A Quantitative Study of the Ringelmann Effect in Software Development Teams

Ingo Scholtes
Chair of Systems Design, ETH Zürich
with Pavlin Mavrodiev and Frank Schweitzer
Computational Social Science

- **data mining & simulation**
  - extract knowledge from (noisy) data on social systems
  - micro- and macro-level simulations of social systems

⇒ **new ways to test hypotheses drawn from sociological theory**
Computational Social Science

- **Computer Science**
  - data mining & simulation
    - extract knowledge from (noisy) data on social systems
    - micro- and macro-level simulations of social systems

- **Social Sciences**
  - social aspects in system design
    - regain control of socially-embedded (cyber-physical) systems
    - better understand & manage development processes

⇒ new ways to test hypotheses drawn from sociological theory
⇒ improve modeling, design and control of technical systems
Context: Collaborative Software Engineering

**technical aspects:** programming language, architecture, code complexity, support tools...

**social aspects:** development process, team composition, communication, coordination, ...
Challenge: Understanding team productivity

software economics: how productive are software development teams?

“Adding manpower to a late software project makes it later.”

Fred Brooks (1975)

Fred Brooks

image: CC-BY-SA SD&M
Challenge: Understanding team productivity

- **software economics**: how productive are software development teams?

  “Adding manpower to a late software project makes it later.”

  *Fred Brooks (1975)*

- **Ringelmann effect** in social psychology: larger teams are less productive

  Maximilien Ringelmann (1861-1931)

  ![image: CC-by-SA Bart Derksen](http://www.sg.ethz.ch)
Challenge: Understanding team productivity

**software economics:** how productive are software development teams?

> “Adding manpower to a late software project makes it later.”
> Fred Brooks (1975)

**Ringelmann effect** in social psychology: larger teams are less productive

1. motivational factors: “social loafing”
2. overhead of coordination

Systems Design
http://www.sg.ethz.ch

A Quantitative Study of the Ringelmann Effect in Software Development Teams
Ingo Scholtes
CSS Wintersymposium Cologne, DE 01/12/2016 | 4 / 8
Challenge: Understanding team productivity

**Software economics:** how productive are software development teams?

> "Adding manpower to a late software project makes it later."
> 
> *Fred Brooks (1975)*

**Ringelmann effect** in social psychology: larger teams are less productive

1. **Motivational factors:** "social loafing"
2. **Overhead of coordination**

**Research questions:**

1. Does the Ringelmann effect apply to Open Source communities?
2. Can we detect the underlying mechanisms?

---

Empirical Software Engineering
Vol. 21, No. 2, April 2016

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

Ingo Scholtes¹ - Pavlin Mavrodiev¹ - Frank Schmitter¹

© Springer Science+Business Media New York 2015

Abstract Complex software development projects rely on the contribution of teams of developers, who are required to collaborate and coordinate their efforts. The productivity of such development teams, i.e., how their size is related to the produced output, is an important consideration for project and schedule management as well as for cost estimation. The majority of studies in empirical software engineering suggest that, due to coordination overhead, teams of collaborating developers become less productive as they grow in size. This phenomenon is commonly paraphrased as Brooks’ law of software project management, which states that “adding manpower to a software project makes it later.” Outside software engineering, the non-additive scaling of productivity in teams is often referred to as the *Ringelmann effect*, which is studied extensively in social psychology and organizational theory. Conversely, a recent study suggested that in Open Source Software (OSS) projects, the productivity of developers *increases* as the team size grows in size. Attributing it to collective synergistic effects, this surprising finding was linked to the Aristotelian quote that “the whole is more than the sum of its parts.” Using a data set of 58 OSS projects with more than 580,000 commits contributed by more than 30,000 developers, in this article we provide a large-scale analysis of the relation between size and productivity of software development teams. Our findings confirm the negative relation between team size and productivity previously suggested by empirical software engineering research, but providing quantitative evidence for the presence of a strong Ringelmann effect. Using fine-grained data on
Data Science approach: commit log analysis

data: GHTorrent GitHub dump (~1.5 TB)

identified **58 most active OpenSource projects**

580,000 commits
30,000 developers
> 10 years of commit history
Data Science approach: commit log analysis

