Modeling Attributes Influencing Intelligent Information Software System using Exploratory Factor Analysis

Manu Banga, Abhay Bansal, Archana Singh

Abstract: In today’s era Software development is hybrid with Software Reliability Model, using model parameters for validating software faulty dataset as it has many pitfalls choosing reliability model being dataset biasing, parameters on which choosing parameter estimation relevant to model based on statistical test, readiness index, functionality preserving, effort consumed on faulty datasets prediction. Model Estimation Modeling based on Least Square Error (LSE) for which factor analyzed using Exploratory Factor Analysis and Confirmatory Factor Analysis and model is validated using p-test with confidence interval of 95% on cloud service. Based on our eleven attribute of our dataset, Continuous Integration is most relevant among other factors. Defect Prediction is the most crucial and critical task in successful operation of software working as can leads to faults which further causes failures and these failures are very ominous, thus reducing and preventing software defects is major challenging work.

Keywords—Software Reliability Models, Cloud Service, EFA, CFA, Intelligent Information System.

I. INTRODUCTION

Software Reliability Growth Models in software development life cycle of debugging failures intensity using testing history for predicting behavior of software in real time environment. Firstly, selection of model, which has eigen-values calculated in a performance matrix, lastly parameters are estimated using LSE based on EFA and CFA. Collectively, these are four steps are known as Model Composition[14,15]

![Model Composition](image)

Among all the software reliability models Non-Homogenous Poisson process based on S-shaped, concave curve software-debugging phenomenon. Based on software failure phenomenon causes monetary losses, time loss, as faults transform to failures [1, 2]. Thus controlling faults at appropriately and careful examine them in a testing/debugging phase by using previous data of software failure and for estimating defects remained under testing phase. Based on Failure History, Optimal handling of the defects mean function m(t), and software intensity function λ(t) of software reliability models is accessed. Among all the available models, Non-Homogenous Poisson Process Model has been employed to estimating &predicting remaining number of defects in a software system under test [4, 9]. Software Reliability Models plays a important role in decision making by top executives for accessing various software reliability engineering. models based on practical software reliability engineering, NHPP SRGMs have been proved successful and robust tool for predicting, controlling, and assessing software reliability.

II. LITERATURE REVIEW:

Preventing failure of software under specified conditions and time. Among several models, Non Homogeneous Poisson Process, 11 attribute are taken into account: Test Driven, Continuous Integration, Coding Environment, Feedback available from customer, SRS documents or documents available for distribution of workload, testing tools using testing in which customer checks the software working in real environment, testing data for 2 point validation, interaction with customer [7, 8]. So in real world scenario model based on Non-Homogenous Poisson Process proved accurate for software reliability prediction [5, 6].

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T. D(Test Driven)</td>
<td>Code Refactoring</td>
</tr>
<tr>
<td>C.I(Continuous Integration)</td>
<td>Integration</td>
</tr>
<tr>
<td>Coding Environment</td>
<td>Defects Detection</td>
</tr>
<tr>
<td>Feedback</td>
<td>Iteratively Detection at each stage</td>
</tr>
<tr>
<td>Documents</td>
<td>For Distribution of workload</td>
</tr>
<tr>
<td>Testing Tools</td>
<td>Beta Testing,</td>
</tr>
<tr>
<td>Testing Data</td>
<td>For Validation Purpose</td>
</tr>
<tr>
<td>Interaction</td>
<td>Periodic Meeting</td>
</tr>
</tbody>
</table>

Revised Manuscript Received on June 14, 2019
Manu Banga, ASET, Amity University, Noida, India.
Abhay Bansal, ASET, Amity University, Noida, India.
Archana Singh, ASET, Amity University, Noida, India.

Retrieval Number: H6881068819/190BEIESP

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication
Failure of software is very ominous so in terms of mathematical modeling depending on input factors. **Factor Analysis:** As per Kaiser Criteria, Eigen values or characteristic roots are criteria for determining a factor. If eigenvalues greater than one then that is consider as a factor otherwise not a factor [3].

**Exploratory Factor Analysis:** It is method based on no prior theory and factors or variable can be associated with any factor [1].

