Empir Software Eng (2007) 12:161–182
DOI 10.1007/s10664-006-9018-0

A replicated empirical study of a selection method for software reliability growth models

Carina Andersson

Published online: 20 October 2006

Editor: Pankaj Jalote

Abstract Replications are commonly considered to be important contributions to investigate the generality of empirical studies. By replicating an original study it may be shown that the results are either valid or invalid in another context, outside the specific environment in which the original study was launched. The results of the replicated study show how much confidence we could possibly have in the original study. We present a replication of a method for selecting software reliability growth models to decide whether to stop testing and release software. We applied the selection method in an empirical study, conducted in a different development environment than the original study. The results of the replication study show that with the changed values of stability and curve fit, the selection method works well on the empirical system test data available, i.e., the method was applicable in an environment that was different from the original one. The application of the SRGMs to failures during functional testing resulted in predictions with low relative error, thus providing a useful approach in giving good estimates of the total number of failures to expect during functional testing.

Keywords Replication · Software reliability

1 Introduction

Many software reliability growth models (SRGMs) have been proposed to estimate the reliability of a software system. Software reliability, one of the most important attributes of software quality, is closely related to defects. It is assumed to grow as defects are corrected and removed from the software. To estimate the remaining number of defects in a software system under test, SRGMs can be applied to guide test management in their decisions whether to continue or stop testing. This paper reports on a replication of a study, originally conducted by Stringfellow and Amschler Andrews (2002), where a method for selecting SRGMs was suggested and applied to make these release decisions.

C. Andersson (✉)
Department of Communication Systems, Lund University, Box 118, 221 00 Lund, Sweden
e-mail: Carina.Andersson@telecom.lth.se

In the last few years, the importance of replicating research studies has received growing attention in the empirical software engineering community. A finding cannot be established as the "truth", based on a single study, since small variations in the execution of a study can have a large effect on the results. Nevertheless, to understand the fundamental principles behind the software development phenomena studied, an attempt to run an exact replication is most often not feasible (Robson 2002), and not even ideal. Instead, replications in a variety of environments are a basis for obtaining more robust and generalizable results. Miller (2005) discusses the topic and argues that to receive meaningful results, a step towards families of studies investigating a single hypothesis is necessary. Miller focuses on replications of controlled experiments, although replications can also be applicable to other forms of studies, either to produce support for a particular theory, or to question the claims of the original study.

Hence, the goal of this study is to explore the applicability of the results of an original study in a different context, rather than repeating the study under the same conditions to verify the exact results. Stringfellow and Amschler Andrews (2002) proposed a selection method for determining the most appropriate software reliability growth model in terms of predictive ability, stability and curve fit. SRGMs are used to estimate the remaining number of defects in a software system of interest to help management with release-decisions during testing. The method is empirically evaluated in a case study on data from three releases of a large medical record system. The results of the original study show that the selection method worked well on the three data sets presented in the case study, although several underlying assumptions of the SRGMs were violated, when applying the models on the real-world data.

This paper presents a replication of the study by Stringfellow and Amschler Andrews. The proposed selection method is implemented and applied in a different environment to evaluate its usefulness and applicability. That is, in a new case study the method's applicability is validated, with failure data from three telecommunication software system projects. The replication is one step towards a generalization of the selection method. However, it is important to notice that one replication is not enough. In the study, some parameters are changed, compared to the original study, while the basic ideas of the selection method are maintained. The selection method in this case is applied to a much larger number of failures (approximately 10 times larger), while the projects' lead-times are once to twice the length of the original study. This meant that the evaluation criteria in terms of the given values of stability and curve fit used in the original study could not be transferred to this new context without adaptation. In addition, the study by Stringfellow and Amschler Andrews examined the method with failure data from system test, while in this study the selection method is applied to failure data from system test, but also to failure data from functional testing to investigate whether useful predictions could be made during this earlier test activity.

The paper is structured as follows: Section 2 describes SRGMs and their underlying assumptions. Also, the original study by Stringfellow and Amschler Andrews is described. Section 3 describes our approach of the replication study, presents the failure data and the differences compared to the original study. In addition, the findings compared to the results of the original study are presented. Finally, in Section 4 the conclusions are presented.

2 Background

Quality, cost and schedule have been declared as the most important software project characteristics (Musa et al. 1987). The latter two are quantitatively measurable, while

quality is more difficult. Software quality has a wide range of attributes, such as functionality, usability, portability, and maintainability (ISO 2000), and thus there is no single concrete measure for software quality. Software reliability, however, could be seen as a key factor in software quality, since it quantifies software failures. Software reliability is defined as the *probability of failure-free operation* of a software program for a *specified time* in a *specified environment* (Musa et al. 1987).

Several definitions of error, fault and failure exist in the literature. A crucial part in applying software reliability measurements concerns separating these definitions. In this paper we use the following terminology, as defined by IEEE (1990): an *error* of commission or omission causes a *fault* in the code, which in turn manifests itself as a *failure* that can be observed during software testing or operation. The program has to be executed for the failure to occur. Hence, the failure is something dynamic. The failure behavior is obviously affected by the number of faults existing in the software being executed (Musa et al. 1987).

According to the definition of reliability given above, another aspect of software reliability measurement is *time*. The reliability quantities are related either to the *execution time* for a software system, which is the CPU time actually spent by the computer executing the software, or the *calendar time*. The third aspect of the software reliability definition concerns the execution environment. The environment is described by the operational profile. Musa et al. (1987) describes the concept of operational profiles. An operational profile consists of the set of operations that a system is designed to perform and their probabilities of occurrence. Thereby, a quantitative characterization of how the system will be used is provided.

2.1 Software Reliability Growth Models

The assumptions of SRGMs generally state that the models are applicable during system test, where cycles of test executions, observed failures, repair, and continued testing are repeated. Changes in failure rate over time can be used by management to make a decision about when to stop testing. Practical experiences of the use of reliability growth models in a variety of contexts are published, e.g., by Musa and Ackerman (1989), Ehrlich et al. (1993), Wood (1996, 1997), and Jeske and Zhang (2005).

Several SRGMs assume that the observed failures occur as a non-homogenous Poisson process (NHPP). This means the failure intensity is not a constant. As faults are detected and removed from the software, it is expected that the observed number of failures per time unit will decrease. The expected number of failures observed by time t is given by $\mu(t)$, with the boundary condition $\mu(\infty)=a$, where a is the expected number of failures to be observed eventually. Four common SRGMs are used in the original study, the basic Musa or Goel-Okumoto (G-O) model (Goel and Okumoto 1979; Musa et al. 1987), the delayed S-shaped model (Yamada et al. 1986), the Gompertz model (Kececioglu 1991), and the Yamada

Table 1 SRGMs used in this study

Model	Type	Equation $\mu(t)$	Reference
G-O	Concave	$a\left(1-e^{-bt}\right)$, $a\geq 0$, $b>0$	Goel and Okumoto (1979)
Delayed S-shaped	S-shaped	$a\left(1-(1+bt)e^{-bt}\right)$, $a\geq 0$, $b>0$	Yamada et al. (1983)
Gompertz	S-shaped	$a\left(b^{c^t}\right)$, $a\geq 0$, $0\leq b\leq 1$, $c>0$	Kececioglu (1991)
Yamada	Concave	$a\left(1-e^{-bc\left(1-e^{(-dt)}\right)}\right)$, $a\geq 0$, $bc>0$, $d>0$	Yamada et al. (1986)

exponential model (Yamada et al. 1986). These models are all based on a NHPP. The same models are used in this replication study. Table 1 gives an overview of the models.

SRGMs can be classified into two major classes, concave and s-shaped models (Wood 1996). The concave models assume a pattern of decreasing failure rate, while the s-shaped models assume that early testing is not as efficient as later testing. An s-shaped growth curve may reflect the initial learning curve at the beginning of the test process, as test teams become familiar with the software system and its testing procedures, followed by growth where the failure detection rate increases, and finally leveling off when the remaining faults become fewer and more difficult to detect.

