Architecture and Engineering in Business and I.T.

tailored, through the use of analogy, to achieve success in large information technology projects

Enterprise Engineering
AUSTRALIA

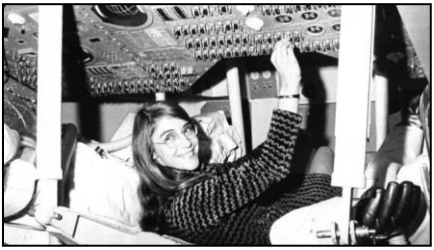
Bill Malkin (Director) BE(Civil), GDipCompSts, MEngSc(Systems), MIEAust
M 0414 838 083 • **E** bill@malkin.com
PO Box 280, Marden, SA 5070

This booklet is dedicated to Margaret Hamilton.

Margaret H. Hamilton led the team that created the on-board flight software for NASA's Apollo command modules and lunar modules. A mathematician and computer scientist who started her own software company, Hamilton co-created the concepts of asynchronous software, priority scheduling, and human-in-the-loop decision capability, which set the foundation for modern, ultra-reliable software design and engineering.[1]

Margaret Hamilton coined the term "software engineering".

It was an ongoing joke for a long time. They liked to kid me about my radical ideas. Software eventually and necessarily gained the same respect as any other discipline.[2]



Receiving the Presidential Medal of Freedom, 2016
Photo by Chip Somodevilla, Getty Images

[1] On receiving the Presidential Medal of Freedom, 2016 *http://time.com/4580807/president-obama-presidential-medal-of-freedom/*

[2] Meet Margaret Hamilton, the badass '60s programmer who saved the moon landing *http://www.vox.com/2015/5/30/8689481/margaret-hamilton-apollo-software*

Enterprise Engineering
AUSTRALIA

On average, large IT projects [Over $15 million] run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted. Software projects run the highest risk of cost and schedule overruns. …17 percent of IT projects go so bad that they can threaten the very existence of the company.

Bloch M, Blumberg S, and Laartz J, *Delivering large-scale IT projects on time, on budget, and on value,* McKinsey & Company, 2012

When reasoning by analogy, managers need not understand every aspect of the problem at hand. Rather, they pay attention to select features of it and use them to apply the patterns of the past to the problems of the present.

Giovanni Gavetti, Jan W. Rivkin*, How Strategists Really Think: Tapping the Power of Analogy*, Harvard Business Review, April 2005

It is useful to be aware that analogies are tools for leverage, and indeed part of their utility lies not just in extracting the insights they bring, but also in insights stimulated if we look for where the analogy falls short.

Ruth Malan, Architecture Consultant at Bredemeyer Consulting, Author, Blogger, Presenter and Trainer in System, Software and Enterprise Architecture, *www.ruthmalan.com*

Preface

Right now the software development industry is in crisis. It seems that we in the IT industry are very capable of carrying out projects worth up to around fifteen million dollars but our likelihood of success falls off very rapidly above that. For instance, there are the huge, billion dollar failures we've had recently in Australia, i.e. the Queensland Department of Health payroll system and the Victorian e-Health system.

My first degree is in civil engineering and I've been working in IT now for thirty-six years. About eight years ago I started to compare software development projects with building construction projects to try and work out what architecture and engineering professionals did, that IT people did not do, in order to greatly lower the risk of project failure.

From this ongoing investigation I've put together this booklet and have given presentations on this topic to the Mastering Complex Projects Conference, which was part of the Engineering Convention in Melbourne in 2014, the Australian Enterprise Architecture Conference in 2013, and in presentation programs run by the Australian Computer Society (ACS) and Engineers Australia.

It has struck me however that the IT industry is not too concerned with these failures. It seems that, because we all still get paid very well whether the project fails or not (failed projects can be swept under the carpet very easily anyway), the IT industry seems quite content to turn a six million dollar project (the original estimate for the Qld Health payroll system) into a one and a quarter billion dollar project failure.

As well as trying to motivate IT people to improve this situation, I now think that some pressure needs to be brought to bear from outside the IT industry, particularly from large companies that have been, or will be, severely affected by IT failures, ie most if not all of them, and from the relevant professional bodies such as Engineers Australia and the Australian Computer Society.

I believe it is about time that large software development projects began to experience the same high level of success that civil engineering projects now enjoy. I also believe that this booklet can substantially help to achieve that objective.

Table of Contents

Section 1: The Problem

1. Introduction

Today, most organisations utilise many technologies in order to source, process, transport and deliver products and services. All of these technologies, as well as most, if not all, of the business processes still performed manually, are underpinned by information technology. As Microsoft's Bill Gates says, *Information technology and business are becoming inextricably interwoven. I don't think anybody can talk meaningfully about one without talking about the other*[3].

Change is occurring in both the business and IT environments at a far more rapid pace now than it has ever been. *The rate of change is not going to slow down anytime soon. If anything, competition in most industries will probably speed up even more in the next few decades*[4].

Due to this extensive use of IT in a rapidly changing environment, the need to continually align an organisation's IT systems with its business direction has therefore become increasingly urgent and increasingly difficult. IT project failure, rather than success, has become the norm.

This booklet seeks to solve the mystery of why the IT industry is riddled with project failure. It does this by comparing IT projects with projects in the Building and Construction industry and others, to see what can be learnt from other project-based industries that have relatively high success rates. It also investigates the relatively recent phenomenon of job titles of architect and engineer being used in business and IT and attempts to better define their respective roles. Finally, it tries to understand why managers and decision makers in IT, although they know of best practice, continually fail to follow it.

---

[3] Bill Gates, 1955-, Philanthropist, Technology Adviser, Microsoft, *Business @ the Speed of Thought*, 1999.

[4] John Kotter, 1947-, Professor of Leadership, Harvard Business School, *Leading Change*, 1996.

The only real mistake is the one from which we learn nothing.

Henry Ford, 1863-1947, American Industrialist, founder of the Ford Motor Company, and sponsor of the development of the assembly line technique of mass production.

The IT industry is very labour intensive, has relatively low capital costs, is extremely exacting, and requires both excellent interpersonal skills and very advanced technical skills. It is one of the very few industries where jobs are not being replaced by automation, although the tools being utilised are becoming very powerful and sophisticated. For this reason and many others discussed in this booklet, the IT industry is simply not coping with the increasing demands being placed upon it.

Most authors of books on engineering, architecture and project management in business and IT, although they carry out very good research and come up with excellent findings and conclusions, really aren't able to look outside the troubled world of software development and peek over the professional walls to see what others may be doing.

For that reason, the authors tend to go over the same ground as each other and really don't come up with anything new. For instance, there are many books telling us what we should be doing, but there are none telling us why very few are actually doing it!

This booklet avoids providing opinions but attempts to (a) define concepts, job roles, etc, in a simple and concise way, and (b) make useful observations of, and comparisons with, other industries. In the light of this additional information, recommendations are made and, furthermore, the reader might themselves be better able to understand what is going wrong in their own IT projects.

Everyone is entitled to his own opinion, but not his own facts.

Daniel Patrick Moynihan (1927-2003), American politician and sociologist, United States' Ambassador to the United Nations and to India.

2. Failure

2.1 Introduction

This chapter was "inspired" by the recent, billion dollar failures in the Queensland Health Payroll and Victorian e-Health Information Technology projects. The approach here is to take a look at projects in the building industry, see how they differ to those in the software development industry, and attempt to determine if there was anything that the software development industry could learn from this.

> *Failure is not fatal, but failure to change might be.*
>
> John Wooden (1910 – 2010) The "Wizard of Westwood". American basketball player and coach. Named basketball All-American player three times, Named national coach of the year six times.

The achievements made in building construction projects for instance are nothing short of astounding. These achievements are not only in relation to the incredible size and complexity of the structures being built around the world, but also to their high quality and very low failure rate. Unfortunately, on the other hand, the *failure* rates occurring in software engineering construction projects are very high. For instance, in 2012 the Standish Group's appropriately named CHAOS Report[5] found that:

a. For large IT projects (i.e. over $10 million), 38% were considered failures (i.e. abandoned) and another 52% considered "challenged" (i.e. not staying within their budget or schedule, and/or not providing the specified functionality requirements). Only 10% were therefore considered to be successful;

b. For small IT projects (i.e. under $1 million) only 4% were considered failures and 20% considered challenged. 76% were therefore considered successful.

---

[5] The Standish Group. The CHAOS Report, 2012, *http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf*

According to a McKinsey report in October 2012, *on average, large IT projects [Over $15 million] run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted. Software projects run the highest risk of cost and schedule overruns*. It then adds, *17 percent of IT projects go so bad that they can threaten the very existence of the company.*[6]

> *The sad fact is that software projects fail because we do not recognize that good engineering principles should be applied to software projects just as they are to building office buildings. We try to defend ourselves by saying that software construction is "different".*
>
> Dr Paul Dorsey, American founder and President of Dulcian Inc, an Oracle consulting firm, and co-author of a number of books on Oracle.

From these statistics and the fact that IT projects are getting extremely large (i.e. with initial project estimates well over half a billion dollars), and because research has been occurring in IT project failures for many years now, it can be concluded that:

a. Some IT projects are getting so large and complex it appears that the probability of their failure, in terms of budget and schedule overrun and functional underrun, may well be approaching one hundred percent;

b. There have been countless IT project failure reviews and recommendations made over many years, but these recommendations seem mostly ignored.

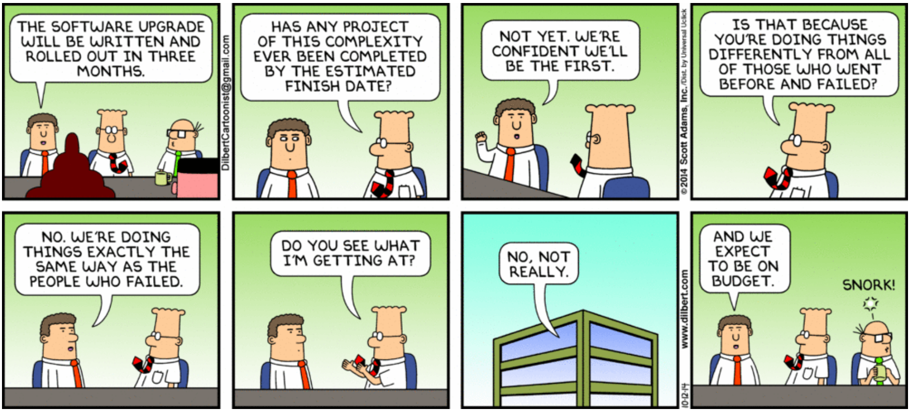
Albert Einstein might suggest that this situation is insane.

> *Insanity: doing the same thing over and over again and expecting different results.* Generally attributed to Albert Einstein (1879-1955), German

---

[6] Bloch M, Blumberg S, and Laartz J, *Delivering large-scale IT projects on time, on budget, and on value,* McKinsey & Company, 2012 *http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value*

Theoretical Physicist



This booklet does not provide solutions to the problems inherent in IT construction projects. The solutions are already well known in any case, and have been talked about at great length over many years. This booklet simply attempts to understand why these solutions are so rarely fully implemented. It is then hoped that, once these barriers to the adoption of the well-known solutions are known, IT Project Managers might have a much better chance of successfully completing IT projects.

