The Challenges of Measuring Productivity in Software Engineering

Leo McClean

Trinity College Dublin

# Introduction

The advent of software engineering came at the dawn of the 1960s, when most of the modern world realised its problems could be simplified by embedded systems that simplified large calculations and operations involving machinery. For example, with space travel, complex calculations needed to be performed constantly by an on-board computer for a shuttle to correctly exit and re-enter the atmosphere. This ‘software engineering’, where the term was born in Cape Canaveral, involved first designing the routines by hand, then moulding them into punch cards to be tested. When the program was proved to be working, it was hard wired into a shuttle, quite literally, using wires threaded through and around magnetic cores (through signalling a binary 1, and around signalling a binary 0) (McMillan, 2015). It wasn’t until years after the mostly successful Apollo launches that software evolved from binary, into the first high-level languages such as COBOL and BASIC. In the thirty-year period from the 60s until the 90s, we leapt from punch-cards to a variety of languages, some imperative or procedural – object-oriented or functional. And within this rapid development of new technologies to better facilitate the creation and application of software, we ultimately failed to properly design any method to successfully determine the required input of labour for designing a system, the output we can achieve with a certain number of programmers, or even an effective measurement of productivity in software engineering. And although we have had another 30-year period of innovation and advancement since 1990, we still struggle to adequately measure the process of software engineering. In this essay we will explore what data we can measure as software engineers, why it is difficult to transform this data into a measure of productivity, how companies today are improving their development cycles, and how, as we look to the future, software engineering may become a fully measurable process eventually.

# The Data of Software Engineering