Modeling the software failure process can hardly be expected to be precise, and underlying assumptions are necessary for each model, some more reasonable than others. The assumptions for each model should be evaluated in terms of the test environment from which the failure data is obtained. Wood (1997) gives an overview of a list of assumptions and discusses their accuracy for the test environment used in his study. For example, in general, all software reliability models assume that the defect detection that occurs during testing follows an operational profile (Lyu 1996; Musa 1999). In addition, with some exceptions (e.g., Fujiwara and Yamada 2003), most reliability models assume a perfect debugging environment, i.e., the defects are assumed to be corrected immediately without inserting any new faults. In practice, software faults may not always be fixed during debugging, and new faults may be introduced.

Although several of the SRGMs' assumptions might be violated, the models can be applied to fit a curve to a set of data points representing the cumulative number of failures. The fit to the data set is examined by evaluating the deviation between the observed cumulative number of failures and the fitted values, by using a statistical test, e.g., χ^2, Kolmogorov-Smirnov and R^2. Gaudoin et al. (2003) evaluated the power of several of this type of tests, applied to a number of reliability models, showing that for example the simple R^2-test had as much power as the other tests and in some occasions more. More information on the specific tests can be found in the literature by Siegel and Castellan (1988) and Montgomery (2001).

2.2 The Original Study

The original study, conducted by Stringfellow and Amschler Andrews (2002), describes an approach applying several SRGMs to cumulative failure data grouped by week to select the model(s) that best fit the data. By a good fit of a model, more accurate predictions of total number of failures are expected, thereby providing decision support for whether to stop testing and release the software product or to continue testing for another week. The predictions are based on test time and failure data. Stringfellow and Amschler Andrews (2002) used calendar time to measure test time, since that was the only time measurement available. The failure data consisted of defect reports, reported during system test.

The four SRGMs used in the study, the basic Musa or Goel-Okumoto (G-O) model (Goel and Okumoto 1979; Musa et al. 1987), the delayed S-shaped model (Yamada et al. 1986), the Gompertz model (Kececioglu 1991), and the Yamada exponential model (Yamada et al. 1986), were selected because they represent a range of assumptions. Important to consider when choosing which models to apply, is simplicity. To persuade practitioners to use a SRGM, it has to be simple in concept and allow inexpensive data collection. A user without extensive mathematical background should be able to understand and apply the model (Musa 1999). This makes the models in this study good candidates.

Three model evaluation criteria are used in the original study, the *goodness of fit measure* (GOF), the *prediction stability*, and the *predictive ability*. Stringfellow and

Amschler Andrews (2002) have chosen to base their GOF measure on the simple R^2 test. The choice is motivated by Gaudoin et al. (2003), who have evaluated the power of several statistical tests for GOF for a variety of reliability models. The larger the R^2 value, the better the fitted equation explains the variation in the data. The evaluation showed that this measure was as least as powerful as the other GOF tests compared. Stringfellow and Amschler Andrews chose a threshold of R=0.95.

In addition to the GOF measure, the models are evaluated in terms of prediction stability. A threshold for the stability is set; the prediction in week i should be within 10% of the prediction of week $i-1$. The threshold value of 10% is subjectively chosen, motivated by a rule of thumb given by Wood (1996).

Prediction accuracy is the last model evaluation criterion used in the study. The predictive ability is measured in terms of *error* (estimate-actual) and *relative error* (error/actual).

The proposed method for selecting SRGMs based on the models that best fits the data consists of several steps. Stringfellow and Amschler Andrews give a detailed description

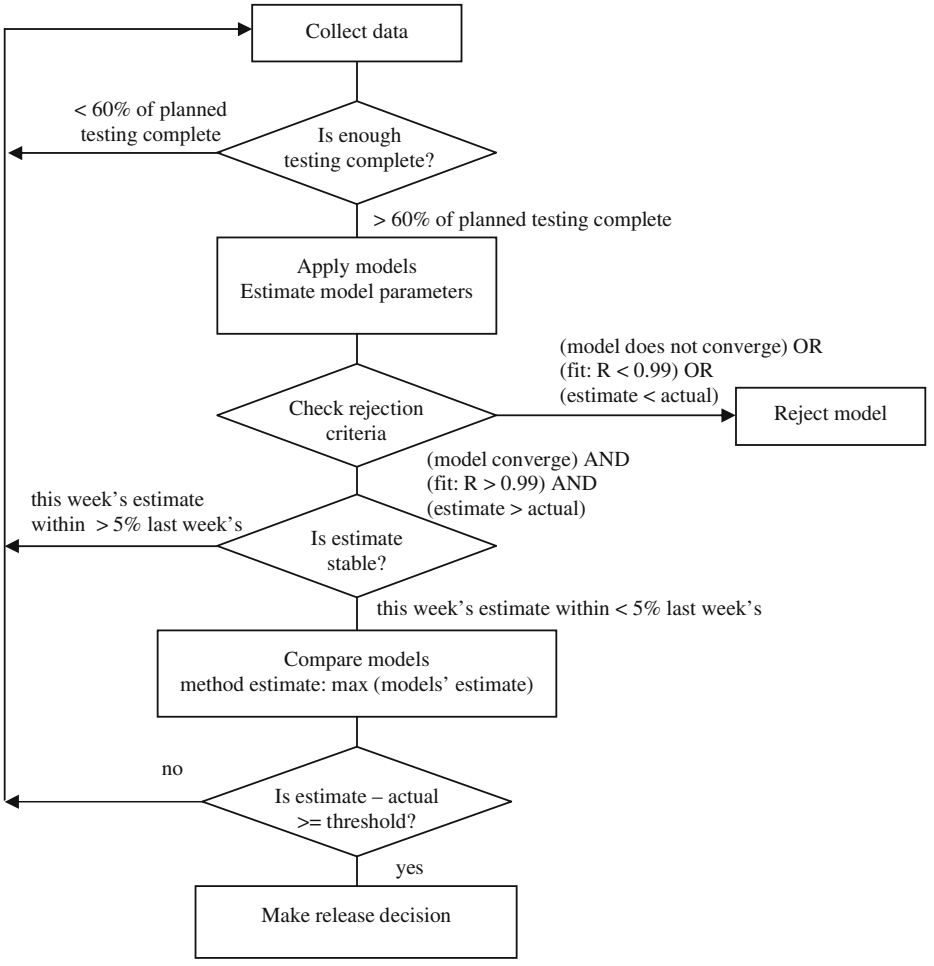

Fig. 1 Flowchart for the selection method. Derived from (Stringfellow and Amschler Andrews 2002). The rejection criteria are based on the threshold values from the replication study

of each step of their approach. A short summary is presented below and illustrated in Fig. 1.

The procedure for the selection method is executed once a week (on the assumption that the cumulative number of failures are grouped by week), starting with recording the cumulative number of failures found. The next step is considering whether it is appropriate to apply the models. In case only a minor part of the execution of the test plan is complete, it may not make sense yet to apply the SRGMs. Stringfellow and Amschler Andrews recommend that at least 60% of the planned testing is complete before applying the models, which also is motivated by Ehrlich et al. (1990). Once testing has proceeded this far, the chosen SRGMs are applied to the data at the end of each week, using a commercial curve fit program. The curve fit program attempts to fit each model to the data. A model diverges if no fit can be performed to the data set. Then the model is considered inappropriate and excluded from the selection procedure in future weeks. However, if a fit is obtained and the model converges, the program outputs the model's parameters such as the estimate for the expected number of total failures. GOF of the fitted curve is also evaluated, based on the R^2 value. Models are considered inappropriate for the data set if the R value is below the threshold value $R=0.95$. These models are excluded from the selection procedure in subsequent weeks. Another criterion could exclude the models from the selection procedure; if the estimate of expected total number of failures given by the curve fit program is lower than the actual number already detected, this could lead to a false sense of security, and will thereby lead to exclusion of the model in subsequent weeks. Thereafter, the model stability is examined with the stability threshold of 10% of the previous week's prediction. If no model has stabilized by giving a prediction within the 10% interval, testing should continue and failure data be collected for another week. With at least one stable model, the final step in the approach gives the estimate that is the maximum estimate of all stable models. A decision based on the difference between the estimate and actual number of failures detected determines whether testing should continue or if the developed software system has reached an acceptable level. This threshold is decided by management and very much dependent on type of system, e.g., whether it is safety-critical or not.

