THE 'SOFT' SIDE OF SOFTWARE ENGINEERING

Dr. David J. Skyrme

Abstract

Many organisations are failing to achieve the benefits they expected from their investment in information systems. At the heart of the problem is lack of attention to human and organisational factors (HOF) - the 'soft' issues. This paper describes how to tackle some these issues, by elaborating on the initials of 'SOFT' as Social, Organisational, Flexibility and T-skills.

1.0 GOOD NEWS AND BAD NEWS

We have all witnessed the remarkable inroads of information technology into daily life - reservation systems, customer ordering systems, automatic teller machines etc. Many businesses today would soon be crippled without their 'mission critical' systems. Other businesses have gain market share and even generated new businesses through innovative use of technology - home banking is just one such example.

On the other hand, every week sees some 'bad press' about IT. Here are some recent news and survey headlines:

- "New system abandoned after £10m investment"

- "Over 30% of systems projects fail to meet expectations"

- "Only 11% of organisations are successful with their IT, according to our objective measures"

There are estimates that over £200m of investment in software is wasted every year in the UK. This situation gives grave cause for concern for all those involved with proposing, designing and implementing IT systems, not least because senior business managers may well be looking for scapegoats to blame!

2. THE PROBLEMS

Not only do systems have to be well engineered (a difficult enough task by itself), but they must do so within acceptable timescales and budgets, and meet or exceed the expectations of users and business managers. There are many reasons why software projects fail to meet expectations. Some of the commonly cited causes are :

- inadequate definition of requirements
- inadequate methods
- poor estimation of resources
- poor project management
- problems with new technology/software
- change of business requirements over duration of project

However, software projects are just part of the wider context of IT and business integration. There is now clear evidence that lack of attention to human and organisational factors (HOF) is the most common cause of problems. Thus A.T. Kearney in their 'Barriers-2' report cites: "human and organisational issues, not technology, are the main barriers to success in IT" [1].

This position is supported by the in-depth research of the 5-year MIT 'Management in the 1990s' (MITN) programme. A strong human and organisational theme runs through the final report's main conclusions [2] which include:

- IT is fundamentally changing ways in which work is done, especially information based work and 'knowledge work';
- Successful application of IT will require changes in management and organisation structures;
- Organisational transformations will be necessary to prosper in the globally competitive environment of the 1990s.

Two other recent research projects shed some light on specific practices that affect the success of IT systems. The first project [3] investigated the linkages between the IS function and business units. Some of the factors associated with successful introduction of systems were:

- the use of more mechanisms to get co-operation between business people and IT specialists, including multi-discipline teams, collocation and job rotation;
- the development of 'hybrid managers', people with business and IT knowledge as well as general management skills;
- projects oriented to 'business improvement', where the IT system is viewed as part of a wider business system.

The second project 'SOFTCASE', is within the DTI/SERC Information Engineering Advanced Technology Programme (IEATP). It studied the use of methods (systems design methodologies) by interviewing method designers, analysts, and end-users [4]. Here are some of the findings from three different perspectives:

From the user/business perspective:

- users did not feel consulted about their work needs, even when this supposedly happened;
- business managers asked to approve project stages, often felt intimidated by data flow diagrams or weighty specifications;
- systems were often difficult to use, and training on them was often inadequate and underfunded;
- sometimes the user representative became so involved with the project team, that they became 'distant' from the real users.

From the analyst's perspective:

- many organisations get the systems they deserve: business objectives are unclear, ownership by the business is low;
- user project managers can act as a barrier between them and the ultimate end-users;
- analysts operate under a number of difficult constraints, in particular time and resources: they are therefore making many systems decisions and trade-offs during the course of a project;
- being trained to use methods is no substitute for real experience;

From the perspective of methods and the attention given to HOF:

- company standards often force the use of specific methods (e.g. SSADM), but in practice analysts are very pragmatic in how they use them: "dipping in and out as appropriate";

- no single method covered all phases of the life cycle, or both content (what to do) and process (how to do it);

- all mainstream methods were of a functional, technical orientation with little consideration of HOF:

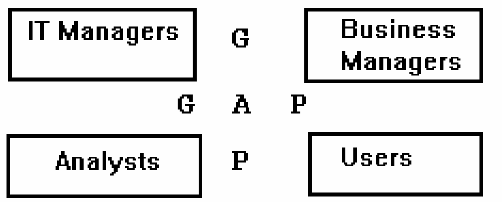
- methods are often not used as their designers intended: for example, there is often little iteration between stages;

- user-centred HOF-oriented methods do exist (e.g. ETHICS, HUSAT) but their commercial uptake is very low.

