

2000

Soft Systems Methodology - A Foundation for a Systematic Approach to SPI

Peter D.C. Bennetts

Gloucester County College, pbennetts@chelt.ac.uk

A. Trevor Wood-Harper

University of Salford, a.t.wood-harper@salford.ac.uk

Follow this and additional works at: <http://aisel.aisnet.org/amcis2000>

Recommended Citation

Bennetts, Peter D.C. and Wood-Harper, A. Trevor, "Soft Systems Methodology - A Foundation for a Systematic Approach to SPI" (2000). *AMCIS 2000 Proceedings*. 369.

<http://aisel.aisnet.org/amcis2000/369>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Soft Systems Methodology - A Foundation for a Systemic Approach to SPI

Peter D.C. Bennetts, Gloucestershire Business School, Cheltenham and Gloucester College of Higher Education, Cheltenham, United Kingdom. pbennetts@chelt.ac.uk

A. Trevor Wood-Harper, Information Systems Institute, University of Salford, Salford, United Kingdom a.t.wood-harper@salford.ac.uk and School of Accounting and Information Systems, University of South Australia, Adelaide, Australia, trevor.wood-harper@unisa.edu.au.

Abstract

This paper develops a model of software process improvement, based on the systems approach and soft systems methodology and an adapted version of GQM to allow for qualitative metrics. The product characterisation from soft systems methodology causes a difficulty as it does not consider time.

Introduction and Background

Looking back at the development of computing, it seems clear that it has always been difficult to develop new systems. In the late 1960s and early 1970s, the DoD were so concerned about the perceived failure of developers to create systems to budget, time and functionality, that they convened a NATO working conference whose title introduced the concept of a "software engineer" (Naur, Randell and Buxton, 1976). In support of this "re-launch" of information systems development (ISD), many argued that more training was required, together with more mathematics and formality (Dijkstra, 1976, Hoare, 1982). While improvements have accrued, many systems are still considered to be failures, and for reasons that are not addressed by a purely technical perspective (Clegg *et al.*, 1996, Standish Group, 1995). The authors take the view that in order to improve new systems' failure rate, both technical and non-technical issues need to be explicitly addressed. This leads to a recognition of systems approaches as being appropriate methodologies to support the development of ISD. In particular, Checkland and Scholes' (1990) Soft Systems Methodology (SSM), is appropriate. Indeed, Bennetts, Wood-Harper and Mills (1999) show that SSM can act as a metaphor for the processes of ISD. The current paper develops the model of this earlier work. However, the product characteristics of SSM are seen to be inadequate as a superset of development goals as they do not include a concept of time.

Lyytinen (1988) recognised that SSM gave "a better and more complete identification of stakeholders, a deeper insight into the dynamic nature of IS problems and a careful perception of several ... problems which are often

... unnoticed". For SSM, a particular problem situation (in this case ISD) is examined through an analysis of the perceived corresponding Human Activity System (HAS), which represents both product and process issues. Software Quality Assurance is seen as an element of the monitoring and control processes of a HAS for ISD. As well as offering a means to resolve problems, SSM can be seen as a means of describing an appropriate set of activities that should be taking place in ISD i.e. as a metaphor for ISD. As part of the thrust of the software engineering approach to ISD, practitioners have tried to measure software quality using 'metrics'. The characterisation of a 'good' metric is purely quantitative (e.g. Wallmüller, 1994). However, it is possible to adapt this characterisation slightly in order to accommodate qualitative issues (Bennetts, Wood-Harper and Mills, 1999). Hence, two forms of metric are perceived ('hard' and 'soft'), depending on which characterisation is used (Bennetts, Mills and Wood-Harper, 1998). (The soft metric replaces objectivity and reliability as prime criteria for a good metric, with related criteria based on team consensus.) An element of the software quality model developed here will be the measurement of recognised characteristics of software in order to assess how closely the espoused goals of software quality have been achieved. Basili's (1995) Goal / Question / Metric (GQM) approach is often used to achieve this. However, this is defined to require quantitative responses only. The approach therefore needs adapting to include metrics which take qualitative values.