The results from the study by Stringfellow and Amschler Andrews show that the selection method worked well. With the chosen threshold values for GOF and stability, at least one model was acceptable by the time testing was approaching a decision point for stopping. The selection method seemed to differentiate between the included models, and the predictions corresponded well with the actual numbers of failures.

3 The Replication Study

The goal of our study is to replicate the study by Stringfellow and Amschler Andrews, and to evaluate their suggested approach for selecting SRGMs. We have monitored the trends of failures detected and attempted to predict the testing process with the selection method for SRGMs. However, as in Stringfellow and Amschler Andrews, our study also had failure data that violated some of the underlying assumptions.

The following section presents our approach of the selection method. A description of the cases study is given, with detailed information of the data. Differences from the approach taken by Stringfellow and Amschler Andrews are presented.

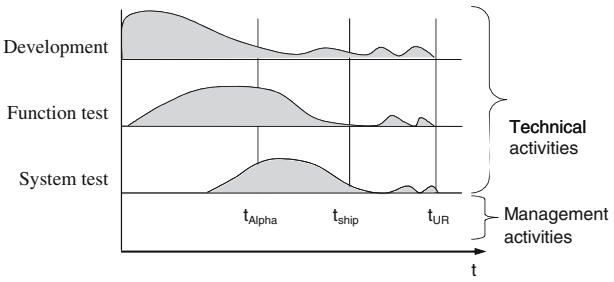

Fig. 2 Project work flow

3.1 Case Study

The failure data used in this replication study comes from three software development projects conducted in the telecommunications industry. The development organization has a high market sensitivity, which implies a strong need for early indicators of project progress and specifically fault content in the developed products, to enable early actions to reduce costs and to plan preventative test activities.

The software projects follow an incremental development process, with an extensive number of iterations. Project duration is a number of months. The development projects are divided into *feature groups*, each having responsibility of a specific part of the functionality to be developed. In each iteration, the feature groups are delivering components with the required functionality of the main software system. In addition to the developers in the feature groups, these groups also include functional testers. The developers do the unit testing themselves. After that the functional testers take over the testing responsibility of the implemented software.

System test is a test organization that is independent from the feature groups. The system test organization runs its test suites on the latest available version of the product. Testing is performed both in the development environment and in the real operational environment.

Due to the incremental development environment, several activities occur in parallel. An overview of the workflow is given in Fig. 2 illustrating overlapping activities. Some important milestones used in the development process are marked. Before *Alpha* the main activities are development and functional test, while the main part of system test starts at *Alpha*. At *ship date* the first version of the system is released. After this milestone testing and debugging continues. The versions with the corrected implementations are released at *update releases (UR)*.

3.2 Data

The data sets used in this replication study are obtained from a large failure report database. Data from the three software projects is extracted from the database, classified by detection date, and whether the failure was detected by a functional tester or a system tester. In this way, failure reports are separated into functional test failures and system test failures. The data sets are not limited to failures detected by system test as is the case in several other studies (Wood 1996; Stringfellow and Amschler Andrews 2002). We chose to use the full data set available, and the subsets of it obtained through data partitioning into functional and system test failures, since this approach enables us to do more detailed analyses. The functional test failures are detected in testing procedures, which are considered to be feature related.

Combining this type of data with ordinary reliability growth testing data is quite contrary to what is advisable (Lyu 1996), at least if the test cases are not selected randomly according to an operational profile. In this case, no operational profile is used for test case selection, but on the other hand, the functional tests do not proceed sequentially. The feature groups execute functional test cases in parallel, each group focusing on its own functionality, resulting in a mixed test suite execution of different features.

The failure data is accumulated per week, using calendar time, since execution time data is not available in the organization. In the development projects, failures were reported in relevance to calendar time, with a granularity of days. Hence, we use this scale. This also was motivated by a rather constant test effort per week. However, there were holidays, such as summer vacations and Christmas, when the test effort was less than normal. There were also regression test periods when the test effort was more intensive than normal, which constituted a significant percentage of the calendar time. For this reason, a modified time scale is used, which takes into account only testing days and compensates for the non-constant test effort.

Duplicate detection of failures is not included in the data sets, that is, only one failure report is kept per observed failure. If the same failure is detected more than once, only the first is entered into the database. Also, if an underlying fault is causing different failures, only one failure report is kept for the analysis. Hence, only unique faults are represented in the study. We did not consider change requests nor problems that are not code faults in the analysis.

The predictions from the SRGMs, based on the available data sets, are compared to the cumulative number of failures detected after t_{ship}. Hence, failures detected post-ship are also separated into failures detected in functional test and system test, i.e., the number of failures detected by each test activity *after* t_{ship}. Thus, to examine predictive ability, the predictions obtained from the data sets, consisting of failures detected by functional test is compared to the actual number of failures detected by the same test activity after t_{ship}. The same goes for system test failures.

3.3 Comparison to the Original Study

Some differences exist between our study and the one conducted by Stringfellow and Amschler Andrews. These differences are presented below with a summary of aspects considered during the application of the suggested selection method.

- Can calendar time be used when the original models assume execution time? Stringfellow and Amschler Andrews investigate the use of calendar time. We also use a sort of calendar time. However, the time used is modified to avoid the reflection of differences in test effort due to holidays and more intensive periods of testing.
- How robust are the models when the underlying assumptions are not met? As mentioned, when applying SRGMs in practice, several of their stated assumptions are often violated, which is the case also with the data in this case study. In this study we can assume an imperfect debugging environment where new faults may be introduced by the correction of detected faults. The code base is not held constant, but grows during the test process when the developers iteratively deliver new functionality to the main software system. In addition, we apply the SRGMs on both failure data from system test and functional test.
- What is a good fit of the models? Analogous to Stringfellow and Amschler Andrews, we use the R^2 value as GOF measure. However, the number of data points of detected failures is not in the same range as in their study (see Table 2), which implies that their chosen value is not appropriate in our study. Stringfellow and

Amschler Andrews base their threshold value on a discussion by Gaudoin et al. (2003) who evaluate a different model, although they find the value associated with a high confidence level. Also based on Gaudoin et al. we chose a threshold value R= 0.99, to better reflect the models' appropriateness in this application, since the critical value depends on the number of data points. An alternative to setting a threshold value is to base the selection method on choosing the SRGM that has the highest R^2 value, and exclude the rest. However, the purpose is to maintain the iterative procedure in the selection method. We do want to keep more than one candidate in the selection process, especially those which may give reasonable predictions later. The chosen threshold value is noticeably higher than the value used by Stringfellow and Amschler Andrews, R=0.95, which might not have resulted in any selection at all applied to our data sets. The effect of choosing the threshold value R=0.99 is further discussed in the subsections presenting the application of the selection method.

