

Understanding the Software Development Practices of Blockchain Projects: A Survey

Partha Chakraborty , Rifat Shahriyar , Anindya Iqbal , and Amiangshu Bosu

Department of Engineering and Technology, Dhaka, Bangladesh

Department of Computer Science, Wayne State University, Detroit, MI, USA

発表者：富田恭平

目次

1 要旨

2 導入

3 RQ

4 研究の方法について

5 回答者とそのプロジェクトの人口統計

6 研究の結果

7 調査結果の意味

8 結果の妥当性に対する脅威

9 結論

1 要旨

背景

ブロックチェーン技術はスマートコントラクト, IoT, 土地登録管理, アイデンティティ管理などの分野で期待されているため, ソフトウェアエンジニアリングの研究がいくつか行われている

目的

これらのギャップを埋めるため, 要求分析, タスク割り当て, テスト, ブロックチェーンソフトウェアプロジェクトの検証などのソフトウェアエンジニアリング手法を
探求する調査を目指す

方法

145の人気あるBCS（ブロックチェーンソフトウェア）プロジェクトのGithubリポジトリを
マイニングして特定した1604人のアクティブなBCS開発者にオンライン調査を行い,
分析の基準を満たす156の回答を受け取った

1 要旨

結果

BCS開発者間ではコードレビューと単体テストが効果的な開発手法であることが分かった

結論

テストやセキュリティを含むソフトウェアエンジニアリング手法を、
ブロックチェーン固有の性質に対処したより適切なものにする必要がある

ブロックチェーンについて

- 分散型, ピアツーピア, パブリック, 不変, 追加専用 のデータストレージ
- トランザクションと呼ばれる永続的なレコードを保持する
- 複数のトランザクションはブロックとしてグループ化される
- 各ブロックにproof-of-workアルゴリズムで計算されたハッシュと前のブロックのハッシュを含んでいる
- **ブロックのデータへの変更は、ハッシュの変更を起こすため不可能**
つまりブロックチェーンは**不可逆**で**不変**

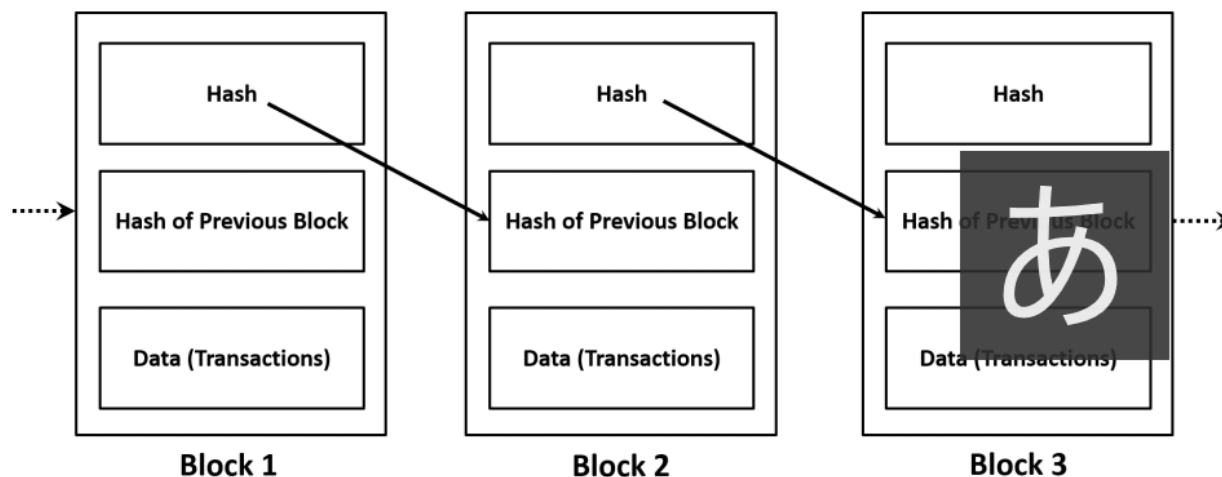


Figure 1: Simplified diagram of a blockchain

2 導入

- ・ ブロックチェーン中心のソフトウェア開発はまだ初期段階だが、需要により急速に拡大
- ・ 2018年1月現在、3,000人以上の開発者がGithub上の約2,087個のブロックチェーンリポジトリに定期的に貢献している。

しかし、急速に変化するエコシステムと激しい展開スケジュールにより、開発者自身や外部の専門家による**堅牢な**アーキテクチャ設計、コードレビュー、機能のテスト、パフォーマンス、スケーラビリティを行うことは**めったにない**

2 導入

適切なシステム設計, 徹底的なテストがなければ,
大規模な分散システムの高性能や拡張性, セキュリティ要件は満たされない

ブロックチェーンのような特殊な技術のための
ソフトウェアエンジニアリングツールと技術が必要
BlockchainOriented Software Engineering(BOSE)を導入すべきである。