The model developed in Bennetts, Wood-Harper and Mills (1999) (and shown as Figure 1) uses the Logic Stream of SSM to develop the software product. The social or cultural stream of analysis, on the other hand, can be used to develop understanding of the new information systems' context, so that, in practice, the model can be seen as contingent.

The Contingent Factors Model

This section develops the Contingent Factors Model (CFM) of software quality assurance (SQA). The aim is to develop a comprehensive approach to software quality

through the use of success factors or goals, which is sensitive to the characterisation of software quality developed by an organisation and its development personnel. (It is accepted that it is rationalistic to even attempt to be 'comprehensive'.) Using this model, an organisation will be able to identify any changes that would enhance the quality of the software that they develop. This feedback loop is recognised in ISD as software process improvement (SPI).

Checkland's (1981) development of his Soft Systems Methodology was identified as being appropriate support for the systems approach. Subsequently, it was recognised that SSM could be used to represent the whole of the process of ISD. Consequently, the ISD process, regarded as a Human Activity System (HAS) will have its "Logic Stream" characterised by the "5Es" - Efficiency, Efficacy, Elegance, Effectiveness and Ethicality (Checkland and Scholes, 1990). Vidgen, Wood-Harper and Wood (1993) regard these characteristics as representing a Use View of software quality. Further, they regard the objective rationality of software engineering as being embedded in the socially constructed reality of the organisation. Consequently, the quality of the product derived from this objectivity is intertwined with this Use View. Again, Vidgen, Wood and Wood-Harper (1994) recognise the need for a Service View of quality. These views can be reconfigured to reflect Linstone's Multiple Viewpoints (Mitroff and Linstone, 1993) - Technical, Organisational and Personal. Further, these views also reflect Mitroff and Linstone's (1993) contention that complex situations need to be addressed from several viewpoints. As the 5Es are held to characterise any complex business situation, it is argued that it is reasonable to suppose that they characterise ISD.

The current model is developed from a systems approach, treating software production as a HAS. The 5Es of Checkland and Scholes (1990) are treated as goals within the context of Basili's Goals / Questions / Metrics (GQM) paradigm (Basili, 1995). The resulting questions are used as an instrument to identify the level of software quality currently achieved by the organisation. From the literature, it is clear that there is a growing consensus about the issues promoting high quality software. However, ethicality is not often considered in that literature. It is included here as a goal as it is a recognised characteristic of Singerian inquiring systems and SSM, both of which support the model.

The idea of making quality factors (or attributes or characteristics) contingent developed from two main areas. Firstly, each organisation is unique, with its own problems and its own attitudes. For example, Linstone's approach (Mitroff and Linstone, 1993) showed that

Organisational and Personal viewpoints were significant in developing a pertinent solution to any business problem. Secondly, Bennetts and Wood-Harper (1996) showed that even where organisations had identified their view of the nature of software quality, this was not necessarily supported by the developers, who had different goals.

The overall model is linked to the adapted GQM model of Basili through the identified characteristics of SSM's Logic Stream. Conceptually, the Logic Stream for SSM connects with GQM in two ways - through the characteristics of the Logic Stream linking with the espoused organisational goals; and through metrics, given values by being applied to the products or processes of ISD linking back to the questions and goals of GQM. For successful ISD, Technical, Organisational and Personal Viewpoints need to be identified and incorporated into the techniques and processes that take place. These can be both rationalistic and interpretative, quantitative and qualitative, so the image of contingent ISD as a HAS is replaced by the image of a corresponding process structure of contingent ISD as SSM. In this image, would-be-problem-solvers consider the current situation in the context of historical decisions, information and data. They develop the organisational culture in a way that is culturally feasible. Within the Stream of Cultural Analysis, the generic Analyses One, Two and Three (i.e. role analysis, a social analysis and a political analysis) (or alternatives, such as structuration theory or actor-network theory) will have been used to identify the options that were feasible. These decisions will include the business goals and quality factors to be addressed by a specific project in that organisation, at that time. The goals and factors identified by the practitioners, now form the framework for the adapted GQM approach. Thus, the answers to the questions representing the goals, will be addressed by metrics - both hard and soft. Consequently, there are two points of contact between the SSM and GQM elements of the model. These points of contact are - through the metrics and also through the relation between Checkland's characteristics and the identified goals. This model is summarised in Figure 2.

