Role and Impact of Feedback and System Dynamics in Software Evolution Processes and their Improvement

EuroSPI 2000
Copenhagen, Denmark
8 Nov 2000

M M Lehman J F Ramil
Dept. of Computing
Imperial College
180 Queen's Gate
London SW7 2BZ
{mml,ramil}@doc.ic.ac.uk
http://www-dse.doc.ic.ac.uk/~mml/feast

FEAST - *F*eedback, *E*volution *A*nd *S*oftware *T*echnology

- Study of **impact** of **feedback** on ***global*** software **process** and **evolution** of its **product**

- **Background**
  - 1968-9: IBM software **process** study, including empirical data analysis
  - 1971: **feedback** phenomenon **identified**
  - 1974-86: **laws** of software evolution
  - 1989: principle of **software uncertainty**
  - 1994: **FEAST** hypothesis
  - 1996 to 2001: **FEAST/1** and **FEAST/2** projects

- **Evolution** of wide variety of systems - addressing different **application areas** and spanning differing **sizes**, **development** and **usage environments, processes**, **evolved**, **marketed** by **ICL**, **Logica**, **DERA**, **Lucent Technologies, Matra-BAe**, **BT**

- Generally **consistent observations** over a thirty year period

What is software **evolution** and **why** must software **evolve?**

E-type Systems

- **Systems** solving a **problem** or addressing an **application** in the real world
- **Results** of **execution** determine **acceptability**
- Validation **criterion** is that system satisfies **current stakeholder needs**
- **Needs**, **desires**, **opportunities**, operational **domain all evolve**
- **Intrinsic** need for **continuing application**, **system evolution**

How are *E*-type systems **evolved** - **developed**, **maintained**?

