# Applying Fault Correction Profiles

Norman F. Schneidewind
Naval Postgraduate School
nschneid@nps.navy.mil

## Abstract

*In general, software reliability models have focused on modeling and predicting the failure detection process and have not given equal priority to modeling the fault correction process. However, it is important to address the fault correction process in order to identify the need for process improvements. Process improvements, in turn, will contribute to achieving software reliability goals. We introduce the concept of a fault correction profile -- a set of functions that predict fault correction events as a function of failure detection events. The fault correction profile identifies the need for process improvements and provides information for developing fault correction strategies. Related to the fault correction profile is the goal fault correction profile. This profile represents the fault correction goal against which the achieved fault correction profile can be compared. This comparison motivates the concept of fault correction process instability, and the attributes of instability. Applying these concepts to the NASA Goddard Space Flight Center fault correction process and its data, we demonstrate that the need for process improvement can be identified, and that improvements in process would contribute to meeting product reliability goals.*

## 1. Introduction

Software reliability models have focused on predicting failures and have not given equal priority to predicting faults and fault correction. This situation could result in reliability predictions of NASA missions that are too pessimistic. In addition, modeling tools should be able to track and predict when uncorrected faults would pose a reliability risk to NASA missions. Furthermore, it is important to track the fault correction process in order to identify the need for process improvements. We report on the development of fault correction models and tools to address the above needs, using GSFC satellite project failure and fault data. In future work, the models and tools will be validated against JPL Mission Data System data and Shuttle avionics software data.

If faults remain uncorrected for prolonged periods of time, their presence in the code could pose a reliability risk for NASA missions. This is obviously the case for critical faults, if they are not removed before the mission. In addition, there is a very interesting and subtle characteristic of faults: even faults that are considered to be non-critical could block the discovery of critical faults during testing. For example, the presence of a non-critical fault might cause paths to be executed that do not contain a critical fault, and the critical fault would not be found. Furthermore, fault correction should not be limited to fault identification and removal. Rather, it should also be concerned with *process improvement*, as in the Shuttle PASS development process [KEL97]. In this process, in addition to finding and removing faults, the deficiency in the development process (e.g., requirements analysis, inspections, testing) that allowed the fault to be inserted in the code, is identified and corrected. Thus, product reliability is integrated with process improvement. We adopt this principle in our research approach. That is, our models and tools are aimed at *contributing* to mitigating the risk of not meeting product reliability goals and, at the same time, contributing to process improvement.

This paper consists of the following sections: 2. Objective, 3. Relevance to NASA's Mission and Goals, 4. How the Models and Tools can be Applied by NASA, 5. Research Approach, 6. Data Analysis, 7. Fault Correction Closed Dates Profile, 8. Goal Fault Correction Profile for Closed Dates, 9. Fault Correction Time Profile, 10. Goal Fault Correction Profile for Correction Times, 11. Prediction Accuracy, 12. Application, 13. Model Predictions, 14. Measuring Deviations from Goal Fault Correction Profiles, 15., Summary, and 16. References.

## 2. Objective

Our objective is to develop a model for predicting two quantities: 1) the delay between failure detection and fault correction dates and 2) fault correction times (i.e., durations). The purpose of 1) is to identify whether fault correction dates increase at an increasing rate relative to failure detection dates. The purpose of 2) is to identify whether fault correction times increase at an increasing rate relative to failure detection dates. If either 1) or 2) is the case, it is indicative of a fault correction process that should be examined for *possible*

Proceedings of the 14th International Conference on Electronics, Communications and Computers (CONIELECOMP'04)

improvement. If the examination does lead to improvements in process, this result would contribute to meeting product reliability goals. With these predictions in hand, software engineers can anticipate problems in both the product and process and take corrective action early when the cost of correction is low.

