

Aspectos Colaborativos no Desenvolvimento de Contratos Inteligentes da Plataforma Ethereum: Um Estudo Exploratório-descritivo Preliminar

Alan Rodrigues¹, Allysson Alex Araújo¹, Matheus Paixao² e Pamella Soares²

¹Grupo de Estudos em Sistemas de Informação e Inovação Digital (GESID)
Universidade Federal do Ceará (UFC) – Crateús – CE – Brasil

²Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Universidade Estadual do Ceará (UECE) – Fortaleza – CE – Brasil

alan.rodrigues434@gmail.com, allysson.araujo@crateus.ufc.br,
matheus.paixao@uece.br, pamella.soares@aluno.uece.br

Abstract. *Smart Contracts (SCs) projects are joining the open source development. However, the development of SCs have to deal with new constraints which motivate questions about the dynamics of collaboration between developers and the community. Based on an exploratory-descriptive scope, this paper aims to discuss preliminary results on collaborative elements regarding the open source development of SCs, in particular: i) the relationship between commits and collaborators, and ii) the report and resolution of issues.*

Resumo. *Presencia-se uma adesão de projetos de Contratos Inteligentes (CIs) ao desenvolvimento open source. Ademais, reconhece-se que o desenvolvimento de CIs lida com novas restrições as quais motivam novos questionamentos sobre a dinâmica de colaboração entre desenvolvedores e para com a comunidade. Enquadrando-se a partir de uma pesquisa exploratória-descritiva, o presente artigo objetiva discutir resultados preliminares sobre elementos colaborativos quanto ao desenvolvimento open source de CIs, em específico: i) a relação entre commits e colaboradores, e ii) quanto ao report e resolução de issues.*

1. Introdução

O *blockchain*, que teve seu desenvolvimento difundido após a criação da criptomoeda Bitcoin [Nakamoto 2009], funciona como uma tecnologia distribuída capaz de armazenar criptograficamente um registro de eventos lineares de transações entre atores em rede [Risius and Spohrer 2017]. Em decorrência da natureza descentralizada baseada em algoritmos de consenso, os próprios atores que compõem o *blockchain* conseguem fazer o papel de validador de transações, contornando, assim, a necessidade de uma entidade intermediadora para garantia da confiança de operações. Entretanto, o cenário de uso do Bitcoin é restrito devido sua aplicação ter como foco principal o armazenamento e a transferência de valores. Visando contornar tal limitação, o surgimento da Ethereum, como uma nova plataforma de *blockchain*, ampliou as possibilidades de desenvolvimento de soluções baseadas em *blockchain* através do uso de Contratos Inteligentes (CIs) [Chen et al. 2020]. Em suma, CIs são códigos executáveis, armazenados de forma distribuída e que contém termos contratuais formalizados para realizar acordos de relacionamento [Oliva et al. 2020]. Por meio desta inovação tecnológica, o desenvolvimento de *Decentralized Applications (dApps)* tem se fortalecido cada vez mais.

Ademais, percebe-se, por parte dos envolvidos no ecossistema de CIs, um movimento de adesão ao desenvolvimento *open source*, inclusive por parte de projetos de alto valor de mercado como Binance USD (\$18 bilhões) e ChainLink Token (\$6 bilhões) [Zou et al. 2019]. Nesse contexto, pressupõe-se que os projetos são desenvolvidos em plataformas sociais de hospedagem de código-fonte, com destaque para o GitHub, até o estágio de *deploy* na Ethereum [Rodrigues et al. 2021a]. Todavia, o desenvolvimento e manutenção de CIs lida com particularidades distintas do convencional como, por exemplo, a natureza inalterável dos CIs após o *deploy*, a garantia da simplicidade para redução de custos operacionais em criptomoedas e transparência de acesso ao código-fonte do CI [Chakraborty et al. 2018, Sillaber et al. 2020]. Tal processo de desenvolvimento intrincado por novas restrições motiva questionamentos sobre a dinâmica de colaboração entre os desenvolvedores, bem como na relação com a comunidade envolvida no projeto.

A pertinência dessa provocação se justifica relevante tendo em vista as particularidades envolvidas na evolução de software *open source* de CIs haja vista que, ao contrário dos softwares tradicionais que podem ser atualizados diretamente, para manter um CI, os desenvolvedores precisam lidar com os ajustes necessários de implantação de um novo contrato, incluindo como proceder quanto a versão antiga. É importante destacar uma crescente presença de trabalhos [Tikhomirov et al. 2018, Pinna et al. 2019, Oliva et al. 2020, Ajenka et al. 2020, Rodrigues et al. 2021a] explorando desafios relativos à manutenção de CIs, no entanto, ainda se constata uma escassez de pesquisas com foco nos aspectos colaborativos. Conforme destacado por Prikladnicki *et al.* (2013), softwares são criados *por e para* pessoas, e entender os aspectos sociais inerentes ao desenvolvimento de software demonstra-se crucial para compreender como os métodos e ferramentas são utilizados e, assim, melhorar criação e manutenção de sistemas.

