Empirical Studies of the Global Software Process - The Impact of Feedback

Position Paper submitted to the Workshop on Empirical Studies of Software Maintenance (WESS'99)
Sept. 3 - 4, 1999, Keble College, Oxford, UK

G Kahen, M M Lehman and J F Ramil
Department of Computing
Imperial College
London SW7 2BZ
tel: +44 (0)171 594 8216 fax: +44 (0)171 594 8215
{gk,mml,ramil}@doc.ic.ac.uk http://www-dse.doc.ic.ac.uk/projects/feast

The Global Software Process

The FEAST hypothesis [leh94] and the nature and results of the FEAST/1 study have been widely disseminated in the last three years, as may be seen from examination of its publications [leh96,leh98c,d,cha99,fea99]. We would claim that these in themselves demonstrate how the FEAST group has and is continuing to address the first three questions that define the WESS'99 workshop theme and its scope [wes99]. One of the key results of FEAST/1 is that the *global* software process is a feedback system, whose behaviour is strongly impacted by the properties of the feedback mechanisms it contains. The global process includes the activities of all involved, for example, developers, managers, marketeers, support personnel and users, as shown in figure 1. The figure and related concepts are explained in detail in [leh98c].

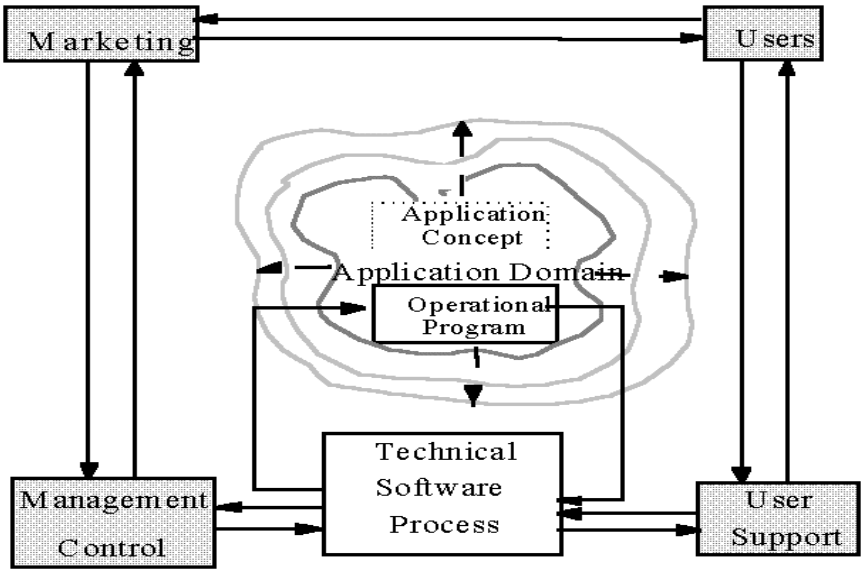

Figure 1 - The Global Software Process

In a recent paper, addressing the empirical evidence in the literature on productivity and quality improvements due to the introduction of software technology, Glass indicated that the available evidence is so far inconclusive [gla99]. Glass referred, in particular, to software technology such as structured techniques, fourth generation languages (4GLs), computer aided software engineering (CASE), formal methods, cleanroom methodology, process models and object orientation. Evaluative studies addressing this issue pose particular challenges. Glass suggests, for example, that learning curve effects have to be considered, since it takes time and effort for a particular technology to yield its potential.

His remarks raise a major issue but appear to overlook one crucial aspect of the problem. As first suggested more than 25 years ago [bel72,leh85], and supported by recent FEAST/1 results [fea99], the *global* software process is a feedback system. This observation applies to actual industrial software production, that is, to *in vivo* settings and, as has been shown [fea99], should not be ignored. Unlike *in vitro* experimentation, the evidence suggests that *in vivo* software process behaviour (for example its performance, however measured) is strongly influenced, possibly dominated, by the interaction of the many forward and feedback mechanisms. This observation includes the impact of software technology. Moreover, management-related and organisational factors [kah97], themselves involving major use of feedback, also play a significant role in determining productivity and performance.

