

Case Study

Postponement of High-Risk Features

Cliff Berg

President, Expressway Solutions LLC



Table of Contents

Background.....	3	Sensitivity Analysis.....	9
The Problem.....	3	Discussion of Results.....	10
Assembling the Sources of Opportunity, Cost, and Risk.....	3	Developing Consensus.....	10
Assembling the Model.....	5	Summary and Conclusions.....	11
Defining Scenarios.....	6	Resources.....	11
Simulation.....	7	Author Bio.....	12
Collaboration With Stakeholders.....	8	About Expressway Solutions.....	12

Background

Organizations that execute IT projects to enhance existing business capabilities or build new ones routinely make implementation decisions that have a major impact on the long term value of the capability. Yet, such decisions are not typically made using analytical techniques. Instead, implementation decisions are usually made in an ad-hoc manner based on the experience and judgment of technical staff.

IT departments need to develop skills to increase the quality of major implementation decisions in order to ensure that those decisions represent the best interests of the organization. The challenge is that there is an existing set of analytical models for analyzing IT implementation decisions. Some baseline models, templates, or examples would help. This whitepaper provides an example of how to build an analytical model for a very common IT implementation decision: whether to postpone some high-risk system features.

The Problem

A for-profit company has a project to build a new customer-facing service that is expected to generate substantial revenue. The **development team has identified two choices** for building the service. One choice is to build the service's core features and release them for use by the customers, but postponing certain important security features that are considered to be difficult to implement. The security features would be implemented in a subsequent "security release". The second choice is to build the security features while the core features are developed.

The **first choice** is considered to put the entire project at risk, because the security features are complex and if development does not go well, release of the service to the customer could be substantially delayed. This

would have the cost of losing revenue, as well as losing some time to market advantage while competitors work on similar services.

The **second choice** is therefore less risky from a development standpoint, but it incurs the risk that a security-related failure will occur before the security release is developed. Such a security failure could cost the company significantly in terms of the trust of its customers, and therefore result in a loss of future revenue.

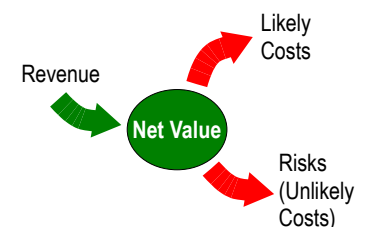


The development team has not been able to quantify any of these risks, and so management does not have a means of balancing the various factors to make a decision about which choice to select. In response, the project architect has proposed developing a model that accounts for the known factors so that the architect can work with the business to acquire a more **quantitative understanding of the tradeoffs**.

Assembling the Sources of Opportunity, Cost, and Risk

The first thing to do when developing an economic model is to identify the various sources of opportunity (positive value), cost, and risk. A risk is merely a potential for loss: that is, it is a cost that is uncertain – and hopefully unlikely. A cost is merely a negative value.

In general, predictions about the future are uncertain, and so even values and costs have uncertainty. We



differentiate between a risk and a cost because a cost is expected to occur, whereas a risk is a cost that is not expected to occur, but that might occur. There is a gray line between what is a risk and what is a cost.

Opportunities

In the situation presented here, the only source of positive value is the opportunity for future revenue.

Revenue

The business claims that once the new service has been deployed, it will generate \$2M/year +/-40% in the first year of deployment, growing at 20% each year, but decreased by 20% for each year of delay, assuming that there are no publicized security failures. If, however, a publicized security failure occurs, revenue is expected to drop by 50%, and permanently remain at 50% below what could have been achieved. The business asserts that these numbers are supported by market analysis.

Costs

Costs are usually the most clear-cut things in a business system, especially direct costs. In our example, we are treating loss of revenue due to a security failure as a risk, so that is effectively our “indirect cost” bucket. The direct costs identified are:

1. Development of core features.
2. Development of security features.
3. Deployment of initial release.
4. Deployment of security release, if any.
5. Operation.

Development of Core Features

The development team estimates the cost for development of core features, without security included, will require six months (@ \$10K/day). An analysis of the historical uncertainty of such estimates by that team indicates an uncertainty of 30%, where “uncertainty” is considered to represent two standard

deviations. These uncertainties are well within industry norms.

It is estimated that building security features into the initial release would increase the expected development time of that release by 50%, with an uncertainty of 80%. This increased uncertainty represents the development risk associated with that strategy.

