# ソフトウェア開発者の年齢がプログラム理解速度に及ぼす影響の分析

村上優佳紗[†1,a)]　角田雅照[†1,b)]　中村匡秀[†2,c)]

ソフトウェア開発において，個々のソフトウェア開発者の能力が開発プロジェクトの結果，例えば開発総工数などに与える影響は無視することができない．開発者の能力は個人により大きく異なることがいくつかの研究で指摘されている．開発者の能力差を小さくすることができれば，プロジェクトの生産性などを改善できる可能性がある．そのためには，能力差に影響する要因を明らかにし，その影響を小さくするためのソフトウェア開発支援ツールなどを用意する必要がある．本研究では，能力差が生じる要因として，開発者の年齢に着目し，年齢が記憶力に影響していると仮定する．そして，年齢が高い開発者の場合，理解のために記憶力を多く必要とするプログラムでは，理解速度が低下するかどうかを実験により確かめた．実験では，理解のために必要となる記憶力が異なる 4 つのプログラム用意し，低年齢の開発者 27 人と高年齢の開発者 8 人のプログラム理解速度を計測した．その結果，理解のために記憶力を必要とするプログラムの場合，高年齢の開発者のほうがプログラムを理解する速度遅いこと，高年齢の開発者では，記憶力を必要とするプログラムとそうでないプログラムの理解速度の差が大きくなることなどがわかった．

**キーワード**：プログラマ，開発スキル，コードリーディング

## 1. はじめに

近年，ソフトウェアは社会の多くの活動において広く利用されており，ソフトウェアの開発規模は年々大きくなっている．例えば，組込みソフトウェアの開発費は 2004 年度には 2 兆円であったが，2008 年度には 3 兆 5 千億円に増加した．さらに，同年には組み込みソフトウェアがインストールされた機器の生産額は製造業の生産額の 50%以上，GDP では 13%以上の比率となっている[8]．このため，ソフトウェア開発プロジェクトの（生産性低下，品質低下，納期遅延などの）失敗を避けることはより重要な課題となってきている．

ソフトウェア開発において，個々のソフトウェア開発者の能力がプロジェクトの結果，例えば開発総工数などに与える影響は無視することができない．例えば，開発工数見積もりに広く用いられる COCOMO II モデル[1]では，プログラマのスキルがモデルの要因に含まれている．これは，プログラマのスキルが異なる場合，その他の条件が同じ場合でも，プロジェクトの開発工数が変化する，すなわち生産性（開発規模÷開発工数で定義される）が変化することを示している．

ソフトウェア開発者の能力は個人により大きく異なることが，いくつかの研究で指摘されている．例えば，Sackman らの研究において，プログラムのデバッグ能力が，個人により大きな差があることが示されている[7]．同様に，Thelin らの研究においても，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[9]．もし開発者の能力差を小さくする，すなわち，能力が比較的低い開発者について，ある程度能力を高めることができれば，ソフトウェア開発プロジェクトの生産性などを改善できる可能性がある．

開発者の能力差を小さくするためには，能力差に影響する要因を明らかにし，その要因の影響を小さくするためのソフトウェア工学教育やソフトウェア開発支援ツールを用意する必要がある．例えば，個人によりコードレビュー能力に差があり，その原因が仕様と実装コードの対応をチェックしていないことが原因である[4]とする．この場合，「仕様と実装コードの対応をチェックする」ことをレビューのチェックリストに追加することにより，レビューの能力差を小さくすることができる．これまで，開発者の能力差の要因について分析した研究がいくつか存在する[10]．

本研究では，開発者の能力差が生じる要因として，開発者の年齢に着目する．すなわち，開発者の年齢が高い場合，プログラミングに関する何らかの能力が低下していると仮定して分析を行う．科学的な根拠は希薄であるが，開発者の能力と年齢の関連が指摘される場合がいくつか見られる．例えば，ある記事[11]では，年齢が若い開発者のほうが，新しいコーディング方法や技術の習得が早いと指摘している．また日本では，35 歳でプログラマとしての能力の限界を迎えるという「35 歳限界説」[12]が散見される．ただし，実際に年齢により開発者の能力が影響されるのかどうか，影響されるとすればどの能力なのかは，我々の知る限り，これまでほとんど分析されていない．

