



Takeflite White Paper on “One-Off Do It Yourself” Software Builds

So you need a new system! Before you build your own, run through your plans against the checklist:

- Can customers make bookings after hours and when all your staff are busy or away?
- Is your software solution housed in a major data-center, so your business data is safe and secure? Major data centers guarantee supply with a maximum outage of 27 seconds a month. They are very serious undertakings costing upwards of AU\$80M for the building alone and come with ISO 27001 & ASIO T4 certifications. They are found near cities like Sydney and larger. This is the new way. Don't invest in hardware sitting under your desk, which becomes obsolete in a few months and can be destroyed with all your business data by burglary or fire.
- If there is a hacking attack on your online site or some other disaster, do you have protection and can you carry on if disaster strikes? Hint ... this is not a part-time job in the online world.
- Will your information solution be OK if your IT person gets run over by a bus?
- Microsoft provides a major release every 3 years and servers need replaced every 3 years if you are not to fall behind. Things get less reliable and more expensive if you fall behind. Does your solution take this into account with the cost of upgrades budgeted for?
- Is your proposed solution network efficient so you don't get big Telecom/Telstra bills?
- If you get a new contract that's a bit different to your usual business, is your software multi-skilled enough to handle it, or flexible enough to have your new workflow added on easily?
- Can all your staff work as effectively remotely as at work?
- Can all your offices work on the same data at the same time? Hint ... this is important because less confusion means less problems and lower toll bills, not to mention staff in quiet offices helping staff in busy offices.
- How many days after you order your new system till you can use it? Hint ... if the answer is more than 30 then by the time it arrives your requirements will have changed and you'll have to start again.
- Does your software provider have centralized error checking, so they can call you up when something goes wrong and tell you how to work around the problem and when it will be fixed?
- Will your software have comprehensive audit trails so you can tell who did something silly!
- Will you be able to automate agent bookings and invoicing? Hint ... we have for many companies!
- Will your online site be optimized for speed and network efficiency (i.e. bandwidth cost)?
- Will you be able to reboot your web-servers without dropping customer bookings?
- Will your email systems be optimized to deliver etickets to the customer not their junk-mail folder?

The reasons for automating your business and going online are now well canvassed. Although it's tempting to think that your business has always done things the same way and there's no need to change, the world is changing rapidly in terms of regulation, the cost of staff and fuel and the level of competition you are facing. To meet these demands, most businesses have put in some sort of computer system, which typically have been pretty expensive and haven't in the end delivered what was expected. There is a new way! A commodity “Software as a Service” “Cloud Computing” solution is:

- Safe, by being housed in a datacenter; the sort of place where banks put their information. You never need to worry about a fire or computer failure again;
- Comprehensive, because it caters for many businesses like yours already;
- Cost effective, because you pay as you go for a slice of a “much bigger and better” pie.

The good news for airline/charter companies (with planes, helicopters, boats, buses or balloons) is that Takeflite is available now. The better news is that your local IT resource can be involved in setting the system up and producing that “must have” report. The best news is that your investment in the old system isn't wasted; if your users have become comfortable with using a computer system, then you'll find implementing Takeflite even easier.

If you are contemplating embarking on building a one-off system though, we have a word of caution. We've come across many customers who've spent AU\$100,000 up to AU\$1M on their one-off and non-online systems, before finally finding they just won't do what's needed. You might think of this as over-capitalizing on a house in the wrong area. The idea you can on-sell whatever you produce is exciting, but the chances are negligible it will be a reality. So here are some things to think about before commencing “one-off” major builds of airline/tour operator software systems.

What is a “one-off” build? Is anyone using your proposed software solution already or will it take less than a month to be in operation? If the answer to either question is no, then you are looking at a one-off build, where you will pay all the costs to begin with and into the future.



Firstly in terms of creating a new airline or tour-op system, you'll need to cater for:

1. Housing your systems in a data-center that has published outage times. 27 seconds per month is appropriate; namely the time it takes to transfer automatically from one AU\$2M router that's failing to another. These data centers are now cost effective and the nervous worry involved with running a server in a back room is just no longer justifiable. Businesses fail every day because of system crashes when the server wasn't backed up. If your proposed new software isn't network efficient and can't be housed in an online data-center, it won't stand the test of time.
2. Online multi-user ability integrating web/phone/accounts i.e. more than one person can look at and change all information at the same time from home or the office. Note that starting off with an Access DB application that worked in the past and then taking that online will not work.
3. A booking engine which handles flights with multiple aircraft and lots of legs (packages/vouchers etc).
4. Credit Card payment gateway integration
5. Accounting integration for invoicing and GL reporting e.g. MYOB. Multi currencies and packages per route.
6. An ability to handle weather/engineering problems in terms of the subsequent customer processes (prioritize customers who have been rescheduled before)
7. Inquiry & quoting facilities (inquires by phone or internet) including ability to quote on multiple aircraft
8. Agent & public online portals catering for all the different browsers i.e. MAC Safari, Firefox ...
9. Gift Vouchers (including online purchase)
10. Management reports (about 100 core reports required)
11. An operations manager view (activity per aircraft, with drag and drop for charters) and booking center view (free seats per package)
12. Integrated flight logs (fuel/oil uplift) and engineering maintenance (inspections/inventory/work packs)
13. Loyalty programs and customer mail-outs
14. Crew scheduling
15. Flexibility to take bookings without flights and flights without bookings and then mix and match e.g. we'll put these 10 people on aircraft A and these 10 on aircraft B and call that a flight, but this other flight isn't going anymore so let's delete the flight and leave these bookings against the aircraft to sort out when we know what's going on. Flexibility is the key with dynamic airline operations.
16. Ad-hoc functions like rescheduling (per booking, but also catering for splitting group bookings, and also online), refunds, user security profiles (i.e. temps can't look at company financials), site settings so can change terms & conditions and logos and e-tickets at the flick of a switch, show last bookings and display daily stats, wait-listing, customer feedback handling, audit trails and notes ...

