



卒業研究報告書

令和3年度

研究題目

プログラム理解における
着眼点の違いと性格特性

指導教員 上野秀剛 准教授

氏名 岩崎琴音

令和4年2月21日 提出

奈良工業高等専門学校 情報工学科

プログラム理解における 着眼点の違いと性格特性

上野研究室 岩崎 琴音

本研究の目的はコードレビューにおいて、性格特性がコードレビュー能力に影響を与えているのかを明らかにすることである。本研究では被験者実験を行い、コードレビュー能力と性格特性を測定した。性格特性はKT性格検査を用い5つの特性の傾向を測定した。コードレビューでは、指摘箇所と指摘時間を記録し、正答数、誤答数、発見までの時間、発見した種類、10分ごとに区切った時の正答数と性格特性との間にそれぞれ相関があるか分析を行った。これらの結果から、プログラムが設計書通りに正しく動作しない誤りに対してS型の傾向が高い人の方が見つけやすく、Z型の傾向が低い人の方が見つけにくいということが分かった。また、性格特性と40分間のコードレビューのうち10-20分間の正答数との間に、E型とN型に正の相関がみられることが分かった。

目次

1	はじめに	2
2	関連研究	3
3	KT性格検査	4
4	被験者実験	6
4.1	概要	6
4.2	提示物	6
4.3	実験手順	8
4.4	分析方法	8
5	結果と考察	10
5.1	性格特性と正答数	10
5.2	性格特性と誤答数	11
5.3	性格特性と種類ごとの正答数	12
5.4	性格特性と発見までの平均時間	13
5.5	性格特性と10分ずつ区切った時の正答数	14
5.6	外的妥当性への脅威	16
6	おわりに	17
	謝辞	18
	参考文献	19

1 はじめに

ソースコード中に混入している誤り（バグ）を発見するために、開発者がソースコードを精読する作業のことをコードレビューという。ソースコードの作成者以外の目でソースコードを見ることでコーディングの間違いや勘違いを発見したり、コンパイラでは検出できないソースコードの問題箇所を検出し、ソフトウェアの品質を向上させることができる。そのため、コードレビューはソフトウェアの品質を高めるうえで重要な作業である。コードレビューの効率を上げるために、さまざまな研究が行われている [1, 2]。例えば、Theelinらの研究において、個人によりコードレビュー能力の差が大きいことが示されている [2]。この能力差に影響する要因を明らかにすることができれば、要因による影響を小さくするためのサポートを行うことによって、能力の比較的低い開発者の能力を高めることができ、ソフトウェア開発プロジェクトの生産性の改善に役立つ。

本研究では、開発者の能力差に影響を与える要因として性格特性に着目する。理由として、几帳面で細かいところまで目が行く人はそうでない人と比べコードレビュー時でも誤りを見落としにくかったり、集中力が長く続く人は長いコードを読んでもペースが落ちない、といったことが考えられるからである。性格特性の違いがコードレビュー時に注目する箇所やバグの発見速度に影響を与えるか被験者実験を行い調査する。被験者実験では、性格診断とコードレビューを行う。性格診断にはKT性格検査を用いて、感情に左右されやすい自己開放型、持久力のある着実型など5つの性格特性を測定する。コードレビューでは発見したバグの種類や数、発見までの時間を測定する。性格特性の傾向によって、違いが見られるか分析する。本研究によって性格特性による違いが確認された場合、性格特性に応じた不得意な部分を補うツールの開発や、それぞれに得意なコードを見てもらうなど、個々のコードレビュー能力の向上に役立つ。またそれによってソフトウェア開発における品質の向上につながるといえる。

本論文は以下のような構成となっている。2章では関連研究について述べ、3章ではKT性格検査について述べる。4章で実験環境や実験手順などを示し、5章で実験結果を述べ考察を行う。6章ではまとめと今後の課題について述べる。

2 関連研究

コードレビューの効率を上げるために、さまざまな研究が行われている [1, 2]. 例えば, Thelinらの研究において, 個人によりコードレビュー能力の差が大きいことが示されている [2]. Thelinらの研究では, Checklist based reading (CBR)と Usage based reading (UBR)の2つのレビュー手法のうちどちらがバグを検出するのに効率的かつ効果的であるのかを比較する実験を行っている. その結果, バグ発見率はレビュー手法の違いによる影響よりも, 個人差による影響の方が大きく, 手法間でのレビュー効率の差は1.2-1.5倍であるのに対し, 個人差は3倍以上あった.

