Spring 4-11-2011

Reaching A Point Of Mutual Understanding: A Practitioners Approach To Systems Development

David Hagan
University of Bolton, d.hagan@bolton.ac.uk

David Roberts
Salford Business School,, d.c.roberts@salford.ac.uk

Follow this and additional works at: http://aisel.aisnet.org/ukais2011

REACHING A POINT OF MUTUAL UNDERSTANDING: A PRACTITIONERS APPROACH TO SYSTEMS DEVELOPMENT

David Hagan
Department of Business and Creative Technologies,
The University of Bolton, Bolton, UK.
Email: d.hagan@bolton.ac.uk

David Caton Roberts
Salford Business School,
The University of Salford, Salford, UK.
Email: d.c.roberts@salford.ac.uk

Abstract:

This practitioner paper explores the background of the development of two separate information systems. The first a case study where a SME was able to commission the design and implementation of an information system that offered a solution to restraints on the growth of the business, and a second case study of an SME, whilst unable to articulate the requirements for an Information system, with the support of a patient programmer, was able to commission a system that was developed, grew, and was sustained, over a ten year period. The paper explores the ideas of how both commissioners and developers can negotiate to a point of mutual understanding benefiting all stakeholders of the system and, as a result of learning from practice, The discussion point of this paper is a reflection on theory from the perspective of the practitioner.

Keywords: Practitioner, Reflection, Learning, Communication, Convergence, SME, Development

Practitioner paper

Introduction

An IT contractor was given the chance of a contract in Frankfurt with a large German leasing company. On his first day he was introduced to the Managing Director and the Board by his agent. There ensued some small talk in German and then the German MD asked the English contractor a question - in German. There ensued a period of silence while the question was digested by the Englishman. The contractor suddenly

realised what the MD had asked and blushed while stammering out "No it won't be a problem at all". The MD had asked -- in German – "Will your lack of the German language cause you any problems?" The contractor answered "no it will not." It had been a close thing, (an inappropriate answer to the question could have meant that the contract was lost) and the contractor was employed. The conversation set the scene for the coming months.

The outcome of the conversation was a re-evaluation in the way the contractor and the client were to approach their future communication. Discussions about the client's requirements followed a pattern of first the client attempting to specify his requirements, the contractor explaining what he thought the client meant by the request, the client would then explain what he meant and the programmer would then, once again, respond with his understanding of what was meant. Patiently the two parties worked towards a point of mutual understanding where they were both able to articulate communicative consistency. This process, which we compare to the Hermeneutic circle (Klein & Myers, 1999), enabled a successful outcome of the system development.

The acknowledgement that parties were communicating in a language that was not mother tongue to both enabled a pause which allowed the other party to reflect on what was being said and understood. This process contrasted the usual temptation in a time constrained commercial operation to just dive in and start to do some work.

The contractor was forced to simplify his language because the domain specific computing/programming exclusive language that was available in his English language was not available in the foreign language. Likewise the client was forced to think more about his language and his terminology. The outcome of the dialogue enabled a much thicker description (Ryle 1967) of both the whole and the detail of the situation.

What developed was a means of communication that helped build a joint and thorough understanding of the problem / requirements. To enlarge on this approach to communication we describe two case studies which illustrate how, both by effective communication and a mutual will and intention produced effective, durable, and long

lasting systems. In keeping with the practitioner stance of this paper the case material will be presented in the first instance, with the theoretical reflection to follow.

Case studies

The following case studies explore scenarios of two systems successes where individuals, working together, both whom trust each other, and where both had a genuine interest in the long term prospects of their business, and the business of their clients, enter a discursive struggle and attempt to move to a point of mutual understanding. As a result of reaching this mutual point of understanding, in both cases the actors, which we will describe as the business managers (manager) and the IT providers (contractor), were able to deliver through a system thinking approach viable information systems. The case studies offer an insight into how a success can be achieved in two contrasting cases, the first to automate an organisation's processes, and the second to transform the business an organisation undertook.

But before the case studies:

Imagine a conversation between a business owner, a programmer, and a business analyst. Is this the way to develop a system?

Owner "*All we need is a computer to speed up processing timesheets*"
Programmer *"It's not that simple - you need to analyse your business processes and go from there"*
Owner *"We don't have any business processes we just get timesheets in, work out the hours and rates, and pay the men"*
Programmer *"Ok - sounds simple enough. I know a guy who is a business analyst - I would like him to look at your business in case there is anything hiding in the woodpile."*

Sometime later.