E-type System Evolution Process

Trigger - Need/Demand/Opportunity not satisfied by current system

⇩

Preliminary **statement** of required **change**

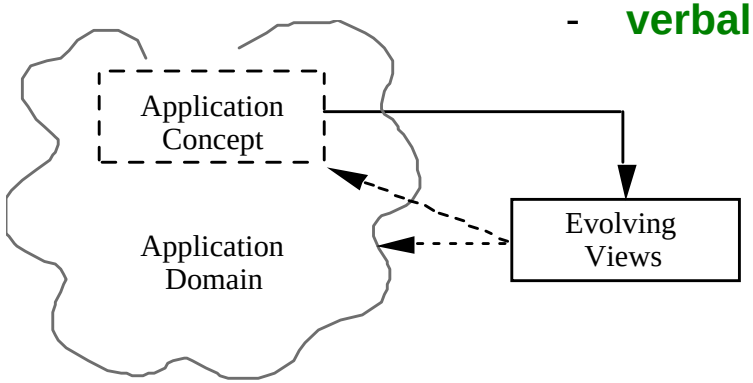
⇩

Yields - new or **changed application concept** - high level **requirement**

- new or **changed domain** of application

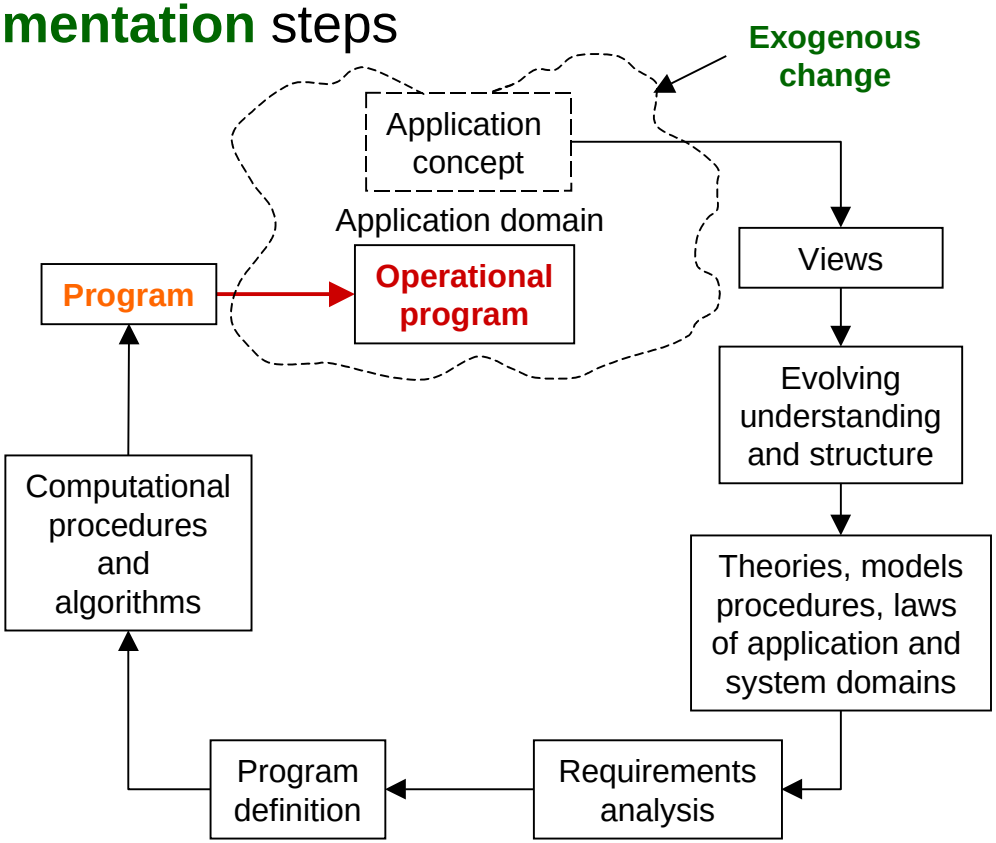
- **Initially both**:
  - **ill defined**
  - **not** fully or precisely **verbalised**
  - not explicitly **bounded**

- Ultimate operational domain - ***universe***
  - **Initial** process **steps**
    - **verbalise**, **bound**



E-type Software Evolution Process

- Series of **development** and **implementation** steps
- In a **changing real world domain**
- Culmination: **validated program**
- Final step: **installation**, **operation**
- **Installation changes domain**
- **Usage changes application**
- System **contains** implicit **model** of **itself**