Alessandraらは, 性格特性がコードレビュー能力に影響していることを示している [3]. Alessandraらはコードレビュー課題の成績に相関する特定の性格タイプがあるかどうかを調べている. 性格タイプはMBTIを用いて測定しており, その結果, NTタイプがほかのタイプに比べ, 成績が優位にいいことが分かった. NTタイプとは, 情報を処理する際には人の気持ちや周りの人がどう思うかではなく, 物事に筋が通っているか・合理性があるかどうかで判断するタイプのことである. ただし, Alessandraらの研究では, 性格特性ごとの正答数に着目しており, レビュー中に発見する誤りの種類や, 発見速度などへの影響に対しては着目していない. そのため, 本研究では性格特性ごとの正答数に加え, 発見する誤りの種類や, 発見速度にも着目し性格特性が影響を与えているのか調べる.

3 KT 性格検査

本章では、本研究で性格特性を計るために用いるKT性格検査について説明する。

KT性格検査はドイツの精神医学者クレッチマー(Kretschmer,E)の精神医学的性格類型論に基づき、金子心理研究所によって開発された、5つの性格タイプの傾向を測定する検査である[4]。判定できる5つの性格タイプを表1に示す。KT性格検査で測定できる性格タイプは、S型(自己抑制型)、Z型(自己開放型)、E型(着実型)、N型(繊細型)、p型(信念確信型)の5つである。S型は、周りの人や周りの世界よりも、自分自身の内面の世界に関心を向けやすく、実際に行動するよりも考えることの方が自分に合っているといった傾向がある。Z型は感情に左右されやすく、気分で行動してしまう傾向がある。また、自分の喜びや悲しみといった感情をそのまま表現しがちで、考えるよりも行動が先行しやすいといった特徴がある。E型は粘り強く、忍耐力、持久力があるといった特徴がある。1つのことを途中でやめず、最後までやり抜く傾向がある。N型は感受性豊かで細かいところによく気がつく一方、ものごとを深く考えすぎてしまう傾向がある。P型は、自主性があり積極的で、実行力があるといった特徴がある。自分なりの信念を持っており、自己主張する傾向がある。

本研究では、表1に示した性格タイプのうち自己開放型、着実型、繊細型の3つが特にコードレビュー結果に影響を与えると考える。その理由を次に示す。

自己開放型において、このタイプは感情に左右されやすく気分で行動する傾向がある。標準的にはバグを見つけてることでモチベーションは上がっても、行動にはそれほど影響しないため、コードレビューにも影響しない。一方、自己開放型の傾向が高い人は行動が感情に左右されやすいため、モチベーションがコードレビューに影響すると考えられる。そのためコードレビューでは、誤りを見つけたときに達成感を得てモチベーションが上がることで、より誤りを見つけようとする。1つのソースコードに複数の誤りがあった場合、発見していない誤りの

表1 性格タイプ

記号	名称	傾向
S	自己抑制型	周りの人や周りの世界よりも、 自分自身の内面の世界に関心を向けやすい
Z	自己開放型	感情に左右されやすく、気分で行動してしまう
E	着実型	粘り強く、忍耐力、持久力がある
N	繊細型	感受性豊かで細かいところによく気がつく一方、 ものごとを深く考えすぎてしまう
P	信念確信型	自主性があり積極的で、実行力がある

多いレビュー序盤に、次々とバグを発見していくと考えられる。一方で、終盤にかけては残っている誤りが少なくなることで見つけづらくなり、モチベーションが下がることでさらに見つけられなくなると考えられる。

着実型において、このタイプは粘り強く、持久力がある。標準的にはレビュー終盤になるにつれ集中力が切れバグを発見できなくなっていく。一方、着実型は粘り強く、持久力があるといった特徴があるためコードレビュー中に集中力が切れにくく、レビュー中の序盤と終盤とのコンディションの違いが最も少ないと考えられる。よって、着実型は序盤から終盤まで安定したペースでバグを発見すると考えられる。

