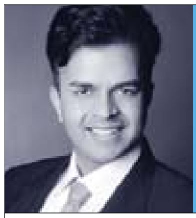
1-2010

Operational risk in trading platforms

M. THULASIDAS
Singapore Management University, manojt@smu.edu.sg

Citation

THULASIDAS, M.. Operational risk in trading platforms. (2010). *Wilmott*. 64-65.
Available at: https://ink.library.smu.edu.sg/sis_research/6145



Manoj Thulasidas

Operational Risk in Trading Platforms

Uses for a "useless" risk

Operational risk is the risk of loss resulting from inadequate or failed internal processes, people, and systems, or from external events. So says Basel II. A trading platform is a system, and therefore comes under the umbrella definition of operational risk. In fact, only one word in that Basel II definition of operational risk (namely "systems") holds the risk arising from using trading platforms.

Banks, focusing on the potential failures of systems, mitigate this risk by putting in place contingency plans. For instance, if the building that houses the trade database goes up in flames, there is a backup location and a preapproved switchover plan. Such business continuity plans are essential, but in the context of trading systems, we have more to do.

In order to appreciate the less severe, but more frequent, risk events associated with trading systems, we need to expand the definition of operational risk to include any potential loss that is not market or credit related. From this all-inclusive view of operational risk, we will examine the possible weaknesses in trading platforms, particularly the ones developed in-house.

Fear not good people, for with this BLACK-SCHOLES MODEL, BIOGRAPHY OF BILL GATES AND BOTTLE OF PETROS '46 I SHALL PENETRATE THE SILOS OF KNOWLEDGE!
BSM
GATES
PETR
WALL ST

A special kind of risk

Operational risk is a "useless" risk from a quant's perspective. Willingness to assume higher risk is supposed to bring in more potential rewards. This axiom works well with market risk and credit risk. But when it comes to operational risk, the riskier position only brings in more pain and suffering – no rewards.

Associated with this lack of rewards is our inability to quantify or model operational risk. Although we can apply some statistical modeling to high-frequency, low-impact events (such as data entry errors, or petty fraud), it is next to impossible to model the rare, but catastrophic events in any meaningful fashion. The success rate in anticipating such events is similar to that in earthquake prediction, which indeed is an operational risk event.

In the absence of pure quantitative methods and hedging possibilities, a large part of operational risk management tends to be process based. To what extent these risk-mitigating processes are facilitated and enforced in a trading system depends on its design.

Trading platforms

A trading platform is essentially a database application that can perform trade pricing, booking, lifecycle and risk management, trade settlement, and accounting. Each one of these tasks is vast in its scope. For instance, trade pricing is what mathematical finance is all about. Booking calls for a robust database layer and specialized programmers. Lifecycle management is what the middle office team spends its time doing. Market and credit risk management keeps huge middle office teams busy, while trade settlement and accounting are the bread and butter for the back office and finance department. As a software solution taking care of these complex processes, the trading platform tends to be vast, complex, and expensive programs. And in vast and complex processes and systems reside tricky operational risks.

Vended systems tend to do well in providing process hooks to manage operational risks. They typically have strong access control mechanisms, robust audit trail implementations, and historization. They also come with a certain level of guaranteed support and continuity, at least to the extent that the vendor company is solvent.

An in-house trading platform, on the other hand, performs poorly from an operational risk management perspective. An extreme example of such a system is a spreadsheet-based pricing and trade-booking solution. To an operational risk manager, spreadsheet is the worst kind of in-house trading platform. It is open to uncontrolled end-user modification, and even fraud. As a trading platform, it may contain live data feeds and external links that may become stale without the users being aware. And spreadsheets have notoriously poor security, and are exceedingly difficult to version-control and centrally manage.

A properly designed and developed in-house trading platform will address most of these drawbacks of a spreadsheet solution. However, given that vended platforms perform better, why develop in-house systems at all? The primary objective of an in-house trading system is to quickly deploy the brilliant pricing models emanating from local quants. And the operative word there is "quickly." Why this need for speed? Because market opportunities are transient. Remember the crude oil price swings of 2008. Such wild swings induce the need for rolling out customized hedging solutions, which can be rolled out only with the celerity afforded by in-house systems.