## 3. Relevance to NASA's Mission and Goals

There is a need for greater emphasis on fault correction modeling and prediction in software reliability models [XIE92]. This need stems from the fact that the fault correction process is vital to ensuring high quality software. If we only address failure prediction, reliability assessment will be incomplete because it would not reflect the reliability of the software resulting from fault correction. In addition, it is important to address the fault correction process in order to identify the need for process improvements. Process improvements, in turn, will contribute to achieving software reliability goals -- the well-known observation that process improvement will lead to product improvement [SCH99].

## 4. How the Models and Tools can be Applied by NASA

To address the problem described above, we developed the concept of a *fault correction profile* -- a set of functions that predict fault correction events as a function of failure detection events [SCH03]. The *fault correction profile* identifies the need for process improvements and provides information for developing fault correction strategies. Related to the *fault correction profile*, is the *goal fault correction profile*. This profile represents the fault correction goal against which the achieved *fault correction profile* can be compared. This comparison motivates the concept of *fault correction process instability*, and the attributes of instability. Applying these concepts to the GSFC fault correction process and its data, we *illustrate* how the need for *possible* process improvement can be identified, and that improvements in process would contribute to meeting product reliability goals, and thus reduce reliability risk.

## 5. Research Approach

We consider the fault correction process to be a non-linear system with feed forward path from errors in software artifacts to faults in code to failures in execution. The feed backward path consists of failure detection to fault correction to process improvement to eliminate the errors in the development and maintenance process that cause the errors in the software artifacts. The backward path has an inherent delay due to the time required to correct faults. In addition, an artificial delay may be introduced due to postponement of fault correction because some faults are not high priority at the moment, as in the case of the Space Shuttle [SCH01]. This is a deliberate polity of withholding fault corrections until a later release of the software [MUS99]. These factors contribute to the non-linear nature of the fault correction process. Specifically, this means the following: fault closed dates and fault correction times, relative to the first failure occurrence date, eventually increase at an increasing rate, with the passage of time. Therefore, we modeled the fault correction process with non-linear regression functions, specifically with third order polynomials. These functions increase at an increasing rate, as is the case with the actual data. In addition, these functions allow for local maxima or minima, as is also the case with the actual data. Although, necessarily, specific functions were used in applying our concepts to the GSFC data, the principles of this approach would remain the same independent of the particular domain and set of data used in other applications.

## 6. Data Analysis

The types of data that are available from the GSFC that are relevant to this research are the following:

