

## System Integration Introduction

**Wikipedia** defines System Integration as in “*engineering as the process of bringing together the component sub-system into one system (an aggregation of subsystems cooperating so that the system is able to deliver the overarching functionality) and ensuring that the subsystems function together as a system, and in information technology as the process of linking together different computing systems and software applications physically or functionally, to act as a coordinated whole. The system integrator integrates discrete systems utilizing a variety of techniques such as computer networking, enterprise application integration, business process management or manual programming. System integration involves integrating existing often disparate systems and is also about adding value to the system, capabilities that are possible because of interactions between subsystems.*” [Ref 1]

Commonly recognized software development methods include vertical integration (creating functional entity silos), star integration (interconnected subsystems), horizontal integration (typically an enterprise service bus type approach), all normally following a common data format methodology.

System Integration involves more than just software development, it covers all aspects of business requirements to ensure that there is technical, functional and capability integrity within the project and associated deliverables, products and services. It involves automation and control technologies, software and test/evaluation systems, networking and enterprise systems, all linked to project management and business best practices, stakeholder engagement and alignment of goals to meet stakeholder expectations. System Integrators must have an appropriate level of understanding of the relevant components, subsystems, interfaces, systems, processes, workflows and support/training systems to make sure that they all work together in a predictable, reliable, efficient and economical manner throughout the whole lifecycle. [Ref 2]

Ten common issues that can quickly emerge and cause significant problems when undertaking system integration type activities includes:

- Leadership (or lack of it) is the most influential factor because it covers aspects such as the political arena, sponsorship and support, communication and reporting, ease of problem resolution, gaining additional resources and not losing the resources you currently have assigned to your programme, and acting as a buffer between demanding stakeholders;
- Not having clear business direction, ranging from not understanding the business issues driving the project, through to not having a good handle on the requirements set (especially the non-functional requirements);
- In appropriate resources to undertake the system integration activities, ranging from insufficient time, funding or skills to do the work through to not having a reasonable amount of time and expertise to undertake the design and build activities;
- Failure to mitigate and manage risks in terms of safety, security and environmental engineering, COTS products integration and the testing and commissioning aspects;
- The level of customizing or be-spoking software and systems (MOTS) presents a difficult issue because the more a product or service is customized, the higher the technical overhead to develop, manage and support becomes, with associated through-life costs;
- Understanding the impact of continuing to use or integrate with legacy systems, especially those that your organization may have invested a significant amount of capital and resources to get running and is now reluctant to move away from;
- Being aware of data migration issues and not bringing old, redundant or poor data across into new systems is always a big concern when introducing new capabilities that replace existing systems;
- Changes in the external environment especially in the fields of the introduction of new technology, the IPR or proprietary nature of open standards or formats, capabilities being withdrawn, removed or overtaken by market forces, or aggressive behaviours and business tactics;
- Obsolescence Management can be a significant showstopper if not managed proactively. The loss of a key component within your solution can result in significant delays, cost overruns and in the worst case whole re-design programmes;
- The cultural effect of change on the people within your organisation. If not handled correctly can be the single most destructive or if managed correctly is the biggest motivator to success.

The complexity and sheer number of systems being developed today mean that change becomes a major issue if not factored in at the start of the process. Methods of combating the impact of change associated with system integration include:

- Having competent, skilled and experienced people and teams to undertake the work is fundamental, and making sure you listen to their concerns and issues, because if you don't, you are potentially pushing major problems down the road;
- Having an Enterprise Architecture Methodology and associated evergreen strategy to easily and quickly swap out any of the architectural building blocks without affecting the other aspects of the enterprise, or making it difficult to re-certify, assure and accredit for deployment/use;
- Having a Grand Design is great to aim towards, but you need roadmaps and transformation programmes to help you get there via logical and coherent stepping stones, interim architectures/states, that bring together strategies around infrastructures, data, security, networking, messaging and applications for both on premise and off premise solutions and services;
- Reducing or ideally eliminating the dependency on any individual manufacturer or product range, and being able to migrate to other platforms and systems should the need to arise (i.e. customers have been finding it difficult to migrate away from their current cloud service provider, so they should always seek to maintain control of the data at all times);
- Maintaining and providing a common look and feel for the user community, to reduce training on the new systems, make the systems more intuitive to reduce the training burden and hiding a lot of the complexities behind well thought out HMI's;
- It is well understood that the best integrated solutions all link to seamless processes and workflows, because that is how outcomes are generated, not by selecting specific hardware items or applications. Systems analysis is needed to determine what will work for your organization, the solution will not be found in a book, it will need to be created based on your unique parameters and needs, so experience is everything;
- Have a backup plan and proven business continuity process working before attempting upgrades, because invariably things will go wrong and you will need to fallback to the previous baseline or have developers available to fix the issues (APIs, IERs, Point to

Point links, Links to legacy, proprietary or supplier systems, links to ERP/Billing and Management Systems are particular vulnerable as they are often not thoroughly tested);

- Takeover and mergers never go as smoothly as anticipated because detailed designs and documentation is never fully created or maintained, a lot of the other systems are dependent on heroic efforts to keep them going and when those people leave the systems start to fall over or the knowledge just walked out of the door over a number of years and nobody noticed it leaving;
- Some legacy systems should have been shutdown years ago but have been allowed to continue. They are often inefficient, cost a lot to maintain, data is locked into any number of sub-systems and they are becoming more complex by the day due to further change requests and workarounds. There is a level of mess and chaos that having been achieved is impossible to reverse or take any further. If you have reached this level of anarchy on your burning platform, with no planned new platform to jump to, then you are in a real big problem space;
- Baseline alignment (especially if linked to separate customer, end user or supplier systems) and configuration management can be particularly challenging, especially when you start introducing the backward and forward compatibility issue between baselines, or want to have an eco-system of mutually exclusive different competing supplier components (i.e. IBM, HP, Dell & Cisco Servers) because although they may say they adhere to the same standards and formats, there are often small subtle differences in implementation that can make live very difficult for integrators.

When you start to venture into the domain of the bespoke, unknown, never been done before or try to go against the market forces, you are immediately on the back foot. The analogue I like to make is that it is pretty easy to research the best components of cars, as listed below:

- Best Engine (2017) – BMW 3.0L Straight-six (Ref 4);
- Safety Car Body (2017) – Audi Q7 (Ref 5);
- Best Tyres (2017) – Continental ContiPremiumContact 5 (Ref 6)
- Best Suspension System (2017) – Jaguar F-Type (Ref 7)
- Best Interior Comfort Factor (2017) – Mercedes S-Class (Ref 8)

So with this quick research, why don't we see lots of custom made vehicles equipped with Audi Body, BMW Engines, Jaguar Suspensions, Mercedes Interior and Continental Tires on the road today? The reason is that it would be very expensive to create, it would probably be horrible to drive and have awful performance and behaviours, plus it would not be very reliable or easy to maintain. The reason is the car makers put a lot of effort into developing cars and along the way there are a lot of decisions and trade-offs to be made, for the sake of road worthiness, economy, performance, balance and handling. Modified cars have a habit of improving one area, but that improvement comes at the cost of another or puts additional strain on other components that are not rated appropriately. Car manufacturers have learnt lessons over a number of years, and people new to the market can easily make mistakes or put together improper installations. Finding replacements or support for custom made components is nearly impossible. There is a natural affinity for certain components to work well with other components, and this has been learned the hard way.

