Optimal parameter choice in modeling of ERP system reliability

Frane Urem, Želimir Mikulić
Department of management
College of Šibenik
Trg Andrije Hebranga 1, 2200 Šibenik, Croatia
zelimir.mikulic@vus.hr,
frane.urem@compco.hr

Krešimir Fertalj
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, Zagreb, 10000 Croatia
kresimir.fertalj@foi.hr

Abstract. *Enterprise Resource Planning (ERP) system is a complex software system that supports network of different business processes. Sales of such systems has expanded in the late 90s. In that period ERP installations dominated in large business organizations, but today are commonplace in small and medium-sized enterprises. It's very important for ERP buyers to check is it possible to adapt preferred ERP package to their business. ERP implementer can modify the software to fit the customer's business process but that can slow down the project and introduce new defects into the system. It would be useful to gather input about the perceived defects during systems usage and to try to assess what a customer can expect in the future. We assumed that suitable models in modeling of ERP system reliability are Software Reliability Growth Models (SRGM). This paper presents how optimal selection of input parameters can be done during a system usage in an attempt to choose the appropriate SGRM model. In presented case study Weibull model is identified and used for reliability modeling and reliability prediction for existing ERP installation.*

Keywords. ERP, reliability prediction, Weibull model

1 Introduction

ERP (Enterprise Resource Planning) system integrates all departments and functions across a company onto a single computer system that can serve all those different department's particular needs [8]. According to [9] the most common reason that companies walk away from ERP projects is that they discover the software does not support one of their important business processes. If they gave up later the project costs a lot more than it should. The problems in ERP implementation can be different but it would be ideal to discover them at the time of the evaluation and implementation of software. The more serious problems may arise later in ERP maintenance. In both cases, the challenge is to make an assessment of the ERP system reliability. That includes two types of activities: reliability estimation and reliability prediction. Reliability estimation represents the measure of achieved ERP reliability and reliability prediction determines future ERP reliability upon available measures.

In this paper we start from the client organization's perspective with assumption that the ERP system is not completed software in early design stage. It is usually a product that has already been delivered to many customers in the past but some adaptation and configuration must be done for specific client. The main problem is how to select an appropriate software reliability model with optimal parameter choice. In section 2 basic definitions are reviewed for parameters that affect the reliability of software and define what types of defects are reported in observed ERP system. All defects are tracked and categorized by proposed scheme from users and vendor in the separate e-service database. Measurement-based analysis framework is presented in section 3, as a good starting point for ERP reliability estimation and prediction. In section 6, all phases from specified framework are validated on presented use case.

2 Basic definitions

The key terms of software reliability are *error*, *fault* and *failure*.

Error has two different meanings [1]:

- A discrepancy between a computed observed or measured value or condition and theoretically correct value or condition.
- A human action that results in software containing a fault.

Fault is the cause of the failure [1]. It is also referred as a *bug*.

Failure occurs when the user perceives that the software ceases to deliver the expected service [1].

Defect can be used as generic parameter in modeling of ERP system reliability to refer to either a fault (cause) or a failure (effect). From the perspective of strict definition in ERP software it often captures the fault, sometimes the error and often the failure.

Software reliability is defined as the probability of failure-free software operation in a specified environment for a specified period of time [1]. The same definition can be used for ERP system reliability but the term defect can be used instead of failure.

All we need is a scheme to capture the semantics of each ERP software defect quickly. It is the definition and capture of defect attributes that make mathematical analysis and modeling possible.

3 How to track ERP defects ?

Typical defect triggers in ERP systems [2] are:

- Training
- Customization
- Integration
- Data conversion

Defect types in ERP systems are usually categorized in a simple scheme [2] as:

- Insufficient understanding and familiarity with the system (user-support)
- Changes to security and authorization profiles
- Bugs fixes
- Changes to existing functionality
- New functionality

Each defect has also attributes like opening date, priority, reason for priority, description and closing date.

Each defect has its own life cycle:

- Opening by customer
- Analysis
- Development
- Testing
- Implementation
- Closing by customer

The instantaneous rate of defects which is denoted by $\lambda(\tau)$, is an important function in reliability measurement.

The mean value of defect distribution (the expected number of defects in the time interval [0,t]) is:

$$E[N(t)] = \mu(t) = \int_0^t \lambda(\tau)\mathrm{d}\tau \qquad (1)$$

where $\mu(t)$ is the expected number of defects at time *t*.

