



12 Deadly Mistakes of Software Selection

#1 – Buying the same software as your industry peers

- First, it assumes that industry peers are qualified to make good software selection decisions.
- Second, it excludes knowledge of how your peers made their selection, what their criteria was, their budget constraints, user needs and so on.
- Lastly, it fails to consider your own unique requirements and differences.

#2 – Buying software based on features alone

- When you buy software on features alone you may overlook other critical factors such as:
 - Scalability – software’s ability to grow with your company.
 - Flexibility – ability to meet your changing needs, rather than you having to conform to the software.
 - Excess Complexity – the software is too complex and cumbersome for you to effectively implement and manage.
 - Technology Fit – you don’t have the technical sophistication to use and manage the software.
 - Cost – the total cost of ownership is more than you can afford.
 - Support – insufficient support resources exist where and when you need them.
 - Cultural Fit – the software, software vendor, and/or implementation vendor are so incompatible with your business culture that you are not able to build a partnering relationship.

#3 – Buying software without consideration of the vendor

- It is important to consider a vendor's
 - momentum,
 - litigation issues,
 - profitability,
 - ability to successfully deliver its new product or revisions,
 - employee and senior management turnover rate,
 - employee morale/attitude,
 - business focus,
 - leadership vision, and,
 - strategic direction.
- This is the general problem with using books, databases, surveys, etc. that compare your requested features, with features the product has. It is static data that does not ebb and flow, not to mention that the data in the survey may be inaccurate or incomplete.

#4 – Buying software without focusing on the implementation partner

- It is not the software that helps you, but the implementation and use of the software that makes the software become a useful tool.
- The implementation partner is a big part of the equation.
- In too many instances, we have seen companies looking to make major software changes when there was nothing wrong with the original software, it just wasn't implemented properly or people were not sufficiently trained to use its features as a tool to do their jobs.

#5 – Look at the true cost of the software

- The true cost of the software includes not just initial acquisition and implementation but
 - Upgrades
 - Manual interventions
 - Consumption of scarce IT staff
 - Reconciliation processes for most concurrent users
 - Etc.

#6 – Buying software buy-in from the organization at large

- You need the input and buy in from the constituent parts that make up your organization. You cannot leave out IT, departmental heads, power users, or foreign branches.
- The value proposition for the usefulness of software begins in the selection process. If inside politics shape the deal, if outside interests prevail, or if a biased attitude is used the implementation will usually be problematic.
- Representative groups should be included in the process.

#7 – Buying popular software without considering all the options

- Some of the major software vendors like to turn the selection process into a popularity contest.
- Too many companies buy software because the vendor has the most customers, the greatest revenue, and/or has the highest magazine or analyst ratings.
- They fail to examine why the vendor has the greatest number of customers or revenue, what the criteria for the ratings were, and whether the software or vendor can meet their own unique requirements and needs.
- What kinds of software packages are generally left out of these popularity contests?
 - New startups with vision and momentum that is taking them well into the scope of execution.
 - Old has-beens that have regenerated themselves to the surprise of onlookers.
 - Software labeled with a description that inadequately describes its potential to your company.
 - Software that does not have a “dot” on a Gartner chart.
 - Software that is not implemented by big accounting firms.
- There are a wealth of viable options out there that could be your best buy, but are overlooked because you were only considering the popular software.

#8 – Buying software that is too complex

- People tend to overbuy not under buy.
- Many companies are so focused on the detailed functionality of the software that they found themselves with a product that was too complex for their situation.
- Their users adopted workarounds and Excel spreadsheets, which defeated the purpose of an integrated solution.
- The results were wasted money, unused software potential, and difficulty using the system.

#9 – Buying software without properly defining your requirements

- Many companies over-define their requirements and end up with a requirements document so large that they lose the clarity, focus, and manageability essential to an appropriate selection.
- Put your “differentiating criteria” of key requirements onto just a few pages. These represent the items that are unique to your organization, and help to point out why a software package is a better fit for your company.
- This helps you avoid buying software that is missing a “deal killer” requirement, thus rendering it impossible to use. It also allows you to determine why certain options should be clearly eliminated from consideration.

#10 – Understanding where the software is in its product lifecycle

- This allows you to understand the risks that you are taking with that product.
 - Products at the end of their lifecycle often have very robust functionality and a large customer base. The danger with these products is that they are written in older technology so compared with new software they are often difficult to modify, they are not as intuitive for users, they have poor reporting capabilities, and they have a high risk of being discontinued in the future. In addition, the vendor may not be providing much support for the product, may not be developing and enhancing the product, and/or is focusing its resources and efforts on its new product. If the vendor discontinues the product customers will have to invest a lot of money in licensing and implementing a new software system.
 - Products at the beginning of their lifecycle take advantage of the latest technology. The problem with these products is that because writing software is such a complex undertaking the new software usually has functional gaps and may be unstable with a number of software “bugs” or anomalies. Even if the software is relatively bug-free, new versions usually do not have adequately trained implementation experts, training resources or full technical support.
- With the many software acquisitions and mergers happening in today’s market, companies need to do due diligence on the genealogical history of the software, how long the software has been around, what the underlying technology and architecture of the software is, and what the vendor’s future plans for the software are.

#11 – Buying software that is based on dying technology

- Buying software based on a hardware platform, database, architecture and/or operating system that is declining, no longer supported, no longer taught in the schools, is too inefficient, inflexible or slow compared to the alternatives, or has too small of a customer base to survive in the future.
- This puts you into a corner with few alternative paths.

#12 – Buying software to fix business problems

- Too many organizations purchase software thinking it will solve their problems when, in fact, the problems are actually due to internal process and/or organizational issues that the software alone will not solve.
- Software is only a tool, although it may help you to focus attention on the bad processes and other internal problems that need to be addressed.