



## The flaws in Operational Risk software

### Summary

Although the bulk of the current software that purports to support Operational Risk fails miserably in that task, this is not an IT issue. It is a consequence of the lack of maturity in Operational Risk thinking. It can be addressed by recognising four areas where ORM has led the software vendors up the garden path: the base data, the analysis, the reporting and the comprehensive fabric or ORM. These need to be addressed by the ORM community before sensible software can be expected.

### Dear reader,

Much has been expected from OpRisk software and little has been delivered. This is a failure of the OpRisk discipline rather than a failure of IT. Even OpRisk professionals are often confused about crucial issues such as the correct scoping of the domain, the use of unambiguous terminology, setting out a consistent workflow and adopting meaningful measurements. In such a landscape, it should not be surprising that supporting software is equally confusing and unhelpful. Four areas of misunderstandings need to be addressed by the ORM community before IT solutions can be developed.

#### 1. Base data: The need for stable framework

Anybody working in ORM is familiar with the triumphant remark “operational risk!” when someone bumps into a door, the copier runs out of paper, or the biscuits ran out in a meeting. It is, however, seldom heard when collateral turns out not to exist, the wrong pricing was used for a trade, projects fail to deliver the promised value or when a payments and settlements system is out of order for a day. And yet these are the real examples of OpRisk events.

This is partly a result of the fact that ORM has not managed to create a consistent methodology, a consistent classification or even a proper object of study. The rush to implement IT solutions for this ill-defined field has led to predictable results: endless configuration issues since the vendors need to cater for ever framework, but usually no thorough support for any specific framework. What is needed is a rigorous framework of base data anyone calls in the software vendors.

#### 2. Analysis: The correct place of quantification

The fact that the structure is often muddled poses no serious problem for the modern software vendors as such. Since everything is modular, they will merrily configure as per requirements. The next step is often to start using some of the acquired data to start calculating something. Anything will do, since the desire for number from senior management is often insatiable. Not to be outdone, the vendors add statistical analysis to the package. This is their field, so they run rampant if they see half the chance. Large vendors (who shall remain unnamed) have thus developed AMA modules, splicing options, weird and wonderful curve fitting techniques and breathtaking extrapolation software. Sadly, the base data does not allow the application of 99.9% of these techniques. We should demand better insight in processes, (changes to) risks within processes and information about (changes to) controls and such like. We should therefore ask for descriptive statistics before moving on to modelling.



## The flaws in Operational Risk software

### 3. Reporting: The correct way to synthesise data

An area that vendors and senior management are equally in love with that of dashboards. The vendors since it offers them a possibility to show what fancy displays they can create and senior management since it shows them a colourful and recognisable summary of the information. This is what they were asking all along: pointers to the identified weak spots in processes, progress charts of roll out, and of course an overview of risks to the bank in a tri-coloured grid.

As long as these dashboards stick to the raw data, they are merely another way of displaying data and there is nothing wrong with it. But the nature of software packages is that they cannot resist 'adding value' by also displaying summaries, roll-up data, implications for the next level or risk maps with impressive arrows from 'inherent risk' to 'residual risk', as well as links across businesses and geographies. The trouble is that these pie charts and diagrams appear as sensible as the display of the raw data (a pie chart is a pie chart after all), but they are anything but. These aggregations etc usually do not stand up to scrutiny, which is something ORM must solve. We must find a way to properly synthesise data to higher levels. Until that time, reporting should be on an atomic level and we should instruct our vendors accordingly.

### 4. The ORM world: How do the areas of attention fit together?

Time was when OpRisk was called 'Other Risk'. That phase has thankfully passed, but the various areas of attention within ORM are still unconnected. Within ORM, it is common to see a split between core ORM (Loss data, RSAs, KRAs etc) and IT security, business continuity management, handling of legal risks etc, to the extent that the governance, the base data, the analysis and the reporting in these areas follow their own paths and are not integrated.

Gathering and presenting ORM data will benefit tremendously if these areas can all be brought under one umbrella. Since banks have an increased need for Integrated or Enterprise wide risk management, the least ORM should do is to unify its own data. That can be done by focusing on the common factor that underlies all ORM activities, namely though a combination of processes, products and organisational structure.

These three dimensions (or four if we include geography) must be applied uniformly across all ORM tools and programmes, using a single reference list for each if we are ever going to see integrated ORM data collection, analysis and reporting.

### Concluding remarks

So is the poor systems support for operational risk all ORM's fault? Not quite. Many software vendors have also taken a shortcut by building on their existing software platforms for, say, the audit world or some statistical notion of risk management. They cannot really be blamed for that, but they can be blamed for the lack of imagination shown since. The ORM software for 2010 and beyond needs to do a lot more than that of 2009 and before.

Before chastising the vendors for not building what ORM needs, the call to action for ORM is to have something concrete, consistent and coherent to put in front of the vendors. Unless and until that happens, ORM software will remain the poor cousin of the risk family.