To put a dollar value on providing the above, at Takeflite we've invested so far more than AU\$4.5M over 9 years, of which about 2/3 is typically relevant to any single given operator.

As well as project costs with creating a new software system, there are people management pitfalls. A typical one-off software development project requires a team of at least 5 to cover the core competencies, namely:

1. Project Management
2. Business Analysis
3. Systems Administration i.e. hardware and maintenance and capacity planning
4. Database Administrator
5. Windows interface development
6. Browser interface development
7. Report development
8. Interface development
9. Testing (Functional, UAT, Stress)
10. Security specialist
11. Network specialist
12. Technical writing
13. Trainer
14. Help Desk & support
15. Change Management
16. Various client liaison people like the Business Sponsor and Call Centre and Accounts staff.

So delivering complex requirements requires a range of complex and expensive people, who need to be managed carefully to ensure go-forward. After 20 years in the industry, I've seen about a 10% success rate for development projects like the above (corresponds to Gartner Group reported results). A project like this is not really achievable with 1-2 IT people and the alternative of using a software development consultancy without specific airline experience amounts to paying people to learn on the job and re-invent the wheel. Our "Software as a Service" approach is targeted towards making a successful solution immediately available to regional airlines at an affordable price.

Overall:

1. Customers often like the thought of owning software, but there are disadvantages:
 - a. New systems can be like new cars; they are worth half as much once they're off the showroom floor. Get started with something you can get advantage from immediately, then pay as you go leaving the Takeflite experts to budget for ongoing maintenance.
 - b. Up-front costs & risk are high for customers contemplating a new software system and although initial progress can seem promising, the devil is in the detail and the last 20% takes 80% of the time and effort and money;
 - c. Ongoing costs can't be shared, which forces economies in the wrong places, particularly further down the track when everyone has had a enough of the hard work and can't be bothered ... "there are real customer issues to attend to!" becomes the catch cry;
 - d. Aside from the requirement for ongoing system development (approx. 1 release per month for a successful system), software application creators can expect to need to rebuild the application completely every 3 years to keep current. Failure to do so means your system can't deliver competitive features and become expensive to run. It takes specialist staff to manage this continuous improvement successfully, both in terms of picking the right IT strategy and in terms of managing changes without interrupting service or losing data.
 - e. Businesses that build their own system become in essence an IT shop as well as an airline, but without the depth of necessary specialist staff. When key staff leave and/or requirements change, the application becomes a business risk and budgets are normally too depleted to fix the issues or start again.
 - f. In our experience, if the business/booking application you build is based upon desktop products like Excel or Access or the shareware product MySQL, it will ultimately be unsuccessful. These products are great, but no good for online high transaction booking systems with a high disk write-to-read ratio. Conversely, there are licensing traps with using an appropriate product like Microsoft SQL-Server, where multi-processor systems become prohibitively expensive, particularly where companies share a server.
 - g. All hardware has to be replaced every 3 years when it comes off manufacturer support, because it's too expensive to do otherwise i.e. a single replacement fan for a more than 3 year old HP server costs upwards of AU\$370 and takes 3 weeks to arrive. We treat hardware as an operational expense not an asset. With Takeflite you get the server hardware in effect thrown in for free ... because by the time we spread the costs over 30+ clients, we can afford to. When you build yourself or host yourself, it's an expensive exercise.
 - h. Although the monthly service fee for a leased solution is sometimes called "clipping the ticket", software systems are not a buy and forget exercise. The bigger you get and the more you do, the more your system will cost and it's sensible to acknowledge that at the outset. Systems need continual maintenance to add value. The monthly cost of Takeflite is not that different to the "Help Desk" fee from legacy per-user priced software providers i.e. the only difference is you miss out on the huge up front purchase costs.
 - i. Your system requirements will change as your circumstances change e.g. more customers means you'll need more resources and probably different business work-flows. By starting with a flexible and scalable software service, you can take all this in your stride without having to start again.
2. Core to the cost efficiency of the Takeflite solution (for all concerned) is the need to provide updates and fixes efficiently and not get bogged down with system failures. Most application providers ultimately fail, because most of their organization after a few years is focused on keeping clients with old versions limping along. By providing "Software as a Service" which is always up to date on out-sourced virtual servers housed in Tier 1 Data Centers, we have created an environment where we can spend the bulk of our time on returning value to our clients, rather than treading water. We have one development project, configurable via site settings by the customer, which we rollout to all customers with the click of a button.
3. Our rollout strategy deploys from the development platform to the UAT platform to pilot users, to the general community. Our infrastructure provides centralized logging, which means we see and can fix problems even if we don't get called. Bearing this in mind and with 30+ clients, the chances of any one client seeing a bug is pretty small.
4. In terms of track record, we haven't lost data in 9 years of hosted business across all our customers, which means if you put data in, you'll get it back out (and we'll often have audit trails to prove it).
5. By joining the Takeflite community, you get access to a "community forum" where you can ask questions of other operators in terms of how they've approached a business requirement using Takeflite. This "best practices advice" is important. It's not always about the software ... it's often about how you use it.
6. The cost of recovery from a hardware failure is too high and the issues too complex to be addressed by non-dedicated personnel e.g. if a power failure drops a busy SQL-Server database because the batteries in the UPS are too old and don't work while the generators are turning on, then the information is probably irrecoverable. If the backup/logging/DR strategy has failed, the situation is dire. With Takeflite, these risk factors are handled by dedicated professionals with 20 full-time years in the industry. We know as much about customer facing online IT systems, as our clients know about Aircraft, Helicopters, Buses & Boats.