- What is considered a stable model? Related to the discussion of the GOF measure, with our larger data sets in terms of data points and number of failures, the higher number of detected failures compared to the study by Stringfellow and Amschler Andrews implies that the stability measure may not be relevant at the 10% level. We have chosen the threshold at 5% of the prediction of the previous week, to keep the number of models remaining in the selection process at an appropriate level.
- Which estimate is appropriate to use when more than one model is stable? Stringfellow and Amschler Andrews chose conservatively, i.e., the maximum estimate when more than one model was viable. Often there is a large difference in the estimates given by the different models and we did not find this approach applicable in every case. We chose to evaluate the selection method on the same aspect, the highest estimate, as in the original study. However, this should be carefully considered when applying the selection method in an industrial context. We noticed that further reflection on the estimates from the non-rejected models must be done to avoid unnecessary time spent on testing.
- What should the predicted values be compared to? Stringfellow and Amschler Andrews use the number of failures reported after release as the comparative value to calculate predictive ability and relative error. Our corresponding value for the calculation of relative error is based on the number of failures detected until t_{UR} according to Fig. 2, separated into the two categories of failures detected by system test and functional test. Since newly corrected code is implemented after t_{ship}, functional test failures continue to occur after this milestone. The parallel activities

Table 2 Project data for original study and replication study

	Original study			Replication study		
	Release 1	Release 2	Release 3	Project 1	Project 2	Project 3
Duration ST (weeks)	18	17	13	22	27	25
Duration FT[a] (weeks)	–	–	–	35	46	41
Number of failures (ST)	231	245	83	585	1,330	3,839
Number of failures (FT)	–	–	–	2,704	4,802	5,343

FT=functional test, ST=system test

[a] The functional test runs in parallel with the system test in the replication study. i.e., the end dates are identical, although the system test starts after the functional test and the duration in number of weeks are shorter

of system test and functional test require this approach, although the main portion of functional test is conducted before system test. Functional test failures are assumed to not be detected by system test. However, the size of the fraction of failures that could be detected both by system test and functional test is not known, but in this study is believed to be small.

3.4 Results

In this section we present the results from the replication study. In Section 3.4.1 the selection method for SRGMs is replicated on system test failures, while Section 3.4.2 presents the new approach where the SRGMs are applied to the data sets of functional test failures.

3.4.1 System Test Failures

In the following we present the data from the three projects while applying the G-O, delayed S-shaped, Gompertz, and Yamada models to failure data detected by system test, starting at 60% of the planned testing and ending at the ship date. The columns show the test week, the cumulative number of failures found, and for each model: the prediction of total number of failures and the adherent GOF-value (R-value). An *S* indicates that the model is stabilizing in the specific week, while a *D* indicates a destabilization of the model. An *R* indicates the model is rejected in the selection method and not considered as an appropriate model in future weeks.

The tables show the prediction data when applied to the failure data detected in system test. Notice that test time is counted from the beginning of system test; in parentheses the functional test time is given.

Project 1 system test data and the application of the SRGMs are shown in Table 3. Both concave models are rejected because they do not converge in test week 16. The rejection is expected when inspecting the cumulative failure curve, which clearly is s-shaped. Both models do converge later on, although they give very high estimates and have R-values well below the threshold value of 0.99.

The delayed S-shaped model and the Gompertz model have better fits to the curve, although the delayed S-shaped model does not stabilize until week 21, while the Gompertz model stabilizes in week 17, destabilizes the week after and stabilizes once more in week 19. Figure 3 illustrates the cumulative failure curve and the s-shaped models' estimates of total number of failures for each week, starting week 16. The appearance of the failure

Table 3 Predicted total number of failures for project 1 (ST)

Test week ST (FT)	Failures found	G-O		Delayed S-shaped		Gompertz		Yamada	
		Estimate	R-value	Estimate	R-value	Estimate	R-value	Estimate	R-value
16 (29)	226	–	– (R)	1,020	0.9963	421	0.9976	–	– (R)
17 (30)	249	–	–	948	0.9970	423 (S)	0.9981	–	–
18 (31)	252	82,300	0.9637	758	0.9965	383 (D)	0.9980	–	–
19 (32)	265	151,000	0.9680	661	0.9963	367 (S)	0.9982	–	–
20 (33)	279	89,300	0.9717	609	0.9965	364	0.9984	–	–
21 (34)	297	90,500	0.9747	590 (S)	0.9969	373	0.9986	127,200	0.9748
22 (35)	306	114,000	0.9775	567	0.9971	376	0.9987	49,830	0.9774

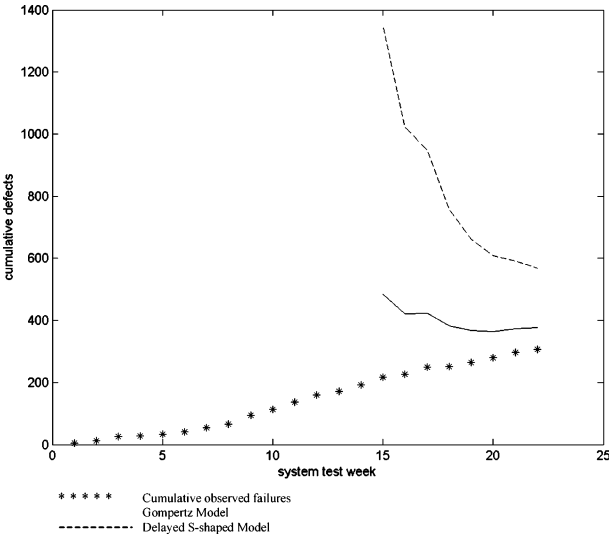

Fig. 3 Plot of project 1 data (ST) and each week's prediction of total number of failures, from SRGMs not rejected

curve from system test in project 1 is obviously rather difficult to fit. According to the original selection method, in test week 22 the delayed S-shaped model is favored, since it gives a higher estimate than the Gompertz model, although its R-value is lower than the Gompertz model's.

Compared to the total number of failures reported, the predicted value of the delayed S-shaped model had a relative error of −0.031, while the Gompertz model had a relative error of −0.357, despite the high value for the curve fit, see Table 4.

Project 2 system test data and the results of applied SRGMs are shown in Table 5. When applying the models to the data set, the concave models give very high estimates of the total number of failures in the beginning of the presented test period, but have R-values above the threshold value. During subsequent weeks, the models' estimates are closer to more reasonable figures, and finally also stabilize in week 27.

The delayed S-shaped model and the Gompertz model stabilize in test week 18, with good curve fit and high R-values for both models. The delayed S-shaped model, which gives the highest estimate of the two, predicts the total number of failures to be 1,110 in week 18, to compare with the actual number of failures detected, 859. If the difference between the actual number and the prediction is considered too large, the decision to continue testing should be taken. Testing did continue for several more weeks.

In week 27, when the G-O model and the Yamada model stabilize, the original selection method recommends the conservative choice, to follow the maximum estimate. In this case it is the G-O model's prediction, which is 2,320 failures to compare with the actual number of detected failures of 1,100. The s-shaped models estimated the total number of failures to 1,160 and 1,180, rather close to the actual number of detected failures, and the estimated

Table 4 Final estimates and error by SRGMs not rejected for project 1 (ST) at week 22

Model	Estimate (true value: 585)	R-value	Error	Relative error
Delayed S-shaped	567	0.9971	−18	−0.031
Gompertz	376	0.9987	−209	−0.357

Table 5 Predicted total number of failures for project 2 (ST)

Test week ST (FT)	Failures found	G-O		Delayed S-shaped		Gompertz		Yamada	
		Estimate	R-value	Estimate	R-value	Estimate	R-value	Estimate	R-value
17 (36)	822	156,000	0.9964	1,100	0.9970	1,080	0.9981	137,900	0.9965
18 (37)	859	167,000	0.9966	1,110(S)	0.9974	1,080 (S)	0.9983	52,770	0.9966
19 (38)	880	33,300	0.9961	1,110	0.9977	1,070	0.9985	10,690	0.9961
20 (39)	899	7,280	0.9954	1,110	0.9979	1,060	0.9986	5,774	0.9954
21 (40)	925	4,420	0.9949	1,110	0.9981	1,060	0.9987	4,406	0.9950
22 (41)	950	3,370	0.9946	1,110	0.9983	1,070	0.9989	3,373	0.9946
23 (42)	992	2,980	0.9948	1,120	0.9983	1,080	0.9988	2,979	0.9948
24 (43)	1,027	2,770	0.9951	1,140	0.9982	1,110	0.9986	2,767	0.9951
25 (44)	1,043	2,560	0.9952	1,150	0.9982	1,120	0.9986	2,557	0.9952
26 (45)	1,069	2,410	0.9954	1,160	0.9981	1,140	0.9985	2,409	0.9954
27 (46)	1,100	2,320 (S)	0.9957	1,180	0.9980	1,160	0.9983	2,318 (S)	0.9957

values had been stable for several weeks (see Fig. 4). In Fig. 4, the predictions of the s-shaped models are shown from week 11. These stabilize already in week 15 (not presented in Table 5, since 60% of the planned testing was not completed at that time). Figure 4 also shows the predictions of the G-O model and the Yamada model (giving nearly the same estimates), starting in week 23.

