Dave Lauer
Written Statement
SEC Technology Roundtable, October 2, 2012

Introduction

High-profile technology failures are becoming commonplace on Wall St. The terms "bug" and "glitch" have already entered the popular lexicon, but such terms misleadingly minimize and understate the threat to the public posed by such technology failures. Consider just a few recent examples:

- A failure of the market to absorb a large sell order, resulting in the $1 trillion Flash Crash.
- A failure during the first IPO on an exchange (BATS).
- A failure during the most widely anticipated IPO ever (Facebook).
- A failure during the launch of a brand new NYSE retail market making program (Knight Capital).
- A failure of a UK exchange during the rollout of a brand new trading system.
- A failure to understand, explain or rectify the daily illiquidity contagions in individual securities.

None of these are fairly or accurately referred to as a "glitch" or "bug" and they shouldn't be talked about or addressed as such.

After many technology incidents over the past two years, by no means starting with the Flash Crash, but certainly that being the most extreme, it is critical that the SEC has convened this roundtable to look to the future of technology and US equity markets. The market is no longer run by people gathered together on the floor of a stock exchange. The "market" is more of a concept, a public good to which exchanges, acting as utilities, provide access. Technology has moved so quickly that most market participants have not been able to keep up.

Even the firms at the technological cutting edge, high-frequency trading firms, have difficulty understanding the ramifications of complex algorithms interacting with each other. The SEC must recognize that solving the problems of market structure, fairness and orderliness are now technology issues, and must confront these issues with technology. I would urge the SEC to uphold the tenets of the most disruptive technology force of our time, the Internet, and focus on openness, transparency and a departure from the proprietary ways that have marked the technology revolution on Wall St. up until now. Every firm on the Technology Roundtable and every technology-centric firm in the industry uses open source software and knows the benefits that the open approach offers.

Questions

1. What are current best practices for ensuring adequate testing, robustness, deployment, and use of software systems? Are these practices sufficient to support market continuity and integrity? If not, what else should be done?

It is very difficult to pin down "best practices" in an industry that views almost everything it does as highly confidential and competitively sensitive, even that is not the case. For that and other reasons, the industry simply does not adhere to any particular set of standards. Considering the impact that

electronic trading strategies can have on the market, this should be a concern to everyone. It should also be of concern that the majority of testing occurs when they are initially built – but not as extensively or robustly when those same strategies are modified. Considering the complexity of many of these software codebases, small changes can have dramatic consequences.

There are two main aspects of technology testing that a new, or newly modified, trading strategy should undergo:

- Behavior testing:
    - How does this strategy act under various market conditions:
        - High/medium/low volatility
        - High/medium/low market data volumes
    - Unexpected conditions:
        - Locked / crossed markets
        - Inconsistent market data across market centers
- Load testing: strategies should be tested against many days of recorded market data. Latency-sensitive strategies listening to direct feeds should undergo substantial testing with temporally accurate replay data – this means using hardware devices to simulate actual market data conditions. Replaying market data with software unnaturally smooths out microbursting conditions, resulting in tests that do not reflect the reality of the market data feeds. This can lead to non-deterministic outcomes once the strategy is pushed to production, and is exposed to untested conditions.

I will spend more time discussing testing and what critical changes are required industry-wide in response to question #2 below.

Of course testing is just one aspect of pushing a new, or newly modified, trading strategy out to production. Knight Capital has brought deployment procedures (or the lack thereof) into focus. Deployment procedures across the industry typically consist of:

1. Login to production box.
2. Download new code.
3. Run new strategy.

Now undoubtedly this is simplifying and generalizing across the industry. There are certainly firms with more robust procedures, but that is not the industry norm. In the software industry, it is customary to have build servers in place, with automated Quality Assurance (QA) test suites that run against every build to check for regressions. That is the first step in a multi-step process that new software releases must satisfy before they are ever released into production environments. Once again, the HFT industry should learn from the much more mature software development industry when designing and adopting testing and release procedures.