繊細型において、このタイプは細かい部分を気にする傾向がある。標準的には少し気になる部分があっても、考えてわからなければ次の部分を見ようとする。また、バグは見落としやすい。それに対し、繊細型は細かいところを気にしたり、ものごとを深く考えすぎてしまう。そのため気になる部分があると解決するまで考え続けると考えられる。また、自分が普段書いているコードと違う書き方のコード（他人が書いたコード）だと違い(変数名やインデントなど)が気になって、見慣れるまで時間がかかると考えられる。一方で注意深くコードを見るため細部のバグを見落としにくい。よって、繊細型は時間があれば他の型に比べて多くのバグを発見できると考えられる。

4 被験者実験

この章では、コードレビュー能力と性格特性との間に関連性があるかを調べるための実験について説明する。

4.1 概要

本実験は、コードレビュー能力と性格特性との間に関連性があるかを知るために、性格特性によるコードレビュー結果の違いを分析する。実験では、KT性格検査と、コードレビューを被験者に行ってもらい、コードレビューでは、誤りの発見と指摘を行ってもらい、Microsoft Wordのコメント機能を用いて指摘内容と発見時間を記録する。また、ソースコードはJavaで書かれており、被験者はクラスからインスタンスの作成やメソッド呼び出しについて学習する講義を受講済みである。本実験では、得意、不得意、学年の違いなどによるプログラミング能力の差は考慮していない。

4.2 提示物

本実験ではJavaで書かれたプログラムをレビュー対象とする。使用したレビュー対象物を表2に示す。

対象物は應治の研究[5]がコードレビューの実験で使用したソースコードと設計書である。Main.javaはファイルから物件データを取得、ユーザの入力条件にマッチする物件を表示する。House.javaはMain.javaからの呼び出しに応じて、駅からの距離や坪数、家賃などの物件情報を返す。設計書は、日本語で書かれており、プログラムの概要や各クラス、メソッドの働きについて書かれている。対象物はすべてPCに接続したディスプレイ上に表示し、被験者に提示する。プログラムのソースコードはMicrosoft Wordを用いて表示し、コメント機能で発見した誤りを指摘できるようにする。また、指摘が行いやすいように行数を表示する。設計書についてはテキストエディタで表示する。表3に、本実験で使用するソースコードに埋め込むバグの一覧を示す。バグについては應治の研究[5]で埋め込まれたものに加え、新たにコンパイラには通るが細かい部分まで見ないといけないようなバグを追加した。本研究では、バグを「動かない」、「表示間違い」、「理解が必要」の3種類

表2 実験で使用するプログラム

仕様	提示物	行数	バグ数
物件を表示する	Main.java	105	13
	House.java	30	1
	設計書	30	0

表3 ソースコードに埋め込むバグ

ファイル名	行数	ミス内容(正解→ミス)
Main.java	11	int i 不要
	25	終了[9]→終了[3]
	42	末尾「.」が抜けてる
	47	「下限」→「上限」
	49	「上限」→「下限」
	54	末尾「です。」が抜けてる
	59	坪数→壺数
	61	&&→&&&&
	64	→&&
	69	&&→&&&&
	75	&&→&&&&
	85	MAXHOUSE→100
91	&& i<MAXHOUSEが抜けてる	
House.java	27	rentの出力がない

に分けた。バグを種類別に分けたものを表4に示す。まず、設計書の概要どおりに正しく動作しないバグを「動かない」とした。例えば、演算子が「||」から「&&」になっているというようなものである。次に、正しく動作するバグのうち、設計書通りに表示しないバグを「表示間違い」、プログラムを理解しないと指摘できないものを「理解が必要」とした。「表示間違い」は設計書には「.」があるのに対しソースコードには「.」が無いといったものであり、「理解が必要」は「下限」と表示すべきところで「上限」と表示してしまっているというようなものである。また、バグの種類、内容は被験者には知らせていない。KT性格検査については、金子書房が出版する”KT性格検査”[4]を用いた。内容は50問の質問形式となっている。質問は、「一人でいる方が楽だ」や「今まで嘘をついたことはない」といった簡単なものである。回答者は質問に対して「はい」、「いいえ」「どちらでもない」の3つから解答する。

表4 バグの種類

バグの種類	行数(Main.java)
動かない	61, 64, 69, 75, 85, 91
表示間違い	42, 54, 59, 27(House.java)
理解が必要	11, 25, 47, 49

