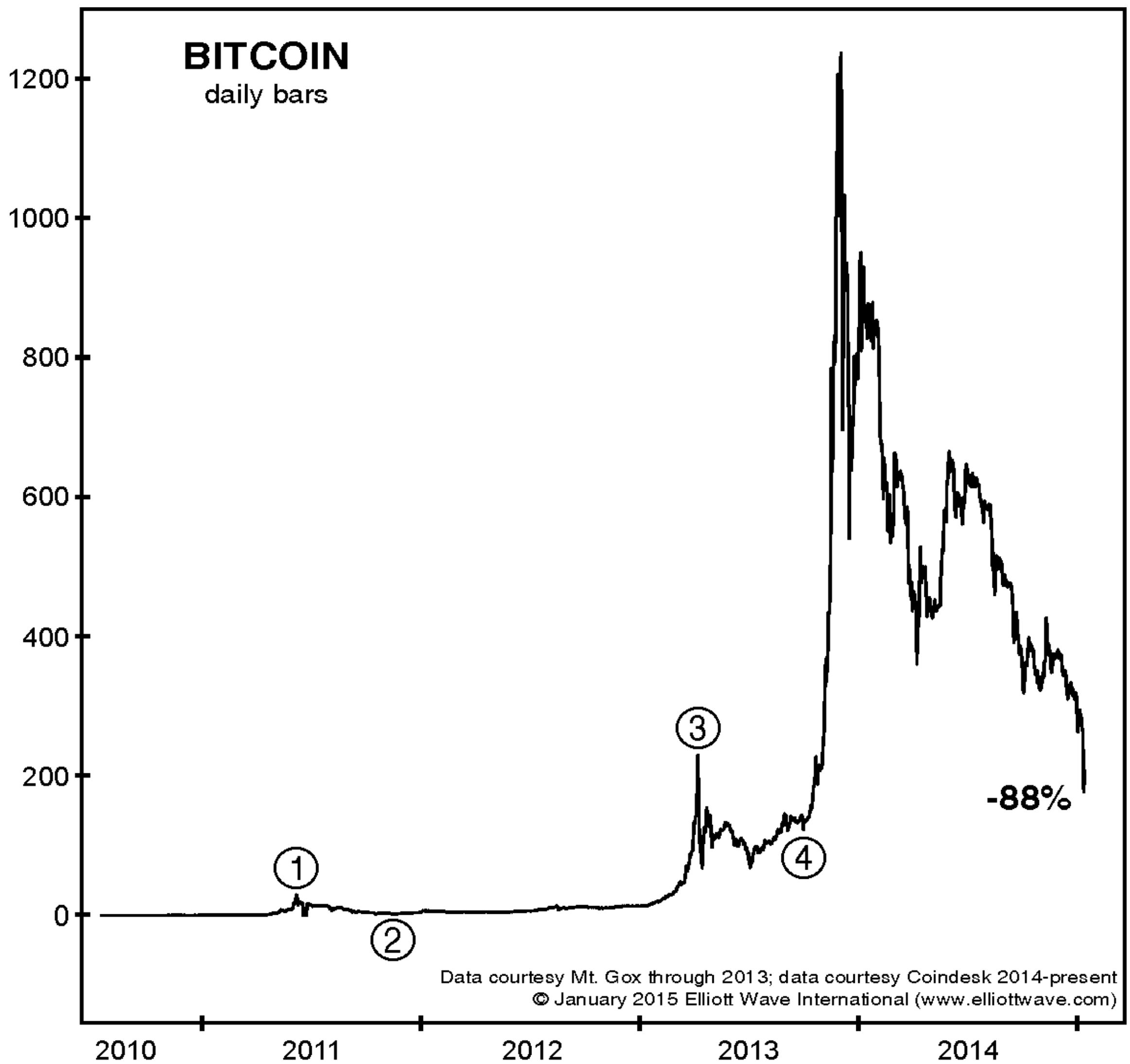
The hope for Bitcoin's future lies in its open-source nature, allowing it to be improved, and Moore's Law. Moore's Law is colloquially used to signify the exponential increases in computer-hardware efficiency over time, including network capacity. But Moore's law—which calls for a doubling of computer speed every two years—has hit a snag in recent years: the rate of improvement in performance has dramatically slowed, causing many experts to call for the end of the operation of Moore's law (http://www.extremetech.com/computing/165331-intels-former-chief-architect-moores-law-will-be-dead-within-a-decade). (For the record, Moore's Law was never intended to refer to computer hardware performance, but the media have confused the term to the point where it is now generally used in this context. Originally, it was intended to refer to the increase in the number of transistors that are packed into microchips.)