Everyone has a plan: until they get punched in the face. Mike Tyson, 1966-, former undisputed heavyweight champion of the world.

2.2 A Definition of Failure

Defining what constitutes failure, and recognising when failure has occurred, are both quite contentious issues themselves, making it a little difficult to compare statistics on IT project failure between research reports. Some maintain that the project has only failed when it has been abandoned [7] [8], and some, such as The Standish Group, say that if the project hasn't

[7] Ewusimensah, K., & Przasnyski, Z. H. (1991). On information systems project abandonment - An exploratory study of organizational practices. *MIS Quarterly, 15*(1), 67-86.

exactly met its goals then it has not been successful, i.e. it has either failed or is challenged (i.e. not meeting the project schedule, budget or functional requirements [9]). Some even say that failure is a judgement rather than an objective state.[10]

It is suggested that a failure has occurred if the item, during the process, or after the completion, of being built, has experienced one or more of the following:

1. Physical failure, i.e. physical collapse or dangerous weakness;
2. Functional failure, i.e. does not perform the functionality required;
3. Financial failure, i.e. runs over budget;
4. Failure to meet deadlines, i.e. runs over schedule.

> *How does a project get to be a year late? One day at a time.*
> Frederick Brooks, 1931-, American computer architect, software engineer, computer scientist and author "The Mythical Man-Month"

However, it does not include failures that occur for any of the following reasons, as they are not under the control of the project team:

a. The functionality, as requested and specified by the client, is not useable (i.e. there is no need for it, and therefore results in an operational loss, for instance an apartment building where the apartments are poorly designed and therefore not functional);

b. The form, as requested and specified by the client, is so unattractive it is unsaleable and therefore results in an operational loss;

---

[8] Ewusimensah, K., & Przasnyski, Z. H. (1995). Learning from abandoned information systems development projects. *Journal of Information Technology, 10*(1), 3-14.

[9] The Standish Group. (1995). The standish group report: CHAOS.

[10] Bussen, W., & Myers, M. D. (1997). Executive information system failure: A New Zealand case study. *Journal of Information Technology, 12*(2), 145-153.

c. The main contractor or a major sub-contractor becomes insolvent, severely impacting the construction process;

d. An act of God, or any other unforeseen event such as war or sabotage, causes the construction process, or the construction itself, to fail;

e. Political interference causes the construction process to fail, for example the early stages of the Sydney Opera House.

The Opera House gave [Davis Hughes, new NSW Minister for Public Works] a second chance. For him, as for Utzon, it was all about control; about the triumph of homegrown mediocrity over foreign genius.[11]

Considering the current sad state of our computer programs, software development is clearly still a black art, and cannot yet be called an engineering discipline. Bill Clinton

A quote in an April Fool's Day joke article in 1994 entitled *President Kills Software Engineering*. Incredibly, it is still being quoted as fact. http://tab.computer.org/fase/fase-archive/v4/v4n08.txt

[11] Farrelly, Elizabeth, 'High noon at Bennelong Point' in *Canberra Times*, Retrieved 1 December 2008 by Wikipedia. (See http://en.wikipedia.org/wiki/ Sydney_Opera_House #cite_note-canberratimes.com.au-41)

Section 2: Architects and Engineers

3. A Short History of Architecture and Engineering

3.1 Introduction

History books are the “Manuals of Life”, but unfortunately most of us don’t like reading manuals. With this in mind, just a very short history of architecture and engineering is included here. For those who do like “reading manuals”, a list of further reading is included at the end of this booklet.

> *Those that fail to learn from history are doomed to repeat it.*
> Winston Churchill (1874-1965), British Prime Minister

3.2 The Creation of the Engineering and Architecture Disciplines

Although civil and military engineering have both been practised over the past few thousand years, the term engineer did not eventuate until around 1325. At that time, a person who constructed a military engine (i.e. any mechanical contraption used for war) became known as an Engineer[12]. A Catapult, for instance, was a “siege engine”.

However, having said that, the word *engineer* is derived from the Latin *ingeniare* meaning *to devise*[13] and even the word *engine* itself originally stood for *any product of the mind*[14].

To distinguish the engineer who carried out similar tasks but in a civilian context, the two professions of civil engineering and military engineering came into existence[15].

---

[12] Oxford English Dictionary, Definition of engineer.

[13] Engines of Our Ingenuity, John Lienhard et al, http://www.uh.edu/engines/epi12.htm

[14] Engines of Our Ingenuity, John Lienhard et al, http://www.uh.edu/engines/epi718.htm

Architecture and Engineering in Business and I.T.

In the 1560s the term *architect* was coined[16]. This word is derived from the Greek *arkhi*- "chief" and *tekton* "builder, carpenter". In fact, two thousand years ago the Romans had a dedicated force of military engineering specialists that they referred to as the *architecti* [17].

> *Believe me, that was a happy age, before the days of architects, before the days of builders.* Lucius Annaeus Seneca, Imperial Advisor to Emperor Nero (4BC – 65AD)

On the face of it however, the two terms *engineering* and *architecture* appear to describe the same profession. Although this apparent confusion was sorted out very many years ago in the building industry, it still causes a great deal of confusion when an attempt is made to apply metaphors for these professions in other industries.

Although there are no restrictions on the use of the title of engineer, the title of architect is protected in many countries including Australia[18], and its use is illegal without first fulfilling the necessary requirements for registration as an architect. However, there are a few exceptions such as naval architect.

> *Architecture is a science arising out of many other sciences, and adorned with much and varied learning; by the help of which a judgment is formed of those works which are the result of other arts.*
> Marcus Vitruvius Pollio, Roman author, architect, and engineer during the 1st century BC, perhaps best known for his multi-volume work entitled De Architectura

---

[15] Engineers' Council for Professional Development definition of engineering, Encyclopaedia Britannica

[16] Online Etymology Dictionary, definition of architect. <u>*http://www.etymonline.com/index.php?term=architect*</u>

[17] Military engineer, Wikipedia

[18] Architects Accreditation Council of Australia, use of the title “architect”. <u>*http://www.aaca.org.au/architect-registration*</u>

3.3 The Dividing of the Engineering and Architecture Disciplines

The timeline below shows how these, initially combined, disciplines formed into their own, very distinct, professions in the United Kingdom.

1700s	Civil and military engineering began to form into two distinct disciplines.
1800s	Architecture and engineering began to form into two distinct disciplines.
1818	Institution of Civil Engineers (London) formed.
1828	Received Royal Charter, i.e. recognised as a profession.
1834	Royal Institute of British Architects formed.
1837	Received Royal Charter, i.e. recognised as a profession.

The story of civilization is, in a sense, the story of engineering — that long and arduous struggle to make the forces of nature work for man's good. Lyon Sprague DeCamp, The Ancient Engineers (1963)

Seven thousand years of human history would establish that the key to complexity and change is Architecture. John A Zachman (2003), "The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing". (Electronic book)

3.4 The Fundamental Difference between Architecture and Engineering

The big questions here are:

1. Why were the architecture and engineering roles split from each other?
2. What are the differences between the two?

3. Can anything be learnt from this by people using the titles of architect and engineer in other professions?

The answers are actually very simple:

1. Why did they split? Because there are two fundamentally different stages in design:
    a. Determining <u>what</u> the requirements are (i.e. what the customer wants in terms of both form and function).
    b. Determining <u>how</u> to build an item that meets those requirements.

> *Design is not just <u>what</u> it looks like and feels like. Design is <u>how</u> it works.* Steve Jobs (2003)

2. What are the differences? The differences result from the fact that each profession requires a very different set of skills in order to be able to answer the two questions of <u>what</u> and <u>how</u>.
    a. Architects work with the customers in order to determine the requirements so, fundamentally, they must have excellent people skills.

    > *Bad architecture is, in the end, as much a failure of psychology as of design.* "The Architecture of Happiness", Alain de Botton (1969-) Swiss/British writer, philosopher, television presenter and entrepreneur. Honorary Fellow of the Royal Institute of British Architects, in recognition of his services to architecture

    b. Engineers work with the technologists in order to build the item, so, fundamentally they must have excellent technology skills.

    > *One has to watch out for engineers - they begin with the sewing machine and end up with the atomic bomb.*
    > Marcel Pagnol (1972), French Novelist, Playwright and

Filmmaker, *Critiques des Critiques*

3. Can people using the title of architect or engineer in other professions learn anything from this? Yes indeed.

> *No matter what engineering field you're in, you learn the same basic science and mathematics. And then maybe you learn a little bit about how to apply it.* Noam Chomsky, 1928-, American linguist, philosopher, cognitive scientist, logician and political commentator. Professor Emeritus, MIT.

3.5 Design

3.5.1 Types of Design

> *Engineering, medicine, business, architecture and painting are concerned not with the necessary but with the contingent - not with how things are but with how they might be - in short, with design.* Herbert Simon, 1916-2001, American leader in information processing, decision-making, problem-solving, attention economics, organization theory, complex systems, and computer simulation of scientific discovery. Winner, Nobel Prize in Economics, 1978.

There are many designers involved in both software development and building construction. Some of them are:

1. Architects, designing the form and function based on the user's requirements.
2. Engineers, designing the structure that will realise the architecture.
3. Interior Designers, designing the interior of a building.
4. UX (User Experience) Designers, designing the look, feel and flow of a software application, the report and user documentation layouts, etc.

There are many other designers outside these industries as well, such as Industrial, Textile, Fashion, Jewellery, Furniture, Exhibition and Display, and TV, Film and Set designers.

The big difference between architectural design and all other forms of design is that, if the architectural design is wrong then the impact on all aspects of the built item can be significant and is often where the project fails. The engineering design can never be correct unless the architectural design is correct. On the other hand, if the graphical design is wrong it does not (or should not) affect the engineering design or the architectural design.

3.5.2 The Design Institute of Australia

The professional body for designers in Australia is the Design Institute of Australia (DIA). Their definition of a designer is *a business professional who develops solutions to commercial needs that require the balancing of technical, commercial, human and aesthetic requirements*[19]*.*

They also state that *a designer plans things for manufacture or construction. The difference between a designer and a craftsperson or artist is that designers usually develop things that have requirements set by others and will ultimately be produced by others.*

Both these statements would also describe architects and engineers; however, for instance, the DIA adds that *design is also a term used by technical professionals such as engineers or software developers*. *These professionals must reconcile the technical and commercial requirements of projects.*

The DIA go on to divide design into three categories:

1. Industrial Design (or Product Design).
2. Interior Design (or Interior Architecture, Spatial Design).
3. Graphic Design (or Visual Communication).

From this it can be concluded that, for the industries being examined here, the relevant design categories are:

- Interior Design for Building Industry projects.
- Graphic Design (which includes UX Design) for IT projects.

---

[19] http://www.design.org.au/

However, it is important to note that Interior Designers do not work exclusively in the Building Industry, and Graphic Designers do not work exclusively in IT.