After week 27 and ship date, the total number of failures detected was $1330 - 1100 = 230$. An amount well below the prediction from the G-O model, but also a little higher than the predictions from the s-shaped models. The values of relative error are presented in Table 6, where also the R-values are presented. These indicated good curve fit for each model, although the predictions were not very good.

Project 3 system test data and SRGMs model results are shown in Table 7. The selection method did not reject any of the SRGMs for the data. However, as soon as week 17, the two concave models have good curve fits, with the highest R-values (the G-O model and the

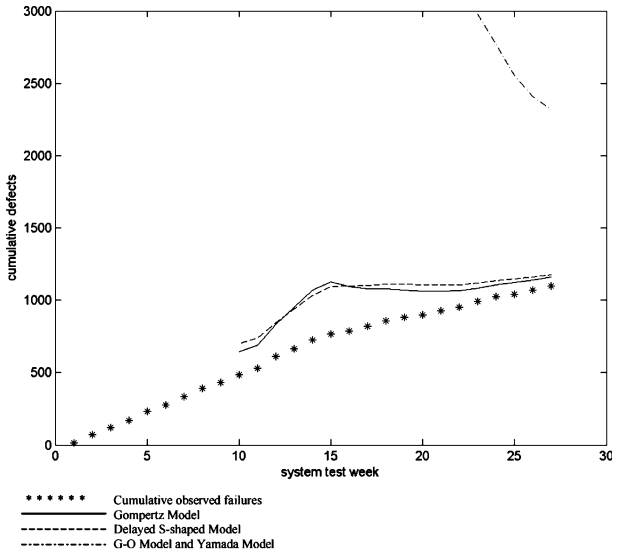

Fig. 4 Plot of project 2 data (ST) and each week's prediction of total number of failures, of SRGMs not rejected

Table 6 Final estimates and error by SRGMs not rejected for project 2 (ST)

Model	Estimate (true value: 1,330)	R-value	Error	Relative error
G-O	2,320	0.9957	990	0.744
Delayed S-shaped	1,180	0.9980	−150	−0.113
Gompertz	1,160	0.9983	−170	−0.128
Yamada	2,318	0.9957	988	0.743

Yamada model have very similar values both for predictions and curve fit for this data set). At the end of the system test period, in week 25, the Gompertz model has the highest R-value, indicating that a selection procedure based on just following the highest R-value would give a different result.

Figure 5 shows the cumulative failure curve of the failures detected by system test in project 3 and the four models' predictions for each week. The estimates from the s-shaped models start in week 18, when they stabilize. The G-O model and the Yamada models are shown from test week 23.

When reaching week 25, when the two concave models have stabilized, the original selection method recommends following these models, as they give the higher estimates (at week 25 the G-O model and the Yamada model are giving the exact same estimate). Making a decision to stop testing based on the estimates from the concave models, would probably have resulted in continued testing. At week 25, only 3,263 failures had been found, while the concave models predicted a total of 5,060 failures.

At week 25, all models had fairly good R-values. Another 576 failures were reported after test week 25. The concave models overestimated, while the s-shaped models underestimated. The values for relative error are presented in Table 8.

For this third system test failure data set, the original selection method suggested the concave models. Nevertheless, it was later shown that these models had a larger relative error than the s-shaped models. The Gompertz model's prediction was closest to the actual number of failures. It also had the best fit according to the high R-value, but, as in previous cases, underestimated.

Table 7 Predicted total number of failures for project 3 (ST)

Test week ST (FT)	Failures found	G-O		Delayed S-shaped		Gompertz		Yamada	
		Estimate	R-value	Estimate	R-value	Estimate	R-value	Estimate	R-value
17 (33)	2,707	7,750	0.9989	3,000	0.9916	3,300	0.9967	7,752	0.9989
18 (34)	2,824	7,330	0.9989	3,090 (S)	0.9920	3,380 (S)	0.9971	7,325	0.9989
19 (35)	2,905	6,860	0.9989	3,160	0.9924	3,420	0.9973	6,859	0.9989
20 (36)	2,993	6,490	0.9989	3,230	0.9927	3,460	0.9976	6,348	0.9989
21 (37)	3,050	6,110	0.9988	3,270	0.9931	3,480	0.9978	6,106	0.9988
22 (38)	3,088	5,730	0.9985	3,300	0.9936	3,480	0.9980	5,733	0.9985
23 (39)	3,151	5,460 (S)	0.9983	3,330	0.9939	3,490	0.9981	5,455 (S)	0.9983
24 (40)	3,196	5,220	0.9980	3,360	0.9945	3,490	0.9983	5,223	0.9980
25 (41)	3,263	5,060	0.9980	3,380	0.9945	3,510	0.9983	5,060	0.9979

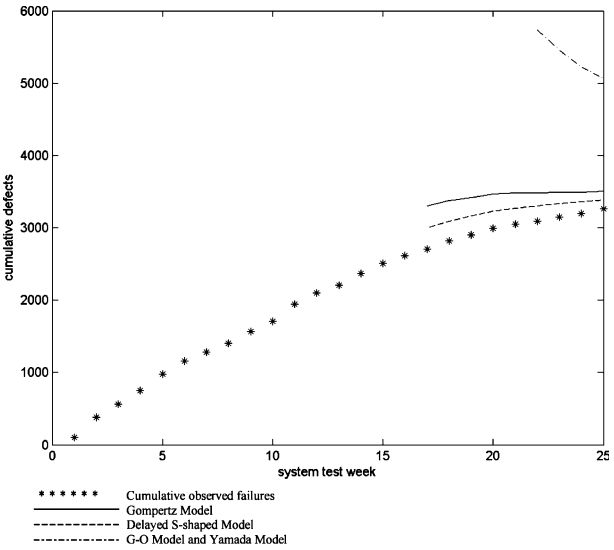

Fig. 5 Plot of project 3 data (ST) and each week's prediction of total number of failures, from SRGMs not rejected

3.4.2 Functional Test Failures

In the following we present the data from the three projects and the application of the G-O, delayed S-shaped, and Gompertz models to failure data detected by functional testing. The data sets, consisting of the cumulative failures detected during functional testing, all followed an s-shaped curve, clearly seen by visual inspection. These s-shaped data sets could not be fitted to the concave Yamada model, and the model did not converge in any of the cases using functional test data. Thus, the model is not included in the following section. The concave G-O model did not perform very well either, though it is included to illustrate how the results from a concave model differed from the s-shaped models'.

Failure data from the first project and the estimates from the SRGMs are presented in Table 9. The delayed S-shaped model and the Gompertz model perform well, as indicated by the R-values. Figure 6 shows the plot of the failure data from project 1, together with the estimates for each week for the delayed S-shaped model and the Gompertz model. The G-O model was rejected as an appropriate model, due to a low R-value, 0.9429, well below the threshold value. The Yamada model, not shown in the table, was rejected since the model did not converge in week 22, indicating that a good fit will not be obtained later on either.