分析では，年齢に影響される開発者の能力は，記憶力であると仮定する．そして，年齢が高い開発者の場合，理解のために記憶力をより多く必要とするプログラムでは，そうでないプログラムと比較して理解速度が低下するかどうかを実験により確かめる．もし年齢と，理解のために記憶

†1 近畿大学理工学部
Faculty of Science and Engineering, Kindai University, Japan
†2 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University, Japan
a) m.yukasa@gmail.com
b) tsunoda@info.kindai.ac.jp
c) masa-n@cs.kobe-u.ac.jp

力を必要とするプログラムを読む速度に関連がある場合，年齢の高い開発者向けに記憶力をサポートする機能を持つ開発ツールが必要となる．そのようなツールを提供できれば，年齢の高い開発者とそうでない開発者の能力差を小さくすることができ，それによりソフトウェア開発プロジェクトの生産性を高められる可能性がある．

あるプログラムが，理解のためにどの程度記憶力を必要とするかを定量的に計測するために，本研究では文献[3][4]において提案されているプログラム理解容易性評価尺度を用いる．これらの尺度はプログラムの理解容易性を，プログラムの複雑度などではなく，理解に必要とする記憶力の多寡に基づいて評価している，すなわち，これらの尺度により理解が容易でないと評価されるプログラムは，理解のために記憶力がより多く必要であることを示している．実験では，年齢の高い開発者の場合，そのようなプログラムの理解速度が相対的に遅いかどうかを確かめる．

## 2. プログラム理解容易性評価尺度

プログラムの理解容易性を評価するために，文献[3][4]では，人間の記憶力に基づく尺度を提案している．これらの尺度では，プログラムを理解するためにメンタルシミュレーション[2]が行われることを前提としている．メンタルシミュレーションとは，プログラムの動作をコンピュータや筆記具を用いずに，思考しながら理解することであり，比較的規模の小さなコード片に対して用いられる．メンタルシミュレーションを行う場合，変数の値を記憶しておく必要があるが，多数の変数の値を同時に記憶しておくことは容易ではない．このため文献[3][4]では，多数の変数の値を記憶しておく必要のあるプログラムの場合，メンタルシミュレーションのコストが高くなり，プログラムの理解容易性が低下すると仮定している．

文献[4]では，人間の短期記憶を FIFO キューとして，メンタルシミュレーションの仮想モデル（Virtual Mental Simulation Model; VMSM）を作成し，そのモデルに基づいて理解容易性評価尺度を定義している．基本的なアイディアは，ある変数の値を参照する際，（大きさに制限のある）キューに変数の値が記憶されている場合はコストが小さくなるとしている．逆にキューに値が記憶されていない場合，その値の変更箇所にまでバックトラックする必要が生じるため，コストが大きくなるとしている．

文献[4]では，VMSM に基づき，以下の 4 つの評価尺度を定義している．

- ASSIGN: 変数代入に関するコスト．
- RCL: 短期記憶内の変数を思い出すコスト)．
- BT_CONST: 定数をバックトラックする回数．短期記憶にない定数を得るコスト．
- BT_VAR: 変数のバックトラックの距離．短期記憶にない変数を得るコスト．

```
int i , t;

t = 11;
t = t - 1;
i = 2;
if(i < t){
        i = i + 2;
        if(i < t){
                i = i + 2;
        }
        if(i < t){
                i = i + 2;
                if(i < t){
                        i = i + 2;
                }
        }
        if(i < t){
                i = i + 2;
        }
}
System.out.println("i = " +i);
```
(a0)

```
int i , t;

t = 11;
t = t - 1;
i = 2;
if(i < t){
        t = t - 2;
        if(i < t){
                i = i + 2;
        }
        if(i < t){
                t = t - 2;
                if(i < t){
                        i = i + 2;
                }
        }
        if(i < t){
                i = i + 2;
        }
}
System.out.println("i = " +i);
```
(a1)

図 1 プログラム a0, a1[3]

```
int a, b, c, d, e, f, g;

a = 2;
b = 4;
c = 3;
d = 6;

c = c + 4;
d = d - 2;
if(c < 5)
        e = d + 5;
else
        e = d + 3;
a = a * 2;
b = b + 6;
if(a > 7)
        f = b - 3;
else
        f = b - 5;
g = e + f;

System.out.println("g = "+g);
```
(b0)

```
int a, b, c, d, e, f, g;

a = 2;
b = 4;
c = 3;
d = 6;

c = c + 4;
d = d - 2;
a = a * 2;
b = b + 6;
if(a > 7)
        f = b - 3;
else
        f = b - 5;
if(c < 5)
        e = d + 5;
else
        e = d + 3;
g = e + f;

System.out.println("g = "+g);
```
(b1)

