

## **Best Practice, Standards, Guidance and Lessons Learned Applicable to the Implementation of the New Switching Arrangements**

Best Practice, Standards and Guidance. A due diligence review was undertaken of best practice, standards and guidance applicable to the implementation of the new switching arrangements; focused in particular on the Testing, System Integration and Post-Implementation strategies. These sources are often powerful and, although they don't in themselves guarantee success, they can greatly reduce programme risk and support successful outcomes as they are generally based on a large underlying source of lessons learned across multiple sectors and programmes developed over a number of years.

First, it is necessary to characterise the Switching programme implementation to understand which best practice, standards and guidance would be most relevant. The new switching arrangements will involve the design, build, integration, test and transition into live operations of a new information-based solution supporting faster, more reliable switching of suppliers across the gas and electricity markets. As such, it has the following assumed characteristics and features:

- The current switching arrangements in place across the gas and electricity markets use a variety of Information based systems and services within and across market participants to store and exchange data in defined formats to support switching business processes.
- The new switching arrangements aim to harmonise switching across gas and electricity, across information, systems and processes where possible, as well as improve switching reliability and speed. This represents a 'complex change programme' that will impact multiple organisations, people, processes, information and technology.
- The new arrangements are likely to be realised through a combination of changes to existing systems and processes as well as the implementation of new solution components, such as the Central Registration System (CRS). The solution will involve a significant change in existing cross-party and cross-system interfaces as well as some new interfaces.
- Continuity of switching capability throughout the transition to the new arrangements will be key so as not to undermine the effective operation of the energy retail market or disadvantage any consumers or parts of the Industry.
- Delivery of the new arrangements will be 'distributed' across multiple dispersed parties and their design teams with no single delivery vehicle uniting these (some parties will respond to license obligations and codes and some will be contracted (e.g. the CRS provider). Appropriate Incentives and controls within governance and assurance will be key to ensure a cohesive, coherent approach to achieving common objectives and outcomes.
- The delivery of the new switching arrangements will take place in a context of wider ongoing changes; e.g. the Smart Metering Implementation Programme (SMIP), Nexus, reforms responding to the recent Competition and Market Authority (CMA) recommendations, etc. This places significant risks and dependencies on the Switching programme.

The new Switching arrangements can therefore be considered as a complex, multi-party change programme involving the delivery of an information-based solution (system or service) comprising people, processes, technology and data that will build on, adapt, harmonise, and improve existing solutions/systems providing switching in the gas and electricity retail markets. Delivery of the solution represents a complex 'system realisation' problem with the 'end to end' system for switching depending on the successful (synergistic) interaction and integration of multiple Industry systems and the CRS. Given this, the following areas of best practice, standards and guidance have been identified as those most applicable to the implementation of this programme:

**Systems Engineering.** An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: Operations; Performance; Test; Manufacturing; Cost & Schedule; Training & Support; Disposal. Systems engineering integrates all disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE 2012)

**Software Engineering.** Software engineering is the application of engineering to the design, development, implementation, testing and maintenance of software in a systematic method. In many ways, its structured methods gave rise to Systems Engineering and indeed over recent years the applicable ISO standards for these two have become harmonised; particularly around processes and life cycles.

**Information Technology (IT) Service Management.** This covers the entirety of activities that are performed by an organisation to plan, design, deliver, operate and control information technology (IT) services offered to customers. It is thus concerned with the implementation of IT services that meet customers' needs, and it is typically performed by the IT service provider through an appropriate mix of people, process and information technology.

**Project and Programme Management.** Project management tends to be concerned with the delivery of one or more 'outputs' within defined time, cost and quality parameters whereas Programme Management is usually concerned with the delivery of one or more 'outcomes' achieved through the integration and beneficial use of a range of integrated project outputs via their transition into normal business operations.

The tables below summarise some current sources of best practice, guidance and standards relevant to these disciplines and draws out those elements particularly relevant to Testing, System Integration and Post-Implementation. These are not exhaustive; there are others not listed, e.g. e-TOM and COBIT for IT Service Management, but the ones listed below tend to be those most favoured in the public sector.