Development of Security Features

If security is built later, the special security release is expected to take 70% of the original core feature development time, with an uncertainty of 90%.

Deployment of Initial Release

Deploy costs for the initial release without the high risk security features are expected to be \$200K with an uncertainty of 40%.

Deploy costs with security security built into the initial release are expected to be 40% higher than without the security features, with an uncertainty of 40%.

Deployment of Security Release

If the high risk security features are deferred, the deployment costs for the subsequent special security release are expected to be 30% of expected core system deploy costs with an uncertainty of 40%.

Operation

Operating costs are expected to be the same with or without the security features, and are expected to be \$1M/year with an uncertainty of \$200K/year.

Risks

Two types of risk are identified:

A risk is merely a cost that is improbable.

1. Development risk.
2. Security risk.

Development risk has been addressed through the uncertainties identified for the development time. Therefore, we have only security risk remaining to consider.

Security Risk

The Security Group within the company tracks security incidents and assigns them a Severity level from 0 to 5, where 0 is an incident of no consequence and 5 is the highest. The security group also tracks the rate of incidents over time.

With security features, the mean Severity of incidents is expected to be 0.2 with variance of 0.25. Frequency of incidents is expected to average 2/day with an uncertainty of 2/day.

Without security features, the mean security incident Severity is 1.1 with an uncertainty of 0.2. Without security, the frequency is expected to be 4/day with an uncertainty of 3.2/day.

Reflection On the Accuracy of Data

By this point you have no doubt noticed that much – if not most – of the data that will be input to the model consists of *little better than educated guesses*. This is the nature of business: business is about predicting the future and capitalizing on it, and the one who makes the best prediction has a tremendous business advantage because they can then make better decisions.

Even so, it is important to *acknowledge the uncertainties* inherent in these educated guesses, and that is why we have explicitly documented each uncertainty as best as we can. The uncertainties will be explicitly included in the model, which is a statistical model, and so *our result will have an explicit uncertainty*. This uncertainty is important for planning because it is helpful to know the level of

certainty around predictions.

Assembling the Model

An *Expressway*[™] model of the situation is shown in Illustration 1.

In this model, each chevron shape represents a source of events. These events represents inflows of value or outflows of cost. Each source has a statistical distribution, defined by the data that we have collected so far. The component at the top right is a “tally” for adding up all of the positive and negative events. If we simulate this model for a specified planning horizon – say five years – the value accumulated by this tally will represent the expected value that has been retained. If we divide this value by the money invested, we will have an expected ROI.

The arrows connecting the various shapes are pathways along which values (or costs) can flow. In some cases, the flow of a value (or cost) triggers some process that eventually produces another event. This is the case when an arrow enters a chevron on its left side: it means that when an event enters the chevron through that pathway, then the chevron causes an event of its own to occur at some time in the future. Thus, we have a cascade of events causing other events.

When an event enters a chevron through its top side, it means that the event modulates the chevron in some way, e.g., by adjusting the rate at which the chevron produces events, or the magnitude of its events.

Finally, a circular arrow present in a chevron indicates that the chevron produces events on an ongoing basis, and not only when it is triggered from the left.

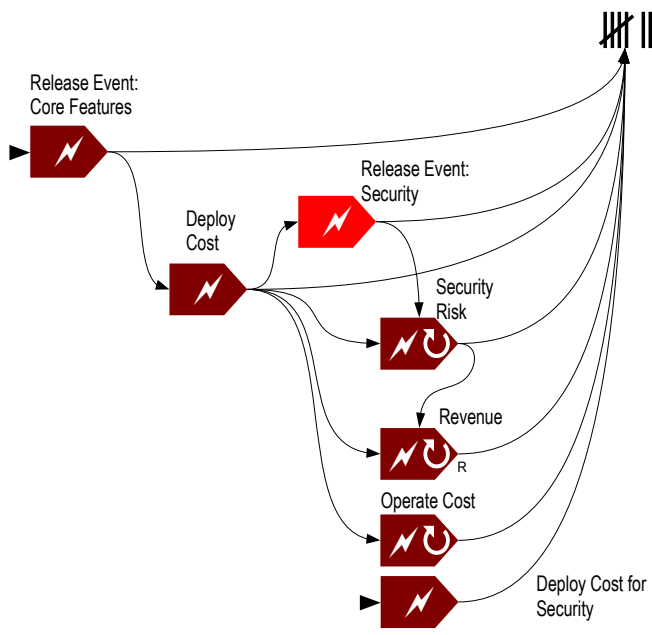


Illustration 1: Expressway™ model.

