

Software Evolution Observations Based on Product Release History

By Harald Gall and Mehdi Jazayeri and Rene Kosch and Georg Trausmuth

Thoughts/Things of Note:

Implies that a decrease in both growth and change rates in a system imply satisfactory evolution → meaning good software practice?

I would argue the assumption of stability based on the information presented without more detail. Growth and change rates if I understand correctly were based on size comparisons: if x number of developers can make y number of changes consistently to a project then as the size of the project grows these numbers will diminish as a matter of course, though the changes are constant. It may be that I don't understand the numbers though.

A course grained analysis is a simple approach, which is appealing. Should allow for almost full-automation of the information gathering part of the process with only a limited intervention by a domain expert to give general knowledge about the structure.

Summary:

This paper is an examination of the changes between versions of a software product. It focuses on the code itself in relation with the features added in an attempt to identify potential shortcomings of the development process based in analysis of the statistics gathered.

Rather than focus on the lines of code this analysis is based on the modules and program modified for a broader or more coarse-grained approach. The result is a focus on the structural changes between releases rather than on any specific source code metrics.

The case study was done on a Telecommunication Switching System (TSS) having more than 10 million lines of code. It has a four level logical tree structure: the system comprised of subsystems, which are comprised of modules, which are comprised of programs. A Product Release Database was created to store the structural information, which was extracted by pre-processors at run-time for populating the DB. The PRDB had entries for each part of the system based on the tree structure along with the relevant version information. Relations between modules for also stored in the PRDB.

The following relevant characteristics from the PRDB were then compared through time based on release versions:

- the size of each system, subsystem and module (defined as the number of programs contained within)
- the changing rate which is the percentage of programs in a system, subsystem or module which changed from one release to the next
- the growing rate – programs added in a system, subsystem or module from one release to another

Observations that were produced:

- the size of the system is growing linearly
- there was no correlation comparing the changing and growing rates for each release

This prompted the conclusion that due to nothing dramatic showing in the charts, this was evidence of satisfactory evolution.

A subsystem with the largest growth and change rates was examined in detail to see of these characteristics made it a candidate for re-factoring. Examining the modules in the subsystem in detail in terms of growth/change showed that one of the three modules was variations much larger than the other two, and was therefore assumed to be a candidate for re-design.

General observations in the conclusion were that as time progressed the system became 'stable' meaning that there was diminishing values in the growth and change rates.