4.3 実験手順

本実験の手順を以下に示す。

1. 実験の説明・準備

実験および、注意事項についての説明を行う。注意事項としては、性格検査やコードレビュー中にほかの人と話さないこと、レビュー中に eclipse や Visual Studio Code といったツールを使わないことを説明する。

2. KT 性格検査

KT 性格検査の冊子を配り、10分間で冊子に書かれた指示に従い回答してもらう。

3. コードレビュー

プログラムの書かれた Microsoft Word ファイルとプログラムの説明が書かれたテキストファイルを配布し、コードレビューを40分間行ってもらう。コードレビューでは、被験者に誤りを見つけてもらう。誤りを見つけたら Word のコメント機能で指摘してもらう。指摘方法は以下の二通りとする。

- (a) 間違っている所をマウスで範囲選択し、コメントをクリックして『間違い』と書き込む。
- (b) 誤り付近にカーソルを移動させ、コメントをクリックして『?行目の「指摘場所」が間違い』と書き込む。

4.4 分析方法

実験結果の分析では、KT 性格検査とコードレビュー2つの結果から分析を行う。

KT 性格検査において、回答時に「はい」、「どちらでもない」、「いいえ」の順に2点,1点,0点が質問と関連のある性格特性のスコアとして加算され、このスコアが高いほど傾向が強くなる。KT 性格検査では、5つの特性の最終的なスコアを用いる。

コードレビュー結果では、以下に示す5つの結果を用いる。

- 正答数：[埋め込まれているバグの指摘数]
- 誤答数：[埋め込まれているバグとは違う部分の指摘数]
- 種類ごとの正答数：[表4に示す3つの種類ごとの正答数]
- 発見までの平均時間：[正答しているバグの発見時間の合計/正答数]
- 10分ずつ区切った時の正答数：[0-10分、10-20分、20-30分、30-40分のそれぞれの時間区間における正答数]

KT性格検査の結果とコードレビュー結果の間関係を見るために、相関係数を求める。また、それぞれ無相関の検定によって有意であるか確認する。

5 結果と考察

本研究では16人の被験者を対象に実験を行った。図1に被験者の性格特性ごとの人数分を示す。

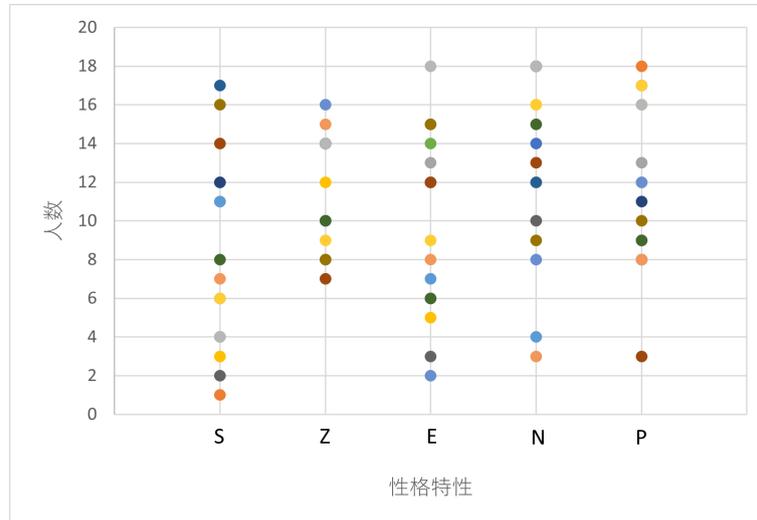


図1 被験者の性格特性ごとの人数分布

本研究で使用したKT性格検査では各性格特性ごとに12点以上が傾向ありと診断される。図5から本実験での被験者で傾向があったのはS型が5人、Z型、E型、P型が8人、N型が11人だった。

5.1 性格特性と正答数

表5に性格特性の結果とコードレビューでの正答数から計算した、各性格特性の相関係数とp値を示す。

表5から、S型(自己抑制型)と正答数の間には弱い正の相関が、Z型(自己開放型)との間には弱い負の相関がみられた。E型(着実型)との間には正の相関がみられ、N型(繊細型)との間にはほとんど相関がみられなかった。P型(信念確信型)との間にもほとんど相関はみられなかった。また、相関係数に対し無相関の検定(有