Discussion

Figure 2 is only schematic, as the anticipated links between the goals or characteristics and the factors or attributes, for example, are too numerous for them all to be shown. Consequently, the many-many 'crows-feet' of SSADM's Logical Data Structures is used to convey this ambiguity. Similarly, no attempt has been made to identify the links between specific factors and sub-factors, or between sub-factors and questions.

The model of Figure 2 represents the products and processes of one ISD project. In practice, the learning experience achieved through obtaining appropriate metric data, as well as debriefing users and developers, needs to be fed back into the decision making processes, to inform the development of the next project. This represents the standard SSM process of unlimited iteration. It also, hopefully, represents the sought-after incremental improvement of software quality through SPI. This may be achieved through an improvement process, based on GQM, in which project information is deposited in a repository. The data and information in this repository can be analysed and synthesised to identify potential improvements to perceived goals, models, techniques and processes. Consequently, a feedback loop, through this improvement process, is included in Figure 2, in the form of a dashed line.

Checkland and Scholes (1990) state that the 5Es are "criteria by which [the] T[ransform] would be judged" (where the transform is the change that happens to the input to create the output to a HAS). In the context of ISD, this means that the quality of the product is characterised by these 5Es. Consequently, an attempt was made to identify the links between the 5Es and the product quality characteristics in the instrument. However, this was abandoned as soon as it became clear that the SSM characteristics lacked a sense of time, so that goals such as Reliability (Is the system accurate all the time?) and Delivery (Will the system be delivered on time?) cannot be met entirely. Again, the goal of having a system which will be easy to code and implement and which will be seen to be successful (Deliverable) causes difficulty. Ease of coding depends on the problem, and success is a political issue. It is therefore concluded that, at the moment, the anticipated links cannot be identified. This is seen as a difficulty for SSM, rather than this model. In practice, this difficulty does not change the model as the level of quality was going to be identified by metrics within GQM. Note that some metrics may well have been used within the ISD processes represented by SSM's Logic Stream. However, on implementation, the characteristics of a project will be added to the repository and analysed in context. The process changes that are suggested by this analysis will reflect the organisation's current situation (in that using SSM should ensure that they are culturally feasible and systemically relevant).

Given the discussion above, Figure 2 can be recast (as Figure 3) so that the 5Es and any corresponding links have been removed (as they are not immediately usable). The connection between SSM and GQM is now being articulated through metrics rather than goals. If no metrics are being used, improvement is expected to take

place along the lines of the development CMM would require to get to the next level.

It has not been possible to test the Contingent Factors Model yet. Moreover, this model is based on both SSM and GQM, so validation and verification processes should reflect both elements. However, this will present some problems with generalisability as the model is contingent. For example, the organisation may not use any metrics, in which case, little use can be made of GQM. SSM has been refined by its use in well over one hundred interventions (Checkland, 1981). It is hoped that the model would be stabilised with fewer studies than that. Each intervention used a form of action research. It is appropriate, therefore, to consider action research to test the model. If Baskerville and Wood-Harper's (1997) characterisation of action research techniques, then it is possible to deduce which technique is appropriate, namely SSM(!).

The Contingent Factors Model has the same limitations as SSM. In particular, SSM demands a 'level playing field' politically among the stakeholders - an openness and willingness to address the issues and arrive at an agreed accommodation, without one particular person or group dominating events. The end result is only as good as the insights gained from the discussions during the process.