FD: Failure Date

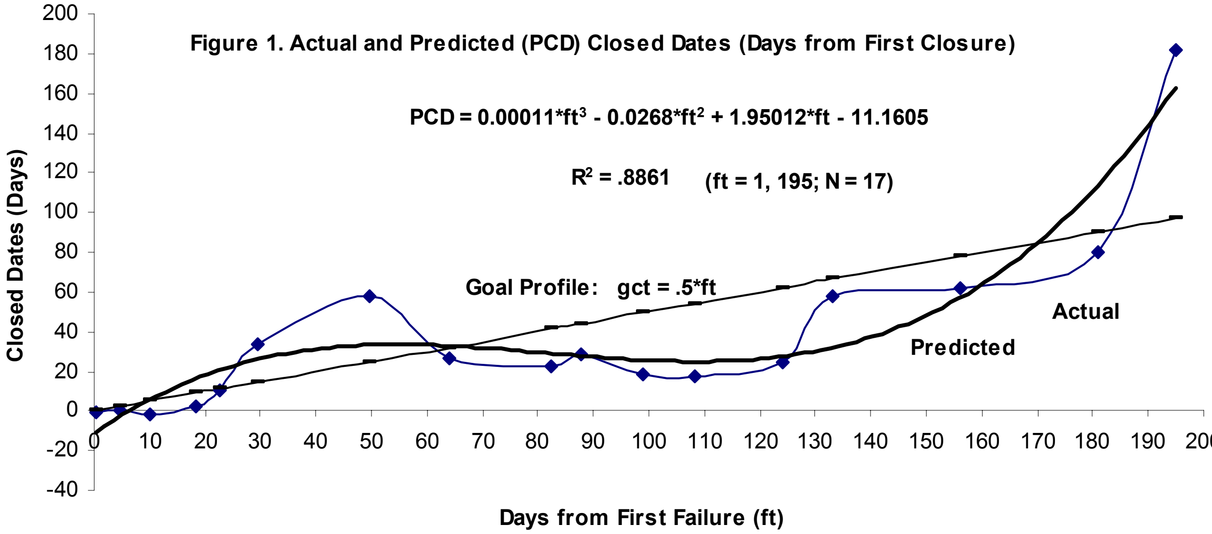
CD: Closed Date

*No fault severity information was available for these data.*

The raw data (i.e., FD and CD) were not meaningful for analysis. However, two transformations on the data allowed us to produce fault correction profiles. First, we averaged the data. This transformation smoothed the data, but the data were still inadequate for deriving prediction functions. Then, we performed a second transformation that involved indexing the data to the dates of the first failure *ft* and the first fault closure. With the second transformation, we were able to produce fault predictor regression functions.

## 7. Fault Correction Closed Dates Profile

In this section, we briefly describe and apply the *fault correction profile* to the GSFC fault correction process. The *fault correction profile* predicts reliability risk metrics (e.g., *fault closed date*) as a function of a failure detection metric (i.e., *failure detection date*).




The *fault correction profile* provides very useful information for developing fault correction strategies. For example, PCD, *predicted closed date*, equation (1), has local maximum and minimum at 55 and 107 days, respectively, (see Table 1 and Figure 1). Figure 1 was obtained by doing *retrospective* prediction, using a sample of size N =17 (see Table 1). The predicted values of 55 and 107 days were obtained by solving equation (2), the derivative of (1) set equal to 0. These values predict when local values of *closed dates* reach a maximum (55 days) and a minimum (107 days). It is the latter that is of major interest because it identifies when the rate of change of PCD (equation (2)) transitions from negative to positive (i.e., increasing rate of change of PCD in the undesirable direction). This is an example of *attribute # 1* of *fault correction process instability*. For the example, as seen in Figure 1, this point is 107 days. *Instability could* be caused by a changing process, product, and personnel.

$$PCD=0.00011*ft^3-0.0268*ft^2+1.95012*ft - 11.1605 \quad (1)$$

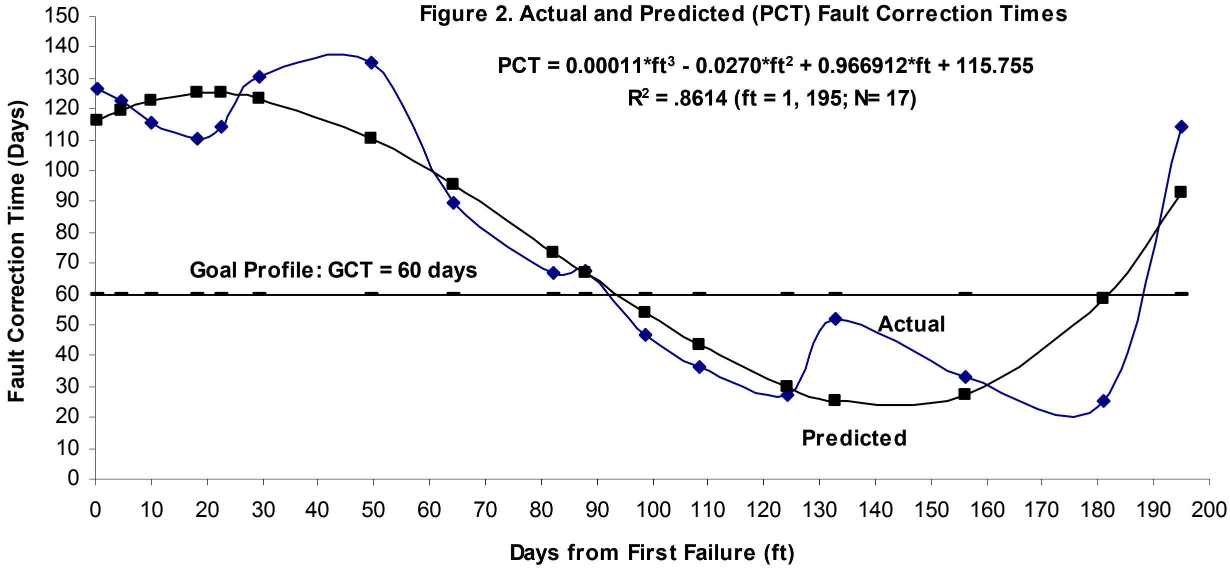
$$d(PCD)/d(ft)=0.00033*ft^2-0.0536*ft+1.95012=0 \quad (2)$$