Figure 6 illustrates how the Gompertz model smoothly follows the cumulative failure curve, while the delayed S-shaped model is closing in from higher estimates. Choosing the less conservative alternative with the lower predicted values instead of the highest, in this case from the Gompertz model, would result in an expected number of failures that is too

Table 8 Final estimates and errors by SRGMs not rejected for project 3 (ST)

Model	Estimate (true value: 3,839)	R-value	Error	Relative error
G-O	5,060	0.9980	1,221	0.318
Delayed S-shaped	3,380	0.9945	−459	−0.120
Gompertz	3,510	0.9983	−329	−0.086
Yamada	5,060	0.9979	1,221	0.318

Table 9 Predicted total number of failures for project 1 (FT)

Test week	Failures found	G-O		Delayed S-shaped		Gompertz	
		Estimate	R-value	Estimate	R-value	Estimate	R-value
22	1,460	104,000	0.9429 (R)	6,170	0.9914	1,750	0.9988
23	1,495	107,000	0.9485	4,850	0.9910	1,740 (S)	0.9989
24	1,575	173,600	0.9536	4,240	0.9912	1,760	0.9990
25	1,657	107,000	0.9576	3,940	0.9918	1,810	0.9989
26	1,707	120,000	0.9614	3,690	0.9923	1,860	0.9988
27	1,761	119,000	0.9648	3,510 (S)	0.9943	1,900	0.9987
28	1,800	107,000	0.9677	3,350	0.9931	1,940	0.9986
29	1,826	128,000	0.9703	3,200	0.9933	1,960	0.9987
30	1,861	106,000	0.9723	3,070	0.9934	1,990	0.9987
31	1,898	103,000	0.9741	2,980	0.9936	2,010	0.9987
32	1,915	122,000	0.9756	2,880	0.9936	2,020	0.9987
33	1,940	112,000	0.9765	2,810	0.9937	2,030	0.9988
34	1,971	106,000	0.9772	2,750	0.9938	2,050	0.9988
35	2,012	103,000	0.9779	2,700	0.9939	2,070	0.9988

low. Considering the failure detection after week 35, the Gompertz model underestimated and had a relative error of −0.234 (Table 10), even though the model had the best curve fit. The estimate from the delayed S-shaped model in week 35 was almost the same as the actual number of failures. Thus, the original selection method would choose the most appropriate model for the prediction of total number of failures to expect during functional testing.

Project 2's failure data from functional test and the estimates from the SRGMs are presented in Table 11. Again, the concave models were rejected as appropriate models. In this case the G-O model was rejected because the model does not reach the threshold value for stability.

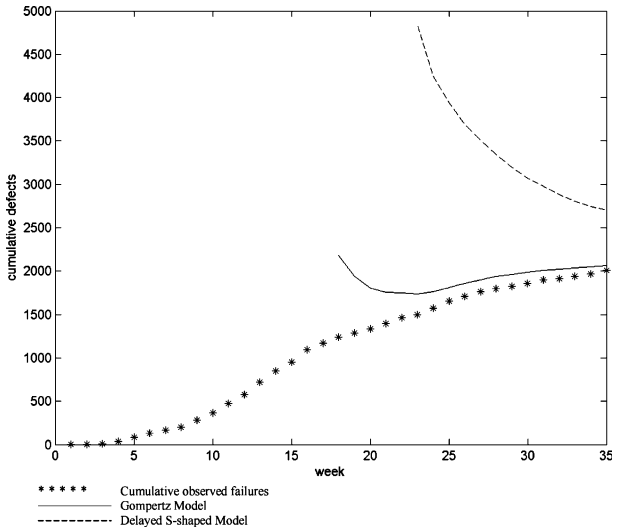

Fig. 6 Plot of project 1 data (FT) and each week's prediction of total number of failures, from SRGMs not rejected

Table 10 Final estimates and error by SRGMs not rejected for project 1 (FT)

Model	Estimate (true value: 2,704)	R-value	Error	Relative error
Delayed S-shaped	2,700	0.9939	−4	−0.002
Gompertz	2,070	0.9988	−634	−0.234

The two s-shaped models both stabilize as early as week 31, with high R-values for both models. The delayed S-shaped model is very stable, especially the last 9 weeks, with estimates ranging from 4,850 to 4,900. Figure 7 shows the predictions of the delayed S-shaped model and the Gompertz model for each week, specifically illustrating the stability of the delayed S-shaped model during the last weeks of testing.

Using the originally suggested selection method, the delayed S-shaped model's predictions should be followed during the test period, since these are higher than the Gompertz model's. In week 46 the difference between actual number of failures detected and the estimate is rather small; we cannot expect to detect many more failures. After week 46 another $4802 - 4437 = 365$ failures were reported. This gives the delayed S-shaped model's final prediction a relative error of 0.020 versus −0.023 for the Gompertz model (Table 12). Thus, the delayed S-shaped model overestimated, while the Gompertz model underestimated.

If the stability threshold would had been set to 10% of last week's prediction, as is the case in the original study by Stringfellow and Amschler Andrews, the G-O model would also have been considered appropriate for the data from functional test in project 2. Following this larger stability threshold would imply that the G-O model's estimate is used for predicting the expected number of failures. Consequently, testing would probably have continued another week, although it is unlikely the model would give estimates close to the actual value of total number of failures. Also, by visual inspection of the cumulative failure

Table 11 Predicted total number of failures for project 2 (FT)

Test week	Failures found	G-O		Delayed S-shaped		Gompertz	
		Estimate	R-value	Estimate	R-value	Estimate	R-value
30	3,453	165,000	0.9928	4,490	0.9995	3,890	0.9987
31	3,537	148,000	0.9932	4,530 (S)	0.9995	3,960 (S)	0.9986
32	3,645	151,000 (S)	0.9936	4,570	0.9994	4,050	0.9985
33	3,740	147,000	0.9940	4,630	0.9994	4,140	0.9984
34	3,803	140,000	0.9942	4,670	0.9994	4,210	0.9983
35	3,908	106,000 (D)	0.9944	4,720	0.9993	4,300	0.9982
36	4,009	61,000	0.9946	4,780	0.9992	4,390	0.9980
37	4,054	39,400	0.9948	4,820	0.9992	4,460	0.9980
38	4,097	29,700	0.9948	4,850	0.9992	4,510	0.9980
39	4,124	23,000	0.9947	4,860	0.9992	4,540	0.9981
40	4,139	18,600	0.9945	4,860	0.9993	4,560	0.9982
41	4,196	15,900	0.9945	4,870	0.9993	4,580	0.9982
42	4,251	14,100	0.9941	4,870	0.9993	4,600	0.9983
43	4,298	12,800	0.9940	4,880	0.9993	4,620	0.9983
44	4,342	11,800	0.9939	4,890	0.9994	4,640	0.9984
45	4,386	11,000	0.9938	4,890	0.9994	4,660	0.9984
46	4,437	10,400	0.9937	4,900	0.9994	4,690	0.9984

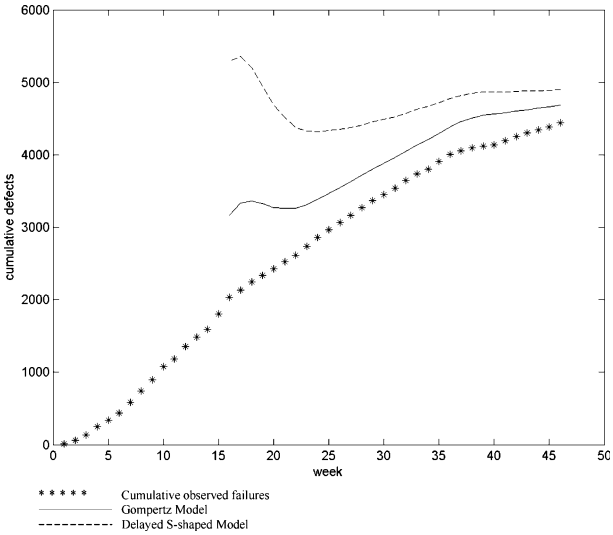

Fig. 7 Plot of project 2 data (FT) and each week's prediction of total number of failures, from SRGMs not rejected

curve in Fig. 7, an s-shaped model would be more appropriate than a concave model. Therefore, the rejection of the G-O model is justified for the data from functional test in project 2, in this case based on the stability threshold.