4. An Overview of Engineering and Architecture

> *If you can't explain it simply, you don't understand it well enough.*
> Albert Einstein (1879-1955), German Theoretical Physicist

4.1 Which engineers work with which architects?

There are many more engineers than architects. Outside of IT most architects work with engineers, however, only a subset of civil engineers, i.e. architectural engineers who work in the Building Industry, work with architects. This is explained below. (Please note that the term *technologists* here include tradespeople.)

1. Most architects work with engineers and technologists;
2. Most engineers work with technologists;
3. However, few engineers, only a subset of civil engineers, work with architects:
   i. It includes only those civil engineers who work on the construction of large buildings and landscaping projects. These engineers have more recently been called *architectural engineers*.
   ii. It does not include those civil engineers who work on the construction, for instance, of roads, oil refineries, power stations, sewage treatment works, bridges, pipelines, harbours, unless some part of the construction requires an architectural design.
   iii. It does not include the other engineering disciplines of aeronautical; chemical; communications; electrical; electronic; environmental; mechanical; mechatronic; metallurgical; mining; petroleum; transport; or robotics engineering.

These last two groups of engineers do not work with architects as the requirements are predominantly technical and not based on a jumble of often conflicting thoughts, ideas and objectives held in the minds of a, at times, changing and elusive group of client representatives.

4.2 Who has ultimate responsibility?

The architect is generally the overseer when working on the construction of large buildings and landscaping projects. On the other hand, the engineer is generally the overseer working on the construction of, for instance, roads, oil refineries, power stations, bridges, sewage treatment works, pipelines, and harbours.

From this therefore two types of project can be identified:

1. The architecture-centric project.

   The architect is the overseer when the architectural element is more critical and complex than the engineering element. Here the architect needs to continually supervise the project, from the very beginning to the very end, to ensure it is meeting the client's expectations. The engineer, in this case reports to the architect.

2. The engineering-centric project.

   The engineer is the overseer when the engineering element is more critical and complex than the architectural element. Here the engineer needs to continually supervise the project, from the very beginning to the very end, to ensure it is meeting the technical requirements. The architect, if involved in some small elements of the project, reports to the engineer.

In either case, (1) or (2) above, the architect is responsible for the architectural element, and the engineer is responsible for the engineering element.

> *Design is what you do when you don't [yet] know what you are doing.*
> George Stiny, Professor of Architecture, Massachusetts Institute of Technology, June 21, 2002.

> *Engineering is the art of modelling materials we do not wholly understand, into shapes we cannot precisely analyse so as to withstand forces we cannot properly assess, in such a way that the public has no reason to suspect the extent of our ignorance.*
> Dr AR Dykes, British Institution of Structural Engineers, 1976

4.3 Summary

The architect and the engineer are both designers.

The architect creates the architectural design, ie what the client wants, in both form and function, without any concern for how it will be built.

The engineer's concern, on the other hand, is how it will be built and they create the engineering design from the architectural design.

The technologists/tradespeople work from the engineering design and ignore the architectural design.

However, the architect and client make sure the item being built adheres to the architectural design. They aren't interested in the nuts and bolts of the engineering design, unless the thing falls over and then everybody is interested!

The architect is the overseer in architecture-centric projects, and the engineer is the overseer in engineering-centric projects.

5. Engineers Who Call Themselves Architects

5.1 Introduction

There is one engineering discipline whose members prefer to be known as architects even though they have engineering degrees and are eligible to join professional engineering bodies. This discipline is Naval Architecture. This is made even more confusing by the fact that they often work with another engineering discipline, ie Marine Engineering.

On the face of it, it would seem that the Naval Architect plays a similar role to the Building Architect but in relation to shipbuilding, and the Marine Engineer plays a similar role to the Civil (Architectural) Engineer.

5.2 Naval Architects

According to Engineers Australia, *A Naval Architect is a professional engineer who is responsible for the safe design and specification of ships, boats, and marine structures, both civil and military, including merchant ships (cargo and passenger), warships, submarines & underwater vehicles, offshore structures (fixed & floating), high speed craft, workboats and pleasure craft*[20].

IMAREST, the international professional body for Marine Engineering, states that *Marine engineers are involved with the design, construction, installation, operation, maintenance and repair of the main propulsion engines and auxiliary machinery and systems found in all kinds of ships, boats and offshore installations. Depending on the job, they may also be responsible for everyone working in the engine room*[21].

From these definitions it is fairly obvious these two disciplines do not at all mirror the architecture/engineering partnership one would see in the Building Industry; it is more like an association between a Structural Engineer (the Naval Architect) and a Mechanical Engineer (the Marine Engineer). However, as shipbuilding would seem a little like building construction, the question arises as to who would be the equivalent of a building architect in shipbuilding.

The answer to this question is that shipbuilding is actually more closely related to the construction industry as the requirements are predominantly technical and not so much based on a jumble of often conflicting thoughts, ideas and objectives held in the minds of a group of client representatives. For this reason there is a much reduced need for naval architects to work with a shipbuilding equivalent of building architects.

---

[20] Engineers Australia, *Eligibility Criteria and Procedures for Registration in the Specific Area of Practice of Naval Architecture*, https://www.engineersaustralia.org.au/sites/default/files/naval_architecture_guideline_issue_1_rev_0-24aug2015.pdf, Retrieved October 2015

[21] Institute of Marine Engineering, Science and Technology, *How About Marine Engineering?* http://www.imarest.org/membership/education-careers/careers-in-the-marine-profession/how-about-marine-engineering Retrieved October 2015

Section 3: Software Architects and Engineers

6. Introduction and Executive Summary

6.1 Objective

The objective of this section is to determine why software development projects do not experience the high level of success as projects in a comparable industry such as the building industry.

6.2 Approach

The approach is to compare software development projects with the more traditional and successful building construction projects, and investigate the differences.

6.3 Findings

In making the comparison between software development and building construction projects, a number of problems have been highlighted.

1. There is little real incentive for the IT industry to improve their success rate as there is little accountability in large software development projects. It is easier to hide project failures than defend them to the taxpayers (for public works) or the shareholders (for private works).
2. Key roles in building construction projects rarely exist in software development projects or, if they do exist, the incumbents are generally very poorly trained. These key roles are Clerks of Works, Quantity Surveyors and Software Architects.
3. Personnel in the software development industry do not distinguish between architecture and structure so there is confusion between software architecture and structural software engineering.
4. The role of architect in software development projects is poorly understood and performed very badly. Hence there is too much focus on the technology and too little on the client's requirements. It is still a "techie" club rather than a disciplined profession.

6.4 Recommendations

To address the problems described above, the following recommendations are presented:

1. All IT projects above a certain cost should be reviewed by an independent authority on behalf of the taxpayer, in the case of public works, or the shareholder, in the case of private works. The findings should then be made available to representatives of the taxpayers/shareholders to ensure accountability on the part of all those involved in the projects.

   However, it is not envisaged that this recommendation would ever be voluntarily adopted by private organisations or government at any level.

2. To ensure that the client is fully aware of the state of the project at all times and that the project work is of a consistently high level, a Clerk of Works should be closely involved in the project at all times and report directly to the client. Furthermore, if the project is later subject to litigation, the client would then have a technical expert available to advise and assist them during the litigation process.

3. In order for project estimates to be more accurate, fully trained, experienced and reputable Quantity Surveyors should be engaged in the estimating process.

4. Software development personnel, particularly engineers and architects, need to be able to distinguish between architecture and structure and between software architecture and structural software engineering.

5. There needs to be a four year university qualification in software architecture available so that software architecture is raised to the same level as software engineering and hence create a balance between finding out (a) what the client's requirements are, and (b) how the requirements are going to be satisfied. At the present time, software development projects are focused too much on the technology and too little on the client.

6. There needs to be more rigour in the choice of job titles and these should reflect the roles the incumbents are playing. Furthermore, the job titles, roles, responsibilities and reporting lines should be fully specified in the contract.

7. The Building and Construction Industry

7.1 Sub-Industries

This industry consists of three sub-industries:

1. The Housing Industry.

   Projects in this industry generally do not require architects and engineers.

2. The Building Industry.

   Projects in this industry generally require architects and civil (architectural) engineers, and are generally architecture-centric, ie the architect needs to be the overseer. These projects include medium to very large building and landscaping construction projects.

3. The Construction Industry.

   Projects in this industry always require civil engineers and, for parts of the project, architects may be required. These are generally engineering-centric, ie the engineer needs to be the overseer.

For the purposes of this booklet, *Building Industry* is defined as that part of the Building and Construction Industry that requires the expertise of both architects and civil (architectural) engineers in the construction projects.

It is specifically the Building Industry therefore that is being compared with the higher end of the Software Development Industry, ie where the projects are too large to not engage software architects and software engineers.

7.2 Ambiguities

The word *construction* is ambiguous and therefore can create confusion. The ambiguity arises as construction occurs in the building industry, specifically the construction of buildings, as well as in the construction industry. Therefore the following terms are valid but could be a little confusing so the differences are explained here.

1. *Building and construction projects* – these are projects in both the building industry and the construction industry.
2. *Building construction projects* – these are projects in the building industry that relate to the construction of buildings and do not include other projects in that industry such as those relating to the maintenance of buildings.

It could be argued therefore, that a *software development project* should really be referred to as a *software construction project*, in order to distinguish it clearly from a *software maintenance project*. This is because it is very possible for “software development” to occur in software maintenance projects as well as in software construction projects.

7.3 The Fundamental Difference between Software Architecture and Software Engineering

As discussed in the previous section, the architecture and engineering disciplines split because there are two fundamentally different stages in design:

1. Determining what the requirements are (i.e. what the customer wants in terms of both form and function).
2. Determining how to build an item that meets those requirements.

Each profession therefore requires a very different set of skills in order to be able to answer the two questions of what and how.

> *Design is not just what it looks like and feels like. Design is how it works.*
> Steve Jobs (2003)

The same situation exists in the software development industry. The software architect of course needs to work closely with the client representatives, and the software engineer needs to work closely with the technologists. Only through a strong architectural/engineering partnership can these two critical areas be brought together in order for the client's requirements to be successfully implemented in Information Technology.

> *Nothing is so inspiring as seeing big works well laid out and planned and a real engineering organisation.* Frederick Handley Page, Aircraft Pioneer and Industrialist, after a visit to Short & Harland where they were building his aircraft, just before WWII

Having said that however, if a software architect, for instance, has both excellent people and technology skills, there is really nothing to stop them playing both roles, assuming it is a smallish project, otherwise they may simply not have time to carry out both roles successfully.

So a software architect needs to document the requirements and design the architectural solution, in a manner that is understandable to both the client and the software engineer. The software engineer needs to document and design the technical solution in a manner that is understandable to both the software architect and the technologists.

The software architect would seek advice from the business analysts and user experience (UX) designers, whilst the software engineer would seek advice from the developers, DBAs, systems administrators, and other technical experts.

7.4 Summary

The software architect and engineer are both designers.

The software architect creates the architectural design, ie what the client wants, in both form and function, without any concern for how it will be built.

The software engineer's concern, on the other hand, is how it will be built and they create the engineering design from the architectural design.

The technologists work from the engineering design and ignore the architectural design.