Business analyst to Owner. *"I need to follow you and your secretary through a working week to see what you do"*

Owner *"How much will that cost?"*

Business analyst *"£xxxx"*

Owner (shrugs his shoulders and grimaces) *"How much! I can't afford that - how about if I tell you what I do"*

Business analyst *"OK Let's start at the beginning - I know nothing of the Construction Industry - How does it Work?"*

Owner *"I go out to construction sites and ask them if they need any men and leave a card. Sometimes they ring me up and I find some men and send them to the site. I invoice the company for their time and pay the workers"*

Business analyst *"Sounds simple - you leave it with me and I will rough out a system specification."*

Some time Later

Business analyst *"OK - you need stand alone relational database system to store details of companies, sites and men - we will have to have timesheet entry and invoicing. We can use a 3g system called Advanced Revelation which will give you data entry, reporting and pay calculations"*

Owner *"Could you say that again in English"*

Business analyst *"erm..."*

Programmer *"What he means is that we will use a system running in Dos to generate character screens that will be used for data entry and retrieval"*

Owner (looking perplexed) *"righty oh"*

Business Analyst *"Don't worry about it the programmer will write the system and install it on your pc"*

After the first prototype is demonstrated.

Owner *"It doesn't generate Invoices automatically when I enter all the timesheets for a job"*

Business analyst *"Do you have separate invoices for every site - even if the sites are owned by the same company?"*

Owner *"Yes but sometimes we generate two invoices on a site if we have a time and materials contract and a labour contract"*

Business analyst *"You have contracts as well!"*

Programmer *" That's going to screw up the data structures"*

Case Study One

Wessington Case Study

This narrative details the development of a single user Construction Agency System.

In the late 1980s a Small Construction Agency in the North East of England was starting to expand from a "One man and his dog" situation of three or four agency contract staff to 10s of staff and fixed price contracts. The secretary could no longer cope with the day to day running of the administrative tasks as it took several days to work out payments, write contracts, and to generate invoices and pay slips. Delays in processing the paperwork meant that cash flow problems arose as invoices were not going out on time and interim payments were being made to agency workers.

It was decided that what they needed was a computer system to speed things up, to automate the administrative processes. There were no relevant software products on the market at that time so they employed a business analyst and a software developer to develop a software solution.

Behind this cry for help was the belief that technology can solve problems. The company had used computers to write documents and calculate wages so it was logical to assume that a computer system would solve this reasonably well-defined problem.

All the business knowledge was locked up in the head of the business owner so the first task was for the analyst to extract it - easier said than done as the owner did not

know how the system would work. Why should he - this was the first time he had commissioned a computer system. He knew that it should be an end-to-end process - so time sheets and contracts in - pay slips and invoices out, but this only became apparent after several interviews as he had to work through the process himself. This coupled with the analyst and developer having no experience of the Construction industry meant that talking at cross-purposes was the rule. The acceptance of the analyst and the developer that they did not know what 'the thinker was doing' enabled them to generate a rich description, an understanding of the whole and also of the parts, of the business. The business owner also worked hard to gain an understanding of what the IT professional could provide – much more than a computer system.

It was quickly realised that a prototyping approach was needed. A product called Advanced Revelation was used to build the system. This was based on the Pick database but ran under Dos on PCs. It used a relational database based on a version of SQL and Screen forms which were character based and connected together in dialogues. The analyst and the developer already had experience of this software and could turn around designs quickly.

It was decided to start with core functionality - all the maintenance functions were written first to get the data in - Contracts, contract rates, agency staff and employment types. Timesheets were developed that were connected to the invoice generation and pay slip generation by functionality. The system took about two months to develop and all the timesheets could be entered in two days and the invoices and payslips printed out in one morning. Further developments to the system were tax and national insurance calculations and end of year runs for P11d and P60 and numerous reports such as debtors etc needed to run a small business - the system is still running after 20 years.

The relationship of the actors is important in attaining a mutually advantageous outcome. The subtext of the situation is this: The programmer is related to the client by marriage. The analyst is a good friend of the programmer and has worked many times with him. The analyst and programmer are looking to sell the software on and are treating the development as a "loss leader". The intention of the client to find a

solution is matched by that of the developers to provide one. Their intentions mutually converge making a good outcome more probable.