Comparing Apples with Apples

When an airline compares building a system versus buying one off the shelf, the comparison is often focused upon existing requirements as they are understood today. Takeflite has learned from (and caters for) the best practices of 30+ operators. Some recent examples of benefits provided by the Takeflite which are often overlooked:

- (a) One of our partner airline's customers booked 3 passengers from Picton to Nelson via Wellington for NZ\$513. At face value, that made no sense because it's a 1 hour drive to get from Picton to Nelson, but a 2 hour flight with the stopover. The point is that the customer got access to a service they desired and which nobody could have foreseen in advance. The Takeflite "web-services fare-engine" makes all combinations of multi-sector routes available using a technique called recursion ... and for one airline that earned an extra NZ\$513 turnover from just one booking, which otherwise would have been lost.
- (b) The same airline was closing off under-utilized flights a day in advance, but discovered that 20% of bookings were made within a day of departure and so have changed policy to reflect this. This information combined with a report that identifies booking requests which weren't fulfilled and other operational reports enables effective yield management.
- (c) We've come across a number of airlines that have a staff member dedicated to tracking the number of VFR/IFR flights for CASA/CAA licensing and the number of landings for monthly airport landing fees. These laborious tasks can be completely automated providing more accuracy and freeing up the staff member for more productive work. It's often not till you use Takeflite that Airlines realize what business process improvements are possible. Be aware: writing a custom system for an airline puts you at risk of locking in your existing inefficiencies making you uncompetitive in an evolving market.

Not convinced yet!?

One other reason airlines look at developing their own solution is the perception that their unique requirements can't be met by an "off the shelf" software service. This is a reasonable argument often born out of experience however Takeflite has already catered successfully for 30 "unique" businesses. Takeflite's point of difference is its ease-of use and deployment combined with a consulting development capacity to deliver on additional & sometimes unique requirements. Our consultancy division will listen to what's needed and where we all agree it makes sense, deliver what's missing. Some thoughts if you need something a bit out of the ordinary:

1. Care is necessary to avoid developing a feature which will cost \$X to deliver, but won't return \$X of savings across your fleet within a year.
2. When developing your requirements list, try to split it into must have's and nice to have's; successful projects use an infrastructure with long term capacity to meet your needs, but concentrate on delivering on the important things quickly, to get user buy in.
3. A common mistake with implementations is getting stuck on a particular feature, when in practice it will not be possible to get staff to enter the necessary data. Implementations can get de-railed spending more and more time on features which for "human" reasons will never ever work.
4. Developing features which can be used by many airlines is something we like to do; it maximizes the returns for everyone from our efforts. Where we get a one-off requirement which is unlikely to be of value to another airline in the community, try to think of the simplest way you could achieve the outcome you need. This will keep costs down and ultimately "keeping things simple" makes them usable.
5. Successful software solutions are never "finished". Tomorrow always brings something new and your overall information technology strategy needs to manage this process.

In Summary, as one of our customers said: "***airlines don't make tires and we don't develop software either***"

More Reading:

http://h30458.www3.hp.com/nz/en/smb/736439.html?jumpid=em_di_450004_NZ_GB_1_003_hpc_gb_749347_acr_oss-bg&dimid=1003437624&dcid=null&mrm=1-4BVUP

<http://talkback.zdnet.com/5208-10532-0.html?forumID=1&threadID=62900&messageID=1162837>

http://www.zdnetasia.com/news/software/0_39044164_39293868_00.htm

<http://www.stuff.co.nz/business/industries/2978340/Plan-for-a-cloudy-future>