Along the lines of market continuity and integrity, a topic that has been woefully neglected is that of security. Once again we must realize that individual firms and even single servers can have a

dramatic impact on the integrity of the market. As part of a drive towards a robust, resilient technology-based marketplace that is open and transparent, I would urge the SEC to consider mandating the adoption of the OSSTMM (Open Source Testing Methodology Manual), an open source security specification to which I have contributed in the past. While we have most likely not seen market disturbances that are the result of security failures (although we cannot be sure), we should not fool ourselves into thinking that this industry is not a target.

2. How do market participants balance speed-to-market against the need for extensive testing, or the costs of additional redundancy and safeguards compared with the potential benefits of innovation and rapid development?

> ***"Depending on the trading firm, the life cycle for the development, testing, and deployment of a new trading strategy ranges from minutes to months to one year. At a few firms, new trading strategies are quickly implemented by tweaking code from existing strategies and placing new code into production in a matter of minutes."***[1]

Due to market, profit and competitive pressures (and no countervailing regulatory or legal requirements), HFT firms are much more likely to place new code in production in minutes or days, rather than months depending on the change. Many HFT firms are organized much more like software startups than firms with significant potential to precipitate massive adverse market events. Even proprietary trading desks at larger institutions have a bleeding-edge mentality, meaning that they adopt technology as soon as it comes out – your standard "early adopters" – as most of the edge today is a technological edge. Firms that don't live on this bleeding edge are quickly and easily overtaken by those firms that do.

While the Flash Crash was an eye-opening event, and gave firms a rich new data set to test against, few have mandated such testing, or any other firm-wide systematic test plan. In general, few firms mandate any rigorous testing standards or have a uniform SDLC (Software Development Life Cycle) process. Testing new software or changes to existing software is done on an ad hoc basis with responsibility given to the developer and their manager. QA groups are almost unheard of within the industry.

One of the most important rules in mature software development groups is Quality Assurance (QA). If the SEC were to focus on one aspect of the software development process, it should be on QA. There are several simple rules to proper QA:

1. QA must be performed by an independent group within the software development organization. It should never be performed by the developer, or those who work directly with the developer. These groups must be staffed by highly competent developers and testers, which is one of the more difficult staffing exercises in a software organization.

[1] Clark, Carol and Rajeev Ranjan, "How Do Proprietary Trading Firms Control the Risks of High Speed Trading?", Federal Reserve Bank of Chicago, March 2012.

2. QA must always run a uniform set of tests when any change is made to a software codebase to check for any regressions that may have been introduced by code changes.

3. QA groups must think creatively about conditions that the software will be used under, and robustly test for those conditions as accurately as is technologically possible. As mentioned before, for latency sensitive applications, this should entail hardware-based temporally accurate market data replay.

Quality Assurance is one of the most under-resourced activities in HFT today, and should be a primary focus for building safe software. The SEC should consider requiring QA workers to have securities registrations (to ensure compliance adherence), which would mean increased compensation levels, thus attracting more qualified testers. HFT firms should recognize how critical the QA function is not only to firm-wide health and profitability, but to market quality and confidence. QA should not be a path towards the more profitable life of a trader, but a destination for those wishing to perform a critical function in today's high-speed, electronic market.

In addition, firms must change their mentality when it comes to fixing bugs and making small software changes. The software used by HFT is very complex. At many firms it's been built up over time, often by developers who have since been promoted or left the firm. It is difficult to know what the ramifications are, even with small code changes. In addition, these algorithms are interacting with a complex marketplace. The interactions of these algorithms with each other can often lead to non-deterministic, non-linear behavior as was witnessed during the Flash Crash, and as we see on a nearly daily basis from "mini Flash Crashes" in individual symbols. While developers may feel confident about a change that they have made to their software, it is impossible for them to know how their software will interact with the other algorithms in the market.

> ***"The need for profitable HFT systems cannot take precedence over the quality—stability and reliability—of the global system."***[2]

The environment on a HFT desk is high-pressure. Large amounts of money are being traded and flowing through the HFT software. When problems are discovered, the focus and priority is in fixing them quickly and getting back into the market as fast as possible. This is the absolute wrong mentality. When bugs are discovered, trading should be halted by the desk, the bug fixed, and a fresh, robust QA cycle initiated before any trading can commence.