アーキテクチャ設計と開発ツールが未熟ということは,
セキュリティの抜け穴を作りやすい

ブロックチェーン技術の変更不可能という性質は, 脆弱性が発見された場合,
回復がほぼ不可能である

2 導入

他にブロックチェーンの開発に重大な影響を与えるのは、
プロジェクトメンバー間の要求の収集と分析, 割り当て

- ・ ブロックチェーンアプリケーションを堅牢なものにするためには、BCSプロジェクトの現在のソフトウェアエンジニアリングプラクティスの不備を具体的に理解しなければならない
- ・ しかしブロックチェーン開発プロジェクトに対するソフトウェア工学手法に関する研究がない
- ・ ブロックチェーン中心の開発プラクティスに関する**開発者の意見を集めた**

3 RQ

RQ1：BCS開発者はどのソフトウェア開発プラクティスに従うのか？

BCSプロジェクトは、ブロックチェーン技術の急速な変化ペースに対応する必要がある

RQ2：BCS開発者は、ソフトウェアの正確性をどのように検証するのか？

BCSプロジェクトのコードの正確性を検証するため、課題とニーズを明らかにし、課題を克服する必要がある

RQ3：BCS開発者は、ソフトウェアのセキュリティとスケーラビリティをどのようにテストするのか？

必要なツールと関連知識の開発につながる

RQ4：BCS開発者は、プロジェクトの要件をどのように識別して選択するのか？

従来のソフトウェアプロジェクトのプロセスと異なるBCSプロジェクトの要件収集プロセスを理解する

RQ5：BCSプロジェクト間のタスク割り当て手順は？

過半数が様々な分野から参加するため、タスク割り当てプロセスは柔軟である必要がある

RQ6：BCS開発者のコミュニケーションチャンネルは何？

BCSプロジェクトの大部分はオープンソースであるため頻繁にコミュニケーションをとる必要があり、その際さまざまな通信プラットフォームを使用する可能性がある

4 研究方法

今回の調査では、

BCSプロジェクトのソフトウェア開発プラクティスに焦点を当てた質問を検討する

表1に各質問、その質問の動機となるRQ、各質問の回答の選択肢を示す

回答者に関する人口統計を収集するために、「RQ #」ではなく「D」で示される質問を含む

Table 1: Survey Questions

#	RQ*	Question Text	Answer Choices
Q1	D	How many years of software development experiences do you have?	[Less than a year, between one to five years, between six to ten years, more than ten years]
Q2	D	How many years have you been developing blockchain software?	[Less than a year, between one to two years, between three to five years, more than five years]
Q3	D	What is your primary Blockchain software project (i.e. the project that you have spent most of your time)?	[#]
Q4	D	What are your roles in your primary project (please check all appropriate roles)?	[Developer, Maintainer, requirement analysis, Testing, User support, Documentation, Social marketing]
Q5	D	Approximately, how many pull requests have you submitted to your primary project?	[Less than 10, Between 11 to 30, More than 30]
Q6	D	Approximately, how many hours on average do you spend per week on your primary project?	[Less than 5, between 6 to 10, Between 11 to 20, Between 21 to 35, I work full time]

Please answer the following questions based on your own experiences with your primary project (i.e., as responded in Q3.)

4 研究方法

Table 1: Survey Questions

#	RQ*	Question Text	Answer Choices
Q7	RQ6	Which of the following channels do you use to communicate with the peers from your primary project? Please check all that apply.	[email, Slack, Discord, Github, instant messenger, Skype, mailing list, Reddit, Medium, others (please specify)]
Q8	RQ1	Which of the following software development practices do you follow (Please check all that apply)?	[peer code review, automated build (aka one-click build using ANT, Maven, Gradle, CMake), continuous integration (e.g., Travis, Jenkins, Teamcity, Bamboo, Buildbot, or Cruisecontrol), pair programming, unit testing, automated testing, test driven development (aka test first development), performance testing, formal verification, others (please specify)]
Q9	RQ1	Please rank the following software development practices based on their effectiveness to improve the quality of your project (you can drag and drop to reorder the following list):	[formal verification, automated testing, pair programming, performance testing, test driven development (aka test first development), unit testing, continuous integration (e.g., Travis, Jenkins, Teamcity, Bamboo, Buildbot, or Cruisecontrol), peer code review, automated build(aka one-click build using ANT, Maven, Gradle, CMake)]
Q10	RQ4	How the requirements of your projects are identified and selected?	[#]
Q11	RQ5	How development tasks are assigned among the project members?	[#]
Q12	RQ2	How do you verify the correctness of your code?	[#]
Q13	RQ3	How do you test your software for security and scalability?	[#]

*numbers refer to the research question that motivated the inclusion of the survey question, 'D' refers to demographic questions