The flow occurs as follows: An initial event (indicated by the arrow entering the first chevron) represents the start of development, and at some time later the first chevron produces an outgoing event, indicating the cost of development. The value is therefore negative, and it flows into the tally. In addition, it flows into the Deploy Cost chevron, triggering it, and that chevron immediately generates an event representing the cost of deployment. That cost flows into the tally, and also triggers other processes (chevrons) that produce events.

Finally, assuming that a separate security release is built, a future event (at a pre-defined time) causes the “Deploy Cost for Security” chevron to trigger a cost event that is tallied. The timing of this event is not important.

The “Release Event: Security” chevron feeds into the top of the “Security Risk” chevron, modulating it. This adjusts the rate and magnitude of security incidents that occur, which are represented here as events emanating from the “Security Risk” chevron. The “Security Risk” events in turn modulate the

“Revenue” chevron, since a large security failure will cause a long-lasting drop in revenue.

We have not discussed the details of how modulation are specified, or how the magnitude and frequency of chevron events are specified. That is a detail of how to use *Expressway™*; but suffice it to say that *Expressway™* allows you to define all of those things, including specifying conditional expressions for the values that are generated. Therefore, all of the requirements for our model can be represented in *Expressway™* using event chevrons, which *Expressway™* calls “generators” (because they generate events).

The Time Value of Money

So far we have not accounted for the fact that a dollar to be received in the future is worth less to us today than a dollar received today. The reason is that if the dollar is received today it can be invested.

Expressway™ handles this by allowing you to define sources of investment capital, each with its own “cost of money” rate;¹ and it also allows you to discount values based on when they are accrued. (This is essentially a “net present value” type of calculation, although *Expressway™* does the calculation during simulation rather than using a formula.)

The time value of money turns out to not be very important for most software development investments. This is because software development projects are usually relatively short term – a year or two – and because the risks and opportunities involved are usually so large that they overwhelm any effects due to the time value of money. Therefore, the time value of money is not included in this case study: it is a second order effect.

Defining Scenarios

We define two scenarios to model: one for each choice that we have. That is, we define the following

¹ This feature has not yet been implemented.

two scenarios for our model:

1. Build security now while core features are also built.
2. Build security later, as a separate release.

Each scenario will be specified as a set of parameters for our model.

Scenario 1: Build Security Now

The first scenario essentially nulls out the Security Release event by specifying zero values for its probabilities. The resulting parameters for the first scenario (“Build Security Now”), based on the data explained in the section “Assembling the Sources of Opportunity, Cost, and Risk,” are given in Table 1.

Description of Value, Cost, or Risk	Estimates
Release Event: Core Features	$m_i=270\text{days}$, $2\sigma_i=216$ $m_v=1$, $2\sigma_v=0$
Deploy First Release	$m_i=0$, $2\sigma_i=0$ $m_v=\$280\text{K}$, $2\sigma_v=112\text{K}$
Security Risk	$m_i=2/\text{day}$, $2\sigma_i=2$ $m_v=0.2$, $2\sigma_v=0.25$
Revenue	$m_i=1\text{yr}$, $2\sigma_i=0$ $m_v=\$2\text{M}/\text{yr}$, $2\sigma_v=800\text{K}$
Operating Cost	$m_i=1\text{yr}$, $2\sigma_i=0$ $m_v=1\text{M}/\text{yr}$, $2\sigma_v=200\text{K}$

Table 1: Values, Costs, and Risks for Scenario 1.

Probability distributions are chosen to fit the data in Table 1. *Expressway*™ provides a tool for fitting the probability distributions, so that no mathematics is required.

Scenario 2: Build Security Later

The second scenario (“Build Security Later”) defines non-zero values for the probabilities pertaining to the Security Release, and adjusts other parameters based again on the data explained in the section “Assembling the Sources of Opportunity, Cost, and

Risk.” The parameters are listed in Table 2.