As the BATS Exchange admits in its comment letter, their IPO was disrupted by a simple bug. "This 'single line of code' programming bug caused the matching engine for ticker symbols in the range A to BFZZZ to enter into an infinite loop, which in turn made those symbols inaccessible on BATS."[3] While their software had undergone extensive testing, this bug was still not caught. As they urge, "The fact that technology errors may be unavoidable, though, does not mean that the industry shouldn't do more to prevent them."[4]

In their comment letter, BATS also urges all firms to do staged rollouts of their trading applications on test symbols in a live environment. I would extend this to include bug fixes as well. The

[2] Van Vliet, Ben, et al, The Rationale for HFT 9000: An ISO 9000-style Quality Management System for High Frequency Trading, August 6, 2012
[3] Swanson, Eric, BATS Comment Letter RE: File No. 4-652, September 27, 2012
[4] Ibid.

Knight Capital incident is just the most visible example of what can happen when established, well-tested software is modified and then rolled out without proper testing or deployment procedures.

3. How do firms test their system for capacity, contingencies, and other unexpected circumstances?

Capacity and load testing can be done in 2 ways:

- Software-based market data replay. This is the predominant method as it is the easiest to setup, run, manage, and is much cheaper. It is also far less accurate for latency-sensitive trading strategies.

- Hardware-based temporally accurate market data replay. The defining characteristic of our current market data system is exceptionally high volume. That volume is not uniformly distributed, and is subject to microbursting – small windows of time, often measured in milliseconds, in which volumes are extremely high. This is an artifact of the extreme level of volume. Because market data feed volumes are so high, it takes several servers from each exchange to produce and send out the data. When those servers happen to put data on the wire at similar times (similar to constructive interference patterns in physics) the bursts can be extreme and produce network saturation conditions for brief periods of time. These microbursts can have non-deterministic effects on extremely latency sensitive, multi-threaded trading strategies, potentially leading to thread contention, positive feedback loops and micro flash crashes. Without this kind of testing, it is difficult to conclusively state that a trading strategy has been fully tested and vetted for load.

Contingency and unexpected circumstances are much more difficult to test for. They are partially accounted for with hardware-based temporally accurate market data replay. That level of load can quickly expose aberrant and dangerous consequences of software that might not be seen with software testing. It can also be used to increase the rate of replay to fully stress test software under conditions that have not yet been seen. What is most important for contingency testing is to test across a very wide range of market conditions.

Aside from using hardware-based replay, firms and regulators can spend time finding these market conditions by identifying days, hours or even minutes of market data that represent a diverse cross section of market conditions, and mandate testing for any new strategy.

4. How is scenario testing performed? Who determines what types of operational risk scenarios a system must be able to withstand?

This is done primarily by the developer of the strategy. Most automated trading strategies do not undergo this level of testing rigor. They are simply backtested against a certain time period of tick data, and as the Chicago Fed report states, "If the results are economically viable"[5] risk limits are set and the strategy is pushed to production.

5. What level of robustness is expected by the market? What is needed? Are there acceptable rates of errors? What levels are practical or achievable?

[5] Clark, Carol and Rajeev Ranjan, "How Do Proprietary Trading Firms Control the Risks of High Speed Trading?", Federal Reserve Bank of Chicago, March 2012.

6. What is the role of independent parties in testing or certifying the many aspects of a robust software development life cycle?

> ***"It is interesting to note that not all trading firms' risk platforms are able to calculate enterprise wide portfolio risk."***[6]

There will never be error-free software. This is a simple fact of software development. Firms must do more to prevent errors, bugs and glitches from spinning out of control and negatively impacting markets, whether through price dislocation or excessive market data volumes.