The fact that all these have been largely ignored in the majority of the studies may well explain the difficulties in obtaining consistent empirical results across different domains of the impact of software technology changes and other aspects of the wider process of software maintenance. *In vivo* empirical studies of software maintenance must consider the *global* process including its feedback mechanisms, if, for example, conclusive evidence about the software process across different domains is to be obtained.

FEAST/2

Progress made in understanding the impact of feedback in the global software process has inevitably led to recognition of new issues and others requiring further clarification. FEAST/2 started in April 99 with plans for a two-year investigation with all the original FEAST/1 collaborators (ICL, Logica, Matra-BAe and Lucent Technologies - Bell Labs) and the addition of BT Labs. In general, the project aims to refine and extend the FEAST/1 models (black box [leh98c] and white box - system dynamics [cha99]) so that they can be effectively exploited by the industrial collaborators in the context of software process improvement and software evolution planning management. To this end the project will refine existing models, drawing on new observations and by refining models to a level at which the impact of management controls becomes explicit.

FEAST/2 goals also include, *inter alia*, effort estimation in the context of software evolution, and the development of refined system dynamics models to identify forward and feedback process mechanisms and to determine how these may be adjusted for process improvement. The project also plans to examine the evolutionary behaviour of commercial off-the-shelf (COTS)-intensive software, components-based and reuse-based software [leh98a]. A detailed view of FEAST/2 plans is given in the project proposal [leh98b], from which the lists presented below have been extracted.

Detailed Tasks, Objectives and Issues

On the basis of what has been learned from FEAST/1, the following list represents tasks that we recognised as being of immediate theoretical interest and practical concern. FEAST/2 does not have the resources to undertake all of them and the selected objectives are listed further below. We list them here and offer them to the wider community interested in the empirical study of software evolution as a challenge.

i derive strategies, guidelines and rules to support decision processes in software evolution management, process design and control, project planning and management and process improvement, with indicators of process capacity, constraints, thresholds and limits for stability and good practice
ii formalise these findings
iii investigate data patterns to uncover correlations and causal mechanisms underlying behaviour
iv investigate short term behaviour of evolution processes, related data and drivers that underlie them
v develop online monitoring, evolution planning, process improvement approaches suggested by i - iv
vi continue monitoring the systems already under study to refine interpretations of observed behaviour
vii extend the investigation of software system evolution by analysis of data from additional application areas and implementation domains, the BT system for example
viii reify models to discover and reflect the influence of lower level mechanisms on process behaviour and product properties and that of dynamics in the evolution domain
ix determine extent to which FEAST models reflect process mechanisms, replicate behaviour and predict it
x attempt generalisation of FEAST models to derive generic models within and across organisations
xi investigate extent and mechanisms of organisational and user feedback and control over the process
xii identify and propose means to modify and adapt the associated mechanisms for maximum utility in the context of organisational goals
xiii investigate the relationship, if any, between feedback dynamics and the workings and validity of paradigms such as CMM for process improvement or COCOMO [boe81,cla98] for estimation. If, as is likely, it is significant it will suggest refinement of these models and/or the development of others
xiv examine relevance of the FEAST findings, including the laws of software evolution, to systems involving a significant proportion of reused or COTS elements [leh98a]
xv integrate results as a foundation/framework for a theory of software evolution and the software process

FEAST/2 will concentrate on tasks i - iv, vi - ix, and xi. Task v is omitted as per the suggestion of the current collaborators. Task x is excluded because more understanding of the influences and mechanisms that determine evolutionary behaviour must be achieved before one can hope to develop generic models. Though they reflect issues of significant concern and opportunity.