**Confirmatory Factor Analysis:** It is a method based on pre established theory, assuming each Factor associated with specified sets of measured variables. It is used of evaluating evidences [2].

<table>
<thead>
<tr>
<th>Test Cases</th>
<th>Checking Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing Method</td>
<td>Iterative Check</td>
</tr>
<tr>
<td>Real Environment</td>
<td>At real time</td>
</tr>
</tbody>
</table>

**Table 1: Attribute affecting testing of Model**

Reliability Models on software development phase.

**Model Parameter Estimation:**

Based on Stochastic Modeling, parameter estimation technique

\[ a = \text{Expected Number of Failure,} \]

\[ b = \text{Faults Detection Rate} \]

\[ m(t) = a(1 - (1 + bt)e^{-bt}), \text{where} \ a > 0, b > 0 \]

represented concave curve,

\[ m(t) = a(1 - e^{-bt}), \text{where} \ a > 0, b > 0 \]

\[ m(t) = N \left( \frac{\beta}{\beta + (c/b) \ln((a + e^{bt})/(1 + a))} \right)^a \]

Non- Homogenous Poisson Distribution:-

**Laplace Trend Test:** By this software reliability growth is predicted. It is based on centroid and after its comparison scope is calculated and if its secure is greater than zero than it is a upward slope and if its secure is less than zero then it is a downward slope and slope of zero means horizontal line [12].

\[ \text{Mean Square Error} = \frac{\sum_{i=1}^{N} (m(x_i) - (m(t_x))^2)}{t-p} \]

**Signal Noise Ration (R_s) = -10 \log(\sum_{x=1}^{N} y_{x+1})/N_{x+1},**

where,

\[ R_s = \text{number of experiments trails,} \]

\[ x_n = \text{Number of times trail sequence,} \]

\[ N_{x+1} = \text{Random Times occurred in sequence} \]

\[ y_{x+1} = \text{Number of generations required for require results.} \]

1. **Generalised Goel NHPP Model:**

Mean Value Function:

\[ m(t) = a(1 - e^{-bt})a > 0, b > 0, c > 0 \]

Software Failure Intensity \( \lambda(t) = abct^{-c}e^{-bt} \)

2. **Goel Okumoto Model:**

Mean Value Function: \( m(t) = a(1 - e^{-bt})a > 0, b > 0 \)

Software Failure Intensity \( \lambda(t) = abe^{-bt} \)

3. **Gompertz Model:**

Mean Value Function: \( m(t) = ake^{-bt} \)

Software Failure Intensity \( \lambda(t) = ab \ln(k)e^{-bt}e^{-bt} \)

4. **Inflection S-Shaped Model:**

Mean Value of function \( m(t) = \frac{a(1-e^{-bt})}{1+ke^{-bt}} \)

Software Failure Intensity \( \lambda(t) = \frac{abke^{-bt}(1+bt)}{(1+ke^{-bt})^2} \)

5. **Logistic Growth Curve Model:**

Mean Value Function:

\[ m(t) = \frac{a}{1 + k e^{-bt}} \]

Software Failure Intensity \( \lambda(t) = \frac{abke^{-bt}}{(1+ke^{-bt})^2} \)

6. **Modified Duane Model:**

Mean Value Function

\[ m(t) = a[1 - (b/(b + t))^c]a > 0, b > 0, c > 0 \]

Software Failure Intensity \( \lambda(t) = acb(b + t)^{1-c} \)

7. **Musa-Okumoto Model:**

Mean Value Function \( m(t) = \alpha \ln(1 + \beta t) \)

Software Failure Intensity \( \lambda(t) = ab/(1 + bt) \)

8. **Yamada Imperfect Debugging Model:**

Mean Value of function

\[ m(t) = \frac{ab(e^{at} - e^{-bt})}{a + b} \]

Software Failure Intensity
Yamada Rayleigh Model:
Mean Value of function
\[ m(t) = \frac{\alpha (1 - e^{-bt}) (1 - \frac{\alpha}{b}) + \alpha at}{1 + \beta e^{-bt}} \]
Software Failure Intensity
\[ \lambda(t) = \frac{\alpha (1 - e^{-bt})(1 - \frac{\alpha}{b}) + \alpha at}{1 + \beta e^{-bt}} \]

