Evolução da Distribuição de Conhecimento de Software em Projetos *Open Source*

Talita Santana Orfanó

Orientadoras: Mariza A. da Silva Bigonha e Kecia Aline M. Ferreira

Universidade Federal de Minas Gerais

Programa de Pós-Graduação
Ciência da Computação
UFMG

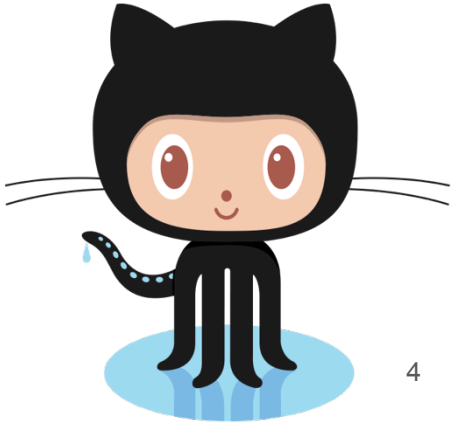
À medida que o software evolui, há uma tendência de ocorrer uma maior degradação de sua qualidade interna [Lehman,1996]

Diversos estudos apontam uma significativa relação entre os **fatores humanos** e a **qualidade de software**

Contexto

- A **falta de conhecimento** dos desenvolvedores sobre o código fonte em que fazem manutenção

- Popularização de **projetos *open source***
  - Desenvolvimento colaborativo
  - Mineração de dados em repositório públicos
    - **GitHub**, BitBucket, GitLab



Contexto

- **Desenvolvedores *heroes***
  - Aqueles poucos que efetuaram a maior parte das contribuições do projeto
  - Também conhecidos como *core developer*, *major*, *hero* ou *main developer*

- **Desenvolvedores periféricos**
  - A maioria dos desenvolvedores do projeto
    - Realizam poucas contribuições
  - Em geral, atuam na evolução de uma pequena funcionalidade ou na correção de *bugs*

Problema

Poucos trabalhos estudam como **a distribuição do conhecimento dos desenvolvedores evolui** no decorrer do ciclo de vida do software

- O conhecimento sobre a evolução da **atuação** dos desenvolvedores ao longo da existência de projetos *open source* ainda não é amplo

Objetivo

Investigar, por meio de estudos de caso, como a distribuição de conhecimento do código evolui ao longo do ciclo de vida do projeto

Questões de Pesquisa (QP)

QP1. O conhecimento do software se difunde ao longo de seu ciclo de vida?

QP2. Os desenvolvedores considerados *heroes* nas primeiras versões permanecem *heroes* durante a evolução do projeto?

QP3. A concentração do conhecimento é impactada pela quantidade total de contribuições do projeto?

Questões de Pesquisa (QP)

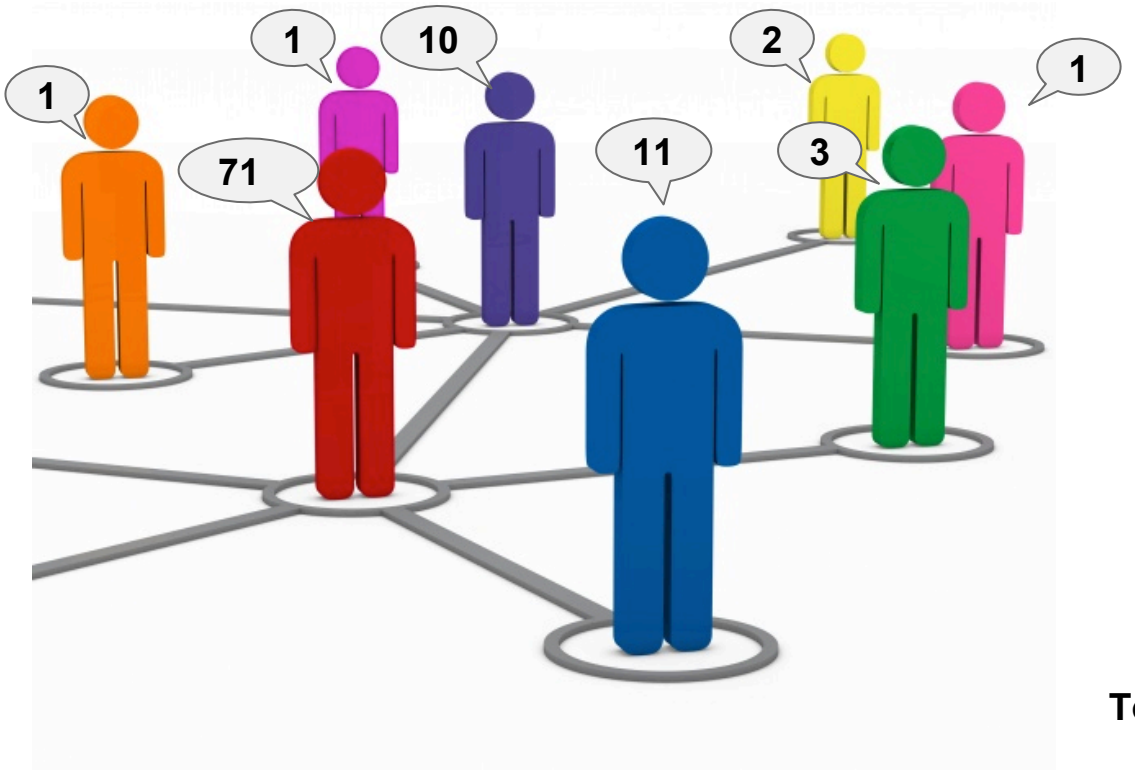
QP4. Como se dá a atuação dos desenvolvedores periféricos nos projetos?

QP5. Como ocorre a disseminação de conhecimento do software entre os desenvolvedores?

Propriedade de Código

- **Principais trabalhos**
  - DOK (*degree-of-knowledge*)
    - Fritz et al., 2014
  - Linhas alteradas como fator de medição da autoria
    - Rahman & Devanbu, 2011
  - Proprietário do módulo
    - **Bird et al., 2011 e Foucault et al., 2014**
  - Métrica *Truck Factor*
    - Torchiano et al., 2011 e Avelino et al., 2016

Propriedade de Código



Total de 100 *commits*

Propriedade de Código

Proprietário do projeto
Valor da métrica: 0,71

Total de 100 *commits*

Metodologia

Coleta de Dados

Bootstrap, Django, Elasticsearch, Panda e Spring Boot

+119.500

commits

+6.900

autores

+850

releases

Coleta de Dados

- **Granularidade por *commits***
  - Não há um consenso na literatura sobre a melhor maneira de avaliar autoria e contribuição, especialmente em projetos *open source*
  - Custo para processar, armazenar e analisar grande gama de informações geradas por coletas de granularidades menores é muito alto
  - Vários trabalhos relacionados utilizaram a granularidade de *commits*

Coleta de Dados

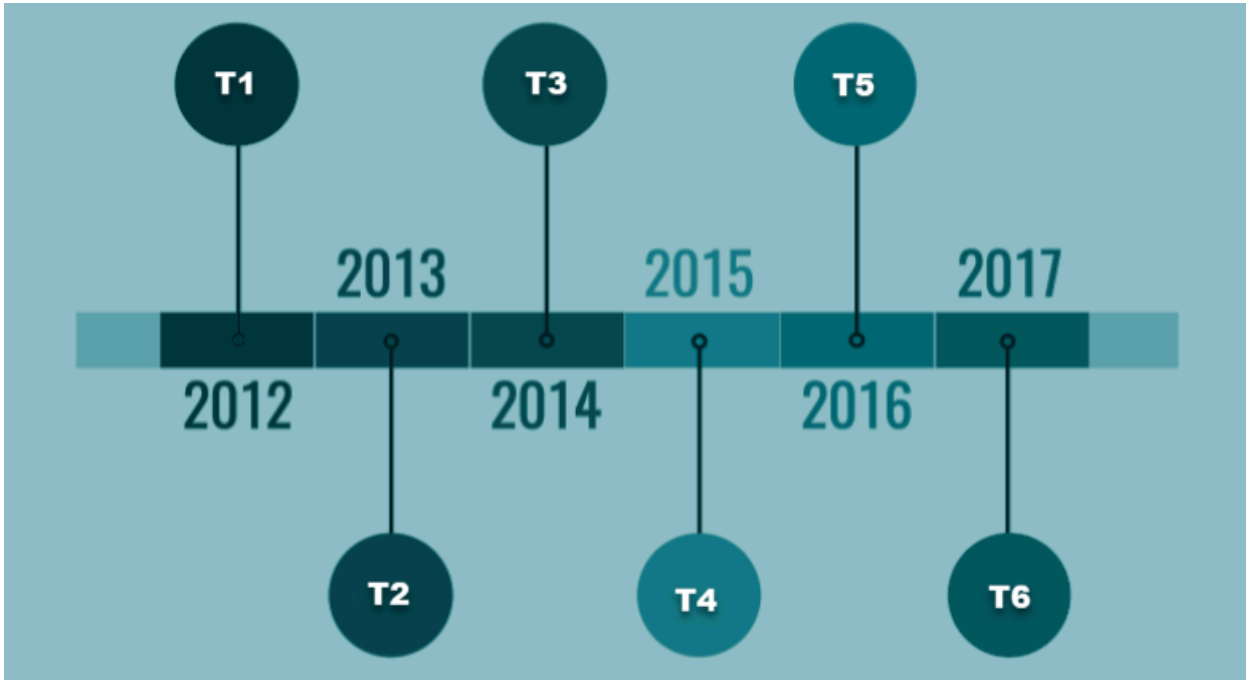
- **Ferramenta para coleta**
  - Dados coletados utilizando a API REST do GitHub
  - Facilidade de utilização e ampla documentação
  - Completude das informações coletadas diariamente dos repositórios originais
  - Resultados obtidos no formato *JSON*