What is critically important is proper risk controls to ensure that when there are methods to mitigate problems at every step. Firms with automated trading strategies must be able to demonstrate robust risk controls at each level:

- Trading software and individual strategies
- FIX gateway
- Desk-wide and/or firm-wide basis

The Chicago Fed found in their report that "[m]ost firms apply fewer pre trade risk checks to some strategies to reduce latency (delays)."[7] This should be considered unacceptable. The SEC should explicitly mandate minimum acceptable risk controls, and have firms demonstrate their compliance through a third-party audit similar to the ARP program for Exchnages. Firms must maintain an auditable record that demonstrates that every outgoing order passed through these minimum risk checks established by the SEC.

Firms are also inconsistent in their risk control review process. Risk limits when done properly can be a very effective way of limiting strategy exposure and order rate explosions. However, those limits are usually changed over time, without a consistent review process. This is another area the SEC could mandate with acceptable minimums, subject to third-party audits.

The argument against many of these ideas is that firms will act in their own self-interests to ensure that their continuity is not threatened. As we have seen in the market, and in the Chicago Fed report, the decisions made at the individual firm level have an impact on the market-at-large and they are currently being made haphazardly and inconsistently. We must, absent substantial market structure changes, accept that the market is fragile and able to be negatively impacted by an individual server (many of which are capable of sending 100,000 orders per second). We cannot rely on individual firms to take proper precautions; otherwise we will continue to see a plethora of problems similar to the ones that have occurred over the past several years.

There is an argument to be made for adopting a robust technology development standard across the industry for any firm with direct market access, whether that is some adaptation of ISO 9000,

[6] Ibid.
[7] Ibid.

CMM, or some other standard. Three professors from Illinois Institute of Technology, Ben Van Vliet, Andrew Kumeiga and Rick Cooper, along with Jim Northey of FIX Protocol, Ltd have made an excellent case for the creation of a standard within ISO 9000 called HFT 9000. They cite the fact that industries in which societal safety must be ensured have all adopted some form of this quality management standard, including Aerospace, Chemicals, Medical devices, Health care and Food safety. **Surely the equity market as the pillar upon which capitalism rests is equally systemically critical.**

Practically the adoption of such a standard will mean a dramatic increase in documentation (very little is documented currently), strictly regimented development and testing organizations, more robust development methodologies and slower time-to-market for new strategies and fixes to existing strategies. The costs would be high to the industry, although we must ask whether the cost of not taking this action is higher. This is certainly an area ripe for further study and consideration.

7. What additional role, if any, might further or different regulations play in these processes?

The best answer to this question is to quote R.T. Leuchtkafer's public comment that "[w]hat's most revealing is that risk management is left almost entirely up to the trading firms themselves to decide, with highly varying practices and highly varying results." As mentioned in the previous answer, this should be of great concern to everybody involved.

I think regulators can play an instrumental role in encouraging firms to adopt more robust software development practices. I also believe that regulators can play a role by auditing any firm with direct market access or requiring third-party certification and mandating minimum requirements for testing any application that has direct market access, including:

- Independent Quality Assurance teams, responsible for testing all strategies before they are pushed to production.
- Sufficient load and scenario testing, including hardware-based temporally accurate replay for latency sensitive and high-performance trading systems.
- Standards should be examined for any firm with direct market access, either with the SEC's ARP reviews or independent standards such as CMM or ISO 9000.
- As James J. Angel said in his public comment, "The SEC should approach system technology the way the FAA and NTSB approach transportation safety by relying primarily on experienced technical experts." The SEC needs to overhaul its approach to regulatory enforcement – the market is a creature of technology now, and the SEC must adjust to that. The SEC should take great care in ensuring the independence of these experts, and not relying so heavily on industry-aligned individuals.

In addition, I'd like to reiterate two statements I made on September 20 in my testimony before the Senate Banking Committee's subcommittee on Securities:

1. The SEC must implement a robust, market-wide surveillance system and mandate individual ID's on a per-strategy basis.
2. Access to historical market data should be made free and open, and access to live market data made available via API. The SEC should implement a prize-based or percent-of-fine-based

incentive system to encourage independent developers to study the data and develop cutting-edge surveillance and pattern recognition algorithms.

