IPA Software Engineering Center
Information-technology Promotion Agency, Japan

What is Process?
~ As stated in the references.~

Katsutoshi Shintani
Presented at SEA-SPIN May 7, 2009

M.M. Lehman, The Programming Process, IBM Research Report RC2722, 1969

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- The programming process is the total collection of technologies and activities that "transform the germ of an idea into a binary program tape." At the present time there is little quantitative knowledge how costs, time delays and difficulties are distributed over that process…

  However, the process is itself a "system" involving many people, many phases, many components and many requirements.

  A characteristic feature of system is that local optimization does not lead to system optimization.

M.M. Lehman, Programs, Life Cycles and Laws of Software Evolution, Proc IEEE Spec. Issue on Software Engineering Sept **1980**, pp.1060-1076

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- S-programs

  S-programs are programs whose function is formally defined by and derivable from a specification.

  As suggested by Fig. 1 (next page), the specification, as a formal definition of the problem, directs and controls the programmer in his creation of the program that defines the desired solution.
- E-programs

  E-programs are inherently even more change prone. They are programs that mechanise a human or societal activity and that are, therefore, embedded in the application. Once the program is completed and begins to be used, questions or correctness, appropriateness and satisfaction arise as in Fig. 4 (next page) and inevitably lead to additional pressure for change.

M.M. Lehman, Programs, Life Cycles and Laws of Software Evolution, Proc IEEE Spec. Issue on Software Engineering Sept **1980**, pp.1060-1076 (cont' d)

SEC Software Engineering for Mo-No-Zu-Ku-Ri

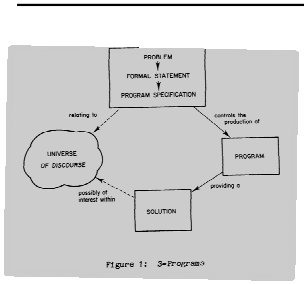

Figure 1: S-Programs

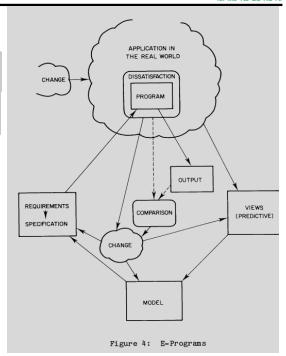

Figure 4: E-Programs

M.M. Lehman, Program Evolution, first at Symp on Empirical Foundation of Computers and Information Sciences, **1982,** JICST, published in J. Info Proc and Management, 1984, Pergamon Press

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- With current practice, an average of about 70% of the life-time expenditure on a program is incurred after initial installation. The main cost of a program is not incurred in its creation but in its subsequent evolution. To achieve an appropriate balance between initial product quality, continuing satisfaction and some desired life-time expenditure distribution, requires improved understanding of both intrinsic and process dependent modes of evolution.

W.S. Humphrey, the IBM large-systems software development process: Objectives and direction, IBM Systems Journal, pp.76-78, **1985**

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- Programming has evolved from a highly individualistic profession into an orderly engineering process.
- The first step in improving the programming development process was learning how to make an meet schedules and estimates.
- Significant progress toward the goal of a zero-defect development process will require advances in tools, methods and process. Of these three, process is the key.
- We are now focusing on the programming process by defining and documenting the steps in the process itself and by educating the people.
- Since defects are disruptive and expensive to our customers and to IBM, quality improvements are an economic necessity. Of even greater importance is the improved usability of our programs.

L.J. Osterweil, Software Processes Are Software Too, Revisited: An Invited Talk on the Most Influential Paper of ICSE 9, Proc. Of ICSE **1997,** pp. 540–548

SEC Software Engineering for Mo-No-Zu-Ku-Ri

The distinguishing characteristic of most of these debates has been the fact that there did not, and still does not, seem to be much possibility that these debates and questions ca be resolved definitively. One reason is that there is no agreed upon definition of what software is. Likewise, there is no firm agreement on what programming is, or what a process is for that matter.

…

It seems just possible that what we have been witnessing is a slow paradigm shift to a view software and software development that is rooted in the centrality of the notion of process as a first-class entity whose properties are very much like those of software itself.

M.M. Lehman, Process Modelling – Where Next, Proc of ICSE 1997, pp.549–552

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- The essential lesson to be derived from current improvement approaches is that one must develop a global view and comprehensive insight as to how, through their processes, organizations achieve and maintain quality products.
- A second major development that signals a change in direction for process modelling arises from the realization that software process is a complex, multi-loop, multi-level feedback system…
  From this it followed that understanding and improving the process required it to be treated a feedback system. Moreover, in these systems, intrinsically, humans play a major controlling role.
- The total process must be modeled in business environment to include all relevant activities and their feedback mechanisms.