There are two fundamentally different periods in modeling of ERP system reliability:

- Implementation (initial software design) with activities like existing data conversion, customization and integration.
- Post-implementation (post-production) with activities connected with product maintenance.

If there is a separate system with defect log database then is possible to extract data, perform analysis and choose an appropriate ERP reliability model [1] for both periods like on Figure 1.

In step 1 necessary information from the defect log is extracted (cumulative number of defects, rate of defects). In step 2 the data are interpreted with *reliability estimation* of operational ERP system in actual environments and issues that must be addressed to improve system reliability. In step 3 and 4 the appropriate model is identified for *reliability prediction* (what-if analysis) that follows in step 5.

4 Suitable ERP reliability models

Most authors are recommending software reliability models according to Software Development Life Cycle (SDLC) phases [3]. One group of models is known as Software Reliability Growth Models (SRGM). This type of models captures failure behavior of software during testing, verification and validation and extrapolates it to determine its behavior during operation. Hence this category of models uses failure data information and trends observed in the failure data to derive reliability predictions.

We have found SRGM interesting for ERP reliability modeling during implementation to assess whether a system is ready for production [4]. We have also found this group of models interesting in ERP production assuming that system will be constantly changed during future maintenance.

Most of SRGM are described in [1]. Model classification scheme was proposed by Musa and Okumoto [1]. We have found particularly interesting classification in that scheme (based on total number of failures that can be experienced in infinite time) that can be applied in ERP reliability modeling on *finite* and *infinite* models. Finite models are applicable before ERP production [4] because in that period we strive for a finite number of defects and it is interesting to use them for estimating the number of undetected defects in a moment when ERP production starts. Infinite models are applicable in ERP post-production and it is interesting to use them for ERP maintenance validation.

The main approach in ERP reliability model selection can be defined by the distribution of the number of the defects experienced by time. During step 1 and step 2 from Figure 1. defect distribution experienced by time can be measured. According to [1] two important distribution types can be expected: Poisson and binomial. When the corresponding distribution is obtained, appropriate models can be selected in step 3 and step 4 and compared with real system in step 5 from Figure 1.

In presented case study, post-implementation maintenance log for one ERP system was available. After data analysis it was found that probability density function for new defects can be interpreted with Weibull distribution. Weibull distribution can be employed for engineering analysis with small sample sizes better than any other method so that was another reason to accept Weibull reliability model from [1].

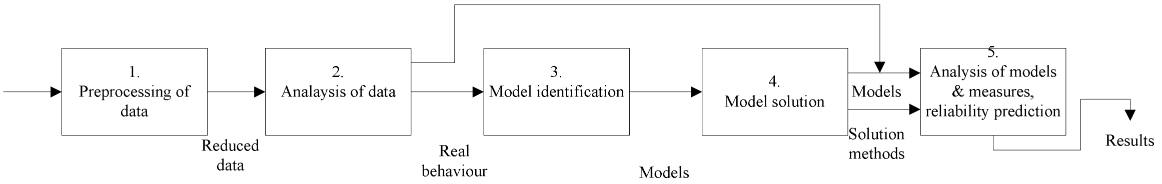

Figure 1. Measurement-based analysis framework of ERP system reliability

5 Case study

As a case study we have selected one Croatian rental company witch implemented RentPRO XL ERP system that handles all the aspects of rental management. RentPRO XL is developed by CarPro Systems. More about RentPRO XL ERP can be found at [6]. The final goal was to validate optimal parameter choice in selection of an appropriate model of ERP system reliability during production. The main source of data was an e-service database of all requests for system enhancement, bug fixes and requests pertaining to customer support. All defects were added by the user during product usage and solved by CarPro customer support centre in Pune, India.

Every defect had attributes like: event date, subject, priority, reason for priority, domain (Insufficient understanding, Bug, Changes to existing functionality, New functionality, Changes to security and authorization profiles) and attached documents.

Insufficient understanding defect in presented case study is defect that is initially reported as a bug, but after analysis it is found that customer did not use the software correctly. In that case additional documentation or customer training is provided.

Every defect can be tracked with more events (like question in by user, answer out by customer support and closing by customer). Customer support can change defect domain (for example: change from bug to insufficient understanding). All defects were solved according to their complexity (defects recognized as insufficient understanding with more training or documentation) and urgency (regular monthly upgrades for all customers or urgent repair for specific customer). All new functionalities or changes to existing functionality were published in regular monthly updates.

