Human, Social and Organisational influences on Component-Based Software Engineering

Douglas Kunda and Laurence Brooks
Department of Computer Science, University of York, UK
Douglas@cs.york.ac.uk

Abstract

This paper discusses some human, social and organisational issues affecting Component based software engineering (CBSE) processes and the introduction of CBSE in organisations. We present some of the non-technical problems we have identified from literature and case studies. In each case we suggest some actions that developers and project managers should consider in order to alleviate these problems.

1. Introduction

Building systems from pre-fabricated software components, also known as Component-Based Systems Engineering (CBSE), changes the focus of software engineering from one of system specification and construction to one of component: identification, qualification, adaptation, integration and upgrade (for system evolution). The CBSE approach relies on the existence of an inventory of existing software components, the emergence of component integration technologies such as Common Object Request Broker Architecture (CORBA) and Component Object Model (COM), and the development of organisational capabilities for CBSE trade-off analysis and design. Growing capabilities in each of these areas is encouraging the migration towards COTS-based systems in a broad range of domains. CBSE can potentially be used to reduce software development and maintenance costs, as well as reducing software development time by bringing the system to markets as early as possible [2][6] . CBSE also improves reuse across programs and promotes a competitive component marketplace.

Software systems do not exist in isolation they are used in social and organisational contexts. Experience and many studies show that the major cause of most software failures is the people rather than technical issues [3][9]. Even with the availability of a wide range of advanced software development methodologies, techniques and tools, serious problems with software are still being faced. It is the people and culture of the organisation that determines how any system is used. For example poor training may result in people not co-operating with the information system leading to failure and project abandonment [1].

Curtis et al, [3] has highlighted that human, social and organisational considerations affect software processes and introduction of software technology. Le Quesne, [3] agreed that certain aspects of the design of information systems would make its likely success dependent on characteristics of the particular organisation environment. Friedman and Kahn Jr., [4] give two examples of computer systems that passed technical muster, but posed ethical concerns or made little sense for the social context of their use.

This paper presents some thoughts on the organisational issues to be considered when developing software application using CSBE approaches.

2. Social and organisational factors

Studies by Le Quesne, Grudin and others [5][7] have identified a number of social and organisation issues that affect the software systems and these can be best summarised in the three level behavioural model by Bill Curtis, et al [3]. Mullins adds the environmental level to the individual level, group level and organisational level identified by Curtis [10]. There are some factors that overlap at different levels and the figure 1 describes the model of the factors that affect software systems.