Coleta de Dados

- **Solução construída**
  - Construídos *scripts* em *Python*
    - Para coleta dos dados via API REST do GitHub
  - Dados coletados armazenados
    - MongoDB e arquivos *CSV*
  - Conjunto de dados coletados
    - código identificador do *commit* (*sha*),
    - informações do contribuidor (nome, e-mail e *login*)
    - data do *commit* e nome do repositório
  - Primeiro *commit* de cada projeto até 2019/1

Coleta de Dados

Janela de Tempo



Coleta de Informações

- **Contribuições por autor**
  - Total de *commits* efetuados por cada autor
  - Identificar quais autores mais contribuíram no repositório desde a sua criação
  - Compreender como é distribuído o total de *commits* entre os contribuidores dos projetos

Coleta de Informações

- **Contribuições por período de tempo T**
  - Total de *commits* efetuados por cada autor em cada período T
  - Compreender como as contribuições cresceram ou diminuíram no decorrer do ciclo de vida do projeto
  - Entender o processo de distribuição de conhecimento entre os autores no ciclo evolutivo
  - Analisar o processo de evolução dos autores considerados *heroes* do projeto

Desambiguação de Autores

- Em projetos *open source* é frequente a existência de contribuidores ambíguos
- Usuários que no mesmo projeto utilizam nomes ou e-mails diferentes
    - Identificados erroneamente como sendo duas ou mais pessoas distintas
- Essas particularidades, se não tratadas, podem impactar os resultados dos estudos

Desambiguação de Autores

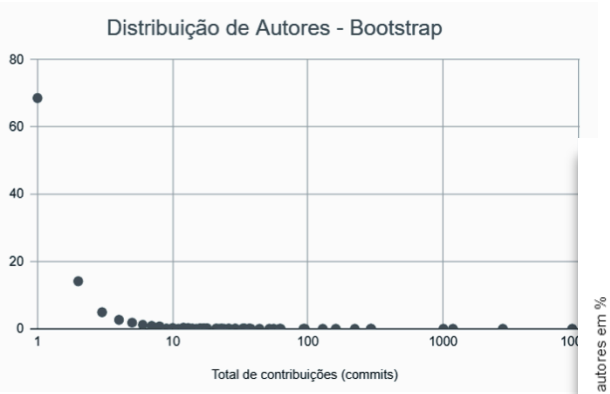
- **Por login**
- **Nome e e-mail - Por heurísticas**
    - Algoritmo de similaridade Bird [Bird et al., 2006]
        - Alto coeficiente de revocação (*recall*) - 100%
        - Baixa precisão em projeto de grande porte - ~30%
- **Nome e e-mail - Manual**
    - Solução adotada

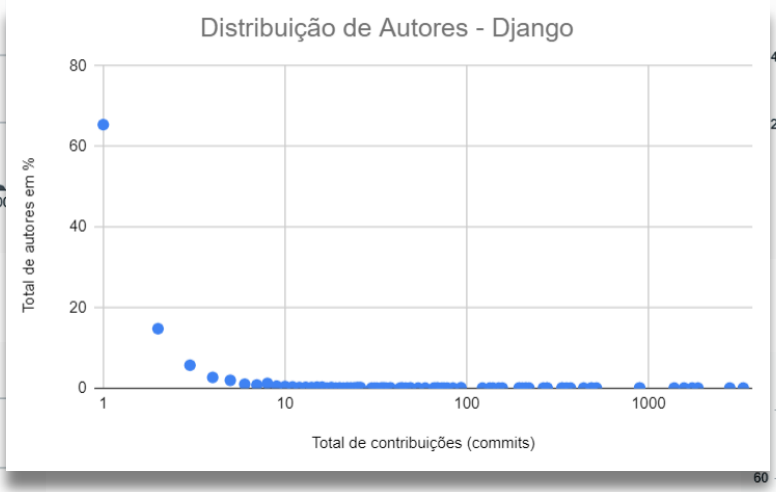
Desambiguação de Autores

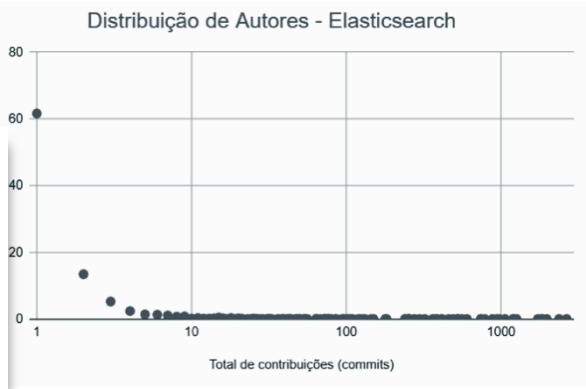
- **Por nome / e-mail - Manual**
  - Processo custoso, porém com maior precisão e confiabilidade
    - Autores ambíguos que trariam **grande impacto** nos resultados: *heroes*, médios e pequenos que possuíam grande número de contribuições
    - Nomes e e-mails ordenados alfabeticamente e comparados entre si
      - Nomes ou e-mails iguais foram unificados
      - Nomes ou e-mails semelhantes foram avaliados individualmente com o auxílio de pesquisas externas

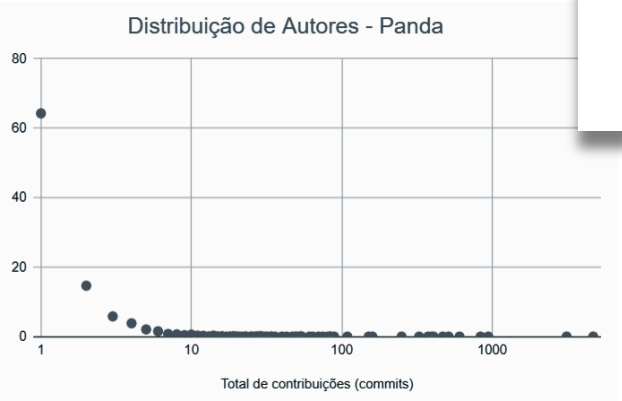
Evolução da Distribuição do Conhecimento Entre os Desenvolvedores

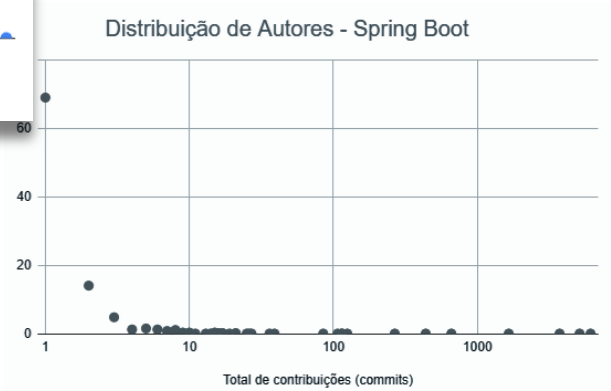
Distribuição da População de Autores











Distribuição da População de Autores

- A maior parcela de autores realizaram poucas contribuições
  - Ao menos 80% dos autores realizaram apenas 2 contribuições
- A distribuição da população de autores possui cauda longa
  - Quantidade de autores diminui exponencialmente e torna-se mais esparso
  - Poucos desenvolvedores realizam uma quantidade alta de contribuição nos repositórios
- Conhecimento concentrado em poucos contribuidores
- Comportamento homogêneo em todos os projetos do *data set*
- Resultado em concordância com trabalhos prévios
  - Avelino et al., 2019b; Greiler et al., 2015; Foucault et al., 2014; Ricca & Marchetto, 2010

Caracterização da População de Autores

- Diferença na atuação dos autores em um projeto *open source*
  - Necessidade de estudá-los de forma segmentada
- ***Heroes***:
  - Soma das contribuições é maior ou igual a 5% do total do projeto
- **Médios:**
  - Soma das contribuições é maior ou igual a 1% e menor que 5%
- **Periféricos:**
  - Soma das contribuições é menor que 1%