Description of Value, Cost, or Risk	Estimates
Release Event: Core Features	$m_i=180$, $2\sigma_i=54$ $m_v=1$, $2\sigma_v=0$
Release Event: Security Release	$m_i=126$, $2\sigma_i=113.4$ $m_v=1$, $2\sigma_v=0$
Deploy First Release	$m_i=0$, $2\sigma_i=0$ $m_v=\$200\text{K}$, $2\sigma_v=80\text{K}$
Deploy Security Release	$m_i=0$, $2\sigma_i=0$ $m_v=\$60\text{K}$, $2\sigma_v=24\text{K}$
Security Risk	$m_i=4$, $2\sigma_i=3.2$ $m_v=1.1$, $2\sigma_v=0.2$
Revenue	$m_i=1\text{yr}$, $2\sigma_i=0$ $m_v=\$2\text{M}/\text{yr}$, $2\sigma_v=800\text{K}$
Operating Cost	$m_i=1\text{yr}$, $2\sigma_i=0$ $m_v=1\text{M}/\text{yr}$, $2\sigma_v=200\text{K}$

Table 2: Values, Costs, and Risks for Scenario 2.

Simulation

We now have two scenarios that we can simulate over a planning horizon for comparison. If we use a five-year horizon (1825 days), and simulate each scenario ten times using *Expressway*™, we obtain two histograms: one for each scenario. These are shown in illustrations 2 and 3.

In the illustrations, the horizontal axis is the “bucket” for the final net business value after five years, and the vertical axis is the number of simulations that fell within each bucket. For each illustration, the first bucket has a range of \$2.5M-\$2.6M, and each next bucket begins where the prior bucket left off. The final bucket has a range of \$3.4M-\$3.5M.

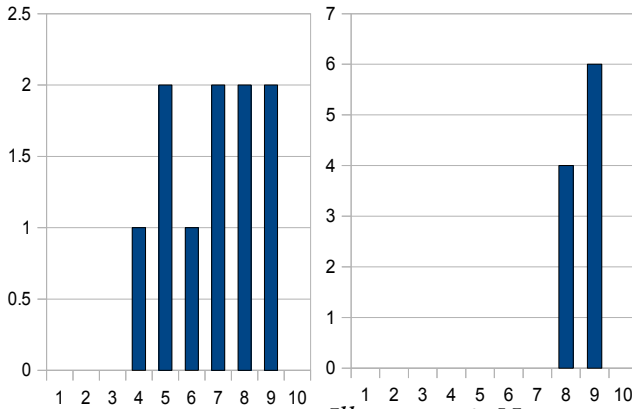


Illustration 2: Net Business Value, Scenario 1
 Illustration 3: Net Business Value, Scenario 2

As one can see from these figures, the expected value of Scenario 2 (Build Security Later) is the higher of the two scenarios. However, there is uncertainty: Scenario 1's histogram has incidents in bucket 4 (\$2.8M-\$2.9M) and also in bucket 9 (\$3.3M-\$3.4M) and each intervening bucket, whereas Scenario 2's histogram has incidents in buckets 8 (\$3.2M-\$3.3M) and 9 (\$3.3M-\$3.4M): so the two histograms overlap in buckets 8 and 9. That is, the simulations show that it is possible that Scenario 1 *could* result in an outcome that is higher than the expected outcome for Scenario 2, even though it is unlikely.

Discerning Trends

The expected outcome at the end of the planning horizon is not the whole story: we should also be interested in the trajectory of net value. Illustrations 4 and 5 show the predicted trend of expected value over time for each scenario, respectively.

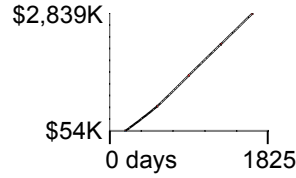


Illustration 4: Trend for Scenario 1 (Build Security Now)

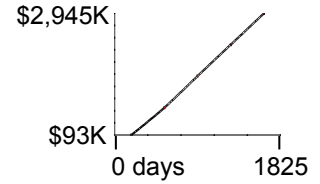


Illustration 5: Trend for Scenario 2 (Build Security Later)

As can be seen from these figures, both scenarios show a positive net value trend (i.e., a money-making one) at the five-year point. The slope of the trend indicates the rate of net income.

A simple calculation reveals that the trend of scenario 1 at five years is \$723K/yr., whereas the trend of scenario 2 at five years is \$726K/yr. Thus, while scenario 2 ends with a steeper trend, it is only very slightly steeper. The implication is that the effect of the early security decision wanes over time.

