

A record number of firms are expected to upgrade their energy trading and risk management systems this year. *Brian Shydlo* of Sirius Solutions addresses the most common questions asked about ETRM upgrade projects and introduces the ETRM upgrade project checklist

Grading your upgrade



Brian Shydlo, Sirius Solutions

★ Upgrading a vendor-supplied energy trading and risk management (ETRM) system is a significant project with various project risks and uncertainties. Because it happens only every few years, the typical programme manager tasked with leading the upgrade is often new to this type of project and can therefore be apprehensive about what to expect and how to plan for success. This article sets out to allay fears and promote clarity by answering the most common questions about an ETRM upgrade project.

To upgrade or not to upgrade?

Firms in the business of energy trading will eventually be faced with the decision to upgrade their ETRM software. Some may decide to upgrade in response to a need for new functionality, improved performance, or updated support for third party software. Most firms sensibly defer upgrading for as long as they can.

While most organisations will eventually choose to upgrade,

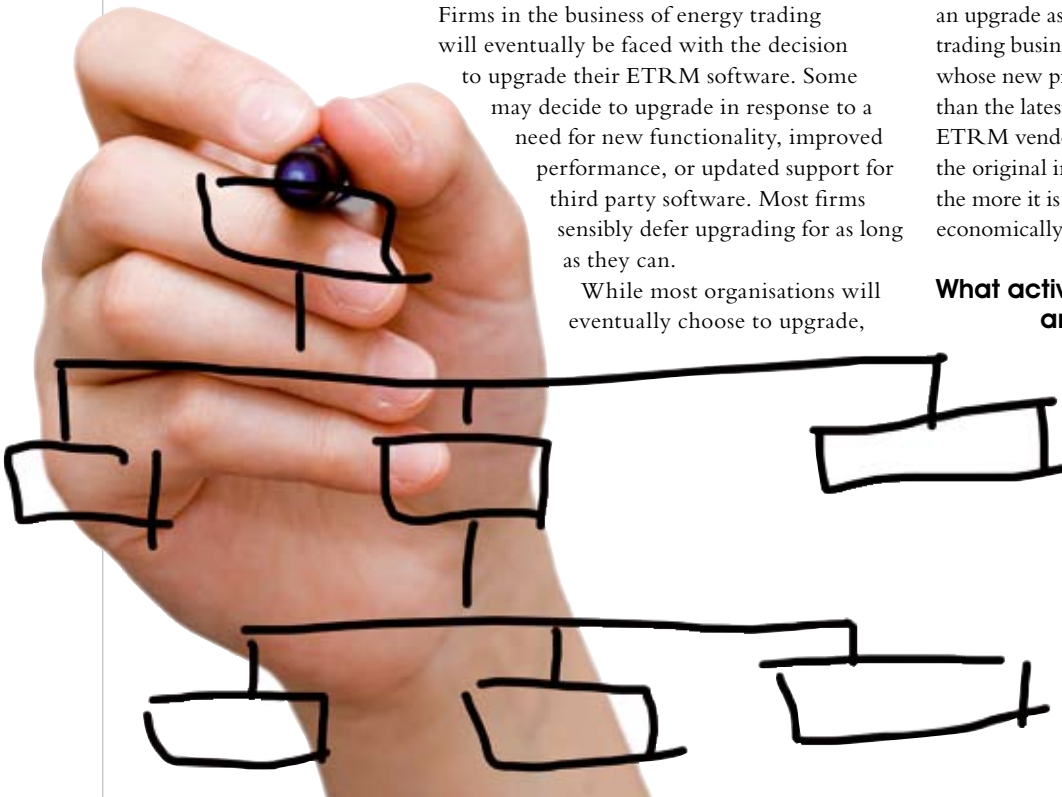
some may consider switching vendors as an alternative, and a growing number have eliminated upgrades altogether by using web-hosted applications.

The decision-making process to upgrade or to switch vendors is similar to the vendor selection process that firms would have originally used to choose their current ETRM vendor. Firms should therefore think of upgrading their software as just another choice in a new ETRM selection process.

The two main decision criteria would still be economics and functionality. Firms considering an upgrade as part of a reaffirmation of their trading business may consider another vendor whose new product offers more functionality than the latest version offered by their current ETRM vendor. The longer it has been since the original implementation or prior upgrade, the more it is likely that switching vendors is economically comparable to an upgrade.

What activities are involved in an upgrade?

The ETRM upgrade project checklist introduces the steps to complete an upgrade (see table 1). This tool was designed by Sirius Solutions to be a starting point for building a project plan. The top level contains the primary categories of activities and subsequent levels provide details. For example, the 'Quality assurance testing' item from the top level is subdivided in the second level into types of QA testing such as



functional, regression, capacity, and performance testing (see table 2). Level three offers additional details on the second level tasks (see table 3).