Taken together, these findings vividly portray some fundamental gaps between:

- the IS organisation and the business
- users and analysts
- business managers and their users
- the theory and practice of methods
- business needs and delivered systems

Fig 1. - Some of the Gaps



They are gaps of communication, understanding and expectations, knowledge and skills, attitudes and philosophy. Let us now consider these further.

3. THE CULTURE GAP AND ALTERNATIVE PARADIGMS

The fact that the gaps are widespread suggests that many of the causes are deep seated. In fact, the differences in technologist and business stereotypes (Table 1) is seen by many as a culture gap.

Table 1 - The Culture Gap between IS and Business

	IS	Business
Perceptions	Mid-long term Explicit in detail	Short-term (yesterday!) Needs emergent, implicit
Methods	Planned Procedural Formal	Intuitive Appropriate Informal
View of System	Elegant 1-off Precise Integrated	Fulfil Needs Ongoing Flexible Autonomous
Psychology	Logical/rational Analytical	Intuitive Action oriented

The left-hand column epitomizes the engineering paradigm. It has served society well by providing reliable, usable and cost-effective tools for use in our daily lives. As a consequence of an engineering approach, systems methods have helped make computer systems that are more reliable and maintainable.

However, today's business and technological environment is vastly different from that when the systems development methods most widely used today were developed:

- More intense global competition
- More complexity and turbulence
- Customer and quality focus
- Need for organisational flexibility and speed of response
- Heightened social and environmental awareness
- Automatic data entry devices (e.g. ATMs)
- Proliferation of personal and portable PCs
- Distributed computing
- Windows and Mouses
- OOPS

How does this environment affect the nature of systems and their development? Some of the characteristics are:

- new business requirements continually emerging
- systems more evolutionary in nature rather than frozen
- users define procedural steps more than the computer
- need for higher flexibility and coping with unexpected changes
- closer in nature to tool sets than pre-programmed machines

Thus, the traditional engineering methods must be adapted to cope with this new situation. Some of the variants to consider are:

- the use of 'spiral' development methods rather than 'waterfall' ones [5]
- increased use of prototyping [6]
- introducing facets of other paradigms.

Examples of alternative paradigms that can provide improvements are:

1. The research paradigm with its heavy emphasis on the exploration of new ideas;
2. Concurrent engineering, which involves people from all parts of the organisation throughout the systems evolution process;
3. An organisation development (OD) paradigm, where the introduction of a system is viewed as a an organisational change.

The power of an alternative paradigm as an approach to systems development is best illustrated by the following example.

4. EXAMPLE OF AN OD APPROACH TO SYSTEMS DEVELOPMENT

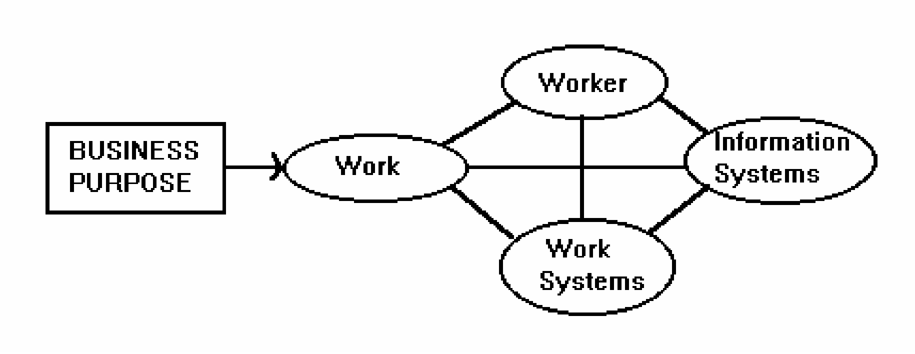
This example is about one of Digital's customer administration systems. There were significant business pressures to overhaul the total order to delivery cycle which involved eight different computer systems. This fragmentation caused unwanted delays and resulted in low perceived customer satisfaction. As the business grew and the product range became more varied, it always seemed that the computer systems lagged behind business needs.