## 8. Goal Fault Correction Profile for Closed Dates

This profile represents the fault correction goal that the achieved *fault correction profile* can be measured against. For example, suppose the goal is to have all faults corrected within 200 days of the first failure. In addition, suppose another goal is a last *closed date* of 100 days since the first closure. Furthermore, since we have shown that *closed dates* that increase at an increasing rate are undesirable, a linear function of *closed dates*, with a constant rate of change of .5, starting at 0,0 and ending at 200, 100, would be appropriate for representing our goals. This is the *goal fault correction profile*, gct = .5*ft, as shown on Figure 1. When the actual *closed dates* are below the goal profile, the trend is favorable; conversely, when the values are above the line, the trend is unfavorable. We see in Figure 1 that the trend is mostly favorable; however, at *ft* = 195 days, *actual*, the trend has become distinctly unfavorable. This result provides further evidence that the fault correction process warrants investigation. This is an example of *attribute # 2* of *fault correction process* instability. To identify favorable and unfavorable trends, the *goal fault correction profile* can be compared to *PCD*, the predicted *closed dates*, in Figure1. We see that the two are equal at *ft* = 9, 65, and 169 days. Between 9 and 65 days, the actual *closed dates* are mostly above the *gct* line (unfavorable); between 65 and 169 days they are always below the *gct* line (favorable), and at 169 days (see Table 2), they start a rapid increase (unfavorable). These curve crossings are most useful when we are predicting beyond the range of the actual data, when only the predictor functions, like *PCD*, would be available.

## 9. Fault Correction Time Profile

A second *fault correction profile* example is *PCT*, the *predicted correction time* in equation (3), which has a local maximum and a local minimum at 20 and 143 days, respectively, (see Table 1 and Figure 2), obtained by solving equation(4), the derivative of (3) set equal to 0. Figure 2 was obtained by doing *retrospective* prediction, using a sample of size N =17 (see Table 1). The local minimum of 143 days indicates that the plot of equation (3) would continue to rise at an increasing rate in the range *ft* = 143, 195, as shown in Figure 2. This indicates a *possible* unstable fault correction




process, according to *attribute # 1* of *fault correction process instability*, which should be investigated.

$$PCT=0.00011*ft^3-0.0270*ft^2+0.966912*ft+115.755 \quad (3)$$

$$dPCT)/d(ft)=.00033*ft^2-.0540*ft-.966912 = 0 \quad (4)$$

Table 1. Fault Profile Characteristics (Data Set 1, Sample Size N = 17, ft = 1, 195 days)

| Predictor | Units | $R^2$ | Value of *ft* at local minimum | Average Residuals (between predicted and actual) |
|---|---|---|---|---|
| Closed Dates (PCD) | Days from First Closure | .8861 | 107 | -0.5072 |
| Correction Time (PCT) | Days | .8614 | 143 | 0.2592 |
| Ft: Days from First Failure | | | | |