However, the architect and client make sure the item being built adheres to the architectural design. The functional, integrated and user acceptance testing phases then compare the built application to the architectural design, ie they make sure that the form and function of the application is as the client requested and which is embodied in the architectural design.

> *There is the satisfaction of watching a figment of the imagination emerge through the aid of science to a plan on paper. Then it moves to realization in stone or metal or energy. Then it brings jobs and homes to men. Then it elevates the standards of living and adds to the comforts of life. That is the engineer's high privilege.*
>
> Herbert Hoover (1874-1964), Mining Engineer, US President
> "The Profession of Engineering" (from his memoirs)

8. Differences Found Between Software Development and Building Industry Projects

8.1 Project Visibility

The first, and very obvious, difference between these two industries is the visibility of their respective projects.

In the building industry there is a very high level of project visibility. If a building construction project fails people may well be killed or seriously injured and built items that have failed remain there for the world to see, sometimes for many, many years.

On the other hand, if an IT project fails it is very easy to hide and, in fact, nobody associated with the project, from the Board/Government (depending upon whether it relates to private or public works) to the user and IT staff, and from the main contractor to the smallest sub-contractor, has any enthusiasm for bringing the reasons for the project failure, or even the project failure itself, to the notice of the shareholder/taxpayer. Failed IT projects can easily be swept under very small carpets.

IT project failures are generally only ever investigated when the failure coincides with a change in Board/Government as was the case for the Queensland Health Payroll project.

A lack of *Project Visibility* results in lower *Project Accountability* which, in turn, decreases the incentive for *Project Governance* which, in turn, decreases the incentive for *Project Discipline* which, in turn, decreases the likelihood of *Project Success*.

> *Accountability breeds response-ability.*
>
> Stephen Covey (1932 – 2012), American educator, businessman, and keynote speaker. Author, *The Seven Habits of Highly Effective People*. Professor, Jon M. Huntsman School of Business, Utah State University.

From this observation, it is almost guaranteed that, unless IT project failures are thoroughly investigated, for instance by bodies directly representative of the shareholders/taxpayers, then new IT projects will continue to demand very little accountability and hence continue to experience a very high failure rate.

8.2 Project Discipline

8.2.1 Introduction

> *We are arguing about the wrong thing… we're arguing about what architecture is. We ought to accept the definitions of architecture that the older disciplines of Architecture and Construction, Engineering and Manufacturing have already determined.* John A Zachman (1934-), CEO, Zachman International, 2009

Projects in both industries:

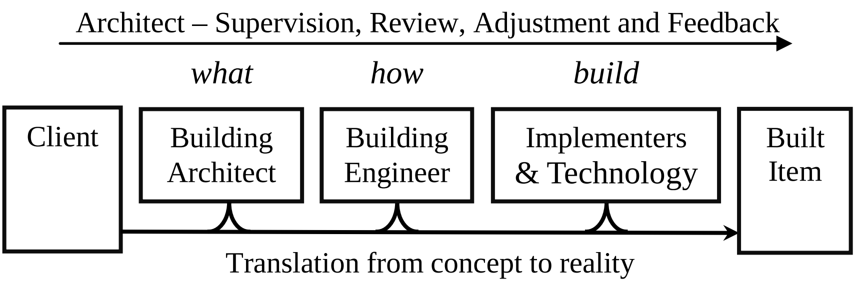
1. Carry out large construction projects, albeit the building industry, in the main, work with components in the physical world and the software development industry, in the main, work with components in the virtual world;
2. Consist of staff with similar titles such as project manager, estimator, architect, engineer, designer and technologist, although the IT technologists have skills in working in the virtual world and the building technologists (i.e. the tradespeople) have skills in working in the physical world;
3. Investigate and document the client's requirements, create the architectural design, and from this create the engineering/technical design. This is then passed to the technologists in order for the

item to be built, with supervision by the engineer and architect and managed by the project manager.

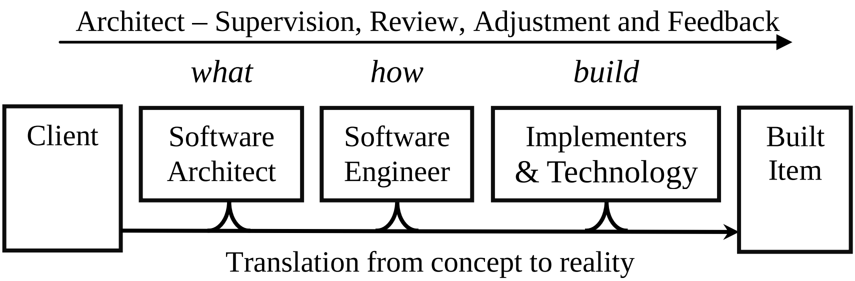
Below are two simple diagrams showing the essential roles involved in (a) the construction of a building, and (b) the construction of a software application, to demonstrate the parallels between the two processes.

> *Make everything as simple as possible, but not simpler.*
> Albert Einstein (1879-1955) German Theoretical Physicist

(a) Constructing Buildings



(b) Constructing Software Applications



There are certainly very strong parallels between the two industries in relation to the roles of engineers and architects. A great area of confusion within the IT industry however, with respect to project discipline, is the industry's adoption of job titles from the building industry without understanding what those titles mean.

This paper goes on to look a little closer at job titles used within the building industry to see how they might be better understood in the software development industry.

8.2.2 People, Roles and Job Titles

> *I handed my passport to the immigration officer, and he looked at it and looked at me and said, "What are you?"* An anecdote from Grace Murray Hopper (1906–1992), American Computer Scientist and United States Navy Rear Admiral. Received the inaugural Computer Sciences, unfortunately named, “Man of the Year” award.

When discussing professions such as architects and engineers in the various professional domains, the discussion is about roles, and not about people and certainly not about job titles – in the IT industry they are too often meaningless or misleading.

Furthermore, as in all areas of human endeavour, one person may well play a number of roles. In a more extreme case, for instance, an IT professional may perform the software architect role, then perform the software engineering role, and then maybe even the sales role.

However, in larger projects, roles tend to be individually allocated as the increased complexity in larger projects makes it necessary to hire specialists for each role, rather than generalists that have some knowledge and expertise across a number of roles.

> *Tell me, Mr Hoover, what are your interests?*
> *Madam, I am an Engineer*
> *Really? I took you for a gentleman.*
> An anecdote from Herbert Hoover (1874-1964), Mining Engineer, US President

8.2.3 Job Titles in Software Development Projects

As mentioned previously, there are two fundamentally different stages in building/software design:

1. Determining <u>what</u> the requirements are - what the customer wants in terms of both form and function. The result of this is the architectural design;
2. Determining <u>how</u> to build the item that meets those requirements. The result of this is the engineering design.

Architecture is the language by which the architect communicates with the client. It is also the term for the deliverables from the architecture process provided to the engineer in order for them to undertake the engineering design. Furthermore, architecture is how the built item communicates to all those that view and experience that built item. In other words, architecture is basically everything you can see, engineering is everything else.

In the IT industry we currently have solution and software architects who, as they should do, work with the Client to produce the architectural design. Unfortunately, there is some considerable confusion as we also have another, very different, group who call themselves architects but are actually performing engineering roles.

This group of people refer to themselves as technical architects, java architects, .Net architects, etc., when, in fact, they are software engineers. For instance, a technical architect is more likely to be a software engineer responsible for designing the structure or maybe the foundation of the application. They therefore correspond exactly with structural or foundation engineers (i.e. specialist civil engineers) who work in both the building and construction industries.

The confusion seems to arise from the fact that ***Architecture*** and ***Structure*** are considered to be the same thing in the IT industry, whereas they are recognised as being very different in the building and construction industries. The architecture is what you can see, feel and experience; the structure is how the architecture is realised and supported.

Another factor is that the term *architecture* comes to the software development industry from the terms *building architecture* and *computer architecture*. Designing computer architecture is inarguably an engineering pursuit, however designing the architecture of buildings is certainly not. We therefore now have two conflicting uses of the term architecture in software development so we have two groups of people calling themselves "architects" who do very different work. They are:

1. Those that do what building architects do, ie determine the client's requirements and the form and function of the object to be built.
2. Those that design the structure of the software application.

This confusion needs to be sorted out, and this is best done by renaming the members of one of these groups of "architects". Furthermore, as the latter group corresponds with structural engineers (i.e. specialist civil engineers), then software architects who design the structure of the software application should really be referred to as structural software engineers.

Architecture is designed by Architects and is what you can see, feel and experience, whereas the structure is designed by Engineers and is how the architecture is realised and supported.

There needs to be recognition in the IT industry of the distinction between architecture and structure so that members of software development and maintenance projects have a much better understanding of their roles, responsibilities and deliverables, and the roles, responsibilities and deliverables of their team members.

Although IT professionals are a little blasé about their job titles, the job titles in both the building and construction industries are controlled as it is essential that all members of the project, all stakeholders, and any others who might take some interest in the project know, for a particular issue, who they need to talk to, who has the ultimate responsibility for specific tasks, and what those people are required to do, who reports to whom and what deliverables they are required to produce. There is currently no such rigour in IT projects.

The IT industry must apply rigour to job titles. For a start it needs to correctly distinguish between architects and engineers and between architecture and structure. Only then can members and stakeholders of an IT project be able to understand the specific roles and responsibilities of each team member, and have a better understanding of the fundamental elements of the system they are building.

8.2.4 Summary

Imagination is everything. It is the preview of life's coming attractions.
Albert Einstein (1879-1955) German Theoretical Physicist

The architecture/engineering process is to start with the client's imagination and end with its reality:

a) The architect documents and pictures ***what*** the client is imagining (i.e. creates the architectural design) - the specific expectations of the client in both form and function.

b) The engineer documents and pictures ***how*** these specific expectations can be turned into reality (i.e. creates the engineering design) - the structure and all components that together will form the physical realisation of that architecture.

The architecture is ***what*** the client sees, experiences and understands, the engineering is ***how*** it will be physically realised.

The architect works with the client, the engineer works with the technologists, and the architect and engineer form a partnership through which the client's imagination is brought to life.

Similarly, the software architect works closely with the client representatives, the software engineer works closely with the technologists, and the software architect and engineer form a partnership through which the client's requirements can be successfully implemented in Information Technology

8.2.5 Business, Solution (or Systems) and Functional Architects

Projects in Business and IT often include specialised business/IT activities other than software development and, in fact, may not include software development at all. As software architects are only required in software development projects, and only in large ones at that, and as the project may involve very complex business processes or the implementation of a suite of large, integrated applications such as an Enterprise Resource

Planning (ERP) system, then other business/IT architects are required. These are:

1. Business Architects, who are required when the task involves very complex business processes and may need a number of projects to accomplish it. This may or may not require software development projects and therefore may or may not require software architects.

2. A Solution Architect, who is required when the task involves two or more projects relating to software development, network installation, IT hardware investigation and installation, application integration, etc. If a large software development project is involved then the software architect on this would report to the solution architect.

