The Cybernetics of Software Development: An Investigation of Organizational Factors in the Successful Development of Software.

P. Kawalek, D.G.Wastell.
Informatics Process Group (IPG), Department of Computer Science,
University of Manchester, M13 9PL. Great Britain.
+44 161 275 6183; +44 161 275 6200 (fax); pk@cs.man.ac.uk

Abstract: This paper seeks to explore the view that organizational factors can have an impact upon software development. It proposes that the Viable System Model (VSM) provides a rigorous framework for an investigation of these organizational factors. The paper describes the VSM and presents two case studies of its application. These case studies provide evidence of how software development can be affected by organizational issues. On the basis of this evidence it is concluded that the VSM can be used to assist the optimal organization of software development.

Keywords: software development, software quality, cybernetics, Viable System Model.

1.0 Introduction.

As software systems have become ever more ubiquitous and critical to our daily lives, it has been widely acknowledged that the successful development of software has become increasingly difficult and more urgent. As a consequence of this, the discipline of software engineering has developed several different perspectives from which issues of software development can be addressed e.g. quality standards and awards, structured methods, process maturity (Paulk et al., 1993) and formal methods. The work described in this paper starts with a broader vantage point than many of the existing investigations of software development. We consider organizational issues such as the identity of different operational teams, their relations with each other, the sharing of resources, the ability to alert management to crises and the effective formulation of policy objectives. These represent a wide-range of issues that affect many forms of cooperative human enterprise (e.g. managing a hospital ward, running a business).

The premise underpinning this work is that these organizational factors can have an impact upon how successfully teams are able to develop software. The organization of any particular team will depend upon what it seeks to achieve and the business environment in which it seeks to achieve it. It follows from this that not all of the variables are under the control of the development team itself. They might find themselves in a changing business environment which threatens to undermine their effectiveness. These observations suggest that it can be valuable to be able to diagnose the organizational effectiveness of a software development team and to be able to adapt the way it is organized in order to address new aims or business environments.

The theoretical foundations of our work come from cybernetics. This has been dubbed “the science of effective organization” (Beer 1985; see also Wiener 1961). Fundamentally cybernetics is concerned with structures of communication and control. Our work utilises a detailed cybernetic model known as the Viable System Model (VSM) (e.g. Beer 1979). Further details of the VSM are set out below. In this paper two case studies of its use are described. The case studies concern two software development teams which faced changing business environments. The studies showed that the teams needed to adapt the way in which they organized their work in order to continue to successfully develop software in the new conditions. We conclude from the case studies that there is merit in taking an organizational perspective upon software development and that the VSM is effective as an analytical and diagnostic device.

2.0 The Viable System Model (VSM).

In a series of original and challenging books, Beer has proposed a cybernetic theory of organization (Beer, 1979; Beer, 1981; Beer, 1985). In these, he sets out a model of what he terms a *viable system,* i.e. a system (an organization of some kind) which is capable of maintaining separate existence, of surviving on its own (1979, p.113). This is the central idea in Beer's philosophy. Organisations (at whatever level) are to be seen as systems (i.e. goal directed entities made up of interacting parts, operating in an environment of some kind). The issue is: what form of internal "architecture" is required if these systems are to be viable? The present section will briefly describe the VSM, further details may be gleaned from the previously referenced texts of Beer as well as other academic writings (e.g. Espejo and Harnden, 1989; Flood and Jackson, 1991).

