Extract of Chapter on "ELEMENTS OF SYSTEM ENGINEERING & IT ARCHITECTURE"

In the complex world of technology today, almost no company can produce any new product or services without the assistance or partnership of other organisations. It is fundamentally necessary to understand what you are buying, why you are buying it, the compromises and risk you are accepting by making those decisions, and then finally the effect on your room to manoeuvre in the future. Figure 21 gives a high level view of the operating environments and key factors that should be taken into consideration when making technology system choices.

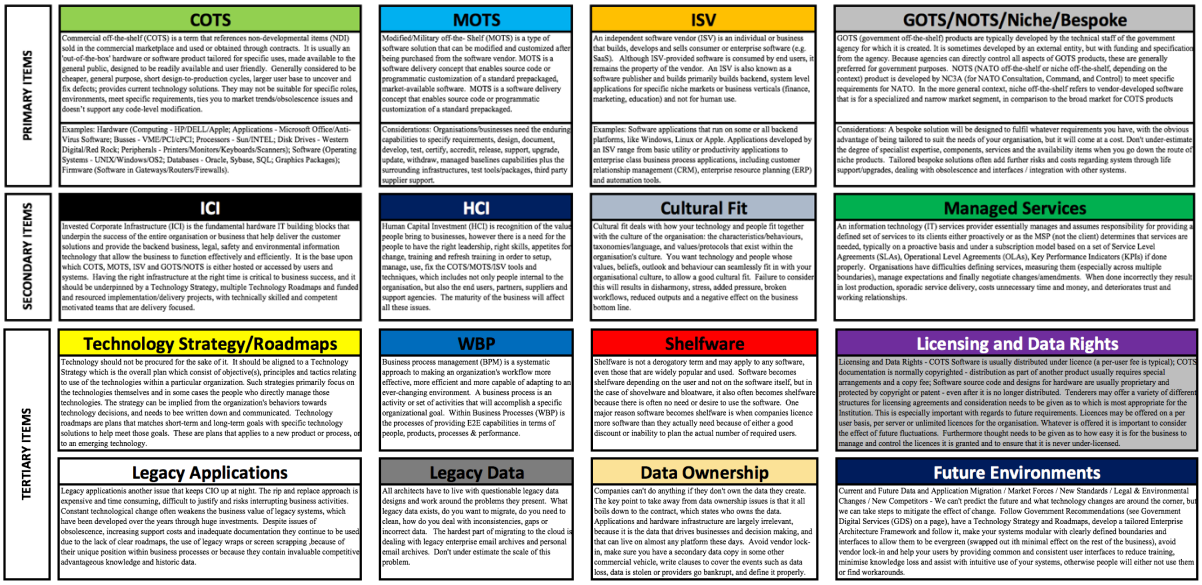

Figure 21: High Level of Key Technology Factors

Follow recommended published best practices where you can, as they have often been lessons learnt through very painful and costly processes of doing things the wrong way. Table 12 provides once such example, however there are lots available to those that are willing to invest a little time and effort to unearth them and take appropriate actions.

Government Digital Services (GDS) on a Page

Aims	1. Job is digital transformation of government
	2. Always start with user needs
	3. Remain Agile
	4. Work to a set of Design Principles
	5. Ensure government data is good data

Principles	1. Start with user needs - Do research, don't make assumptions
	2. Do less - Govnt do only Govnt work; Reusable and Shareable Platforms
	3. Design with data - Data driven-decisions; prototype & iterate after live
	4. Do the hard work to make it simple - it is hard work to make things simple
	5. Iterate. Then iterate again - Release MVP early; test with users; reduce risks
	6. This is for everyone - Accessible design is good design; Build for User Needs
	7. Understand context - think hard about the context of using the services
	8. Build digital services, not websites - It all adds up to what people need
	9. Be consistent, not uniform - Same Language; Same Design Patterns
	10 Make things open: it makes things better - Share code, designs, ideas, intentions, failures

Elements	1. Service Toolkit - Standards; Design & Style; Components; Monitoring; Buying
	2. Digital Marketplace - Buyer's Guide; Templates; Writing Reqmts; Shortlisting & Evaluation
	3. Introducing Verify - a new way to prove who you are online
	4. Introducing Notify - will reduce the need for many services to post letters to people
	5. Common Technology Services - Apps; End User Compute; Hosting; Identity & Integration; Networks; Service Management
	6. Buying - G-Cloud, Digital Outcomes and Specialists or Crown Hosting Data Centres framework

