Are Unreachable Methods Harmful? Results from a Controlled Experiment

Simone Romano, Christopher Vendome, Giuseppe Scanniello, and Denys Poshyvanyk
software evolution
changes cause a decay of software design
Software decay increases the **effort** and the **cost** to understand and maintain code.
what is one of the therapy for software decay?
a lightweight therapy

“The process of changing a software system in such a way that it does not alter the external behaviour of the code yet improves its internal structure”

--M. Fowler
what are the symptoms?
bad smells in code

“Symptoms of poor design or implementation choices”

--M. Fowler
many kinds of bad smells in code
Dead code refers to computations whose results are never used

--S. K. Debray et al.
“Code that is unreachable can never be executed”

--S. K. Debray et al.

“dead” different from “unreachable”
software engineering

“Dead code is code that isn’t executed”

--R. C. Martin

dead instead of unreachable
to avoid confusion we will use the term **unreachable**
public class M{
    public static void main(String[] args){
        C c = new C1();
        c.m1();
    }
}

public class C{
    public void m1(){...}
}

public class C1 extends C{
    public void m1(){
        this.m2();
    }
    private void m2(){...}
    private void m3(){...}
}
percentage of unreachable methods ranges from 5% to 10%
so what?
A long-term investigation
are unreachable methods really harmful?
how actual developers perceive unreachable methods?
how do developers deal with unreachable methods?
when and why are unreachable methods introduced and/or removed
Impact of unreachable methods on comprehensibility and modifiability
Analyze their presence for the purpose of evaluating their effect w.r.t. comprehensibility of unknown code and w.r.t. modifiability of familiar code from the point of view of researchers and practitioners in the context of novice developers and Java code.
design: one factor with two treatments
NoUM vs. UM
costructs: correctness of understanding and of modification, and effort
tasks:
- pre-questionnaire
- comprehension
- modification
and 2(b) for understanding regardless of the used measure (see Figures 2(a) is a difference between NoUM and UM for correctness of correctness of the modification task. On the other hand, there to the time to accomplish comprehension and modification significant difference between NoUM and UM with respect descriptive statistics and boxplots indicate that there is no deviation for the used metrics. The distributions of the values A. Descriptive Statistics and Exploratory Data Analysis

D. Conclusion Validity

INDICATE THAT THE CORRESPONDING NULL HYPOTHESIS WAS REJECTED

Hn3 Time 0.575 negligible (0.094) 2.417% 0.937

Hn2

Hn1 Time (+) 0.542 negligible (0.179) 4.006% 0.946

Hn0

IN BOLD

TABLE II

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Metric</th>
<th>p-value</th>
<th>Cohen/Cliff’s d</th>
<th>Perc. difference</th>
<th>Stat. Power / β-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hn0</td>
<td>$F_1$ (+)</td>
<td>&lt; 0.0001</td>
<td>large (1.317)</td>
<td>14.395%</td>
<td>0.925</td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.008</td>
<td>medium (0.428)</td>
<td>24.065%</td>
<td>0.835</td>
</tr>
<tr>
<td>Hn1</td>
<td>$F_1$ (+)</td>
<td>0.542</td>
<td>negligible (0.04)</td>
<td>1.00%</td>
<td>0.915</td>
</tr>
<tr>
<td>Hn2</td>
<td>$F_1$</td>
<td>0.256</td>
<td>negligible (0.011)</td>
<td>0.421%</td>
<td>0.955</td>
</tr>
<tr>
<td>Hn3</td>
<td>Time</td>
<td>0.575</td>
<td>negligible (0.094)</td>
<td>2.417%</td>
<td>0.937</td>
</tr>
</tbody>
</table>

We used a parametric test to verify null hypotheses only when the assumptions for its application fishing and the error rate. Worth mentioning that we involved 47 participants. Random heterogeneity of participants. In this section, we present the results of our data analysis have been rejected considering proper p-values. It is also concerns its ability to reveal a true pattern in the data. Low statistical power. In addition, the number of participants in our experiment was large enough. Concerning the fact that we did not impose any time limit concerns the reliability of treatment implementation. Reliability of treatment implementation. Hypotheses Testing

In Table I, we report the values of mean and standard deviations for the used metrics. The distributions of the values were verified. We used non-parametric tests otherwise. Statistical tests.

Fig. 2. Boxplots for correctness of understanding ((a) and (b)), understanding effort (c), correctness of modification (d), and understanding effort (e) group.

