Jens Knoop, Uwe Zdun (Hrsg.): Software Engineering 2016,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016 53

From Aristotle to Ringelmann: A large-scale analysis of team productivity and coordination in Open Source Software projects

Ingo Scholtes[1], Pavlin Mavrodiev[2], Frank Schweitzer[3]

Abstract: The productivity of software development teams, i.e., how their size relates to their output, is an important question for project management. Most studies suggest that teams become less productive as they grow larger, a phenomenon paraphrased as *Brooks' law* in software engineering and as *Ringelmann effect* in social psychology. Conversely, a recent study suggests that the productivity of teams in OSS projects *increases* as they grow larger. Attributing it to synergetic effects, this was linked to the Aristotelian quote that "the whole is more than the sum of its parts". Using data on 58 OSS projects with 580,000 commits by 30,000 developers, we perform a large-scale analysis of productivity in development teams. We confirm the negative relation previously found by software engineering research, providing quantitative evidence for the Ringelmann effect. Taking a network perspective on developer-code associations, we investigate mechanism behind this effect and show that the magnitude of the productivity decrease is related to the growth dynamics of coordination networks.

Most of today's software projects are so complex that they cannot be developed by a single person, instead requiring large teams of collaborating developers. This necessity of large teams raises a simple, yet important question: How productive is a team of developers compared to a single developer? Or, in other words: How much time do n developers need to finish a project compared to the time taken by a single developer? This question is of significant importance not only for project management but also for the development of cost estimation models for software engineering processes. One may naively assume that the productivity of individual team members is *additive*, i.e., that, compared to the time taken by a single developer, n developers will speed up the development time by a factor of n. However, this misses out two important factors that can give rise to a non-additive scaling of productivity. First, the collaboration of developers in a team can give rise to *synergy effects*, which result in the team being *more productive* than one would expect from adding up individual productivities of its members. Under this assumption, the average output per team member can be *increased* by adding developers to the team, a fact that has recently been related to Aristotle's quote that *"the whole is more than the sum of its parts"* [SMG14]. A second, contrary factor that influences the productivity of developer teams is the communication and coordination overhead which is likely to increase as teams grow larger. In particular, this can lead to situations where the average output per team member *decreases* as the size of the team is increased. Studies showing that growing team sizes negatively affect productivity can be traced back to early studies of Maximilian Ringelmann [Ri13]. In the context of software engineering, it can be related

[1] Chair of Systems Design, ETH Zürich, CH-8092 Zürich, Switzerland, ischoltes@ethz.ch
[2] Chair of Systems Design, ETH Zürich, CH-8092 Zürich, Switzerland, ischoltes@ethz.ch
[3] Chair of Systems Design, ETH Zürich, CH-8092 Zürich, Switzerland, ischoltes@ethz.ch

to "Brook's" law of software project management, which states that *"adding manpower to a late software project makes it later"* [Jr75].

Using a data set covering the history of 58 Open Source Software (OSS) projects hosted on the social coding platform GITHUB, in [SMS15] we quantitatively address the question how the size of a software development team is related to their productivity. Based on a time-slice analysis of more than 580,000 commit events over a period of more than 14 years, we analyse the output of projects in terms of code and study how their time-varying productivity relates to the number of active software developers. Using the *distribution of inter-commit times*, we first identify reasonable time windows for the definition of team size and the analysis of commit activities in OSS projects. We measure the contributions of individual commits based on a microscopic, textual analysis of commit contents. Our analysis confirms the intuition that the actual contribution of commits exhibits a large variation, thus requiring an analysis of commit contents rather than the mere number of commits. We define a measure for the contribution of developers which is based on the *Levenshtein edit distance* [Le66] between consecutive versions of source code files. Using this fine-grained measure, we quantitatively show that in all of the studied OSS projects the average productivity of developers decreases as the team size increases, thus providing quantitative evidence for the Ringelmann effect. Finally, we take a network perspective on the association between developers and the source code files they have edited. Aiming at a file-based and language-independent first-order approximation for coordination structures, we analyse the growth dynamics of co-editing networks constructed from repository data. For all projects in our data set, we observe a *super-linear* growth of co-editing networks, which can be seen as one potential mechanism behind the observed Ringelmann effect. We argue that both our results as well as our methodology are useful to refine and calibrate existing software development cost models based on empirical data from software development repositories.

References

[Jr75] Jr., Frederick P. Brooks: The Mythical Man-Month. Addison-Wesley, 1975.

[Le66] Levenshtein, Vladimir I: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet physics doklady. volume 10, p. 707, 1966.

[Ri13] Ringelmann, Maximilan: Recherches sur les moteurs animes: Travail de l'homme. Annales de l'Institut National Agronomique, 12(1):1–40, 1913.

[SMG14] Sornette, Didier; Maillart, Thomas; Ghezzi, Giacomo: How Much Is the Whole Really More than the Sum of Its Parts? 1 + 1 = 2.5: Superlinear Productivity in Collective Group Actions. PLoS ONE, 9(8):e103023, 08 2014.

[SMS15] Scholtes, Ingo; Mavrodiev, Pavlin; Schweitzer, Frank: From Aristotle to Ringelmann: A large-scale analysis of team productivity and coordination in Open Source Software projects. Empirical Software Engineering, 2015. accepted for publication on September 23 2015, to appear.