4 研究方法

次の4つの基準に基づいて、145のBCSプロジェクトを特定した。

- ・ ブロックチェーン, 暗号通貨, アルトコイン, イーサリアム, ビットコイン, スマートコントラクトの**6つのトピックの少なくとも一つタグ付けされている**
- ・ 10人以上のユーザーによる「**スター付き**」
- ・ 少なくとも**5人の貢献者**がいる
- ・ リポジトリの手動検証により, **リポジトリがBCSプロジェクトである**ことを確認した

Github APIを利用して、145のそれぞれのプロジェクトの一つに少なくとも5つの変更を送信した1604人の貢献者を特定した

Gitコミットログのマイニングから電子メールアドレスを収集した

4 研究方法

- ・パイロットサーベイ

SE経験のあるコンピューターサイエンスの教授と大学院生に質問を確認して、質問が簡潔で正確なものであることを確認した

フィードバックでは、マイナーな編集のみ提案された

行った変更

- ・いくつかの質問に選択肢を追加
- ・3つの質問に例を追加

4 研究方法

調査方法（RQ, 参加者の選択, 募集メール, 同意書, データ収集, データ管理）が, SIU (Institutional Review Board) によってレビュー/承認された

2017年12月13日 - リスト内の1,604人のBCS開発者に電子メールを送信した
すべてのメールが届いたとすると, 約62人分がバウンスし,
最大1,542人の参加者が残った

2017年12月21日 - リマインダーメールを送信した

2018年1月5日 - 調査を終了した

4 研究方法

- ・ 終了時, 調査URLは合計358回クリックされていた (招待の約23%)
- ・ これらの内, **200人**が約13% (200/1542) の回答率で**調査に参加した**
全ての質問に回答したのは115人だけ
- ・ 「少なくとも75%の質問」, 「少なくとも1つの自由回答式の質問」 の
いずれにも答えなかった44個の回答を除外して, **合計156個の回答が残された**

4 研究方法

- ・ 自由回答形式の質問は、体系的な**定性的データ分析プロセス**を用いた
- ・ 2人の著者が、各質問に対する75個の回答から一般的なテーマを抽出した
- ・ このテーマを用いて著者が書く質問に対して、
合意の取れたコーディング方式を開発するための討論会を行った
- ・ その結果、二人の著者はCoding Analysis Toolkit (CAT) ソフトウェアを用いて各解答を個別にコーディングした
- ・ **カッパ係数を用いて手動コーディングプロセスの評価者間信頼性のレベルを計算した結果、0.62と測定された**
- ・ 0.61から0.80の間の値は一般に「**実質的な一致**」として認識される
- ・ CATを使用してコーディングの不一致を特定し、すべての競合を解決するためのディスカッションセッションを行った

5 人口統計

Table 2: Projects represented

Multiple occurrences			Single occurrences			
Ethereum (22)	Bitcoin (9)	Bitshares (7)	Ambisafe	Basic Identity Token	Bytom	Cpuminer
Monero (6)	Sia (6)	Waves (5)	Dash	Distense	DNSChain	Ebets
Solidity (5)	Lbry (4)	Ripple (4)	ESKU	Etherbet	Etherplay	Fabric Labs
Nem (3)	Cardano (3)	Decred (3)	Golem	Haskoin	ZeroLink	Icofunding
EOS (3)	Hyperledger (3)	IOTA (3)	Ind	Iroha	JS Miner	Keyrun
Factom (2)	Feather coin (2)	Lisk (2)	Libsnark	LiteCoin	Payroll System	PHP-Mpos
Metamask (2)	Namecoin (2)	Neo (2)	Populus	Progmathon	Pycoin	Shapeshift
Remix IDE (2)	Stratis (2)	Trezor (2)	Snapcoin	Status	Steller	Storj
Zcash (2)	Undisclosed /Private (14)		Swiftly	Vandal	Vcash	Viper

回答者と回答者自身が代表するプロジェクトの人口統計について

- ・ 表2は回答者の主要プロジェクトに関するQ3（表1）の結果を示している
- ・ カッコ内の数字はそのプロジェクトをリストした回答者の数を表す
- ・ 回答者は61個の異なるBCSプロジェクトを代表している
- ・ 回答者の37%は上位10プロジェクトからきており、調査にトップBCS開発者がかなり参加していることを示している

5 人口統計

ソフトウェア開発経験 (Q1) , BCS開発経験 (Q2) , 役割 (Q4) ,
BCSプロジェクトへのコミットの合計数 (Q5) ,
BCS開発に費やした週あたりの平均時間数 (Q6)
に関する回答者の人口統計について

