Knights Trading Loss

How a company lost $440 million in 45 minutes

By David Hack, William Pearson, Dylan Kumar, Minping Yang and Krishna Kapadia

Introduction:

This report is about the Knights Trading Loss tragedy that occurred on August 1st, 2012, which lead to the trading firm losing $440 million in under an hour. This event caused the Knight Capital Group share price to drop by over 70%, taking the company from being the largest trader in US equities in 2012 to being one of the largest software engineering failures to ever affect the stock market. This report will be focused on what was the failure, why the software failure occurred and how the failure could have been avoided. The topics will include, what was the underlying cause, how did it result in the failure, why was the fault not discovered earlier, which violation of software engineering practices led to the introduction of the fault and also if there would have been a way to catch the problem before the damage was caused. This introduction will give a rough overview of what happened at Knight Group to cause such a monumental failure.

Prior to this event, Knight Capital made changes to its system in order to accommodate a new program that the New York Stock Exchange had put in place. A major change was made that added new code to SMARS, which is a high-speed algorithmic router which would send stock orders to the market. This change was intended to replace code that had previously allowed for a function called Power Peg. Further, this function had been discontinued for 9 years, which in theory meant that no issues would occur while overwriting it as the function was not being used in any of the system's core functions. However, instead of fully erasing Power Peg, a portion of the code was moved to an earlier point in SMARS without any further integration tests. Also, a boolean flag that was once used to activate the Power Peg function had been repurposed for use with the new code.

Once these changes had been completed the updated software was deployed to SMARS which consisted of 8 servers. However, the update was only deployed on 7, which meant that 1 server still contained the outdated code while the other 7 were using the new code which also utilized the Power Peg activation flag. What could go wrong? Once the market opened, the system began to buy and sell at a rapid rate causing Knight Capital to lose $440 million in under an hour.

What was the underlying cause

On

Upon deployment, the new RLP code in SMARS was **intended to replace unused code in the relevant portion of the order router.** This unused code previously had been used for functionality called “Power Peg,” which Knight had discontinued using many years earlier. Despite the lack of use, the Power Peg functionality remained present and callable at the time of the RLP deployment. The new RLP code also **repurposed a flag** that was formerly used to activate the Power Peg code. Knight intended to delete the Power Peg code so that when this flag was set to “yes,” the new RLP functionality—rather than Power Peg—would be engaged.

Underlying Causes

A cumulative quantity function counted the number of shares of the parent order that had been executed, when Knight used the Power Peg code. This

When Knight used the Power Peg code previously, as child orders were executed, a cumulative quantity function counted the number of shares of the parent order that had been executed.

One of the underlying causes this failure occurred is that Knight Knight did not retest the Power Peg code after moving the cumulative quantity function to determine whether Power Peg would still function correctly if called. Because, in 2005, Knight moved the tracking of cumulative shares function in the Power Peg code to an earlier point in the SMARS code sequence.

Other underlying causes is that Knight did not have a second technician review this deployment, and they also did not have written procedures that required such a review. As this report mentioned before, there is one of Knight’s technicians did not copy the new code to one of the eight SMARS computer servers. Furthermore, no one technician realized that the Power Peg code had not been removed from the eighth server, nor the new RLP code added. When the Power Peg flag on the eighth server was activated the Power Peg functionality began routing child orders for execution, but wasn’t tracking the amount of shares against the parent order – somewhat like an endless loop.

* How was the fault introduced?

Upon deployment, the new RLP code in SMARS was **intended to replace unused code in the relevant portion of the order router.** This unused code previously had been used for functionality called “Power Peg,” which Knight had discontinued using many years earlier. Despite the lack of use, the Power Peg functionality remained present and callable at the time of the RLP deployment. The new RLP code also **repurposed a flag** that was formerly used to activate the Power Peg code. Knight intended to delete the Power Peg code so that when this flag was set to “yes,” the new RLP functionality—rather than Power Peg—would be engaged.

* Why was the fault not discovered earlier?

The underlying cause of the failure occur is that, Knight does not have good software engineering practices regarding their documentation. The fault was introduced by that one of Knight’s technicians did not copy the new code to one of the eight SMARS computer servers

The fault was not discovered earlier because Knight did not have a second technician review this deployment, and they also did not have written procedures that required such a review.