Using a standardised checklist is helpful as it increases task visibility and ensures important items are included in the project plan. Without a checklist, required activities may initially be missed and lead to unanticipated project delays, particularly for tasks requiring a long lead time.

The second benefit that the checklist provides is that it can be converted into a project scorecard by replacing checks with letter grades and it can be suitably formatted for use as a management presentation.

What is the best way to structure the project team?

The best project structure, based on experience, is one where responsibility is divided among four workstreams and additionally, a separate project steering committee comprised of representatives from each stakeholder group performing an oversight role.

1. The project management office (PMO):

- creates the project charter containing project scope and methodology;
- distributes meeting minutes and status reports tailored to the needs of the recipients based upon an agreed communications plan, specifying the level of detail and distribution frequency for each phase of the project. For example, the PMO might send the immediate project team members a detailed status report at the end of each week, while the project's steering committee might receive a high level status report once a month at the beginning of the project, moving to a weekly frequency by the end of the project;
- organises training on the new version of the software;
- supports other workstreams.

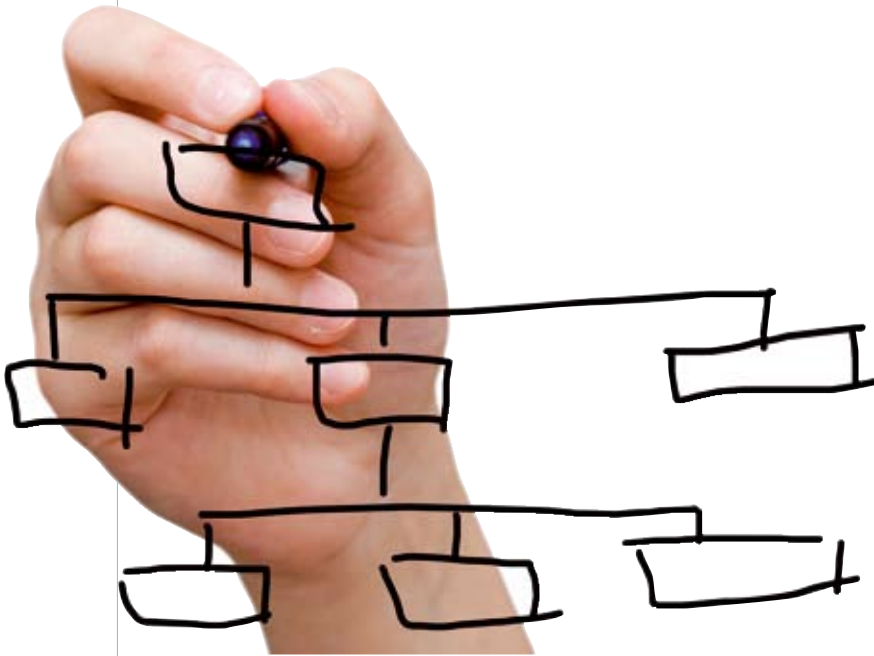
2. The testing stream:

- documents a testing plan specifying the method and scope of testing
- creates and executes test cases
- obtains user approval that both the testing results are acceptable and that the testing scope is comprehensive. User approval is often referred to as user sign-off from the formal procedure of obtaining a signature,

T1. ETRM Upgrade project checklist			
1	Business case written & budget approved		✓
2	Creation of project management office		✓
3	Project charter		✓
4	Resources mobilised		✓
5	Production roll-out plan		✓
6	Release notes review		✓
7	Remediation of custom code / interfaces		✓
8	Quality assurance testing		✓
9	User acceptance testing		✓
10	Security permissions sign-off		✓
11	Training		✓
12	Dress rehearsal		✓
13	Go Live		✓
14	Project closure		✓
Source: Sirius Solutions			

T2. Upgrade checklist			
8	Quality assurance testing		✓
	8.1	Functional testing	✓
	8.2	Regression testing	✓
	8.3	Capacity testing	✓
	8.4	Performance testing	✓
	8.5	Integration testing	✓
Source: Sirius Solutions			

T3. Upgrade checklist				
8	Quality assurance testing			
	8.1	Functional testing		
		8.1.1	Functional requirements	✓
		8.1.2	Test-case creation	✓
		8.1.3	Test-case sign-off	✓
		8.1.4	Environment ready	✓
		8.1.5	Test cycle 1	✓
		8.1.6	Test cycle 2	✓
		8.1.7	Test cycle 3	✓
		8.1.8	Testing passed	✓
Source: Sirius Solutions				



Firms that disapprove of a design change may try to persuade a vendor to reintroduce the old behaviour

though in practice email confirmations have replaced signatures. This process is valuable as it makes explicit that both users and testers are jointly responsible for the success of the testing efforts and thus prevents users from placing all of the blame on the testing team if something goes wrong.