		Domain/Discipline				
		Systems Engineering	Software Engineering	IT Service Management	Project Management	Programme Management
Best Practice	Lead professional bodies	INCOSE	BCS (Chartered Institute for IT) Axelos (formerly OGC)		APM & PMI, Axelos, DSDM (for Agile DSDM)	
	Best Practice Guidance / Frameworks	INCOSE Body of Knowledge		ITIL (+ SIAM)	Prince 2 & APM BoK	MSP APM Body of Knowledge
	Aspects applicable to Testing and System Integration	Integration, Verification & Validation		ITIL Service Transition: Validation & Testing	Solution and Scope Mgmt (APM) Quality (Prince 2)	Assurance Quality
	Aspects applicable to post Implementation	Operation & Maintenance		ITIL Service Transition: Release and Deployment	Closing a project (Prince 2)	Transition & Benefits Realisation
Standards	Higher level/ umbrella standards	ISO15288, ISO9000 and ISO9001. IEEE730		ISO20000	None?	Management of Portfolios
	Life Cycle Delivery Models	ISO12207 (System & Software Lifecycle Processes) – 'V model'		Service Life Cycle	Waterfall, Agile, V model, Prince 2 Processes	Transformational Flow
	Testing and Integration Specific Standards	IEEE1012 & ISO15026 (pt2 – Assurance Case)	ISO29119, IEEE829, IEEE1008, IEEE1028	None	None	None
	Post Implementation Specific Standards	None	None	None	None	None