Project 3's failure data from functional test and the estimates from the SRGMs are presented in Table 13. Similar to projects 1 and 2, the cumulative failure data are more s-shaped than concave. Thus, the G-O model and the Yamada model were rejected. The G-O model was rejected because of a low R-value in week 27, clearly below our threshold of 0.99, although above the threshold value chosen in the original study by Stringfellow and Amschler Andrews. Even if the more conservative threshold from Stringfellow and Amschler Andrews was followed, the G-O model would be rejected. The model destabilizes in week 37 and does not stabilize again, using either stability threshold.

The s-shaped models remain in the selection procedure and both stabilize in week 28. Again, the delayed S-shaped model gives the highest estimate in week 28, and the original selection method tells us that decisions should be based on this model's prediction. Figure 8 illustrates the stability of the estimates from the delayed S-shaped model. Meanwhile the Gompertz model also has a good fit with high R-values, but is adjusting its estimates each week to meet the changes in the failure data. Unfortunately, this behavior could lead to misjudgment, letting management believe that the remaining number of failures is lower than it really is. This can be avoided by having a set of models to use in the selection procedure, and in this case the delayed S-shaped model is favored, partly because of its higher estimate but also its stable behavior.

Table 12 Final estimates and errors by SRGMs not rejected for project 2 (FT)

Model	Estimate (true value: 4,802)	R-value	Error	Relative error
Delayed S-shaped	4,900	0.9994	98	0.020
Gompertz	4,690	0.9984	−112	−0.023

Table 13 Predicted total number of failures for project 3 (FT)

Test week	Failures found	G-O		Delayed S-shaped		Gompertz	
		Estimate	R-value	Estimate	R-value	Estimate	R-value
27	3,403	148,000	0.9847 (R)	5,090	0.9976	3,880	0.9979
28	3,510	145,000 (S)	0.9861	5,100 (S)	0.9978	3,980 (S)	0.9979
29	3,632	157,000 (D)	0.9873	5,140	0.9980	4,090	0.9978
30	3,754	150,000 (S)	0.9884	5,180	0.9981	4,200	0.9977
31	3,900	151,000	0.9893	5,270	0.9982	4,340	0.9975
32	3,954	152,000	0.9902	5,300	0.9983	4,440	0.9975
33	3,993	159,000	0.9908	5,310	0.9984	4,500	0.9976
34	4,054	150,000 (D)	0.9912	5,300	0.9985	4,550	0.9977
35	4,128	143,000 (S)	0.9914	5,290	0.9986	4,590	0.9978
36	4,207	129,000	0.9916	5,290	0.9987	4,640	0.9978
37	4,309	150,000 (D)	0.9918	5,310	0.9987	4,700	0.9979
38	4,385	65,600	0.9919	5,330	0.9988	4,760	0.9979
39	4,447	40,600	0.9919	5,350	0.9989	4,810	0.9979
40	4,511	30,200	0.9920	5,360	0.9989	4,850	0.9979
41	4,552	23,600	0.9920	5,370	0.9990	4,890	0.9979

After week 41 and ship date, a rather large amount of failures were reported. However, by following the suggested delayed S-shaped model, this could be predicted. The Gompertz model underestimated the number of failures once again, see Table 14.

The application of SRGMs worked well on the failure data obtained from functional testing. The suggested method for selecting a model could also be used on the data sets consisting of functional test failures. It might not be appropriate as support for making a stop test decision, although it could be useful as a guide on how many more functional test failures to expect. The predictions of the total number of failures for all three projects are all very close to the actual number detected. The cumulative failure curves for the three projects were s-shaped, and, as expected, the s-shaped models performed better than the concave. The

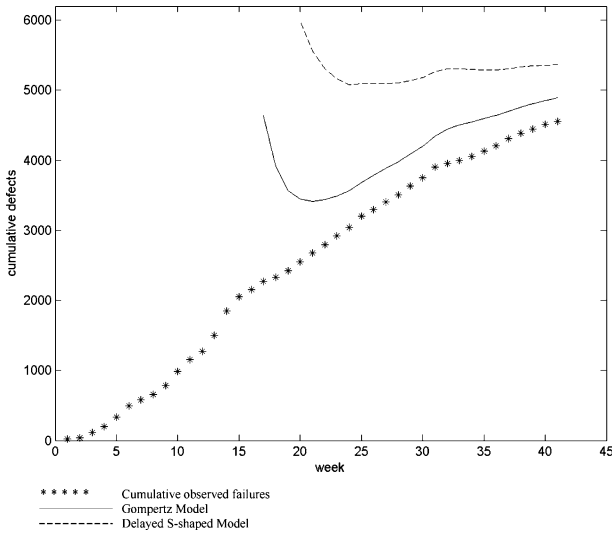

Fig. 8 Plot of project 3 data (FT) and each week's prediction of total number of failures, from SRGMs not rejected

Table 14 Final estimates and error by SRGMs not rejected for project 3 (FT)

Model	Estimate (true value: 5,343)	R-value	Error	Relative error
Delayed S-shaped	5,370	0.9990	27	0.005
Gompertz	4,890	0.9979	−453	−0.085

delayed S-shaped model and the Gompertz model both give reasonable estimates; however, the delayed S-shaped model was considered to be the most appropriate in all cases. The Gompertz model underestimated in all three cases. The concave G-O model was not appropriate for any of the 3 projects. It also gave very high estimates. The Yamada model did not converge for any of these data sets in the beginning of the selection method, indicating that a good curve fit is not obtainable. For the six data sets in this study, the Yamada model did not converge on several occasions and required considerable effort to use because of its sensitivity to starting values in the fitting process. Also, the model, when converging, gave predictions with wide confidence limits, which made the model less practical.

3.5 Findings Compared to the Original Study

Stringfellow and Amschler Andrews concluded that the selection method worked well when applied to their empirical data. Our data differ from theirs with respect to development environment and software system domain. In addition, our data consists of more data points, both in terms of number of failures and test weeks. Similar to Stringfellow and Amschler Andrews, our results show that when applying the selection method to system test data, at least one model is applicable for the data set by the time testing was approaching a decision point for stopping. Also, our modified calendar time, adjusted with regards to vacations and regression periods, did not provide any obstacles. The models gave reasonable estimates.

Compared to the original study, we did obtain some diversification among the selected models. By following the conservative choice (i.e., the highest estimate), the delayed S-shaped model was considered most appropriate for the data set of system test failures from project 1, and the concave models were considered appropriate for system test failures from project 2 and 3. However, in the latter two cases a different selection criterion would be suitable, since the concave models' high estimate resulted in a rather high relative error. In a situation like this, the selection method criterion should not be followed. Rather, each accepted model's estimate must be evaluated to find out whether it is realistic and might be considered reliable for making a stop test decision or not. Generally, the s-shaped models performed better than the concave models in predicting the total number of failures, and they had a rather low relative error, often ranging in the small amount of a couple of percents, see Table 15. However, the s-shaped models underestimated in each prediction of the system test failures, even though the absolute value of the relative error is lower than for the concave models. As seen in Table 15, the Gompertz model often had a low relative error when applied to the six data sets in this study, but in each occasion, underestimated the number of failures. For the two data sets in the study by Stringfellow and Amschler Andrews for which the Gompertz model was accepted, the model also underestimated more than the delayed S-shaped model, although the GOF measure was higher than for the delayed S-shaped model.

