The effects of UML modeling on the quality of software
Nugroho, A.

Citation

Version: Corrected Publisher’s Version
License: Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from: https://hdl.handle.net/1887/16070

Note: To cite this publication please use the final published version (if applicable).
Appendix A

UML Survey Questionnaire

Generally, UML designs are the basis for an implementation of a system. For example, an implementation has the same classes and dependencies as are mentioned in the design. When an implementation closely resembles a design we say there is a high correspondence. Thus, maintaining correspondence is the effort to keep design and implementation identical.

**Question 1**
How would you rate the importance of correspondence between design and implementation?

- Not important
- Somewhat important
- Important
- Very important
- Extremely important

**Question 2**
What methods are used in your project to maintain the correspondence between UML designs and the implementation? Please choose all that apply.

- No special effort spent on maintaining correspondence
- Manually review and update the model
- Reverse engineer the implementation code
- Using round-trip engineering technique

**Question 3**
How frequently do you maintain the correspondence between UML designs and the implementation?
• No special effort spent on maintaining correspondence
• After final software release
• On monthly basis
• On weekly basis
• Continuously during coding activities

Question 4
How important is the correspondence between an implementation and the associated UML design in your projects?

• Unimportant
• Fairly important
• Important
• Very important
• Extremely important

Question 5
In implementing a UML design, please mark how strict you think the following statements should apply (Loosely - Somewhat Loosely - Neutral - Somewhat Strict - Strictly).

• The package structure in an implementation should correspond to the package structure in the design.
• The dependencies between classes in an implementation should correspond to the dependencies between classes in the design.
• The inheritance relations in an implementation should correspond to the inheritance relations in the design.
• The names of classes and methods in an implementation should correspond to names in the design.
• The order of method calls in an implementation should correspond to the order of messages in the design.

The purpose of a software design is to specify a system that is going to be developed. We regard a design as having a high degree of completeness if it specifies all parts of a system.

Question 6
Based on your experience, how would you (on average) rate the degree of completeness of the provided UML designs in describing the systems to be developed?

• Very Low
• Low
• Somewhat Low
• Somewhat High
• High

Question 7
In your experience, what is the frequency in which programmers go back to the designers to ask for clarification of the design in projects that use UML compared to projects that do not use UML?

• Very less often
• A bit less often
• Neutral
• Somewhat more often
• Much more often

Question 8
In case there is an inconsistency in a UML design (e.g. a class that is never called in any sequence diagram) that you have to implement, what would you do as a programmer?

• Implement the class as specified
• Discuss the inconsistency with other colleagues
• Infer the role of that class and make it functional to other classes
• Clarify the inconsistency to the designer
• Leave out the class

Question 9
Please indicate how frequent the following factors force you to deviate from a UML design in an implementation.

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>Sometimes</th>
<th>Often</th>
<th>Very Often</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting approaching deadline</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impractical design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incomplete design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design does not satisfy requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In modeling a system, designers can decide to specify different parts in various levels of detail. We say a part of a system as being modeled in high level of detail if all aspects (e.g. structures, behaviors, interactions) of that part are explicitly specified in the design.
Question 10
As a programmer, indicate how you would prefer that detail is used in UML models of a system that you have to implement (Disagree - Somewhat Disagree - Neutral - Somewhat Agree - Agree).

- All parts of a system should be specified in an equal amount of detail.
- Different parts of a system should be specified in a level of detail that is proportional to the complexity of the parts being modeled.
- Parts that are more critical for the functioning of the system should be specified in more detail.
- A model should explain how the system works, but allow programmers freedom to determine implementation details.

Question 11
Does the use of UML help or hinder you in being more productive in different activities of software development? Please indicate by marking a circle in each row of the following table.

(1 = Very Hinders; 2 = Hinders; 3 = Somewhat hinders; 4 = Neutral; 5 = Somewhat helpful; 6 = Helpful; 7 = Very Helpful)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis/Problem Understanding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Imperfections in UML designs may appear in various forms. Inconsistency reveals contradicting information in portraying a design concern. Understandability relates to the degree in which concepts are easily inferred from a design. Inaccuracy relates to the lack of preciseness in specifying design concerns. Finally, incompleteness refers to a design’s low coverage in specifying all parts of a system.

Question 12
How often do the imperfections in UML models listed below lead to problems in implementing a design?

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>Sometimes</th>
<th>Often</th>
<th>Very Often</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inconsistency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understandability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inaccuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incompleteness</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question 13
Please indicate how the use of UML influences the following quality properties of the final software product.

<table>
<thead>
<tr>
<th></th>
<th>Reduce</th>
<th>Somewhat Reduce</th>
<th>Neutral</th>
<th>Somewhat Improve</th>
<th>Improve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfying / covering Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correctness (number of defects)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modularity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understandability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Your professional background is very useful for us in interpreting your answers accurately. Therefore, please continue with the following questions that concern your professional background.

