

## Playing the Numbers: Assigning EFT Testing Priorities

---

*No organization can test everything. So how do you make intelligent decisions about which tests to include in your EFT regression testing? How do you create a test plan that makes the most of your test resources? This article offers one approach to categorizing and prioritizing your regression testing of electronic payments.*

### What Every EFT Tester Knows

While managers at financial institutions agree in principle that EFT software testing is necessary and important, these principles are not always reflected in the time and resources allotted to testing. More often than not, testing is constrained by limited budgets, or squeezed by delivery promises and aggressive deadlines. Unlimited testing is simply not an option. The only course of action, then, is to determine the best use of the limited resources of Quality Assurance/Quality Control (QA/QC) departments. The question is: how do you determine that?

### Identifying Electronic Payment Events and Evaluating Probability

You likely already have a list of events that you want to test before moving a new feature or release into production. How do you begin to prioritize those EFT tests?

- **First, you evaluate your list in terms of which events potentially have the most severe financial impact** on your financial institution. Put simply, which events will “take money out of your pocket,” causing losses that directly affect the bottom line? Certainly, cases of ATMs dispensing cash in excess of account balances, or of incorrectly posting cash-back amounts associated with POS debits, are extreme examples of events that can have a severe impact on your organization.

Some events do not result in direct losses, but will potentially decrease or limit your profitability. Events that limit ATM or POS usage, or affect switch availability, may fit into this category.

- **Secondly, you determine probability, or the likelihood of these events occurring.** By examining why and how such events occur, and incorporating this knowledge into development of your QA scripts and procedures, you can reduce the probability of these events.

## A Simple Example of Calculating Risk Exposure

After determining the potential for financial loss and the probability of each event, you can begin a simple risk exposure calculation. In our example, the scale is 1 to 4, where 1 is highly unlikely or very low impact, and 4 is highly likely or very high impact. (Your risk assessment scale will likely be more complex and can be devised by QA personnel or other stakeholders in your organization.)

Probability/Likelihood (x value)	
Highly likely =	4
Likely =	3
Unlikely =	2
Highly unlikely =	1

Financial Impact (y value)	
Very high =	4
High =	3
Low =	2
Very low =	1

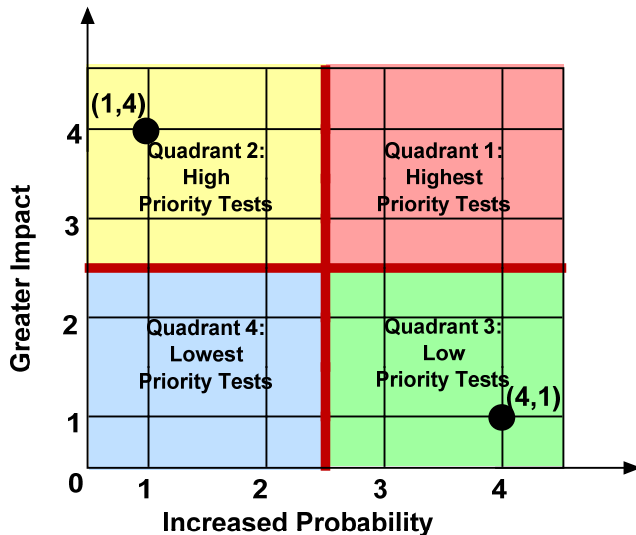
Use these scales to assign two numbers to each event: one representing the likelihood that the event may occur, and one that indicates the severity of the financial impact to your organization. After you have assigned the number pair, you can plot each event on a graph to begin assigning event priority.

## Electronic Payment Event Examples and Their Resulting Number Pairs:

Event Examples	Probability/Likelihood Score	Financial Impact Score	Resulting Number Pair
Approval of a transaction for an account without sufficient funds	Unlikely (x=2)	Very high (y=4)	(2,4)
Approved transaction because of improper PIN checking	Highly unlikely (x=1)	Very high (y=4)	(1,4)
Approval of multiple (fraudulent) POS purchases from a single cardholder due to a failure to enforce the velocity limit on the cardholder account	Likely (x=3)	High (y=3)	(3,3)
POS transactions are approved for a cardholder in error because pre-authorizations are not processed correctly	Highly likely (x=4)	Very high (y=4)	(4,4)
An ATM has a reduced service level because its supply of receipt paper is low	Highly likely (x=4)	Very low (y=1)	(4, 1)

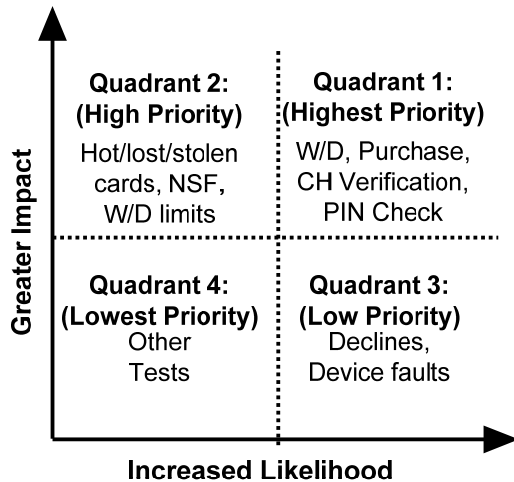
## Using Risk Exposure to Assign Event Priority

As you plot various events, imagine that you are dividing your graph into quadrants that contain four categories of events: priority 1 (highest priority) events through priority 4 (lowest priority) events. Logically, you should first focus EFT testing on the highest priority events, and then test the next priority events, and so on.



