

Ten Common Mistakes with Software Adoption

This whitepaper contains ten of the most common mistakes that hinder or prevent successful rollout and adoption of any collaborative software. This list is by no means comprehensive, but is aimed at pinpointing some of the common misconceptions and oversights in the adoption process.

1. Lack of clearly defined Goals

Like the old adage says, if you don't know where you're going, you'll never know when you've arrived. Successful implementation of any collaborative software involves a lot of people, perhaps even everyone at your organization. It is imperative that you have a unified vision of the end goal. What do you want this tool to do for you? What efficiencies do you expect to gain by using this tool? How much money do you plan on saving by using the software? Without these types of goals, you have no metrics with which to measure success.

2. Limited Sponsorship

Collaborative software often impacts every level of your organization. Too often, executive sponsorship is fostered during the selection process, then all but forgotten during implementation and rollout. Executive sponsorship is key throughout the entire process, from selection through final adoption and continuous improvement. Let the executives and upper-level managers know that their continued and public involvement is desired and necessary. Lack of visible involvement from executive sponsors sends a message to all other users that the tool has little value for the overall organization.

3. No clearly defined processes

Trying to implement software without clearly defined processes is like trying to direct a caravan across the country without a map. If you are rolling out a piece of software to a handful of people, it may be possible to "feel your way in the dark," but trying to include more than a few people without designing the process interactions, is almost certainly going to fail. These processes should define who will interact with the system and in what way. They will be the basis for configuring access to the system and workflow through the tool. Without these clearly defined and documented processes, you will have little chance of widespread success.

4. Poor Communication of Processes and Responsibilities

Once the processes are defined, they need to be communicated to the entire team. Not just the executives and managers, but the users as well. Everyone should understand how the system will be used. These are not secrets to be kept by "those in the know." It is difficult, perhaps impossible, to achieve widespread adoption, if there is confusion about how to use the system. User training, custom usage guides and providing a support channel for users are all key to good communication.

5. Confusing the tool for proper training and discipline

No matter how well designed or easy to use a system is, it will not be sufficient to just let your users loose to learn on their own. They need guidance and direction. This includes

proper training and defined responsibilities and discipline. If users are not trained to use the system in a consistent manner, then they will, find different ways to use it, which it leads to inconsistent data and worthless reporting.

6. Lacking formal agreement of new responsibilities

When a new system is implemented, there are new tasks and responsibilities that naturally arise. People will be assigned to handle these responsibilities and it should be made clear to everyone who those people are and what responsibilities they have regarding the new system. This is important, not only to let users know who to contact when they have questions, but also to foster ownership at different levels of the system. Poorly defined and communicated responsibilities can lead to confusion, disinterest and, eventually, to utter failure of the adoption.

7. Believing that Managerial Mandate can drive adoption at the user level

Widespread adoption of any software requires commitment and loyalty, or, at least conformance, from the users. This individual “buy-in” is rarely motivated solely by an order to use the software. Users need to understand why the software is being implemented and they need to understand the benefits of the software. Without this, they may follow orders for a while, then mutiny. Now, that’s not to say that there won’t be dissenters; there will be, however, your implementation will be more successful if you get support from individuals.

8. Waiting for the perfect implementation

“Waiting for the perfect implementation,” means trying to get everything right the first time. It means that all users will be brought on the system in one day. It means that an organization will try to use all of the advanced features on Day One. It means that all integration with external systems will be completed before anyone will have access to the system. This approach is flawed, because it tries to do too much, too fast. Almost any collaborative software is going to involve a dramatic change in culture. Doing too much, too fast can lead to confusion and pushback. Just remember that, no matter how hard you try, no matter how many people you talk to, no matter what decisions are made, there are things about your implementation that WILL change. This doesn't mean that you should throw everyone in on the first day – far from it, but it does mean that you need to be careful to avoid “paralysis through analysis.”

9. Failure to Encourage End-User feedback

End users generally make up the majority of the software users. All too often, input from end users is ignored, or not solicited at all. This leads to implementations and design decisions that don’t meet the needs of the users. It is important to get feedback from all

levels of the organization. Executives and Managers want good reports and end users want something that is easy to use and provides value to them. Implementations without end user feedback almost certainly end in mutiny.

10. Believing that training is a one-time event

Studies have shown that people learn best by repetition. A single training, while sufficient for some, will not achieve the adoption results you expect and hope for. Many users will need to be reminded. Often it works best to have an introduction training session where the processes are explained and the product is introduced, then to have the users actually use the system for a short period of time, then to conduct follow-up trainings to allow users to ask questions, to provide additional product instructions, or to introduce advanced topics or share ideas on usage. And, of course, providing a support channel for the users that makes it easy for them to get questions answered, is key.

Conclusion

Implementing a new software system can be a daunting task. It is a lot of work and requires patience and diligence, however, you can greatly increase your chances for success by following the ideas presented above. Just keep in mind that a successful adoption doesn't happen by itself, but it can happen with organization, diligence, commitment and communication.

Additional information on successful software adoption can be found at <http://www.attask.com>.