


## **0. our first ideas**

# first thoughts

- collaboration among users
- communities formation
- recommender
- focus on code modification/code review/else
- focus on projects from one category and analyze the practices of dev
- starring behavior? i.e. what people star and to what extent it is useful
- mine communication patterns using issues and commits

# first thoughts

- **collaboration among users**
- communities formation
- recommender
- focus on code modification/code review/else
- focus on projects from one category and analyze the practices of dev
- **starring behavior?** i.e. what people star and to what extent it is useful
- mine **communication patterns** using issues and commits 

# Communication patterns on GitHub

but we are ok with stackoverflow as well



# **1. research questions**

# research questions

**RQ1:** How communication patterns differ between users and developers?

**RQ2:** What place do the newcomers hold and how they build their reputation and extend their social network?

(что происходит с новенькими, one-commit developers, всякое такое)

**RQ3:** Who gets more negative comments, and why?

## 2. hypotheses

# hypotheses

**H1:** contributions of developers differ from each other

**H2:** communication patterns differ between users and developers

**H3:** newcomers can receive more hate?

**H4:**





### 3. a bit of literature review



# literature review

1. Ortu, Marco & Hall, Tracy & Marchesi, Michele & Tonelli, Roberto & Bowes, David & Destefanis, Giuseppe. (2018). Mining Communication Patterns in Software Development: A GitHub Analysis. 10.1145/3273934.3273945.
2. Yu, Yue & Yin, Gang & Wang, Huaimin & Wang, Tao. (2014). Exploring the patterns of social behavior in GitHub. 31-36. 10.1145/2666539.2666571.

## literature review /

### 1. Mining Communication Patterns in Software Development: A GitHub Analysis

**Данные:** ~650K comments from 130K issues of 64K contributors, GHTorrent dataset

#### Методы:

- выделили эмоции (love, joy, anger, sadness) и politeness
- выделяют несколько групп пользователей и разработчиков
- используют классификатор Ortu + Murgia, чтобы определять эмоции (был обучен специально на Software Engineering domain)
- для politeness: использовали библиотеку, обученную на данных со stackoverflow

#### Вопросы:

- Do developers and users communicate differently with respect to fan-in, fan-out, emotions (love, joy, anger, sadness), and politeness?
- What are the differences in communication typology between one-commit and multi commit developers?

## literature review/

### 1. Mining Communication Patterns in Software Development: A GitHub Analysis

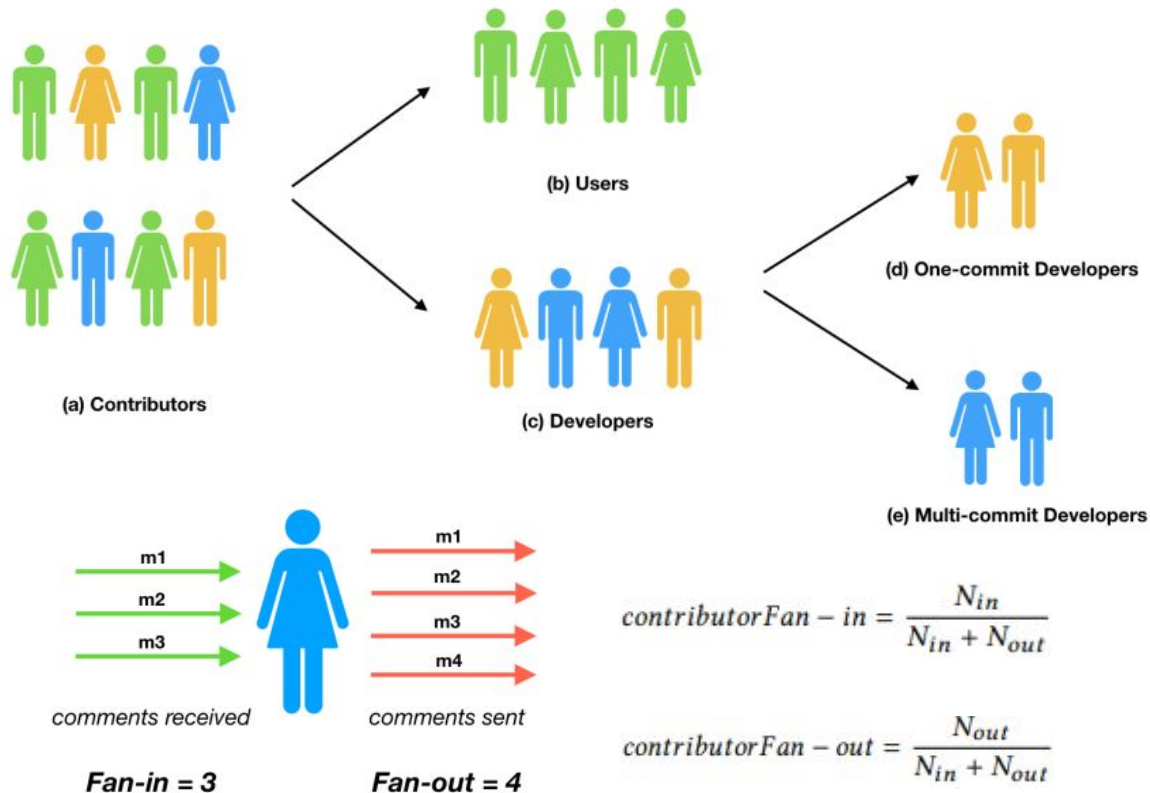
**User** - a project contributor without commits