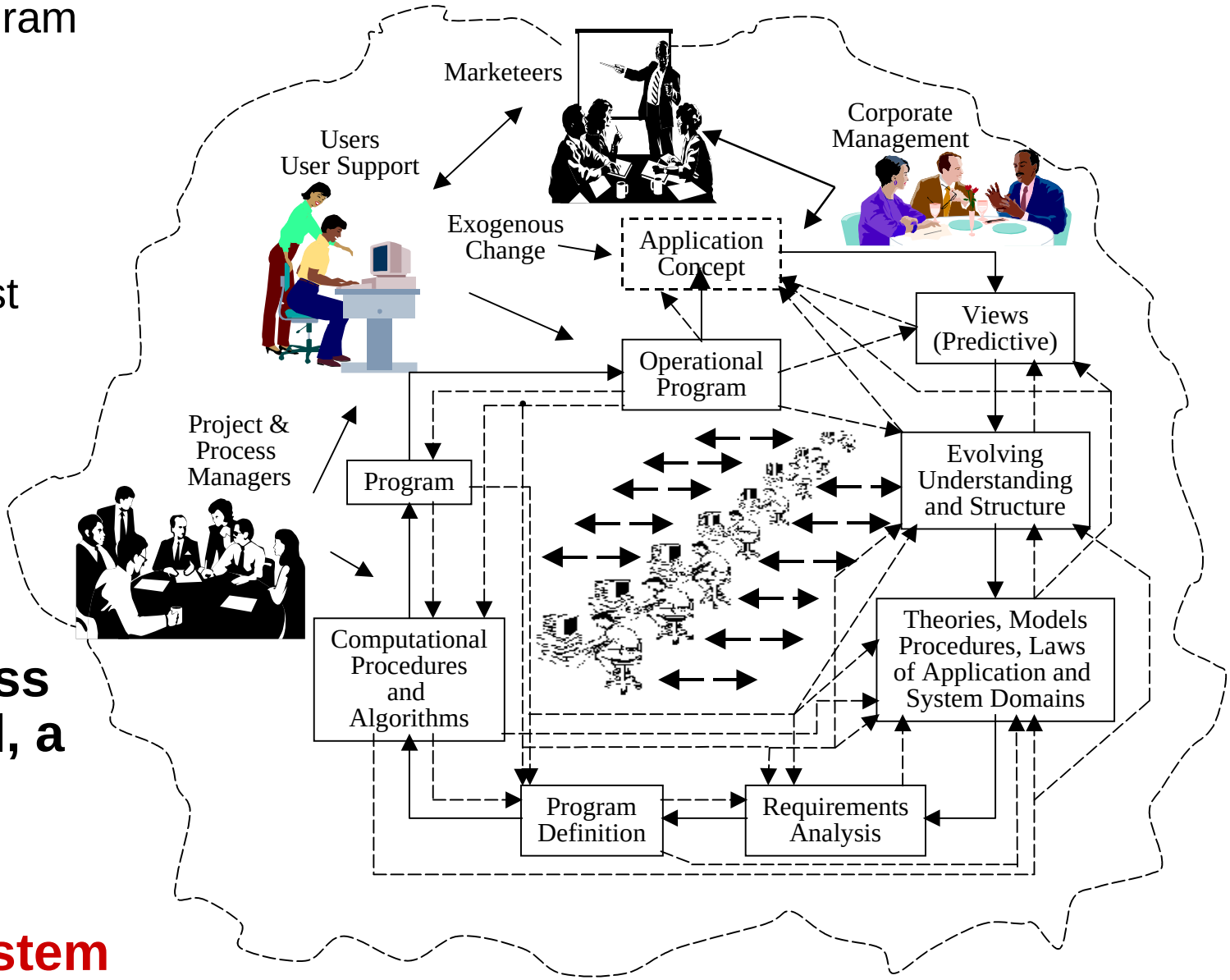


Installation, introduction into **usage**,closes major **feedback loop**

Driver of **continuing** software ***evolution***

More Realistic Picture

- Previous diagram a **fiction**
- Process **not** sequential
- **More** than just technical development activity



Global process is, in general, a multi-level multi-loop multi-agent feedback system

The Global Process

- Process the aggregated effect of **all activities** that **transform concepts**, **ideas**, **needs**, **resources** into **code** and other **deliverables**
- Satisfies engineering **definition** of **feedback** - "***Output from part of system (process) modifies one or more** of its **inputs***"
- **Complex**, **non-linear**, possibly **time varying**, loop structure
- Feedback **must** be **considered** when **modelling**, **planning**, **assessing**, **predicting**, **controlling**, **improving** process **properties**, **behaviour**

These and other **observations encapsulated** in
laws of software evolution

leading to
rules and tools for software evolution planning and management

The Laws

No.	Brief Name	Law
I 1974	Continuing Change	*E*-type systems must be continually adapted else they become progressively less satisfactory in use
II 1974	Increasing Complexity	As an *E*-type system is evolved its complexity increases unless work is done to maintain or reduce it
III 1974	Self Regulation	Global *E*-type system evolution processes are self-regulating
IV 1978	Conservation of Organisational Stability	Average activity rate in an *E*-type process tends to remain constant over periods or segments of system lifetime
V 1978	Conservation of Familiarity	In general, the average incremental growth (long term growth rate) of *E*-type systems tends to decline
VI 1991	Continuing Growth	The functional capability of *E*-type systems must be continually enhanced to maintain user satisfaction over the system lifetime
VII 1996	Declining Quality	Unless rigorously adapted to take into account changes in the operational environment, the quality of *E*-type systems will appear to be declining
VIII 1996	Feedback System (Recognised 1971, formulated 1996)	*E*-type evolution processes are multi-level, multi-loop, multi-agent feedback systems

Nature and Implications of the Laws

- **Focus** on **organisational**, **human** (e.g. cognitive) factors and on **process** as a **system**
- Termed **laws** because they reflect behaviours common to current software management and development **methods**, **tools**, **techniques**
- Encapsulate **behaviour** and **regularities** observed over the years - FEAST data **supports** 6 of the 8 laws
- The **laws** and their roots in the **feedback structure** of the process suggests **quantitative modelling** as a tool for achieving process **understanding** and **improvement**

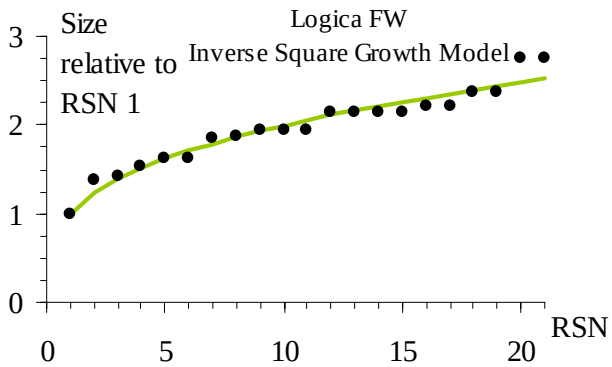
Black box and ***white box*** **models** of industrially evolved systems