主なプロジェクトの役割について、回答者の大半が複数の役割を持っていた

「開発者」は回答者の中で最も一般的な役割(93%)で、「保守担当」(45%),
「QA」(34%)がそれに続いた

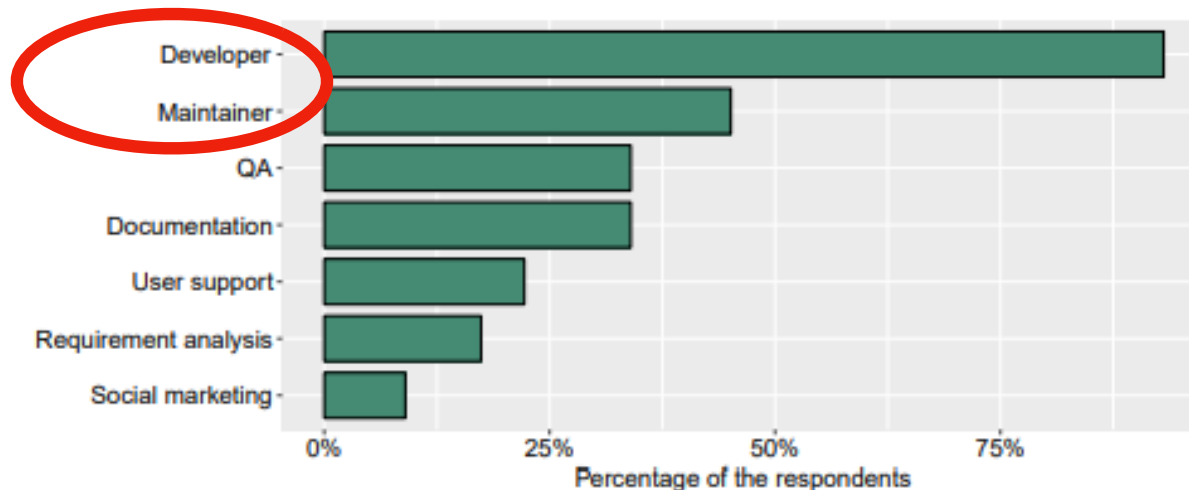
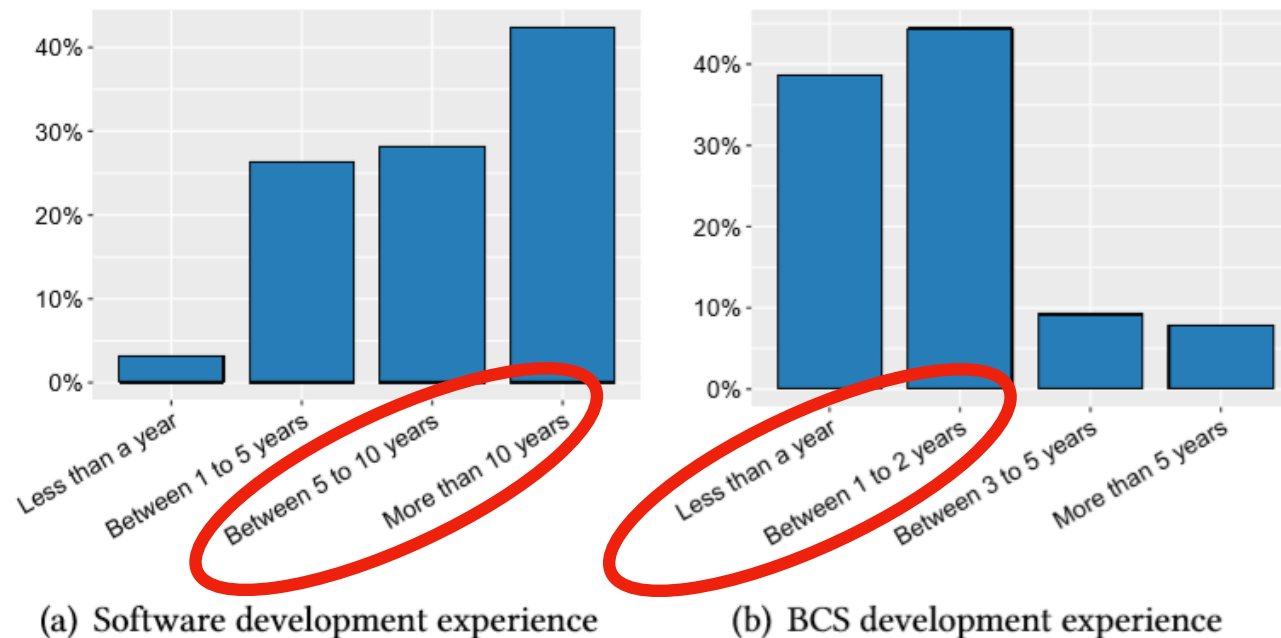


Figure 3: Roles of the survey respondents

5 人口統計

- ・ソフトウェア開発の経験に関して、回答者の70.5%は5年以上の開発経験があり、42.3%は10年以上ある
- ・BCS開発の経験に関して、回答者の81.4%は2年未満の経験があり、37.8%は1年未満
- ・BCS開発の経験がない多数のソフトウェア開発者が、最近BCSプロジェクトに参加したことを示す