## 10. Goal Fault Correction Profile for Correction Times

As an example of a *Goal Fault Correction Profile*, consider the goal of correcting faults in an average of about two months, or 60 days. (The actual average for the sample size of 17, referred to in Table 1, is 81 days). Then, this condition can be portrayed as in Figure 2, where the *Goal Fault Correction Profile*, *GCT*, intersects *PCT*, the predicted fault correction times, at *ft* = 93 and 181days. We see that between *ft* = 0 and 93 days, the actual and predicted fault correction times are above GTC (unfavorable), and at 181 *predicted* (see Table 2) and 190 *actual* the trend has become distinctly unfavorable. Thus, there is evidence that *attribute # 2* of *fault correction process instability* has taken effect.

## 11. Prediction Accuracy

The $R^2$ values and average residuals for *PCD* and *PCT* are shown in Table 1. The purpose of $R^2$, along with residual plots, is to judge the goodness of fit between the predicted and actual data. The residual plots for *PCD* and *PCT* (not shown) do not show bias over the range *ft* = 1, 195 days.

## 12. Application

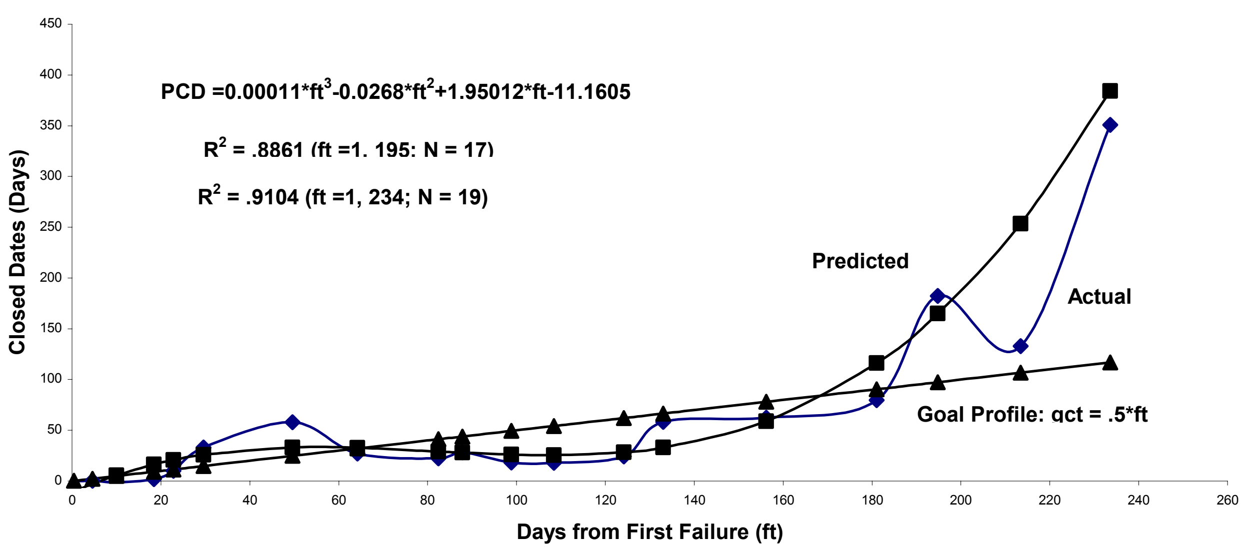
If these profiles hold for other software in the same domain, it could be used as an alarm to investigate the fault correction process for *possible* improvement. This process would be implemented by plotting the actual data points, along with making predictions, for example weekly, as shown in Figures 1 and 2, and observing whether the local maxima or minima occur. If the start of any unfavorable trends are observed, such as the predictions of *ft* = 169 days in Figure 1 and *ft* = 181 days in Figure 2, the fault correction process would be examined to determine the cause of instability. If this examination reveals that there are systemic problems in the process, remedial action is initiated to improve the process. Once sufficient data has been collected, the prediction equations (1) – (4) and their plots can be used to predict unfavorable *future* process events.