The Meridian Case Study

Meridian-Int was formed to offer sales and marketing solutions. The main offering of the organisation was of a customised professional outbound telephone sales operation which would allow Meridian's clients to build and maintain relationships with their customers.   This longitudinal case study tracks the development of a CRM system over a twelve-year period. Initially a paper based system used by two employees in an office in Manchester, then the use of a packaged off the shelf solution, later a bespoke system utilising Microsoft Visual Basic, and finally a web based system used by multiple users across Europe.   It also tracks the strained relationship and communication between the MD of Meridian and the lead developer as the system grew.

Whilst in large organisations the challenges of system development are (we hope) understood this is not always the case in a SME where there is an emergent situation and information technology, whilst important, may be seen as a necessity rather that a source of business advantage.

Project Description

Following on from a successful trial project around the sales and marketing of blood glucose systems for a global pharmaceutical company, the MD of Meridian was invited to share with the client the 'secret of success' and as a result was offered the opportunity to sign a long term contract with a client.

This secret of success was the 'system' that was used at Meridian to organise contacts with customers. The system involved a pen, a diary, a very relaxed sales call where the sales people worked hard to really understand the needs of the customers, and a commitment from the employees working on the contract to return calls to customers

when they promised that they would do. In explaining this 'secret' to the client it was understood by the client that the 'system' used was computer based and built on technology. (The client, embedded in the organisational culture of a technologically driven company assumed that the 'system' was technologically driven). As a result of the system the client, impressed with the results that Meridian had achieved in the trial programme, contracted the company to provide its 'CRM system' for use by 10 of the client's field based work force.

The MD of Meridian returned to his office in Manchester and asked his colleague who looks after computers to 'knock up a system' that his client could use. (8 weeks to launch).

Reflection from the MD.

The MD didn't know what he knew - his knowledge was an accumulation of his personal experience and, by nature of the entrepreneurial work he had undertaken as a generalist, was not from one discipline. Moreover his client did not know what he really wanted - his client was certain of the problem that he was trying to solve; he was trying to keep his job. Many years later the client explained that at the time market share for his product was collapsing and he was trying a myriad of projects in an attempt to arrest the decline and to save his job. The trial exercise undertaken by Meridian had been a success so he wanted to scale it up.

Both the MD and the IT professional invested a considerable amount of time trying to understand what the overall picture was and also to understand how they could deliver a system. The IT professional explained what might be possible and how a solution could work, whilst the MD was completely transparent about the strategic, financial, and managerial accounts of the company. Patiently, the programmer, using at first technical language, secondly layman's terms, and finally metaphor described to the MD the technical options, their advantages and disadvantages. Patiently the MD listened, questioned, and sought to understand. Patiently the programmer listened, questioned, and sought to understand. Together they worked towards a mutual point of understanding of the problem and possible alternative solutions.

An off the shelf version of Sage Telemagic (A CRM solution that linked to a standard accountancy package) using PC anywhere (a piece of software that enabled data to be transferred between two computers) and a rack of dial-up modems was used to bring together a 'system' that the client's employees could use from remote locations. Very brief details of their sales transactions could be recorded and then, at the end of each week, the representatives would dial up to the Meridian office in Manchester and upload their data.

This system worked well between January 2000 and December 2000. There were some problems however which developed, first around users being unfamiliar with using keyboards and IT equipment; and second around users claiming that the system was not working and that they were unable to complete their work. The client's representatives were charged with uploading their weeks work by close of business on a Friday evening. Whilst both the MD and the IT colleague had succeeded in communicating and understanding the initial requirement of the client this second problem – the problem of micro politics and resistance within the client's employees became an area for misunderstanding, and a potential contractual problem.

A pattern emerged of the client's representative claiming that 'over the weekend' the system was not working; (we now know that this is the classic user resistance familiar in IS implementation scenarios) the MD questioning Lead developer as to what was happening and why the system was not working over the weekend; the Lead developer contacting the client 's representative explaining that all was well; the MD reporting to the client representative that all was well; The clients representative uploading his work; and the lead developer unable to explain what the problem was to the MD; MD becoming perplexed with Lead Developer.

At a more Senior Level the client now wanted to roll the CRM system out to 30 more of their users. (Senior managers at the client had been using the computerised system to compare the apparent performance of individuals in an attempt to increase productivity. The original use of the system -- a mechanism to help customer service staff keep a track of relationship with customers -- had been usurped and the reports from the system were now being used as a tool to control employees.) This was to be the start of a period of more serious difficulty for the IT developer and also the MD of

Meridian. Additionally managers were requesting and consuming reports at a voracious rate. Sage Telemagic did not seem to offer facilities for reports to be automated.