表5 性格特性と正答数の相関

性格特性	相関係数	p値
S	0.332	0.209
Z	-0.382	0.144
E	0.424	0.101
N	0.023	0.934
P	-0.140	0.606

意水準5%)を行ったところ、全ての相関係数に対し有意な相関はみられなかった。この結果に対して、まずS型は周りの人や周りの世界よりも、自分自身の内面の世界に関心を向けやすい、実際に行動するよりも考えることの方が自分に合っているといった傾向があります。そのためソースコードや説明書を理解し間違っているか考える必要のあるコードレビューに向いていると考えられる。よって弱い正の相関が出たと考えられる。次にZ型は感情に左右されやすく、気分で行動してしまう、考えるより行動が先行しやすいといった傾向がある。そのため、S型とは異なり考える必要のあるコードレビューに向いていないと考えられる。よって弱い負の相関が出たと考えられる。次にE型は粘り強く、忍耐力、持久力があり、一つのことを途中でやめないで最後までやり抜くといった傾向がある。そのため、E型の傾向が高い人ほど最後まで集中しバグを発見したと考えられる。また、真面目で几帳面といった傾向もあるため、日頃から苦手なものにもしっかり取り組み対処していると考えられる。よって、いろいろなバグに対して対応できたと考えられる。これらの理由から正の相関が出たと考えられる。次にN型は感受性豊かで細かいところによく気が付く一方、ものごとを深く考えすぎてしまう傾向がある。そのため細部まで見る必要のあるバグに対して特に正答数が高くなると考えていた。しかし、今回細部まで見る必要のあるバグとしてソースコードに埋め込んだものが単調だったため、N型の傾向が高くなっても一つ気づくことができれば他の似たバグを見つけることができたため、ほとんど相関が出なかったと考えられる。最後にP型は自主性があり積極的で、決断力、実行力がある、自分に自信を持ち負けず嫌いな傾向がある。そのためレビュー時に、積極的に探し最後まで順調にバグを見つけた人は正答数が高くなった、もうすべて見つけたと決定し終盤にバグを見つけられなかった人は正答率が低くなったなど、人によって自信や決断力による影響が異なったためほとんど相関が出なかったと考えられる。

また3章で述べた、Z型、E型、N型の順に正答数が高くなるという仮説に対して、E型は正の相関が出たが、Z型は弱い負の相関、N型はほとんど相関が出なかったため仮説通りの結果にはならなかった。

5.2 性格特性と誤答数

表6に性格特性の結果とコードレビューでの誤答数から計算した、各性格特性の相関係数とp値を示す。

表6から、S型(自己抑制型)と誤答数の間にはほとんど相関がみられなかった。Z型(自己開放型)との間には弱い負の相関がみられた。E型(着実型)、N型(繊細型)、P型(信念確信型)との間にはほとんど相関がみられなかった。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、全ての相関係数に対し有意な相関はみられなかった。この結果に対して、まずZ型について、5.1節で正答数との結果を見たとき、考える必要のあるコードレビューに向いていないと考えられるた

表6 性格特性と誤答数の相関

性格特性	相関係数	p 値
S	0.067	0.804
Z	-0.369	0.160
E	-0.067	0.804
N	0.093	0.731
P	-0.113	0.676

め弱い負の相関が出たと考えた。誤答数に関しても、考えることに向いおらず、そもそも発見自体が少なくなっているため誤答数も少なく、弱い負の相関が出たと考えられる。次にS型について、5.1節でS型は弱い正の相関があった。傾向が高いほど正答数が高くなったことを、ソースコードと設計書をよく理解しているためだとすると、傾向が高いほど誤答数は減ると考えられる。また、Z型の傾向が高いほど誤答数は減った。これを、考えることに向いておらず発見自体が少ないとすると、S型の傾向が低い人も似た傾向になり、誤答数が少ないと考える。これらから、傾向が高い人と低い人は誤答数が少なく、真ん中あたりの人は誤答数が多かったためほとんど相関が出なかったと考えられる。また、Z型は全員の数値が7以上と高めになっており1や2といった低い人はいなかったため相関が出たと考えられる。E型もS型と同じように傾向が高い人と低い人ほど誤答数が少なく、真ん中あたりの人ほどの誤答数が高かったためほとんど相関が出なかったと考える。次に、N型は細部まで見る必要のあるバグ以外で誤答する人、しない人がいたためほとんど相関が出なかったと考える。最後にP型は5.1節と同じようにレビュー時に、人によって自信や決断力による影響が異なったためほとんど相関が出なかったと考える。