2.1 System and meta-system.

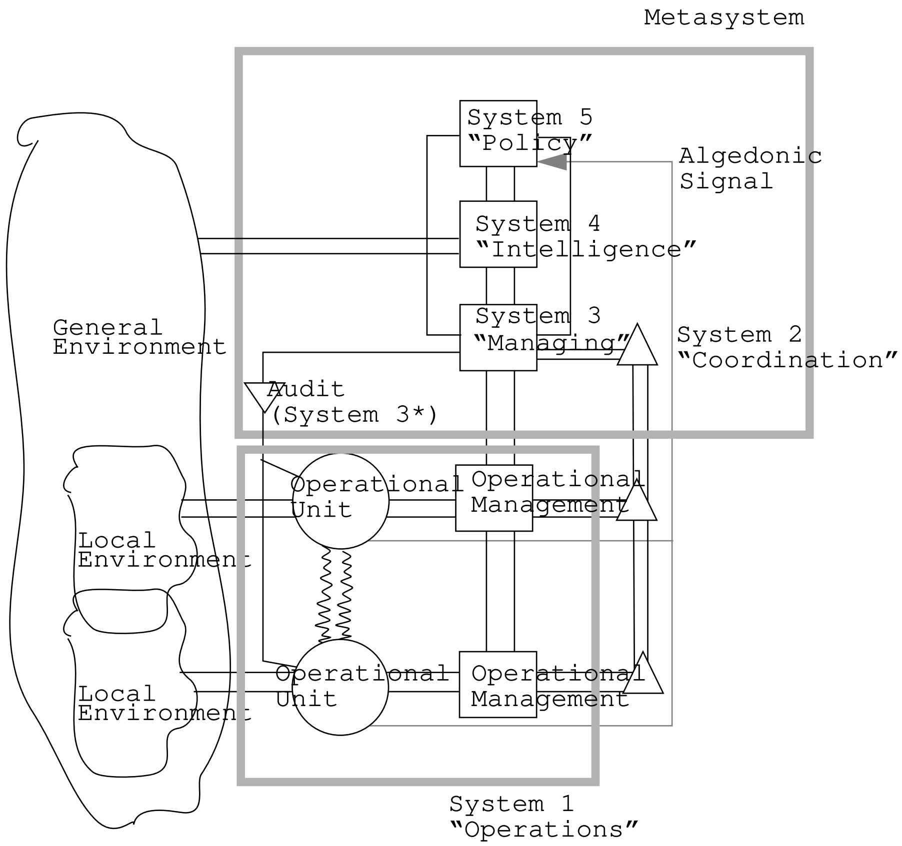
The concepts of operational system and meta-system are fundamental. All viable systems comprise these two elements: an *operational system* which performs productive work, and a *meta-system* which is the means of regulating the operational system. These concepts are recursive in that the combined structure of operational system/meta-system at one organizational level together constitute the operational system at another 'higher' level in the hierarchy (e.g. project teams organised in departments, departments nested in enterprises). At each level, though, the relationship of operational system and metasystem is invariant. Note that the use of the terms 'lower' and 'higher' does not infer a conventional, hierarchical structure of command and control. VSM proposes a distinctive view of control in organizations which emphasises the need for self-organization and localised management. Hierarchy simply denotes the referential context of a particular level of "organizational recursion", not some other notion of significance or authority.

The role of the meta-system is to manage the operational systems under its jurisdiction. This brings us to a fundamental concept in cybernetic thinking, namely the Law of Requisite Variety (Ashby, 1965). This law stipulates that the "variety" of the regulator must equal or exceed the variety of that which is being regulated (variety is defined as the number of possible states of whatever it is whose complexity we want to measure). In the terminology of modelling, the law means that an effective regulator must a) embody a model of the system under its control and b) that the model must be equal in complexity to the regulated system.

The operational system in VSM is referred to as System One. The rest of the model is made up of four further functional elements (referred to as Systems Two to Five) and the control and information loops between them. Each of these components is shown in Figure 1 and then described in more detail in the later sub-sections.

Figure 1 follows the diagrammatic conventions established by Beer. The environment is shown to the left of the diagram by the cloud like shapes. The operational teams are represented as circles. The operational management, like all management units, is represented by a square. Upright triangles denote System Two which stands outside both the System One operations and outside the metasystem. An inverted triangle represents the System Three audit. "Squiggly" vertical lines between system one elements represent informal interactions (the "grapevine"). Straight vertical lines represent formal interactions between the various elements: the two lines connecting System Three and the team leaders in System One (i.e. operational management) symbolise the regulation of System One through the resource bargain; goals/resources are propagated downwards though "variety amplifying" links whilst aggregated management information is transmitted upwards through "variety attentuating" channels.

FIGURE 1. The Components of the VSM.



2.2 System One.

System one is the basic building block of VSM. It comprises a collection of operational systems, each comprising an area of operational activity (*operational unit*) and the local management structure responsible for directly managing the operational unit (*operational management*). System One constitutes the set of these systems visible *at the current level of analysis.* This latter emphasis is absolutely crucial. VSM is a recursive model of viable organization: organizations are seen as nested arrangements of systems within systems. Given that a particular level of analysis has been chosen (the "system-in-focus" in Beer's terminology), internal details of the next level down are not visible, the next level is a "black box". For example, if our system-in-focus is a software department, we are not concerned with the internal organization of the project teams that make up the "operational systems" working within the department. These are viable systems in their own right, and are properly opaque to management at this level (this follows from Ashby's law: management at the departmental level simply does not have the detailed knowledge to intervene at the next level down). The issue at the current level of recursion is to manage relationships between the operational systems in view at this level, nor their internal workings.