Collaboration With Stakeholders

Each model parameter is an assumption, and the old adage “garbage in, garbage out” certainly applies.

In some cases it is possible to derive model parameters based on existing data about the business. This is the goal of business intelligence: to develop insights by looking at extracts of actual operational data.

Unfortunately it is often the case that useful data is not available for some parameters, and so informed “guesses” must be made by subject matter experts (SMEs). Such SMEs are usually reluctant to make any numerical guesses, but allowing them to specify ranges of uncertainty usually puts them at ease. Thus, they are often very willing to say something like, “We encounter that situation at least 2 time a day, and at most 6 times a day, and if I *had* to guess, I would say that it averages about 3 times a day.” The range that they provide in this way is the basis for estimating the parameters for your model, since you can make reasonable assumptions that, e.g., the “at least” value

is, say, two standard deviations from the most likely value. From this, you can derive a probability distribution using a tool that is built into *Expressway*™.

Even so, it is important to reflect on each parameter for the model that you develop, and discuss the parameters with stakeholders. This reflective process is best done once you have a basic model up and running so that you can demonstrate output. This makes things more concrete for stakeholders so that they can consider the model's output and then challenge each parameter.

Sensitivity Analysis

It is important to determine how sensitive the predicted results are to initial assumptions. If one knows the sensitivity to the parameters, one can be more diligent in validating the parameters to which the model is most sensitive.

To keep this paper short, we will focus on the question of how the projected time to develop features affects the predicted outcome. In a real world setting we would be concerned with more than this, and we would try many different variations on model parameters to assess the sensitivity of the results to each parameter.

Given our narrow focus here, we will vary the following parameters in each scenario:

- a. The time to develop core features. (Plot E(ROI) versus this time.)
- b. The time to develop the security release. (Plot E(ROI) versus this time.)

We will define the additional scenarios listed in Table 3 in order to perform this sensitivity analysis on the time to develop core features.

Additional Scenario	Description
1.a.+	Same as 1, but with m_t for core features 10% larger.
1.a.–	Same as 1, but with m_t for core features 10% smaller.
2.a.+	Same as 2, but with m_t for core features 10% larger.
2.a.–	Same as 2, but with m_t for core features 10% smaller.

Table 3: Additional scenarios for determining sensitivity to the cost of development of core features.

We will also define the additional scenarios listed in Table 4 in order to perform sensitivity analysis on the time to develop the security release.

Additional Scenario	Description
2.b.+	Same as 2, but with m_t for the security release 10% larger.
2.b.–	Same as 2, but with m_t for the security release 10% smaller.

Table 4: Additional scenarios for determining sensitivity to the cost of the security release.

These additional scenarios can then be simulated and compared in order to determine how different their predictions are given a 10% change in assumptions. The results of these simulations are shown in Illustrations 6 through 8.

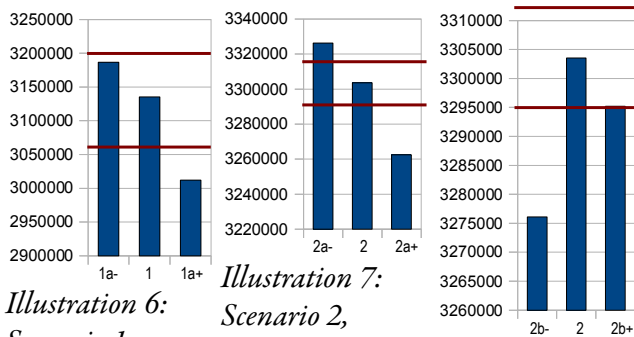


Illustration 6: Scenario 1, varying time to develop core features.

Illustration 7: Scenario 2, varying time to develop core features.

Illustration 8: Scenario 2, varying time to develop security release.

Illustration 6 reveals that slightly increasing our assumption in scenario 1 about the time to develop core features by 10% results in a slightly lower expected return over five years. The horizontal red colored lines are one standard deviation apart, showing the level of uncertainty. It appears from Illustration 6 that the slight downward sensitivity is of about the same magnitude as the uncertainty. Therefore, we do not need to be more than about 10% correct in our assumptions about the cost of core feature development in order to be able to trust our projections.

Illustration 7 provides the same sensitivity analysis, but for scenario 2. That is, if we are 10% off in our estimate of the time to develop core features, what will be the impact on the projections for scenario 2? Illustration 7 shows that a 10% error will have a slightly larger effect than the standard deviation, but not overwhelmingly so.