The approach adopted was based on the principles of high performance work systems which had successfully been introduced into Digital's Ayr manufacturing plant [7]. Traditional work flow analysis suggested ways of improving productivity by cutting out unnecessary work and removing inter-departmental barriers. The approach adopted was a systemic one where several factors were considered from a 'total systems' perspective (Fig 2).

Over several months the facilitated OD process:

- clarified the business purpose to "turning orders into payable invoices";
- empowered the administrators with 'cradle to grave' responsibility for an order;
- described work, separating core from non-core;
- investigated culture and beliefs;
- designed the organisational 'system'.

Fig 2 - System View of Administrative System



At this point the legacy of existing systems became the main stumbling block. The IS systems team proposed a traditional IT solution, but the time to design and implement would be too lengthy. In the event, some internally developed software, affectionately known as 'Jabberwocky', came to the rescue. It provided a common user interface and a way of integrating the information flows between the different systems. Only four specific systems changes, taking person-weeks rather than person-years were required.

The close cooperation between end-users, IT experts and human resource facilitators resulted in a ready adoption of new work systems with significant measurable improvements:

- 40% reduction in administrative staff
- One third the number of managers
- Doubled throughput
- Cost and space savings

It was highly unlikely that a traditional 'engineering' approach, by itself, would have achieved benefits on such a scale. The users effectively designed the work system themselves using prototyping tools and expert help as required. There were also quality improvements, a better working atmosphere, but most important - increased customer satisfaction.

5. S = SOCIAL

One way to consider the 'soft' aspects of software engineering is by expanding its component letters - S O F T. The first letter emphasises the all important social factor.

Look around any office and you will see a lot of social interaction. This affects the design of systems in several ways:

- the resultant systems must take account of the effects on organisation structures, people job's and roles, and their psychological needs;

- users need to be actively involved in the design of systems that affect the way they will work;

- developers should recognise that users may not see systems in such a benign way as their developers; in the extreme, they could be perceived as instruments of management oppression;

What happens when such considerations are ignored? In one case researched by MITN, the introduction of a new system led to a 300% increase in labour turnover, very costly for the employer. The reason was that the users considered the new systems deskilled their jobs. It could be construed that the designers had considered only the technical system and that they had taken a 'machine' view of the organisation. There are other equally valid views of organisations. Some interesting ones are described in Gareth Morgan's book 'Images of Organization' [8] - they include organisations as organisms, brains and even psychic prisons!

Taking account of the social factors requires design of the overall business system, using the principles of sociotechnical design [9]. Some of the aspects that need to be given more attention by designers are the following.

Allocation of Function

One of the trade-offs to be made in any system design determining the allocation of function between human and computer. Computers are generally better at more predictable, routine and repetitive functions, while humans are better at creative thinking, coping with the unexpected, and even pattern recognition. There are economic considerations, too. One computer centre manager recently told me that 60% of his programming effort went into coding for 'exceptions', those paths in a transaction flow that happen only occasionally. Why not, when these occur, accept that the terminal operator has some skill (even if it's simply knowing to call the manager for help) and perform that activity manually? Therefore, more systematic checking of each function against the human-machine criteria should be carried out during system design..

Job Design

Related to the above are considerations about the quality of a user's job. Too much repetition leads to boredom and scope for human error. Social scientists many years ago identified the characteristics of a 'good' job. Unfortunately, systems designers often ignore them. Some of the characteristics of a 'good job' are:

- active involvement (not passive monitoring)
- some mental activity, but not too intensive for extended periods
- sense of completeness
- perception of a degree of control over how performed
- relatively high degree of autonomy
- variety
- receive feedback relative to goals
- social contact
- opportunity to learn and develop new skills

Remember, too, that individuals are different in their needs and are at different points on the learning/familiarity curve. This requires a certain amount of adaptability and ability to 'personalise' (tailor) the system. Also consider a work-group, as well as an individual, viewpoint. The total task set of a work-group can be combined in different ways to create jobs. Therefore do think of assigning system functions to roles rather than individuals.