2.3 System Two.

System Two provides a co-ordination service to System One. Without it, System One would be potentially unstable. For example, a number of software development projects may share resources. They might all rely upon the same highly skilled test team. In the normal course of events the sharing of this resource may be unproblematic, the individual project managers might "lose" minor delays within their specified development schedules. However, there remains the possibility of instability. The number of projects may increase or there may be coinciding urgent requirements for the use of the test facility. In such circumstances damaging effects may arise from increasing competition over the shared resource: Products may be late or not thoroughly tested.

System Two is the element which 'dampens' the instability caused by conflict between parts of System One; its sole function is anti-oscillatory. In the example above we can imagine that System Two might simply be an agreed protocol (e.g. delayed projects must hire in test resource from another part of the company) to which the operational elements will comply. Another example of System Two functionality is a process model which is used to provide a framework for cooperation within and between teams of programmers.

2.4 System Three.

The role of System Three is to steer the organization towards its current objectives. It interprets the policy decisions of higher management and maintains the operations of System One. To System Three, the internal operations of System One are opaque; perforce, the management of System One is based upon a resources-results basis.

There are three information loops in VSM between Systems One and Three. These are: legal and corporate requirements, resources and accountability. Legal and corporate requirements cover those matters that are bound into the organization through legislation (e.g. Companies Acts) or the organization's own constitution with regard to matters such as the scope of operations, safety standards and so forth. The resource and accountability loops together form what Beer calls the "resource bargain". This bargain represents a continuous process of negotiation between central and operational management about the balance of central and local control: it embodies an agreement stipulating that, in return for certain resources, System One will achieve certain goals. It is, of course, a matter for System One to "amplify" these goals into operational targets and to set up internal structures with sufficient variety to regulate performance. This is properly beyond the competence of System Three.

Having said this, there will be times when System Three will want more knowledge of how System One operations are accomplished. This special and important prerogative of System Three is known as an audit. Through this mechanism, System Three by-passes operational management in order to gain knowledge of operations directly. There are two important constraints upon the process of audit. First, it must be sporadic. Secondly, it must be used to enable more informed decision making and not to interfere in the operations and management at a lower level.

2.5 System Four.

System Four is concerned with enabling the organization to learn and adapt. It is an intelligence gathering and reporting function that seeks useful information about the environment of the current system-in-focus. It searches the environment for opportunities and threats. It provides a model of the organization and the environment. This serves as a basis upon which hypotheses are explored and changes proposed to the organization as a whole (e.g. "move into object oriented development"). The significance of System Four can be highlighted by the fact that the modern business environment is often regarded as generating an unprecedented rapidity of business and technological change (a state of affairs which is especially true of software development.) In these circumstances, the ability to perceive opportunities and threats quickly, and to foster appropriate responses, is clearly critical to the well-being of the organization.

2.6 System Five.

We have noted that System Three gives the organization the ability to steer itself towards current objectives and System Four allows it to generate alternative directions and objectives. System Five sets policy. The values and beliefs espoused through System Five should be shared with all other elements of the organization. An important part of this role is thus to arbitrate between System Three and System Four. Conflict can arise because of the different functional emphases on the *status quo* (System Three) and change (System Four).

Similarly, System Five should be open to the other elements of the viable system. It should be able to respond to all significant signals that pass through the "complexity filters" of Systems One, Two, Three and Four. A special kind of signal is the *algedonic signal* which by-passes all the normal communication channels and reaches directly from System One to System Five. In a well-functioning organization the signal will simply say that all is well, but it can also quickly alert System Five to a sudden crisis (e.g. "new product specification cannot be met.")

3.0 A Case Study of Daffodil Systems.

The first case study concerns Daffodil Systems (DS)[1], a medium-sized software development department in a large chemical company. DS consists of two teams of sixteen programmers. The first team specialises in mainframe based applications whilst the second team is responsible for PC based applications. Each team has a leader who reports to a more senior manager who oversees the whole operation of DS.