Sage Telemagic no longer gave the flexibility that the client required of the system, and the Lead Developer at Meridian had programmed an updated system using MS Visual Basic that would allow for more flexibility and more importantly it would more easily enable the data that was being collected to be interrogated and reported upon. In June 2002 the first of many iterations of the MS Visual basic system were introduced and patterns emerged of difficulties with both the system and also with the use of the system. Users of the system resisted the new ways that their company was measuring them - and they found new language that would excite their managers. Cries from vocal sales representatives of "we feel that there might be some data corruption as sales we have entered now no longer show" were certain to start a series of questions and fruitless audits of the Meridian system. There were also problems with the internal users of the system at Meridian - the system was prone to crashing. It was now complex and twenty users within the meridian sales office and thirty remote users employed by the main client we all operation on replica data sets. Synchronisation of the replica data sets became a challenge and the health of the Lead programmer declined. The health of the MD also declined. The resources of the company declined too as more and more finances were invested in hardware in an attempt to keep the system operational. We now know that the lead engineer spent his nights on bulletin boards engaging with other practitioners in a community of practice (Wenger, 1998) sharing experiences, breakthroughs, and ideas. (The replication farm that he had developed was even of interest to one of the lead VB programmers at MS.)

Despite these difficulties both the MD and the developer continued to search earnestly for a point of mutual understanding. Unfortunately the validity of their understanding had been undermined by factors outside of their knowledge, such as, the resistance of the client's employees and the attempted use of reports of the system by the client to control their employees.

In 2004 there was a breakthrough as an unusual relationship, identified by the developer at Meridian, was shared with the client. Representatives who were

identifying difficulties with the Meridian system were in fact using the system to hide their own shortcomings, or as an excuse for their lack of sales achievement.

Dates stamps on the replica databases identified that some representatives had not even entered any data into the system even though they were claiming that the system was at fault. The revelation, once accepted by the client's senior management, was then used as another tool to control and monitor their workforce. Representatives who had difficulty using the system were given intense training and monitored very closely.

By late 2005 the MD had fully embraced the Internet and began to ask the Developer about how the Meridian system could be moved on-line. At this stage there were over 100 users of the CRM system - most of whom were remote and most using partial replica sets. The suite of reports being produced for the client had escalated to a level where over 195 different reports were being produced on a monthly basis.

Information Technology for the SME

As the argument between the Technological Determinists and the Social Constructivists continues, Pollock and Williams (2009) point out that according to social accounts organisations like SAP, and their products, should not and cannot work. In contrast we see in the case of the UK NHS the determinist promise of technology once again falling short. For the SME different pressures apply. Supporting the status quo is not necessarily the objective of the owner manager where serving the needs of the customer dominate and sometimes to the detriment of the organisation (Drucker 1989). Different forces are at play: survival and, maybe, growth.

The English philosopher Gilbert Ryle introduces the idea of a *Rich Description,* (Ryle, 1971) which is later expanded upon by the anthropologist Clifford (Geertz, 1973) , which to gain a holistic understanding of a situation. The example explained by Ryle is that of the sculpture of Rodin's thinker. Ryle asks 'What is the thinker (Le Penseur) doing?' In both the case studies we follow Ryle's and Geertz ideas of the

Rich Description and how by genuinely seeking to understand the reason for a human action. Our example of appreciating, and adapting to, the difficulties of using a foreign language resulted in reaching a rich description of the requirements of the system.

According to the Chinese philosopher Confucius we become wise by three methods; first by reflection, second by imitation, and third by experience. Confucius, demonstrating his wisdom, provides further insights - reflection being the noble way of becoming wise; imitation being the easiest; and experience the bitterest. We now share our reflections of the case study examples, with the intention of supporting the reader in achieving wisdom without tasting the bitterness of experience.

Reflexive Discussion from Practice

Following on from a comparison of the two case studies, it is apparent that communication forms the basis of a mutually assured development process. The language used by the participants must change over time to achieve a commonality of goals. The participants in these cases formed a cooperative relationship forged in the furnace of a common intention to succeed. This cooperative relationship can be construed as legitimate peripheral participation (Wenger 1998), where those outside of the practice i.e. the owner of Wessington, become experienced members of the 'community' of system makers. Without that common interest it is doubtful that these projects would have succeeded as they show many of the risk factors which can lead to failure (in bold).