Conclusion

This paper has presented the CFM model of software process improvement, which was based on systems theory and SSM, as well as GQM. A problem was identified with SSM's product characterisation. The Social or Cultural Stream of SSM is extended from the standard SSM analysis to potentially include the use of structuration theory or actor-network theory. These analyses develop a rich insight into the organisational, political and social issues of the moment, which provide the milieu in which the new IS is to be developed. The Logic Stream is a representation of the process of the technical development of this information, benefiting from the greater understanding of the context as provided by the Social Stream. The achievement or otherwise, of the organisational goals is measured by appropriate hard or soft metrics. These metrics are then related through the corresponding questions to the corresponding goals.

The Contingent Factors Model combines SSM and GQM, in order to improve software quality. If we reflect on Pirsig's (1974) quote on needing to care, and the sometimes destructive input of politics, it will be clear that, without the requisite support, the process will fail. Therefore, there is an imperative need for manifestly

strong support from top management for the implementation of this model. The Contingent Factors Model is expected to improve the ISD process because, not only does it explicitly require the project leader to examine the social aspects of the current organisational situation, but also, the organisation will need to have some interest in software quality. Again, it encourages software development goals to coincide with organisational goals by focusing on goals and ensuring that the project leader is aware of the “big picture” for the organisation.

References

Basili, V.R. “Applying the Goal / Question / Metric Paradigm in the Experience Factory” In Fenton, N., Whitty, R., and Iizuka, Y. (eds.) *Software Quality Assurance and Measurement A Worldwide Perspective* International Thomson Computer Press, London, 1995, p. 23-44 (Chapter 2).

Baskerville, R.L. and Wood-Harper, T. “A Taxonomy of Action Research Methods” *Work in progress University of Salford*, 1997.

Bennetts, P.D.C. and Wood-Harper, T. “Multiple Viewpoints and Software Quality Goals” In Samson, B., Marshall, I.M., Edgar-Nevill, D.G. (eds.) *Proceedings of 5th Software Quality Conference* (Dundee, 9-10 July 1996) University of Abertay Dundee, Dundee, 1996, p. 66-75.

Bennetts, P.D.C., Mills, S. and Wood-Harper, T. “A Qualitative Approach to Organisational Goal Setting for Information Systems Development” In Hawkins, C., Ross, M. and Staples G. (eds.) *Proceedings of the Sixth International Conference on Software Quality Management, Amsterdam* Springer-Verlag, London, 1998, p. 38-49.

Bennetts, P.D.C., Wood-Harper, T. and Mills, S. “The Soft Systems Methodology as a Framework for Software Process Improvement” In *Software Process Improvement: Concepts and Practices* McGuire, Eugene (ed.) Idea Group Publishing, 1999.

Checkland, P.B. *Systems Thinking, Systems Practice* John Wiley & Sons, Chichester, 1981.

Checkland, P.B. and Scholes, J. *Soft Systems Methodology in Action* Wiley, Chichester, 1990.

Clegg, C., Axtell, C., Damodaran, L., Farbey, B., Hull, R., Lloyd-Jones, R., Nicholls, J., Sell, R., Tomlinson, C., Ainger, A. and Stewart, T. *The performance of Information Technology and the role of human and organizational factors* Report to the Economic and Social Research Council, UK, 1996, (current Dec. 10, 1998) <http://www.system-concepts.com/stds/clegg.html>.

Dijkstra, E.W. *A Discipline of Programming* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976.

Hoare, C.A.R. *Programming is an Engineering Profession* Technical Monograph PRG-27 May 1982 Oxford University Computing Laboratory, Programming Research Group, 1982.

Lyytinen, K. “Stakeholders, Information System Failures and Soft Systems Methodology: An Assessment” *Journal of Applied Systems Analysis* (15), 1988, p. 61-81.

Mitroff, I.I. and Linstone, H.A. *The Unbounded Mind - Breaking the Chain of Traditional Business Thinking* Oxford University Press, New York, 1993.

Naur, P., Randell, B. and Buxton, J.N. (eds.) *Software Engineering - Concepts and Techniques* Petrocelli / Charter, New York, 1976.

Pirsig, R.M. *Zen and the art of Motorcycle Maintenance* William Morrow & Co., New York, 1974.