5.3 性格特性と種類ごとの正答数

表7に性格特性の結果とコードレビューでの誤りの種類ごとの正答数から計算した、各性格特性の相関係数とp値を示す。

表7において、プログラムが正しく動作しない誤りを「動かない」、プログラムが正しく動くもののうち、設計書通りに表示しない誤りを「表示間違い」、プログラムを正しく理解できていないと指摘できない誤りを「理解が必要」としている。

表7から、S型(自己抑制型)と「動かない」との間には、正の相関がみられ、Z型(自己開放型)の間に負の相関がみられた。E型(着実型)の間には弱い正の相関がみられた。N型(繊細型)、P型(信念確信型)の間にはほとんど相関がみられなかった。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、S型とZ

表7 性格特性と種類ごとの正答数

性格特性	動かない		表示間違い		理解が必要	
	相関係数	p値	相関係数	p値	相関係数	p値
S	0.665	0.005	-0.022	0.937	-0.199	0.461
Z	-0.506	0.045	-0.035	0.898	-0.098	0.717
E	0.232	0.387	0.294	0.268	0.327	0.216
N	-0.035	0.897	0.010	0.970	0.096	0.724
P	-0.189	0.482	-0.210	0.435	0.191	0.479

型について有意差がみられた。その他については、有意な相関は見られなかった。

表7から、E型(着実型)と「表示間違い」との間には、弱い正の相関がみられた。S型(自己抑制型)、Z型(自己開放型)、N型(繊細型)、P型(信念確信型)の間にはほとんど相関がみられなかった。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、全ての相関係数に対し有意な相関は見られなかった。

表7から、E型(着実型)と「理解が必要」との間には、弱い正の相関がみられた。S型(自己抑制型)、Z型(自己開放型)、N型(繊細型)、P型(信念確信型)の間にはほとんど相関がみられなかった。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、全ての相関係数に対し有意な相関は見られなかった。これらの結果から、プログラムが設計書通りに正しく動作しない誤りに対してS型の傾向が高い人の方が見つけやすく、Z型の傾向が低い人の方が見つけにくいということが分かった。この結果に対して、まずS型について、5.1節でも述べたようにS型は行動するよりも考えることの方が自分に合っているといた傾向がある。そのためプログラミングに必要な論理的思考をするのに向いていると考えられる。よってプログラムを理解しないと指摘できない「動かない」バグに対して、正答数が高くなったと考えられる。Z型はS型値は逆に考えるよりも行動が先行しやすいといった傾向があり、論理的思考に向いていないため低くなったと考えられる。

5.4 性格特性と発見までの平均時間

表8に性格特性の結果とコードレビューでの誤り発見までの平均時間から計算した、各性格特性の相関係数とp値を示す。

表8から、S型(自己抑制型)と平均時間の間には弱い正の相関がみられた。Z型(自己開放型)、E型(着実型)、N型(繊細型)の間にはほとんど相関がみられなかった。P型(信念確信型)の間には弱い負の相関がみられた。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、全ての相関係数に対し有意な相関はみられなかった。この結果に対して、まずS型の傾向が高い人はレビュー時に5.3節でのバグの種類の中の「動かない」バグを最初に見つけ、大体見つけ終わった

表8 性格特性と発見までの平均時間

性格特性	相関係数	p 値
S	0.291	0.274
Z	-0.176	0.515
E	-0.084	0.757
N	-0.179	0.506
P	-0.217	0.419

後に「表示間違い」や「理解が必要」のバグを見つけている人が多かった。「動かない」バグはソースコード中の後半に多いため見つけるまでに時間がかかり弱い正の相関が出たと考えられる。次にZ型は感情に左右されやすく、気分で行動してしまう傾向がある。そのため個人個人でバグを探し出す場所が変わり、発見時間もバラバラになったことでほとんど相関が出なかったと考えられる。次にP型は決断力、実行力がある、自分に自信を持ち負けず嫌いな傾向がある。そのためレビュー時に、もうすべて見つけたと決定し終盤にバグを見つけられなかった、見つけなかった人が多かったため平均時間が短く弱い負の相関が出たと考えられる。平均では、10-30の間に2個見つけたのと0-10と30-40の間に1個ずつ見つけたのが同じくらいの数値になる。S型は序盤に時間がかかる、P型は終盤にあまり見つけていないといったように、レビュー序盤、終盤に少し偏っていたため相関が出たと考えられる。E、N型は中盤あたりに多く見つけるために平均では違いが出にくかったため、ほとんど相関が出なかったと考えられる。

