

Name : **Isiram OUKAKI**

Name of your level 1 : Morgane Gaspar

Paper Title : On the Evolution of Lehman's Laws

Keywords specific : Professor Manny Lehman, Software, Evolution, Development, Software Architecture, Computer Systems, Interfaces, Complexity, Monolithic, Components, Open Source

This paper honors Prof. Manny Lehman's contributions to software evolution, discussing his background, his laws of software evolution based on industrial experience, and the changing nature of software development and its implications for future studies.

1. LEHMAN'S INTELLECTUAL JOURNEY

Manny Lehman's unconventional career path began with repairing radios during World War II. Despite finding the work repetitive, he aspired to become a tester, highlighting the importance of thinking in maintenance tasks. Transitioning to computer science, Lehman worked on early computers and later investigated programming practices at IBM. He observed a focus on productivity metrics like SLOCs and unit tests, but noted a potential loss of sight of the overall project goals.

Lehman, alongside Laszlo Belady, challenged prevailing software maintenance models at IBM and pioneered the study of software evolution. Transitioning to Imperial College London in 1972, Lehman's engineering-influenced perspective emphasized the dynamic nature of software development, contrasting with Edsger Dijkstra's formal, specification-driven approach. While recognizing the value of Dijkstra's model, Lehman postulated that software could be categorized into S-type programs, derived from rigorous upfront specifications, and E-type programs, adaptable to changing environments. This distinction underscored the importance of adaptability in real-world software systems.

2. LEHMAN'S LAWS OF EVOLUTION

Lehman's seminal work revolves around his laws of software evolution, developed over years with collaborators like Laszlo Belady. He argued that real-world software cannot be fully specified due to the ever-changing environment it operates in. This dynamic interaction leads to a continuous need for adaptation as user needs and environmental conditions evolve. Lehman emphasized that software evolution is inherent and essential, shaping his renowned Laws of Software Evolution.

3. SOME OBSERVATIONS ON MODERN SOFTWARE EVOLUTION AND LEHMAN'S LAWS

Lehman's laws of software evolution, developed in the 1960s and 1970s, may need to adapt to the changing landscape of software development. Recent trends, such as agile practices and open-source collaboration, challenge the traditional notions of release planning and software metrics. The shift towards more ad hoc release schedules and variable development artifacts raises questions about the applicability of Lehman's laws in today's software environment.

3.1. Software Architecture Emergence:

- Initial stages of large system development prioritize creating a robust software architecture.
- Over time, internal boundaries form within the system, establishing clear interfaces.

3.2. De-Monolithization of Software:

- Traditional monolithic designs give way to systems interacting with various components.
- Modern development involves understanding available services and specifying deployment details.

3.3. Open Source and Agile:

- Development shifts towards open-source components and agile practices.
- Evaluating contributions and measuring product characteristics become challenges.

3.4. Emergent Software Uses:

- Software systems adapt to evolving environments, receiving feedback from users.
- Internet infrastructure enables diverse software uses, leading to fundamental shifts in software evolution.