The Market-Wide Surveillance System

Equity market surveillance is currently done by individual exchanges. In the new electronic marketplace, no firm sees an exchange as a center to be traded on in isolation. Such an approach to surveillance is outdated and ineffective.

One of the most important functions the SEC could take on would be to supervise the provision of new trading strategies and the surveillance of automated trading strategies across market centers. The SEC is uniquely positioned to build a market-wide surveillance system in order to revolutionize the policing of markets and enforcement of existing rules and regulations.

The first step would be to build a Strategy Registration System. The Strategy Registration System borrows from a concept at a leading HFT firm. It should be mandatory for firms to have this type of application internally and eventually rolled out by the SEC as part of the Market-Wide Surveillance System. This would be a very simple web application that would allow firms with direct market access to register automated trading strategies with the SEC. This system would request that the firm fill out the following information:

1. Strategy Name
2. Supervisory individual responsible for strategy's actions
3. Contact information for supervisory individual and emergency contacts
4. Strategy Profile
    a. Average/Min/Max cancellation rate
    b. Average/Min/Max orders per second
    c. Exchanges strategy will trade on
    d. Will strategy send ISO orders?
    e. Etc.
5. Group sign-off
    a. Strategy developer
    b. Strategy trader
    c. Trading desk manager
    d. Operations group manager
    e. Head of trading or other executive at the firm

This should be a modern web app, with the ability to save and edit these forms, and use a distributed system for sharing the forms in order to review or sign-off. The form will assign a globally unique ID to the strategy. Exchanges will have to extend their FIX and proprietary electronic order entry protocols to support the receipt of this ID, and participants will have to attach the ID to every quote they submit. Most importantly, the values in the Strategy Profile must be **empirically measured** – not estimated. This will ensure a minimum level of Quality Assurance and backtesting so the firm can be assured that these values are reasonable and realistic.

The SEC should work with existing surveillance operations at market centers to build a distributed market-wide surveillance system that operates both centrally at the SEC and has a software presence at each market center. The SEC can monitor individual strategies across market centers, and build robust software systems that can quickly recognize aberrant behavior and cut off the offenders. These pattern recognition algorithms can start by using the Strategy Profile information from the Strategy Registration System, but over time can build up its own profile of that particular strategy and develop more advanced heuristics based off of this data. **In addition, the SEC should provide an open API-based interface to this system and incentivize independent developers to build novel and advanced pattern recognition algorithms by offering them a percentage of fines collected or using a prize-based mechanism.**

The market-wide surveillance system can be used to drive the kill switches that appear to be the best solution to cutting off disruptive behavior. While the Industry Working Group has proposed kill switches, their idea of using "Peak Net Notional Exposure" is not sufficient to ensure that a disruptive algorithm would be identified. The proposal is based on doing the minimum amount possible and avoiding any centralization of surveillance with the SEC, rather relying on the DTCC to take action on a "near real-time basis." **Anything that starts with "near real-time" is non-viable in this new electronic marketplace in which every minute, second and even millisecond count.**

Response times must be immediate, and low-latency systems must be utilized by regulators in order to regulate and provide surveillance of the low-latency HFT systems. Further, focusing the effort at the DTCC ensures that only filled trades are incorporated into the monitoring system, rather than quote traffic. This is an extreme shortcoming of their proposal, and as such should be considered a non-starter.

In contrast, the market-wide surveillance system proposed here has many advantages because rather than using static kill switches, it enables **dynamic, adaptive kill switches** that are strategy-specific, and can be targeted to only kill the individual ports that a specific strategy is trading over. **It also has the advantage of a market-wide perspective, which is a system that cannot fairly be compared to the simplistic view of exchange-based kill switches and surveillance groups in isolation**.

One concern from the industry will inevitably be the same that they have raised time and again – the fear of information leakage about their strategies and the ability for regulators and exchange surveillance teams to learn what their strategies are doing, and then run to a competitor to ruin the profitability or efficacy of their strategy. Whether advertent or inadvertent, this is a concern that must be addressed through laws, rules, regulations and/or agreements protecting confidential, proprietary and/or competitive information.