When assigning event priority, the overriding impetus is **minimizing the financial impact on the institution**. In every case, an event that has a high financial impact but is unlikely to occur will be rated as a higher priority event than one that is highly likely to occur but has a small financial impact. So an event with a low probability of occurrence ( $x=1$ ) and a high financial impact ( $y=4$ ) falls into priority 2 (1, 4), while an event with a high probability of occurrence ( $x=4$ ) but a low financial impact ( $y=1$ ) falls into priority 3 (4, 1).

## Examples of Electronic Payment Events and Their Priorities



Let's look at some examples of events in the quadrants.

- **Priority 1 events represent testing standard functionality** of the system, including withdrawals, transfers, authorizations, purchases, returns, PIN checking, cardholder verification, and reversals.

A financial institution stands to suffer the greatest losses should something go awry here, such as when the system functions, but in a compromised condition. The financial institution can be assessed switch penalty fees for non-availability, or a compromised system can dispense money in violation of normal safeguards, resulting in substantial losses. When ATMs are dispensing "free money," or when unlimited POS purchases are approved, there will always be dishonest consumers ready to quickly take advantage of the situation.

Tests for these events guard against failures that can result in loss or corruption of data, or partial or complete loss of application functionality. These tests should be run during the development process and on the final build, and should always be executed prior to deploying an application in production. Regression scripts should also include these tests.

Problems during everyday standard operations will have the greatest financial impact on the institution, and so must be assigned the greatest focus or largest share of test resources.

- **Priority 2 events test system functionality involving velocity limits, hot cards, or address verification.** As these are some of the major avenues of fraud, these events can also have a significant financial impact on an institution. While incorrect functioning of an application during these events will not normally stop system operation, problems here can cause substantial financial losses.

Priority 2 tests guard against failures that will result in an unacceptable loss of functionality. They must be run as soon as practicable. When development is complete, they should be run again. Priority 1 and 2 tests should be executed immediately before deploying the application in production.

- **Priority 3 events place a financial institution's customer applications in an operationally compromised condition**, with a diminished capacity to process transactions. Perhaps an ATM or kiosk is declining all transactions, or a fault has occurred that may not be fatal, but the defective terminal-driving application takes the unit out of service. Similarly an institution may experience periodic communications failure to a POS/ATM authorization network, and its stand-in algorithms fail due to inadequate testing. Although the institution may not suffer as great a financial loss as from fraud events, it can experience a loss of revenue nonetheless—and to some degree, loss of customer confidence, which may result in financial losses which are more difficult to measure.

Tests for Priority 3 events guard against failures that may result in minor loss of functionality, but can likely be remedied by intervention by support staff. These tests should be run after development in this area is complete.

- **Next, Priority 4 events are those that would be desirable to test if the schedule permits.** For example, the addition of multiple languages, or new “bells and whistles” features or functionality, designed to attract a new population of customers, may not be available due to application failure. When the new functionality is properly tested prior to deployment, these features are readily available and serve to enhance the institution's competitive advantage. However, if EFT testing is rudimentary and failures result, these new customers are dissatisfied by the service they receive from the institution.

Tests for Priority 4 events guard against failures that may create only trivial problems. While it would be desirable to run these tests, they need not be completed if time or resources are severely constrained.

- **Finally, there are Ad hoc tests.** These tests don't show up on our risk assessment graph (and may not show up in your initial test plan), but are still considered essential to comprehensive EFT testing. Often these tests are variations of scheduled tests—tests that are the result of “what if...” thinking on your part. For example, you might add some Ad hoc tests because of problems or anomalies you uncover during the execution of your test plan.

## Designing the Best Tests for Each Electronic Payment Event

After the events are prioritized according to the risk to the financial institution, the next step is to determine which test cases will best verify the features or functions associated with the prioritized events. Test cases describe inputs, actions, or events, as well as expected application responses. The process of developing test cases requires detailed analysis of application operation, so this development is best assigned to personnel most familiar with the industry in which an application is to be deployed.

## Conclusion

By determining which tests are critical to the proper operation of applications, and then prioritizing those tests to take maximum effective advantage of your EFT testing resources, your organization can ensure that applications are deployed with a minimum of “undesirable features” (commonly known as bugs). With careful planning and prudent use of time and personnel resources, your organization can implement upgraded and new feature-rich applications with fewer post-production headaches.

## About Automated EFT Test Tools

If you intend to test early and often, it makes sense to try to automate the reliable and repeatable manual test cases that you have created. With an automated test tool, you can quickly run (and re-run) scripts of detailed test cases on demand. However, with the best automated test tools, you are not limited to simply playing back a rote test script. Imagine taking your current manual tests and manipulating fields, messages, and transaction sequences—enhancing your current test plan or expanding it with completely new tests. In this way, you are not only testing *faster*, you are also testing *more thoroughly* than you could previously.

## About Paragon Application Systems

Paragon Application Systems is a leading global provider of ePayment simulation, configuration and testing software tools to the financial industry. More than 400 financial institutions in over 80 countries use Paragon tools to improve quality and reduce time-to-market. Paragon’s broad customer base includes major interchanges, processors, leading software providers, banks and credit unions. Visit Paragon Application Systems at [www.paragonedge.com](http://www.paragonedge.com).

## About the Author

Lynn Tjepkema, ePayment Specialist in Paragon’s Professional Services department, is responsible for analyzing customer requests and developing working solutions to address testing needs. Lynn is ISTQB certified and has over 20 years of software development and testing experience in the financial services industry. She has extensive knowledge of financial network message protocols as well as authorization and settlement processing, with special emphasis on Visa, MasterCard and American Express.