A. Fuggetta, Software Process: A Roadmap, The Future of Software Engineering, pp.27–34, ACM, 2000

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- The notion of software process builds on the notion of lifecycle and provides a broad and comprehensive concept to frame and organize the different factors and issues related to software development activities.
  - Software development technology
  - Software development methods and techniques
  - Organizational behavior
  - Marketing and economy
- We have often considered software processes as a "special" and "unique" form of processes… We should heavily invest in finding and evaluating commonalities and similarities…
- Practitioners' most important need is to describe processes with the purpose of understanding and communicating them.
- We cannot forget that software development is carried out by teams of people involved in a highly creative activity. It is, indeed, a human-centered process as many others engineering and design processes in our society.

R.B. Hunter, R.H. Thayer(ed), Software Process Improvement, IEEE, 2001

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- pp.541, Chapter 12 Software Process
  Models of the software process have been used for a variety of purposes, such as:
  - To design software development environments;
  - To study software evolution and maintenance;
  - To simplify (and hence improve) the software process;
  - To identify critical parts (bottlenecks) of the software process;
  - To design tools to support the software process.

  What constitutes a good process model depends on the purpose for which it is to be used.

T. Leishman, Extreme Methodologies for an Extreme World, Cross Talk, June 2001, pp.15–18

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- Extreme methodologies take a different approach from traditional software development methodologies. These methodologies accept the notion that, change happens during all phases of the development processes. Further, it is anticipated that the system users may not know exactly what they want the final product of the development effort to be.
- These methods attempt to satisfy customer requirements by speeding development and delivering useful products in rapid iterations. Underlying each of these methods is a foundation of basic characteristics common to each.
  - An iterative approach
  - Risk driven
  - Results oriented

T. Leishman, Extreme Methodologies for an Extreme World, Cross Talk, June 2001, pp.15–18 (cont' d)

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- J. Highsmith in Software Testing & Quality Engineering:
  The old world was one of optimization where efficiency, predictability, and control dominated. The new world is one of adaptation in which change, improvisation, and innovation rule.
- Although they emphasize the need to rethink the traditional methods and eliminate processes that are not value added, they recognize the need for sound development processes.
- Extreme methodologies all share the idea of collaboration…
  Having the right people dedicated to a project will ensure that stakeholder and developer alike have agreement on issues from primary requirements through final implementation.
- Ward Cunningham, one of the fathers of XP:
  Extreme Programming is a lot of simple little things. It' s lots of things that have been done before…

D.J. Reifer, the "3P's" of Software Management, Software Management, 6th Ed., IEEE, 2002

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- 3P's = Processes, products and people
- Maturing the Process

  Producing software involves more than just writing programs.
  - Principle 1 – Recognize that good processes add value.

    Having either a good process or good people is not enough.. Getting your people to use the process is the challenge. This can be handled most easily by making your process the preferred way of doing business.
  - Principle 2 – Use your processes to share your lessons learned

    Direct your process definition efforts towards institutionalizing a preferred approach to doing business. Such an infrastructure enables you to share your experiences and build on your lessons learned, both positive and negative.
  - Principle 3 – Stress continuous process improvement

    Make sure that the process you improve is the one that your people use. Be flexible and try to build on the past in a manner that lets you address the future.

Chapter 9 Software Engineering Process, SWEBOK 2004 Version, IEEE

SEC Software Engineering for Mo-No-Zu-Ku-Ri

The term "software engineering process" can be interpreted in different ways, and this may cause confusion.

- One meaning, where the word *the* is used, which could imply that there is only one right way. There is no such a thing. Standards such as IEEE12207 speak of software engineering processes, meaning there are many processes involved.
- A second meaning refers to the general discussion of processes related to software engineering.
- Finally, a third meaning could signify the actual set of activities performed within the organization.
- Breakdown of topics for the software engineering process KA
  - Process implementation and change,
  - Process definition
  - Process assessment,
  - Process and Product measurement

ISO/IEC 12207:2008

SEC Software Engineering for Mo-No-Zu-Ku-Ri

- ISO/IEC 12207:2008 establishes a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry. It contains processes, activities, and tasks that are to be applied during the acquisition of a software product or service and during the supply, development, operation, maintenance and disposal of software products. Software includes the software portion of firmware.
- ISO/IEC 12207:2008 applies to the acquisition of systems and software products and services, to the supply, development, operation, maintenance, and disposal of software products and the software portion of a system, whether performed internally or externally to an organization. Those aspects of system definition needed to provide the context for software products and services are included.
- ISO/IEC 12207:2008 also provides a process that can be employed for defining, controlling, and improving software life cycle processes.
- The processes, activities and tasks of ISO/IEC 12207:2008 – either alone or in conjunction with ISO/IEC 15288 – may also be applied during the acquisition of a system that contains software.