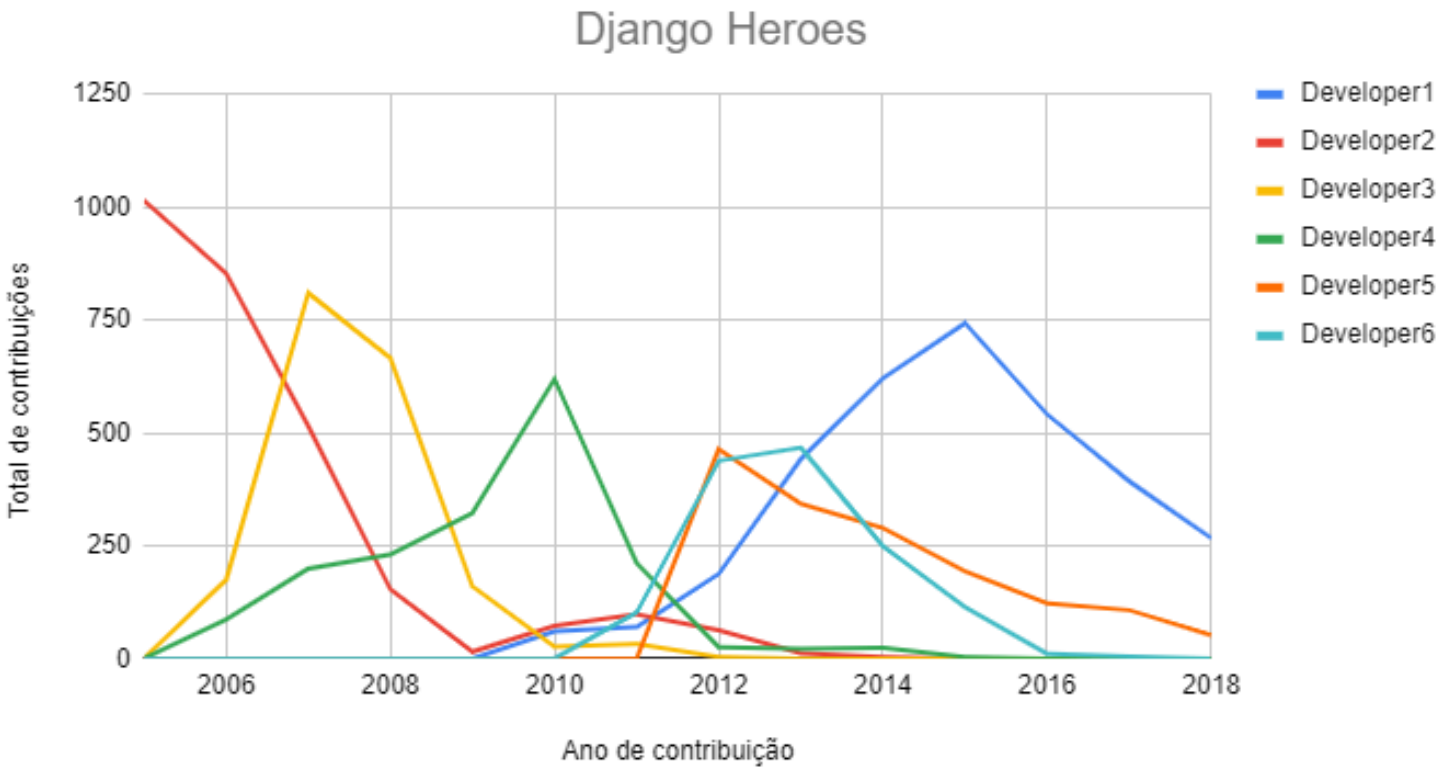
Caracterização da População de Autores

Projeto	Categoria	∑ autores	∑ *commits*	Percentual de contribuição (%)
Bootstrap	Autor periférico	1.278	3.579	19,49
	Autor médio	3	811	4,42
	Autor *heroe*	4	13.969	76,09
Django	Autor periférico	2.065	7.869	29,70
	Autor médio	13	5.928	22,38
	Autor *heroe*	6	12.694	47,92
Elasticsearch	Autor periférico	1.196	8.963	24,77
	Autor médio	21	14.870	41,09
	Autor *heroe*	6	12.352	34,14
Pandas	Autor periférico	1.690	5.624	30,27
	Autor médio	9	4.187	22,53
	Autor *heroe*	3	8.770	47,20
Spring Boot	Autor periférico	664	1.970	9,86
	Autor médio	3	1.365	6,84
	Autor *heroe*	4	16.618	83,3

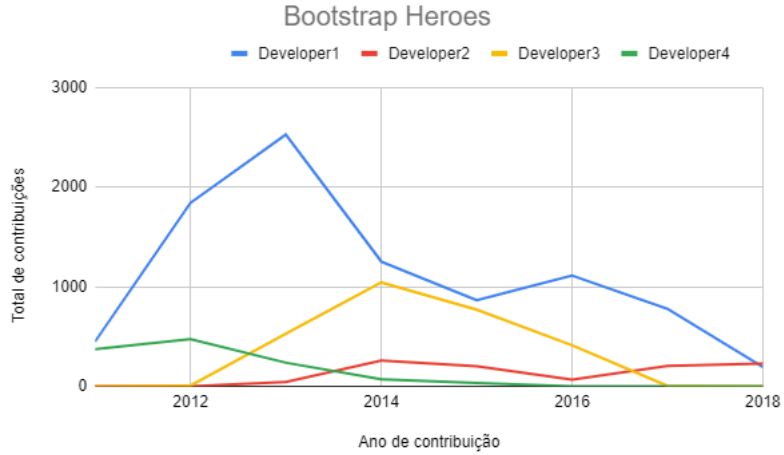
Distribuição dos *Heroes* no Ciclo Evolutivo

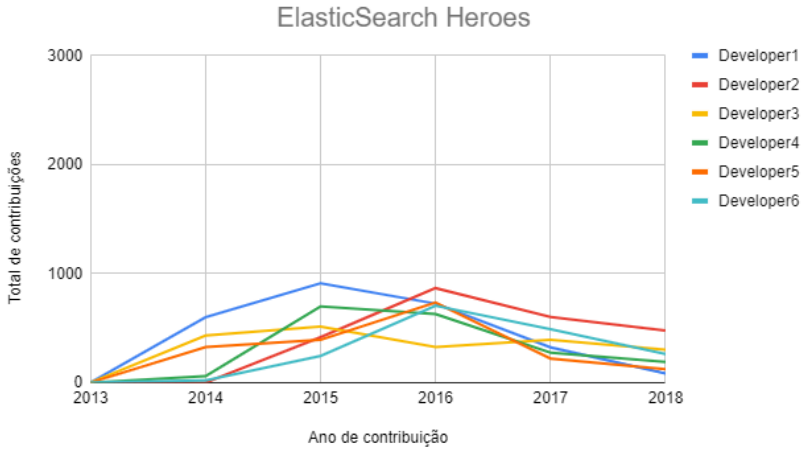
- Concentração expressiva do conhecimento do código nos autores *heroes*
- Finalidade de compreender como as contribuições dos *heroes* evolui
    - Se estão em constante crescimento ou não
    - Se ocorre um rápido pico ou se o aumento das contribuições é gradual

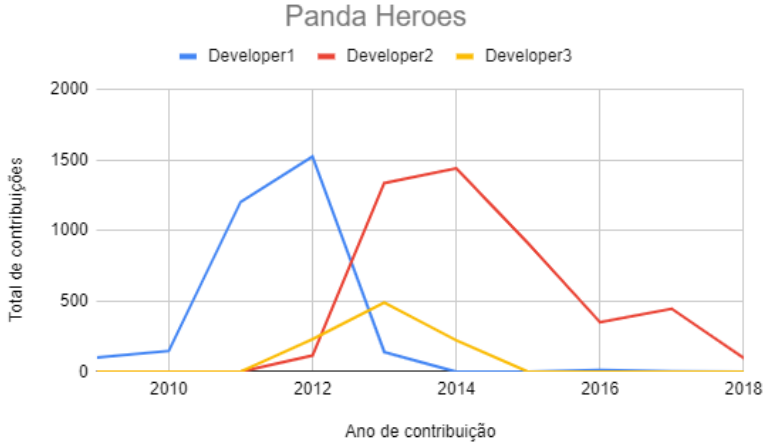
Distribuição dos *Heroes* no Ciclo Evolutivo

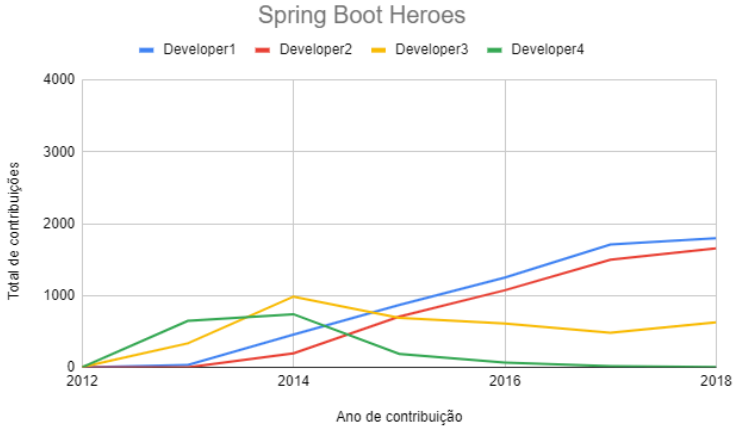


Distribuição dos *Heroes* no Ciclo Evolutivo









Propriedade de Código no Ciclo Evolutivo do Software

- Adaptando as notações desenvolvidas por Foucault et al. [2014] para o contexto do presente estudo
- A propriedade de código é dada pela razão das contribuições feitas por um desenvolvedor pelo total das contribuições do projeto

$$own_{p,d} = \frac{\omega(p,d)}{\omega(p)}$$

- O valor mais elevado dessa razão representa a **propriedade de código do projeto** e o desenvolvedor responsável por essa contribuição é considerado o **proprietário**