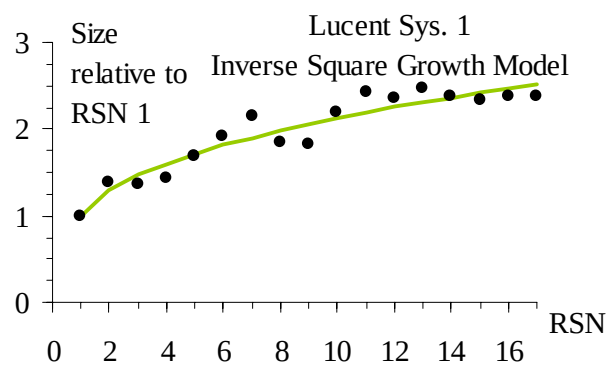
Quantitative Modelling

- **Quantitative models** can provide guidelines for process **planning**, **management and** improvement
- And a **tool** for evolution **planning** and **management**
- **Black box** models for globally observed input output behaviour
- **White box** models for reflecting behaviour of internal mechanisms and dynamic behaviour
- Model construction and analysis requires wide variety of specialised **knowledge**, **skills**
- Useful at **maturity level** beyond the most primitive
- **They represent an organisational asset**
  - characterise process **performance**
  - **reasoning** about process **changes** and **policies**
  - support for **cost**, **interval**, **performance estimation**
  - become part of organisational **learning**, **familiarisation**, **experience base**

Example of Black Box Modelling

- **Growth in modules**





Five out of six systems over 5 - 15 years

inverse square trend

- **Steady**, but **declining** growth rate with **ripple**
- **Example** of common **behavioural pattern** across systems studied - leads to **method** and **tools** to replace arbitrary by **disciplined release content control**
- **Report** - *rules and tools for software evolution and planning* - thirty six general **implications** on **process practice improvements**

Examples of Rules and Tools - Relating to Assumptions

Underlying observation - *embedded **invalid** **assumptions** **major source** of defects*