図 2 プログラム b0, b1[3]

文献[3]では，上記のメトリクスが変数の更新回数に基づく再計算コストを考慮していないことと，バックトラックに関するメトリクスがプログラムの行の入れ替えに敏感すぎることが問題点であるとし，新たな評価尺度を 2 つ提案している．具体的には，各変数の値の更新回数をベクトルの要素とし，ベクトルの要素の和に基づくメトリクス SUM_UPD とベクトルの要素の分散に基づくメトリクス SUM_VAR を定義している．

## 3. 実験

### 3.1 概要

実験の目的は，年齢が高いソフトウェア開発者の場合，理解のために記憶力をより多く必要とするプログラムでは，

表 1　各プログラムの理解容易性[3]

| プログラム | ASSIGN | RCL | BT_CONST | BT_VAR | SUM_UPD | VAR_UPD |
|---|---|---|---|---|---|---|
| a0 | 12 | 6 | 0 | 80 | 7 | 1.25 |
| a1 | 12 | 6 | 0 | 48 | 7 | 0.25 |
| b0 | 18 | 8 | 1 | 30 | 11 | 0.24 |
| b1 | 18 | 6 | 1 | 54 | 11 | 0.24 |

そうでないプログラムと比較して理解速度が低下するかどうかを確かめることである．そのために，必要とする記憶力が異なる，複数のプログラムを用意し，年齢の異なる被験者がコードを理解するために掛けた時間を計測した．

必要とする記憶量の異なるプログラムは，石黒らの研究[3]で示されているもの 4 つ（a0, a1, b0, b1．図 2，図 1 参照）を利用した．各プログラムは 20 行から 30 行の規模である．あらかじめ指定された変数の値が，プログラム実行後にどうなるかプログラムを読んで答え，それが正しかった場合，プログラムを理解できたとした．例えばプログラム a0 の場合, プログラム実行後の変数 i の値を答えさせた．プログラムの理解はメンタルシミュレーションにより行うこととし，メモなどは利用させないようにした．

各プログラムで理解のために必要とする記憶力の多寡については，2 章で説明した 6 つのメトリクス（ASSIGN, RCL，BT_CONST，BT_VAR，SUM_UPD，VAR_UPD）に基づき評価した．これらのメトリクスに基づく各プログラムの理解容易性を表 1 に示す（石黒らの研究[3]からの引用）．ASSIGN，BT_CONST，SUM_UPD より，プログラム b0，b1 を理解するためには，比較的記憶力が必要とされることがわかる．

被験者を低年齢グループと高年齢グループに分け，それぞれのグループの解答時間の平均値や中央値などを算出し比較した．低年齢グループは，近畿大学理工学部情報学科に所属する学部 4 年生から修士 2 年生の 24 名であり, 高年齢グループは同学科の教員 8 名である．高年齢グループの平均年齢は 45 歳（最小 33 歳から最大 64 歳）である．

プログラムを読む順番が実験結果に影響することを避けるために，4 つのプログラムを読む順番を被験者ごとに変更した．例えばある被験者ではプログラムを a0, a1, b0, b1 の順に読むとし，別の被験者では b1, a0, b0, a1 の順に読むなどとした．

分析にあたり，以下の 2 つのリサーチクエスチョンを設定した．

- RQ1: 高年齢グループと低年齢グループでは，どちらがプログラムを理解する速度が速いのか?
- RQ2: 理解のために記憶力を必要とするプログラムの場合, 高年齢グループのほうがプログラムを理解する速度が速いのか?
- RQ3: 高年齢グループは，記憶力を必要とするプログラムとそうでないプログラムの理解速度の差が大きいのか?