Table 12: One Example of IT Best Practice

We often forget that data and knowledge exist in different forms within the technology stack, and what you can do with it depends on where it exists and what standard or form it has been created within. Figure 22 is a modified example of the traditional 7 layer ISO, that attempts to show the interconnection of data within the standards, processes, applications and IT systems. There are many interpretations of this in the literature, and this is only one view point, and others may be more applicable. They key to a lot of things in life is to tailor what you need, keep or reinforce the good stuff, remove or reduce the bad stuff, and bear in mind in different environments and circumstances, priorities and previous criteria may well change.

11 Stage (Modified) Open Systems Interconnection (OSI) Data Model

	Layer	Protocol Data Unit (PDU)	Example Protocols/Standards	X-Layer
User Layer	10. Gov/Legal Compliance	User/Sector/Governmental/Legislation	ISO, British Standards, NATO STANAG, International Mil-Std	Enterprise Architecture
	9. Org/Political/Sector	User/Organisational/Political	Def-Stan, JSPs, Service Design Packages & Patterns, ICDs	
	8. User/Financial/Function	User (Capability Mgmt/Standards Board)	RFC, MODAF, DLoD TEPID-OIL, FRC 102, UK GAAP	
Host Layer	7. Application	Data (App Method/Protocols)	TLS, FTP, HTTP, HTTPS, SMTP, SSH, Telnet	Security Service, Mgmt Functions, MPLS, ARP, DNS
	6. Presentation	Data (App/Network format translation)	CSS, GIF, HTML, XML, JSON, S/MIME	
	5. Session	Data (Local/Remote App Connections)	RPC, SCP, NFS, PAP, AppleTalk	
	4. Transport	Segment (TCP) / Datagram (UDP)	NBF, TCP, UDP	
Media Layer	3. Network	Packet (Datagrams on same network)	AppleTalk, ICMP, IPsec, IPv4, IPv6	
	2.5 MPLS	Multiprotocol Label Switching (MPLS)	IP Packets, Native ATM, SONET, Ethernet frames	
	2. Data link	Frame (Node Transfer/Error Correction)	IEEE 802.2, L2TP, LLDP, MAC, PPP, ATM, MPLS	
	1. Physical	Bit (Raw Data/Encoding)	DOCSIS, DSL, Ethernet Physical Layer, ISDN, USB	

Layer	User/Software	Hardware Platform Components	System Software	X-Layer
11. Info Layer	User Generated Artefacts	Compute Hardware	Graphical User Interface (GUI)	Encryption & Info Handling
	Templates/Forms	Compute Memory	Operating Systems & Firmware	
	Rule Sets/Filters	Data Store Storage/Databases	Device Drivers	
	Workflows/Processes	Connectivity Elements	Mgmt Utilities (Service Access, Sign-in, Security, Cyber, Backup, Recovery)	

Figure 22: 11 Stage (Modified) OSI Data Model

Data in a very simplistic viewpoint exists in four main electronic forms, as depicted in Figure 23. When computers were first developed, data existed in mechanical form in terms of transient wheel/cog settings, but when the ability to capture and save data arrived on disks, it was in the form of Transferable Formatted Protocol Data (TFPD). The vast majority of data that we access today is in this form. As more complex systems were developed, we started putting data in containers, in the form of Containable Application Based Data (CABD) that was accessible through specific applications and queries. The arrival of the 'interweb' made Linkable Web Resources Data (LWRD) possible, and static data was now reachable around the world. We are entering a new age of Requestable Web Resources Data (RWED) where we can make requests and receive replies, from which we can make decisions that result in actionable workflows, trigger events or set up alerts.