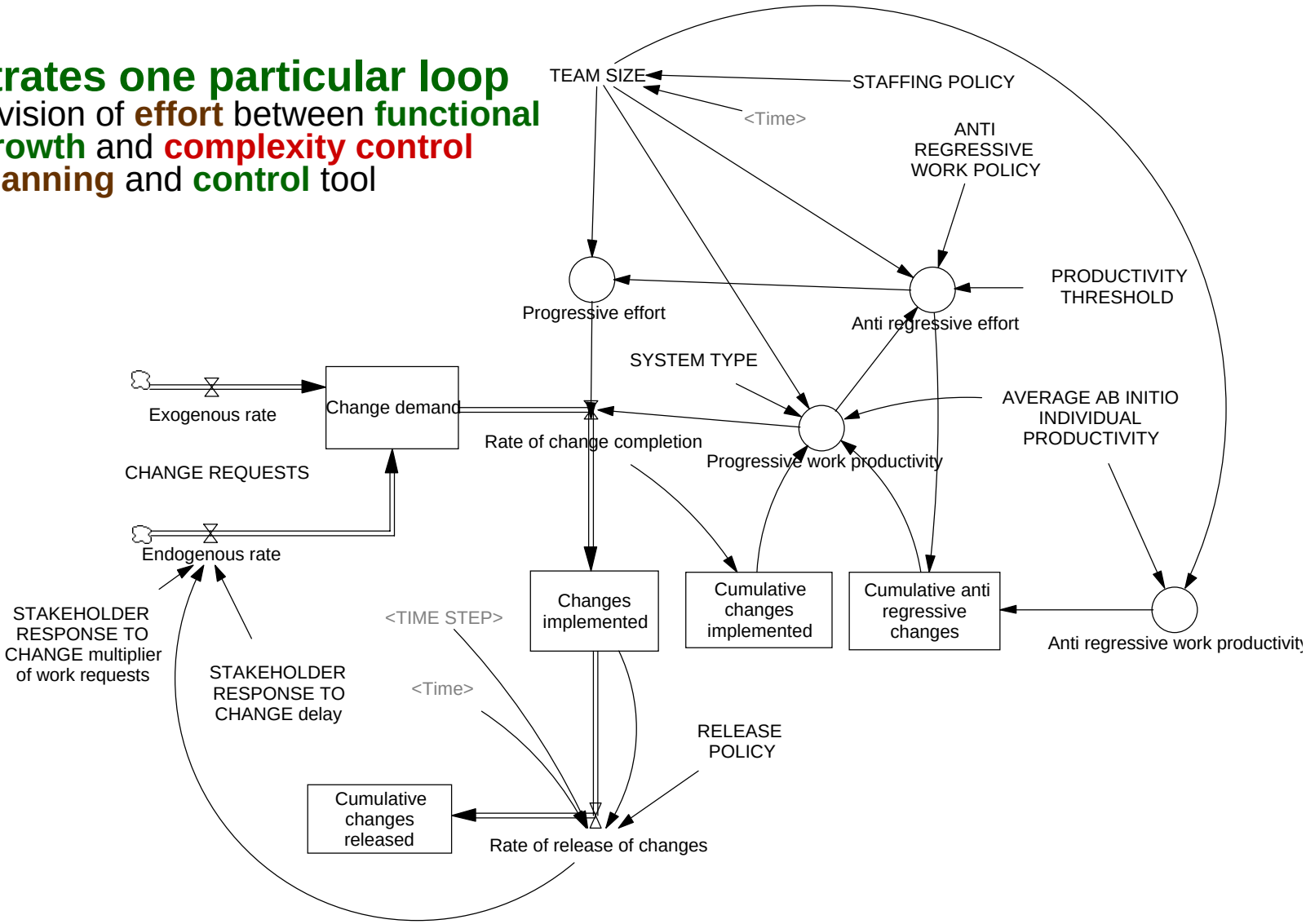
- **Capture**, **document**, **structure**, **retain** **assumptions** in specifications
- **Verify** assumptions while verifying specifications
- **Continuing** **capture**, **updating**, full **documentation** of assumptions during **design**, **implementation**, **integration**, **validation** work
- **Develop**, **use** tool support for **recording** of assumptions in **structured** form, **classified** by appropriate categories, **throughout the process**
- **Document** assumptions, design, implementation **rationale** underlying change
- **Periodically** **review** assumption set
- **Validation** of **all** changes must address interaction with and impact on that part of the system that is **not changed**
- Validation must include **check** of the **continued validity** of assumptions

Planning, **managing**, **controlling** a**ssumptions** about **application**, **operational domain** **key** to successful SPI