Choosing the right set or suite of products and services for your organization is always going to be a challenge, without it becoming too complex, too costly or doesn't satisfy the business need. Avoid the creation or perpetuation of monolithic systems, because integration becomes a real challenge. Different people and business units in the organization see data differently and have different/competing requirements, where the focus is on modelling the data and information flows, whilst implementing a minimal canonical data model. Integrator must meet the needs of the business today, but be aware they need to keep an eye on the future (supportability, performance, scalability, reliability, maintenance and upgradability) and will be expected to re-run their testing routines a number of times so must give some thought to automation techniques. Be wary of non-technical management that is focused on delivering immediate business centric outputs, at the expense of good development pipelines/methodology, use of best architectural principles and ease of upgradeability/problem identification. There is a careful balance to be achieved, and this is a very difficult line to follow. Don't under-estimate the factors of regulatory compliance issues, data privacy, the introduction of more and more IT related legislation, recording of design decisions and the audit functions, should anything go wrong in the future and the case ends up in a court of law. What was known, when will be a critical issue that could determine the outcome of the case and lead to significant losses (including reputation) for the liable plaintiff.

A glimpse into the future would seem to indicate that organisations will be driving for more and more integration within and outside their boundaries, however there are still some functions that are having difficulty being integrated (e.g. HR, Compensation & Talent Management Systems, Learning & Knowledge Management Systems, Big Data Systems and Collaborative Systems especially with external partnerships), generally because the market is still immature in these areas or there is no perceived large revenues to be generated from these areas at the moment. Little is being done or delivered in terms of building/facilities management and energy management systems are largely stand-alone systems, except when they feed excess energy back into the grid. Mergers and acquisitions continually pose large system integration problems due to the multitude of different systems, processes and cultures. It takes the right amount of leadership, planning and preparedness, creativity, tapping into the right resources and on-hand team of experts to resolve local issues that arise to stop them becoming major showstoppers, in the grand scheme of things.

Making system integration easier can be achieved by having a clear understanding of the requirements (including business needs and expectations), the existing infrastructure ('As Is' State) the strategy that helps direct the aim points ('To Be' States) and personalities involved in the leadership, decision making, negotiations, disputes and change management activities (can you work with them?). Having an established process, workflows and prototyping regimes for research/discovery, design, development, testing implementation, rollout and support are essential. You can never do enough testing (even with beta products, but make sure it is focused, prioritized, linked back to requirements, use cases and user journeys. It is always less risky to stick to the Tier 1 vendors/suppliers as they often have more experience in depth and have greater compatibility with heterogeneous environments. As systems get more complex, it can be more difficult to determine who is responsible, accountable, owns, coordinates, manages or oversees not only the products or services, but the data and reports. Finally, don't forget the often overlooked elements of system integration with regards to configuration/set-up procedures, establishing the business rules that help drive automated events/activities, the training of the people who will be using the system, and having a robust helpdesk, issue/problem/resolution support organization to help keep things running when (not if) issues are discovered. People make processes, workflows and products work at the end of the day, and whatever infrastructure is put in place, it must be

people-centric, intuitive, easy to use, fits together with other processes/facilities and easy to spot/rectify mistakes, in order to be truly effective and deliver business winning advantage.

### **What does it all mean?**

With most of the modern world economies driven by computer systems, the way they are integrated into this networked world is critical to what we do everyday. However, the way organisations approach the creation of enterprise IT solutions is far from being ideal, with IT being cited as a dead weight slowing growth and not working as an accelerator to help drive growth. The drive towards the next new product and 'cool' tools (shiny new toys continues to be a big problem), obsession with purist solutions, hides the fact that the hardest part of enterprise architecture is system integration, because it remains the most complex undertaking, with an infinite number of possible combinations of solutions, and continues to be more of an art form rather than a pure science. Established vendors are trying to maintain the status quo and tie you into proprietary solutions, whilst the new kids on the block try to undermine this situation. There is no 'ideal solution' that will work for everyone, and optimal solutions tend to be compromised trade-offs balancing the old with the new, as few organisations can afford to start from a greenfield site again. Monolithic object orientated systems are totally incompatible with the mobile dynamic distributed models and systems that are coming onto the market, and trying to merge, misuse both systems together is a disaster waiting to happen. You have to choose your horse (approach) and accept the limitations that it can't fly or go underwater (can't do everything).