   The Solution Architect is often called a Systems Architect. *Large systems architecture was developed as a way to handle systems too large for one person to conceive of, let alone design. Systems of this size are rapidly becoming the norm, so architectural approaches and architects are increasingly needed to solve the problems of large systems. In general, increasingly large systems are reduced to 'human' proportions by a layering approach, where each layer is composed of a number of individually comprehensible sub-layers-- each having its own principal engineer and/or architect. A complete layer at one level will be shown as a functional 'component' of a higher layer*[22].

3. Functional Architects, who are required for projects relating to the implementation of a suite of large, integrated applications such as an Enterprise Resource Planning (ERP) system.

   The functional architect would gather the requirements from the client representatives and perform gap analyses between the

---

[22] Wikipedia, *Systems Architect*, https://en.wikipedia.org/wiki/Systems_architect, retrieved in October, 2015.

requirements deemed essential and the functionality provided by the “vanilla version” of the ERP.

They would then configure the ERP, or sometimes instigate a software development sub-project, in order to address each functional gap. Software development sub-projects that are instigated in this manner would generally not be large enough to require a software architect or software engineer.

8.3 Project Governance

8.3.1 Clerk of Works

The next, and also very obvious, difference between software development projects and projects within both the building and construction industries is the fact that most, if not all, IT project methodologies do not include the equivalent of the Clerk of Works.

In building and construction projects right around the world, the Clerk of Works reports directly to either the Client or the Client’s architect and ensures that the work undertaken by the main contractor and subcontractors is carried out as per the specification. The Clerk of Works carries out detailed inspections of the materials used on the project and the workmanship of the builders and their staff.

In some countries the title of Clerk of Works has been replaced by titles such as Site Inspector, Architectural Inspector or Quality Inspector. [23] (Despite its similar name, the Quality Inspector role bears no resemblance to the IT role of Quality Manager.)

Their main responsibility is to make sure that work is carried out to the client's standards, specification and schedule. They need to be vigilant in their inspections of a large range of technical aspects of the work.

Clerks of Works are not only inspectors but also superintendents. This means that they can advise the contractor about certain aspects of the

[23] Wikipedia, “Clerk of Works”, http://en.wikipedia.org/wiki/Clerk_of_works#cite_note-1

work, particularly if something has gone wrong. They can also agree to minor changes. They cannot, though, give advice that could be interpreted as an instruction, particularly if this would lead to additional expense.

They keep detailed records of various aspects of the work, which they put together in regular reports for the architect or planner and the client.

Clerks of Works liaise closely with the contractor's staff. They must, however, maintain their independence, as they are responsible for working in the best interests of their employer or client. If the work involves maintenance, alterations or additions to buildings by directly employed workers, Clerks of Works may be responsible for supervising them.[24]

On the other hand, in most IT projects very little visibility is given by the main contractor to the client. Feedback to the client is provided by progress reports and canned demonstrations of parts of the application under construction. This means that the client has little visibility in relation to the software quality, progress and adherence to standards and specifications, and cannot therefore make any reasonable judgement on value for money, project risks and likelihood of success. Furthermore,

1. The client would be incapable of making any informed decisions or taking early, corrective courses of action, if major project problems were being experienced.

2. The client would find it quite difficult to win any legal action they might need to take against the main contractor if they have very little (a) technical knowledge relating to the project, or (b) knowledge of the processes and procedures followed or not followed in the development of the software.

It is absolutely essential therefore, for all organisations commissioning an IT project to ensure they have the equivalent of a Clerk of Works, who has excellent technical and communications skills, working within the project

---

[24] The role of Clerk of Works/Site Inspector (Institute of Clerks of Works and Construction Inspectorate of Great Britain – Version 2 March 2010) http://www.icwgb.org/_files/duties.pdf

team. Furthermore, the Clerk of Works must report to the Client or to the Client's architect.

8.3.2 Software Architect

Although there are four-year degrees in software engineering offered by many universities and accredited by professional engineering bodies, there seem to be very few accredited degrees offered in software architecture.

To make matters worse, many software engineering degrees cover what they call "software architecture" when it is really structural software engineering that they are alluding to.

In Australia, the professional engineering body, Engineers Australia, recognises that software development requires a very important engineering component, and hence they accept graduates in these degrees for membership of this professional body.

On the other hand, the professional architecture body in Australia, the Australian Institute of Architects, does not recognise that software development requires a very important architectural component, and hence does not accept software architects, with any level or type of qualification, into the professional body.

This situation creates a critical imbalance in the software development process, where it still remains more of a "techie" process rather than a balance of the client-facing skills of the architect with the technology-facing skills of the engineer, i.e. there is not enough *what* and too much *how*.

8.3.3 Quantity Surveyor

Some pertinent quotes:

Mega-projects impose extra demand on effort estimation. Not only are more values at stake, but there will also be less relevant experience and historical data available. Many of the activities typical for mega-projects, such as organizational issues with many stakeholders involved, seem to be very hard to estimate accurately because they typically involve business *process changes, and complex interactions between stakeholders and with existing software applications.*[25]

Estimation of the software development effort has always been among the most difficult of processes. Firstly, the abstract nature of software makes it a difficult product to characterise, and design changes are often introduced into software before, during and after its pure code-production phase. Secondly, there are often too few projects in process within a particular organisation to provide a good basis for estimating the cost of new developments. Lastly, software production has evolved in recent years from being more of a black art into an engineering discipline in which there is a very rapid evolution of techniques. This makes the process of basing new estimates on anything but fairly recent developments somewhat precarious.[26]

The IT industry is notoriously incapable of estimating the cost of software development projects and is almost always substantially short of the actual cost. This will never be rectified until there are professional cost estimators whose reputation depends on their ability to estimate accurately. The IT industry therefore needs the equivalent of the building and construction industry's Quantity Surveyor to overcome this massive problem.

The Quantity Surveyor estimates and monitors construction costs, from the feasibility stage of a project through to the completion of the construction period.[27]

---

[25] *What We Do and Don't Know about Software Development Effort Estimation*, Magne Jorgensen, InfoQ, Aug 29, 2014. http://www.infoq.com/articles/software-development-effort-estimation (viewed 10 Sep 2015).

[26] *Engineering Costing Techniques in ESA [European Space Agency]*, D. Greves &: B. Schreiber, Feb 1995. http://www.esa.int/esapub/bulletin/bullet81/greve81.htm (viewed 10 Sep 2015).

[27] *What is QS*, Australian Institute of Quantity Surveyors (AIQS) http://www.aiqs.com.au About | What is QS (viewed 1 Sep 2015).

9. Conclusions

9.1 Findings

In making the comparison between software development and building construction projects, a number of problems have been highlighted.

1. There is little real incentive for the IT industry to improve their success rate as there is little accountability in large software development projects. It is easier to hide project failures than defend them to the taxpayers (for public works) or the shareholders (for private works).

2. Key roles in building construction projects do not exist as equivalent roles in software development projects or, if they do exist, the incumbents are generally very poorly trained. These key roles are Clerks of Works, Quantity Surveyors and Software Architects.

3. Personnel in the software development industry do not distinguish between architecture and structure so people who are the equivalent of structural engineers are confused with the equivalent of building architects.

4. The role of architect in the software development industry is very poorly understood, there is no formal university qualification as there is for software engineering, and there is no formal body promoting and protecting the profession

   Hence there is too much focus on the technology and too little on the client's requirements. It is still a "techie" club rather than a disciplined profession.

9.2 Recommendations

To address these problems the following recommendations are presented:

1. All IT projects above a certain cost should be reviewed by an independent authority on behalf of the taxpayer, in the case of public works, or the shareholder, in the case of private works. The findings should then be made available to representatives of the

taxpayers/shareholders to ensure accountability on the part of all those involved in the projects.

However, it is not envisaged that this recommendation would ever be voluntarily adopted by private organisations or government at any level.

2. To ensure that the client is fully aware of the state of the project at all times and that the project work is of a consistently high level, a Clerk of Works should be closely involved in the project at all times and report directly to the client. Furthermore, if the project is later subject to litigation, the client would then have a technical expert available to advise and assist them during the litigation process.
3. In order for project estimates to be more accurate, fully trained, experienced and reputable Quantity Surveyors should be engaged in the estimating process.
4. Software development personnel, particularly engineers and architects, need to be able to distinguish between architecture and structure and between structural software engineering and software architecture.
5. There needs to be a four year university qualification in software architecture available so that software architecture is raised to the same level as software engineering and so that there is a balance between finding out what the client's requirements are, and how the requirements are going to be satisfied. At the present time, software development projects are focused too much on the technology and too little on the client.
6. There needs to be more rigour in the choice of job titles and these should reflect the roles the incumbents are playing. Furthermore, the job titles, roles, responsibilities and reporting lines should be fully specified in the contract.

Section 4: Enterprise Architects and Engineers

10. Introduction and Executive Summary

10.1 Objective

The objective of this section is to determine why enterprise planning projects do not experience the high level of success as projects in comparable industries such as city planning.

10.2 Approach

The approach is to compare enterprise planning projects with the more traditional and successful building city planning projects, and investigate the differences.

10.3 Findings

Enterprise Planning is a difficult field to investigate as there is no single definition of what this is, what its objectives are, what skills it requires and what the acceptable terminology might be. Because of this, this investigation simply suggests how enterprise planning could best be carried out within the business. This is based on the approach used and roles played in similar, more traditional industry projects, particularly city planning.

10.4 Recommendations

The problems seen in the enterprise planning field are the same sorts of problems found in any relatively new field of study and activity. In the more traditional industries, these problems were sorted out by professional bodies, standards organisations and statutory authorities.

The very first recommendation is that a professional body be created to manage and protect the enterprise planning profession, and a standards body be set up, with members from all interested parties, in order to determine the relevant definitions, terminologies, methodologies, objectives, required skills, etc that related to enterprise planning.

This booklet demonstrates how enterprise planning, presently carried out by enterprise architects, would greatly benefit from a partnership between enterprise architects and enterprise engineers. Furthermore, it suggests how the roles of Enterprise Engineers and Enterprise Architects sit in relation to the enterprise strategists, enterprise technologists, the project management office and the business itself. Please see Figure 5.1 later in this section.

11. Comparison: Enterprise Planning with City Planning

11.1 Introduction

The comparison of Enterprise Architecture with City Planning has been discussed extensively in the literature. For instance, the summary from the ZDNet article, *Enterprise Architecture or City Planning*, states that *"City planning" is an easily understood metaphor that architects can employ to communicate more effectively the nature and value of architecture by relating the "unseen" enterprise architecture to real-world concepts that are well understood*[28].

Furthermore, on page 612 of TOGAF Version 9.1, it states, *the role of the enterprise architect is more like that of a city planner than that of a building architect, and the product of the enterprise architect is more aptly characterized as a planned community (as opposed to an unconstrained urban sprawl), rather than as a well-designed building or set of buildings*[29].