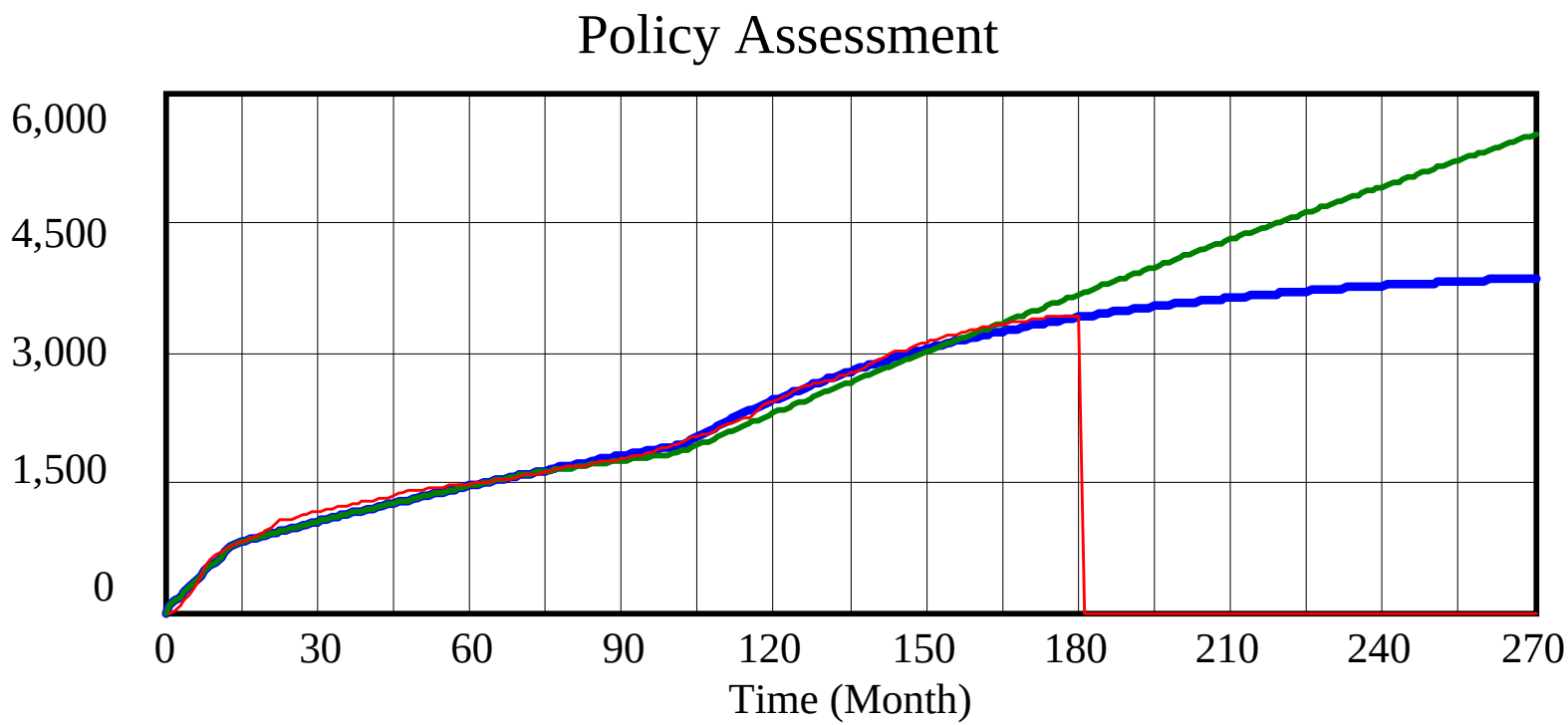
Example of White Box Model

- System dynamic **model** of **long term** process behaviour

- **Illustrates one particular loop**
  - division of **effort** between **functional growth** and **complexity control**
  - **planning** and **control** tool



White Box Model Output



Case AR = 0.0
No significant complexity control effort, long term growth constrained

Case AR = 0.3
Complexity control investment enables further long term growth

Preliminary and incomplete calibration of a commercial system

Final Remarks

- Evidence for presence of **feedback-based dynamics**
  - observation of industrial process
  - **simulation models** that include **feedback mechanisms** and **reproduce** behaviour
- **Local optimisation** of global process has **little value** when system dynamic forces **constrain** impact of local process changes
- Major **process improvement** requires **adjustment** of **feedback mechanisms**
- The nature of the process as a **feedback system** makes **modelling** an **essential** tool
- **Combining** black and white box modelling techniques provides additional power to the approach
- **Feedback** nature of process **explains slow progress** in achieving **major** improvement
- **Formal theory** of **Software Evolution** currently subject of a funding proposal - **Deutsche Bank, DERA, Ericsson, IBM, ICL**

References

References identified with an * are reprinted in [leh85]

[bel72]* Belady LA, Lehman MM, *An Introduction to Program Growth Dynamics*, in Stat. Comp. Perf. Ev., W. Freiburger (ed.), Ac. Press, NY, 1972, pp. 503-511

[cha99] Chatters BW, Lehman MM, Ramil JF, Wernick P, *Modelling a Software Evolution Process*, ProSim'99, Softw. Process Modelling and Simulation Workshop, Silver Falls, Oregon, 28-30 Jun. 1999, also as *Modelling a Long Term Software Evolution Process* in J. of Softw. Proc.: Improvement and Practice, 2000, v. 5, iss. 2/3, Jul. 2000, pps. 95-102

[cox66] Cox DR, Lewis PAW, *The Statistical Analysis of Series of Events*, Methuen, London, 1966

[fea94,95] *Preprints of Three FEAST Workshops*, Lehman MM (ed.), Dept. of Comp., Imp. Col., 1994/5

[fea00a] FEAST, *Feedback, Evolution and Software Technology*, web site: http://www-dse.doc.ic.ac.uk/~mml/feast

[fea00b] *Preprints of FEAST 2000 International Workshop on Feedback and Evolution in Software and Business Process*, Ramil JF (ed.), Dept. of Comp., Imp. Col., London, 10 - 12, Jul. 2000, 124 pp. http://www-dse.doc.ic.ac.uk/~mml/f2000

[for61] Forrester, J., *Industrial Dynamics*, The MIT Press, 1961.