Thus, the objectives of FEAST/2, with primarily related tasks indicated in brackets, are:

- refine set of models and their interpretations and formulate laws and rules derived from them (i, ii)
- develop and refine FEAST methods and conclusions to forms suitable for transfer to industry (i, ii)
- develop models of mechanisms underlying observed behaviour (iii, iv, viii, ix, xi)
- monitor systems studied in FEAST/1 and extend techniques to new systems and data sets (vi, vii)

In addition to these objectives, the following issues will be addressed:

- formalisation and presentation of a discipline for software evolution metrics data analysis, modelling and interpretation
- extraction of practical techniques for the analysis and management of software evolution
- their refinement and formulation for transfer to and independent application by industry
- further development and phenomenological validation of the emerging theory of software evolution and demonstration of its application to software technology and process improvement

FEAST/2 will also establish technical criteria for management decision support, planning and tools to support software evolution and process improvement. It is hoped this will help to open up the way for industry wide exploitation. However, the collaborators have suggested that we concentrate on research issues and exclude development of tools of this type from the FEAST/2 program. They see the development of such tools as requiring strong familiarity with organisational concerns and culture as well as with methods, procedures and tools being used elsewhere in the organisation. Given the technical guidelines, tool development is therefore best undertaken internally.

Final Remark

FEAST/2 is focusing on control and exploitation of process behaviour. It will determine more fully the implications of the FEAST hypothesis [leh94] and its significance for the management of software evolution and its support. In general, it seems to be important that future empirical studies be extended and adapted to consider the global software process and the effects of feedback mechanisms it contains. This proposal will proved vital if further understanding and mastery of the universal software evolution phenomenon are to be achieved.

Acknowledgements

Financial support from the UK EPSRC (FEAST/1 and FEAST/2) is gratefully acknowledged.

Note

The authors expect that, when the workshop will be taking place (Sept. 99), preliminary results of FEAST/2 will be available and will appreciate the opportunity to present and to share them with the rest of the participants.
Two of the authors (GK and MML) will be able to attend the workshop only during one of the days (Sept. 3rd). The third (JFR) will be available to attend during the two days.

References

[bel72] Belady LA and Lehman MM, *An Introduction to Program Growth Dynamics*, in *Statistical Computer Performance Evaluation*, W Freiburger (ed), Academic Press, New York, 1972, pp. 503 - 511

[boe81] Boehm BW, *Software Engineering Economics*, Prentice-Hall, 1981

[cha99] B W Chatters, M M Lehman, JF Ramil JF, P Wernick, *Modelling a Software Evolution Process*, to appear, ProSim'99, Softw. Process Modelling and Simulation Workshop, Silver Falls, Oregon, 28-30 June 99

[cla98] Clark B, Devnani-Chulani S and Boehm BW, *Calibrating the COCOMO II Post-Architecture Model*, Proc. ICSE98 Vol. I, IEEE C.S. Ord. Nr. PR08368, April 19-25, 1998, Kyoto, Japan , pp. 477-480

[fea99] FEAST *(F*eedback, *E*volution and Software *T*echnology) home page: http://www-dse.doc.ic.ac.uk/ projects/ feast/

[gla99] Glass RL, *The Realities of Software Technology Payoffs*, Comm. ACM, Feb. 1999, Vol. 42, No. 2, pp. 74 - 79

[kah97] Kahen G, *Devising the Convergence Manufacturing Strategy for Productivity Improvement: Effectiveness based on the Human Element*, Int. J. Of Materials and Product Technology, Vol. 12, No. 1, 1997, pp 18-26

[leh85] Lehman MM and Belady LA, *Program Evolution, - Processes of Software Change*, Acad. Press, London, 1985, 538 pps.

[leh94] Lehman MM, *Feedback in the Software Evolution Process*, Keynote Addr., CSR Eleventh Annual Workshop on Softw. Evolution: Models and Metrics. Dublin, 7-9th Sept. 1994, in Information and Softw. Tech., sp. Iss. on Softw. Maintenance, v. 38, n. 11, 1996, Elsevier, 1996, pp. 681-686

[leh96] M M Lehman and V Stenning, *FEAST/1: Case for Support*, ICSTM, DoC, EPSRC Proposal, Nov. 1995/March 1996, 11 pps.