### 3.2 実験用ツール

実験のために，問題を出題し，回答時間や誤回答の回数を計測するためのツールを作成した．図 3 にツールのスクリーンショットを示す．本ツールは表計算ソフトのマクロを用いて開発した．ツールの動作を以下に示す．

1. 「解答する」をクリックすると問題とテキストボックスを表示する．
2. 問題に正解するまでテキストボックスを表示し続ける．

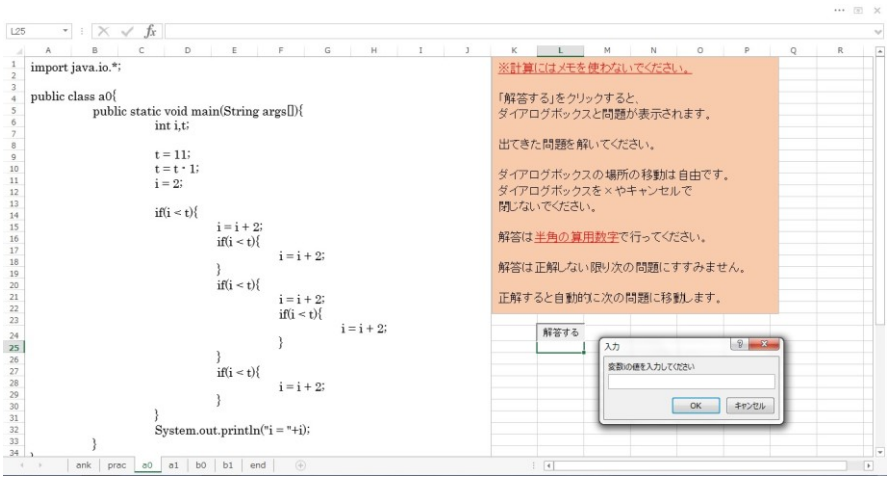


図 3　実験用ツールのスクリーンショット

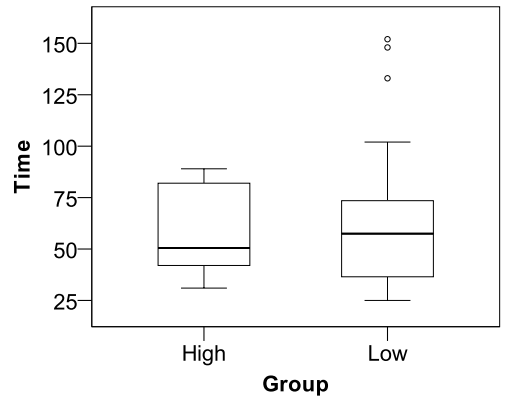


図 4　プログラム a0 の回答時間

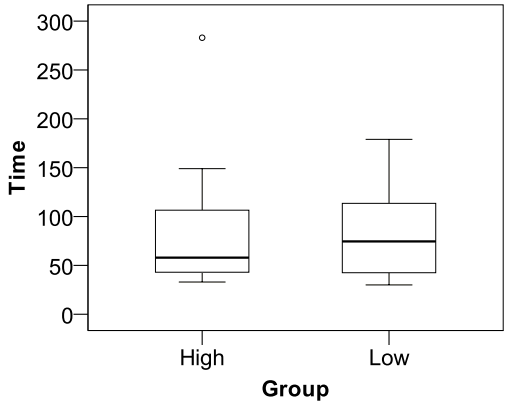


図 5　プログラム a1 の回答時間

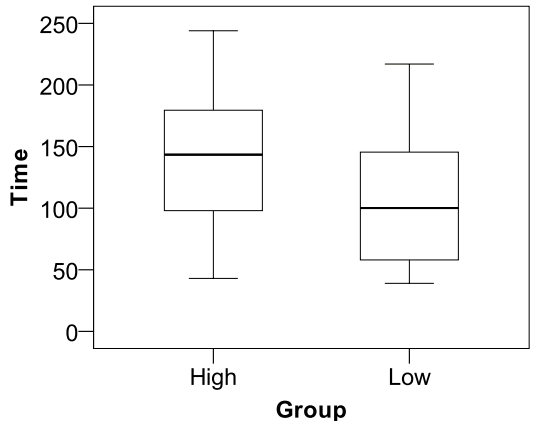


図 6　プログラム b0 の回答時間

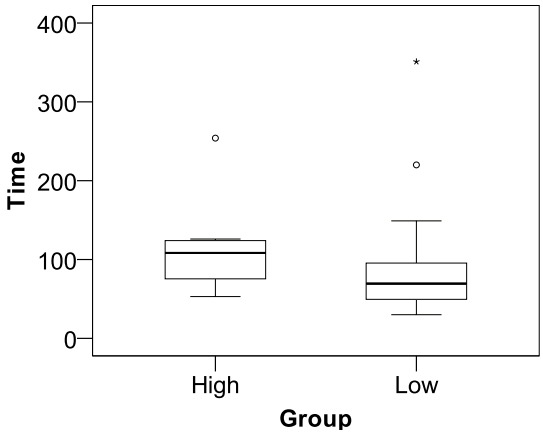


図 7　プログラム b1 の回答時間

表 2　各グループの回答時間（秒）

| | | a0 | a1 | b0 | b1 |
|---|---|---|---|---|---|
| 低年齢 | 平均値 | 65 | 82 | 105 | 89 |
| | 中央値 | 58 | 75 | 100 | 70 |
| 高年齢 | 平均値 | 59 | 91 | 141 | 115 |
| | 中央値 | 51 | 51 | 144 | 109 |