**Question 14**
What is the highest education degree you received?

- None
- B.Sc / HBO (Dutch)
- M.Sc. / Drs. (Dutch)
- Ph.D. / Dr. (Dutch)

**Question 15**
Please indicate the number of year you have been working as a programmer.

- Less than 2 years
- 2 – 5 years
- More than 5 years

**Question 16**
Please indicate the number of software project in which you were involved as a programmer over the past 10 years.

- Less than 4 projects
- 4 – 6
- 7 - 10 projects
- More than 10 projects
**Question 17**
Choose the programming languages in which you have more than 2 years of experience.

- C
- C#
- C++
- Java
- Basic
- Cobol
- Other, __________________

**Question 18**
Indicate the location where you are currently working as a programmer.

- Netherlands
- India
- United Kingdom
- United States
- Other, __________________

**Question 19**
How have you learnt UML?

- I am not familiar with UML
- From course(s) in industry
- From course(s) at university
- Self-study from books

**Question 20**
What percentage of software projects that you were involved in over the past 10 years used UML?

- Less than 25 % of the projects
- 25 % of the projects
- 50 % of the projects
- 75 % of the projects
- More than 75 % of the projects
Appendix B

Model Comprehension Questionnaire

Q1) According to the diagram, there are two categories of item, i.e., ReferenceItem and LoanItem. Which of the following statements is true about those two items?
   a. ReferenceItem is not allowed to be borrowed
   b. LoanItem is not allowed to be borrowed
   c. Both ReferenceItem and LoanItem are allowed to be borrowed
   d. Both ReferenceItem and LoanItem are not allowed to be borrowed
   e. Cannot determine based on the model

Q2) In the process of lending a book, the system must first check the type of the requested book item. What information is required as an input to perform this validation?
   a. isbn
   b. itemId
   c. userId
   d. title
   e. Cannot determine based on the model

Q3) According to the diagram, a loan of a book cannot be renewed (extended) when other borrower has reserved it. Which of the following classes hold the information about the number of reservations of a book title?
   a. Title
   b. Reservation
   c. Loan
   d. LoanItem
   e. Cannot determine based on the model

Q4) When adding an item to the system, object item is created. Besides ItemControl, the creation of object item implies its association with which of the following class objects? a. MonitorControl
Q5) One of the conditions that has to be satisfied before removing a user from the system is that the user must not have any outstanding fine/charge. From which of the following classes can object UserControl obtains this information?
   a. Title
   b. LoanControl
   c. Loan
   d. User
   e. Cannot determine based on the model

Q6) What is the main functionality of class UserTerminal?
   a. It provides user interface functionality to class User
   b. It provides user interface functionality to class Title
   c. It provides user interface functionality to class Loan
   d. It provides user interface functionality that is common to all system users
   e. Cannot determine based on the model

Q7) Which of the following describes the functionality of the controller classes (e.g., TitleControl, UserControl, etc.)?
   a. Providing logic for all system functionality
   b. Providing logic for database functionality
   c. Providing logic for interacting with system users
   d. Configuring or managing for the user interface functionality
   e. Cannot determine based on the model

Q8) When an item of a book title is borrowed or returned, the system has to immediately update the status of this book item. Which of the following classes holds this information (i.e., status)?
   a. Title
   b. Loan
   c. LoanItem
   d. Item
   e. Cannot determine based on the model

Q9) When a borrower returns a book, the system has to check whether it is returned before the due date. Which of the following classes holds the due date information of a borrowed book?
   a. Title
   b. User
   c. Loan
   d. LoanItem
   e. Cannot determine based on the model
Q10) A borrower will be charged when returning books later than their due dates. Which of the following classes calculates the amount of charge for a particular loan?
   a. Title  
   b. User  
   c. Loan  
   d. LoanItem  
   e. Cannot determine based on the model

Q11) In order to make a reservation of a book title in the system, which of the following classes must instantiate a Reservation object?
   a. Title  
   b. ReservationControl  
   c. BorrowerTerminal  
   d. User  
   e. Cannot determine based on the model

Q12) According to the diagram, which of the following statement is TRUE with regard to scenario Remove Item?
   a. The status of item must be first assessed in class Item  
   b. The status of item must be first assessed in class LoanItem  
   c. An item can be deleted regardless of its status  
   d. An item can be deleted only if the corresponding title is deleted  
   e. Cannot determine based on the model

Q13) According to the diagram, which of the following statement is TRUE with regard to scenario Remove Item?
   a. The status of item must be first assessed in class Item  
   b. The status of item must be first assessed in class LoanItem  
   c. An item can be deleted regardless of its status  
   d. An item can be deleted only if the corresponding title is deleted  
   e. Cannot determine based on the model