Source	Scope and Summary
INCOSE BoK	The Guide to the Systems Engineering Body of Knowledge (SEBoK) provides a widely accepted, community-based, regularly updated baseline of <a href="#">systems engineering</a> (SE) knowledge. It is maintained by: the International Council on Systems Engineering (INCOSE), the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS), and the Systems Engineering Research Center (SERC)
APM BoK	The Association for Project Management (APM) Body of Knowledge (6 <sup>th</sup> edition) contains the APM's assembled best practice and guidance on Portfolio, Programme and Project Management.
ITIL (+SIAM)	This is a best practice guide issued and maintained by Axelos (formerly the Office of Government Commerce) which comes in 5 parts: Service Strategy; Service Design; Service Transition; Service Operation; Continuous Service Improvement. The Service Transition part covers the majority of topics relevant to Delivery of Switching (i.e. design, build, test, deployment and release). Service Integration and Management (SIAM) is effectively an application of ITIL that considers how the integrate and manage services that are sourced from a large number of individual suppliers. The main goal of SIAM is to coordinate internal and external suppliers and their services in a cost-effective way to achieve the end-to-end service levels needed to support the goals of the business
Prince 2	PRINCE2® (Projects IN Controlled Environments) is a structured project management method maintained by Axelos (formerly OGC) based on experience drawn from thousands of projects and from the contributions of countless project sponsors, Project Managers, project teams, academics, trainers and consultants. PRINCE2 isolates the management aspects of project work from the specialist contributions, such as design and construction. The specialist aspects of the project can be easily integrated with the PRINCE2 method and, when used alongside PRINCE2, provide a secure overall framework for the project work.
MSP	Managing Successful Programmes (2011) is another Axelos (formerly OGC) maintained best practice publication of Programme Management focusing on complex change programme delivery. The parts mainly applicable are Delivering the Capability and Realising the Benefits (through effective transition of the capability into business as usual operations)
Agile	Agile is a collective term embracing a range of methodologies that originated in Software Engineering. The best known of these include DSDM, Scrum and Extreme Programming. More latterly, a set of Agile Principles have been agreed in the form of an 'Agile Manifesto'....; albeit still largely being applied to software delivery. The latest government study into public IT programmes (Institute for Government, <i>System Error – Fixing the Flaws in Government IT</i> , 2012) advocates applying Agile principals more widely to any IT based programme.
ISO15288 (2015)	System Lifecycle processes for System and Software Engineering. Sections: 6.4.7 (Implementation); 6.4.8 (Integration); 6.4.9 (Verification); 6.4.10 (Transition); 6.4.11 (Validation); 6.4.12 (Operation); 6.4.13 (Maintenance). Are the most applicable parts
ISO12207 (2008)	Systems and Software Engineering – Software life cycle processes. Sections: 7.1.6 (Software Integration Process); 7.1.7 (Software Qualification Testing Process); 7.2.3 (Software Quality Assurance Process); 7.2.4 (Software Verification Process); 7.2.5 (Software Validation Process) are the applicable parts
ISO20000 (2012)	IT Service Management is a service management system (SMS) standard. It specifies requirements for the service provider to plan, establish, implement, operate, monitor, review, maintain and improve an SMS. The requirements include the design, transition, delivery and improvement of services to fulfil agreed service requirements.
ISO9001/9000	Quality Management System requirements and definitions – high level
IEEE730 (2014)	Requirements for initiating, planning, controlling, and executing the Software Quality Assurance processes of a software development or maintenance project are established in this standard. This standard is harmonized with the software life cycle process of ISO/IEC/IEEE 12207:2008 and the information content requirements of ISO/IEC/IEEE 15289:2011.
IEEE1012	Verification and validation (V&V) processes are used to determine whether the development products of a given activity conform to the requirements of that activity and whether the product satisfies its intended use and user needs. V&V life cycle process requirements are specified for different integrity levels. The scope of V&V processes encompasses systems, software, and hardware, and it includes their interfaces. This standard applies to systems, software, and hardware being developed, maintained, or reused [legacy, commercial off-the-shelf (COTS), non-developmental items]. The term software also includes firmware and microcode, and each of the terms system, software, and hardware includes documentation. V&V processes include the analysis, evaluation, review, inspection, assessment, and testing of products.
ISO15026 Part 2	ISO/IEC 15026-2:2011 specifies minimum requirements for the structure and contents of an assurance case to improve the consistency and comparability of assurance cases and to facilitate stakeholder communications, engineering decisions, and other uses of assurance cases. An assurance case includes a top-level claim for a property of a system or product (or set of claims), systematic argumentation regarding this claim, and the evidence and explicit assumptions that underlie this argumentation. Arguing through multiple levels of subordinate claims, this structured argumentation connects the top-level claim to the evidence and assumptions.
ISO29119	ISO/IEC/IEEE 29119 Software Testing is an internationally agreed set of standards for software testing that can be used within any software development life cycle or organisation. By implementing these standards, you will be adopting the only internationally-recognised and agreed standards for software testing, which will provide your organisation with a high-quality approach to testing that can be communicated throughout the world. There are currently five standards: <a href="#">ISO/IEC 29119-1: Concepts &amp; Definitions (published September 2013)</a> <a href="#">ISO/IEC 29119-2: Test Processes (published September 2013)</a> <a href="#">ISO/IEC 29119-3: Test Documentation (published September 2013)</a> <a href="#">ISO/IEC 29119-4: Test Techniques (at DIS stage, anticipating publication in late 2014)</a> <a href="#">ISO/IEC 29119-5: Keyword Driven Testing (at CD stage, anticipating publication in 2015)</a>
IEEE829	Test processes determine whether the development products of a given activity conform to the requirements of that activity and whether the system and/or software satisfies its intended use and user needs. Testing process tasks are specified for different integrity levels. These process tasks determine the appropriate breadth and depth of test documentation. The documentation elements for each type of test documentation can then be selected. The scope of testing encompasses software-based systems, computer software, hardware, and their interfaces. This standard applies to software-based systems being developed, maintained, or reused (legacy, commercial off-the-shelf, Non-Developmental Items). The term "software" also includes firmware, microcode, and documentation. Test processes can include inspection, analysis, demonstration, verification, and validation of software and software-based system products.

Examination of these various sources reveals that, in terms of Testing, Software Engineering as a discipline seems to provide the most detailed and comprehensive guidance and standards underpinning the best practice. IT Service Management (ITIL) provides useful frameworks for

testing in the context of service 'transition' (i.e. implementation in the language of the switching programme) and illustrates well how Testing fits with related areas such as change and configuration management, design and build, integration and so on. These areas have therefore primarily been used to define best practice for testing of the switching arrangements.