Standish Group *The CHAOS Report*, The Standish Group, 1995. (Current Sept. 27, 1999) <http://www.standishgroup.com/chaos.html>.

Vidgen, R.T., Wood, J.R.G. and Wood-Harper, A.T. “Customer satisfaction: the need for multiple perspectives of information system quality” In Ross, M., Brebbia, C.A., Staples, G. and Stapleton, J. (eds.) *Proceedings of the 2nd International Conference on Software Quality Management*, Edinburgh, July 1994, p. 487-495.

Vidgen, R., Wood-Harper, T. and Wood, J.R.G. “A Soft Systems Approach to Information Systems Quality” *Scandinavian Journal of Information Systems* (5), 1993, p. 97-112.

Wallmüller, E. *Software Quality Assurance - A practical approach* Prentice-Hall International, Hemel Hempstead, 1994.

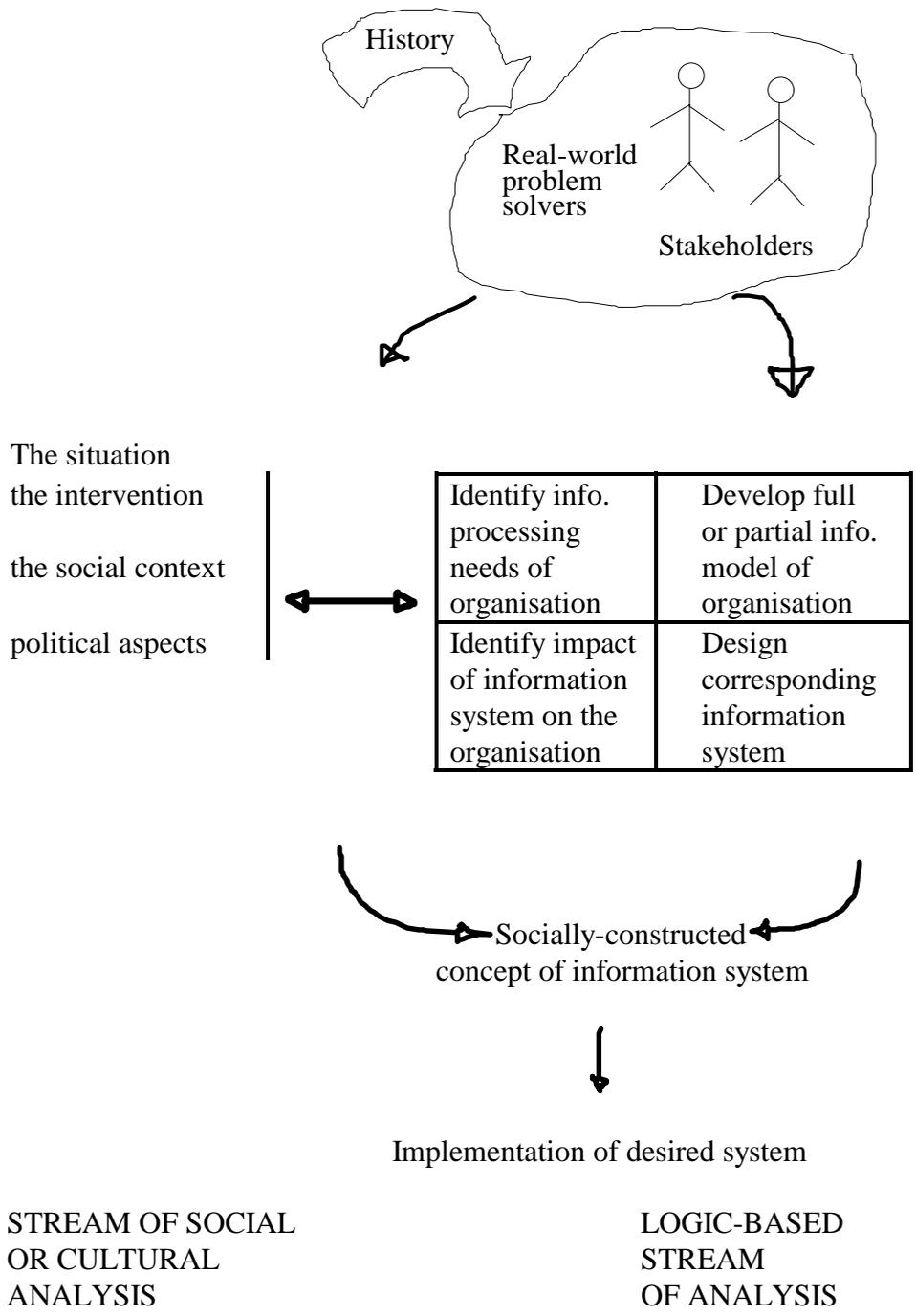


Figure 1 A Framework for the ISD Process (after Bennetts, Wood-Harper and Mills, 1999)