O presente estudo se insere no contexto de uma pesquisa [Rodrigues et al. 2021a, Costa et al. 2021] em andamento a qual visa compreender os aspectos colaborativos e de evolução de software envolvidos no desenvolvimento de CIs na plataforma Ethereum. Para avançar em tal projeto, conduziu-se um estudo exploratório-descritivo de natureza quali-quantitativa baseado em Mineração de Repositório de Software (MRS). Assim, fundamentado a partir de um experimento computacional, este trabalho investigou 27 CIs de projetos listados entre aqueles com maior valor de mercado de acordo com o *ranking* “ERC20 Top Tokens” do Etherscan (*block explorer* da Ethereum) e que, por sua vez, também dispusessem de um repositório *open source* no GitHub. Complementando os resultados apresentados em [Rodrigues et al. 2021a], cujo objetivo foi caracterizar padrões que denotam diferentes comportamentos evolutivos dos CIs, este artigo contribui ao avançar, de forma preliminar, no entendimento de elementos colaborativos quanto ao desenvolvimento *open source* de CIs, em específico: a relação entre *commits* e colaboradores, e quanto ao *report* e resolução de *issues*.

2. Procedimentos Metodológicos

Inicialmente, realizou-se uma análise no Etherscan para identificação de CIs do tipo ERC-20 aptos para o estudo a partir do *dataset* curado por Rodrigues *et al.* (2021). Dada as limitações de tempo e processamento de dados, avaliou-se uma amostra inicial de 27 CIs oriundos da lista ERC-20 Top Tokens do Etherscan, ou seja, os projetos com maiores valores de mercado no dia da seleção (03/08/2020). Em seguida, a partir dos CIs identificados, executou-se uma busca manual junto ao GitHub para identificação dos repositórios *open*

source. Entretanto, constatou-se que alguns dos CIs não dispunham de repositórios no GitHub. Assim, substituiu-se os contratos em questão pelos contratos subsequentes da lista ERC20 *Top Tokens* e que também estivessem disponíveis no GitHub. Para garantia da correção do repositório selecionado, considerou-se tais critérios: 1) o *nome do repositório* no GitHub deve ter alguma relação com o nome do CI no Etherscan; 2) o *tipo de linguagem de programação* no repositório deve ser prioritariamente a linguagem Solidity; 3) devem existir arquivos escritos no *formato de CIs* e com extensão “.sol”; 4) a *descrição do projeto*, quando disponível, deve ter relação com o CI disponível no Etherscan.

Em seguida, utilizou-se da API REST do GitHub para coleta dos dados de evolução dos repositórios. Através de *scripts* em Python (disponibilizados no repositório de apoio desta pesquisa [Rodrigues et al. 2021b]), coletou-se, para cada um dos 27 projetos no GitHub, a lista de *commits* e códigos-fonte dos CIs, lista de colaboradores, dados relacionados às *issues* como seus respectivos números, título, nome do colaborador que submeteu, *status* (aberta ou fechada), data de criação, data de atualização e data de fechamento (se a *issue* estiver fechada), além da quantidade de *commits* de cada colaborador desde a início do projeto no GitHub. A partir dos dados evolutivos coletados, executou-se a análise dos aspectos colaborativos oriundos dos dados quantitativos e qualitativos obtidos a partir do repositório no GitHub de cada projeto. Essa discussão foi pautada por duas análises principais: 1) a relação entre *commits* e colaboradores e 2) a resolução de *issues*. Para a primeira perspectiva, portanto, tem-se a oportunidade de analisar quantitativamente o nível de atividade de cada projeto. Quanto à segunda questão, tem-se uma breve discussão qualitativa sobre o engajamento da comunidade em relação ao *report* de *issues*, bem como atenção empregada na resolução das mesmas. Tais análises também se respaldaram na classificação de padrões de evolução de software propostas por [Rodrigues et al. 2021a] as quais oportunizaram o entendimento sobre a similaridade dos projetos desenvolvidos de forma *open source* no GitHub em contraste à versão do CI disponível no Etherscan.