There is no worldwide standardization, because this would create a monopoly, so there will continue to be a variety of systems and solutions, you just need to pick the one that works for you and hope for the best. The solution vendors are focused on the richest part of the market, not on simplifying the most complex technologies to make it easier for their customers to go anywhere they want. IPR is a big issue, and remember that even open source products have a natural lifecycle and don't go on forever. To get any true value out of your IT system, you probably need the very best caliber of people and there are only so many out there (smart people can still be perceived as being dangerous in IT). If you are a medium sized enterprise, or a large public body with pay ceiling for the people you bring in or trying to retain, you are already behind the curve. If your leadership still make decisions based on sales pitches, brochures

and vendor claims, then it is time to look for a new job, because the organization will not be around for very much longer.

As you consider System Integration consider trying to adopt the following points:

- It is you people that make it all happen not automated tools and techniques, so make sure you look after them, train them well and support best practices wherever you can. Challenge the competency and skills of your people;
- You can't go wrong by keeping your data structures simple, but recognise that the devil is often found in the detail and you can't simplify the problem space or make wild assumptions and expect them to be true; the real world is not like that – simple solutions are hard to define and obtain;
- Keep your design documentation up to date and model based (i.e. to make it easier to manage and control);
- Always have a number of Plan B's because even your best ideas and architectures may not all work, so you make have to design and implement other approaches;
- There are times when you shouldn't at this time do system integration at an electronic level, where the technology or products are just not mature enough, you don't want to be a first adopter or it is just too costly. Sometimes it is best to just wait, do the process manually until you can work out what you want, how to do it and what technology to choose;
- Too many times, people are fixated on the first series of solutions that appear. Listen to other sides of the argument, other view points or other team members to get glimpses of the full picture. Many times, people surround themselves with 'yes' people that is not a good thing;
- The whole organisation, product and programme needs to get over the line for true success to be achieved. It is not like a typical competitive event where there is a single winner. In system integration land there is no point having a world class billing system if your product quality is poor, you need to get them all over the line;
- As things get complex, there is real value in looking for patterns, similar conceptual functions/capabilities, modularity, bounded entities with clear, concise and controlled boundaries;
- There is no one solution that will work for all cases. Solutions need to be tailored, crafted and evolved to meet your current and future

needs (both known and those that emerge along the way), so it is useful to build in some level of flexibility, resilience and robustness into your approaches so you are some way prepared for the changes that will occur just around the corner;

- And remember you can have anything, you just can't have everything. There is a real art and skill to obtaining the right balance, saying no to some of the customer requirements to allow real value to be obtained (not just a compromised tick-box where everything has been achieved, but a real working system is not achieved).

## References

1. [https://en.wikipedia.org/wiki/System\\_integration](https://en.wikipedia.org/wiki/System_integration)
2. <http://www.controleng.com/single-article/dealing-with-system-integration-issues/7342276ac9cc8e47a2a77b3224ac5902.html>
3. <https://www.cwjobs.co.uk/careers-advice/it-glossary/10-things-to-know-about-system-integration>
4. [https://en.wikipedia.org/wiki/Ward%27s\\_10\\_Best\\_Engines](https://en.wikipedia.org/wiki/Ward%27s_10_Best_Engines)
5. <http://money.cnn.com/2017/05/25/autos/safest-and-deadliest-cars-on-the-road/index.html>
6. <http://www.autoexpress.co.uk/accessories-tyres/100247/continental-contipremiumcontact-5-review>
7. <https://www.autobytel.com/car-buying-guides/features/10-cars-with-the-best-suspension-131378/>
8. <https://www.carwow.co.uk/best/the-most-and-least-comfortable-cars-you-can-buy>
9. <http://www.whishworks.com/blog/mulesoft/5-challenges-with-systems-integration>
10. <http://searchitchannel.techtarget.com/feature/Best-practices-for-combating-integration-problems>
11. <https://www.infoq.com/news/2010/06/RESTForward>

Article written by Innov8or Solutions Ltd. Copyright © 2017. Innov8or Solutions Ltd. All Rights Reserved.