Many of the applications developed by DS are safety critical. DS had achieved a reputation for producing high quality software and had been consistently scoring highly in internal and external quality appraisals. Anecdotal evidence gathered during the study also supported the team's claim to be a high quality supplier.

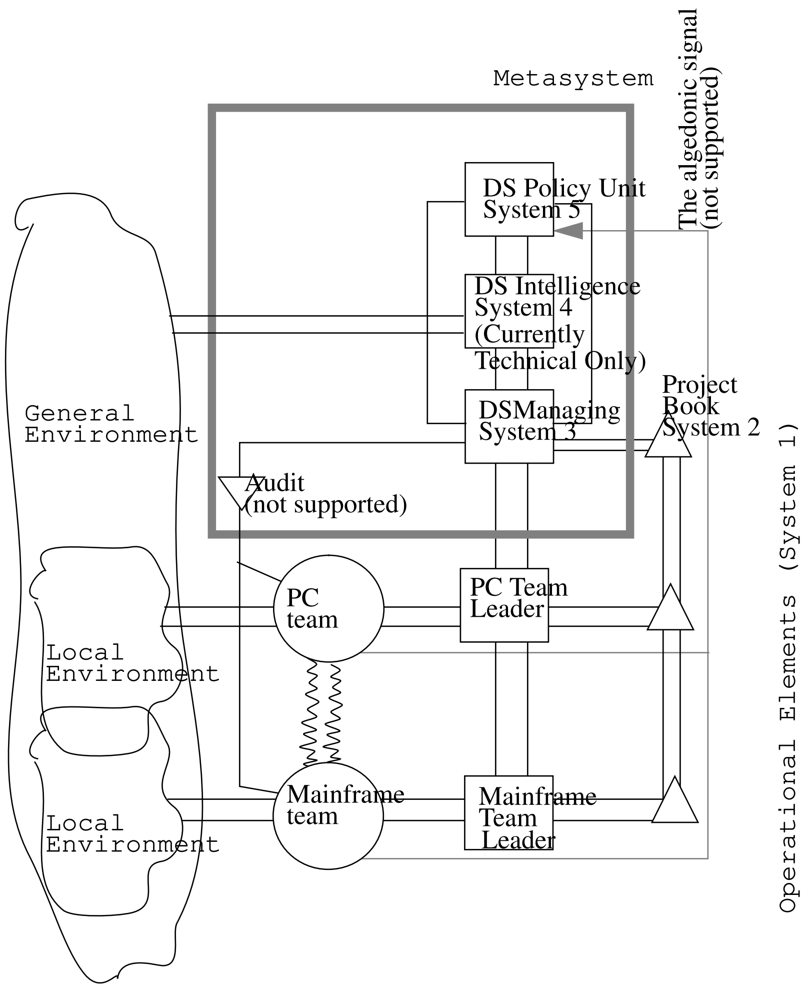
At the time of the study the department was undergoing major organizational change. Traditionally it had been entrusted with producing the vast majority of the software developed for major projects undertaken by the company. This was enforced through corporate procedures which required project leaders in the company to obtain software through DS if at all possible. These procedures were about to be changed. In the future project leaders would be allowed to buy in software services from outside the company and, as a result, DS would have to compete with outside contractors for work. Our study sought to evaluate the likely impact of these changes on the validity of the current organizational structure of DS and, in particular, to identify any threats to viability.

The VSM of DS relied on information gained through interviews with users. The aim was to be able to describe the features of DS by reference to the VSM. A diagrammatic depiction of DS was developed at the same time. Part of this is given below at Figure 2.

---

1. Daffodil Systems and Violet Computing are pseudonyms.

FIGURE 2. Level One of the VSM of Daffodil Systems Example.



Creating the model and comparing the characteristics of DS with those set out in the VSM provided a diagnostic of the department's viability. The diagnostic highlighted several features of DS's operation which were considered to be problematic. One of these posed an especially pernicious threat to the viability of DS. This is presented below.

3.1 The Inadequacy of System Two.

In the new business environment, DS was likely to be handicapped by an ineffective System Two capability. There was a need for a new approach to the shared management of resources of all kinds, but most especially the staff resource. In the past, when one group had become busier than the other, the custom had been to ask for assistance from the other. A programmer from one team might then be instructed to take on a number of tasks to help the other. Although no formal measuring was done, it is generally believed that this resource sharing balanced out. What each team gave, it got back. This practice of 'give and take' had worked well for many years; as a result, each team had come to depend upon the other to help it through times of crisis.

