

List the individuals and teams that will be affected by the new software. Think about how the change will affect them. What will it mean for their day-to-day work?

Are there any particular teams or individuals who are likely to be particularly challenging? Will anyone require significant training, or support changing their mindset?

Are there any particular teams or individuals who are likely to be particularly keen on the change? How can they help bring others on board?

Is there anyone outside the business who you'll need to engage? Perhaps suppliers or customers directly affected by a new system or related problems, such as downtime.

## 1. Understand your users



How will you sell the new software to people? Some people won't change unless you give them a compelling reason to do so.

Think about what's in it for them and frame it that way. Will their tasks be faster? Will boring tasks be removed? Will they have a new option to work from home?

It's never too early to start warming people up to the idea of new software. Launch day should not be the first they hear of it! Put together a quick comms schedule.

Many people have been through difficult technology projects before. Lay out the potential implementation challenges and how you plan to deal with them.

## 2. Prepare your message



Who will need to be trained on the new system? Will different teams need different levels of expertise? Will you have super-users training the rest of the team?

What kind of training will work best with your users? Is it best delivered in person? Remotely? As an online package? Is it best delivered by an external specialist or someone internal?

How long will training take? Will it be ongoing or one-off? Factor this time investment into people's schedules and performance targets.

### 3. Schedule the training



Do you need to introduce incentives to encourage people to use the new system? For example, reward vouchers for those who input all their data fully in the first few weeks.

What repercussions will there be for not using the new system? You could present this as a risk, for example that uncompleted timesheets may lead to someone being underpaid.

Will you go as far as hard consequences? For example, and providing your employment contract allows for it, a hard consequence such as withholding commission payments from salespeople if they haven't entered a sale in the system.

### 4. Encourage uptake



Have a clear plan for how you'll move data from one system to the other, and how you'll check whether it's worked properly.

Have a plan for how the new software will be integrated with your other systems. This should be carefully tested, and the risk of downtime on the old system should be built into planning.

If you're delegating implementation to someone else, make sure you've given them enough time and resources to do it well, cope with problems and make decisions as they go.

Test it thoroughly before you roll it out to everyone. If possible establish a small user group using real data. Make sure it's compliant and secure before you go live.

## 5. Make sure it works



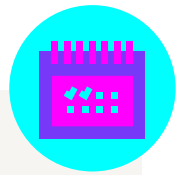
Set a transition/launch date when it won't matter so much if there's a lot of disruption, for example, over a weekend or during a period when the business is quiet.

Account for the fact that business as usual might take longer than usual for a while. Build contingency into your client deadlines and delivery promises.

Account for a period of dual running as a backup if the transition is not successful or the new software doesn't quite work properly.

Make sure your support resources – vendor helpline or internal helpdesk – are in place and accessible by those who are most likely to need them.

## 6. Go live



**With your implementation, testing and training planned out, you're all set. The next step is to see how the new software has bedded in and to make sure you're getting your money's worth.**