The Berkeley Partnership

10 things we've learned about SaaS...

Software as a service - accessed over the internet, paid for through a subscription and hosted centrally by the vendor - is increasingly replacing traditional business software running in your own data centre. The Berkeley Partnership has 25 years experience of delivering the most complex, critical, technology-enabled change. Here are 10 things we've learned about SaaS.

1 You can't customize the solution – that's the "ease/power" trade-off.

SaaS software is usually designed to be configured by non-technical people – often business power-users – and to be easy to use. However, it is usually "closed" – technical staff are prevented from modifying the solution, to protect other clients sharing the same solution from inadvertently being negatively impacted. This means that while SaaS solutions can be implemented quickly, you have to mould your business processes to fit the solution – not the other way around. You can ask the SaaS vendor to include changes on their roadmap – but that's under their control, and they'll be balancing multiple competing customer priorities.

2 You start paying from the moment you start the project.

Unlike more traditional "on premise" software – where you buy the licence, but usually only start to pay for ongoing support and maintenance at the point the solution goes live – SaaS contracts usually mean that you start paying the full "per user per month" fee at the start of the implementation project. Complex implementations with significant interface development and data transition can take time, so you'll be incurring the SaaS usage cost and the legacy system running cost in parallel. That needs to be factored into your business case.

3 Scheduled outages will be outside your control

Most SaaS vendors reserve the right to make the system unavailable for a defined series of regular outages for updates and maintenance. These may not be required, but business planning needs to assume that they will – and you need to think about how you will accommodate those outages during your planning and business case development activities.

4 Do you know where your data is?

The physical location of the "cloud" servers holding your data is a material consideration. We have recently been involved in a project where an EU-hosted SaaS solution could not be used for an organization based in Russia due to recently-introduced legislation mandating that personal information for Russian citizens must be physically stored in Russia. German privacy laws are much stronger than most other European countries. And recent legal cases in the US have cast doubt over the validity and security of "Safe Harbor[1]" provision, although most cloud vendors maintain that their adoption of "EU Model Clauses" and ISO27018 provide all the protection that their clients could want. This is a fast-changing and complex world, and the ultimate accountability for ensuring the compliance of your data lies with you.

5 Who's budget line is it anyway?

The accounting treatment of the cost of SaaS systems is often very different from traditional, on-premise software. There is no up-front spend on licences (which can often be capitalized) – the operational costs are incurred on a "pay as you go" basis - and depending on cost-centre rules, they may be incurred directly by the owning business unit (rather than the IT Department).

[1] Safe Harbor was supposed to facilitate international commerce by ensuring that EU-sourced personal data stored on servers in the US would be treated according to the EU data protection principles.

7 Not everything about SaaS is "Agile"…

Most SaaS suppliers will extol the virtues of the agility of their solution; business users can change configuration themselves, and continually iterate to get the best design and workflow. But a successful SaaS implementation will be dependent on data-loads, interfaces, and integration with other systems which don't work in that way. You can't continually iterate an interface design to a legacy system.

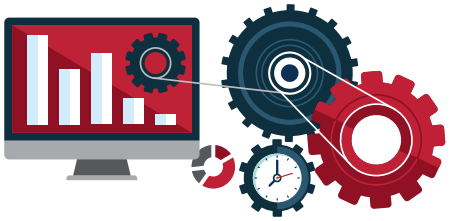
Equally, recognize that as the user of a multi-tenancy solution, you will have limited control over the release of new functionality introduced by the vendor. Although most enhancements are introduced "switched off" – and therefore their release into your own tenancy can be done at a time to suit you – we have experienced large-scale UI upgrades which are released to all users at the same time, giving each client no veto over the change, and no choice over the scheduling of implementation.

9 Don't take performance for granted

It can be tempting to think that SaaS solutions are bound to perform well, be stable, etc. In our experience, running performance and stability tests during implementation is still important. Performance can be adversely impacted by the behavior of other uses if the multi-tenancy architecture isn't well designed. And testing interface performance – at operational "peak load" while simultaneously catching up from a previous outage – is also important to ensuring operational resilience post go-live.

6 "Click here to accept terms and conditions"

SaaS contracts need to be carefully reviewed and negotiated. The fees payable are often for "the product", regardless of how much of the functionality offered is really being used. There may be significant savings on offer if you are prepared to commit to longer contract durations. Considering (and contracting for) the consequences of vendor bankruptcy may be important, particularly given how new (and financially immature) some SaaS suppliers are. And in our experience, most SaaS vendors are extremely reluctant to negotiate the detailed terms and conditions of their contracts. Their business model is geared around the scale economies of a standardized, multi-tenant offering, and they are therefore reluctant (if not unable) to accommodate bespoke services and contract terms. This can be a shock to procurement teams used to dealing with their on-premise competitors – but it is worth testing the likelihood of the controversial clauses being debated even being invoked (and therefore the real value of the negotiation).



8 Data is still difficult

The challenges of data migration (in our experience, the critical path activity on 90% of large ERP implementations) remain on SaaS projects. They can often be more difficult in the SaaS world, as SaaS solutions are often built with a "clean start" (and no historic data load) in mind. Beware the temptation to turn off all validation rules to get legacy data to load – it can give the illusion of a successful migration, but saves the challenges of poor data quality and data-related functional problems for later.

10 Think about what happens if you want to leave before you join.

Given the constraints on access to the system (you can't access the underlying database in the same way you can on an on-premise alternative), make sure that you think though a plan to extract your data and migrate it to an alternative platform, **before** you sign your SaaS contract. Contract for guaranteed access to your data and quantified lead-times for data access and support while you have some leverage over the supplier. If you only do this when you decide to leave, there is no incentive for the SaaS vendor to be in any way helpful.

If you'd like to get in touch with Berkeley and find out more about our SaaS approach, please visit our website or email dave.machin@berkeleypartnership.com