The practice hinged upon the use of a shared ‘project book’. This described (in a simple bar chart format) the projects that each team planned to undertake. Team leaders were experienced and highly skilled. They were able to judge with reasonable accuracy how long a project would take and when it was most likely to consume the highest amount of person hours. They would then schedule it in the project book in a way which tried to avoid the busiest periods of other projects being undertaken by either team.

Although the practice was proven over a number of years, analysis showed that it was founded upon some unsafe assumptions about the business environment. In short, it had only worked because DS was operating as a monopoly. In order to schedule projects so that they did not clash with the busiest periods of other projects, the team leaders had to be able to dictate to their clients the date on which a system would be completed. This they could do because these clients could not take their business anywhere else. It was fully expected by DS that this informal cooperation would continue to work in the new business regime where they competed with outside contractors.

The inadequacy of this System Two arrangement in the new business environment can be shown by considering a number of likely business scenarios. At the time of the study a major project to extend the chemical plant was nearing completion. As a result, the mainframe team was likely to experience a notable decline in the number of commissions they receive. As a natural defensive reflex they would be particularly keen to ensure that their customer base is kept as broad as possible by preventing the outflow of business to the software houses with whom they now have to compete. In such circumstances they would no longer be able to dictate completion dates to their clients. The project managers would gain much greater influence and be able to insist upon earlier completion. The completion of the plant extension was likely to put the PC team in exactly the same situation. However in its case the situation was likely to be further exacerbated by the fact that, in order to replace the lost work, the team might find itself competing more and more to produce general applications (i.e. non-safety critical) on PCs. The competition with external suppliers is likely to be at its most intense in this non-safety critical sector where DS will be less able to rely upon its existing reputation for high quality. Therefore the PC team will also be unable to dictate completion dates. In such circumstances it is highly likely that the cooperation between the teams over resources will be replaced by competition as both struggle to meet coinciding urgent deadlines.

4.0 A Case Study of Violet Computing.

The second case study considers Violet Computing (VC), a large operating systems development team of over fifty staff. VC have achieved a reputation for high quality, their product is well known and highly regarded in its market sector. Their efforts to engender a culture of quality amongst staff had gained recognition through a number of prestigious quality awards. However, the organization of VC had been rationalised and revised over a number of years and at the time of the study a further set of initiatives was proposed. These changes were contentious and provoked some opposition from within VC. The study sought to identify whether the changes were likely to have any implications for the ability of the organization to continue to successfully develop software.

Creating the VSM models proved to be difficult. It was clear that VC were used to surviving in a state of organizational flux and there was some difficulty in clearly defining what the implications of the proposed changes would be. Nonetheless, as the study progressed several significant issues emerged. This presentation highlights the most critical of these; the metasystemic management of technical strategy in VC.