表 3　プログラム a0 との回答時間の比

| | | a1 / a0 | b0 / a0 | b1 / a0 |
|---|---|---|---|---|
| 低年齢 | 平均値 | 1.6 | 2.0 | 1.4 |
| | 中央値 | 1.2 | 1.5 | 1.4 |
| 高年齢 | 平均値 | 1.5 | 2.4 | 2.2 |
| | 中央値 | 1.0 | 2.4 | 2.4 |

3. テキストボックス（問題）が表示されてから正答するまでの回答時間と誤回答の回数を記録し，被験者にも表示する．
4. 正解すると次の問題に自動的に遷移する．

### 3.3 結果

表 2 に各グループの回答時間の平均値と中央値を示す．平均値に着目した場合はプログラム a0 以外で低年齢グループの回答時間の方が短く，中央値に着目した場合はプログラム b0, b1 で低年齢グループの回答時間が短かった．それぞれのグループの回答時間の分布を，箱ひげ図を用いて図 4 から図 7 に示す．プログラム a0 場合，低年齢グループの箱の位置は低かったが, 中央値および外れ値が大きく，どちらのグループの回答時間が短いとはいえなかった．プログラム a1 の場合，高年齢グループのほうが，低年齢グループよりも若干ながら回答時間が短い傾向があった．プログラム b0, b1 の場合，箱ひげ図おいても，低年齢グループのほうの回答時間が明確に短かった．これらの結果より，記憶力を特に必要とするプログラムでは，低年齢グループのほうの理解速度が速いといえる．従って，RQ1 に対する答えは「プログラムによって異なる」，RQ2 に対する答えは「はい」となる．

次に，RQ3 に答えるために，比較的記憶力を必要としないプログラム a0 と比べ, 何倍の回答時間が掛かっているかを調べた．具体的には，プログラム a0 以外の回答時間÷プログラム a0 の回答時間を求めた．この値が大きいほど，a0 と比較して回答時間が多く掛かっていることを示している．結果を表 3 に示す．低年齢グループでは，b0 / a0，b1 / a0 の平均値と中央値のうち，b0 / a0 の平均値以外は 2 を下回っていたが，高年齢グループではそれらの値全てが 2 を上回っていた．このことから，高年齢グループのほうが，理解のために記憶力を必要とするプログラムの理解速度が，相対的に遅いといえる．よって，RQ3 に対する答えは「は