$$\max(\{\ own_{p,d} \mid d \in D\})$$

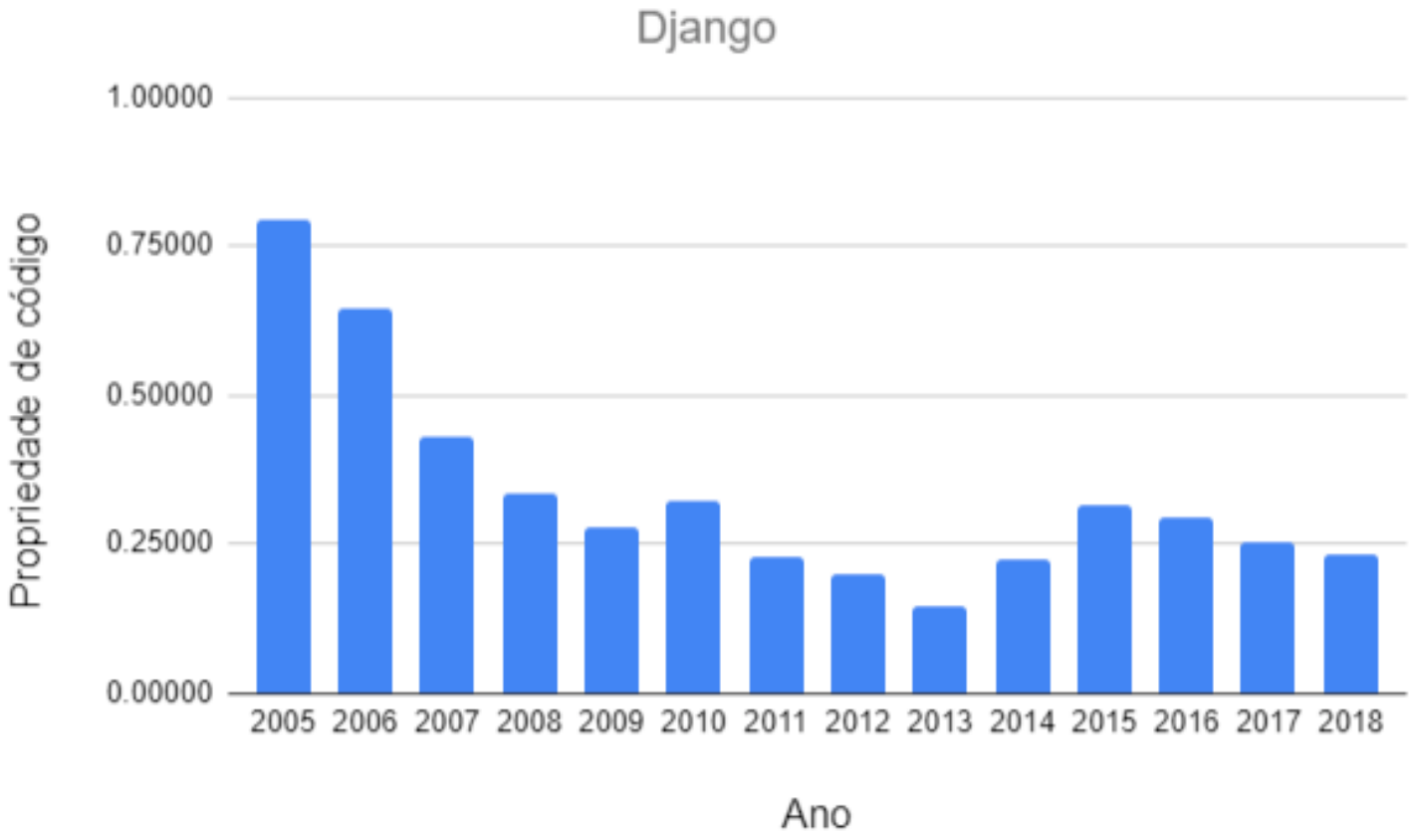
Propriedade de Código no Ciclo Evolutivo do Software

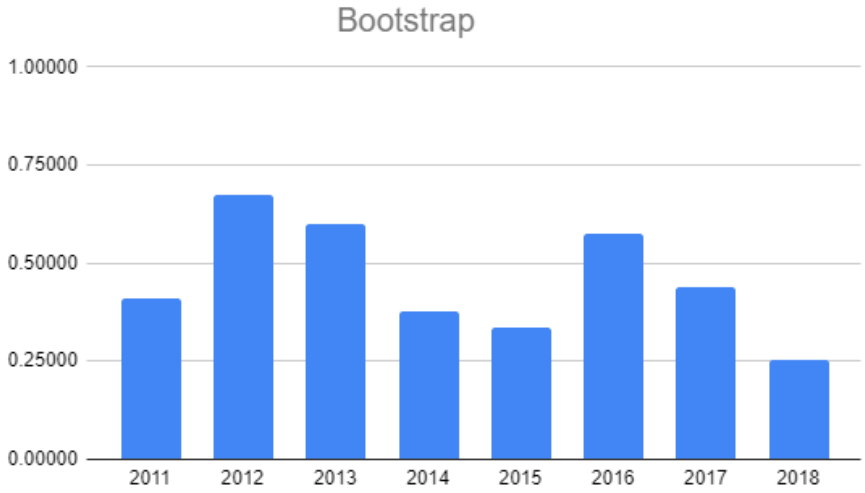
- A métrica foi calculada para cada **período de tempo T**
- Considerando que:
  - ω (*p,t*) é a soma de todas contribuições realizadas no projeto *p* em uma janela de tempo *t*
  - ω (*d,t*) é a soma de todas contribuições realizadas pelo desenvolvedor *d* em uma janela de tempo *t*

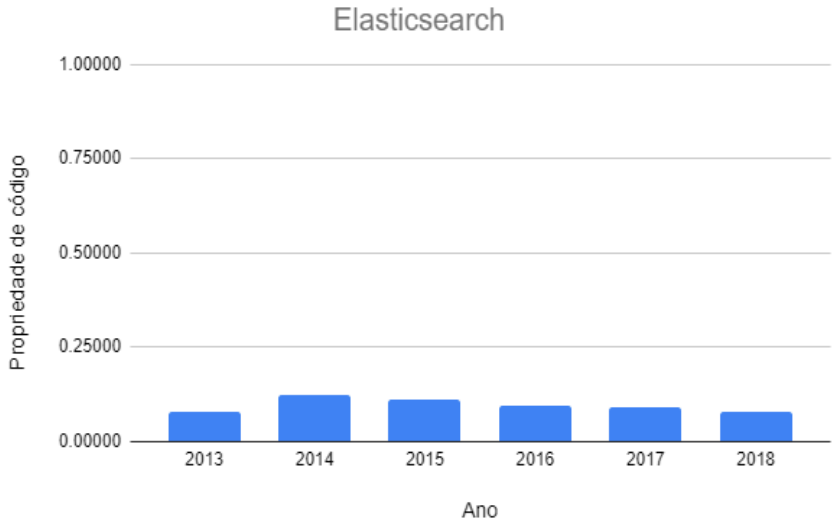
$$own_{t,d} = \frac{\omega(d,t)}{\omega(p,t)}$$

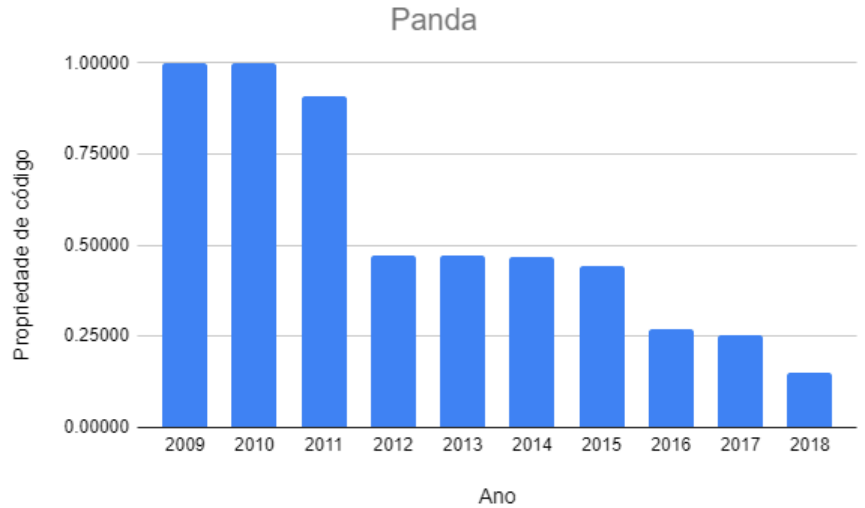
$$\max(\{\ own_{t,d} \mid d \in D\})$$

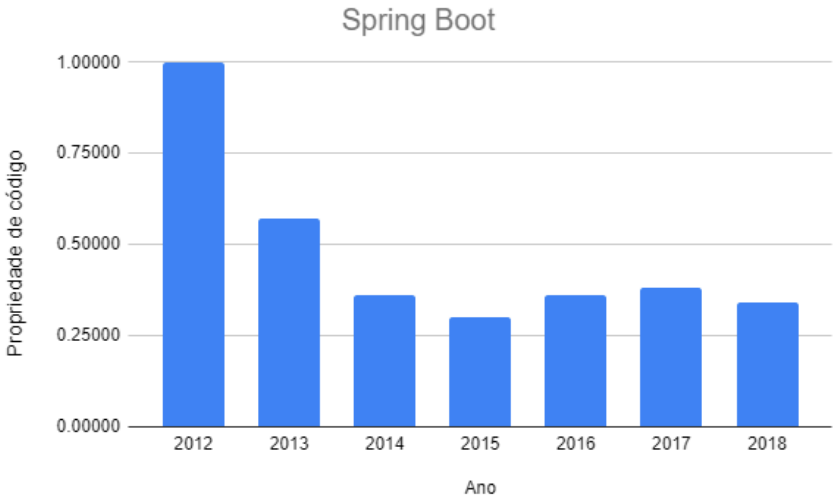
Propriedade de Código no Ciclo Evolutivo do Software