4.1 The management of technical strategy in the metasystem of VC.

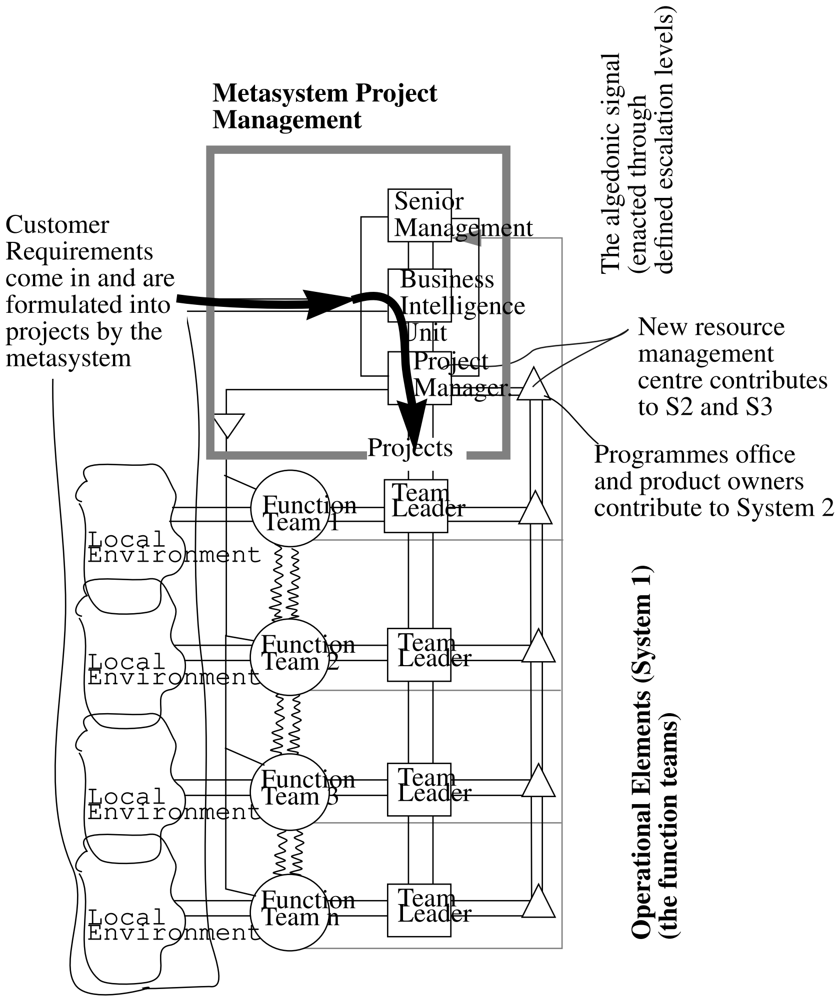
The review that VC undertook of their organizational structure was motivated by their wish to improve customer responsiveness and to control the number of product enhancements. This latter concern was especially acute. Despite the maturity of the product, even the best understood and most proven product functions continued to undergo a greater number of enhancements than had been anticipated. This was felt to endanger product quality and cost. Using the VSM we were quickly able to identify the reason why this was the case. The System One operational system is made up of a number of semi-autonomous 'function teams,' each of which is responsible for developing a defined part of the product. The teams had the power to propose enhancements to their part of the product in order to ensure that it was able to satisfy renewed technical and market expectations. It became obvious that these teams were exploiting this power to propose enhancements in order to generate work for themselves. In other words, even where there was insufficient technical or market justification, the function teams still argued that their part of the product should be enhanced. This is an example of what Beer regards as "pathological autopoiesis" (Beer, 1979; p.412). It was in the interests of the function teams to behave in this way so that they were able to secure resource from management and, ultimately, to ensure that their team continued to exist.

In order to address this problem, VC decided to reduce the power of the function teams by putting greater emphasis on a project structure of organization. This meant that in future, resource would be allocated only in response to customer requirements which were grouped and organised into defined work packages (i.e. projects). Although the function teams would continue to exist and would have sole responsibility for the technical standards within their remit, they had lost their power to argue for product enhancements on technical criteria alone.

These changes were part of the programme of organizational change that was under way as our VSM study was carried out. Although many parts of the programme were contentious, the VSM diagnostic helped to reveal their rational basis and we were generally supportive of them. However, we were concerned about the change to a project structure of organization. One implication of this would be that there would be no metasystemic ownership of technical issues. The management at this level of concern was to be focused only on market issues. As a result it would be unable to arbitrate between conflicting technical objectives at the operational level and, in the face of sudden market opportunities or threats, it would be unable to organize any radical reworking of the product. This shortcoming was felt to threaten organizational viability.

The new project arrangement is depicted as a VSM diagram at Figure 3. The bold black arrows are used to interpret the flow of customer requirements from the business environment, through the metasystem and to the operational system.

FIGURE 3. Level One of the new Structure Proposed by VC.



The rational analysis facilitated by the VSM revealed the precarious nature of this new arrangement. The ownership of technical issues was entrusted to the function teams. However none of these had a holistic view across different parts of the product and all were prohibited from proposing enhancements on technical criteria alone. Clearly then, the market would become the sole source of technical enhancements and yet, if ever these enhancements were wide-ranging, there was no mechanism to provide holistic management of them. Product quality would be endangered by this. At the same time VC would have no mechanisms for promoting enhancements on technical criteria or in anticipation of market benefit. Thus, technical management would be fragmented and reactive. A retrospective analysis of VC helped us to understand why this state of affairs had come about. It was instructive because it revealed how the team had responded to changes in the business environment over a number of years and how as change was laid on change, ultimately, the integrity of the organizational structure was compromised.