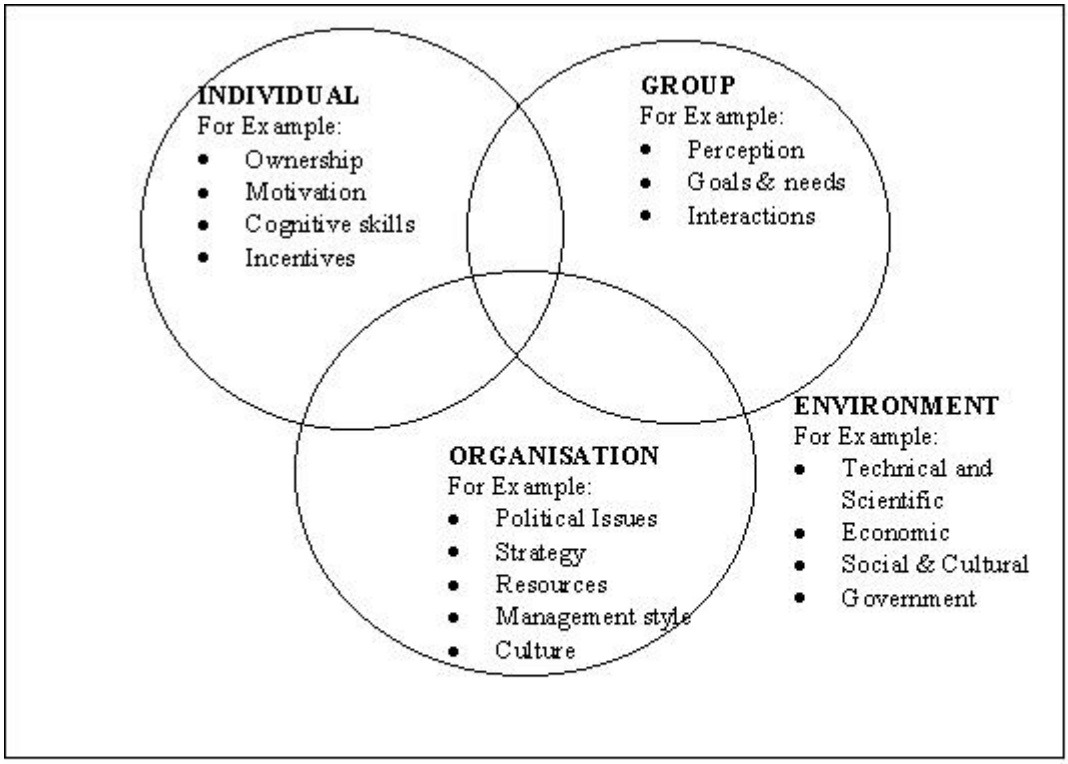

Figure 1: Organisational factors affecting software systems.

Individual behavioural factors, organisations are made up of individual members who may be stakeholders to the system being developed. Where the needs of the individual and the demands of the organisation are incompatible, this can result in frustration and conflict. The following are some of factors that should be addressed by developers and project managers during software development at individual level.
End user ownership,

Motivation and gradient of enthusiasm ,
Cognitive and education,
Incentives.

Group behavioural factors, individuals within organisation belong to one or more groups. Informal groups arise from the social needs of people within organisation. People in groups influence each other in many ways. The following are some of factors that should be addressed by developers and project managers during software development at the group level:
different perception,
different goals, and
Interactions and communication,

Organisational behavioural factors, individuals and groups interact within the structure of the formal organisation. However people sometimes relate in an informal manner and this can impact the success of the software system. The following are some of factors that should be addressed by developers and project managers during software development at organisational level:
Political issues,
Organisational and business strategy,
Organisational resources and support,
Organisational setting and management style, and
Organisational culture.

External environmental factors, the organisation functions as part of the broader external environment of which it is part. The environment affects the organisation through for example, technological and scientific development, economic activity, social and cultural influences and governmental actions [10]. This in turn will affect the software development.

3. Thoughts about actions to tackle organisational issues

A lot of effort has been made by researchers and software developers to address these organisational issues, for instance a phased development strategy. In this strategy the product idea is identified and product objectives are determined, and then at some point during the process, a portion of the product is selected and developed first before developing other portions. Incremental and evolutionary are two examples of phased development strategy [11]. The advantages are that the selected portion is delivered early, increased user acceptance, adaptation. However this approach does not solve other organisations issues such as organisational culture. Table 1 below gives a summary of the organisational problems and the proposed solution.

Table 1: Organisational factors affecting CBSE success

Organisational issue	***Problem description***	***Possible solution***
Customer ownership	Customer resistance because the user does not identify with the software system, that it belonging to them.	User participation in the software process can give a sense of user ownership.

Motivation and gradient of enthusiasm	These are factors or circumstances that induces a person to act in a particular way, they are explained in terms of the person's drives and needs. For example it is difficult to convince developers used to structured programming to migrate to CBSE, there is a threat to professionalism.	Educating stakeholders on the benefits of CBSE and training of developers in CBSE methods can help.
Incentives	The way incentives are given to developers discourages reuse and CBSE because some developers are paid the number of lines of code they write and managers are paid based on annual outputs while CBSE may take some time before the benefits are realised.	Changing the policy on incentives and bonus.
Cognitive skills	The customer/ developer's cognitive and education skills are important and can affect the software system. For instance the skills set in most organisation is currently based on structured methodology rather than CBSE.	Identification and provision of appropriate CBSE training is an important activity.
Different perception	Individuals working together in a group may have different perceptions and bias, an example is the perception of what makes good software interface.	Bringing individual to talk to each other in order to harmonise differences.
Different goals	Individuals working together in a group may have different needs and objectives and these need to be harmonised, an example is that one individual may be thinking about retirement while the other about career advancement.	Understanding individual goals and ensuring they do not sharply differ from organisational goals
Interactions and communication	Individuals working together in a group will normally interact both formally and informally. If individuals within a group are not in harmony, the success of software system will be affected.	Informal communication should be encouraged among during CBSE development process.