[leh98a]* Lehman MM and Ramil JF, *Implications of Laws of Software Evolution on Continuing Successful Use of COTS Software*, Res. Rep. 98/8, Dept. of Comp., Imp. Col., London, June 1998, to appear in ACM publication, 1999
[leh98b]* Lehman MM, *FEAST/2: Case for Support*, Dept. of Comp., Imp. Col., London, EPSRC Proposal, July 1998, 11 pps.
[leh98c]* Lehman MM and Ramil JF, *Feedback, Evolution And Software Technology - Some Results from the FEAST Project*, Keynote Lecture, Proc.11th Int. Conf. on Software Engineering and its Application, Preprints, Vol. 1, Paris, 8 -10 Dec. 1998, pp. 1 - 12
[leh98d]* Lehman MM, *FEAST/1 Final Report - Grant Number GR/K86008*, Res. Rep. 99/1, Dept. of Comp., Imperial College, London, Dec. 1998
[wes99] WESS'99 call for papers, http://dec.bmth.ac.uk/ESERG/WESS99cfp.html

Dr Goel Kahen is a senior research associate at Dept. of Computing, Imperial College, working on the FEAST/2 project. Previously he was a senior lecturer at Tehran University and a Visiting Research Fellow at Imperial College. He holds a BA in economics, a BEng and MSc in engineering (Tehran), an MBA (Tehran) with distinction and a PhD from Imperial College, University of London. He has over 20 years' experience in management, engineering and technology development with special expertise in the area of human factors, information technology and organisational integration. He has published a total of 11 books and his most recent textbook has been reprinted 4 times. Dr Kahen's work has been published at international and national conferences and journals in the fields of technology, management, software development, ergonomics and energy.

Professor Lehman has a 29 year record of software process research dating back to 1969 that has focused on various topics within an overall theme of increased understanding and improvement. Topics addressed include:
Program evolution and *evolution dynamics*: As part of a process study at IBM in 1968 he investigated the growth of OS/360. The study suggested that the evolution process necessary to maintain a software system satisfactory develops feedback-system-like behaviour (1969), its own systems dynamics (1972). These constrain system growth despite managerial efforts at control of growth rates and technical direction. His final report, The Programming Process (1969) led to an invitation to join the prestigious IFIP WG 2.3 on Programming Methodology. In follow up work at Imperial from 1972 supported by US ERO, and later by SRC, Lehman extended the OS/360 results and found similar behaviour in other systems (1980). This suggested that process dynamics is an inherent property of large software systems and not, as was then suggested, an IBM phenomenon. All this early work was, however, largely limited to one person exploring the immediately available evidence obtained from limited data.
Laws of Program Evolution, Program Classification: These results provided a basis for Laws of Software Evolution which reflect some consequences of the observed phenomena (1974,1980). It also led to a software classification scheme (l980) whose significance in software process design and improvement is now widely recognised. More recently (1989,1991) he proposed a Principle of Uncertainty based on the observation that embedded assumptions in *E*-type software tend to become invalid as the operational domain changes leading to uncertainty in the outcome of any execution.
FEAST: In 1994 he formulated the FEAST hypothesis which was the subject of study in the FEAST/1 project. The project addressed issues in relation to feedback phenomena in the software process and their practical exploitation. A related successor project, FEAST/2, has recently started.

Juan F. Ramil is a research associate at Dept. of Computing, Imperial College, working on the FEAST/2 project. He was as research assistant to the recently concluded FEAST/1 (1996-1998). Since 1996 he has participated in the preparation of some 10 publications. His research interests include empirical studies of the software process. For seven years he participated in plant automation projects, responsible for specification, development and maintenance of supervisory control and data acquisition (SCADA) systems at fuel loading terminals, tank farms and pumping facilities. He is currently a PhD candidate at the Dept. of Computing, Imperial College, exploring the relationships between effort and schedule estimation for software projects and the software evolution phenomena. He holds a BSc in electronic engineering, *Cum Laude*, first in the class, (U.Simon Bolivar, Caracas) and MSc in business engineering.

* Available from links at http://www-dse.doc.ic.ac.uk/projects/feast/