11.2 The architecture and engineering roles in planning

This City Planning metaphor, however, needs to be extended as it talks about the architecture role and omits the engineering role. City Planning and Enterprise Planning each require a partnership between an architectural and engineering role in order for the plan to be adequately specified and thereby useful to the people who will implement it. City Planning requires a partnership between a City Planner and a City Engineer, and Enterprise Planning requires a partnership between an Enterprise Architect and an Enterprise Engineer. If the engineering

---

[28] Brian Burke, *Enterprise Architecture or City Planning*, 2004, at http://www.zdnet.com/article/enterprise-architecture-or-city-planning/, viewed on 11th April 2015. Originally published by the META Group, now part of Gartner, on 1 December 2003

[29] Open Group Standard TOGAF Version 9.1, The Open Group (2009-2011), p 612.

component is not performed then the project managers, engineers and architects will simply not be able to make use of it.

Below is a definition of each of City Planning, Enterprise Architecture, City Engineer and Enterprise Engineering. This gives an idea of how closely related are City Planning and Enterprise Architecture, and also how each relies on an engineering component in order to be able to fully design their respective plans.

City Planning is the *activity or profession of determining the future physical arrangement and condition of a community, involving an appraisal of the present condition, a forecast of future requirements, a plan for the fulfillment of these requirements, and proposals for constructional, legal, and financial programs to implement the plan*[30].

The Australian Government defines **Enterprise Architecture** in a very similar way, i.e. Enterprise Architecture *describes the 'current architecture' and 'target architecture' to include the rules and standards and systems life cycle information to optimise and maintain the environment which the agency wishes to create and maintain by managing its ICT portfolio. The EA must also provide a strategy that will enable the agency to support its current state and also act as the roadmap for transition to its target environment. These transition processes will include an agency's capital planning and investment control processes, agency EA planning processes and agency systems life cycle methodologies*[31].

---

[30] Dictionary.com, at *dictionary.reference.com/browse/city+planning*, viewed on 11th April 2015.

[31] Australian Government Architecture Reference Models (AGRM) Version 3.0, page 290, at www.finance.gov.au/sites/default/files/AGA-RM-Final-v3.0-July-2013.pdf, viewed on 11th April 2015.

The City of Edgewater, Florida, defines the **City Engineer** quite simply as being *responsible for reviewing development that impacts the city's streets and utilities [ie, infrastructure]*[32].

The IEEE Technical Society on Enterprise Architecture and Engineering defines **Enterprise Engineering** quite simply also as *the engineering skills and activities necessary to construct and implement enterprise architecture*[33].

The main conceptual difference is quite obviously that city planning relates to city infrastructure and enterprise planning relates to enterprise infrastructure. However, both rely on an architecture/engineering partnership, and both are concerned with a managed transition from their current infrastructure to their target infrastructure.

The main difference between city planning and enterprise planning is that, with city planning, most of the building is carried out by private consortiums requiring the city planning process to ensure that they fit within their city plan. Enterprise planning, on the other hand, *should* result in projects that are handed over, through the Project Management Office, to the relevant business managers, solution architects and consultants who will carry out the project work. This means that city planning is somewhat reactive, whereas enterprise planning is somewhat proactive. However, in most enterprises, enterprise planning is still a piecemeal, reactive process.

> *A city is not an accident but the result of coherent visions and aims.*
>
> Leon Krier (1946 -), *The Architecture of Community*, Luxembourg architect, architectural theorist and urban planner. First laureate of the Driehaus Architecture Prize in 2003

---

[32] City of Edgewater, at http://edgewaterco.govoffice3.com/index.asp?SEC=8C88D9D6-B940-4E72-BA15-9735ADD5A77C&Type=B_BASIC, viewed on 11th April 2015.

[33] The IEEE Technical Society on Enterprise Architecture and Engineering, at www.ieeesmc.org/technical-activities/systems-science-and-engineering/enterprise-architecture-and-engineering, viewed on 11th April 2015.

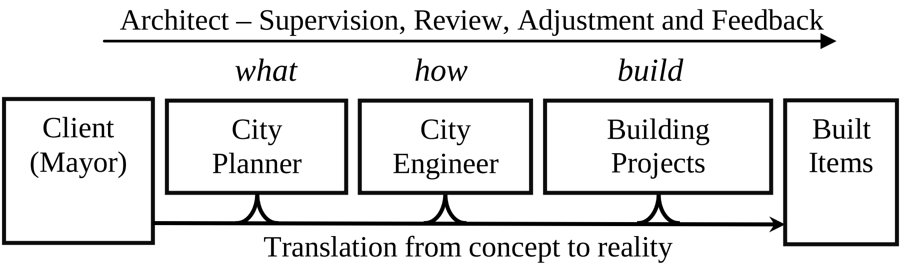


Diagram 1: The City Planning and Implementation Process

Every successful enterprise has a very clear strategic purpose.

Mitch Daniels (1949 -), American academic administrator, former Governor of Indiana and now President of Purdue University

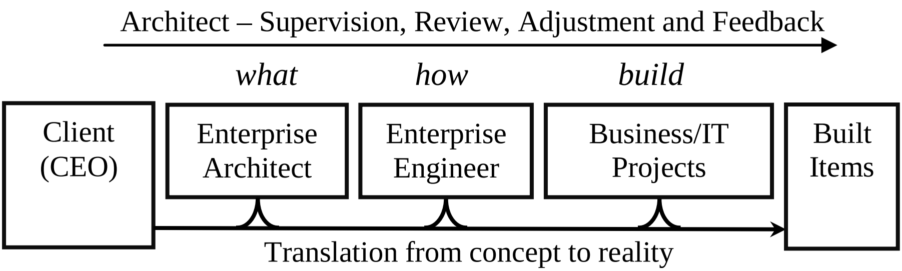


Diagram 2: The Enterprise Planning and Implementation Process

What people want, above all, is order.

Stephen Gardiner (1924-2007), British Architect, Teacher and Writer

12. Enterprise Architecture and Enterprise Engineering

12.1 Introduction

For many years, departments within enterprises have been fairly independent of each other and have determined requirements specific to their own, rather than the enterprise's, needs, and have either purchased or custom built applications to serve them. Once installed, it was often found that the departmental systems did not work well with each other and, furthermore, did not fit the strategic, or even tactical, needs of the enterprise or the critical day-to-day enterprise reporting requirements of upper management.

Any enterprise CEO really ought to be able to ask a question that involves connecting data across the organization, be able to run a company effectively, and especially to be able to respond to unexpected events.
Tim Berners Lee, British Computer Scientist and "inventor" of the World Wide Web

Up to that time, architects, engineers and project managers working in business and IT worked at the project level, where the project was aimed at meeting a subset of the needs of a department.

In order to avoid the problems of incompatible departmental systems and the inability of the systems, both IT-based and manual, to address the strategic aims of the enterprise, the need for two new enterprise-wide functions was identified. These are now performed by the Enterprise Architecture Office and the Project Management Office.

Architects however, always work in partnership with engineers – one working with the Client to determine the requirements, i.e. the *what*, and the other working with the Technologists to determine the implementation, i.e. the *how*. Therefore, over the past twenty years, the Enterprise Architecture Office has devolved into two disciplines, mainly in Europe and America, *Enterprise Architecture* and *Enterprise Engineering*.

In short, the Enterprise Architect needs to be located, and work closely, with the Corporate Strategists and managers at the CXO level, and the Enterprise Engineer to be located, and work closely with, the Corporate

Technologists. Only through a strong Architectural/ Engineering partnership can these two critical areas be brought together in order for corporate strategy to be successfully implemented in corporate technology. Please see Figure 5.1 for a diagram of the information flow from the Business, through Enterprise Architects and Enterprise Engineers and the PMO that is necessary to create, define and finally implement the projects that will take the enterprise from the current to the target state.

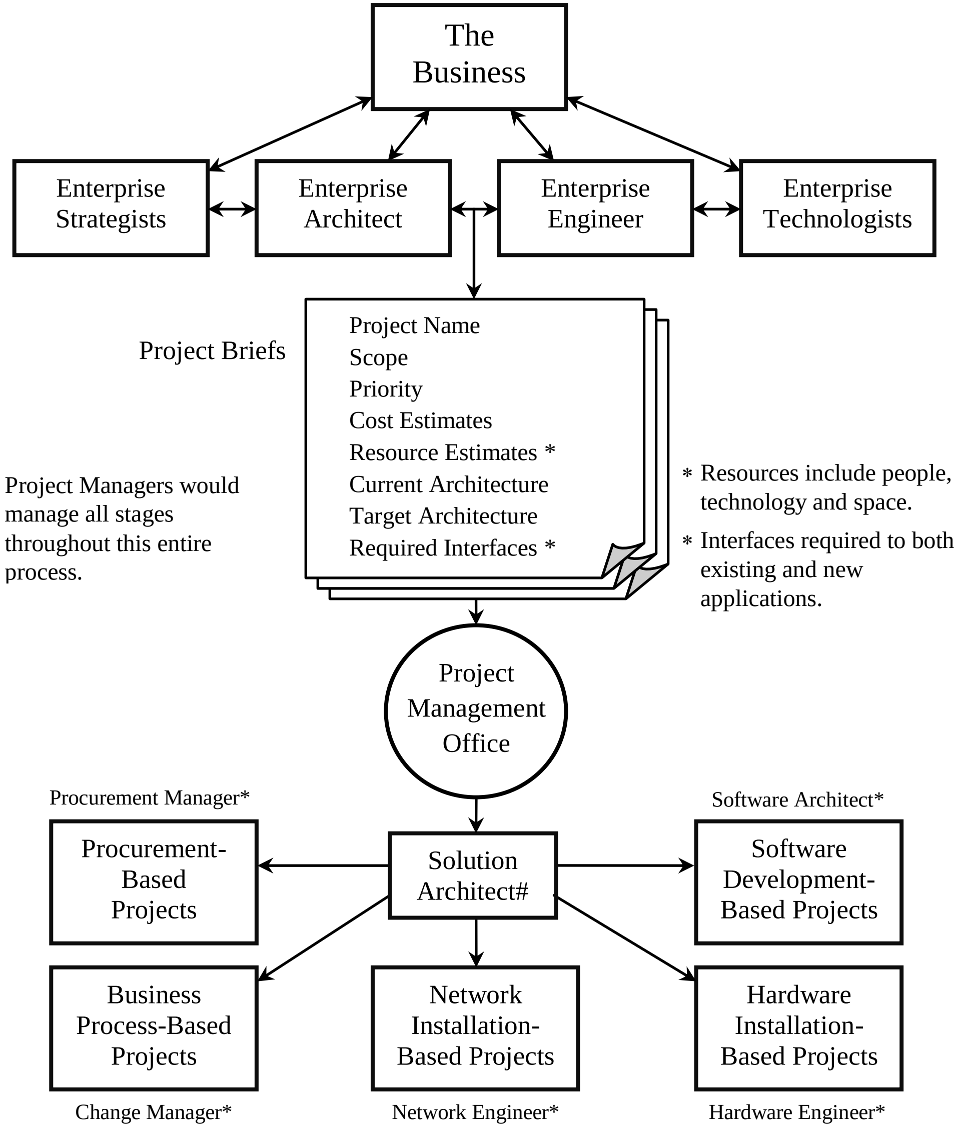
The initial steps in the process of aligning IT systems with the business strategy tend to be quite broad, abstract, political and dynamic. The process must therefore not be permitted to impact the IT specialists until the high-level requirements have been fully examined and the process has identified the building blocks required for the alignment.