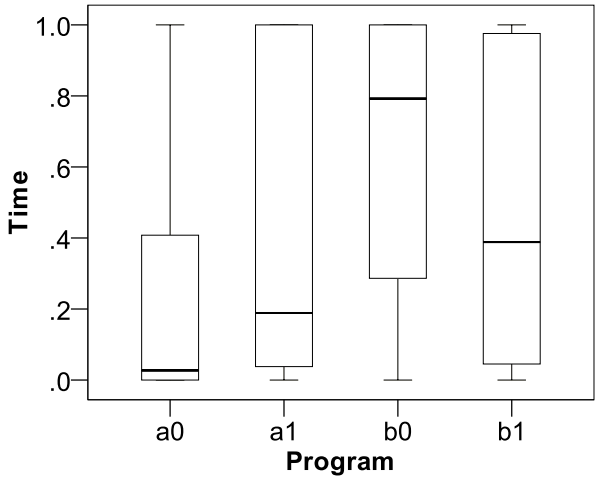


図 8　低年齢グループの回答時間正規化

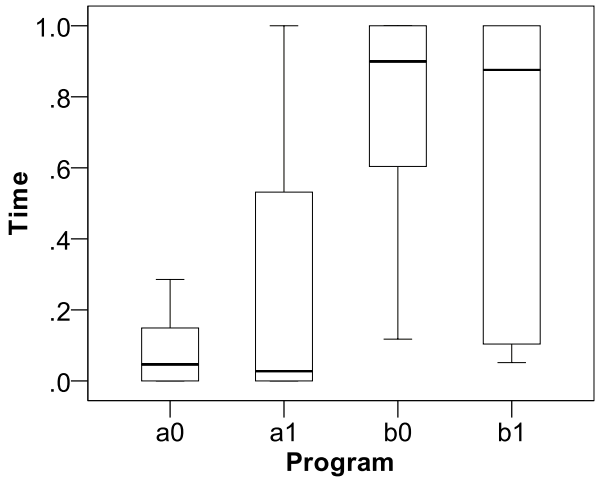


図 9　高年齢グループの回答時間正規化

い」となる．

RQ に対する分析を別の観点で行うために，被験者ごとに解答時間を正規化して比較した．正規化は(回答時間 – 最小解答時間)÷(最大回答時間 – 最小回答時間)により行った．箱ひげ図を図 8，図 9 に示す．低年齢グループの場合，プログラム a0 以外はばらつきが大きい，すなわち記憶力を比較的多く必要とするプログラム b0，b1 の場合でも，相対的に回答時間が長くなるとは限らなかったが，高年齢グループの場合，特にプログラム b0 において相対的に回答時間が長い傾向があった．

表 4　誤回答数の基本統計量

| | | a0 | a1 | b0 | b1 |
|---|---|---|---|---|---|
| 低年齢 | 平均値 | 1.0 | 0.5 | 1.3 | 0.8 |
| | 中央値 | 0.0 | 0.0 | 0.0 | 0.0 |
| | 標準偏差 | 2.5 | 0.8 | 2.2 | 2.3 |
| 高年齢 | 平均値 | 0.1 | 0.4 | 1.1 | 0.5 |
| | 中央値 | 0.0 | 0.0 | 1.0 | 0.0 |
| | 標準偏差 | 0.4 | 0.7 | 1.0 | 0.8 |

誤回答数の基本統計量を表 5 に示す．プログラム b0 を除き中央値は 0 であり，どちらのグループも誤回答を多く行った被験者は少ないことがわかる．プログラム b0 については，高年齢グループの中央値が 1 となったことから，このプログラムは高年齢グループにとって，特に理解が難しかったと考えられる．なお，低年齢グループでは，一部の被験者の誤回答数が多く，プログラム a1 を除き，標準偏差が 2 を超えていた．ただし，高年齢グループと比較して，平均値と中央値に大きな差はないため，全体としては正答率に高年齢グループと差はないといえる．

### 3.4　考察

高年齢グループの被験者が，実験に集中して取り組んでいなかった可能性は低いと考えられる．これは，プログラム a0 の出題順序が被験者によって異なっていたにも関わらず，a0 の回答時間が低年齢グループよりも低かったためである．また，仮に実験に十分に集中して取り組んでなかったとしても，表 3 の結果より，記憶力を必要とするプログラムは，高齢者グループでは相対的に理解速度が遅いといえる．

実験後のアンケートで出題したプログラムについてアンケートを行った．その内容は以下の 2 つである．

I. 1 問目と 3 問目について，具体的にどの部分がどのように変更されたかについて気づいたか．
II. 2 問目と 4 問目について，具体的にどの部分がどのように変更されたかについて気づいたか．

回答は以下の 3 つとした．

1. 気づいたうえで早く読めたと思う
2. 気づいたが早く読むのに役立たなかった
3. 気づかなかった

低年齢グループの有効解答数は 24，高年齢グループの有効解答数は 7 であった．アンケート結果を表 4 に示す．低年齢グループと比較して，高年齢グループは問 I，問 II とは「1 : 気づいたうえで早く読めたと思う」と解答した被験者が多かった．図 9 において，高年齢グループのプログラム a1 の回答時間のばらつきが比較的小さかったのは，このことが影響している可能性がある．記憶力を必要とするプログラム b0, b1 について，もし回答 2, 3 が多かった場合，回答時間が低年齢グループと比較して，より長くなった可能性がある．

## 4. 関連研究

ソフトウェア開発者の能力は個人により大きく異なることが，いくつかの研究で指摘されている[7][9][10]．例えば，Sackman らの研究において，プログラムのデバッグ能力が，個人により大きな差があることが示されている[7].