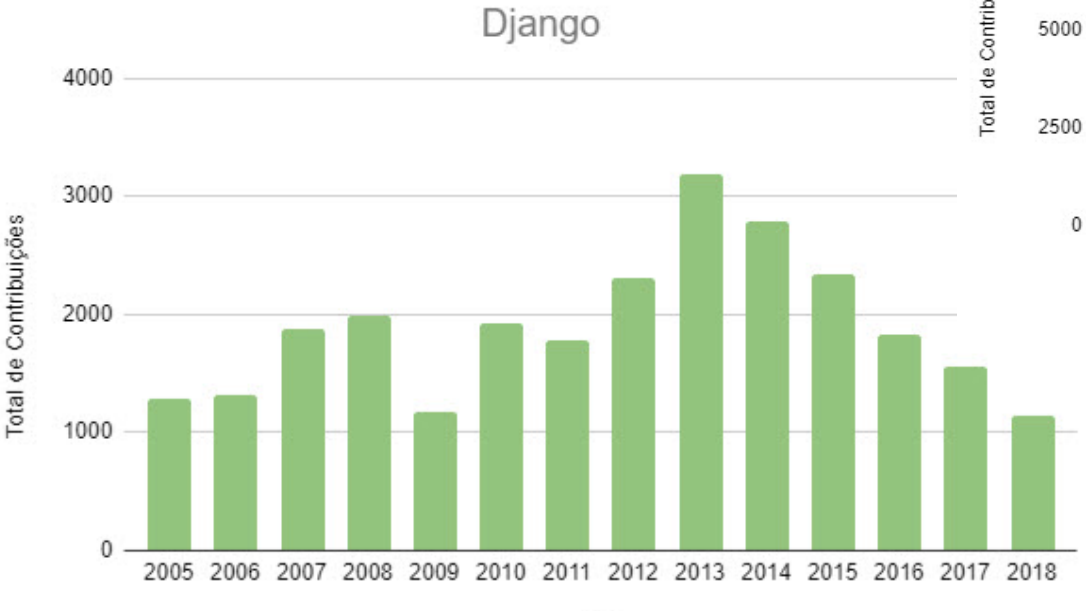

Bootstrap
1.00000
0.75000
0.50000
0.25000
0.00000
2011
2012
2013
2014
2015
2016
2017
2018

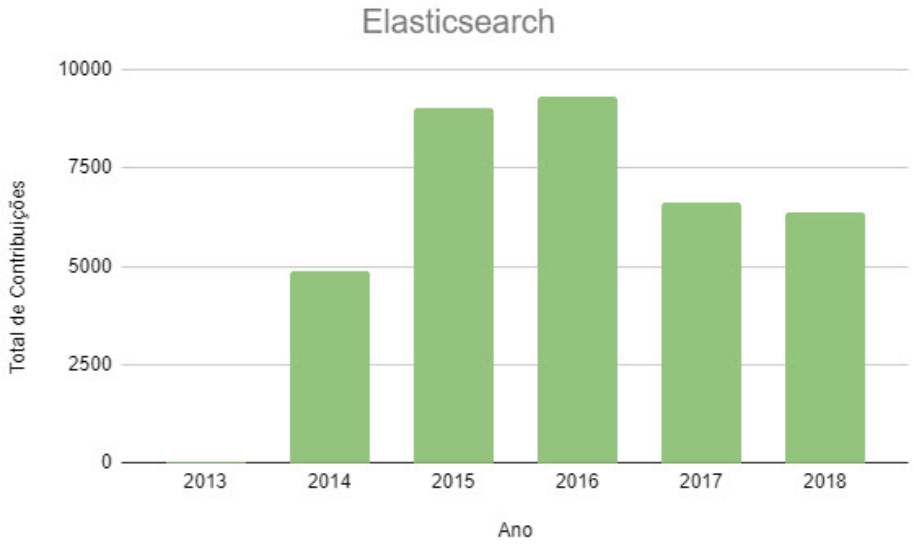

Elasticsearch
1.00000
0.75000
0.50000
0.25000
0.00000
Propriedade de código
2013
2014
2015
2016
2017
2018
Ano

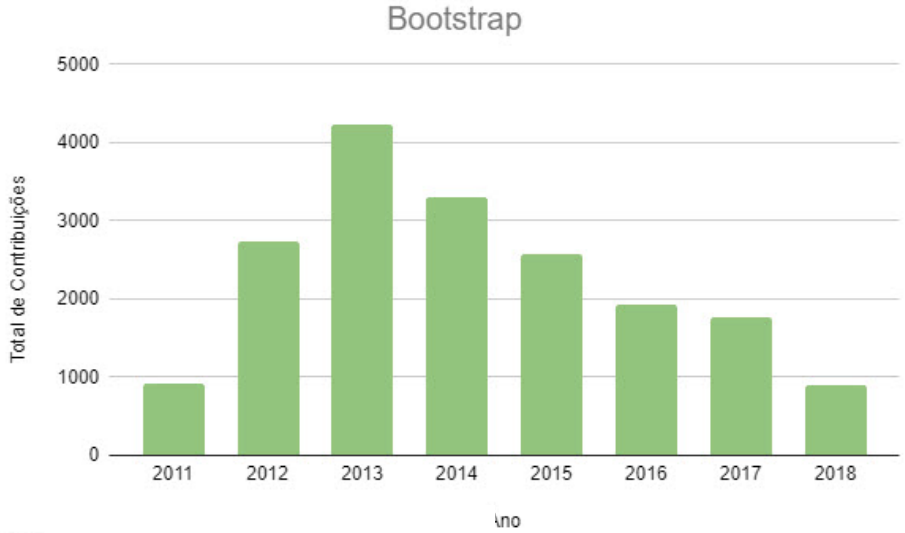

Panda
1.00000
0.75000
0.50000
0.25000
0.00000
Propriedade de código
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Ano

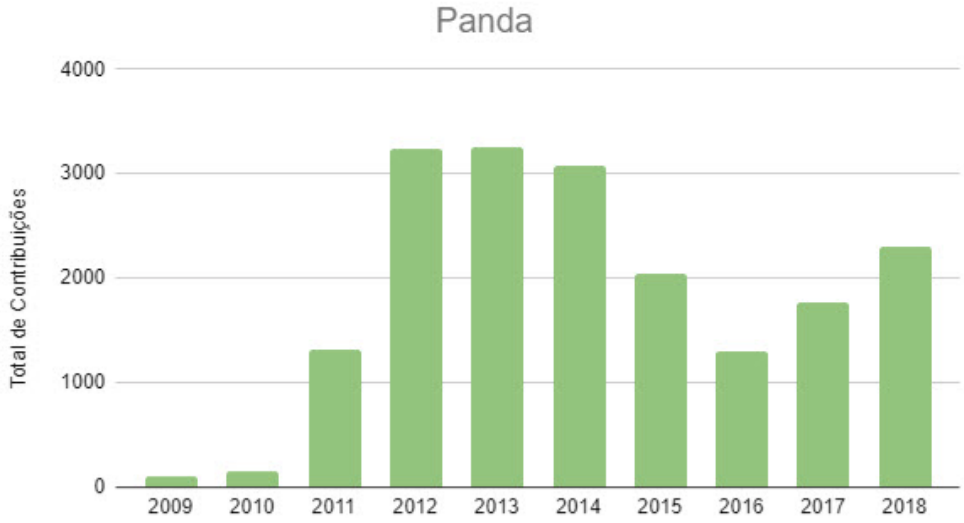

Spring Boot
1.00000
0.75000
0.50000
0.25000
0.00000
Propriedade de código
2012
2013
2014
2015
2016
2017
2018
Ano

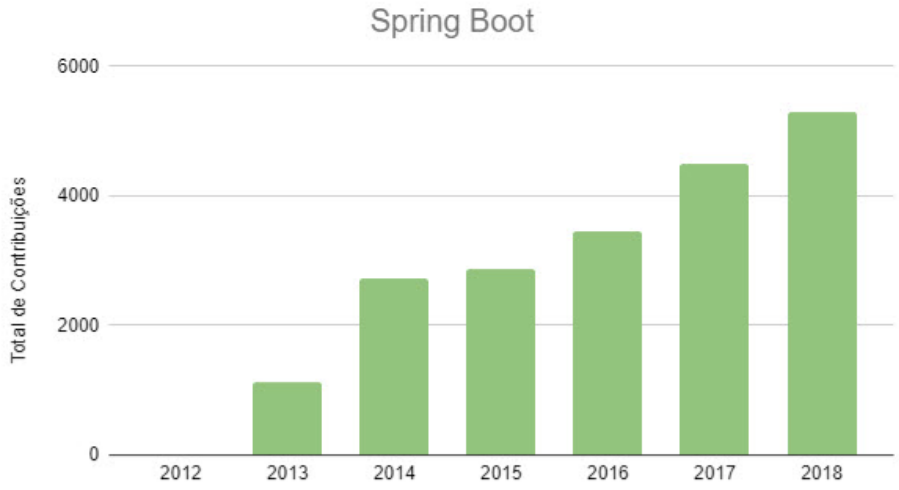
Frequência de Contribuição no Ciclo de Vida do Software






Bootstrap
5000
4000
3000
2000
1000
0
Total de Contribuições
2011
2012
2013
2014
2015
2016
2017
2018
no


Panda
4000
3000
2000
1000
0
Total de Contribuições
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018


Spring Boot
6000
4000
2000
0
Total de Contribuições
2012
2013
2014
2015
2016
2017
2018

Correlação entre a propriedade de código e o total de contribuições

Projeto	Correlação de *Spearman*	Classificação
Bootstrap	0,5	correlação direta moderada
Django	-0,376	correlação inversa fraca
Elasticsearch	0,257	correlação direta fraca
Pandas	-0,339	correlação inversa fraca
Spring Boot	-0,571	correlação inversa moderada

- Para os projetos analisados **não existe correlação**, entre o valor da propriedade de código e a variação da quantidade de contribuições efetuadas no repositório

Correlação entre as contribuições do proprietário e o total de contribuições do período

Projeto	Correlação de *Spearman*	Classificação
Bootstrap	0,952	correlação direta muito forte
Django	0,2	correlação direta fraca
Elasticsearch	0,841	correlação direta forte
Pandas	0,672	correlação direta moderada
Spring Boot	0,964	correlação direta muito forte

- Há uma **correlação direta** entre as contribuições do proprietário do período e o total de contribuições do período nos projetos

Respostas das Questões de Pesquisa

QP1, QP2 e QP3

QP1: O conhecimento do software se difunde ao longo do ciclo de vida?