[kah00] Kahen G, Lehman MM, Ramil JF, Wernick PD, *Dynamic Modelling in the Investigation of Policies for E-type Software Evolution*, ProSim 2000, 12 - 14 Jul. 2000, London, UK

[leh69]* Lehman MM, *The Programming Process*, IBM Research Report RC 2722, IBM Research Centre, Yorktown Heights, NY, Sept. 1969

[leh74]* Lehman MM, *Programs, Cities, Students, Limits to Growth?*, Inaugural Lecture, in Imperial College of Science and Technology Inaugural Lecture Series, Vol. 9, 1970, 1974, pp. 211-229. Also in Programming Methodology, (D. Gries. ed.), Springer Verlag, 1978, pp. 42-62

[leh85] Lehman MM, Belady LA, *Program Evolution—Processes of Software Change*, Academic Press, London, 1985.

[leh89] Lehman MM, *Uncertainty in Computer Application*, Comm. of the ACM, Vol. 33, No. 5, May 1990, pp. 584-586

[leh90] Lehman MM, *Uncertainty in Computer Application*, Technical Letter, Comm. of the ACM, vol. 33, no. 5, pp. 584, May 1990

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

[leh98] Lehman MM, Perry DE and Ramil JF, *On Evidence Supporting the FEAST Hypothesis and the Laws of Software Evolution*, Proc. Metrics'98, Bethesda, MD, 20-21 Nov. 1998, pp. 84-88

[leh00] Lehman MM, *Rules and Tools for Software Evolution Planning and Management*, FEAST 2000 Workshop, 10 - 12 July 2000, Imp. Col.

[pro00] Prosim 98, 99 and 2000, *Software Process Simulation and Modeling Workshops*, http://www.prosim.org

References

[ram00] Ramil JF, Lehman MM, Kahen G, *The FEAST Approach to Quantitative Process Modelling of Software Evolution Processes*, Proc. PROFES'2000 2nd International Conf. on Product Focused Software Process Improvement, Oulu, Finland, 20-22 Jun. 2000, in Frank Bomarius and Markku Oivo (eds.) LNCS 1840, Springer, Berlin, 2000, pp. 311-325

[raj00] Rajlich VT and Bennet KH, *A Staged Model for the Software Life Cycle*, Computer, July 2000, pp. 66 - 71

[sim96] Simon HA, *The Sciences of the Artificial*, 3rd. ed. The MIT Press, Cambridge, MA, 1996, 231 pp, first pub. in 1969

[tur96] Turski WM, *A Reference Model for the Smooth Growth of Software Systems*, IEEE Trans. Softw. Eng., v. 22, n. 8, Aug. 1996, pp. 599 - 600

[wer98] Wernick P and Lehman MM, *Software Process White Box Modelling for FEAST/1*, ProSim '98 Workshop, Silver Falls, OR, 23 Jun. 1998. As a revised version in J. of Sys. and Softw., v. 46, n. 2/3, 15 Apr. 1999

[wid89] Widman LE and Loparo KA, *Artificial Intelligence, Simulation, and Modeling - A Critical Survey*, in Widman LE, Loparo KA and Nielsen NR (eds.), Artificial Intelligence, Simulation and Modeling, Wiley, NY, 1989

[zur67] Zurcher FW and Randell B, *Iterative Multi-Level Modelling - A Methodology for Computer System Design*, IBM Res. Rep. RC 1938, Nov. 1967, IBM Res. Centre, Yorktown Heights, NY 10594. Also in Information Processing 67, Proc. IFIP Congr. 1968, Edinburgh, Aug. 1968, pp. D138 - 142

