Extended Abstract

Software metrics have a reputation in industry of not being very useful. I believe one reason for this is that for most metrics one important aspect of them is usually not provided, namely the “entity population model”. In measurement theory, an entity population model defines the typical values for measurements from a metric for a given set of entities. Having these models is necessary in order to interpret the measurements. For example, without knowing the entity population model for the body temperature of humans we would not know that someone with a temperature of 40 degrees would be a cause for concern. In order for software metrics to be useful we need to have a good understanding of their entity population models.

In fact, we know very little about the entity population models for software metrics for anything but the simplest forms of measurements. We do have speculations, expectations, and even some theories as to what they should be, but there has been very little data published that can help us know which are correct and which are not. There are various reasons why we do not have this data. Often it is because we do not know how to measure something, reuse for example. Sometimes there is disagreement as to what to measure - there are more than 20 metrics for cohesion of object-oriented software for example. But it is also the case that we simply have not made a consistent and sustained attempt to make and report such measurements. The few empirical studies that do exist suffer from lacking sufficient detail to allow them to be reproduced, or are from such a small sample that little can be determined from them. This is the situation I and others are trying to change.

In this talk I will discuss my experience in measuring Java software. I have found that just measuring a large collection of software provides interesting insights as to the state of current software development. It seems that no matter what is measured, the results are usually interesting and sometimes surprising. I will present some of these results. I will also discuss the issues involved in doing this kind of research. One such issue is making measurements that are reproducible. To address this issue, I advocate basing software metrics research on the use of standard software corpora, that is, creating collections of software whose contents are well-defined. However creating such a corpus is not just a matter of downloading stuff off the ‘net. I discuss some of the difficulties that arise in developing a corpus of open source Java software.