6 Data analysis and interpretation

In **step 1**, we have extracted data from the existing maintenance log (E-service log) in one table with events like: defect creation date, defect closing date, defect type.

In **step 2**, we have prepared data with the structure from Table 1.

We have chosen monthly time distribution because we have assumed that most important cause of new defects are regular monthly updates.

In **step 3**, we have analyzed measured defect PDF from the Table 1. in period of 24 months with EasyFit software [7] and found good fit with Weibull distribution (visible at Figure 2).

Weibull reliability model is binomial model type [1] and defect rate function is obtained from the defect probability density function (PDF) $f_a(t)$ as:

$$\lambda(t) = \mathrm{N}f_a(t) \qquad (2)$$

where the N is the expected number of defects in infinite time *t.*

According to (1), expected number of defects $\mu(t)$ at time t is calculated from the defect rate function $\lambda(t)$.

For the binomial model types, according to [1], expected number of defects $\mu(t)$ is in turn related to the defect cumulative distribution function (CDF) $F_{a(t)}$ as:

$$\mu(t) = \mathrm{N}F_{a(t)} \qquad (3)$$

Table 1. Formatted structure for extracted data

No.	Attribute	Description
1	Month	Month of system usage
2	Number of new defects	Number of new defects, detected in month of system usage, extracted from E-service log
3	Probability density function (PDF) for the new defects	Statistical distribution of new defects, detected during ERP usage. Calculated from [2].
4	Cumulative number of defects	Cumulative number of defect, calculated from [2]

The equation for the Weibull probability density function is:

$$f_a(t) = \frac{\alpha}{\beta^{\alpha}} t^{\alpha-1} e^{-(\frac{t}{\beta})^{\alpha}} \qquad (4)$$

The equation for the Weibull cumulative distribution function is:

$$F_a(t) = 1 - e^{-(\frac{t}{\beta})^{\alpha}} \qquad (5)$$

Defect PDF and CDF function can be calculated from measured data from Table 1. and compared with calculated Weibull distribution values with estimated parameters α and β.

In **step 4**, we have found estimated parameters for the Weibull distribution with EasyFit software [7] as: α= 1,3842 and β= 8,4355.

Goodness of fit with Kolmogorov-Smirnov test is presented for estimated Weibull distribution in Table 2

With estimated α and β, Weibull distribution is calculated with (4) and (5). Result is presented at Figure 2.

According to (3) it is possible to compare cumulative number of defects from measured data and proposed Weibull model on Figure 3.

In **step 5,** we have tried proposed Weibull model for ERP system reliability prediction. Reliability prediction is useful if the existing ERP system is changed in any way (new functionality, regular upgrade etc.). That can be done by the manufacturer for all reported defects (from all customers) or for the specific (one customer) installation like we did in our case study. In that case, our task is to set the procedure to automatically determine the Weibull parameters α and β for the new version of software that will be developed [5].

Table 2. Kolmogorov-Smirnov test for estimated Weibull distribution

Sample size	73				
Statistic	0,0581				
P-Value	0,95411				
α	0,2	0,1	0,05	0,02	0,01
Critical value	0,1233	0,1409	0,15649	0,17498	0,18776
Reject?	No	No	No	No	No

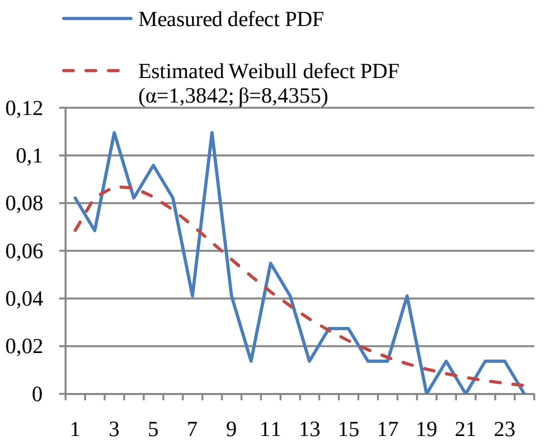

Figure 2. Measured and estimated Weibull defect distribution

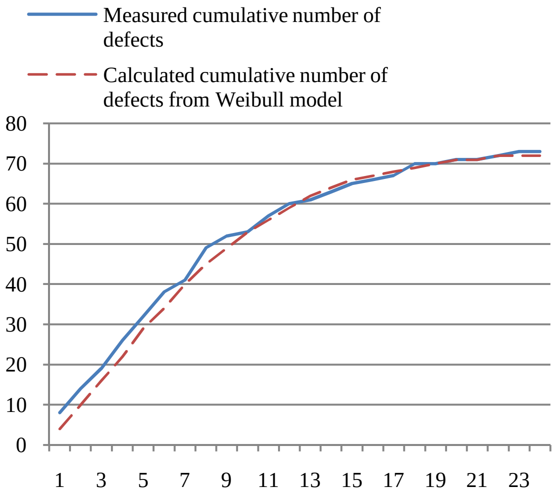

Figure 3. .Measured and calculated cumulative number of defects

Example: Reliability prediction in the 17th month of usage, after system upgrade.