- documents steps to recreate perceived defects (bugs) in the software.
- submits bug reports to the mitigation stream to investigate and either confirm and fix the bug or respond that the item submitted as a bug is in fact caused by an intentional design change. Firms may then decide to accept willingly the changed behaviour. Firms that disapprove of a design change may try to persuade a vendor to reintroduce the old behaviour and may sometimes succeed, or if they fail, be forced to live with the design change without recourse as vendors are technically responsible only to fix bugs.
- confirms bug fixes by rerunning the test that initially identified the bug.

3. The mitigation stream:

- resolves issues found during testing;
- tracks the status of issues from inception to closure;
- co-ordinates and deploys new releases with bug fixes.

4. The production rollout stream:

- creates the production deployment plan, detailing the upgrade weekend event;
- plans and performs dress rehearsals;
- deploys the updated software and configurations into the production environment.

The average upgrade project is staffed with internal resources, external consultants, software vendor personnel, and users of the software. Software vendor resources primarily have a role limited to the mitigation stream and the production rollout stream is typically staffed with internal resources.

The strategy of first brainstorming the specific activities within each stream and then assigning people to those tasks can offer benefits relative to the strategy of selecting the people first.

How long will it take?

The typical upgrade takes six to ten months, which can seem like a long time until people understand the rationale. The chief project bottleneck is the number of testing cycles and frequency of the corresponding bug fix releases.

Practically speaking, the minimum number of testing cycles needed for an upgrade plan is three. The first cycle will find most of the bugs, the second cycle retests the bug fixes and almost always finds a small number of bugs that were not fixed completely and new issues.

In the best case, a third testing cycle is needed to retest and confirm there are no remaining open issues. Plans often include one or two extra testing cycles as a contingency in case issues remain after the third testing cycle.

A new test cycle can begin only after receiving and installing a bug fix release from the vendor. The average time between releases is four weeks, and two weeks is considered a speedy turnaround time. When creating your plan, take care to consider that it can take a month or more until the upgrade team is able to test a fix for an issue.

How much will it cost?

Firms pay ETRM software vendors annual maintenance fees that entitle them to new versions of software with new features and bug fixes, so the primary contributor to cost is the people working on the project. Many organisations charge projects for the use of internal resources to better capture true project costs. This practice serves to show correctly that external consultants can be close to internal resources in terms of hourly rate equivalents.

Total costs vary and are usually in the range of \$1m to \$5m. One key contributor is the number of design changes and bug fixes in the new version relative to existing version; upgrading multiple versions simultaneously will cost more than upgrading a single version. Because of the wide range of potential project costs, the standard for consulting firms is to charge based on time and materials.

While cost is always paramount, for most firms, the goal is not solely to minimise costs. Firms may instead want to optimise cost, project length, and project risk. For example, firms may want to engage external consultants to reduce the risk of limited user availability by having consultants assist with user acceptance testing. The most effective way to reduce cost, time and risk is to engage consultants who have experience upgrading your specific ETRM software.

What are the biggest risks and how to mitigate them?

The biggest risk to your upgrade is the unknown. Besides following the steps suggested here, the best way to mitigate this risk is to reduce the number of unknowns by limiting the upgrade to existing functionality and not changing how the system is used. A separate project to implement new features could run in parallel and be deployed post-upgrade.

Another risk is insufficient vendor support, as upgrades rely on vendors to fix bugs and provide information. Original implementations have milestone payments tied to objectives, which inspire vendors to provide speedy issue resolution. That is not the case with upgrades.

However, if you obtain even a small number of dedicated resources from the vendor to help manage project issues, it can make a big difference in terms of improved turnaround times. Providing the vendor with early notice of your plans and frequent updates can also help.

The risk that QA testing may not find a bug can be mitigated by the surprisingly simple though effective technique of setting a well thought-out testing objective. A stated goal of “ensuring that the firm can run its trading business if all test cases pass” is more practical and easier to achieve than an alternate goal

The longer it has been since the original implementation, the more it is likely that switching vendors is economically comparable to an upgrade

of ridding the system of defects, in particular because firms use only a subset of features in vendor-supplied software. Requiring users to sign off test cases is another way to promote appropriate testing scope.

Conclusion

It is true that an ETRM upgrade is a significant project for most organisations. A typical upgrade will keep an issues log to track both bugs discovered by the testing team and any questions posed to the vendor regarding new or changed functionality. By the end of the project the log could contain hundreds of items. The upgrade will also have to contend with the dozens of bug fixes that the vendor is likely to make to the new version prior to the start of your upgrade project, e.g., based on the findings of other firms’ upgrades.

For example, if there was a bug in the way that the older version valued deals, you may find that upgrading your software changes the valuation of deals, because of the bug fix which has the effect of making reconciling data more difficult than you anticipated. And there can be numerous other challenges, particularly those brought about by insufficient planning. By raising awareness of the activities involved in an upgrade, programme managers can create better project structures and plans and have more confidence in their organisation’s ability to successfully execute those plans. [ER](#)

Brian Shydlo is a director of Sirius Solutions, a software consultancy and implementation specialist.
Email: bshydlo@sirsol.com