Compared to the original study, we changed the threshold values for GOF and stability. The GOF R-value was set to 0.99 instead of 0.95. The change manifested itself once, in project 3 for functional test failures. If the G-O model had been accepted, the model's very

Table 15 Relative errors for the SRGMs not rejected for the original study and the replication study

	Data set	G-O	Delayed S-shaped	Gompertz	Yamada
Original study	Release 1	–	−0.022*	−0.165	–
	Release 2	–	–	–	0.016*
	Release 3	–	0*	−0.036	–
Replicated study	Project 1 ST	–	−0.031*	−0.373	–
	Project 2 ST	0.744*	−0.113	−0.128	0.743
	Project 3 ST	0.318*	−0.120	−0.086	0.318*
	Project 1 FT	–	−0.002*	−0.234	–
	Project 2 FT	–	0.020*	−0.023	–
	Project 3 FT	–	0.005*	−0.085	–

*Indicates which model was selected by the selection method

high estimate would have an impact on the assessment, since the estimate is a lot higher than the other non-rejected models. On the other hand, the G-O model was rejected anyway since it did not stabilize for the data set from project 3 functional test failures, making it unnecessary to address the question about the high estimate. The stability threshold was set to 5% of last week's estimate, instead of 10%. With the latter threshold value, generally the models would stabilize earlier, and provide estimates that could give more confidence. For the data sets in this study, the concave models would stabilize earlier, and affect the choice of which estimate to follow in the selection method. This phenomenon is not entirely eligible, shown by the results where the s-shaped models obviously provide estimates with lower relative error. On one occasion, functional test failures from project 2, the G-O model was rejected based on our threshold value for stability. With the threshold value from the original study, the model would have been accepted instead and would have had impact on the choice of which estimate to follow. The high estimate from the G-O model would probably confuse the prediction of the total number of functional test failures to expect, having to consider this value compared to the s-shaped models' estimates. On their own, the s-shaped models were shown to perform rather well in their predictions.

The results of applying the SRGMs on the data sets consisting of functional test failures were unexpectedly positive. Despite the fact that by applying the SRGMs to functional test data instead of system test data, basic assumptions of the SRGMs are violated, the application of the models provided good predictions of how many failures to expect during functional testing. The three data sets were s-shaped and thereby the two s-shaped models fitted very well and resulted in low relative error, while the concave models did not fit very well at all.

4 Conclusions

In this study, we replicated a selection method for software reliability growth models, and applied the method to empirical software failure data. Replicating the original study and comparing the results of the selection method across various types of projects, increases confidence in the results. Hence, the replication study is one step towards generalization of the procedure. In the replication study, the main selection method has been held constant, while other parameters were varied compared to the original study. Parameters such as the threshold values for the evaluation criteria have been changed. In addition, the failure data were obtained from a different development environment. The selection method may be further evaluated by

varying other parameters than the ones discussed in this study, to gain even more confidence in the selection method. One suggestion is to apply the selection method with a different set of software reliability growth models. The decision of which parameters to change when applying the selection method, has to be based on the environment where the application occurs. The method has been shown to perform best while adapted to the circumstances that are specific to each development environment.

Evidently, other selection methods for software reliability models could be used. Fenton and Pfleeger (1997) present several procedures for comparing different models, such as examining the basic assumptions given for each model, degree of bias, and prequential likelihood. The selection method suggested by Stringfellow and Amschler Andrews (2002) consists of several well-defined steps, defining a complete procedure. Its direct applicability to an industrial case motivated our choice of further investigating this selection method.

The results of the replication confirm that the basic ideas of the selection method works well. Generally, the failure detection pattern converges toward values predicted by the models. This projection allows management to stop testing at an appropriate time with confidence in the quality of the developed software. Nevertheless, the selection of which model to base the stop testing decision upon is not always in accordance with the choice suggested by the original study. Indicated by the data sets in this study, by following the conservative choice of trusting the model giving the highest estimate, unnecessary amount of testing might be spent. This replication study thus suggests that the model selection and thereby conclusions from Stringfellow and Amschler Andrews to some extent will depend on the chosen values for stability and GOF, and if there is more than one stable model, the maximum estimate. As shown in this study, these criteria might need to be changed, to support the selection method in an appropriate way and make it applicable in other development environments. Additionally, the selection method works appropriately for selecting SRGMs in a systematic way, and often results in predictions with a low relative error when compared to the actual values of total number of failures. Nevertheless (this is also mentioned by Stringfellow and Amschler Andrews), the selection method should be used with complementary techniques for assessment of the developed software (Stringfellow 2000), and not be trusted alone.

In addition to applying the selection method and its SRGMs on system test data, the procedure was applied to functional test data. The models provided good predictions of the total number of functional test failures to be expected. The results show that the application of SRGMs could be valuable to other types of data, even though several of the underlying assumptions of the models are violated.

Acknowledgment The author would like to thank Prof. Catherine Stringfellow for being generous with her time and willing to answer my questions about the selection method. Thanks also to Prof. Anneliese Amschler Andrews and Prof. Per Runeson who provided valuable comments on the paper.

References

Ehrlich W, Lee S, Molisanim R (1990) Applying reliability measurement: a case study. IEEE Softw 7(2):56–64

Ehrlich W, Prasanna B, Stanpfel J, Wu J (1993) Determining the cost of a stop-test decision. IEEE Softw 10(2):33–42

Fenton NE, Pfleeger SL (1997) Software metrics: a rigorous and practical approach, 2nd edn. PWS Publishing Company, Boston

Fujiwara T, Yamada S (2003) A testing-domain-dependent software reliability growth model for imperfect debugging environment and its evaluation of goodness-of-fit. Elec Commun Jap Part 3 86(1):11–18

Gaudoin O, Yang B, Xie M (2003) A simple goodness-of-fit test for the power-law process, based on the Duane plot. IEEE Trans Reliab 52(1):69–74

Goel AL, Okumoto L (1979) A time dependent error detection rate model for software reliability and other performance measures. IEEE Trans Reliab 28(3):206–211
Huang CY (2005) Cost-reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency. J Syst Softw 77(2):139–155
Institute of Electrical and Electronics Engineers (1990) IEEE standard glossary of software engineering terminology, IEEE Std 610.12-1990
International Standards Organisation (2000) Information technology—software product evaluation—quality characteristics and guidelines for their use, ISO/IEC FDIS 9126-1. Geneva, Switzerland
Jeske DR, Zhang X (2005) Some successful approaches to software reliability modeling in industry. J Syst Softw 74(1):85–99
Kececioglu D (1991) Reliability engineering handbook, vol. 2. Prentice-Hall, Englewood Cliffs, NJ
Lyu MR (ed) (1996) Handbook of software reliability engineering. McGraw-Hill, New York
Miller J (2005) Replicating software engineering experiments: a poisoned chalice or the Holy Grail. Inf Softw Technol 47(4):233–244
Montgomery DC (2001) Design and analysis of experiments, 5th edn. Wiley, New York
Musa J (1999) Software reliability engineering. McGraw-Hill, New York
Musa J, Ackerman A (1989) Quantifying software validation: when to stop testing? IEEE Softw 6(3):19–27
Musa J, Iannino A, Okumoto L (1987) Software reliability measurement, prediction, application. McGraw-Hill, New York
Robson C (2002) Real world research. Blackwell Publishers, UK
Siegel S, Castellan NJ (1988) Nonparametric statistics for the behavioral sciences. McGraw-Hill, Singapore
Stringfellow C (2000) An integrated method for improving testing effectiveness and efficiency. PhD Dissertation, Colorado State University
Stringfellow C, Amschler Andrews A (2002) An empirical method for selecting software reliability growth models. Empir Softw Eng 7(4):319–343
Wood A (1996) Predicting software reliability. IEEE Comput 29(11):69–78
Wood A (1997) Software reliability growth models: assumptions vs. reality. Proceedings of the Eighth International Symposium on Software Reliability Engineering, pp136–141
Yamada S, Ohba M, Osaki S (1983) S-shaped reliability growth modeling for software error detection. IEEE Trans Reliab 32(5):475–478
Yamada S, Ohtera H, Narihisa H (1986) Software reliability growth models with testing effort. IEEE Trans Reliab 35(1):19–23

Dr. Carina Andersson is a research associate at the Department of Communication Systems, Lund University, Sweden. She received a Ph.D. in software engineering from Lund University in 2006 and her M.Sc. in engineering physics with industrial management in 2001. Dr. Andersson's main research interests are software development verification and validation processes and software quality metrics and models.