Towards Systems Engineering -
a Personal View of Progress

David Talbot
Commission of the European Communities
Directorate General XIII
Telecommunications, Information Industries and Innovation
rue de la Loi 200
B - 1049 Brussels, Belgium

Abstract

The paper is written from the standpoint of an Industrial consumer of Tools, Methods and Theories as they concern the building of software intensive IT systems. In recent years a good deal of effort has been applied to improving the foundations of Software Engineering. However, customers buy systems - the integration of a number of technologies aimed at providing a solution to a problem. A personal view is given of the contribution that Software Engineering advances and advances in related disciplines have made or might make to the real job of improving the way in which systems might be constructed in a predictable, cost effective manner that meets the customer's requirements.

Introduction

Technical breakthroughs (hardware and software) daily increase the range of software intensive IT applications. In turn the users are demanding that the inherent complexities of their systems are better hidden, that the systems themselves should be more cost effective and that, overall, they display higher levels of "quality". These increasing needs can only be satisfied if we consider an application not just as a software engineering exercise but rather as a systems engineering one where various components (hardware, software, users ...) interact and communicate together and where the system itself is part of a wider environment (organisation, process) with which interactions should be carefully addressed. Integration, standardisation and interoperability are key concepts which will support the systems engineering approach.

The paper aims to review three topics:

1. How far software engineering has developed in terms of meeting its more limited aims, and how this contributes to the systems engineering goal.

2. How far associated disciplines concerned with hardware, with issues of the human interface and knowledge-based systems have moved to also encompass the system view.

3. Above all, how far these notions are really entering the industrial "soul" and having clear and measurable effects on providing customers with "quality" systems.

The systems company

My view of the "systems company" (shown in simplified form in Fig 1) contains a number of straightforward notions. These include: a view of requirements elicitation which includes the need for an understanding of the "market" being investigated; the notion of component re-use to assure quality and raise productivity; and the notion of dealing with components and processes over which the engineer has only limited control. This last feature raises the issue of how to establish design and engineering integrity in an environment where the use of imported product components needs to be considered the natural way of doing business.

Note also that the model is recursive and is true for successive points on an integration chain where one man's system must be considered simply as another man's component. Scale is a constant feature; any method, theory or tool needs to accommodate itself to systems enterprises that are large. Since successive integration is a reality, the chances of needing to deal with complexity are high.

Software Engineering

Design: The notions of design "in the large" are well established. Good work has been done in theories and methods for dealing with decomposition, improved modularity and other aspects; and in developing notations that help express design more precisely. This good progress has been accompanied by the development of some tooling that extends just past the "prototype" standard. The prospects for improved tooling that will match up to real industrial expectation are good, but we must concentrate on design in the large.
Requirements: Real progress can be claimed in requirements elicitation and specification. There is an increasing recognition that IKBS and human factors work has much to offer, but integrating these possibilities has been slow. An understanding of the "market need" will continue to be more powerful than the exercise of systematic techniques supported by clever tools.

Components: The systematic and natural re-use of components remains a weak spot, and there is an increasing recognition that this is less of a technical problem than a "management" and "attitudes" issue.

Process: While we have useful models for the engineering process at the various stages of the systems building chain we lack good models of the overall process. Such a model can be the main force for the integration of methods and tools across all stages of the chain. If good design supports the business of handling intrinsic product complexity, then the successful management of scale as well as the management of project complexity is concerned with improved theories, methods and tools that support the engineering process.

Other systems contributors

Hardware: To the hardware engineer the notion of components and their repeated use is a natural one. In hardware design, the most impressive work continues to be concerned with design "in the (relatively) small". Consideration of how best to handle design of large complexity moves towards convergence with many of the ideas developed by software engineers. Increasingly, good decision support techniques will be needed to provide guidance on which functions to realise in silicon and which in software. Both hardware and software engineers need to develop what might be called a "meta-process" for the overall chain of activity.

IKBS: IKBS techniques and paradigms still represent largely unfulfilled promise in improving techniques for design, process management and requirements elicitation and specifications. Too many practitioners give insufficient attention to the industrial requirements of predictability, maintainability and other attributes of a well engineered system or component.

Human interface: Human interface teams are now beginning to make real contributions to both the value of the system product and to the system development process. But this has been slow to be integrated into the standard working environment of the systems engineer.

Penetration into industrial practice

The rate at which the new techniques are finding their way into the fabric of routine industrial use is too slow. Generalisations about management reluctance to invest, and reluctance to adopt new methods until they are fully proven, are simplistic. We need to look at general models of innovation in order to understand why it may take 18 years from the definition of an idea to its commercial take-up.

Points for the conference

The Conference is clearly centered around software engineering activity: IKBS, human interfaces, and penetration into industrial practice are addressed in a software engineering perspective. The system's view puts more emphasis on integrated solutions.

The presentations show progress in the better understanding, modelling and control of some of the constituent processes, but we still do not know how best to integrate these to deliver, more predictably, well engineered "system components". We have even less understanding of the "meta-processes that are required to deliver large scale, multi-stage systems.

Finally, how to accelerate the take-up of improved theories, methods and tools? Clearly, we must continue to worry about developing better metrics, agreeing these and proving effectiveness; but to learn to understand better the relationships of this specific problem with the more general problems of the innovation process is vitally important. If we do pursue these goals, however, such exploration will take us well outside our own familiar surroundings.

Acknowledgments

This is based on a paper given at the last UK Alvey Project Conference and a version later appeared in the ICL Technical Journal; permission to use material in this paper is gratefully acknowledged.

Thanks are also offered to Peter Cropper, Brian Warboys and others in my old company ICL who have been worrying professionally about this subject for some years and whose ideas have been a rich source of "inspiration".

Acknowledgments

This is based on a paper given at the last UK Alvey Project Conference and a version later appeared in the ICL Technical Journal; permission to use material in this paper is gratefully acknowledged.

Thanks are also offered to Peter Cropper, Brian Warboys and others in my old company ICL who have been worrying professionally about this subject for some years and whose ideas have been a rich source of "inspiration".