In terms of Post-Implementation, the various sources above contain little in terms of detailed guidance and standards in this area, although IT Service Management (ITIL) has the most comprehensive description of post-implementation; as part of what it terms 'Release and Deployment' the ITIL framework covers arrangements for 'Early Life Support' which aligns closely with post-implementation as defined for Switching. This source of best practice has therefore mainly been used to define the post-implementation strategy for Switching

#### Comparison of best practice frameworks and observations in terms of wider Delivery

Looking wider than just testing and post-implementation, these various sources of best practice, guidance and standards bring different perspectives to consideration of 'what good looks like' for the delivery of a complex, information system-based change programme such as the new Switching arrangements. These best practice frameworks naturally overlap in many areas, but their comparison and contrasting with the current Delivery Strategy work on Switching provides some useful additional insight that is discussed below.

The Project and Programme Management frameworks (MSP and Prince 2) come from the perspective of planning, organising and controlling the delivery of the project activities to achieve the required outputs and outcomes to time, cost and quality including associated change management aspects to realise the benefits. These frameworks are less concerned with 'what' the outputs are and 'how' the outputs are designed, developed, built, tested, etc. They assume a generic process and project life cycle that could be equally applicable to the delivery of IT projects, construction projects, or indeed any type of project that has a start, middle and end.

The Software and Systems Engineering frameworks (ISO15288) and the IT Service Management Frameworks (ITIL) are more concerned with realisation of the actual products or services being delivered and how these are taken through a product or service development life cycle. As such, they are more specific to the type of product and service and how these products and services are designed, delivered and maintained through life.

However, all these frameworks include elements of each other but provide greater depth in certain areas depending on their perspective. An exercise has been undertaken to compare, contrast and combine the main best practice frameworks applicable to Switching to produce a complete Work Breakdown Structure (activity) view of what a complete, best practice delivery phase should include. Based on this work, the following observations are made in respect of the Switching Programme:

**Agile.** Software and System Engineering, IT Service Management and Project and Programme Management are all generally now adopting or evolving to embrace Agile principles and methods. The area of Agile warrants further discussion in terms of its applicability to Switching and this is covered in more detail below.

**Design Management.** All the best practice frameworks include a range of common enabling themes or processes to manage and control of the design as it evolves through its development lifecycle. These invariably include: Change & Configuration Management; Issue and Defect Resolution; and Information & Knowledge Management. These processes are vital for effective testing and for delivery generally and the strategy and approach for these, linked with the Governance and Assurance arrangements, warrants focused attention; especially noting the multi-party delivery environment applicable to Switching.

**System Integration.** Best practice System Engineering and, increasingly, IT Service Management includes functions and processes to manage the integration aspects of the

solution. Given the complexity of the switching arrangements and the multi-party delivery environment, the whole solution or system will have to be divided into numerous parts for design, build and component level testing before being brought together (integrated) and tested at system and service level. Experience and lessons from multiple sectors suggests that where this has to happen, then risks occur across the boundaries; both with the technical boundaries (system interfaces) or between the interaction of the organisations delivering the separate parts of the system (e.g. due to a lack of communication and information sharing across organisational boundaries).

System Integration is about understanding these cross-boundary business and technical risks and proactively mitigating them as early as possible to avoid them materialising during formal integration testing or later where the cost and time penalties of resolution are much larger. Lessons learned from SMIP and Nexus below strongly reinforce the need for an effective system integration strategy and it is recommended that this is separately developed for Switching and that the testing strategy is aligned to it.

Agile. Agile has its roots in IT and software development. IT projects were being seen more and more as initiatives that delivered late and cost far more than originally planned. If that wasn't enough, what was delivered often didn't meet business requirements at time of delivery – either being out of date or full of misunderstandings. This was happening in a fast moving, competitive business environment and with a world economy in recession. Traditional approaches (often called “waterfall”) implemented a process-driven, almost production line, structure on what is essentially a creative and innovative discipline. Too much was determined up-front without allowing the evolution needed to ensure the end result will meet requirements of today, necessarily embracing change as it happens. Too little was delivered along the way – keeping everyone waiting potentially for years before being able to benefit.