Perhaps the most prominent feature of the traditional organization of VC was the role played by a ‘technical commission.’ This commission, which was headed by a single individual, was made up of a number of team members who were selected for their technical and personal skills. The principal goal of the technical commission was to maximise product quality and integrity. In the language of the VSM, the commission dominated System Three, Four and Five functionality and to as great a degree as possible made long-term technical values preordinant. The prime responsibility of the commission was to prepare and update the functional specification for the product. This was done through the monitoring of technical possibilities and requirements (System Four), the evaluation of their potential benefit *vis a vis* current practice (Systems Three and Four) and the arbitration of conflicts between the two (System Five). Control of the functional specification certainly afforded the technical commission considerable power in the organization, however the VSM exercise suggested that the uncompromising way in which the commission were able to wield this power actually depended upon a transitory set of environmental conditions that existed at that time.

In order to ensure that their technical standards were adhered to, the commission took a strategic role in the organization by providing the interface between the business environment (i.e. customers, salesmen etc.) and the function teams (i.e. detailed designers, programmers, testers). From this vantage point the technical commission were able to work in two interdependent ways. First, they were able to provide an environment of relative stability for the function teams by insulating them from the worst fluctuations of market demands (e.g. changing requirements). This made it easier for the function teams to work to the specification and to concentrate upon achieving high quality. Secondly, they could filter out market requirements that threatened to undermine technical strategy. This was done by putting an alternative technical case to potential customers or even rejecting some demands. This latter role was crucial. It meant that as the technical commission were able to attenuate market variety from the point of view of the function teams, so too they were able to amplify the technical variety from the point of view of market. However the VSM study suggested that their ability to provide this amplification was only possible because of some temporary features of the business environment. In particular, in the market segment occupied by VC there was little competition. For many purchasers there was actually no practicable alternative to the Violet system and many of these purchasers were relatively uninformed about technical possibilities.

Today there is more competition and customers of VC have a higher level of market and technical awareness. This means that VC can no longer rely upon being able to put an alternative technical case to potential customers or to reject some of their demands. Therefore the environmental conditions within which VC seek viability are quite different to those they encountered earlier in the project. The VSM cogently describes the interconnection of environmental, managerial and operational variety and through this reveals such environmental change was likely to precipitate management and operational change within VC.

Indeed it became apparent that over a period of time the technical commission had been reduced in size and influence. Instead, greater authority was given to the function teams through the reformulation of the resource bargain between System Three and System One. The function teams were also allowed to be more directly involved with the customers of VC and, in this way, could be more responsive to market requirements. Thus, in empowering the function teams at the expense of the technical commission, VC had solved one set of problems only to find themselves in new difficulty when the “pathological autopoiesis” of the function teams became evident.

Clearly, VC needed to devise a way of ensuring that the function teams had sufficient power whilst preventing them becoming too powerful and promoting enhancements to their part of the product at the expense of the technical integrity of the product as a whole. We were able to highlight that the crux of the problem lay in the way the resource bargain was negotiated between Systems One and Three. Although the domination of this bargain by the technical commission had been untenable for some time, so too the prospect of awarding resource solely on the basis of customer requirements was a possible threat to viability. Instead we argued for a hybrid approach which would return some ownership of technical issues to the metasystem. We proposed that resource should then be distributed to the function teams either on the basis of customer requirements (the project stream) or technical issues (the technical stream). The balance between these streams should be determined by System Three acting in concert with the demands of Systems Four and Five. We further recommended that this two stream approach should be facilitated by a resource pool (a recent VC innovation, see Figure 3) acting in a System Two capacity to attach staff to function teams as needs arose. Through this, each function team could be reduced to notional status when there was no work to be done in that area. At other times, when enhancements were needed, the function team could be reactivated and staff placed within it. This design was felt to address the need for management of technical strategy whilst avoiding the pitfall of "pathological autopoiesis." Our proposals were arrived at through the use of the diagnostic capability of the VSM and were facilitated by the use of cybernetic terminology to clearly identify issues of concern. The proposals have gained considerable support amongst senior managers in VC and a further review of the organizational structure is now planned.