Only once these building blocks have been scoped, defined “well enough” in terms of requirements and interdependencies, estimated in terms of cost, time and resources, and prioritised, should they then, via the Project Management Office (PMO), be taken on by the IT and relevant business departments to undertake their detailed design, implementation and operation, under project managers allocated by the PMO.

The difficult part however is that the initial, Enterprise Planning, process generally requires input and advice from key members of IT. It is the Enterprise Engineer who must negotiate with the CIO and their senior people to be able to gain this input while avoiding disruption to the essential processes being performed within the IT Department.

This chapter attempts to explain what Enterprise Architecture and Enterprise Engineering are, and how the Architect and Engineer work together and with senior management, other key business people, and senior IT software development and maintenance staff.

Figure 5.1 The Information Flow from the Business to the Projects that, together, will take the Enterprise from the Current to the Target State.



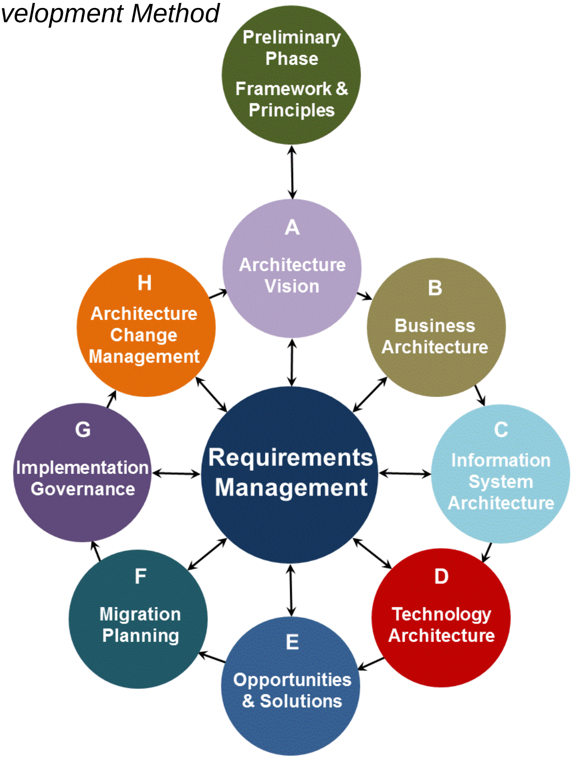
* Responsible for the form and functional fit of that project component
\# Client Liaison and coordination role responsible for the form and functional fit of the entire solution being built.

Enterprise Engineering
AUSTRALIA

12.2 Would A Different Perspective Be Useful?

Like any profession dealing with complexity, Enterprise Architecture needs to be studied from all angles in order to be better understood. For instance, below is how we generally view the TOGAF (The Open Group Architecture Framework) Architecture Development Method (ADM) diagram [34]. It becomes obvious that the ADM is much clearer when also viewing it side-on, as shown in Diagram 2[35].

Diagram 1: *The Architectural (Plan) view of the TOGAF Architecture Development Method*

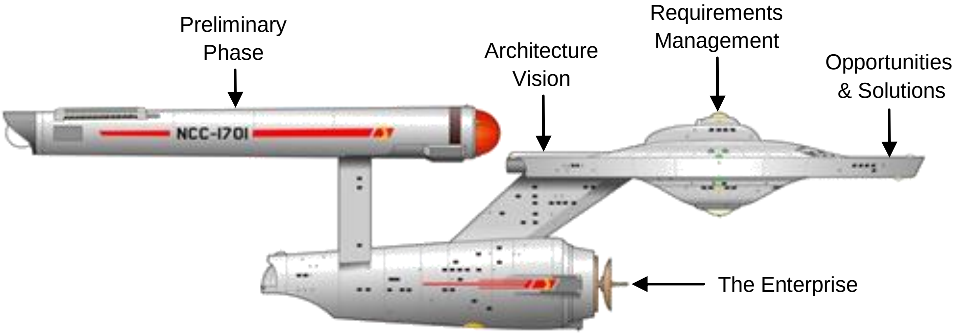


---

[34] Architecture & Governance Magazine http://architectureandgovernance.com/content/inside-togaf%E2%84%A2-9

[35] Ex Astris Scientia http://www.ex-astris-scientia.org/articles/new_enterprise_comment.htm

Diagram 2: *The Engineering (Side Elevation) view of the TOGAF Architecture Development Method*



By applying both an historical and a "lateral" examination, it is hoped that the role of the Enterprise Engineer, and their partnership with the Enterprise Architect, can be made a little clearer but, if that cannot be achieved, then at least make a fairly dry topic a little more palatable!

The Matrix.
Trinity: *Neo... nobody has ever done this before.*
Neo: *I know. That's why it's going to work.*

The remaining sections consist of simple questions and simple answers, in order to build a basic understanding and arrive at conclusions, hopefully without getting too bogged down in theoretical, abstract or political arguments.

12.3 The Enterprise

12.3.1 What is an Enterprise?

An Enterprise is any organisation of any size, public, private, not-for-profit, charitable, religious, etc. However, the enterprises we are mainly interested in are those that have become so big it is getting close to impossible for management to keep track of:

1. All their software applications and business processes.

2. All the interfaces between applications and between the enterprise and other enterprises, for instance between the enterprise and their customers and suppliers.
3. All the places their data is stored, eg databases, spread sheets, filing cabinets, people's heads and desk drawers, offline backups, etc, etc.
4. How the users, internal and external to the Enterprise, interface to each other, eg email, phone, text, paper, etc, and to the software applications, and what information is involved in each of these interfaces.
5. What processing and storage each application and person actually performs and how accurately and efficiently it is done.
6. How information is versioned, backed-up and restored.
7. How information is gathered and reported across the Enterprise.
8. How the health of the Information system can be monitored.

12.3.2 What Part of the Enterprise are We Interested in?

We are only looking at information, i.e. information formats, flows, processes, storage, interfaces, reporting and monitoring, whether or not it is handled by humans or some computerised or other, non-human, mechanism.

How the enterprise is structured, how the people are motivated and managed, training, legal and HR issues, how the enterprise produces its goods and services, and therefore basically all aspects of the enterprise other than those relating to Information, are not considered. (Obviously however, it would still be useful knowledge to have.)

Business, more than any other occupation, is a continual dealing with the future; it is a continual calculation, an instinctive exercise in foresight.
Henry R Luce, 1898-1967, American magazine, called "the most influential private citizen in the America of his day".

12.4 Enterprise Architecture

12.4.1 What is an Enterprise Architect?

Architects work toward a vision of something that does not yet exist. The vision should be exciting. It should inspire employees to expend vitality on bringing about change. James Martin, 1995. *The Great Transition: Using the Seven Disciplines of Enterprise Engineering to Align People, Technology, and Strategy*. Page xiii. amacom, 1995.

An Enterprise Architect attempts to determine, document and map, at a high level, all aspects relating to Information in the area of interest within an Enterprise.

They are generally required to first determine, document and map the "As-Is" (or Baseline) situation in relation to the Enterprise Information. The resultant documentation and mappings are referred to as the As-Is or Baseline Enterprise Architecture.

With high level management, the Enterprise Architect(s) then determine which Information areas need to be improved and/or replaced. The outcomes of these discussions are then also documented and mapped and this is referred to as the Target Architecture.

The Enterprise Architect resides in the organisation within the Business Strategy group and builds close ties with the key people in that group.

The follow-on objective then is to design ***how*** the Information systems, whether computerised or human, can best be moved towards that Target Architecture. That is when the Enterprise Engineer is required, and this design process results in the fully scoped and costed project briefs that can be passed on, through the PMO, to the solution architect for more detailed investigation and then, through the relevant engineers, their implementation.

As an architect, you design for the present, with an awareness of the past, for a future which is essentially unknown. Norman Foster, 1935-, English architect, famous for high-tech architecture. Awarded Pritzker Architecture Prize, Prince of Asturias Award in the Arts category, AIA Gold Medal.

12.4.2 Why doesn't the Enterprise Architect do the Implementation Design?

Like all other types of architect, eg Building Architects, City Planners, Software Architects and Landscape Architects, the Enterprise Architect has very basic technical skills and knowledge. Their job, which again is the same for all types of architect, is to find out <u>what</u> the client already has and <u>what</u> the client wants to have in the future.

This is an incredibly difficult job as it requires the architect to meet with different client representatives, who often each have a different idea of what is required and try to come to some compromise between them all. They then need to document these requirements very specifically so that (a) the client representatives know exactly what they are getting and, (b) the client representatives get exactly what they asked for.

> *All the revision in the world will not save a bad first draft: for the architecture of the thing comes, or fails to come, in the first conception, and revision only affects the detail and ornament, alas!* T. E. Lawrence (Lawrence of Arabia), 1888-1935, British Archaeologist, Writer and Soldier

12.5 Enterprise Engineering

12.5.1 What is Enterprise Engineering?

Here are some quotes from the experts:

Enterprise engineering *is a subdiscipline of systems engineering, which applies the knowledge and methods of systems engineering to the design of businesses. The discipline examines each aspect of the enterprise, including business processes, information flows, and organizational structure.* As quoted in Wikipedia - Jan Dietz, (2006). Enterprise Ontology - Theory and Methodology, Springer-Verlag Berlin Heidelberg

Enterprise Engineering *(EE) is defined as the engineering skills and activities necessary to construct and implement enterprise architecture.* IEEE Technical Committee on Enterprise Architecture and Engineering

Architecture and Engineering in Business and I.T.

Enterprise engineering *is a means of applying engineering method to Enterprise Architecture, developing and evolving the mapping [of] enterprise strategy to its resources.* This is from a presentation called the *Business Service Definition in Enterprise Engineering - A Value-oriented Approach*, given at the Enterprise Distributed Object Computing Conference Workshops (EDOCW), in the 2012 IEEE 16th International Conference

Engineering is the art or science of making practical.
Samuel Florman, The Existential Pleasures of Engineering (1976)

12.5.2 What is an Enterprise Engineer?

An engineer is someone who is good with figures but doesn't have the personality of an accountant. Anon

The **Enterprise Engineer** is responsible for investigating the technological aspects of the Enterprise Architect's design, i.e. the target architecture, determining, with the Enterprise Architect, the projects required to achieve that target architecture, and working with the relevant technical staff of the organisation to determine the technical feasibility of each project, cost and time estimates, required technical resources, interfaces, priorities, etc, ie the necessary information to fully describe the project briefs that will, together, realise that target architectural design.

In the organisation the Enterprise Engineer resides in the IT Department and builds close ties with the key people within that department.

Engineering ... to define rudely but not inaptly, is the art of doing that well with one dollar, which any bungler can do with two after a fashion.
Arthur Mellen Wellington, The Economic Theory of Railway Location (1887)

12.5.3 What does an Enterprise Engineer do?

The Enterprise Engineer, just as any other engineer who works with an architect, works out ***how*** the architecture can be realised, at a high level where the necessary implementation resources and technologies can be estimated and costed. They need to have excellent technical knowledge and be able to work with the architect on one side and the technologists and technology on the other.