## 13. Model Predictions

### 13.1 Closed Date

Up to this point we have described *retrospective* predictions and developed fault correction profiles from them. As mentioned, these are important for sounding alarms concerning the quality of the fault correction process. However, it is also important to predict future fault correction events. With these in hand, software engineers can anticipate fault correction process problems and proactively undertake countermeasures (e.g., strengthen the testing process). We begin by presenting the results obtained by using *PCD*, equation (1), to predict beyond the range of Figure 1, which is a sample of size N = 17, averaged fault *closed dates*, in the range *ft* = 1, 195 days from the first failure. Figure 3 shows the full range of faults, N = 19, averaged fault *closed dates*, in the range *ft* = 1, 234 days from the first failure. Figure 3 compares the actual and predicted fault *closed dates* over the full range of faults. The $R^2$ is .9104 (.8861 for N = 17) and the average residuals are -8.5630 (-0.5072 for N = 17), as summarized in Table 2. The application of *PCD* is to use the fault data in the range 1, 195, to estimate the coefficients of equation (1), and then to predict fault *closed dates* in the range 196, T, where T would be determined by the total time allocated to fault correction from the first failure. In this example, T = 234 days. We observe that *PCD* does predict accurately that at *ft* = 169 days, the *Goal Fault Correction Profile* will be crossed at an increasing rate.

**Figure 3. Actual and Predicted (PCD) Closed Dates (Days from First Closure) Over All Faults**



### 13.2 Fault Correction Time

As in the case of using a sample of N = 17 for *PCD*, the same approach is used for *PCT*, the predicted *fault correction times* in equation (3), as shown in Figure 4. Thus, with a sample of N = 17, in the range *ft* = 1, 195 days, we predict fault correction times in the range *ft* = 196, 234 days. These results are summarized in Table 2, where we see that over the full range of the data, N = 19, the prediction accuracy of *PCT*, as given by $R^2$, is worse than that of *PCD*, but its residuals are lower. We observe that *PCT* does predict accurately that at *ft* = 181 days, the *Goal Fault Correction Profile* will be crossed at an increasing rate. Both predictors, *PCD* and *PCT*, should be used to identify *possible* instability in the fault correction process; we should not rely on a single metric One metric can be used to confirm the result obtained from the other metric.

Table2. Fault Profile Characteristics (Data Set 1, Sample Size N = 19, ft = 1, 234 days)

| **Predictor** | **Units** | $R^2$ | **Value of *ft* where PCD or PCT = Goal Profile at an *Increasing Rate*** | **Average Residuals (between predicted and actual)** |
|---|---|---|---|---|
| Closed Dates (PCD) | Days from First Closure | .9104 | 169 | -8.5630 |
| Correction Time (PCT) | Days | .7316 | 181 | -7.2096 |