**Developer** - a contributor with commits on the source code

**One-commit & Multi commit:** comes from "One-Time Code Contributor" - некоторые коммитят только 1 раз по нескольким причинам

**Fan-in** - сколько комментариев contributor получает к своим issue

**Fan-out** - метрика, отражающая тенденцию публиковать комментарии и к другим issues



## literature review /

### 1. Mining Communication Patterns in Software Development: A GitHub Analysis

#### Результаты:

- пользователи и разработчики общаются по разному

#### RQ1:

- комментариев от разработчиков меньше, чем комментариев от пользователей
- разработчикам отвечают больше
- politeness комментариев от разработчиков оказалась **ниже**, чем пользователей,
- разработчики получают более вежливые комменты
- у пользователей более высокий fan-out и более низкий fan-in
- politeness комментов пользователей выше, чем у разработчиков
- более низкий уровень politeness наблюдается в комментах, получаемых пользователями

## literature review/

### 1. Mining Communication Patterns in Software Development: A GitHub Analysis

#### RQ2:

- OCDs are more active in posting issues and more polite when commenting issues.
- one-commit developers общаются больше как пользователи
- politeness MCD ниже, чем у OCD
- разработчики комментируют OCDs' issues менее вежливо, и наоборот
- issues с менее вежливыми комментариями и commits с низким уровнем politeness имеют меньшие шансы на то, чтобы их приняли (to be merged into the main project)

## 2. Exploring the patterns of social behavior in GitHub

**Данные:** a dataset of activities of 1,838,805 users

**Методы:** network analysis

**Вопросы:**

- What are the differences between the growth modes of GitHub and traditional OSS communities, and is there any sociological theory that supports the special growth mode of GitHub?
- Whether or not the social connections among developers form some distinctive behavior patterns in GitHub, and if it is true, what are these patterns?

# literature review/

## 2. Exploring the patterns of social behavior in GitHub

### Результаты (RQ1 - Growth mode)

supporting vs. human factor

Diffusion of Innovations theory: if there were 2.5% innovators and 13.5% early adopters hosting their projects on GitHub and promoting to others, the “tipping point” would be achieved.

Why?