Illustration 8 provides the sensitivity for scenario 2's assumptions about the time to develop the special security-related release. It appears that the range of uncertainty is similar to the variation in the results produced by increasing and decreasing the estimated cost by 10%. However, this is a very small range of actual total value: about \$30K.

The sensitivity analysis shows us that our projected results are fairly stable with regard to small changes in assumptions, and that a 10% error in those assumptions will change the results by about one standard deviation.

Discussion of Results

Based on the analysis, the following conclusions are evident:

- Given the numeric assumptions presented here, the expected ROI is higher if one defers risky features and focuses on early deployment of core features.
- The expected total return is moderately sensitive to the actual time required to build and deploy the core software: a 10% error of assumptions results in a variation of about one standard deviation, or about \$0.2M for errors in core feature estimation.
- The expected total return is not vary sensitive to the actual time required to build and deploy the security release.

These results do not imply certainty, but rather are projections (see illustrations 6-8) that scenario 2 is consistently better even if assumptions are wrong by 10%. As projections, they can be wrong. Business is about predicting the future: if there were a foolproof method, then there would be no more business opportunities. Uncertainty is the cornerstone of business.

Developing Consensus

It is extremely important that all stakeholders invest in and support the model that is finally produced. By developing the model with the stakeholders you can ensure their investment; and if they are comfortable with the model's predictions then they will be supportive of it. This consensus development process

is something that should be facilitated by an experienced manager, facilitator, or consultant, in order to ensure that personalities, political considerations, and other real-world challenges are effectively dealt with.

Not only is it important to achieve consensus, but I have found that one should help stakeholders to acquire a “feel” for the model's predictions. The model's results should be understood

The primary value of a model should be to enhance people's understanding of cause-and-effect relationships – *not to replace understanding with calculations.*

intuitively. Sensitivity analysis can be helpful for this purpose: stakeholders should be able to predict what effect – and *how large an effect* – small changes in assumptions should have on the projected results. Otherwise, they do not truly understand the model. An intuitive understanding is necessary for people to fully accept a model. In the end, the primary value of the model should be to enhance people's understanding of cause-and-effect relationships – *not to replace understanding with calculations.*

Summary and Conclusions

This analysis work is not easy to perform, and it requires experience, knowledge of the techniques, care, collaboration, trial and error, and considerable time to prepare and develop a trustworthy model and consensus around it. A considerable investment is therefore required in order to build the skills to do this kind of analysis within an IT organization.

The benefits of doing so include the following.

1. Results can be obtained that would be hard to compute without simulation. In the example presented here, the time until

deployment – and therefore the time until benefits start to accrue – is very uncertain, and therefore one cannot simply add up the expected value of each cost and benefit.

2. Simulation enables quick turnaround what-if analysis when assumptions are challenged.
3. The uncertainty of the result is explicit and quantified.
4. The sensitivity of the results to particular risks, such as the time to develop certain features, is explicit and enables decisions about what features to defer.
5. The impact of risk mitigations can be assessed by the model, simply by adjusting the risk-related parameters according to the level of risk mitigation.
6. The flow of value over time (see illustrations 4-5) is valuable for preparing stakeholders for what to expect, and for identifying cash flow constraint exposures.
7. IT practitioners become more focused on value and have a powerful new way of communicating with the business side of the organization.

Resources

Value-Driven IT. The book *Value-Driven IT* explains the concepts behind the analysis used in this article.

Principles of Corporate Finance, by Brealey & Myers. This is the standard reference for business analysis.

Author Bio

Cliff Berg is President of [Expressway Solutions](#), and is a consultant in organizational change management with respect to improving the effectiveness of the IT function. Cliff is the author of four books, most recently *Value-Driven IT* and before that *High-*

Assurance Design. Previously Cliff was co-founder and CTO of Digital Focus, now Command Information. Prior to that Cliff worked for Intermetrics Inc. and CAD Language Systems Inc. where he built compilers and simulation tools. Cliff has a ME in Operations Research from Cornell University, as well as degrees in physics and nuclear engineering.

About Expressway Solutions

Expressway Solutions was formed to develop the *Expressway*™ product and to provide assistance to organizations to implement the use of analytical processes for IT decision-making. Expressway Solutions is privately held.