ft: Days from First Failure

## 14. Measuring Deviations from Goal Fault Correction Profiles

### 14.1 Closed Date

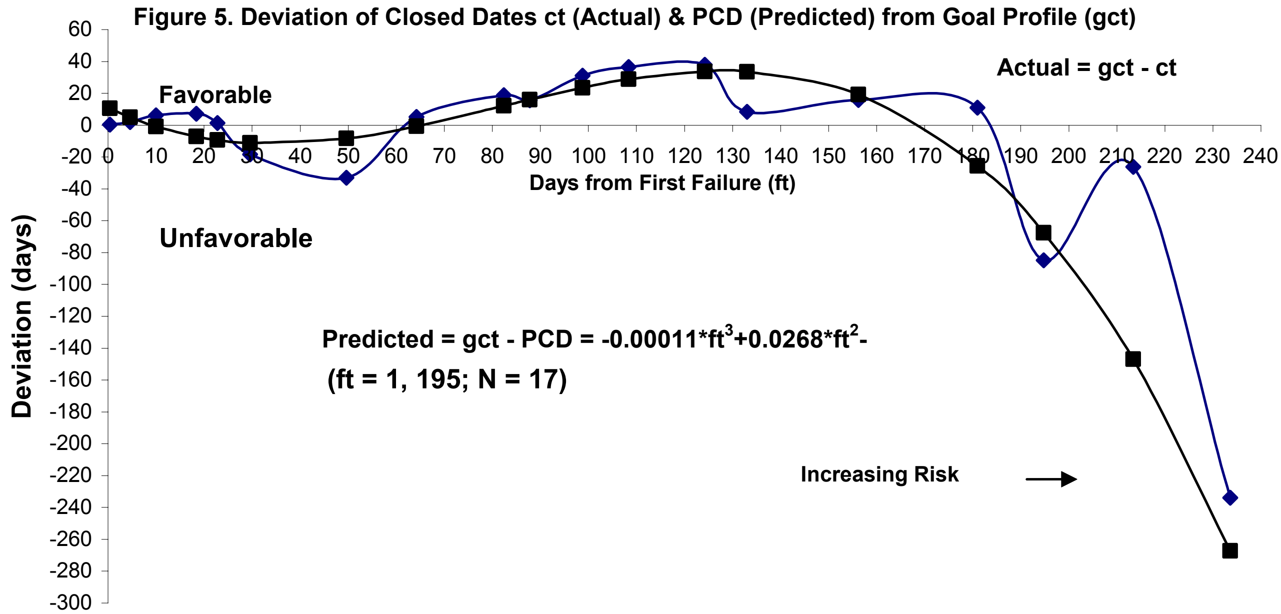
The deviation between the *fault correction profile* and the *goal fault correction profile* is plotted in Figure 5 for *closed date* for actual and predicted deviations. The deviations are computed by equations (5) and (6), respectively.

$$dct = gct - ct = .5*ft - ct \quad (5)$$

$$Pdct=gct-PCD=-0.00011*ft^3+0.0268*ft^2-1.45012*ft+11.1605 \quad (6)$$

The roots of equation (6), where PCD = gct, are *ft* = 9 days, 65 days, and 169 days, as shown in Figure 1, where PCD = gct, and in Figure 5, where gct – PCD crosses the X-axis. We see that at *ft* = 169 days, *predicted* (see Table 2), and 183 days, *actual* (approximate), the deviations go negative (unfavorable); the deviations and the risk start to increase significantly. Of course, since the prediction range is *ft* = 1, 195 days, this would be a *retrospective* prediction. However, the *prospective* prediction in the range *ft* = 196, 234 days, as seen in Figure 5, suggests a highly unstable fault correction process (i.e., *attribute# 1* of *fault correction process instability*). Therefore, this prediction result calls for an investigation of the cause(s) of a rapidly deteriorating fault correction process.

In addition to the values of *ft* where PCD = gct, we want to predict the values of *ft* in equation (6) where a local minimum and a local maximum occur. This is accomplished in equation (7) and shown in Figure 5, where the roots are 34 days (minimum) and 128 days (maximum). The latter value, corresponding to a predicted local maximum, confirms the fact, *retrospectively*, that the X-axis crossing at *ft* = 169 days, is indeed the start of the deviation going negative.




$d(Pdct)/d(ft) = -.00033*ft^2 + .0526*ft - 1.45012 = 0$ (7)

### 14.2 Fault Correction Time

The deviations between the *fault correction profile* and the *goal fault correction profile* are plotted in Figure 6 for *fault correction time* for actual and predicted deviations. The deviations are computed by equations (8) and (9), respectively.