10 Delayed S-Shaped Model:
Mean Value of function
\[ m(t) = a \left( 1 - (1 + bt)e^{-bt} \right) \]
Software Failure Intensity
\[ \lambda(t) = ab^2 e^{-bt} \]

11 Yamada Imperfect Debugging Model 2:
Mean Value of function
\[ m(t) = a \left( 1 - e^{-bt} \right) \left( 1 - \frac{\alpha}{b} \right) + \alpha at \]
Software Failure Intensity
\[ \lambda(t) = ab^2 e^{-bt} \left( 1 - \frac{\alpha}{b} \right) + \alpha a \]

12 Yamada Exponential:
Mean Value of function
\[ m(t) = \alpha \left( 1 - e^{-\alpha t(1 - e^{-bt})} \right) \]
Software Failure Intensity
\[ \lambda(t) = \alpha e^{-\alpha(1 - e^{-bt})} \]

13 Pham-Nordmann-Zhang Model (PNZ model):
Mean Value of function
\[ m(t) = \frac{a(1 - e^{-bt})(1 - \frac{a}{b}) + at}{1 + \beta e^{-bt}} \]
Software Failure Intensity
\[ \lambda(t) = \frac{ab^2 e^{-bt}(1 - \frac{a}{b}) + \alpha ab^2 e^{-bt}(1 - \frac{a}{b}) + a}{1 + \beta e^{-bt^2}} \]

14 Pham-Zhang NHPP (PZ Model):
Mean Value of function
\[ m(t) = \frac{1}{(1 + \beta e^{-bt}) \left( c + a \right)(1 + e^{-bt}) - \frac{ab}{b - a} \left( e^{-at} - e^{-bt} \right)} \]
Software Failure Intensity:
\[ \lambda(t) = \frac{b(c + a)(1 + \beta)e^{-bt} - \left[ be^{-bt}(1 + \beta e^{-bt}) - ae^{-at}(1 + \beta e^{-bt}) \right]}{(1 + \beta e^{-bt})} \]

15 Pham-Zhang Imperfect Fault Detection (Pham-Zhang IFD model):
Mean Value of function
\[ m(t) = \alpha - ae^{-\beta t} \left( 1 + (\beta + \gamma) \tau + \beta d \tau^2 \right) \]
Software Failure Intensity:
\[ \lambda(t) = ae^{-\beta t}(1 + (\beta + \gamma) \tau + \beta d \tau^2) \]

16 Zhang-Teng-Pham Model:
Mean Value of function
\[ m(t) = \frac{\alpha}{\rho - \beta} \left[ 1 - \left( 1 + e^{-bt} \right) \left( 1 + ae^{-\beta t} \right)^{\frac{1}{\beta - \rho}} \right] \]
Software Failure Intensity
\[ \lambda(t) = \frac{\alpha c}{1 + e^{-bt}} \left( 1 + ae^{-\beta t} \right)^{\frac{1}{\beta - \rho}} \]

III. PROPOSED APPROACH:
In this paper, a new algorithm is proposed in which we are doing factor analysis on 11 attribute on standard datasets available over cloud relevant. Based on estimation model, Matrix was made on the variance vales, standard deviation was calculated of the dataset.

Membership Function
\[ F(x) = \frac{1}{1 + e^{-w_2z_1^{2w_1}}} \]
Fuzzy Set Theory is based on dataset defined faults caused by them may lead to software failure To determine intensity of failure on Non- between (0... ... 1). Defects are assigned to the priority basis as some defects are very as important in causing failures.