- the effect of leader
- herd behavior

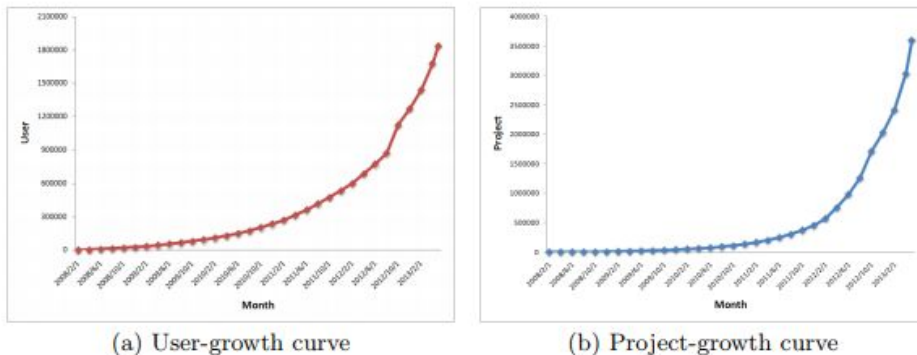


Figure 1: The growth figure of Github

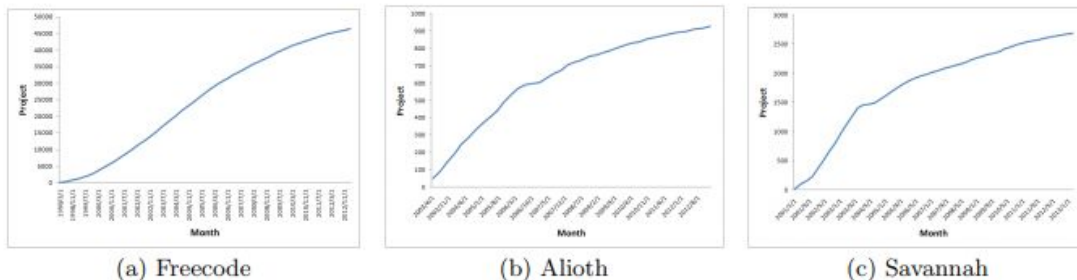


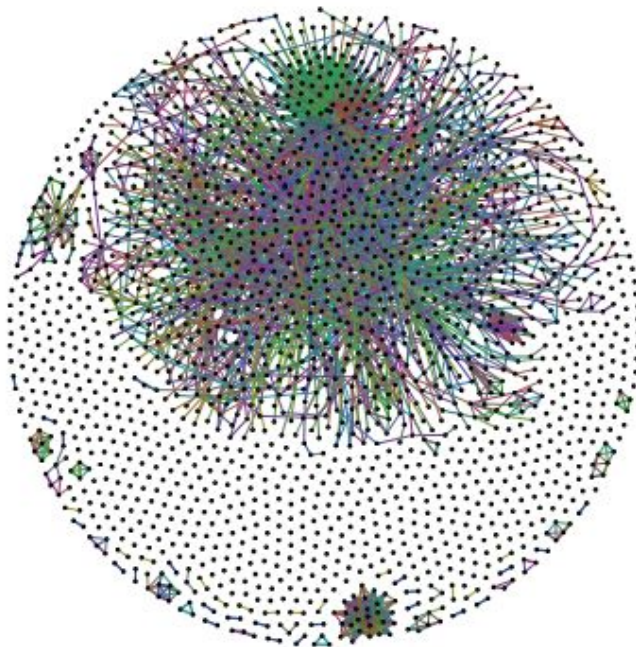
Figure 2: The growth figure of traditional OSS community



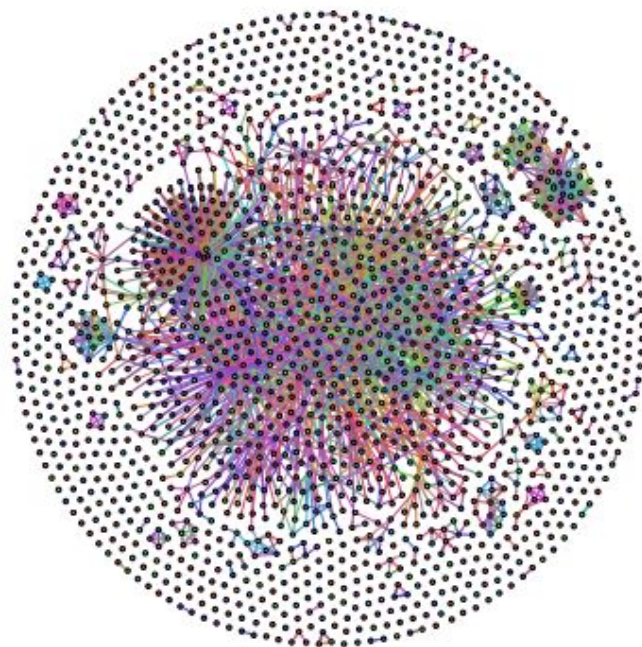
## 2. Exploring the patterns of social behavior in GitHub

### Результаты

(RQ2 - Social networks  
- Follow-network)



(a) 2012-08 subset



(b) 2012-09 subset

Figure 3: Two examples of follow-network

## 2. Exploring the patterns of social behavior in GitHub

### Результаты

(RQ2 - Patterns of social behavior)