* Lack of top management commitment to the project
* **Misunderstanding the user requirements**
* Not managing change properly
* **Failure to gain user commitment**
* **Lack of adequate user involvement**
* Conflict between user departments
* **Changing scope and objectives**
* Number of organizational units involved
* **Failure to manage end-user expectations**
* **Unclear / misunderstood scope and objectives**
* **Improper definitions of roles and responsibilities**

* Lack of frozen requirements
* Introduction of new technology
* Lack of effective project management skills
* Lack of effective project management methodology
* Lack of required team knowledge / skills
* Insufficient / inappropriate staffing

From Al-Ahmad, W et al. (2009) A Taxonomy of an IT Project Failure: Root Causes
The importance of the relationship between the client and developer cannot be understated. To contrast the two case studies, Meridians client/developer relationship was complicated by the fact that the client was liaising with the developer through the MD. Because of this there was a layer of translation that facilitated rather than hindered communication. In Wessington, the client/developer relationship was personal through marriage. The developer was experienced and had some prior exposure to the business. The developer and analyst had to modify the way they communicated, as the client had no prior experience of commissioning an IT project. The client had to explain what he wanted in terms that the developer/analyst could understand. When client and developer meet they will communicate on several levels, it is impossible not to communicate it is part of our behaviour as human beings (Condon & Yousef, 2002). When confronting a situation where neither party knows what they want It is sometimes necessary to tease out the unspoken subtext which goes with tacit knowledge in order to interpret and gather system requirements. For instance someone may agree fervently with you in order to placate but at the same time disagree in the body language displayed. It is important to be cognisant of the subtext when politics comes into play, there may be hidden agendas directing the course of the meeting. The degree of causal ambiguity can be a limiting factor in the transfer of knowledge between two parties. When it is low and the causes of a businesses competitive advantage are obvious then the information can be trusted, when it is high there will be underlying assumptions made on both parties, which do not reflect the actual situation (Szulanski 2003). It is not only necessary to align intention to succeed there must be sufficient transfer of knowledge to make that knowledge trustworthy and to ensure that any assumptions made are accurate.

Conclusion and further research

For practice and for further research - reflection on the case studies reveals that the unconscious process that took place in understanding what was meant by 'can you build me a system mate' and also the 'will we be able to communicate in German even though it is not your language' reveals that this search for understanding acts as an enabler for the successful conception, design, and implementation of an information system. For practitioners, the learning is to resist the temptation to start to make systems and software and instead approach a problem from a perspective of 'not until I fully understand, and I can explain to the commissioner that I understand does the actual development take place'. For the commissioners the lessons learned, by giving space and time, and a commitment to explain, understand, and re explain, (for the overall good of the project) then the likelihood of the project being a success is greatly enhanced.

For educators the idea of threshold concepts was introduced by Mayer and Land (2003) and attempt to explain that certain concepts that were held to be central to the mastery of their subjects. These threshold concepts would be the keys to the door of a discipline or subject and once understood by the learner would provide access to the discipline. What are the threshold concepts for those of us involved in IT and Information systems education? Do we concentrate on the subject specifics or do we make a concerted effort to support students with skills of understanding communication? In both these cases the concerted effort to both understand and seek to understand meant for a successful outcomes.

References.

Al-Ahmad, W., K. Al-Fagih, K. Khanfar, K. Alsamara, S. Abuleil, and H. Abu-Salem. 2009. A Taxonomy of an IT Project Failure: Root Causes. International Management Review 5, no. 1, (January 1): 93-104,106.

Condon, J., C., and Yousef, F., S., (2003) Managing Knowledge Chapter :Communication Perspectives 215

Drucker, P. F. (1989) The practice of management. Heinemann Professional.

Geertz, C. (1973) The interpretation of cultures; selected essays. New York,, Basic Books

Meyer, J.H.F. and Land, R. (2003), Threshold concepts and troublesome knowledge: linkages to ways of thinking and practising,
In: Rust, C. (ed.), Improving Student Learning − Theory and Practice Ten Years On. Oxford: Oxford Centre for Staff and Learning Development (OCSLD), pp 412−424.Ryle, G. (1971) Collected papers. New York,, Barnes & Noble.

Pollock, N., and Williams, R., (2009) The sociology of a market analysis tool: How industry analysts sort vendors and organize markets. Inf. Organ. 19, 2 (April 2009), 129-151.

 Szulanski, G., (2003), Sticky Knowledge, Barriers to knowing the firm : 63

Wenger, E. (1998) Communities of practice : learning, meaning, and identity. Learning in doing.Cambridge, Cambridge University Press.