In this example we assume that Weibull reliability prediction model is first time applied on presented ERP system installation one year after the software was deployed and then every time when software was upgraded (every month). Weibull parameters α and β are estimated from measured data before upgrade. After two or more estimations of α and β from measured data, prediction could be made for future values of α and β with Microsoft Excel trend estimation statistical technique (Figures 4 and 5). All the results obtained are presented at Table 3.

After 16 months, Weibull parameters are estimated from measured data and prediction for α in the 17^{th} month of usage is made from trend estimation, presented at Figure 4.

$$\alpha_{17p} = -0{,}0185*17+1{,}5895 = 1{,}4785$$

Prediction for β in the 17^{th} month of usage is made from trend estimation, presented at Figure 5.

$$\beta_{17p} = 0{,}2186*17+5{,}9706 = 7{,}2822$$

Measured values for α and β for 16^{th} month (Δt_{16}), from Table 3. are:

$$\alpha_{16m} = 1{,}4986$$

$$\beta_{16m} = 7{,}0432$$

According to (4), from measured α_{16} , β_{16} in Table 3:

$$f_{a16m}= 0{,}01048$$

According to (4), from predicted α_{17p} , β_{17p} in Table 3:

$$f_{a17p}= 0{,}01204$$

According to (2):

$$\lambda_{17\mathrm{p}}/\lambda_{16\mathrm{m}} = f_{a17p}/\ f_{a16m}$$

$$\lambda_{17\mathrm{p}} = 1{,}1489$$

$$\lambda_{16\mathrm{m}}= \mu_{16m}\text{-}\mu_{15m}= 66\text{-}65=1$$

Predicted cumulative number of defects for the 17^{th} month is:

$$\mu_{17\mathrm{p}}= \mu_{16\mathrm{m}} + \lambda_{17\mathrm{p}}\ \Delta t_{17} \approx 68$$

Table 3. Prediction for Weibull parameters with trend estimation from measured values

month	α measured	β measured	α predicted	β predicted	measured cumulative defects	predicted cumulative defects
12	1,5673	6,2268			60	
13	1,5583	6,3465			61	
14	1,5348	6,6158	1,5493	6,4662	63	
15	1,5103	6,8994	1,5208	6,7854	65	
16	1,4986	7,0432	1,4938	7,0939	66	
17	1,4857	7,1956	1,4785	7,2822	67	68
18	1,4437	7,6913	1,4645	7,4428	70	
19	1,4437	7,6913	1,4354	7,7909	70	
20	1,4315	7,855	1,4199	7,974	71	
21	1,4315	7,855	1,4063	8,1459	71	
22	1,4172	8,0354	1,3983	8,2457	72	
23	1,4007	8,2321	1,3876	8,3707	73	
24	1,3842	8,4355	1,376	8,5147	73	