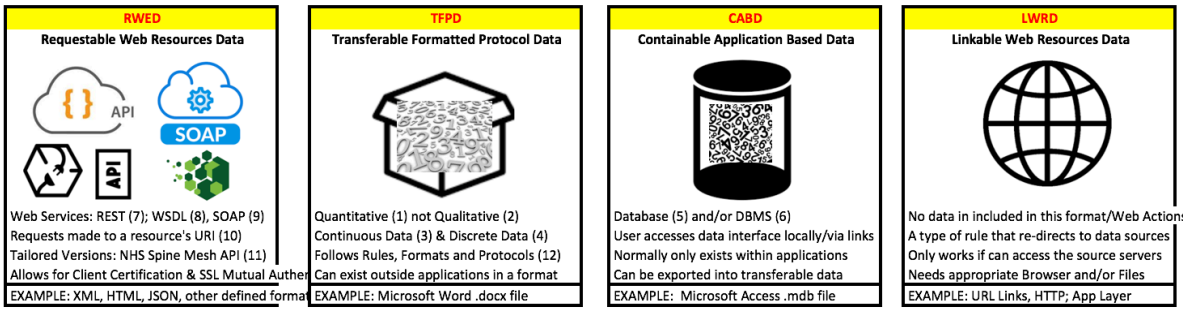

Figure 23: Data Main Forms

In this world of ever present data, we often overlook that fact that every piece of data we access has to be created, hosted, protected, powered, paid for by someone, maintained and cared for. We may think of the application server which we need to connect to, but how much thought is given to the surrounding systems that need to be in place in order to support the application server and to allow it to do its job. Figure 24 shows a generic hardware view of some typical servers and user access devices that are needed to access data. The viewpoint is not supposed to be comprehensive, only illustrative in that fact that for every application server there are multiple (30 in this case) back-end systems that need to be architecture, designed, funded, built, housed in a building, supported and maintained, to make the system work. Get any of it wrong and it could have catastrophic business impacts, cause large delays, costs lots of money to get right and take critical resources off other important activities thereby having huge knock-on effects that could take years to recover.

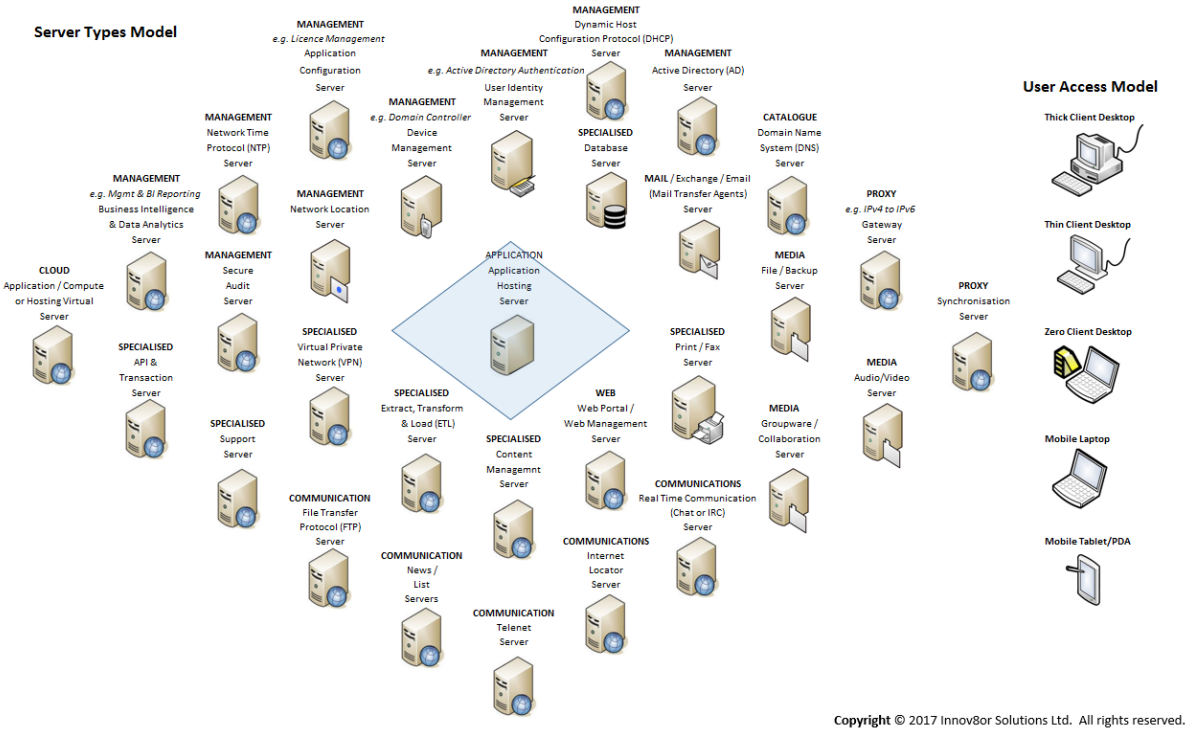

Figure 24: Elements of IT Architecture – A Server and User Access Viewpoint

Data is nothing without the people who are empowered to create it, classify it, cluster it, store it, access it, modify it, keep it up to date, aggregate it, ask the right questions of it, apply the right tools to it, apply the right techniques to it, sift it, process it, optimise it, filter it, analyse it, enrich it, check it, contextualise it, learn from it, make decisions on it and ultimately seek to gain business or personal value from it. These people need to be trained and re-trained in order to allow them to keep data mining to obtain the best business value, and recognise that however good the systems are … Rubbish in, still equals rubbish out.