Reliability of Cloud Based Modeling: Discrete Fourier Transform: In this modeling 1/N parameter factor is calculated based on the value of repeat occurrence of faults or frequency for faults, from previous dataset its occurrence could be validated by I

\[ F_{X<Y}(t) = Pr(X<Y)(T \leq t) \]
\[ = \int_{t}^{t-\tau_1} \int_{0}^{\frac{\tau_2}{\tau_1}} \left( y_{1,2} e^{-\gamma_1 r_1} \right) \left( y_{1,2} e^{-\gamma_2 r_2} \right) d\tau_2 d\tau_1 \]
\[ y_{i,2}^{-I} = f(\sum_{j=1}^{N_{i,2}} w_{i,j} \cdot y_{i,2}^{-1} + \theta_{i,j}) \]
\[ i = 1, ..., N_{i,2}, l = 1, ..., L \]

Homogenous Poisson Process Model we apply our novel approach:
1. Preprocessing the Dataset using Factor Analysis based on faulty dataset.
2. Applying Exploratory Factor Analysis on the faulty dataset, parameters are investigated
3. After step 2, Confirmatory Factor Analysis was applied on the dataset, validating factors relevant to estimating parameter relevant to our model
4. New Variance, $\sigma_i$ again calculate of the dataset and $\sigma_i$ compared with $\sigma_{i+1}$, based on this Matrix is based.

Figure 2: Modeling Attribute using Exploratory Factor Analysis

Table 2: Attribute with Eigen Values

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Eigen-values</th>
<th>% of Variance</th>
<th>Total</th>
<th>Attribute</th>
<th>Eigen-values</th>
<th>% of Variance</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Driven Development</td>
<td>4.348</td>
<td>25.175</td>
<td>25.175</td>
<td>Requirement</td>
<td>0.414</td>
<td>2.585</td>
<td>89.16</td>
</tr>
<tr>
<td>Continuous Integration</td>
<td>3.365</td>
<td>21.034</td>
<td>45.209</td>
<td>Test Cases</td>
<td>0.366</td>
<td>2.288</td>
<td>91.448</td>
</tr>
<tr>
<td>Test Environment</td>
<td>2.391</td>
<td>11.945</td>
<td>60.154</td>
<td>Method Used</td>
<td>0.334</td>
<td>2.087</td>
<td>93.535</td>
</tr>
<tr>
<td>Feedback</td>
<td>1.41</td>
<td>11.811</td>
<td>71.965</td>
<td>Customer</td>
<td>0.284</td>
<td>1.776</td>
<td>95.311</td>
</tr>
<tr>
<td>Documentation</td>
<td>0.688</td>
<td>3.301</td>
<td>75.266</td>
<td>Defects</td>
<td>0.278</td>
<td>1.735</td>
<td>97.045</td>
</tr>
<tr>
<td>Tools</td>
<td>0.635</td>
<td>3.969</td>
<td>80.235</td>
<td>Realistic</td>
<td>0.215</td>
<td>1.343</td>
<td>98.388</td>
</tr>
<tr>
<td>Data</td>
<td>0.547</td>
<td>3.421</td>
<td>83.656</td>
<td>Testing</td>
<td>0.167</td>
<td>1.043</td>
<td>99.431</td>
</tr>
<tr>
<td>Meetings</td>
<td>0.467</td>
<td>2.919</td>
<td>86.565</td>
<td>Test Driven Development</td>
<td>0.091</td>
<td>0.569</td>
<td>100</td>
</tr>
</tbody>
</table>
IV. CONCLUSION:

In this paper, a new technical analysis was done on Exploratory Factor Analysis and then applying Confirmatory Factor Analysis for Model Estimation mathematical analysis based on factor selection relevant among the given eleven factors based on Readiness Index values, Parameter Dependency.

REFERENCES:


AUTHORS PROFILE

Manu Banga, Ph.D. Scholar (CSE), ASET, Amity University Noida. He has done M. Tech and B.Tech and published various papers in conference and journals.

Dr. Abhay Bansal, Jt. Head ASET / Director, DICET, HOI. ASET, Amity University, Noida. His area of research is Data mining, big data, and software engineering. He guided more than 15 Ph.D. scholar. He is life member of ACM, IEEE, CSI & IET.

Dr. Archana Singh, Ph.D. (CSE), M.Tech (CSE). She is working as Professor at ASET, Amity University, Noida. Her area of research is Data mining, big data, software engineering. Presently she is Ph.D. guide of 8 students. She is life member of ACM, IEEE, CSI & IET.