Hence a new approach was needed, and Agile filled this gap resulting in a range of emerging Agile methodologies such as Dynamic Systems Development Methodology (DSDM - 1994), Scrum (1995) and Extreme Programming (XP - 1996). [The Agile Manifesto](#) was subsequently signed in 2001 and defined a set of principles. These can be summarised as:

- Agile puts the customer firmly in the centre. The driving factor is ensuring whatever is being developed will satisfy customer needs.
- As much as possible, an incremental approach is taken. Value is delivered to the customer early and often.
- Empowered, multi-functional teams ensure that delivered capabilities meet requirements. Although documentation is still important, it does not drive the outcome.
- The emphasis is on delivering items that add real value and delivering them on time.
- A culture of openness, honesty and transparency is fostered, ensuring that potential issues are surfaced before they become critical.
- Constant feedback is vital to ensure that the final result meets the needs of the organisation.
- Planning is vital, but plans will change.

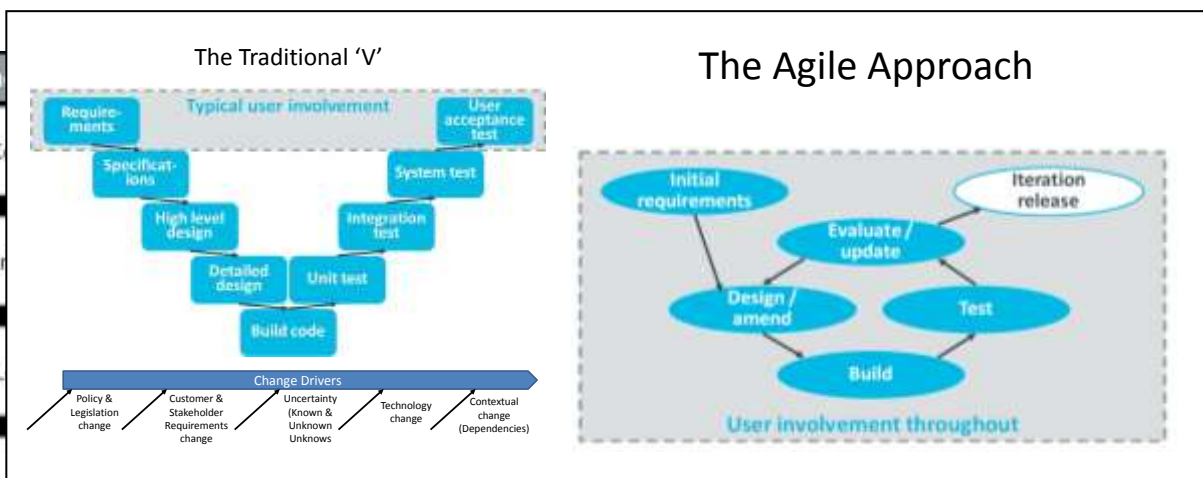
Good agile approaches ensure this happens in a controlled way, incorporating just enough planning, governance and design. There has been a concerted move to incorporate these agile principles into other frameworks such as Systems Engineering and P3M and the Saleable Agile Framework (SAFe) attempts to provide a tailorable application of Agile for a range of project situations and complexities.

Whilst the manifesto tends to focus on software development, the true concept of agile is far more. In fact, it is a philosophy that concentrates on empowered people and their interactions and early and constant delivery of value into an enterprise. The best agile approaches are highly disciplined and can and should be integrated into corporate procedures such as governance. This enterprise level agile is becoming more and more popular, with even relatively conservative business areas, such as the finance and public sectors adopting it.

On that latter point, a review of public IT programmes was undertaken in 2012 by the Institute for Government entitled 'System Error – Fixing the Flaws in Government IT'. This review established a high level Taskforce of government and private sector CIOs, senior civil servants and thought leaders to provide expert support and guidance on how to radically improve government IT. One of the key recommendations was to adopt Agile principles for areas of IT facing technological change or new ways of working as these are much more likely to deliver benefits when adopting an agile approach allowing innovation and experimentation to flourish.