- Conhecimento é restrito a uma pequena parcela de autores, nomeados *heroes*
    - 80% dos contribuidores realizam até 2 *commits*
- **Tendência de disseminação do conhecimento** à medida que o software evolui
    - Ao analisar a evolução do valor da métrica de propriedade de código
    - Métrica tem valores menores nos últimos períodos analisados

QP2: Os desenvolvedores considerados *heroes* nas primeiras versões permanecem *heroes* durante a evolução do projeto?

- Os *heroes* iniciais **não** estão presentes dentre os *heroes* finais
  - O autor que mais contribuiu no primeiro período apresentou nenhuma ou uma quantidade pequena de contribuições no último período analisado
  - Comportamento **unânime** nos cinco projetos
- Os *heroes* **não** ocupam essa posição de modo **simultâneo**
  - Em cada período T, concentra-se em um ou dois deles
- Diminuição das contribuições dos heroes ocorre de modo **gradual**
- Evolução de um autor, considerado periférico, até tornar-se *hero* também é gradual

QP3: A concentração do conhecimento é impactada pela quantidade total de contribuições do projeto?

- Os resultados indicam que **não** se pode inferir nenhuma relação, direta ou inversa, entre o valor da métrica de propriedade de código e a variação no número de contribuições totais do projeto

- Apenas a avaliação da variação da quantidade de contribuições **não é o suficiente** para compreender a variação no valor da métrica de propriedade de código no ciclo evolutivo

Distribuição do Conhecimento de Software Análise Qualitativa

Survey

PERFIL DO DESENVOLVEDOR

DISSEMINAÇÃO DO CONHECIMENTO

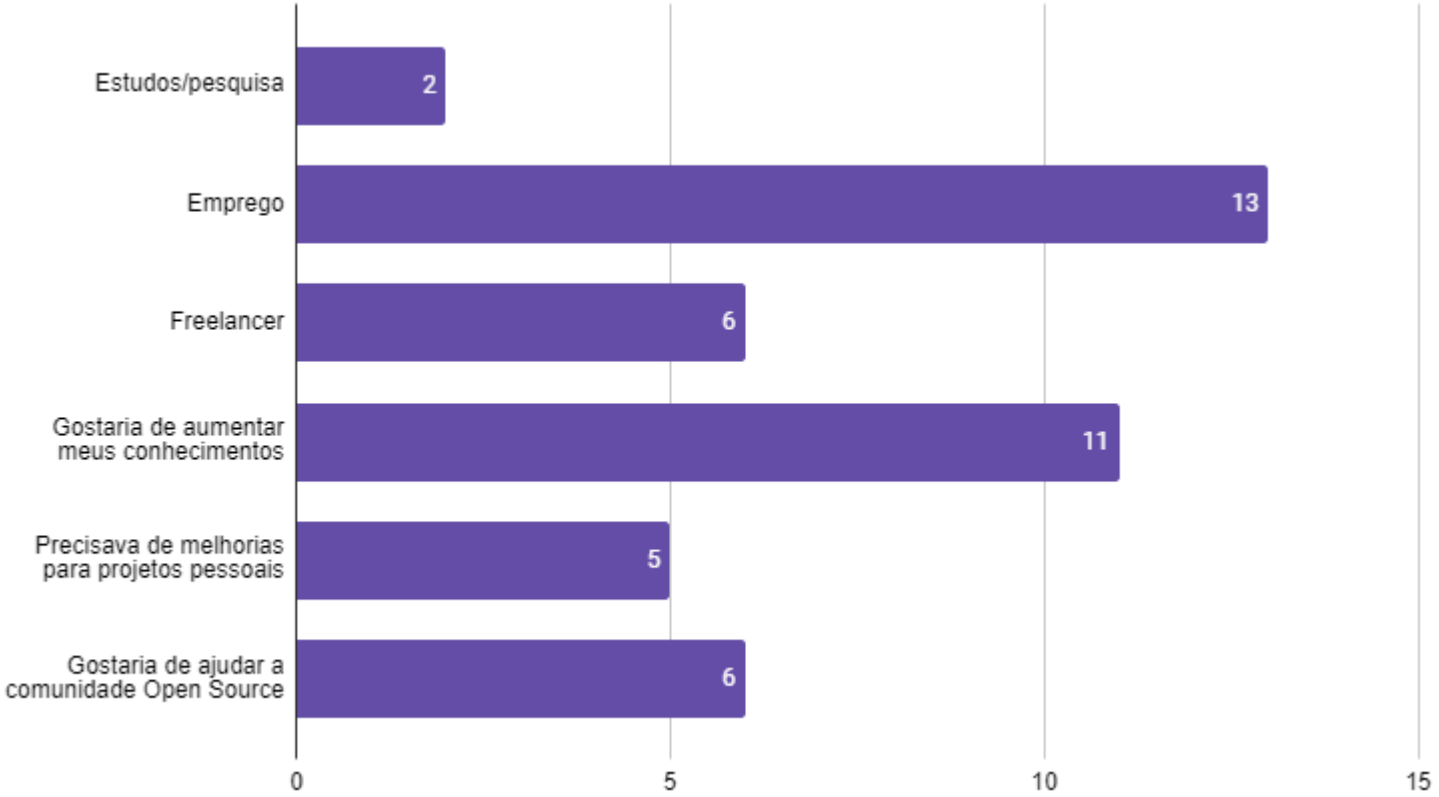
Público Alvo

Categoria	Total de autores selecionados (%)	Total de e-mails enviados	Total de respostas
Hero	100%	21	1
Médio	100%	44	5
Periférico	5%	347	41

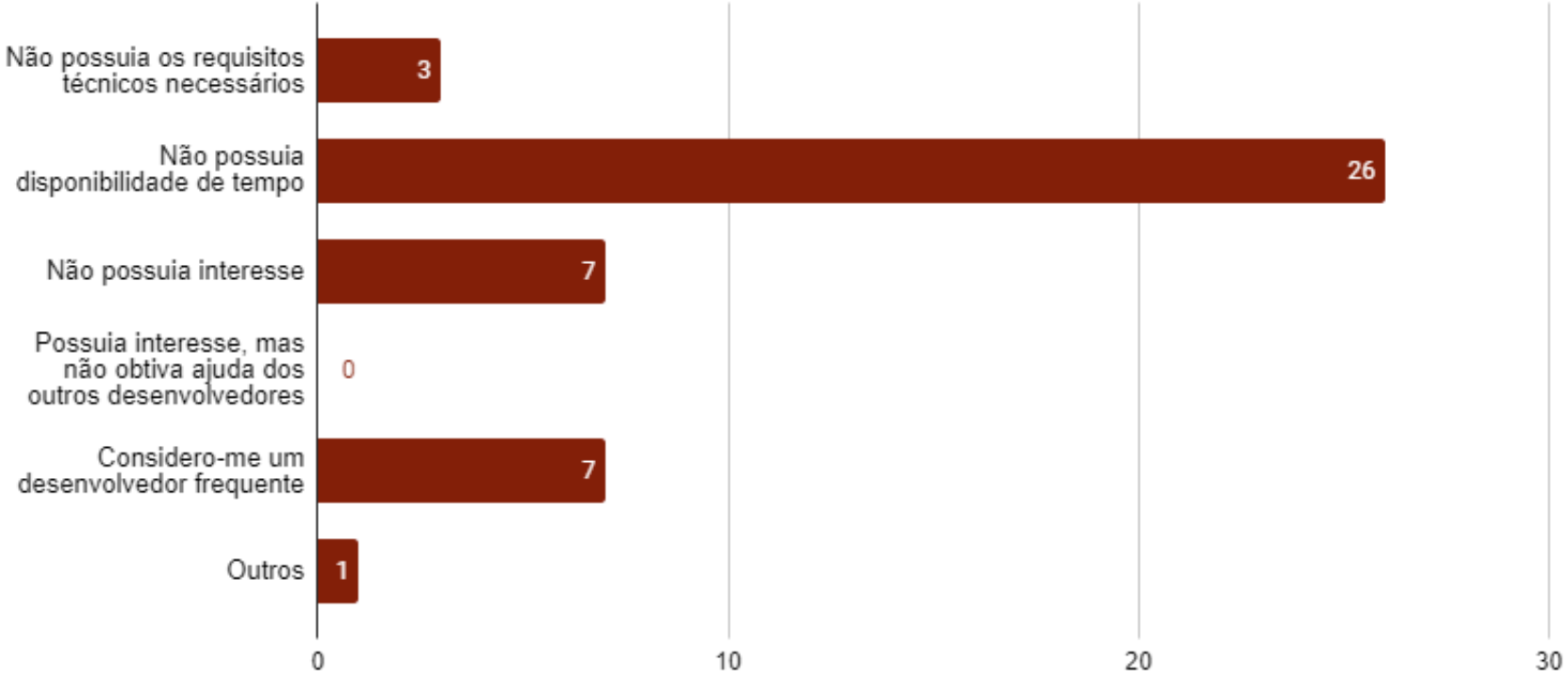
Total de 47 respostas (11% do total enviado)