Q14) Class LibrarianTerminal implements method removeTitle. According to the diagram, which of the following pseudo code represents the implementation of method removeTitle?
   a. 
   ```java
   Class LibrarianTerminal {
       TitleControl a;
       ...
       Function removeTitle(isbn) {
           a.removeTitle(isbn);
       }
   }
   ```
   b. 
   ```java
   Class LibrarianTerminal {
   ```
TitleControl a;

...  
Function removeTitle(isbn) {
    a.removeItem(isbn);
    a.removeTitle(isbn);
}

...  
}

c. Class LibrarianTerminal {
    ItemControl a;
    TitleControl b;
    ...
    Function removeTitle(isbn) {
        create a;
        a.removeItem(itemId);
        b.removeTitle(isbn);
    }
    ...
}

d. Class LibrarianTerminal {
    TitleControl a;
    ItemControl b;
    ...
    Function removeTitle(isbn) {
        create b;
        a.removeTitle(isbn);
        b.removeItem(isbn);
    }
    ...
}

e. Cannot determine based on the model

Q15) According to the diagram, which of the following indicates the correct implementation of making a book reservation (instantiating a reservation object)?

a. Class ReservationControl {
    Reservation a;
    User b;
    ...
    Function requestMakeReservation(isbn) {
        create a;

b. reservationAdded();
}
...
}

b. Class Title {
  Reservation a;
  ...
  Function requestMakeReservation(isbn) {
      create a;
      ...
  }
  ...
}

c. Class ReservationControl {
  Reservation a;
  ...
  Function requestMakeReservation(isbn) {
      create a;
  }
  ...
}

d. Class BorrowerTerminal {
  Reservation a;
  ...
  Function requestMakeReservation(isbn) {
      create a;
  }
  ...
}

e. Cannot determine based on the model
Appendix C

Descriptive Statistics of the NS-OFI Data Set

Table C.1 and C.2 provides descriptive statistics of Java classes that are modeled in class diagrams and sequence diagram respectively, when no sampling method is performed: defect types are not taken into account.

Table C.1: Descriptive statistics of classes modeled in class diagrams – NS-OFI data set

<table>
<thead>
<tr>
<th>Measures</th>
<th>N</th>
<th>Med</th>
<th>Mean</th>
<th>SDev</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD_{aop}</td>
<td>48</td>
<td>0.00</td>
<td>0.32</td>
<td>0.44</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>CD_{asc}</td>
<td>48</td>
<td>0.16</td>
<td>0.44</td>
<td>0.47</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Coupling</td>
<td>48</td>
<td>15.00</td>
<td>18.89</td>
<td>18.22</td>
<td>2.00</td>
<td>119.00</td>
</tr>
<tr>
<td>Complexity</td>
<td>48</td>
<td>14.00</td>
<td>38.20</td>
<td>56.34</td>
<td>0.00</td>
<td>297.00</td>
</tr>
<tr>
<td>Size (KSLoC)</td>
<td>48</td>
<td>0.24</td>
<td>0.49</td>
<td>0.80</td>
<td>0.02</td>
<td>5.26</td>
</tr>
<tr>
<td>Defect Density</td>
<td>48</td>
<td>12.47</td>
<td>19.06</td>
<td>14.88</td>
<td>2.67</td>
<td>58.82</td>
</tr>
</tbody>
</table>

Table C.2: Descriptive statistics of classes modeled in sequence diagrams – NS-OFI data set

<table>
<thead>
<tr>
<th>Measures</th>
<th>N</th>
<th>Med</th>
<th>Mean</th>
<th>SDev</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD_{obj}</td>
<td>61</td>
<td>2.00</td>
<td>1.97</td>
<td>0.05</td>
<td>1.66</td>
<td>2.00</td>
</tr>
<tr>
<td>SD_{msg}</td>
<td>61</td>
<td>1.88</td>
<td>1.72</td>
<td>0.57</td>
<td>0.30</td>
<td>2.56</td>
</tr>
<tr>
<td>Coupling</td>
<td>61</td>
<td>13.00</td>
<td>17.96</td>
<td>17.51</td>
<td>5.00</td>
<td>119.00</td>
</tr>
<tr>
<td>Complexity</td>
<td>61</td>
<td>24.00</td>
<td>46.24</td>
<td>66.57</td>
<td>0.00</td>
<td>366.00</td>
</tr>
<tr>
<td>Size (KSLoC)</td>
<td>61</td>
<td>0.26</td>
<td>0.56</td>
<td>1.10</td>
<td>0.02</td>
<td>7.01</td>
</tr>
<tr>
<td>Defect Density</td>
<td>61</td>
<td>12.35</td>
<td>26.09</td>
<td>39.99</td>
<td>1.73</td>
<td>217.39</td>
</tr>
</tbody>
</table>
Bibliography