**data:** GHTorrent *GitHub dump* (~1.5 TB)

identified **58 most active OpenSource projects**
580,000 commits
30,000 developers
> 10 years of commit history

Q1: What is the team?
>> active developers in reasonable time window
**Data Science approach: commit log analysis**

**data:** GHTorrent GitHub dump (~1.5 TB)

identified **58 most active OpenSource projects**
580,000 commits
30,000 developers
> 10 years of commit history

Q1: What is the team?
>> active developers in reasonable time window

Q2: What is productivity?
>> Levenshtein distance between commits
Data Science approach: commit log analysis

Systems Design
http://www.sg.ethz.ch

A Quantitative Study of the Ringelmann Effect in Software Development Teams
Ingo Scholtes
CSS Wintersymposium Cologne, DE 01/12/2016
Data Science approach: commit log analysis

```
range = [1:n]
result = {}
for i in range:
   n1 = str(s[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
range = [1:m]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
range = [1:100]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=0
```

```
range = [1:200]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = n1
   result[key]=0
```

```
x = 'ASC'
dat = []
dat = getData()
dat = sort(dat)
return clean(dat)
```

```
x = 'DESC'
dat = []
dat = getData()
dat = sort(dat,x)
c = clean(dat)
return s
```

```
range = [1:200]
result = { 0:0 }
for i in range:
   n1 = int(t[0])
   n2 = int(t[1])
   l.append(n1)
   l.append(n2)
   key = n2
   result[key]=1
```

```
```

<table>
<thead>
<tr>
<th>code contributions</th>
<th>45 characters</th>
<th>17 characters</th>
<th>29 characters</th>
<th>16 characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>team size: 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean contribution:</td>
<td></td>
<td></td>
<td></td>
<td>26.75</td>
</tr>
</tbody>
</table>

Systems Design

http://www.sg.ethz.ch

A Quantitative Study of the Ringelmann Effect in Software Development Teams

Ingo Scholtes

CSS Wintersymposium Cologne, DE 01/12/2016 | 5 / 8
Data Science approach: commit log analysis

```
range = [1:n]
result = {}
for i in range:
   n1 = str(s[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
range = [1:m]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
range = [1:100]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = n1
   result[key]=0
```

```
range = [1:200]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = n1
   result[key]=0
```

```
x = 'ASC'
dat = []
dat = getData()
dat = sort(dat, x)
c = clean(dat)
return s
```

```
x = 'DESC'
dat = []
dat = getData()
dat = sort(dat, x)
c = clean(dat)
return s
```

```
range = [1:200]
result = { 0:0 }
for i in range:
   n1 = int(t[0])
   n2 = int(t[1])
   l.append(n1)
   l.append(n2)
   key = n2
   result[key]=1
```

```
x_list = []
dat = []
dat = getData()
dat = sort(dat)
dat = clean(dat)
return dat
```

```
v_list = []
dat = []
dat = getData()
dat = sort(dat)
dat = clean(dat)
return clean(dat)
```

```
r = [1:n]
result = {}
for i in range:
   n1 = str(s[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
r = [1:n]
result = {}
for i in range:
   n1 = str(s[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
r = [1:m]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
r = [1:m]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1
```

```
r = [1:100]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = n1
   result[key]=0
```

```
r = [1:100]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = n1
   result[key]=0
```

```
r = [1:200]
result = { 0:0 }
for i in range:
   n1 = str(t[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = n1
   result[key]=0
```

```
r = [1:200]
result = { 0:0 }
for i in range:
   n1 = int(t[0])
   n2 = int(t[1])
   l.append(n1)
   l.append(n2)
   key = n2
   result[key]=1
```

Team size: 4
Mean contribution: 26.75
Data Science approach: commit log analysis

- Analysis across 58 projects
- Robust log-linear regression yields negative relation

$$\alpha = -0.86 \pm 0.02, r^2 = 0.25$$

We confirm and quantify the Ringelmann effect in Open Source communities!
Data Science approach: commit log analysis