All forms of architect work with an engineer. For instance a Building Architect works with a Building (Civil) Engineer, Software Architect with a Software Engineer, City Planner with a City Engineer, and a Landscape Architect with a Building/Foundation/Soils (Civil) Engineer.

So the architect has very specialised skills in working with people to determine ***what*** is required. The engineer has very specialised skills in working with technology to determine ***how*** to meet those requirements.

Engineers like to solve problems. If there are no problems handily available, they will create their own problems. Scott Adams, Cartoonist (www.dilbert.com) 1957-.

12.5.4 Why Are Enterprise Engineers Only Appearing Now?

The future has a way of arriving unannounced. George Will, 1941-, American columnist, journalist, author. Pulitzer Prize winner

A couple of things had to happen before Enterprise Engineers could be considered useful, let alone essential.

First of all, Enterprises had to reach a size and complexity where a specialisation at the Enterprise Architecture and Enterprise Engineering level would be needed.

When that happened, the Enterprise Architecture role was first developed, with the Architect initially performing both the architectural and engineering roles.

Enterprises have now reached such a complexity that these roles need to be done by specialists in each area, i.e. in the architectural process and in the engineering process.

Engineering is practicing the art of the organized forcing of technological change... Engineers operate at the interface between science and society. Dean Gordon Brown (1907-1996), born in Australia, Professor, Electrical Engineering, MIT. Received distinction of Institute Professor, MIT's highest academic honour

12.5.5 Why is Enterprise Engineering Needed Now?

The history of Enterprise Engineering goes back almost as far as Enterprise Architecture. When Enterprise Architecture became known to Enterprise managers around ten years ago, it took off very quickly and those who were prepared for it are now providing great benefits to the enterprise and, in turn, greatly benefitting themselves from having entered this relatively new discipline.

However, Enterprise Architects are presently receiving a lot of criticism for not being technical experts. This still occurs as the IT industry is only just starting to realise that Enterprise Architect is not a technical role.

As this realisation is spreading throughout IT and through the Enterprise management, they are now starting to look for the high-level technical people with the experience, knowledge and expertise to be able to perform the high level technical and interface design and costings.

Enterprise Managers are starting to see that, as with all other architectural domains, Enterprise Engineers are required for this step in bringing enterprise technology into line with enterprise strategy.

Opportunity is missed by most people because it is dressed in overalls and looks like work. Thomas A. Edison (1847-1931) American Inventor and Businessman

12.6 Summary

The imagination is man's power over nature.
Wallace Stevens (1879-1955), Lawyer, Insurance Company Executive and Poet. Awarded Pulitzer Prize for Poetry in 1955

The Architecture/Engineering process is to start with the Client's Imagination and end with its Reality:

a) The Architect documents and pictures ***what*** the Client is imagining, i.e. creates the Architectural Design - the specific expectations of the Client in both form and function.

b) The Engineer documents and pictures ***how*** these specific expectations can be turned into reality, ie creates the Engineering Design - the structure and all components and interfaces that together will form the physical realisation of that Architecture.

The Architecture is ***what*** the Client sees, experiences and understands, the Engineering is ***how*** it will be physically realised.

The Architect works with the Client, the Engineer works with the Technologists, and the Architect and Engineer form a partnership through which the Client's imagination is brought to life.

Similarly, the Enterprise Architect resides and works with the Strategists, the Enterprise Engineer resides and works with the Technologists, and the Enterprise Architect and Engineer form a partnership through which Enterprise Strategy can be successfully implemented in Enterprise Technology.

Our focus is getting organizations to recognize the architect's role and to understand what architecture is really focused on in terms of delivering value, and to recognize that architecture is the key success criteria to their business and their IT projects.[36]

Paul Preiss, CEO and Founder of IASA Global, (2008).

[36] ADTmag, *Q&A: IASA's Paul Preiss*, https://adtmag.com/articles/2008/06/18/qa-iasas-paul-preiss.aspx, retrieved October 2015

Conclusions

12.7 Findings

Enterprise Planning is a difficult field to investigate as there is no single definition of what this is, what its objectives are, what skills it requires and what the acceptable terminology might be. Because of this, this investigation simply suggests how enterprise planning could best be carried out within the business. This is based on the approach used and roles played in similar, more traditional industry projects, particularly city planning.

12.8 Recommendations

The problems seen in the enterprise planning field are the same sorts of problems found in any new field of study and activity. In the more traditional industries, these problems were sorted out by professional bodies, standards organisations and statutory authorities.

The very first recommendation is that a professional body be created to manage and protect the enterprise planning profession, and a standards body be set up, with members of all interested parties, in order to determine the relevant definitions, terminologies, methodologies, objectives, required skills, etc that related to enterprise planning.

This booklet demonstrates how enterprise planning, presently carried out by enterprise architects, would greatly benefit from a partnership between enterprise architects and enterprise engineers. Furthermore, it suggests how the roles of Enterprise Engineers and Enterprise Architects sit in relation to the enterprise strategists, enterprise technologists, the project management office and the business itself.

Section 5: Bibliography

Education is a progressive discovery of our own ignorance.
Will Durant (1885-1981) American author, *The Story of Civilization*

Alejandro Bernabeu Larena, *Origin of the Collaboration between Engineers and Architects in Great Britain in the Thirties*, undated, p367, http://www.arct.cam.ac.uk/Downloads/ichs/vol-1-357-378-bernabeu.pdf, retrieved October 2015

Architects Accreditation Council of Australia, use of the title “architect” *http://www.aaca.org.au/architect-registration*

Architecture & Governance Magazine http://architectureandgovernance.com/content/inside-togaf%E2%84%A2-9

Australian Government Architecture Reference Models (AGRM) Version 3.0, page 290, at www.finance.gov.au/sites/default/files/AGA-RM-Final-v3.0-July-2013.pdf, viewed on 11th April 2015.

Bill Gates, 1955-, Philanthropist, Technology Adviser, Microsoft, *Business @ the Speed of Thought*, 1999.

Bloch M, Blumberg S, and Laartz J, *Delivering large-scale IT projects on time, on budget, and on value,* McKinsey & Company, 2012 *http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value*

Brian Burke, *Enterprise Architecture or City Planning*, 2004, at http://www.zdnet.com/article/enterprise-architecture-or-city-planning/, viewed on 11th April 2015. Originally published by the META Group, now part of Gartner, on 1 December 2003

Bussen, W., & Myers, M. D. (1997). Executive information system failure: A New Zealand case study. *Journal of Information Technology, 12*(2), 145-153.

City of Edgewater, at http://edgewaterco.govoffice3.com/index.asp?SEC=8C88D9D6-B940-4E72-BA15-9735ADD5A77C&Type=B_BASIC, viewed on 11th April 2015.

Architecture and Engineering in Business and I.T.

Dictionary.com, at *dictionary.reference.com/browse/city+planning*, viewed on 11th April 2015.

Engineering Costing Techniques in ESA [European Space Agency], D. Greves &: B. Schreiber, Feb 1995. http://www.esa.int/esapub/bulletin/bullet81/greve81.htm (viewed 10 Sep 2015).

Engineers' Council for Professional Development definition of engineering, Encyclopaedia Britannica

Engines of Our Ingenuity, John Lienhard et al, http://www.uh.edu/engines/epi12.htm

Engines of Our Ingenuity, John Lienhard et al, http://www.uh.edu/engines/epi718.htm

Enterprise engineering: building 21st century organizations - James H. Brown*, Julian Watts 2002 http://www.sciencedirect.com/science/article/pii/096386879290013M

Ewusimensah, K., & Przasnyski, Z. H. (1991), On information systems project abandonment - An exploratory study of organizational practices. *MIS Quarterly, 15*(1), 67-86.

Ewusimensah, K., & Przasnyski, Z. H. (1995), Learning from abandoned information systems development projects. *Journal of Information Technology, 10*(1), 3-14.

Ex Astris Scientia http://www.ex-astrisscientia.org/articles/ new_enterprise_comment.htm

Farrelly, Elizabeth, 'High noon at Bennelong Point' in *Canberra Times*, Retrieved 1 December 2008 by Wikipedia. http://en.wikipedia.org/wiki/Sydney_Opera_House #cite_note-canberratimes.com.au-41

Giovanni Gavetti, Jan W. Rivkin*, How Strategists Really Think: Tapping the Power of Analogy*, Harvard Business Review, April 2005.

Architecture and Engineering in Business and I.T.

The IEEE Technical Society on Enterprise Architecture and Engineering, at www.ieeesmc.org/technical-activities/systems-science-and-engineering/enterprise-architecture-and-engineering, viewed on 11th April 2015.

Jan Dietz, 2008. Architecture - Building strategy into design. Academic Service.

John A. Zachman, 2003. The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing. (electronic book)

James Martin, 1995. The Great Transition: Using the Seven Disciplines of Enterprise Engineering to Align People, Technology, and Strategy.

James Martin, 1994. Enterprise engineering: the key to corporate survival, Volumes I - V. Savant Institute.

John Kotter, 1947-, Professor of Leadership, Harvard Business School, *Leading Change*, 1996.

Oxford English Dictionary, Definition of engineer

Online Etymology Dictionary, definition of architect. http://www.etymonline.com/index.php?term=architect

Ruth Malan, Analogy, www.ruthmalan.com

Open Group Standard TOGAF Version 9.1, The Open Group (2009-2011), p 612.

The Standish Group, The CHAOS Report, 2012, http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf

The Standish Group, 1995, The standish group report: CHAOS.

What is QS, Australian Institute of Quantity Surveyors (AIQS) http://www.aiqs.com.au About | What is QS (viewed 1 Sep 2015).

What We Do and Don't Know about Software Development Effort Estimation, Magne Jorgensen, InfoQ, Aug 29, 2014. http://www.infoq.com/articles/software-development-effort-estimation, retrieved September 2015

The Author

Bill Malkin

Director, Enterprise Engineering Australia

Experience

Thirty-six years' experience at all levels and in all disciplines in IT, particularly within software development and integration projects.

Academic Qualifications

Bachelor of Engineering (Civil), James Cook University

Graduate Diploma in Computing Studies, University of Canberra

Master of Engineering Science (Systems), James Cook University

Certifications

Sun Certified Enterprise Architect

TOGAF 9 Certified

Professional, Technical and Industry Memberships

Member, Institution of Engineers Australia

Member, Association of Enterprise Architects

Member, IEEE

Board and Committee Memberships

Technical Committee on Enterprise Architecture and Engineering (IEEE)

National Committee on Software Engineering (Engineers Australia)

Professional Activities

Presenter, First Enterprise Architecture Conference, Melbourne, 2013

Presenter, Mastering Complex Projects Conference, Australian Engineering Convention, Melbourne, 2014

Founder and Convenor of the Software Engineering and Architecture Joint Technical Program (sponsored by Engineers Australia) www.seajtp.org

Founder and Past Secretary/Treasurer of the Adelaide Chapter of the Association of Enterprise Architects

Enterprise Engineering
AUSTRALIA

Enterprise Engineering
AUSTRALIA

CERTIFIED Enterprise Architect

THE Open GROUP
TOGAF® 9 Certified

www.enterpriseengineering.com.au