Figure 2: Demographics of the respondents



5 人口統計

- ・ BCSプロジェクトへの貢献数について、回答者の57.6%が10件以上、42.3%が10年以上ある
- ・ BCSプロジェクトに費やされた時間について、回答者は32.7%がフルタイムで働いていて、42.7%は少なくとも1週間で20時間働いている
- ・ 回答者のコミット数と費やされた時間数を合わせて鑑みると、
回答者にはアクティブなBCS開発者のサンプルが含まれているといえる

Figure 2: Demographics of the respondents



6 研究結果

- ・ 6つのRQについて調査結果を示す
- ・ 定性分析では, 各自由回答は複数のコードに一致する可能性があるため, パーセンテージの合計は100%を超える場合がある

RQ1：ソフトウェア開発プラクティス

図4はBCS開発者の様々なソフトウェアエンジニアリング手法を示す

図4から, コードレビューが最も一般的な方法であることがわかる

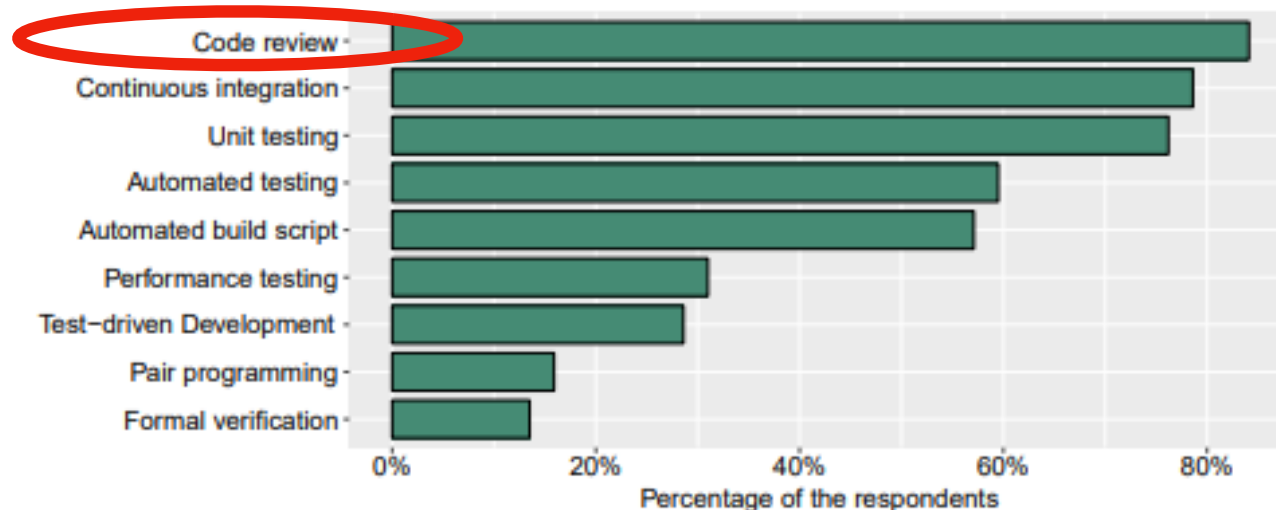


Figure 4: SE practices of BCS developers

6 研究結果

図5は、調査のQ8とQ9への回答から明らかになった有効性に基づいてランク付けされたもの

- ・ BCS開発者によると、コードレビューも最も効果的なものである
- ・ BCSプロジェクトでは適切なテストツールがないため、主に人間がバグを発見する

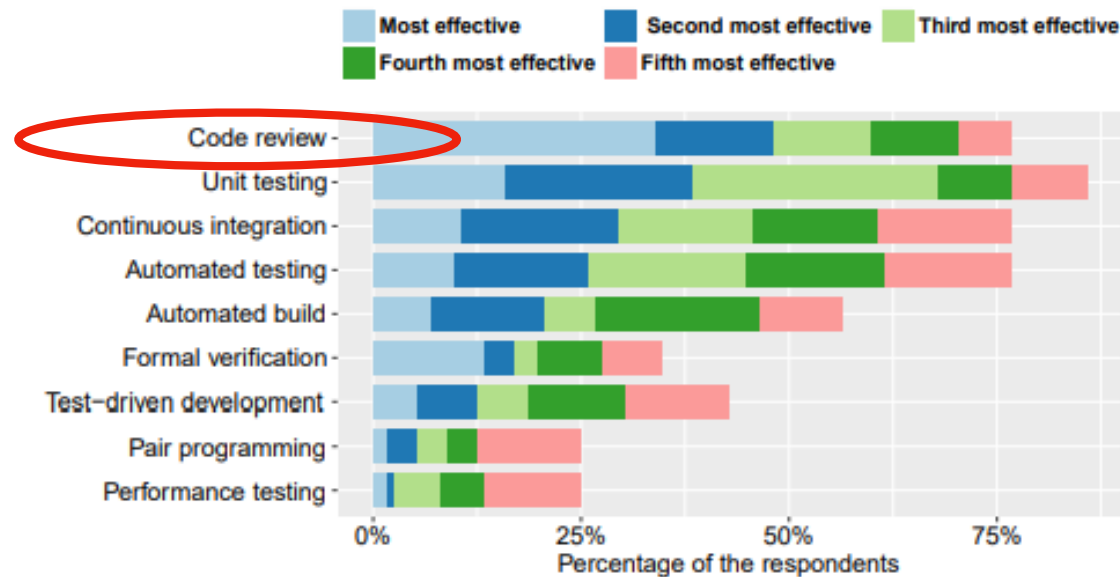


Figure 5: Ranks of software development practices based on effectiveness

6 研究結果

RQ2：検証と検証