TABLE I

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Metric</th>
<th>p-value</th>
<th>Cohen/Cliff’s d</th>
<th>Perc. difference</th>
<th>Stat. Power / β-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hn0</td>
<td>$F_1$ (+)</td>
<td>&lt; 0.0001</td>
<td>large (1.317)</td>
<td>14.395%</td>
<td>0.925</td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.008</td>
<td>medium (0.428)</td>
<td>24.065%</td>
<td>0.835</td>
</tr>
<tr>
<td>Hn1</td>
<td>$F_1$ (+)</td>
<td>0.542</td>
<td>negligible (0.04)</td>
<td>1.00%</td>
<td>0.915</td>
</tr>
<tr>
<td>Hn2</td>
<td>$F_1$</td>
<td>0.256</td>
<td>negligible (0.011)</td>
<td>0.421%</td>
<td>0.955</td>
</tr>
<tr>
<td>Hn3</td>
<td>Time</td>
<td>0.575</td>
<td>negligible (0.094)</td>
<td>2.417%</td>
<td>0.937</td>
</tr>
</tbody>
</table>
A. Descriptive Statistics and Exploratory Data Analysis

In Table I, we report the values of mean and standard deviation for the used metrics. The distributions of the values were verified. We used non-parametric tests otherwise. The results for the Shapiro test suggested (p-value was 0.079 respectively). We applied parametric analyses when the assumptions for its application were met. Hypotheses only when the assumptions for its application were met. Statistical tests.

We used a parametric test to verify null hypotheses concerning the fact that we did not impose any time limit to perform the tasks.

B. Hypotheses Testing

The power of a statistical test concerns its ability to reveal a true pattern in the data. Low statistical power.

In Table II, we present the results of our data analysis with participants belonging to this sample. As for maintainability, a statistically significant difference was observed in favor of NoUM. As for correctness of understanding, a statistically significant difference was observed in favor of NoUM.

The distributions of the values were confirmed (p-value was 0.079 respectively). We applied parametric analyses when the assumptions for its application were met. Statistical tests.

In this section, we present the results of our data analysis. The effect size is large for correctness of understanding. That is, for correctness of understanding, a statistically significant difference was observed in favor of NoUM. As for correctness of modification, a statistically significant difference was observed in favor of NoUM.

C. Experimental Design

In this study, we manipulated the type of information presented to participants (NoUM and UM) and measured their performance on a maintainability task. The experimental hypotheses were stated as follows:

1. The time to accomplish comprehension and modification will differ between NoUM and UM.
2. The correctness of understanding will differ between NoUM and UM.
3. The correctness of modification will differ between NoUM and UM.

The hypotheses were tested using both descriptive and inferential statistical methods. The results are summarized in Table II. We rejected the null hypothesis that the corresponding null hypothesis was rejected.

D. Conclusion Validity

Statistical tests were performed to evaluate the validity of the experimental hypotheses. All hypotheses were tested using appropriate statistical tests.

The power of a statistical test concerns the fact that we did not impose any time limit to perform the tasks.

Low statistical power.

We used a parametric test to verify null hypotheses concerning the fact that we did not impose any time limit to perform the tasks.

Reliability of treatment implementation.

Random heterogeneity of participants.

Reliability of treatment implementation.

Low statistical power.

If the power is low, there is a high risk that an erroneous result will be obtained.

Validity of the experimental results.

Low statistical power.

We used a parametric test to verify null hypotheses concerning the fact that we did not impose any time limit to perform the tasks.

Reliability of treatment implementation.

Low statistical power.

We used a parametric test to verify null hypotheses concerning the fact that we did not impose any time limit to perform the tasks.

Reliability of treatment implementation.

Low statistical power.

We used a parametric test to verify null hypotheses concerning the fact that we did not impose any time limit to perform the tasks.

Reliability of treatment implementation.

Low statistical power.

We used a parametric test to verify null hypotheses concerning the fact that we did not impose any time limit to perform the tasks.
RQ1: Does the presence of unreachable methods penalize correctness of understanding software engineers are not familiar with source code?

RQ2: Does the presence of unreachable methods penalize the effort to comprehend source code if software engineers are not familiar with source code?
RQ3: Does the presence of unreachable methods penalize modifying source code if software engineers are familiar with source code?

RQ4: Does the presence of unreachable methods penalize the effort to maintain source code if software engineers are familiar with source code?
software engineering

“Dead code is code that isn’t executed”
-- R. C. Martin

dead instead of unreachable

percentage of unreachable methods ranges from 5% to 10%

are unreachable methods really harmful?

RQ1: Does the existence of unreachable methods penalize developers to maintain software engineers are not familiar with source code?

RQ2: Does the existence of unreachable methods penalizes software engineers to maintain source code if software engineers are not familiar with source code?

RQ3: Does the existence of unreachable methods penalize software engineers to maintain source code if software engineers are not familiar with source code?

RQ4: Does the existence of unreachable methods penalize software engineers to maintain source code if software engineers are not familiar with source code?

Impact of unreachable methods on comprehensibility and modifiability

giuseppe.scanniello@unibas.it