Resultados do Survey

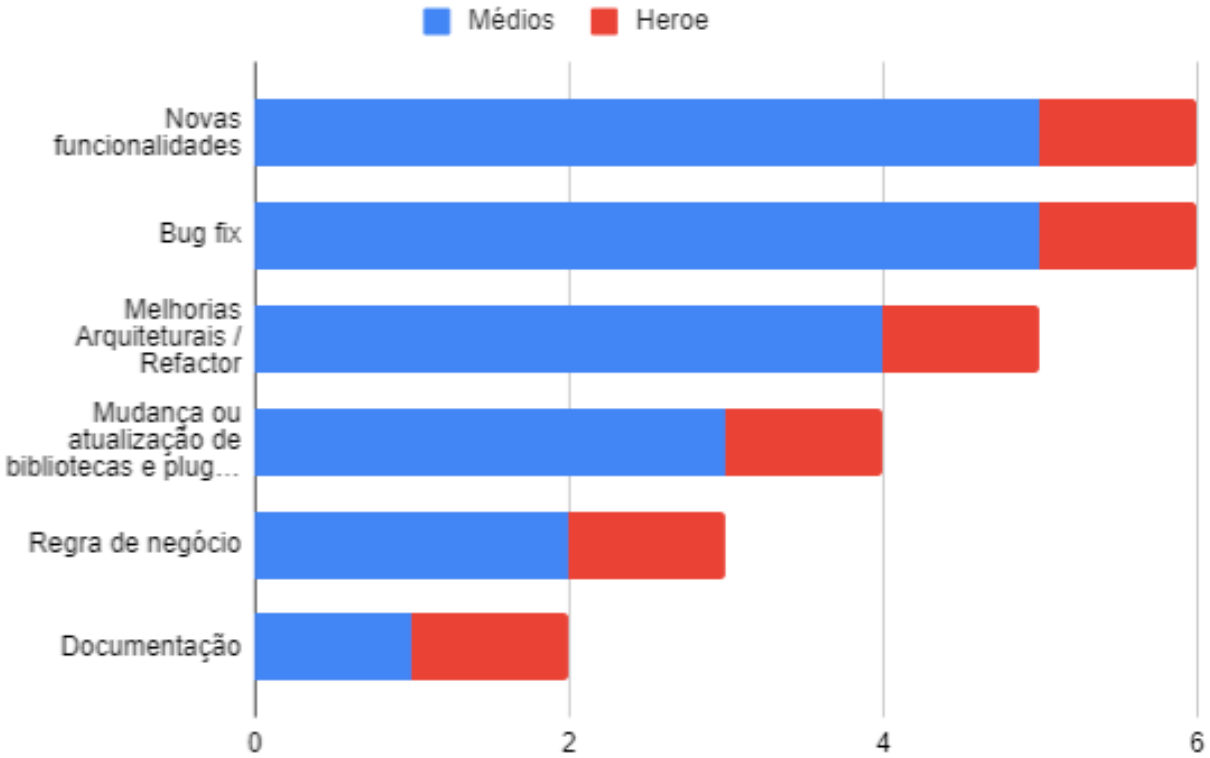
Autores periféricos - Motivação principal para contribuições no projeto



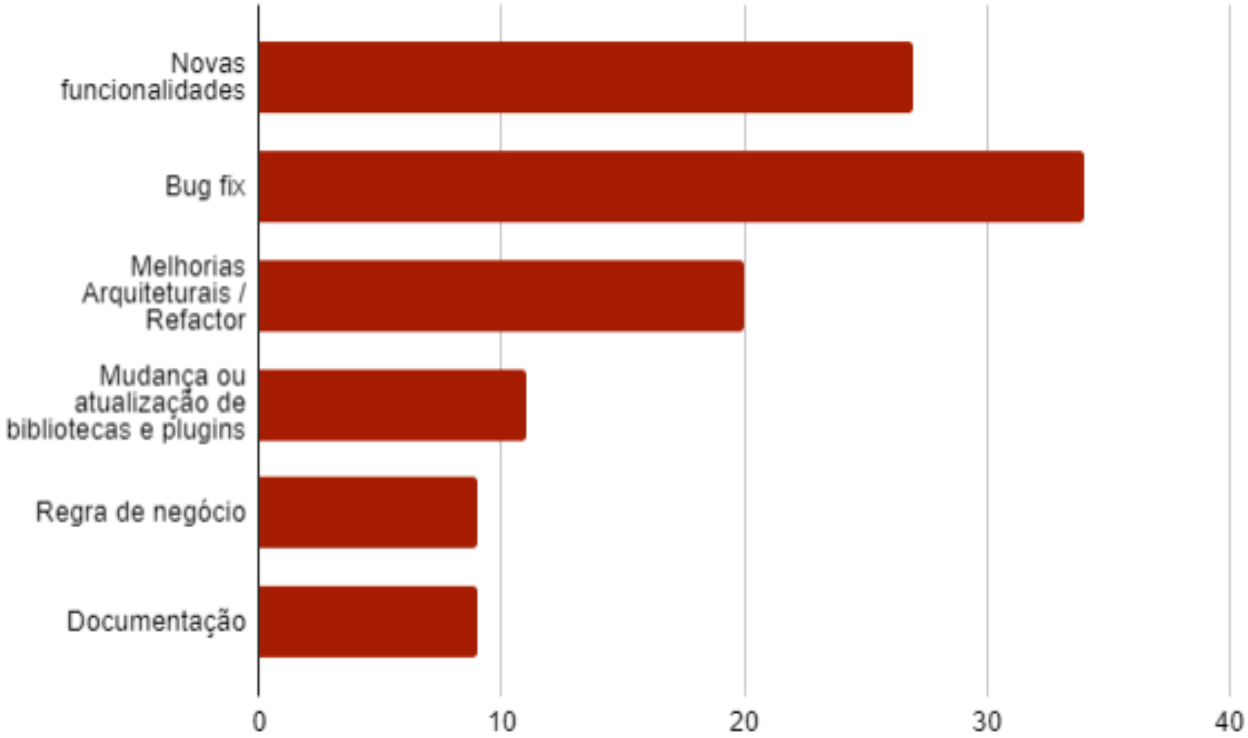
Autores periféricos - Razões para não ter se tornado um autor frequente no projeto



Autores médios e heroes - Tipos de contribuições efetuadas
Disseminação do conhecimento



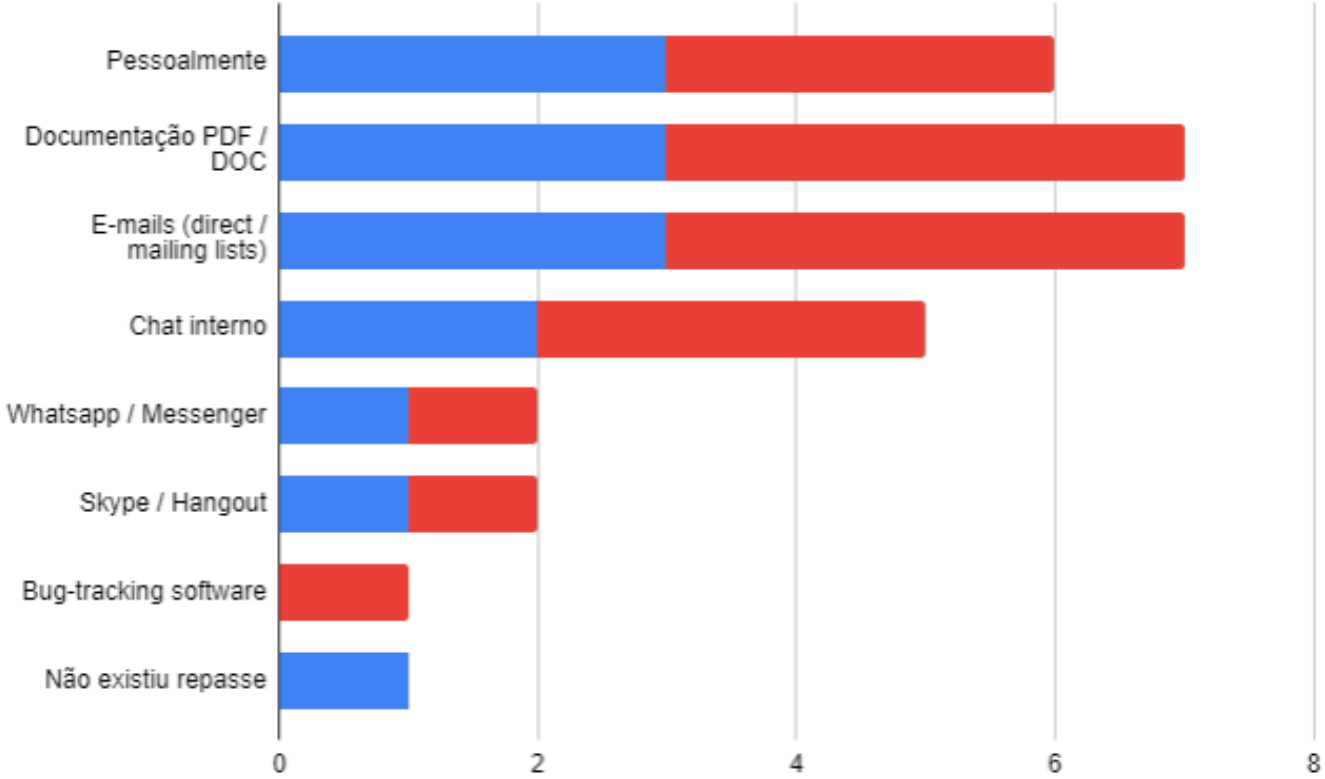
Autores periféricos - Tipos de contribuições efetuadas
Disseminação do conhecimento



Autores periféricos - Recebeu ajuda de outros colaboradores do projeto?