5.5 性格特性と10分ずつ区切った時の正答数

バグを発見する時間が性格特性によって違う可能性がある。そこで、本実験のレビュー時間である40分を10分ずつ区切り、各区間の正答数と性格特性との関係进行分析することで、性格特性によってバグを発見しやすい時間があるのかどうかを調べる。表9に性格特性の結果とコードレビューでの10分ずつ区切ったときの正答数から計算した、各性格特性の相関係数とp値を示す。

表9から、全ての型と0-10分間での正答数との間においてほとんど相関がみられなかった。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、全ての相関係数に対し有意な相関はみられなかった。

表9から、S型(自己抑制型)、Z型(自己開放型)と10-20分間での正答数との間にはほとんど相関がみられなかった。E型(着実型)、N型(繊細型)の間には正の相関がみられた。P型(信念確信型)の間には弱い正の相関がみられた。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、E型とN型について有意差がみられた。その他については有意な相関は見られなかった。

表9 性格特性と10分ずつ区切った時の正答数

性格特性	0-10分		10-20分		20-30分		30-40分	
	相関係数	p値	相関係数	p値	相関係数	p値	相関係数	p値
S	-0.147	0.588	0.056	0.837	0.084	0.757	0.241	0.369
Z	0.076	0.778	-0.182	0.499	-0.140	0.605	-0.149	0.583
E	0.108	0.691	0.538	0.031	-0.098	0.719	-0.047	0.861
N	-0.051	0.852	0.504	0.046	0.002	0.995	-0.263	0.325
P	-0.087	0.749	0.242	0.367	0.024	0.929	-0.325	0.219

表9から、全ての型と20-30分間での正答数との間においてほとんど相関がみられなかった。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、全ての相関係数に対し有意な相関はみられなかった。

表9から、S型(自己抑制型)と30-40分間での正答数との間に弱い正の相関がみられた。Z型(自己開放型)、E型(着実型)の間にはほとんど相関がみられなかった。N型(繊細型)、P型(信念確信型)の間には弱い負の相関がみられた。また、相関係数に対し無相関の検定(有意水準5%)を行ったところ、全ての相関係数に対し有意な相関はみられなかった。

5.1節において、性格特性と40分間での正答数との間には全ての型において有意な相関は見られなかったが、10-20分の間では、E型とN型に正の相関がみられることが分かった。この結果の考察として、まず0~10分間に被験者それぞれがすぐに見つけられるバグを見つけていき、10分からは少し考える必要が合ったり、細部まで見る必要のあるバグを見つけ出したと考えられる。そのため、継続して考えられるE型と、細かいところまで見ることの出来るN型の傾向が高い人の正答数が高くなったと考えられる。20分以降はバグを見つけられない人や見つけても1個だけなどの人が増え、データが少なくなったため有意な相関がみられなかったと考えられる。図2に10分ごとのバグ発見数についてのグラフを示す。