RQ1は一般的にBCS開発者が従うソフトウェア開発プラクティスに対応し、
RQ2は**コードの正確性を検証するために使用したツール/手法**に対応する

- ・ 図6は調査のQ12への回答から明らかになった**BCSプロジェクトの正確性を検証されるために使用される様々な手法**を示す
- ・ BCS開発者はコードの正確性を検証するために、**コードレビューと単体テストを最もよく使用**している

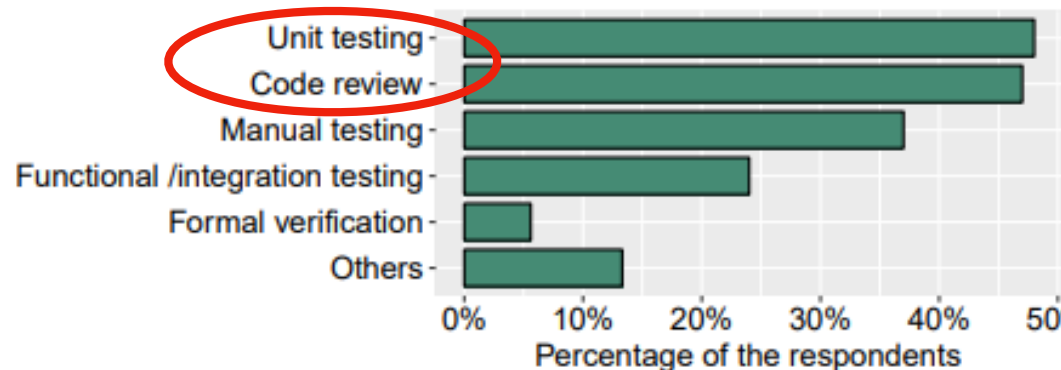


Figure 6: Different Technique to Verify Correctness of BCS projects

6 研究結果

RQ3：セキュリティとスケーラビリティ

図7はBCSプロジェクトのセキュリティとスケーラビリティをテストするために使用される様々な手法を示す

調査のQ13に対する回答から明らかになった

・BCS開発者はプロジェクトのセキュリティとスケーラビリティをテストする際に、コードレビュー, ストレステスト, ユニットテストを最もよく使用している

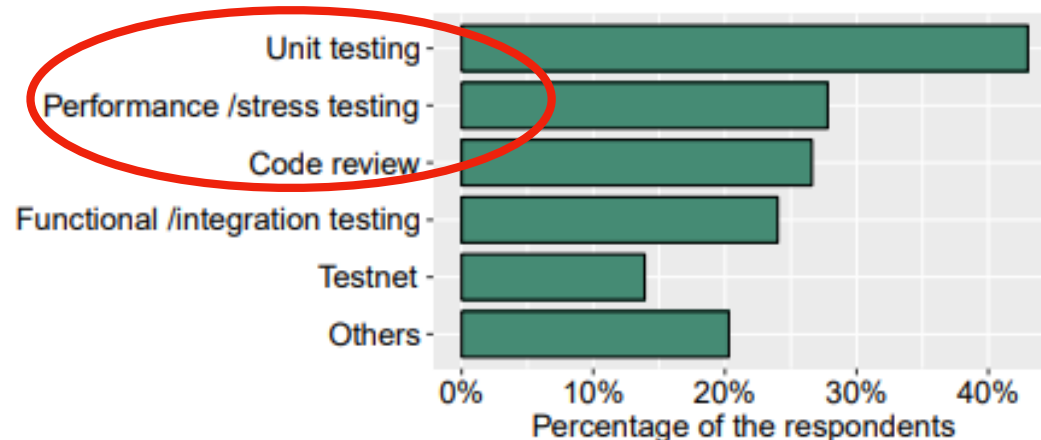


Figure 7: Different Technique for Testing Security and Scalability of BCS projects

6 研究結果

RQ4：要件分析

図8は**BCSプロジェクトの様々な要件選択プロセス**を示す調査のQ10への回答から明らかになった

- ・ BCSプロジェクトの要件が、**コミュニティディスカッションとプロジェクトリーダーによってほとんど特定/選択されている**ことを示す
- ・ 従来のOSSプロジェクトとBCSプロジェクトの要件は両方とも顧客のフィードバックを通じて特定されるが、BCSの場合**大部分はコミュニティディスカッションを通じて要件が特定される**