General Bibliography

• Belady LA & Lehman MM, *An Introduction to Program Growth Dynamics*, in Statistical Computer Performance Evaluation, W Freiburger (ed), Academic Press, New York, 1972, pp. 503 - 511
• Boehm BW, *Software Engineering*, IEEE Trans. on Comp., v. C-5, n. 12, Dec. 1976, pp. 1226 - 1241
• *id*., *A Spiral Model of Software Development and Enhancement*, Computer, v. 21,, May 1988, pp. 61 - 72
• Brookes FP, *No Silver Bullet - Essence and Accidents of Software Engineering*, Information Processing 86, Proc. IFIP Congress 1986, Dublin, Sept. 1-5, Elsevier Science Pubs. (BV), (North Holland), pp. 1069 - 1076
•Lehman MM, *Programs, Cities, Students - Limits to Growth*?., Imperial College. Inaugural Lecture Series, v. 9, 1970 - 1974, also. in [GRI78], pp. 42 - 69, Lehman and Belady, 1985, pp. 133 - 163
• *id*., *Laws of Program Evolution - Rules and Tools for Programming Management*, Proc. Infotech State of the Art Conf., Why Software Projects Fail, - Apr 9 - 11 1978, pp. 11/1 - 11/25
• *id*., *Programs, Life Cycles and Laws of Software Evolution*, Proc. IEEE Sp. Iss. on Softw. Eng., v. 68, n. 9, Sept 1980, pp. 1060 - 1076
• Lehman MM, Stenning V & Turski WM, (1984). *Another Look at Software Design Methodology*, ICST DoC Res. Rep. 83/13, Jun 1983. Also, Software Engineering Notes, v. 9, no 2, Apr 1984, pp. 38 - 53
• Lehman MM & Belady LA, *Program Evolution,- Processes of Software Change*, Academic Press, London, 1985, 538 p.
• Lehman MM, *Process Models, Process Programs, Programming Support*, Inv. Resp. To A Keynote Addr. By Lee Osterweil, Proc. 9th Int. Conf. on Softw. Eng., Monterey, CA, 30 Mar. 2 Apr 1987, IEEE Comp. Soc. pub. n. 767, IEEE Cat. n. 87CH2432-3, pp. 14 - 16
• *id,. Evolution - The Cause of Iteration*, Third Process Workshop, Breconridge, CO, Nov. 1986. In Iteration in the Software Process - Proc. 3rd Int. Process Wrkshp., Dowson M (ed), IEEE Comp. Soc. Press, Mar. 1987, pp. 29 - 32
• *id*., *Software Engineering, the Software Process and Their Support*, IEE Softw. Eng. J. Spec. Iss. on Softw. Environments. and Factories, Sept 1991, v. 6, n. 5, pp. 243 - 258
• *id.*, *Software - Promise and Threat*, Inv. paper, Proc. Software Engineering in the Nineties,. Software Eng. Res. Centrum, Utrecht, The Netherlands, Oct. 1988. Also, inv. paper, Shell Conf. on Logistics, Appledoorn, Neths., Nov. 1988, Pergamon Press, 1989, pp.172 - 183
• *id., Uncertainty in Computer Application and its Control through the Engineering of Software*, J. of Softw. Maint., Res. and Pract., v. 1, 1 Sept 1989, pp. 3 - 27
• *id.*,*Uncertainty in Computer Application*, Technical Letter, CACM, vol. 33, no. 5, pp. 584, May 1990
• *id.*, • *id., Models and Modelling in Software Engineering*, Ency. of Softw. Eng., J Marciniak (ed), Wiley and Co, 1994, vol. 1, pp. 698 - 702
• *id*., *Software Evolution*, loc cit, vol. 2, pp. 1202 - 1208
• Osterweil L, *Software Processes are Software Too, Iteration in the Software Process*, Proc. of the 3rd Int. Proc. Worksh., Breckenridge, CO, 17 - 19 Nov. 1986, IEEE cat. n. TH0184-2, IEEE Comp. Soc. order n. 709, 1987, pp. 79 - 80
• Perry DE, Policy and Product-Directed Process Instantiation, Proc. of the 6th Int. Softw. Process Workshop, 28-31 October 1990, Hakodate, Japan
• Turski WM, *And No Philosophers' Stone Either*, Information Processing 86, Proc. IFIP Congr., Dublin, Sept. 1 - 5, 1986, Elsevier Sci. Pubs, London, pp. 1077 - 1080
• [wil51] Wilkes M V, Wheeler D J and Gill S, *The Preparation of Programs for an Electronic Digital Computer*, Addison Wesley Press Inc., 1951, 167 pp.
• Wirth N, *Program Development by Stepwise Refinement*, CACM, v.14, n.4, Apr 1971, pp.221-227
• Woodside CM, *A Mathematical Model for the Evolution of Software*, J. of Sys. and Softw. vol. 1, no. 4, Oct 1980, pp. 337 - 345 and in Lehman and Belady 1985, pp. 339 - 354