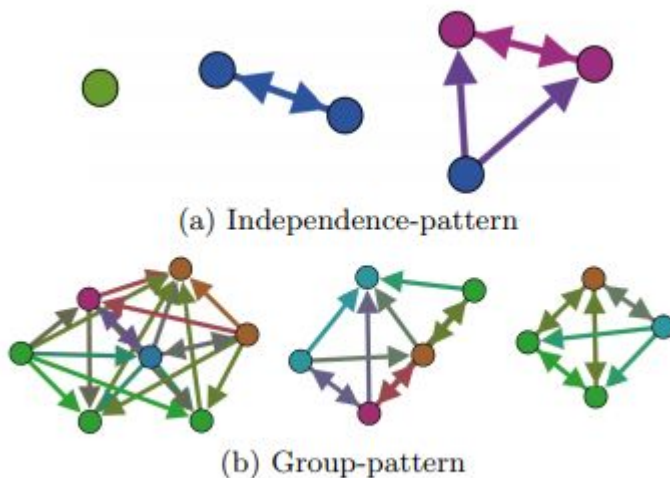


Figure 4: The typical patterns in the isolated part

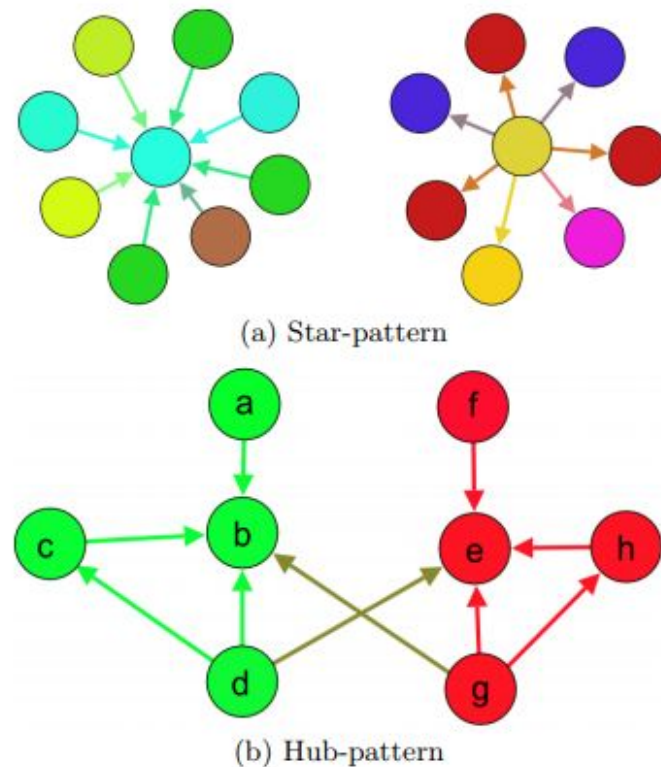


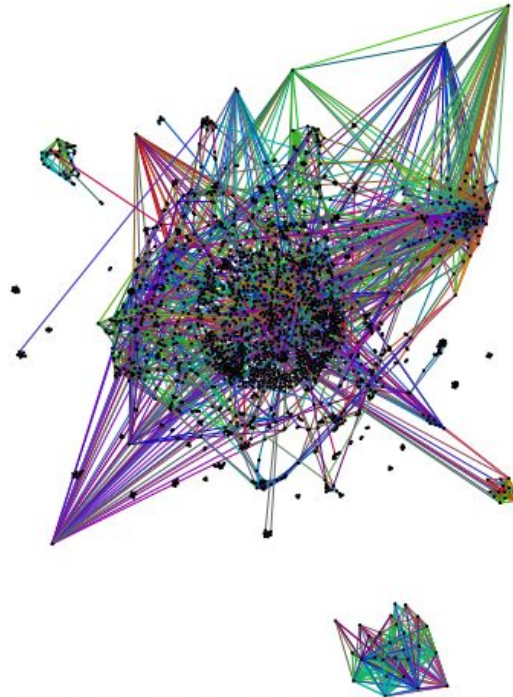
Figure 7: The typical patterns in the interlaced part

# literature review /

## 2. Exploring the patterns of social behavior in GitHub

### Результаты

(RQ2 - Patterns of social behavior)



(b) The Network redrawn by *Force Atlas 2* algorithm

Figure 5: The redrawn follow-network of 2012-08 subset

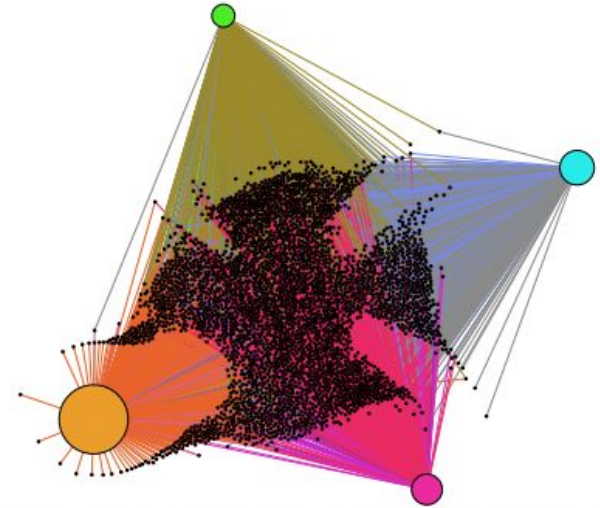
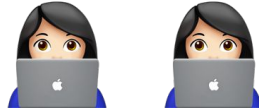


Figure 6: Community structures in the *follow-network* of 2012-08 subset

## 4. methods



# methods

- › собрать комментарии к issues & commits (более-менее крупных проектов)
- › попробовать найти там различия
- › ещё можно кластеризовать!
- › собрать данные о самих юзерах

# а как?

- › [topic modeling](#), чтобы понять, о чем (и как) пишут
- › [clustering](#), чтобы выделить группы/типы пользователей
- › построить [сети](#) и посмотреть на связи между пользователями (например, отвечают друг другу, фолловят)