6. O = ORGANISATIONAL

The impacts of IT on organisations are widespread yet often subtle. For example, easy access to information can erode the power base of those who have formerly had control over its flow. Here are some organisational factors to consider as early as possible in a project.

Culture and Management Style

Culture and style can be measured in many ways. Here are some dimensions that can be used to evaluate the existing situation:

- degree of openness and trust
- care for people as individuals
- delegation of authority for decision making
- amount of bureaucratic interference with work activities

- sources of power
- channels of information: how varied, how free flowing
- level of cooperation (vs. competition) between departments

Each of these dimensions - and this is just a simplified list - has an implication for how a system is designed and implemented. The systems design may need to be consistent with these factors. Alternatively, as is often the case, the system will be used to change such attitudes and behaviours, in which case the extent of change can be pre-determined and the appropriate change management programme planned for system implementation. Too often, otherwise functionally excellent computer systems fail at this hurdle.

Structures

In a short survey conducted last year, nearly every organisation had made major structural changes during the previous two years. Further, most believed that such changes would be even more frequent in the future. It also turned out that computer systems were often a millstone impeding the rate at which the new changes could become effective.

Systems are too often built around existing organisation structures - and even the forthright, but often short-term views, of powerful personalities (many of whom move on before complete system implementation!). This raises the need to develop 'organisation change proof' systems. Relational databases help, allowing aggregation at different levels and reporting along a number of dimensions e.g. country, product, industry, customers, geography.

Alternative Organisational Scenarios

Many plans prescribe a single approach, usually based on experience and precedent. However, any organisation 'system' can be designed in more than one way. It is possible to construct alternative organisational scenarios, such as:

1. Operations will be carried out by semi-skilled operators supported by highly skilled specialists;
2. Operations will be carried out by skilled operators, who support each other.

These fundamentally different ways of organising work require totally different computer systems, levels of training, development and ongoing costs, and result in different levels of motivation, productivity and degree of flexibility. Systems designers should develop 'organisational prototypes' that allow evaluation of alternative scenarios against key business, technical and organisational criteria. Such prototypes make the implications of alternatives clearer, as well as providing a way of testing assumptions with business management. It is they, after all, who should be confronted with alternatives and made to choose between them.

7. F = FLEXIBILITY

As much as 70% of programming effort goes into maintenance. Little of this concerns 'bug' fixing. Mostly it is for 'minor' enhancements to cope with changes, such as :

- business environment changes e.g. new legislation
- technology e.g. obsolescence
- delivered system did not meet users' expectations
- business changes e.g. sales volumes, new products and markets
- changes in organisation factors - structure , policy

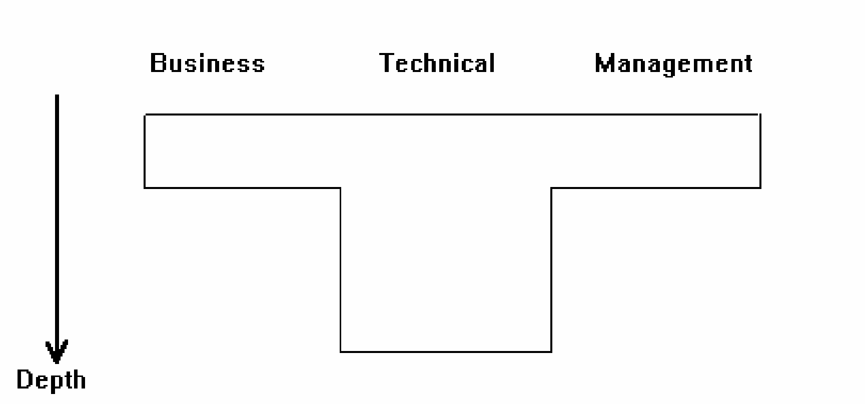
Fitzgerald [10] found that over 70% of the causes were business and organisational. He also noted that formal methodologies did little to address the problem - they generally assume a stable situation. Many a business manager has lamented "I didn't know we had made a design decision that limits us in this way".