Implementation options

Once we convince ourselves that we need a tailor-made trading platform, we have a few options to put it in operation. The safest option, if we want to minimize the associated operational risks, would be to ask our favorite vendor to do it for us. However, true to the proverbial absence of free lunches and such, this option does come with its downsides. One downside is that the response time for deploying a new quant model in this framework would be unacceptably long. Secondly, vendor development tends to be expensive, especially if we demand our intellectual property, thereby nullifying any profit potential that we can glean from the transient market conditions we are after.

A second viable option of deploying a customized trading platform is to use a vendor-provided application programming interface (API). In principle, this option retains all the good things about outsourcing our development aspiration to the vendor, while minimizing the associated perils. But the free-lunch principle should cause us to be vigilant, and look at the situation from a practical point of view. The vendor-provided APIs tend to be incomprehensible and inflexible, which has to be expected because vendors of trading systems have no incentive in encouraging in-house development. Besides, vendor APIs are not cheap – after all, it is not in the vendors' interest to help us be totally self-reliant. (But they do tout the existence of the API as a key selling point.)

In addition to the shortcomings of the API, we end up battling the process-issues related to release cycles as well. The vended systems are deployed by the IT team, not by quantitative developers. And the deployment involves the vendors heavily. Thus, deploying new products through the API may still be delayed by the scheduling priorities of other teams over which the product innovators of the front office have no control.

Since the vendor API is usually complicated, it is only one or two key developers in the quantitative development team who turn out to be familiar with it. This concentration of a crucial skill results in significant key-person risk to the financial institution.

Despite these shortcomings, in-house development using vendor APIs is the chosen route for a large number of midtier players in the financial industry. But if we do not want to compromise on speed and flexibility, we have to do it ourselves, which is our third option. If we choose to go with the in-house approach, we can control the release schedule, resulting in a near ideal response to the front office demands. A well-designed in-house system can be flexible, responsive, rapid, and extensible, although it might prove to be more error prone than using inflexible vendor APIs. In addition, supporting such a trading platform may turn out to be costly because of the nature of in-house development.

Another potential issue with an in-house trading system is a less than ideal integration with the existing settlement and risk management systems. Again, a sound understanding of the downstream systems and processes and a good design and implementation plan can help to avoid nasty surprises during the integration phase. Almost all the investment banks (of the pre-2008 financial meltdown era) had well-developed in-house trading platforms.

Risks and mitigation

Most of the difficulties related to in-house trading systems stem from the lack of communication between various business units, and the consequent emergence of what we can call the silos of knowledge. For instance, quants, whose work drives the need for the in-house system to begin with, are way too mathematical for anyone else in the bank. The developers, who bring the pricing models to the trading systems, are computer science professionals – again, far removed from the rest of the banking world. Traders, who end up using the models on the in-house systems, tend to be more nimble, market-oriented people, unlike the quants and the developers.

The knowledge gap is even bigger when we move to other aspects of trade lifecycle management. Although risk management professionals use methodologies similar to the front office staff, their philosophical focus (on risk reduction rather than risk taking) puts them at variance, and often in conflict, with them.

One efficient way to fight operational risk issues arising from the development of an in-house system is to spread the specialized knowledge residing in the various business units – break the silos, as it were. This need is what prompted me to embark on writing my book *Principles of Quantitative Development* (to be published by John Wiley & Sons Ltd. early in 2010). In it, I endeavor to develop in each professional team a healthy respect for all other business units, by pointing out the various functions, their needs, and the associated trade perspectives.

While designing an in-house trading platform, the architects have to spend a long time understanding the trade perspectives and the work paradigms in various business units, and fully appreciate the requirements arising from them. It may be wise to reuse as much of the existing infrastructure (for trade settlement, accounting rules, etc.) without trying to reinvent the wheel.

Because of the compelling reasons for their existence, in-house trading platforms are here to stay, and we need to bring them under Operational Risk Management.

About the Author

The author is a scientist from the European Organization for Nuclear Research (CERN), who currently works as a senior quantitative professional at Standard Chartered in Singapore. The views expressed in this column are his own, which have not been influenced by considerations of his employer's business or client relationships. More information about the author and his forthcoming book (*Principles of Quantitative Development*, to be published by John Wiley & Sons Ltd.) can be found at his blog: www.Thulasidas.com.