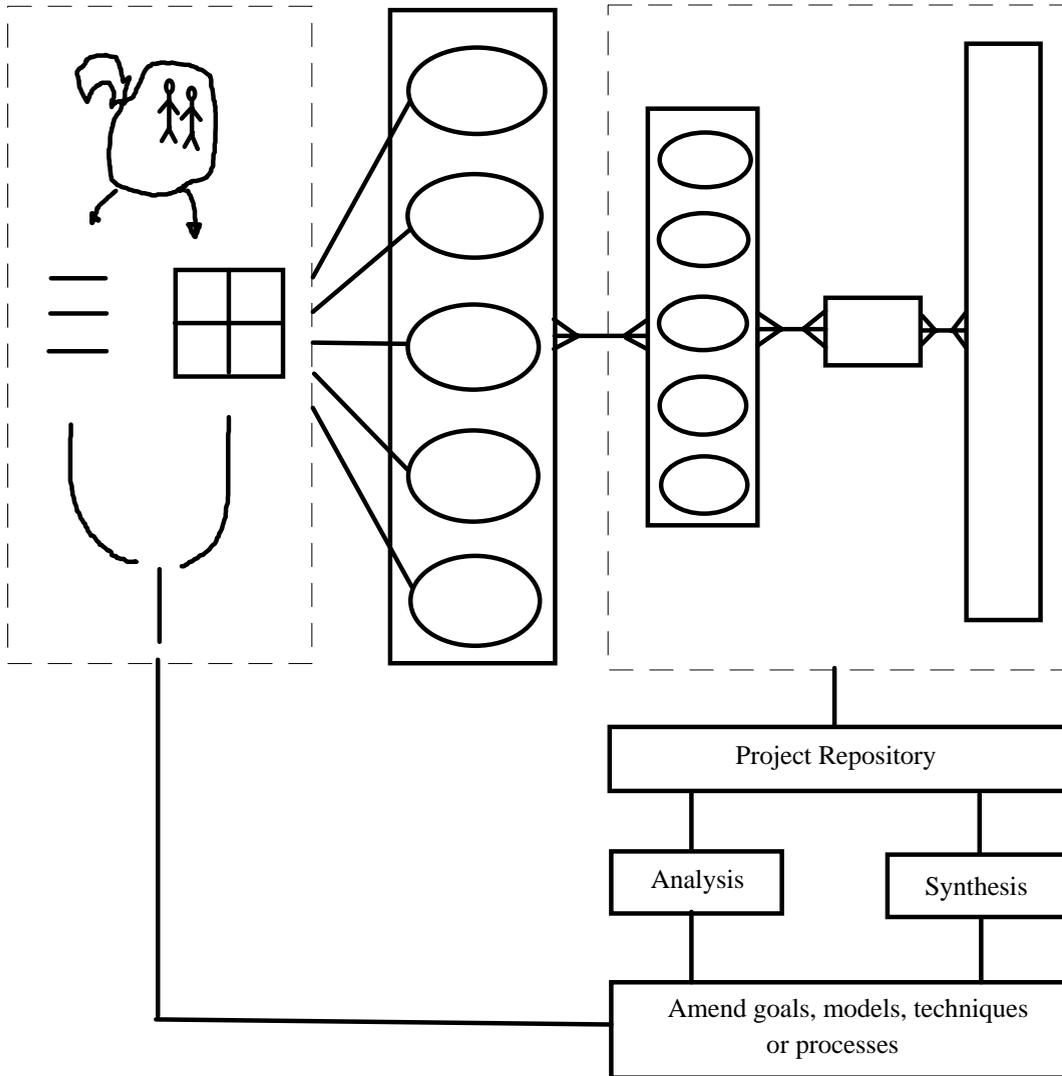
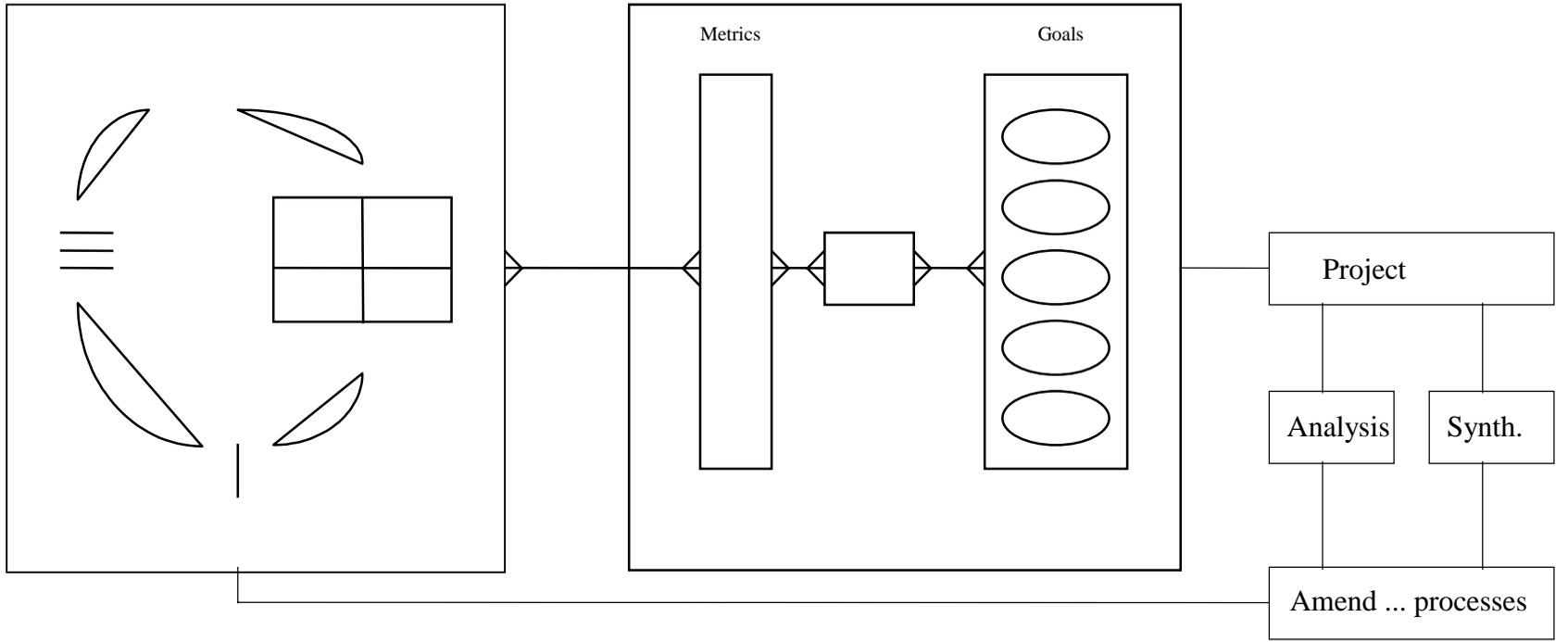


Figure 2 A Contingent Model of Software Quality for SPI

Contingent ISD as an SSM process

Basili's GQM - Adapted



428

Figure 3 Alternative image of the Contingent Factors Model

Where "Amend ... processes" represents "Amend goals, models, techniques or processes"