The advert of big data means the right balance has to be achieved been personal and organisational data privacy and ensuring that people obtain the full value from the data. We have never had so much data at our finger tips, but there is a danger that we could be drowning in data but starved of information and knowledge. We need to surround data with processes, procedures and workflows in order to go through the 6 stages of data maturity, as displayed in Figure 25. Starting off at the Identifying, Dreaming and Imagination Phase

(coloured Yellow), working through the Collating, Realist and Planning Phase (coloured Orange) leading to the Critical Challenge Phase (coloured Red) where the Go/No Go decision is made to commit further resources to the idea. If successful, it enters the Setting Conditions for Action Plans Phase (coloured Green), followed by the PDCA Method Phase (coloured Light Blue) and ultimately ending up in the Benefits Realisation Phase (coloured Dark Blue), all the time data is matured through systems and processes.

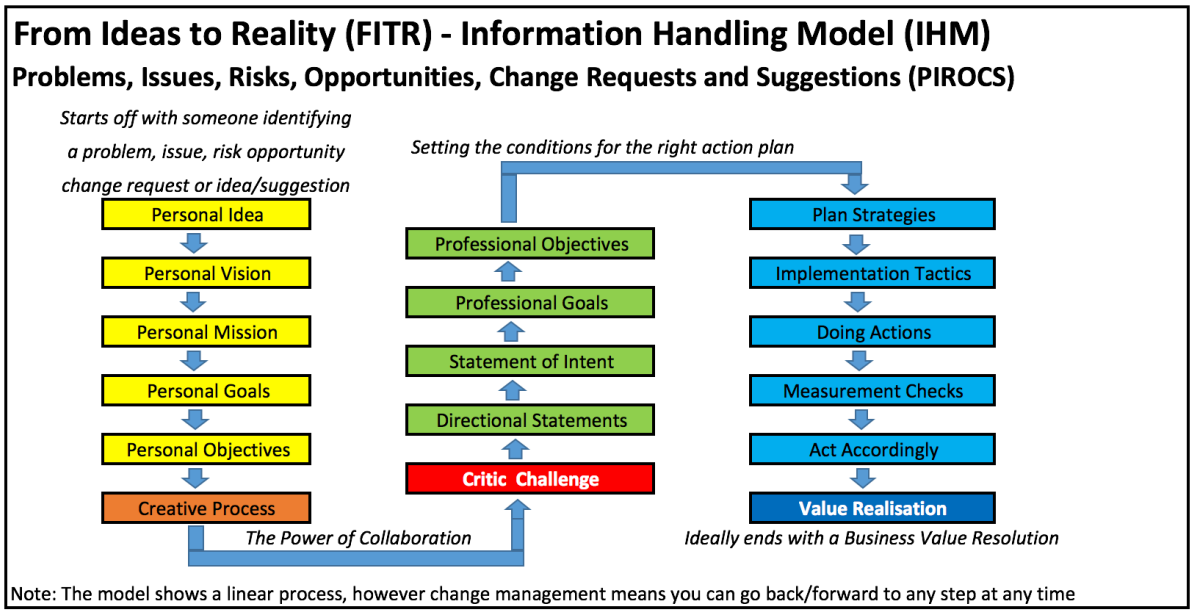

Figure 25: The From Ideas to Reality (FITR) Information Handling Model (IHM)

Traditional it was very clear what needed to be done to resolve a problem (IR1), as shown in Figure 26. As systems become more complex, inter-dependencies explode, emergent behaviours arise and not all the variables are known or even identifiable at the start, then IT Architectures are needed to augment human capabilities and help develop progressive solutions (IR6). Without the right people/processes, tools/architectures and techniques/methodologies there are plenty of scenarios (IR2 to IR5) that lead to failure or incomplete solutions.