5.0 Analysis: The Implications for Successful Software Development.

Both DS and VC were known as high quality, successful software development teams. They had both undertaken a range of quality initiatives and reaped reward in good reputation which, in the case of VC, led to official industry recognition. However working from an organizational perspective, we were able to raise issues which it was felt would affect the efficacy of the organization's operations and, ultimately, could endanger their ability to continue to successfully develop software. Taken together, these two case studies suggest that there is merit in the further investigation of an organizational perspective on software development. One implication is that organizational issues can arise in spite of process or technological excellence when some environmental factor is acting to frustrate the best efforts of the team. This supports the view that the successful development of software will be advanced by the ability to diagnose organizational effectiveness, to take cognizance of the business environment and to correct organizational defects.

DS is a particularly acute example of this. They had hitherto operated in an environment of relative calm, with a guaranteed market. The study suggested that the proposed changes to the business environment would had wide-ranging impact upon the organization of the team. This was not generally appreciated by DS at the time of the study. Although it was anticipated that the organizational changes would cause some perturbation to their operations, it was not generally expected that there would be any implications for the System Two coordination of team resources. The study suggested that the informal give and take arrangements based around the use of a shared 'project book' would not be appropriate in the new business environment. Both teams would be forced to offer shorter completion dates to customers and so, finding themselves more rushed, would be unable to rely upon support from the other team. The effect would be to heighten the impact of the shorter time scales. The resulting pressure could mean less rigorous application of the development process, worse time management, and the hurried introduction of new practices and technology. Inevitably then, this would carry a threat to quality and, in the longer term, to the viability of the team itself.

In VC the adaptations to environmental change were less sudden than those confronted by DS but had reached a critical and contentious stage. Over a number of years these changes had undermined the traditional dominance of the technical commission and lead, via the empowerment of functional teams, to a proposed project basis for the operation. The VSM study revealed the rationality of many of the new proposals. However, it also highlighted how the shift of power from the technical commission to the function teams had pushed technical issues out of the metasystem. This was a risky state of affairs which meant that the organization would be vulnerable in a number of ways (e.g. whenever conflict arose between function teams or if ever a wide-ranging re-working of the product was needed). In order to counter this difficulty, the VSM study proposed that the need for technical overview should be balanced against the need to have flexible and adaptive development teams. This could be done by instituting a new twin-stream arrangement for the System Three to System One resource bargain, and could be facilitated by a new System Two function.

6.0 Conclusion.

This paper has introduced an organizational perspective upon issues of software quality. The general theoretical basis comes from cybernetics and from the Viable System Model in particular. The VSM constitutes a complete description of the functions (i.e. the internal architecture) that an organization must possess for it to be viable (i.e. able to survive on its own in some environment). Two case studies of the use of the VSM have been presented. DS's reputation for high quality was endangered by the profound changes taking place in their business environment. The ongoing rationalisation of VC, whilst generally assisting the viability of the team, ultimately began to raise issues in respect of the technical management of the product. It follows that these studies suggest that an organizational perspective on software development might usefully complement other perspectives (e.g. process, metrics, technology) in addressing the difficulties of producing high quality software systems. This organizational perspective allows researchers to consider how some generic human concerns (e.g. how teams are organized, how communication is facilitated) are manifest in the challenging endeavour of software development. The DS and VC studies also provide validation of the usefulness of the VSM which has proved valuable as a diagnostic device. It is proposed that the investigation can be developed by using the VSM to undertake further studies and especially those where a longitudinal perspective can be cultivated.

7.0 References.

Ashby, W.R., (1965), *Introduction to Cybernetics*, Chapman and Hall, London.

Beer, S., (1979), *Heart of Enterprise*, John Wiley & Sons, Chichester.

Beer, S., (1981), *Brain of the Firm*, Second Edition, John Wiley & Sons, Chichester.

Beer, S., (1985), *Diagnosing the System for Organisations*, John Wiley & Sons, Chichester.

Espejo, R., Harnden, R., (1989) *The Viable System Model, Interpretations and Applications of Stafford Beer's VSM,* John Wiley & Sons, Chichester.

Flood, R.L., Jackson, M.C., (1991) *Creative Problem Solving, Total Systems Intervention*, John Wiley & Sons, Chichester.

Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V., (1993), *Capability Maturity Model For Software*, Technical Report, Version 1.1, CMU/SEI-93-TR-24, Software Engineering Institute, Pittsburgh, February 1993.

Wiener, N., (1961) *Cybernetics,* Second Edition, MIT Books and John Wiley, London.