表 5　アンケート結果

| | 問 I | | | 問 II | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 2 | 3 |
| 低年齢 | 43% | 43% | 14% | 33% | 50% | 17% |
| 高年齢 | 57% | 29% | 14% | 46% | 46% | 8% |

ただし，年齢が開発者の能力に影響するのかどうか，影響するとすればどの能力なのかについて分析した研究は，我々の知る限り，Morrison ら[5]の研究など，ごく少数である．

Morrison らはソフトウェア開発者の知識と年齢が関係するかどうかを Stack Overflow という Q&A サイトのデータを用いて分析している．その結果，開発者の評価が年齢とともに上昇することなどを示している．ただし，Morrison らの研究では開発者の知識に着目しているが，記憶力には着目しておらず，年齢とプログラムの理解速度の関係などは明らかにしていない．

## 5. おわりに

本研究では，ソフトウェア開発者の能力に影響する要因として，開発者の年齢に着目した．分析では，年齢に影響される能力は記憶力であると仮定し，年齢が高い開発者の場合，理解のために記憶力をより多く必要とするプログラムでは，そうでないプログラムと比較して理解速度が低下するかどうかを確かめた．実験の結果より，以下が明らかとなった．

- 高年齢グループと低年齢グループでは，プログラムの理解する速度は，どちらかが常に速いとはいえない．
- 理解のために記憶力を必要とするプログラムの場合，高年齢グループのほうがプログラムを理解する速度遅い．
- 高年齢グループでは，記憶力を必要とするプログラムとそうでないプログラムの理解速度の差が大きかった．

今後の予定は，記憶力以外の，年齢に影響される能力を特定し，年齢の能力に対する影響の大きさを明らかにすることである．

**謝辞**　本研究の一部は，文部科学省科学研究補助費（挑戦的萌芽：課題番号 26540029，基盤 C：課題番号 25330090）による助成を受けた．

## 参考文献

[1] Boehm, B., Clark, Horowitz, Brown, Reifer, Chulani, Madachy, R. and Steece, B.: Software Cost Estimation with Cocomo II, Prentice Hall PTR (2000).
[2] Dunsmore, A., Roper, M. and Wood, M.: The role of comprehension in software inspection, Journal of Systems and Software, vol.52, no.2–3, pp.121-129 (2000).
[3] 石黒誉久，井垣宏，中村匡秀，門田暁人，松本健一：変数更新の回数と分散に基づくプログラムのメンタルシミュレーションコスト評価，電子情報通信学会技術報告，SS2004-32，pp.37-42(2004).
[4] 栗山進，大平雅雄，門田暁人，松本健一：プログラム理解度がコードレビュー達成度に及ぼす影響の分析，電子情報通信学会技術報告，SS2004-53．pp.17--22 (2005).
[5] Morrison, P. and Murphy-Hill, e.: Is programming knowledge related to age? an exploration of stack overflow, In Proc. of Working Conference on Mining Software Repositories (MSR), pp.69-72, 2013.
[6] Nakamura, M., Monden, A., Satoh, H., Itoh, T., Matsumoto, K., and Kanzaki, Y.: Queue-based Cost Evaluation of Mental Simulation Process in Program Comprehension, Proc. of International Software Metrics Symposium, pp.351-360 (2003).
[7] Sackman, H., Erikson, W. and Grant, E.: Exploratory experimental studies comparing online and offline programming performance, Communications of the ACM, vol.11, no.1 (1968).
[8] 田丸喜一郎：10 年で変わった「組込み」、変わらない「組込み」，ZIPC WATCHERS，pp.6-7 (2008).
[9] Thelin, T., Andersson, C., Runeson, P., Dzamashvili-Fogelström, N.: A Replicated Experiment of Usage-Based and Checklist-Based Reading, Proc of International Symposium on Software Metrics, pp.246-256 (2004).
[10] Uwano, H., Nakamura, M. Monden, A. and Matsumoto, K.: Exploiting Eye Movements for Evaluating Reviewer's Performance in Software Review, IEICE Transactions on Fundamentals, vol.E90-A, no.10, pp.317-328 (2007).
[11] Wadhwa, V.: Silicon Valley’s Dark Secret: It’s All About Age, http://techcrunch.com/2010/08/28/silicon-valley%E2%80%99s-dark-secret-it%E2%80%99s-all-about-age/ (2010).
[12] 山田モーキン：Web エンジニア「35 歳限界説」は本当か？ http://next.rikunabi.com/01/closeup_1251/index.html