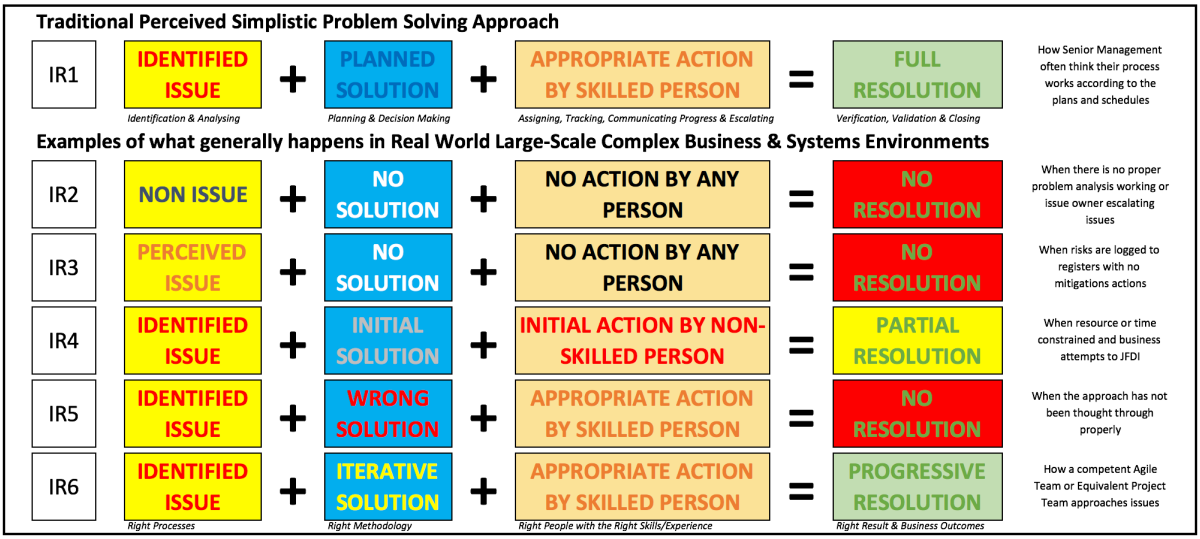

Figure 26: Approaches to Problem Solving

System Engineering can be considered the interdisciplinary approach and method to enable the realisation of successful systems from the point of view of satisfying the needs of the end users (in-service personnel), the customers (budget holders and setting of requirements), the business (achieving profit margins, maintaining/enhancing reputation and delivering capabilities) and other stakeholders (political, legal, safety and environmental). It is a proven structured comprehensive approach to solving today's complex technical challenges, particularly those related to the design and development of highly sophisticated systems, such as aircraft, ships, automobiles, telecommunications networks, energy infrastructures or information management systems. It is the role of the Systems Engineering Department to integrate all the key elements in a system including hardware, software, firmware, people, information, techniques, facilities and services, to achieve the required system capability (functional and non-functional) and performance.

It is not an exact science, there are multiple methods, taxonomies and standards, however whatever specific approach is adopted, but to be successful it normally needs the following elements:

- an analytical approach to model, understand, quantify and characterise problems;
- ability to understand customer needs and define functional plus non-functional requirements;
- creativity to explore innovative 'out of the box' approaches and solutions;
- to take a holistic view of the problem space;
- an awareness and knowledge of the many disciplines that impact on system design;
- strong communication and team building skills, because it is team sport to get the solution across the line;
- leadership skills to help motivate, train, mentor and provide purpose, plus people skills to give people autonomy and develop mastery of the skills needed to do the jobs.

Society has an expectation that technology will help us solve our biggest and world spanning societal problems ahead, however in order to do this we must:

- Understand, listen and sympathetic to the world we live in and its constraints and limitations to bounce back to the negative things we do to it;
- Find holistic synchronised solutions that have the ability to take into account a lot more eventualities and responses than we understand today, through detailed observation, critical analysis and iterative learning;
- Protect against unintended consequences, and provide solutions that are smarter, have a less impact on the environment being resource efficient and sustainable, self organising, robust, modular, safe and secure;
- Develop our understanding of the emergent and resultant behaviours of complex systems that will help us solve the problems;
- Finally look at ourselves, our behaviours/attitudes/needs/expectations, our skills and ability to create an evolving, diverse workforce that has the capability to use more complex tools, capacity to embrace change, can innovate and respond to competitive pressures.