

Successful Software Upgrades

Author: D. Sommer

Date: October 18, 2002

Short Description: Upgrading software systems can be as risky to your business as implementing a brand new system. These 10 step best practices will reduce your risk and increase your chances for a successful production cut over.

Do you remember the first time you purchased a software application for your business ? From start to finish it took you a year or more, there were many modifications made to the software since your business is unique, and afterward there was a big party because the system was finally installed. After the software was in production you performed daily support and never thought of future upgrades.

Now, fast forward 12 months. There is a new release available, and you would like to upgrade your software system. You think back to the initial software evaluation, when the vendor told you that they provide new releases every 6 to 18 months and all you need to do is load the new release. Piece of cake.

Well, the truth is, a software upgrade is not as simple as you would hope. Software upgrades will require much of the same work performed during the original implementation and must be managed as a “project” to limit the risks to your business.

Lets take a look at the costs, timeline, and steps required for a software upgrade.

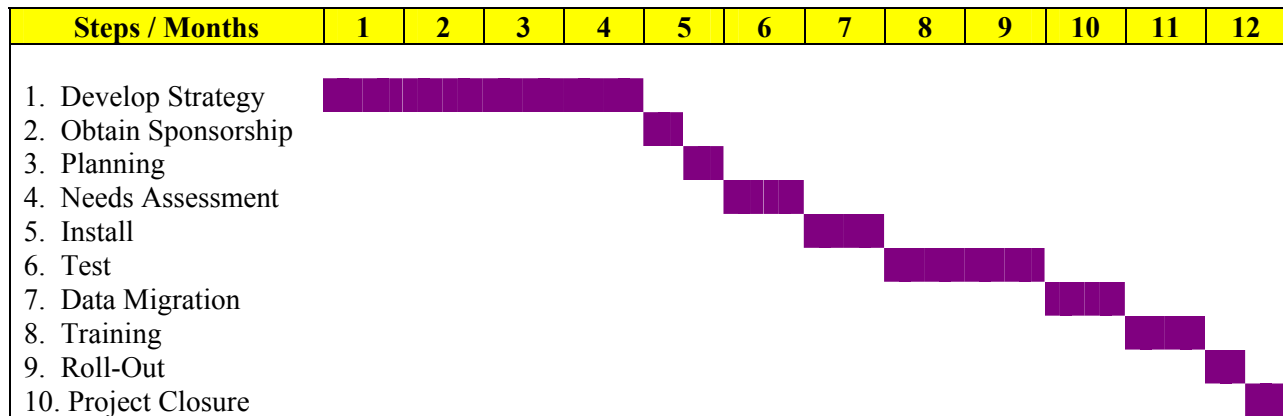
Upgrade Cost Allocation

Upgrading a software system will require more capital than your annual maintenance contract. When reviewing upgrade costs, keep the following cost allocations in mind. These statistics show how most companies spend their money on software upgrades.

Upgrade Component	% of Total
Hardware	33 %
Professional Services	26 %
Employee Labor and Training	19 %
Software for Upgrade	11 %
Additional Software	11 %

Upgrade Timeline

The time spent on upgrading software varies little between small and large systems. From the time you begin thinking about upgrading to getting it installed, you will spend more than 1 year on this project. We have provided an average upgrade timeline below.



10 Steps to Success

Treating a software upgrade like other new implementation projects is the first step to avoid a potential disaster. The following ten best practice steps will ensure the upgrade project runs smooth, is implemented on time, and within your planned budget.

1. Develop Strategy

Software upgrades (ERP Systems, databases, warehouse systems, etc.) are expensive and time consuming. These projects require a thorough examination of the available options, costs, and timelines. A detail business case and return on investment should be developed to justify the upgrade project.

During this step consider:

- Was the current version heavily modified? If yes, determine if the effort required to implement this upgrade is worth the future benefits.
- Do the new features justify the cost and resources required for the upgrade.
- Do the new features better match your business needs or should you consider a different vendor solution.

2. Obtain Sponsorship

Once the strategy, business case, and return on investment is complete, obtain sponsorship (authorization) to move forward with the project. There are three primary sponsors that should be targeted: 1) Business Sponsor 2) IT Sponsor 3) Financial Sponsor. Depending on your organization, this could be the same person or three different individuals.

3. Planning

Software upgrades are costly, time consuming, and adds additional risk to your organization. With that said, treat this as a typical IT project. A project manager, business and IT resources should be assigned. Project roles should be defined. A detail plan of activities and deliverables should be developed covering the following seven steps.

4. Needs Assessment

Software upgrades are the perfect opportunity to eliminate customized code and replace it with standard functionality from the software vendor. This is also the perfect time to select new features that will benefit the business. A “Gap” analysis should be performed with the business units to determine what features can be eliminated or changed and what new features best match the business strategy.

5. Install

Out of all the steps, this will be the easiest. Purchase and install all the new hardware and load the software.

6. Test

Now for the most important step. Testing will make or break you software upgrade implementation. Some software solutions like Enterprise packages are highly integrated with other modules. Make a minor change in one module and that output funnels through the rest of the system. Testing these changes can be very time consuming. Also, take into account that new releases have historically contained large quantities of software bugs. Taking these two factors into consideration, you can understand why proper testing is critical to avoiding project failure.

7. Data Migration

This is the process where you map and cleanse the data for the new system. For example, you have two different versions of a product record that must be reformatted for the new software database. Complexity, duration and cost of this step will be determined by new software version requirements and the quality of your current data.

8. Training

No software upgrade is successful unless the end user's needs and objectives have been met. Minor changes to the software can have major impacts on the business process or the look and feel of application screens. Training should take each user through their job function showing them how their day to day work will be completed.

9. Roll-Out

This is the time when you turn on the new system and shut down the old. The roll-out usually occurs during a long holiday weekend when the project team works around the clock.

10. Project Closure

Once the new system is up and running, the project team will perform a "Project Post Mortem". During this process the team will review what went well, what should be done differently next time, compare cost and duration actual to the budget, and obtain a final project signoff from the project sponsors.