This recommendation was based on an analysis of the root causes of failure in government IT programmes, summarised as follows:

- It is easy to understand why traditional methodologies have been followed in the past. The V-model or Waterfall approaches appear to be the logical response to the core challenges facing any project: the desire to come up with the best solution and deliver it in the most effective way.
- Under these traditional approaches, the best solutions are considered to be those that capture all the requirements up-front and design a solution to incorporate as many of them as possible. The greater the depth of the requirements, the more the solution will fit the business need.
- From these detailed requirements, the shape of the whole solution can then be developed. By planning and designing in as much detail as possible at the outset, showing exactly how everything fits together, the number of errors discovered in the later test phases are reduced.
- In a perfectly predictable world these approaches would work very well. **In the real world, in which requirements, technologies and ministerial priorities are constantly evolving, they quite literally build failure in to the system.**



### Lessons Learned

In addition to the consideration of best practice frameworks, standards and guidance, which are themselves built from multiple lessons learned, it is useful to also consider some recent and highly relevant individual case studies where lessons can be drawn that are directly relevant to the implementation of the new Switching arrangements.

Below are summarised some key lessons from a number of; particularly those relevant to Testing, System Integration and Post Implementation and to delivery generally.

### **Smart Metering Implementation Programme (SMIP)**

Sources: Interviews; DCC Annual Service Report 2015/2016. It is important to stress that the points below are not official lessons learned exercises for these programmes but are relevant

points gathered from publicly available documentation and interviews with key personnel involved in those programmes.

### Interviews

Need to separate detailed technical specifications from codes in a legal sense as it makes them difficult to write and very cumbersome to change.

Avoid writing specifications by committee.

Instil urgency and speed into the process of delivery.

Don't under-estimate complexity.

SMIP now has 'Joint Test Strategy' which is good – but was not the case from the beginning.

Avoid testing getting 'squeezed'; will run out of time. Get adequate testing signed up to early on and then defined it.

Specifications need to be mature/stable before go into design/build/test; if not this leads to a large change burden.

Late regulation/policy changes can have massive impact. Need to manage and control change.

Need to make sure code plus procurement vehicles (CRS) are agile enough to cope with change – build this in from the start and get the right commercials in place.

Within SMIP, no one was really looking at (or responsible for) end to end service performance; in reality this falls to suppliers as they have interaction with customers; DCC SMIP service is 'behind the scenes (i.e. a business to business service)

SMIP test phases include 'informal' testing ahead of Interface Testing and System Integration Testing (pre-IT and pre-SIT) phases. This has greatly helped de-risk and identify issues and improve User readiness ahead of formal SIT.

Need strong relationships between design and test teams; handover of design information; triaging issues; issue resolution and change.

De-risking ahead of formal testing has also been key – e.g. through use of emulators, test stubs, enabling test environments to connect to sandpit development environment, etc. (good example is GBCS for Industry - GFI).

Clear roles and responsibilities are key – need agreed RACI matrix.

Need strong interface specifications. Even with this, ambiguities will lead to different interpretations and engagement across parties is key – SMIP Design Forums have been highly successful in this regard; everyone needs to be brought along the 'design journey' together and avoid 'forcing' a design onto them. This will de-risk testing.

SEC effectively acts as the binding 'contract' unifying parties; but are SLAs and end-to-end service monitoring needed to allow someone to define and monitor overall service performance?

Governance on SMIP has improved over last 2 years, but still tends to work like 'legislative change'. An alternative would be to appoint some body to deliver the overall solution (separate change from BAU governance) recognising this might not be possible due to 'right of appeal'

Consider the use of prototyping/Agile – gives good insight – but need to think about procurement vehicles for build and operate to enable this

The need for a strong, capable independent system integrator is vital to de-risk design and identify issues ahead of formal integration testing

Improve quality of design specifications; avoid writing them 'by committee'. Avoid building detailed design specifications into regulatory codes

Instil some urgency into the process.

Ambiguities in specifications/codes lead to multiple problems only revealed during testing.

Providing 'test stub' early to Users [of the central service] de-risks their development and hence overall programme

Reducing 'exceptions' will help reduce early life problems.

Consider phased release Vs big bang.

Observe SMIP and Nexus in early life to get lessons learned from and early life (post-implementation) perspective

#### DCC Annual Service Report

The [defect] triage process was slower than anticipated, a key reason being reliance on a limited number of technical experts to conduct root cause analysis, resulting in delays in fixing defects.

Core systems were delivered into Release 1.0 Systems Integration Testing in three stages: SIT1, SIT2 and SIT3. This enabled DCC to complete development activities and Pre-Integration Testing with its External Service Providers and introduce code into integration testing in an incremental manner ensuring a high degree of control and greater visibility during testing.