1. Não recebi ajuda e não era necessário **25%**
2. Não recebi ajuda, mas era necessário **0%**
3. Recebi ajuda **75%**
    - Por meio do envio de **sugestões**, conselhos e **feedbacks** por parte dos autores mais experientes
    - **Code review** e co-contribuição
    - **Conversas** pelo *chat*, e-mail e *issue tracker*
    - Reuniões e *meetups*
    - Orientação sobre padrões de codificação

Autores médios e heroes - Transferência de conhecimento do software



Respostas das Questões de Pesquisa

QP4 e QP5

QP4: Como se dá a atuação dos desenvolvedores periféricos nos projetos?

- Falta de disponibilidade de tempo
    - Principal razão para não aumentar a quantidade de contribuições
- Correções de *bugs*, acréscimo de novas funcionalidades, melhorias arquiteturais e correção na documentação do projeto
- Principais motivações
    - Emprego, desejo de aumentar conhecimentos técnicos ou contribuir com a comunidade *open source*, *freelancer* e melhorias em projetos pessoais
- Dificuldade de comunicação não é um fator preponderante que os levou a contribuir pouco

QP5: Como ocorre a disseminação de conhecimento do software entre os autores?

- Por meio da contribuição no código
  - Pelo desenvolvimento de novas funcionalidades, correção de *bugs* e melhorias arquiteturais
- Compartilhamento de conhecimento extra-código nas duas vias
- Troca de informações
  - Documentações de projeto
  - E-mails, *chats*, pessoalmente
  - *Bug-tracking*
- Nenhum autor relatou dificuldade de comunicação ou obstáculos na aquisição dos conhecimentos necessários
  - Facilita entrada de novos colaboradores

Conclusão

Conclusão

- Estudo de caso com cinco grandes projetos *open source*
  - Compreender como ocorre a evolução da distribuição de conhecimento entre os desenvolvedores durante o ciclo de vida do software
- Coleta de dados do GitHub
  - Disponibilização de dados e *scripts* para replicação e extensão do trabalho
- Estudo da atuação dos *heroes* no ciclo evolutivo do software
  - Existe uma alternância entre os heroes
    - *Series of generations"* [Robles et al., 2009]
  - Investir em ações de retenção
    - Importante para **evolução** e sustentabilidade do projeto

Conclusão

- Métrica de propriedade de código no decorrer da evolução de projetos *open source*
    - Propriedade de código do projeto tende a diminuir

- Resultados sugerem que o conhecimento torna-se mais distribuído entre os desenvolvedores no decorrer de seu processo evolutivo
    - Não há relação entre a variação do total de contribuições do projeto e a métrica de propriedade de código

Conclusão

- O desejo de aumentar os conhecimentos técnicos e contribuir com a comunidade *open source* são fortes motivos que levam periféricos a contribuir nos projetos
- Falta de disponibilidade de tempo e interesse são fatores que atrapalham a continuidade das contribuições
- Importância da **distribuição de conhecimento**
    - Cenários extra-código
    - Compartilhamento de informações é vital para subsistência de todo projeto

Trabalhos Futuros

- Replicar estudo utilizando *dataset* maior e comparar com diferentes métricas e granularidades
- Construção de algoritmos de desambiguação com maior valor de precisão
- Construir ferramenta que permite a coleta e caracterização dos dados de forma automatizada

Evolução da Distribuição de Conhecimento de Software em Projetos *Open Source*

Talita Santana Orfanó

Orientadoras: Mariza A. da Silva Bigonha e Kecia Aline M. Ferreira

Obrigada!

Programa de Pós-Graduação
Ciência da Computação
UFMG

ANEXOS

Trabalhos Relacionados

Fatores Humanos e Organizacionais

Autoria e Propriedade de Código

Heroes vs Periféricos

Desambiguação de Autores

- **Por login**
  - O dados de login retornados pela API corresponde ao ***committer***, ao invés do **autor**
  - Em muitos casos, pode coincidir de serem a mesma pessoa
  - Contudo, assumir isso apresenta sérios riscos aos resultados da coleta e a respectiva análise
  - Desambiguação por meio de login não é adequada para o nosso estudo

```
"sha": "55ffcf8e7b414a39e2dfc9c9eb4c5d3fa548e78e",
"node_id": "MDY6Q29tbWl0NDE2NDQ4Mjo1NWZmY2Y4ZTdiNDE0YTM5ZTJkZmM5YzllYjRjNWQzZmE
"commit": {
  "author": {
    "name": "Raúl Cumplido",
    "email": "raulcd@tid.es",
    "date": "2012-06-07T11:46:06Z"
  },
  "committer": {
    "name": "Tim Graham",
    "email": "timograham@gmail.com",
    "date": "2012-06-30T21:16:40Z"
  },
"url": "https://api.github.com/repos/django/django/commits/55ffcf8e7b414a39e2df
"html_url": "https://github.com/django/django/commit/55ffcf8e7b414a39e2dfc9c9eb
"comments_url": "https://api.github.com/repos/django/django/commits/55ffcf8e7b4
"author": null,
"committer": {
  "login": "timgraham",
  "id": 411869,
  "node_id": "MDQ6VXNlcjQxMTg2OQ==",
```

Desambiguação de Autores

- **Por nome / e-mail - Uso de heurísticas**
  - Estudo e comparação de cinco heurísticas de desambiguação de autores em projetos *open source*
  - Bird, Canfora, Simple, Robles e Goeminne
  - Resultados
    - Alto coeficiente de revocação (*recall*) - 100%
    - Baixa precisão em projeto de grande porte - ~30%
  - Mesmo sob essas circunstâncias, optou-se por avaliar os resultados geradas por uma dessas heurísticas: Bird

```
1  início
2  |  ambiguidade ⟵ falso;
3  |  t ⟵ 0,93;
4  |  Normalização dos dados (autor1, autor2);
5  |  se Similaridade do nome completo(autor1.nome, autor2.nome) >= t OU
6  |  (Similaridade primeiro nome(autor1.nome, autor2.nome) >= t E
   |   Similaridade último nome(autor1.nome, autor2.nome) >= t ) OU
7  |  Similaridade entre os prefixos de email(autor1.email, autor2.email) >= t
   |   OU
8  |  Prefixo contêm primeiro e último nome(autor1.email, autor2.nome) OU
9  |  Prefixo contêm primeiro e último nome(autor2.email, autor1.nome) OU
10 |  Prefixo contêm primeiro nome e inicial do último nome(autor1.email,
   |   autor2.nome) OU
11 |  Prefixo contêm primeiro nome e inicial do último nome(autor2.email,
   |   autor1.nome) OU
12 |  Prefixo contêm inicial do primeiro nome e o último nome(autor1.email,
   |   autor2.nome) OU
13 |  Prefixo contêm inicial do primeiro nome e o último nome(autor2.email,
   |   autor1.nome) então
14 |  |  ambiguidade ⟵ verdadeiro;
15 |  fim
16 |  retorna ambiguidade
17 fim
```

Desambiguação de Autores - Heurística Bird

- A heurística foi executada para os autores dos cinco projetos
- Os resultados foram avaliados manualmente
  - Existência de problemas consideráveis na identificação dos autores
  - Algoritmo e resultados estão disponíveis para consulta

Bootstrap:
- contact@helmutgranda.com, contact@weirdog.com e contact@dominicbarnes.us
- cesidio.dilanda@gmail.com, sidp@knights.ucf.edu e sid@sidroberts.co.uk

Django:
- bphillips (bphillips@caktusgroup.com) e Joshua Philips (jphillips@imap.cc)
- Thomas Orozco (thomas@orozco.fr) e Thomas Sorrel (thomas@pandeiro.fr)

Elasticsearch:
- ludo.helder@gmail.com e ludovic@xwiki.com
- Tim B (tim@uncontended.net) e Tim Vernum (tim@adjective.org)

Panda:
- joejev@gmail.com e joe@quantopian.com
- John Freeman (jfreeman08@gmail.com) e Jesse Farnham (jfarnham20@gmail.com)

Spring boot:
- alexander.abramov.pub@gmail.com e alexander.constantin@gmail.com
- Jay Anderson (jaanderson@shutterfly.com) e Jayaram Pradhan (jayaramimca@gmail.com)

Propriedade de Código no Ciclo Evolutivo do Software

- Adaptando as notações desenvolvidas por Foucault et al. [2014] para o contexto do presente estudo, pode-se considerar que:
    - ω (*p*) é a soma de todas contribuições dos desenvolvedores realizadas em um projeto *p*
    - ω (*d*) é a soma de todas contribuições realizadas por um desenvolvedor *d*
    - ω (*p,d*) é a soma de todas contribuições realizadas por um desenvolvedor d em um projeto *p*


100M*
repositories worldwide
50M*
developers worldwide
2.9M*
businesses & organizations worldwide

Autoria de Software e Propriedade de Código

Autoria de Software

O termo autoria de software é utilizado para referenciar o autor responsável por um trecho de código ou módulo

Propriedade de Código

O conceito de propriedade de código pode ser utilizado para distinguir dentre os autores do projeto aquele que possui uma maior responsabilidade, ou seja, aquele desenvolvedor que criou ou alterou a maior parte do código existente no módulo ou projeto analisado