Commercial Off The Shelf Implementations

GENERAL AWARENESS

A Commercial Off the Shelf (COTS) Package is catch all term for software supplied by vendors to perform a specific business function. They are built to deliver commodity functions which fulfil a similar need across many organisations. It is usually the case that a COTS package can be tailored or configured to more closely fit the needs of a specific business.

It is most common now to see COTS packages supplied in a Software as a Service (SaaS) model where the vendor or their partner supplies not just the software but also the hosting, maintenance and full support for the implementation.

It is theoretically possible to deploy a SaaS based COTS package as a ready to use solution directly to end users. In reality many COTS packages – even those supplied as SaaS require a level of configuration before they are usable by the end users. In addition to configuration most implementations require integration with other systems within the wider business systems landscape or with external services from other organisations.

In Financial Services companies the following applications are most commonly provided as COTS packages:

- Customer Relationship Management Systems.
- Enterprise Resource Planning Systems (encompassing Finance, Purchasing and Planning).
- Policy and Claims Administration Platforms
- Document Management and Storage solutions.
- Content Management Systems
- Unified Communications Systems
- Customer Experience Management Platforms

LICENSING

COTS packages using a SaaS model are usually licensed in one of the following ways:

Named User Licence – Users are grouped into roles with the roles priced based on the functionality they can consume within the application. Each individual is allocated a licence for the software which may optionally be deployed to a new user if the original user permanently ceases to need the application.

Concurrent user licence – A set number of users are allowed to use the application at one time. It is common for this to be associated with a maximum number of users allowed to be registered for the application e.g. you can have a maximum of 50,000 users but only 500 can be actively using the application at once. This is a common model for customer facing applications. It is sometimes the case that a “slack number” is allowed where the number of users is allowed to exceed the stated maximum. This is usually monitored by the vendor and may incur future extra charges if limits are consistently breached.

Usage or Transaction Based Licensing – A set amount of transactions are allowed to be consumed in a given period. This is a common model for packages that interact with other systems rather than end users. It is often combined with the named user licence whereby each user has limited usage for a given transaction e.g. each named user is limited to 500 automated workflow actions per day.

Storage or record-based licensing – The application is paid for based on the number of records recorded in the system. This is common with Financial Services Policy Administration systems where the cost for the application is based on the number of customer policies held in the system. This model is also sometimes combined with the named user licence where each user is allocated a storage or record limit e.g. each named is limited to storing 10GB of documents.

UPDATES AND UPGRADES

Purchasing a COTS package on day 1 means you are purchasing a version of the application that will get updated and improved as time goes on. For SaaS implementations the application may update in one of two ways:

Auto Updates. The vendor will update the application automatically (sometimes monthly, quarterly or annually). You will need to test your application prior to the auto update and it is generally the case that the vendor supplies an upgraded version of your configured application in a separate instance to allow you to do this. You will also need to test interfaces still work as expected. At the point of update your application will be updated and there is generally no going back unless the problems are in the core application itself.

Customer Applied Updates. The vendor will advise customers when a new version is available and make available an upgraded version of their configured application in a new instance. The customer can test the application along with their interfaces and bespoke code. They can choose whether or not to deploy the upgrade and are normally given a choice of upgrade time slots. When they are happy they initialise the upgrade of the application. It is generally the case that the vendor mandates that a customer may not be more than a certain number of version behind the latest release and some vendors revert to an auto update for all customers on older versions.

HOW TO AVOID COMMON IMPLEMENTATION PROBLEMS

COTS Package Implementation Projects are often unfairly seen as expensive, long running and notorious for not delivering what was required. Here are some of the reasons why:

Choose the right package. A lot of time and money is wasted in turning a package into something it was never designed to be. Perform gap analysis against your required features and the standard features in the application – if the gap is too wide move on and find a different package (maybe even a different type of package).

Aim to adopt best practice business processes that come with the package. Configuration of underlying business processes is possible but remember the packages are built on best practice processes developed alongside companies for many years. If your process differs markedly from the in built process maybe you should change your process rather than the tool.

Closely examine the licensed user base needs. Many packages come with different tiers of license. It is often the case that only a small number of users require the top expensive tier, many users may suffice with a simpler cheaper licence.

Look at what you have bought before you change it. Most of the COTS packages are supplied with a basic configuration and a working sample application (sometimes available before you actually buy the tool). Socialise this with key users to identify gaps in functionality that need filling and determine how much of the core app is actually fit for purpose already.

Configure the tool – don't rebuild it. Many packages allow you to expand the data schema, configure business rules and change the user interface using straight forward low code frameworks alongside more complex code. Always default to the simpler low code where possible and minimise (or preferably eliminate) complex code.

Train everyone that is involved in the project. The people doing the configuration and deployment will obviously need to be highly skilled in the package, but the Analysts, Testers, Project Managers and key business users should also receive training Everyone involved in the project needs at least a general awareness of the package to prevent the package being pushed to do things it is not meant to do.

Implement bespoke complexity outside of the package. Most modern COTS packages are able to interface with other packages or bespoke applications and have connectors for the most popular integration technologies. Rather than configuring or coding within the package consider if there is another package available that can be interfaced to or as a last resort consider building a bespoke application that can be called from the package (or the integration toolset).

Plan your support and maintenance cycle from day 1. Handover of a completed package to support is often the task that takes the most time (there are many horror stories of SaaS packages being supported by the original project team years after implementation). Plan how the hand over to "business as usual" will be done from day 1. Involve the teams who will be responsible to ensure the package is being configured and deployed in the most suitable way to make support easier.

Involve the vendor or their partner throughout. Most COTS packages are purchased in a SaaS model. This includes, support and advice from the vendor or their preferred partner. Contact them for advice, participate in user groups to get best practice from other organisations. Importantly make the vendor aware of the changes you need. Vendors need feedback from the user community to improve the product – rather than building them yourself in a bespoke way they may make their way into the base product.

HOW ALTUS CAN HELP

We have teams of Technical and Business experts who can assist in the full lifecycle of COTS package Implementations.

Our standard Industry Models are built to map the capabilities you need to fulfil your requirements. We use these model to perform gap analysis against a range of standard COTS packages to determine the correct fit for your requirements. We can assist in or run the package selection exercise right from market analysis through to full RFP.

Our Technical Teams have a wealth of experience in defining the target architecture based around popular packages we have seen implemented in the FS sector. In addition to the pure technical knowledge, we bring our Financial Services sector knowledge to play to implement in the way most suitable for your business.

We have a network of implementation experts ranging from project managers to help you run the implementation through to testers and configuration experts to make the changes you need. All of this is underpinned by our technical experts who will govern the project. We will ensure best practice Altus methods are being followed and the solution is ready to be accepted into full production support when the project ends.

Find out more:
www.altus.co.uk
+44 (0)1225 438 000
enquiries@altus.co.uk