3. Resultados e Análises Preliminares

A Figura 1 apresenta, para cada projeto, uma visão geral sobre a quantidade de *commits*, quantidade de colaboradores e quantidade de *issues*. Destaca-se também que todos os dados granulares encontram-se disponíveis no repositório de apoio da presente pesquisa [Rodrigues et al. 2021b]. Conforme pode-se perceber, há uma ampla diversidade de composições quanto aos dados coletados. Ao analisar a quantidade de *commits*, constatou-se que 50% dos projetos apresentam menos de 11 *commits*, enquanto 26% possuem mais de 100 *commits*. Os três projetos com mais *commits* foram Dai-Stablecoin (586), Golem (365) e Decentraland (333). Por sua vez, tais projetos são os que apresentaram a maior quantidade de colaboradores: 21, 10 e 11, respectivamente. Em contrapartida, pode-se verificar a presença de dois projetos (Byton e DxChain-Token) com somente 1 *commit* e 1 colaborador denotando, assim, ausência de interações. Revela-se, portanto, a incidência de cenários com o mero propósito de criação do repositório, mesmo com a falta de engajamento da comunidade. Dessa forma, verifica-se o uso do GitHub como mecanismo de exposição do código e não como apoio ao processo de desenvolvimento propriamente.

Quanto ao tempo de desenvolvimento e nível de atividade dos projetos, verifica-se que, por exemplo, no caso do AION, houve um intenso progresso de desenvolvimento (15 *commits*) entre os meses de Julho e Setembro de 2017. Já o projeto Binance-USD possui apenas dois *commits*, um com data de Setembro de 2019 e outro somente em Julho de

2020, onde não se verificou modificações no código-fonte do contrato. O motivo do *commit* realizado em 2020 foi para correção do arquivo `package.json` visando a redução de vulnerabilidades. O projeto Zilliqa, por sua vez, dispõe de 7 *commits* ao todo, cobrindo o período de 06/01/2018 e 09/01/2018. O projeto DxChain-Token possui somente um *commit* no GitHub realizado no dia 15/07/2018. Salienta-se, por fim, a existência de repositórios com *commits* após a data de *deploy* (vide ChainLink-Token, por exemplo), gerando, portanto, a possibilidade de aprofundar o entendimento sobre tal fenômeno.

Adicionalmente, é possível analisar para cada projeto a relação entre quantidade de *issues* abertas e fechadas. Nesse sentido, pode-se verificar que 76% dos projetos possuem menos do que sete *issues*. Por outro lado, 20% dos projetos apresentaram 30 *issues* reportadas. Os três projetos com maior quantidade de *issues* foram Dai-Stablecoin (30), Golem (30) e Decentraland (30). Analisando em específico a relação entre *issues* abertas e fechadas para projetos com mais de uma *issue*, verifica-se que em 70% dos casos a quantidade de *issues* fechadas se demonstrou superior a quantidade de *issues* abertas. Todavia, em particular, o Paxos-Standard (13) apresentou uma situação com mais *issues* abertas (7) que fechadas (6). Contudo, é válido destacar que as 6 *issues* em aberto eram oriundas de um bot automático para atualizações de dependências do projeto. Já os projetos AION (1), Theta-Token (4) e Nexo (1) não apresentaram nenhuma *issue* fechada. Ao analisar especificamente esses projetos, verificou-se a incidência de *issues* relacionadas a remoção de dependências (AION) obsoletas e configuração de arquivos `.yaml` (Nexo), bem como, no caso do Theta-Token, havia uma dúvida sobre o *software development kit* utilizado pela carteira web, correções textuais, solicitação para adicionar fonte do contrato e ABI ao Etherscan e comentário no código do CI a respeito da criação de *tokens*.

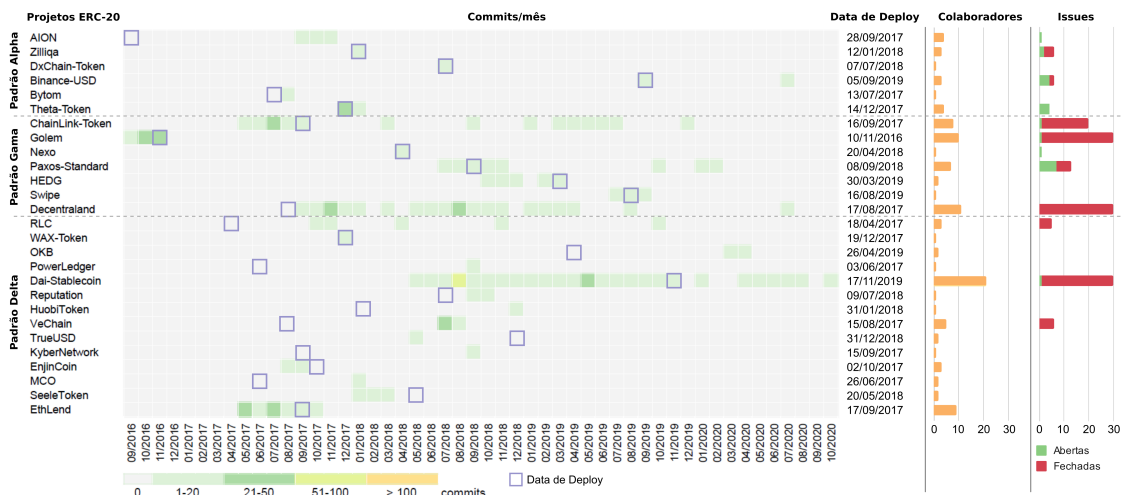


Figura 1. Análise de *commits*, colaboradores e *issues* dos projetos investigados.