Political issues	These are organisational processes or principles affecting power, authority, status, etc. Some people within organisations are more powerful than others by virtual of their positions while others by their connections with powerful people within an organisation. A good example is a junior staff who can not be disciplined because of their relationship with the chief executive.	A recognition of power differences and their causes can aid in the design and development of information systems that support the organisation, its functions and individuals within it.
Organisational and business strategy	Business environment is becoming complex and dynamic - even turbulent - leading to the need for systems with shorter lives and greater adaptability. Short-term approach can discourage CBSE since benefits of CBSE may only be seen after a longer period of time.	Encourage medium to long term strategies
Organisational resources and support	Organisations work overload, skill shortage and budgetary pressure can affect the success of software system. That is not to say that all organisations with adequate resources will always have successful software systems. However this means that it can be difficult to get management (sponsor) support.	Educate management using incremental approach and successful case studies
Organisational setting and management style	The arrangement of organisational subsystems and the accompanying division of labour and hierarchy of authority relations can inhibit CBSE success.	Re-structuring the organisation according new business strategies.
Organisational culture	Organisational culture affects systems requirements and system acceptance. Customer, developer and management resistance to change in most cases is due inherent organisational culture.	Long term education and training can contribute to the change in people's attitude and organisational culture.

Therefore in this position paper we wish to recommend a social-technical development approach

as the best method to deal with organisational issues. Social-technical development is oriented to developing both social and technical subsystems in an integrated way, so that the integrated system functions in an optimal way. According to Jirokta [8] there are several approaches that have been introduced to incorporate organisational (or social) issues, called social-technical approaches. One such approach is where the social issues could be integrated with existing requirements engineering methods. In this case, an extra level of analysis could be added that incorporates the social. This would preserve the separateness and apparent strengths of each in addressing different issues, which are to be combined subsequently in some way. Multiview is a good example of this approach [1].

Other social-technical approaches include participative design and ethnography during requirement engineering phase. Participative design involves the participants directly in the requirements engineering process. Here analysts use materials drawn from meetings between participants and designers, or from user trials of prototypes. ETHICS is good example of this approach [1]. In ethnographic designs, the social and technical are seen as thoroughly intertwined and this approach attempts to develop analytic categories from the participants themselves. Here the technical is thoroughly embedded within the social.

References

1. Avison D. E and Fitzgerald G., Information Systems Development: Methods, techniques and tools, McGraw-Hill Book Company, London, 1995
2. Clements Paul C., From Subroutines to Subsystems: Component-Based Software Development, American Programmer, V8#11, Cutter Information Corp., November 1995.
3. Curtis Bill and Krasner Herb, A field study of the software design process for large systems, Communication of the ACM, 31(11):1268-1286, November 1988
4. Friedman Batya and Kahn, Jr. Peter H., Educating Computer Scientists: Linking the social and technical, Communication of the ACM, 37(1):65-70, January 1994
5. Grudin Jonathan, Eight challenges for developers, Communications of the ACM, 37(1):93-105, January 1994
6. Haines Capt Gary, Carney David and Foreman John, Component-Based Software Development/ COTS Integration, Software Technology Review, Available WWW (online) <URL:http://www.sei.cmu.edu/str/ descriptions/CBSE_body.html>, 1997.
7. Hirschheim R. and Newman M., Information Systems and User Resistance: Theory and Practice, Computer Journal, 31(5):398-408, 1988
8. Jirotka Marina and Goguen Joseph A., editors, Requirements Engineering: social and technical issues, Academic Press Limited, London, 1994
9. Le Quesne P. N., Individual and Organisational factors and the Design of IPSEs, Computer Journal, 31(5):391-397, 1988
10. Mullins Laurie, Management and Organisational Behaviour, Pitman Publishing, London, 1996
11. Wieringa R. J., Requirements Engineering: Frameworks for understanding, John Wiley and Sons, Chichester, 1996