The past four years have been an exciting ride for Bitcoin. But the evidence says the Bitcoin bull market is done for now. It would be best to put Bitcoin out of your mind for the duration of the deflationary wave that is curling toward the financial world. Due to the psychology surrounding Bitcoin, it is too risky to buy now. Due to its open-source nature, however, Bitcoin's infrastructure should continue to improve over the years.

For the long run, I agree with Roger Ver, the CEO of memory dealers and one of Bitcoin's earliest adopters, who recently said, "It is just getting started." But one could have said that about the U.S. stock market in 1966. It would have been visionary only if you were patient and willing to hold through a very deep valley. Our position is that Bitcoin will never again sell for 6 cents, as it did when EWT first wrote it up. But there will be another time to buy it for relative peanuts alongside stocks, real-estate, gold and silver. When the time comes, no one will be interested.

Bitcoin (Excerpt from The January 2015 *Elliott Wave Theorist*)

EWT was an early supporter of bitcoin, with contributor Elliott Prechter writing up the crypto-currency in August 2012 when it was off investors' radar and selling for 6 cents. On November 29, 2013, bitcoin reached the incredible height of $1242 and reversed violently, as it had done many times. In the December 2013 issue, we published Elliott's case that a Cycle-degree top had occurred. It was based on wave structure, manic optimism, the sudden proliferation of copycat currencies and even some overlooked fundamentals. Bitcoin is now down to $152.40. It has entered the range of the previous fourth wave, the minimum downside target. Near term, it's due for a rally.



FREE TO SUBSCRIBE, FREE TO SHARE

The *EWAVES Flash* publication is open-access. Feel free to share the link ewaves.com/1501EWF for this issue, or visit ewaves.com and click on the "newsletter" tab to read other issues or to sign up for alerts when new issues are published. To subscribe to Flash Services, use the "services" tab or go directly to ewaves.com/flash-services

CITATIONS

[1]Air Accidents Investigation: 4/1990 G-OBME. (n.d.). Retrieved December 17, 2014, from http://www.aaib.gov.uk/publications/formal_reports/4_1990_g_obme.cfm

[2]Chain of events (aviation). (n.d.). Retrieved December 17, 2014, from https://en.wikipedia.org/wiki/Chain_of_events_(aviation)

[3]Pairwise Testing. (n.d.). Retrieved December 17, 2014, from http://www.pairwise.org/

[4]GIOVE-A - Search EO Satellite Missions. (n.d.). Retrieved December 17, 2014, from https://directory.eoportal.org/web/eoportal/satellite-missions/pag-filter/-/article/giove-a

[5]Robot Check. (n.d.). Retrieved December 17, 2014, from http://www.amazon.com/Code-Complete-Practical-Handbook-Construction/dp/0735619670

[6]Rear Admiral Grace Murray Hopper. (n.d.). Retrieved December 17, 2014, from http://ei.cs.vt.edu/~history/Hopper.Danis.html

[7]Leveson, N. (1995, January 1). Medical Devices: The Therac-25*. Retrieved December 17, 2014, from http://sunnyday.mit.edu/papers/therac.pdf

[8](n.d.). Retrieved December 17, 2014, from http://www.zib.de/zuse/Inhalt/Kommentare/Html/0684 /universal2.html

[9]John Fund - Can This Website Really Be Saved? (n.d.). Retrieved December 17, 2014, from http://www.nationalreview.com/article/362299/can-website-really-be-saved-john-fund

[10]Robot Check. (n.d.). Retrieved December 17, 2014, from http://www.amazon.comThe-Pragmatic-Programmer-Journeyman-Master/dp/020161622X/ref=sr_1_1?ie=UTF8&qid=1396897893&sr=8-1&keywords=the pragmatic programmer

[11](n.d.). Retrieved December 17, 2014, from http://en.wikipedia.org/wiki/Occam's_razor#cite_note-31

[12]Spurious Correlations. (n.d.). Retrieved December 17, 2014, from http://tylervigen.com

EWAVES

EWAVES Flash is published by Qualitative Analytics, Inc. Mailing address: P.O. Box 3102, Gainesville, Georgia, 30503, U.S.A. http://www.ewaves.com.