

“THE BOTTOM LINE

Breadth and repeatability are the key points to remember when deploying or upgrading an application. Help a lot of people many times, and you increase the likelihood that you'll receive a positive return from your technology investment. Conversely, avoid applications that do not have breadth and repeatability — you're likely to spend more than you'll receive.”

ROI Maximizing: Breadth and Repeatability

Is there a key to maximizing the return on investment from a software application? It turns out there is, and it can be summed up in two words: breadth and repeatability. Throughout the process of choosing and deploying an application, continue to evaluate the project in terms of breadth and repeatability and you will increase your chances that the application will deliver a positive return to your company.

Breadth

Rule 1: The more people affected by the application, the higher the potential ROI. This sounds simple enough: touch a lot of people and you increase the impact of the application. Using this rule, even a simple application can offer great returns.

By its nature, the corporate intranet delivers broad reach across the company and offers an excellent platform for a high ROI application. One example is an intranet-based purchasing application one company developed. Instead of a paper-based request employees go to an internal web page and choose stock supplies such as pens and paper. Pictures and descriptions are really links to the outside supplier that is responsible for keeping the information current. Once submitted, a workflow process is started for approval, ending in an e-mail to the supplier with a request to drop-ship the items the next day to the employee's desk. This application saves a small amount of time compared with a paper request but it does so across an entire company, making the investment in the application development pay back in increased company productivity.

Repeatability

Rule 2: The more often you help the employee, the greater the ROI. Are you deploying an application that will help the employee in their daily activities, or once every three years? Obvious, right? But Decision Interface has seen numerous examples of applications that violate this rule.

ROI Maximizing

One common example is the 401K plan. For tax reasons, employees in the United States often participate in a work-sponsored savings plan. When you join a company you elect to deduct a percentage from your paycheck and you select mutual funds in which to invest that money. You can make changes to these selections throughout the year. Here's the problem: it may seem like a good idea to eliminate the paper forms used in this process and allow employees to change these selections using an intranet site—it helps every employee so it has breadth. What it doesn't have is repeatability. In practice, few employees make changes more than once a year, making the time spent re-learning the application greater than the savings. In most cases, hiring another employee in human resources is more cost effective for the company than creating this type of application.

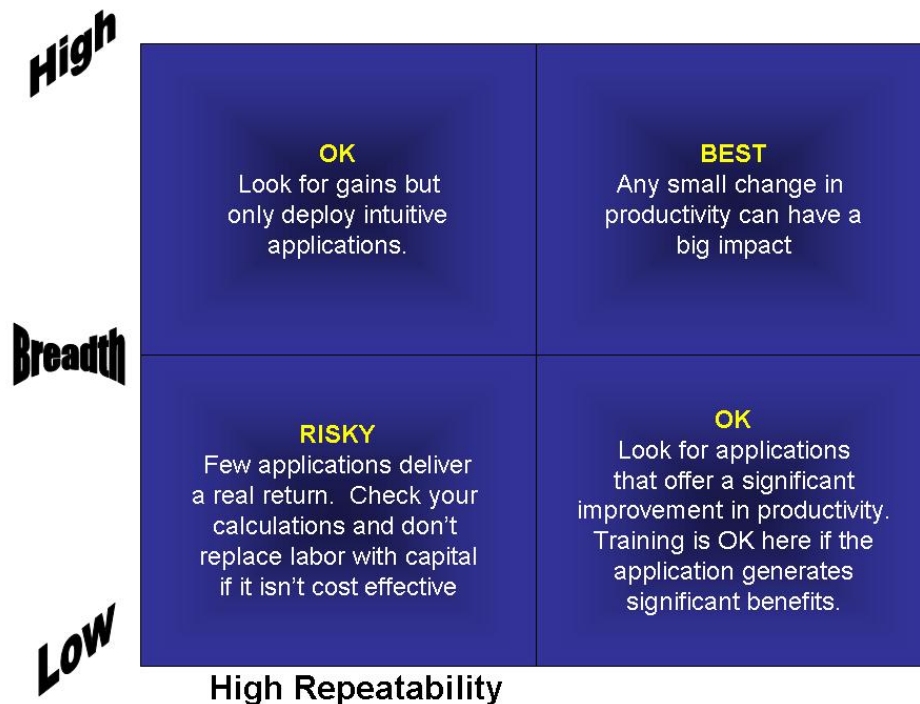


Figure 1: Breadth and Repeatability Model

The Application

The rules of breadth and repeatability apply even after the application is deployed, as features within applications offer varying opportunities for returns. For instance, the real benefit from a document management system is often found in the advanced searching features those applications offer the user -- bringing

ROI Maximizing

powerful searching to the employee increase the reuse of information and thus productivity. Therefore, keep the searching features in mind when training these employees and focus training effort there rather than areas that offer less benefit (e.g. merging versions).

Upgrading to the next version of an application requires a similar look at breadth and repeatability. Are there features within the new version that provide breadth and repeatability? Small changes to minor application features may not warrant the cost in time and effort to upgrade.

Financial calculations are still important in understanding the real returns possible from an application and in effectively targeting the deployment of features to maximize return. Still, the basic rules of breadth and repeatability apply. Keep these rules in mind and you are unlikely to embark on an effort that in the end will offer little in the way of real returns