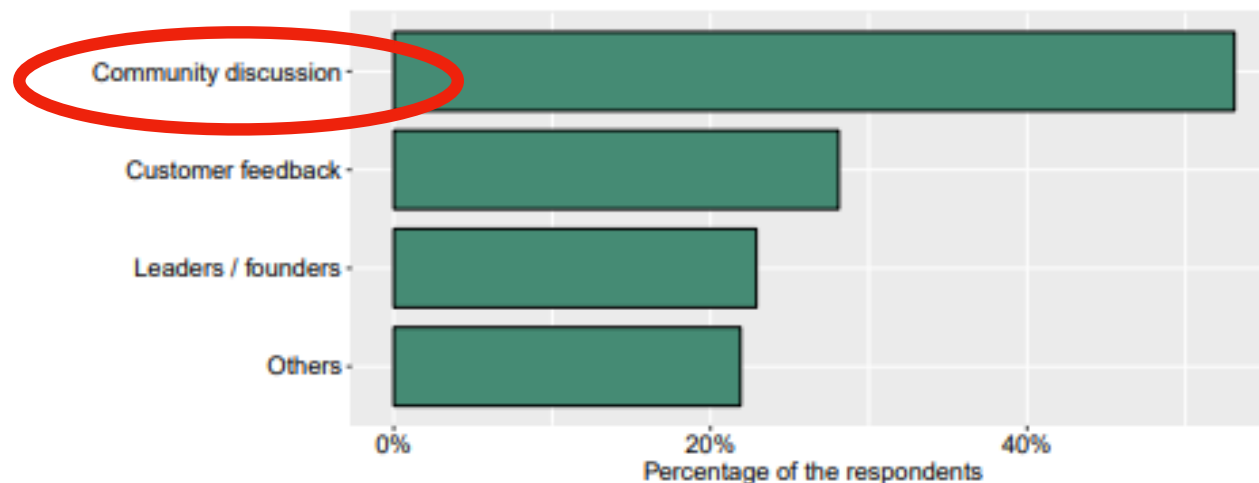


Figure 8: Requirements collection

6 研究結果

RQ5：タスクの割り当て

図9は, BCSプロジェクトの様々な開発タスク割り当てプロセスを示す調査のQ11への回答から明らかになった

- ・ 開発タスクが主に自発的に割り当てられることを示している
- ・ OSSプロジェクトは, 自発的な参加とタスクの自発的な選択が重要で, BCSプロジェクトは同様のタスク割り当て戦略に従っていることがわかる