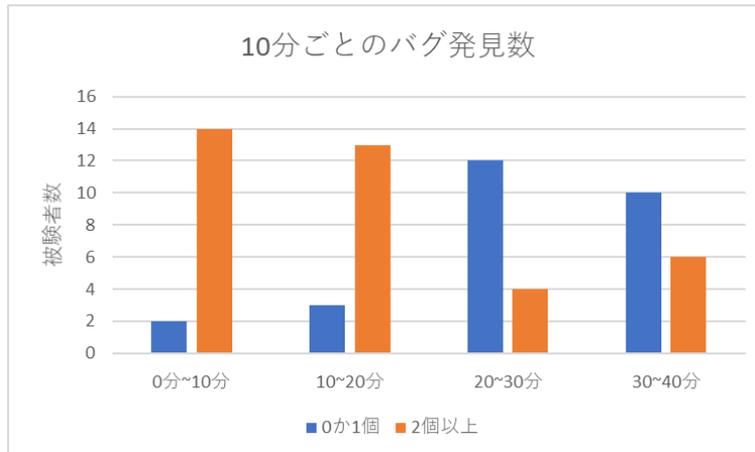


図 2 10 分ごとのバグ発見数

5.6 外的妥当性への脅威

本実験では被験者のレビュー方法を指定しない Ad-hoc Review という手法でレビューを行ってもらった。そのため、Checklist-Based Reading(CBR) といった Ad-hoc 以外のレビュー手法では本研究の結果がどれくらい使えるのはわからない。そのため、本節では Ad-hoc 以外のレビューでどのように本研究の結果が使えるのかについて考察する。

Checklist-Based Reading(CBR) 手法とはチェックリストを基にコードをレビューする方法のことである。この手法に対して、チェックリストが「何行目のこの部分、このメソッドの条件式に注意してください」といったようにソースコードのどこを見るのかを詳しく書いている場合は今回の性格特性によって発見しやすいバグの種類の結果は使えなくなると考えられる。一方で、チェックリストが「条件式、変数名に注意してください」といったようにざっくりとした指示の場合は本研究の結果のように S 型はプログラムを理解しないといけないバグを見つけやすいという結果は使えると考えられる。また、本実験では間違いの指摘だけを行ってもらい修正まではしていないため、「Yes/No」だけで答えるようなチェックリストに関しても同様に本研究の結果は使えると考えられる。本研究では、バグ発見時間について 10 分以降に E 型と N 型はバグを見つけやすいという結果になった。この結果に対して、チェックリストなどの指示があれば、集中力ややる気に影響が出ることで本研究と結果は変わると考えられる。

6 おわりに

本研究では開発者のコードレビュー能力と性格特性との間に関連性があるかを調べるため、被験者実験を実施した。被験者実験では性格特性とコードレビュー能力を測定した。性格特性では、KT性格検査を用いて自己抑制型、自己開放型、着実型、繊細型、信念確信型の5つの傾向をそれぞれどれくらい持っているのかを0-18の数値で測定した。コードレビューでは、正答数、誤答数、発見する誤りの種類、発見までの平均時間、10分ごとの正答数の5つを測定した。その結果、プログラムが正しく動かない誤りに対してS型の傾向が高い人ほど見つけやすく、Z型の傾向が低い人ほど見つけにくいということが分かった。また、40分間のコードレビューを開始後10-20分間ではE型とN型の傾向が高い人ほど多く見つけるということが分かった。以上のことから、性格特性と発見するバグの種類、発見しやすい時間との間には関連性があると考えられる。

今後の発展としては、今回の実験では被験者数が少なくデータ数があまりなかったため、被験者数を増やせばさらに信頼性の高い結果が得られると考えられる。また、性格特性を測るものを変えたり、レビューするコードの難易度を変えることでも今回測定できなかった性格特性の影響を明らかにすることができると考える。

謝辞

本論文の執筆および研究を進めるにあたり、多くの方々にご指導、ご協力を賜りました。指導教員である上野秀剛准教授にはお忙しい中、研究の知識やアドバイス、論文のチェックなど多くの面でご指導いただきました。内田教授には査読教員として多くのご指摘を頂きました。心より感謝いたします。研究や課題でお忙しい中、被験者として実験に協力して頂いた皆様にも深く感謝しています。ありがとうございました。

参考文献

- [1] 二宮凌真, "コードレビューの視線データの分析によるレビューパターンの傾向の違い", 奈良工業高等専門学校情報工学科令和2年度卒業研究論文 (2021).
- [2] Thelin, T., Andersson, C., Runeson, P., Dzamashvili-Fogelstrom, N, "A replicated experiment of usage-based and checklist-based reading", Proc. 10th Int'l Symposium on software Metrics (2004).
- [3] Alessandra Devito Da Cunha, David Greathead, "Does personality matter?: an analysis of code-review ability", Communications of the ACM, Vol.50, No.5, pp.109-112 (2004).
- [4] 金子心理研究所, "KT性格検査", 金子書房 (2006).
- [5] 應治沙織, "ソフトウェアレビューにおける読み方の教示によるレビュー効率の変化", 奈良工業高等専門学校情報工学科平成25年度卒業研究論文 (2014).