

Assignment - I

Software Maintenance(SE-811)

Analysis of Lehman's Eight Laws of Software Evolution

Submitted by

Mahir Faisal

BSSE-1316

Institute of Information Technology

University of Dhaka

Submitted to

Toukir Ahammed

Lecturer

Institute of Information Technology

University of Dhaka

Date of Submission: 27-10-25

Analysis of Lehman's Eight Laws of Software Evolution

Let's analyze the provided graphs to determine which ones support each of Lehman's Laws, and provide a clear explanation of the reasoning behind their alignment.

1. Number of System Calls

Law of Continuing Change:

The progressive rise in system calls over time, from approximately 140 in 1994 to over 325 in 2008, demonstrates that the system is under constant evolution. The interface continuously expands as developers introduce modifications and enhancements.

Law of Increasing Complexity:

The growth in system calls signals an expanding interface. This trend implies that unless deliberate efforts are made to manage complexity, the system's structural intricacy will increase, placing additional cognitive and operational load on users and developers.

Law of Continuing Growth:

The upward trajectory in system calls represents a measurable increase in the system's functional content, reflecting an ongoing effort to enhance capabilities and maintain user satisfaction.

2. Lines of Code (All Directories)

Law of Continuing Change:

The non-decreasing slope in the LOC graph, from roughly 50,000 to 5,000,000 lines, indicates sustained modification and enlargement of the software, consistent with perpetual evolution.

Law of Continuing Growth:

The consistent and significant rise in LOC directly reflects the system's expanding functional content, responding to user requirements and environmental demands.

Law of Conservation of Organizational Stability:

The relatively steady growth rate over long periods suggests a roughly constant allocation of organizational resources, including personnel and budget. This linear trend in later stages reflects controlled development under stable organizational capacity.

3. Number of Functions (All Directories)

Law of Continuing Change:

The continuous increase in functions confirms ongoing updates, additions, and modifications to the system's capabilities.

Law of Continuing Growth:

The steady growth in function count, a key measure of system size and functional content, demonstrates that the system is expanding to satisfy evolving user demands.

4. McCabe Cyclomatic Complexity (All Directories)

Law of Continuing Change:

As the system undergoes modifications, complexity naturally rises.

Law of Increasing Complexity:

The persistent increase in McCabe Cyclomatic Complexity aligns with the expectation that unmitigated additions and modifications tend to make the system harder to maintain and comprehend. Without deliberate efforts to manage complexity, maintainability declines.

5. Oman's Maintainability Index (All Directories)

Law of Conservation of Organizational Stability:

The generally stable or slightly rising maintainability index indicates that the development team is effectively allocating resources to maintain or improve structural quality.

Law of Conservation of Familiarity:

Despite increases in system size and complexity, the maintainability index has remained steady or improved (from 105 to 115). This reflects conscious efforts to counteract the natural decay in quality predicted by the Law of Declining Quality.

Disagreeing Laws

Law of Increasing Complexity: The index does not show the expected decline.

Law of Declining Quality: Maintainability improvements contradict the expectation that quality inevitably deteriorates without intervention.

6. Releases per Month

Law of Conservation of Organizational Stability:

The average release rate over time reflects a finite and stable organizational capacity for development and deployment.

Law of Feedback System:

The highly variable release patterns, characterized by peaks and troughs, indicate a multi-loop feedback system. Backlogs may trigger bursts of releases, followed by periods of recalibration, demonstrating adaptive management of development workflows.

7. Kernel Release Dates

Law of Conservation of Organizational Stability:

The near-constant slopes within major kernel versions, like v2.1, v2.5 suggest stable organizational output, reflecting consistent application of resources across releases.

Law of Feedback System:

Shifts between different kernel versions (v2.1 → v2.3 → v2.5) indicate dynamic adjustment of development focus based on feedback and evolving requirements.

8. Incremental Change in Number of Files

Self-Regulation:

The erratic fluctuations around a horizontal mean line Average Increment reveal a self-stabilizing, feedback-regulated system. This demonstrates ongoing adjustments to maintain equilibrium.

Law of Conservation of Organizational Stability:

Despite variability, the increments oscillate around a mean value, reflecting the maximum change the organization can handle per release.

Law of Feedback System:

The non-linear, fluctuating increments suggest a self-regulating, multi-loop feedback process. Small changes may prompt larger subsequent adjustments, ensuring that the system's growth remains aligned with organizational norms and capacity.