What is needed is more attention to system flexibility, to cope with unexpected changes. This requires flexibility analysis during design and the use of more flexible and adaptive systems. The way that modern systems are currently layered into applications platforms, database, user interface etc. should help. While the design limits of systems may be very conscious during the design phase, they tend to be forgotten about over time, until the business runs into a systems problem. Therefore, in these dynamically changing times, more attention needs to be given to identifying the flexibility parameters of systems, and making them continually visible to managers, together with the costs and time to increase them.

8. T = T-SKILLS

In order to bridge some of the cultural gaps shown earlier, a range of skills are needed in project teams. Technical depth needs to be combined with a breadth of knowledge and skill in business and management. When plotted on a skill profile map (Figure 3) this generates a T-profile, hence the term T-skills. Contrast this to the technical specialist whose profile is an 'I'. People with T-skills are also one type of 'hybrid manager', for which the BCS has defined a career stream in its Professional Development Scheme.

Figure 3 - A Skills Profile map



A good T-person will typically have the following blend of knowledge and skill:

- detailed knowledge and experience of a business or IT discipline
- competence at the other discipline
- organisational knowledge
- general management and 'soft' skills

The level of competence in the secondary discipline needs to be several years experience for a fully competent hybrid. This usually means that such a person has spent some time in their career doing an assignment or secondment in another function. Experience working on multi-disciplined project teams, located together, also helps.

The key differentiator between more successful and less successful hybrids are the management skills. Particularly important are the abilities to communicate, negotiate, set and manage expectations, and to build relationships between IS and business. It also helps if you are of an appropriate personality profile, extrovert in nature.

Certainly every project team would benefit by having one or more people with T-skill profiles. Their presence has been shown to improve the success rate of projects, not just in terms of completion on time to budget, but in meeting the needs and expectations of the business.

9. CONCLUSIONS

This paper has shown the importance of 'soft' side of software engineering. Several 'SOFT' factors were introduced in the second half of the paper. As computer systems become more pivotal in the work-place, and organisations need to adapt more quickly to change, these factors become increasingly important.

Introducing 'soft' approaches into established practices is surprisingly hard. For most organisations it poses a major challenge of changing methods and attitudes. It is unrealistic that professional systems engineers should carry this burden alone. Mastery of engineering practice is itself a lifetime job, without adding a vast repertoire of organisational behaviour and social science skills. However, it is realistic for engineers to do the following:

- take a systemic view of systems development, considering alternative organisational and social perspectivesand the overall sociotechnical system, not just the technical one;
- develop a broad appreciation of the SOFT issues;
- get exposure to the knowledge, methods and skills used by social scientists and organisation behavioural specialists;
- use facilitated workshops for eliciting business and user needs;
- seek expert assistance and support from a human resources colleague; make them part of your engineering team.

Taking the first step is often the most difficult, but my personal experiences in multi-disciplinary teams that included business managers, IT specialists and human resource consultants have been very rewarding.

REFERENCES

[1] A.T. Kearney Ltd (1991) "Breaking the Barriers: IT Effectiveness - A Management Perspective".

[2] ed. Scott Morton, M. (1991) "The Corporation of the 1990s" Oxford University Press.

[3] Skyrme, D.J. (1992) "From Hybrids to Bridging", Research and Discussion Paper RDP 92/1, Oxford Institute of Information Management, Templeton College, Oxford.

[4] Clegg, C., Maclaren, R. (1991) et al., "Systems Design Methods - The Human Dimension", Digital.

[5] Boehm, B.W. (May 1988) "A Spiral Model of Software Development", IEEE Computer, pp61-72.

[6] Smith, M. F. (1991) "Software Prototyping: Adoption, Practice and Management, McGraw-Hill.

[7] Buchanan, D.A. and McCalman, J. (1989), "High Performance Work Systems: The Digital Experience", Routledge.

[8] Morgan, G. (1986) "Images of Organization", Sage Publications.

[9] Mumford, E. (1988) "Sociotechnical Systems Design: Evolving Theory and Practice", Chapter 2 in "Computers and Democracy: A Scandinavian Challenge", ed. Bjerknes G. et. al., Avebury .

[10] Fitzgerald, G. (1990) "Achieving Flexible Information Systems: The Case for Improved Analysis", Journal of Information Technology, 5, pp 5-11.