Software metrics for the Boeing 777:
a case study

ROBERT LYTZ
Boeing Commercial Airplane Group, PO Box 3707 MS 02-MJ, Seattle, WA 98124-2207 USA

This article describes rapid, midstream introduction of elementary software metrics into a large engineering-development programme. It is presented as a case study for use by organizations considering deployment of software metrics. The 777 airplane, under development by The Boeing Company, will contain over two million lines of newly-developed source code. The 777 marks the first time The Boeing Company has applied software metrics uniformly across a new commercial-airplane programme. This was done to ensure simple, consistent communication of information pertinent to software schedules among Boeing, its software suppliers, and its customers - at all engineering and management levels. In the short term, uniform application of software metrics has resulted in improved visibility and reduced risk for 777 on-board software. Looking to the longer term, the metric information collected provides a basis for analysis of internal and supplier processes, and it will serve as a baseline for future commercial airplane programmes of The Boeing Company.

1. Introduction

The Boeing Commercial Airplane Group (BCAG) is the operating branch of The Boeing Company which designs, produces, and markets all Boeing commercial jet transports. Programme-wide application of software metrics is a new development within BCAG. This is in contrast to thoroughly-established use of software standards by BCAG and within the commercial aircraft industry generally (RTCA, 1992). The BCAG Avionics Computer Software Technical Standard (BCAG, 1990), for example, first released in 1983, is now undergoing its fifth revision. Such standards, along with the practices they govern, reflect the safety-critical nature of the software that is increasingly embedded in commercial aircraft.

Late in 1992, upper management of the 777 Development Programme of BCAG directed the programme to collect and apply software-metric data consistently. The goal was to facilitate management of software engineering within the programme. Prospective metrics were to communicate status, over time, for software development and testing and for resource utilization in target machines. They were to provide comparison of plans to actuals (and where applicable, to estimated totals) on a two-week cycle. They were to be useful, uniform, simple, and as unintrusive to developers as possible. They were to lend themselves to straightforward summary for use by engineers and by all levels of management. Finally, metric reporting requirements, metric-tracking tools, and participation of all systems were to be in place within three months.

This article describes the path taken to meet the demand for software metrics. In brief, the software metric programme that was introduced did meet the requirements levied by programme management. Its effects to date are outlined in Section 10. Overall, they can be summarized simply as improvement in communication and reduction of schedule risk.
It is worth examining the reasons communication and schedule have been important for the 777 Programme.

- The 777 Programme has introduced practices to enhance communication among BCAG, its customers, and its suppliers. Software metrics is one of these.
- Early detection of schedule problems was judged likely to improve cost and quality performance, in addition to schedule performance.
- The 777 Programme has pressed to start early integration of software using the new BCAG Software Integration Laboratory and Flight Controls Test Rig. This change potentially strains established software development processes.
- Each new 777 will contain over two million lines of new source code, a fourfold increase over the amount for the last significant BCAG development effort, the 747-400. (With commercial, off-the-shelf software and optional ‘buyer furnished equipment’ software included, the 777 total is upwards of four million lines.)
- Reliance on software for functionality and for integration of functionality has increased dramatically in the 777 over its predecessors.
- Aircraft manufacturers and their customers have traditionally taken delivery schedules very seriously.
- Software has increasingly become a critical-path element in BCAG development programmes.

Before the 777 software metric programme existed, managers and engineers in BCAG and supplier organizations used their own metric data. They continue to use measures different from, and in many cases more detailed than, those provided by the software-metric programme. It would be surprising if the situation were otherwise. The goal of the metric programme has not been to prevent people from using measures appropriate to their purposes. It has been to provide a simple, common basis of communication across a programme.

2. Historical perspective

The total of software-metric data gathered uniformly in the 777 Programme is a small fraction of the worthwhile information that could be collected under the banner of software metrics – for example, see Grady (1992). This fact is well recognized within the 777 Programme. In fact, members of the metrics team good-naturedly refer to their activity as ‘kindergarten metrics’, due to its elementary scope. Part of the limited scope is due to a desire not to impact 777 software suppliers. Historical perspective, though, suggests the elementary scope of 777 software metrics was a reasonable starting point in any case.

For example, a report published under NATO auspices in 1969 (Naur and Randell, 1969) briskly summarized 15 problems endemic at that time in software engineering. The list is reproduced as Table 1. On reflection, most software engineers would probably agree, with regret, that the NATO list is still timely.

Within BCAG, the Software Engineering Lessons Learned Archive contains numerous entries from the recent past which echo the 1968 NATO problems. The correspondence is most striking for items 1 (requirements), 2 (gaps between estimates and actuals), 5 (difficulty in monitoring progress), and 7 (poor communication). The highest correlation between the NATO list and the BCAG archive was for the ‘communication’ problem.

Anyone contemplating deployment of software metrics in their organization would probably do well to look at the NATO summary as they consider priorities for what to measure and what not to measure.