Different assumptions had been made by Arqiva, Telefonica and Device manufacturers in terms of interpretation of GBCS, which differed from CGI's assumptions

Concerns were raised during Systems Integration Testing that the scope of the Systems Integrator was inadequately defined and focused, and under-resourced.... the service provided by the Systems Integrator did not meet expectations and was not adequately scaled to handle the size and complexity of the integration challenge.

Concerns raised during Systems Integration Testing led to DCC appointing a specialised IT consultancy firm with complex systems integration experience, Hunter Macdonald Ltd, to carry out a review of Systems Integration Testing and the role of the Systems Integrator

The change process has been thoroughly challenged during PY 2015/16, both in terms of the volume and complexity of Change Requests....Some of the difficulties experienced included:

- Poor quality or incomplete information
- Timescales for all stages of the process exceeded by both DCC and External Service Providers
- Resource conflicts as same small group of subject matter experts are required on several Change Requests as well as maintaining focus on other priorities
- Aligning External Service Provider responses when multiple External Service Providers are impacted by a single Change Request
- Pressure to deliver the change before the change process has been completed.



Sources: PwC Assurance Report; Interview with Nexus Programme. As for SMIP, it is important to stress that the points below are not official lessons learned exercises for these programmes but are relevant points gathered from publicly available documentation and interviews with key personnel involved in those programmes.

#### PwC Project Nexus Positioning Paper

The reality of an [unplanned] incremental delivery of functionality to Market Trials has had the impact of compressing market trial activity, with the trial of core functionality, defect resolution, the trial of new Unique Sites functionality and regression testing now having to occur at the same time

The issues that have contributed to this [lack of progress in testing] position have included the **'blocking' defects that have halted, and in some cases, continue to halt the efficient execution of end-to-end test scenarios, the complex coordination of testing across participants for certain scenarios such as Change of Ownership ('CoO'), issues with file confirmations and the downtime experienced from the recent priority 1 incident regarding incorrect data being included in some data files**

The aggregation of the challenges and issues above is making multi-party testing more complex than first envisaged by the Market Trials approach

The **high risk nature of the cutover and single major release go-live strategy** require a consequently very high level of confidence. There is, however, a misalignment across the market around the breadth and depth of activity that is required during transition.

Whilst Xoserve have used the additional time, gained from the previous go-live delay, to develop a standardised and robust SAP data migration approach, there **are no clearly articulated criteria defined around data quality** and how good this needs to be for go-live

The current schedule, however, allows no time for participants to test using data that has been loaded and transformed using the new rules. This creates a risk that issues will emerge in live operations that were not identified during Market Trials

#### Interview

Governance arrangements are overly reliant on assurance through regulation rather than through programme and technical controls;

Poor documentation of existing industry processes (comparing unfavourably with electricity) – for example interfaces and data items are defined, but conditionality of data attributes is not; relying on people with long service to know what works and what doesn't.

When new people come on board to undertake end to end testing, they're not aware of the unwritten rules and the data transfers fail.

Nexus is the biggest implementation of SAP IS-U (the SAP module specifically for utilities) in terms of volume, with 30m gas meter points, but not in terms of complexity; there is no customer billing for example – billing is done on an aggregated basis.

Xoserve are co-ordinating the (regulatory) code updates.

The programme is at the market trials stage now, with PwC measuring the delta between defect discovery and fix rates in the end to end solution – fix rates are lower than discovery, meaning the existing go live date is at risk.

Configuration management is an issue in the test environments, meaning that the Programme is in a shallow testing death spiral (i.e. a fix breaks some other components that were previously working, those are fixed, which breaks other components and may impact processing times / security controls, and so on).

The go live is going to be a 'big bang, all or nothing cut-over' with no roll-back, given the 40+ participants and the scale of the work that each will have to undertake to be ready if they don't work, settlement in the industry could be fundamentally compromised.

Applicability to Switching:

- Don't let participants hold the programme to ransom – if some aren't ready, this should not hold up the whole industry
- Phase the go-live as much as possible – e.g. a 'friends and family' approach, i.e. a very restricted customer base to start with.