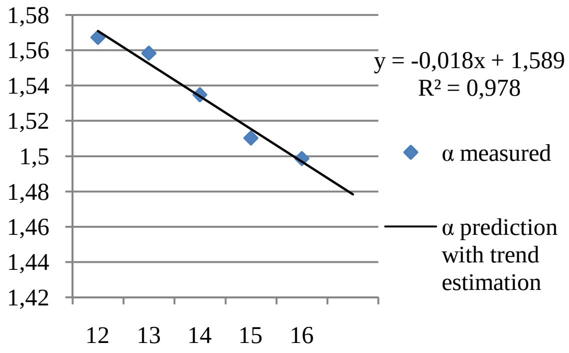

Figure 4. Trend estimation for parameter α at 16th month of usage

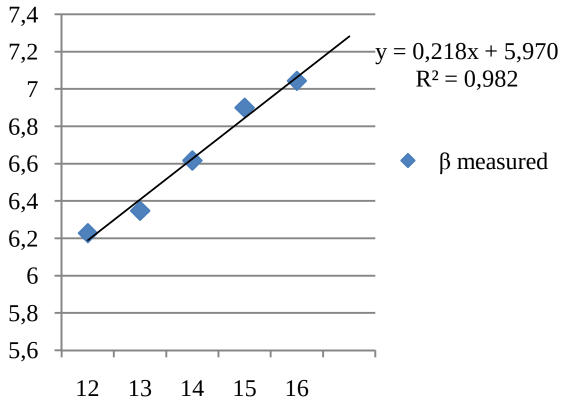

Figure 5. Trend estimation for parameter β at 16th month of usage

Similar prediction can be made for longer period, for an example for the 20th month, at the 16th month:

$$\mu_{20p} = \mu_{16m} + \lambda_{17p}\,\Delta t_{17} + {} + \lambda_{18p}\,\Delta t_{18} + \lambda_{19p}\,\Delta t_{19} + \lambda_{20p}\,\Delta t_{20}$$

$$\Delta t_{17} = \Delta t_{18} = \Delta t_{19} = \Delta t_{20} = \Delta t = 1 \text{ month}$$

$$\lambda_{18p} = \lambda_{17p} f_{a18p} / f_{a17p}$$

$$\lambda_{19p} = \lambda_{18p} f_{a19p} / f_{a18p}$$

$$\lambda_{20p} = \lambda_{19p} f_{a20p} / f_{a19p}$$

According to trend estimation for α and β from Figures 4 and 5 at 16th month for next periods and (4):

$$\mu_{20p} \approx 72$$

The obtained results for expected cumulative number of defects correspond well with measured values from Table 3. It is clear that longer predictions could be made in situations when the observed ERP system is stable. In presented case study that moment is 10th month of usage (from Figure 3). In that moment user has implemented all new functionalities and employees have learned how to use installed ERP.

Parameter α is shape parameter of the Weibull distribution and according to presented case study and mathematical definitions [10] can be interpreted as follows:

- Since the parameter α is decreasing over time (Figure 4), Weibull distribution becomes exponential and defect rate is constant (for α=1) or decreasing over time (defects are rare and existing installation is tending to be stable)
- Since the value of parameter α is increasing over time, defect rate is increasing over time (defects are often and existing installation is tending to be unstable).

7 Conclusion

Measurement-based analysis represents a good foundation for the future work in modeling of ERP system reliability. This paper gives example how to use existing user-generated ERP software defect reports in choosing of an appropriate SRGM in reliability modeling and later in reliability prediction. It is important to emphasize that we have analyzed data for only one installation of described ERP product, collected during system usage after software acquisition. It is good approach for the situation when user wants to validate product maintenance and predict the future trends during product usage. Another important point is that we have analyzed maintenance log from the moment when the product was in full production so the number of reported defects is relatively small.

The future research should cover the ERP software implementation phase before full production when amount of reported defects is bigger in smaller time interval. It will be also interesting to use presented analysis for one ERP product with reported defect data from different customer installations.

References

[1] Michael R. Lyu, **„Handbook of Software Reliability Engineering“**, Computer Society Press. ISBN: 0-07-039400-8. 1996

[2] James, D. and Wolf, M. L. **"A Second Wind for ERP"**, The McKinsey Quarterly, (:2), 2000, pp. 100-107.

[3] Asad, C.A.; Ullah, M.I.; Rehman, M.J.-U., **"An Approach for Software Reliability Model Selection",** Computer Software and Applications Conference 2004., Proceedings , Page(s): 534 - 539 vol.1

[4] Frane Urem, Želimir Mikulić, **"Using NHHP model for ERP software module reliability growth**", MIPRO 2010, , Proceedings of the 33rd International Convention,page(s): 165-168

[5] Hribar L., Duka D., **"Software component quality prediction in the legacy product development environment using Weibull** and other mathematical distributions ", SoftCOM 2009. , Page(s): 254 - 259

[6] CarPRO Systems, `http://www.carprosystems.com`

[7] EasyFit Distribution Fitting Software, `www.mathwave.com`

[8] Thomas Wailgum,"ERP Definition and Solutions", `http://www.cio.com/article/40323/ERP_Definition_and_Solutions, 2008.`

[9] `http://www.erpweb.com/erpforever.htm`

[10] Weibull, W.,**"A statistical distribution function of wide applicability**", 1951.,J. Appl. Mech.- Trans. ASME 18 (3): 293–297