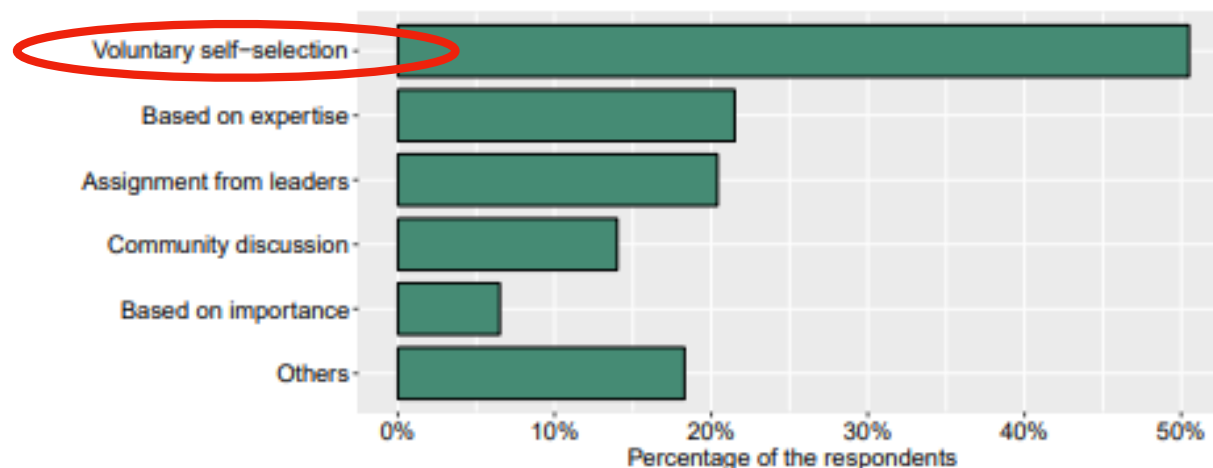


Figure 9: Different Task Assignment Process for BCS projects

6 研究結果

RQ6：通信チャンネル

図10は, BCS開発者が使用する様々な通信チャンネルを示している
調査のQ7への回答から明らかになった

- GithubとSlackが最も使用する通信チャンネルであることを示している
- 一般に, BCS開発者の通信チャンネルは, 従来のOSS開発者とは異なる

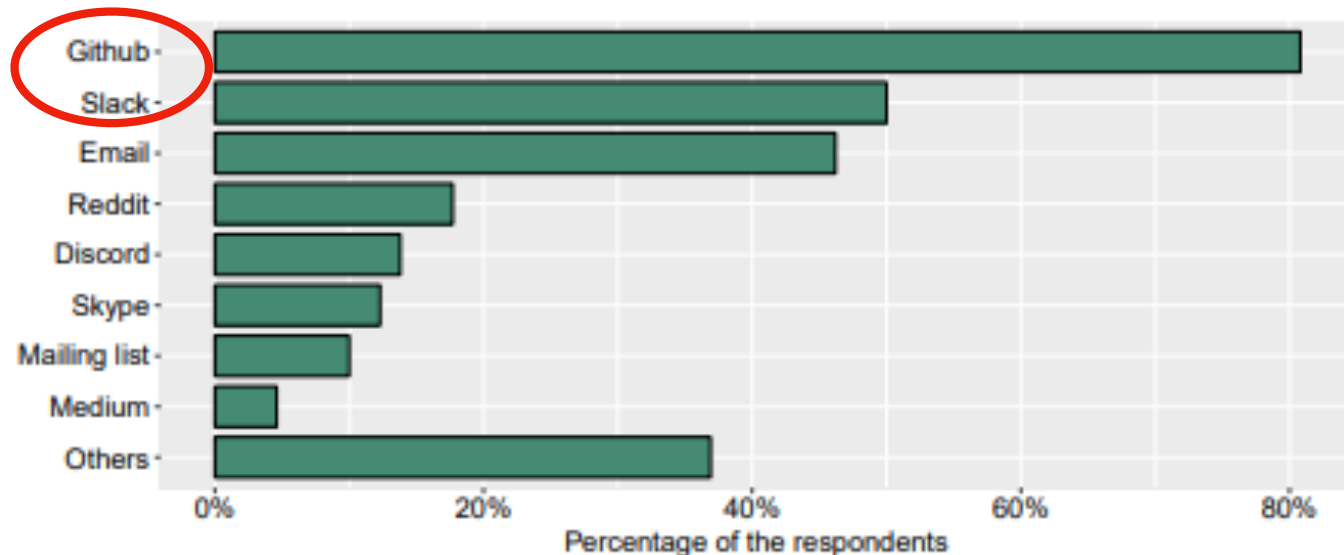


Figure 10: Communication channels of BCS developers

7 この研究が含む意味

- ・ この研究では, BCSプロジェクトにおいて, ソフトウェアエンジニアリング/セキュリティ/分散システムの研究者からの貢献が重要である分野がわかった
- ・ BCSは急速に拡大している競争の激しい分野であるため, ユーザーを引き付けるための**革新的なアイデア**, そして同時に**システムの信頼性と効率が損なわれないこと**が重要
つまり将来の技術に先んじて考えるコツを持っている研究者の協力が必要
- ・ BCSの開発者はアプリケーションの機能の正確性を検証するために,
主にコードレビューと単体テストの実践を行う必要がある
- ・ セキュリティ及びスケーラビリティのテストには**ストレステストを実施**
- ・ テストの自動化が難しいため, **主に手動での作業**を行う
- ・ 関連する研究コミュニティからの協力はBCSを堅牢かつ効率的にするのに重要

7 この研究が含む意味

- ・ この研究で, BCSプロジェクトの開発では, **従来のOSSプロジェクトの開発とは違う**慣行で行われていることが分かった
- ・ 一般的なOSSプロジェクトは開発者や顧客のフィードバックから要件を集めるが, **BCSプロジェクトは主にコミュニティディスカッション**で特定
- ・ BCSは開発タスクの大部分が分散型プロジェクトで自発的に制御されているため, 自発的に割り当てられ, **一般のOSSプロジェクトと似ている**
- ・ 重要な機能, スケーラビリティ, セキュリティのテストに関して, **BCSの特性に対処できる高品質のツールは存在せず, 高度に自動化されたテスト方法もない**

8 有効性に対する脅威

- ・ 調査の回答者は61種類の大規模BCSプロジェクトからきているが、
小規模なプロジェクトにおける開発では、この調査結果に対応しない
場合がある
- ・ 回答率がわずか13%ほどなため、潜在的に「**非応答バイアス**」が生じる
可能性がある
- ・ それでも、分析した156個の応答はこの論文で説明した洞察を明らかに
するデータソースとしては十分であるといえる

9 結論

- ・ ブロックチェーンプロジェクトの数は急激に増加しているが、
この分野の研究は足りていない
- ・ BCSの場合、一部の重要なテストやセキュリティ分析要件では、
技術的サポートの不足を感じる
- ・ この調査で報告されているような、**研究が貢献できる範囲のツールやソフトウェアエンジニアリングのメソッド、リソースが必要**