**Project A**

Productivity coefficient
\[ \alpha = -0.12 \pm 0.07, \ r^2 = 0.23 \]

**Project B**

Productivity coefficient
\[ \alpha = -1.9 \pm 0.2, \ r^2 = 0.47 \]
Coordination networks and productivity

network of coordination requirements

Commit History

commit 1
range = [1:n]
result = {}
for i in range:
   n1 = str(s[0])
   n2 = str(t[1])
   l.append(n1)
   l.append(n2)
   key = (n1, n2)
   result[key]=1

commit 2
dat = getData()
dat = sort(dat)
dat = clean(dat)
return dat

commit 3
v_list = []
dat = []
dat = getData()
dat = sort(dat)
dat = clean(dat)
return clean(dat)

commit 4
v_list = []
dat = []
dat = getData()
dat = sort(dat, x)
c = clean(dat)
return s

commit 5
range = [1:200]
result = { 0:0 }
for i in range:
   n1 = int(t[0])
   n2 = int(t[1])
   l.append(n1)
   l.append(n2)
   key = n2
   result[key]=1

Commit References

- Ingo Scholtes
- CSS Wintersymposium Cologne, DE 01/12/2016
Coordination networks and productivity

**Project A**

- Mean degree
- Densification coefficient: $\beta = 0.007 \pm 0.005, r^2 = 0.036$

**Project B**

- Densification coefficient: $\beta = 0.035 \pm 0.007, r^2 = 0.24$

---

**Systems Design**

http://www.sg.ethz.ch

---

A Quantitative Study of the Ringelmann Effect in Software Development Teams

Ingo Scholtes

CSS Wintersymposium Cologne, DE 01/12/2016 | 6 / 8
key finding:

statistically significant relation between densification of coordination networks and decrease in productivity as teams grow in size
Take-away messages

1. We provide quantitative evidence for the Ringelmann effect in Open Source software development teams.

2. We uncover coordination overhead as one mechanism behind the observed scaling behavior of team productivity.

3. We show that social science theories provide actionable insights into software engineering processes.

Abstract

Complex software development projects rely on the contribution of teams of developers, who are required to collaborate and coordinate their efforts. The productivity of such development teams, i.e., how their size is related to the produced output, is an important consideration for project and schedule management as well as for cost estimation. The majority of studies in empirical software engineering suggest that due to coordination overhead, teams of collaborating developers become less productive as they grow in size. This phenomenon is commonly paraphrased as Brooks’ law of software project management, which states that “adding manpower to a software project makes it later”. Outside software engineering, the non-additive scaling of productivity in teams is often referred to as the Ringelmann effect, which is studied extensively in social psychology and organizational theory. Conversely, a recent study suggested that in Open Source Software (OSS) projects, the productivity of developers increases as the team grows in size. Attributing it to collective synergetic effects, this surprising finding was linked to the Aristotelian quote that “the whole is more than the sum of its parts”. Using a data set of 58 OSS projects with more than 580,000 commits contributed by more than 30,000 developers, in this article we provide a large-scale analysis of the relation between size and productivity of software development teams. Our findings confirm the negative relation between team size and productivity previously suggested by empirical software engineering research, thus providing quantitative evidence for the presence of a strong Ringelmann effect. Using fine-grained data on...
Take-away messages

1. we provide quantitative evidence for the **Ringelmann effect** in Open Source software development teams

2. we uncover **coordination overhead** as one mechanism behind the observed scaling behavior of **team productivity**
Take-away messages

1. We provide quantitative evidence for the Ringelmann effect in Open Source software development teams.

2. We uncover coordination overhead as one mechanism behind the observed scaling behavior of team productivity.

3. We show that social science theories provide actionable insights into software engineering processes.
Thank You!

I Scholtes, M Strohmaier, F Schweitzer: *Is Computer Science becoming a Social Science?*, Under review, October 2016


@ingo_s

http://www.ingoscholtes.net

ischoltes@ethz.ch