4. Considerações Finais e Próximos Passos

Sob a forma de uma pesquisa exploratória-descritiva baseada em MRS, este estudo preliminar analisou aspectos colaborativos de 27 CIs implantados na Ethereum. Pautado numa análise quali-quantitativa, contribuiu-se ao investigar elementos colaborativos quanto ao desenvolvimento *open source* de CIs, como relação entre *commits* e colaboradores,

bem como o engajamento quanto ao *report* e resolução de *issues*. Como implicações, evidenciou-se que enquanto existem projetos com considerável nível de atividade (em termos de *commits*, colaboradores e *issues*) no respectivo repositório, existem iniciativas que aderem ao GitHub apenas como mecanismo de exposição do código-fonte. Como consequência desses achados, pretende-se aprofundar i) a análise do efeito que o processo de *deploy* na Ethereum exerce sobre os projetos *open source* em termos colaborativos, ii) na compreensão sobre o que efetivamente mudou nos repositórios após tal processo e iii) o entendimento quanto à criticidade e relevância de tais mudanças e de *issues*, a fim de compreender os benefícios da análise e da interação com a comunidade antes da *release*.

Referências

- Ajienka, N., Vangorp, P., and Capiluppi, A. (2020). An empirical analysis of source code metrics and smart contract resource consumption. *Journal of Software: Evolution and Process*, 32(10):e2267.
- Chakraborty, P., Shahriyar, R., Iqbal, A., and Bosu, A. (2018). Understanding the software development practices of blockchain projects: a survey. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–10.
- Chen, J., Xia, X., Lo, D., Grundy, J., and Yang, X. (2020). Maintaining smart contracts on ethereum: Issues, techniques, and future challenges. *arXiv preprint arXiv:2007.00286*.
- Costa, S., Araújo, A. A., and Souza, J. (2021). Investigando o efeito do deploy na ethereum em repositórios open source de contratos inteligentes: Uma proposta metodológica. In *Proceedings of the 13th Information Systems in Latin America (ISLA)*.
- Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system.
- Oliva, G. A., Hassan, A. E., and Jiang, Z. M. J. (2020). An exploratory study of smart contracts in the ethereum blockchain platform. *Empirical Software Engineering*, pages 1–41.
- Pinna, A., Ibba, S., Baralla, G., Tonelli, R., and Marchesi, M. (2019). A massive analysis of ethereum smart contracts empirical study and code metrics. *IEEE Access*, 7:78194–78213.
- Prikladnicki, R., Dittrich, Y., Sharp, H., De Souza, C., Cataldo, M., and Hoda, R. (2013). Cooperative and human aspects of software engineering: Chase 2013. *ACM SIGSOFT Software Engineering Notes*, 38(5):34–37.
- Risius, M. and Spohrer, K. (2017). A blockchain research framework. *Business & Information Systems Engineering*, 59(6):385–409.
- Rodrigues, A., Araújo, A. A., Paixao, M., and Soares, P. (2021a). Caracterizando a evolução de software de contratos inteligentes: Um estudo exploratório-descritivo utilizando github e etherscan. In *Anais do IX Workshop de Visualização, Evolução e Manutenção de Software*, pages 11–15. SBC.
- Rodrigues, A., Araújo, A. A., Paixao, M., and Soares, P. (2021b). Repositório de apoio: <https://zenodo.org/record/5216861>.
- Sillaber, C., Walzl, B., Treiblmaier, H., Gallersdörfer, U., and Felderer, M. (2020). Laying the foundation for smart contract development: an integrated engineering process model. *Information Systems and e-Business Management*, pages 1–20.
- Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Marchenko, E., and Alexandrov, Y. (2018). Smartcheck: Static analysis of ethereum smart contracts. In *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, pages 9–16.
- Zou, W., Lo, D., Kochhar, P. S., Le, X.-B. D., Xia, X., Feng, Y., Chen, Z., and Xu, B. (2019). Smart contract development: Challenges and opportunities. *IEEE Transactions on Software Engineering*.