DCT = GCT – CT = 60 – CT (8)

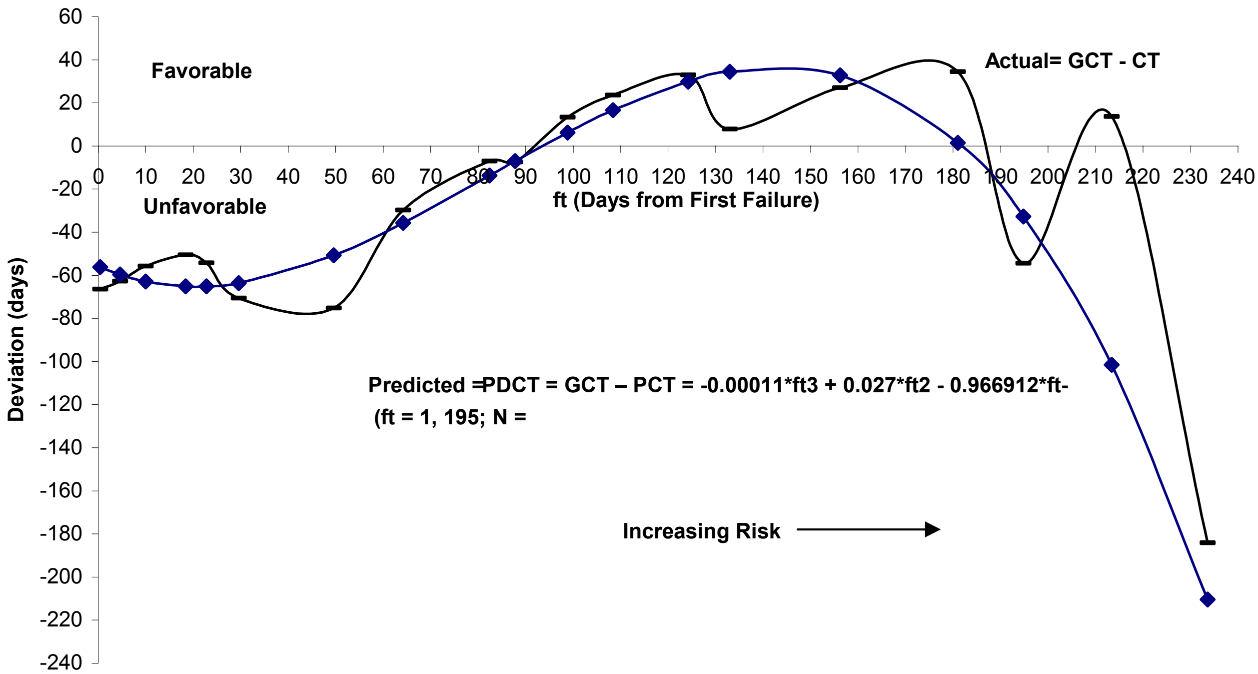
$PDCT=GCT-PCT=-0.00011*ft^3+0.027*ft^2 - 0.966912*ft- 55.755$ (9)

The positive roots of equation (8), where PCT = GCT, are *ft* = 93 days and 181 days, as shown in Figure 2, where PCT = GCT, and in Figure 6, where GCT – PCT crosses the X-axis. We see that at *ft* = 181 days, *predicted* (see Table 2), and 187 days, *actual* (approximate), the deviations go negative (unfavorable); the deviations and the risk start to increase significantly. Again, this would be a *retrospective* prediction in the range *ft* = 1, 195 days. Figure 6 shows a *prospective* prediction of a highly unstable fault correction process in the range *ft* = 196, 234 days.

In addition to the values of *ft* where PCT = GCT, we want to predict the values of *ft* in equation (9) where a local minimum and a local maximum occur. This is accomplished in equation (10) and shown in Figure 6, where the roots are 20 days (minimum) and 143 days (maximum). The latter value, corresponding to a predicted local maximum, confirms the fact, *retrospectively*, that the X-axis crossing at *ft* = 181 days, is indeed the start of the deviation going negative

$d(PDCT)/d(ft)=-0.00033*ft^2+0.054*ft-0.966912=0$ (10)

**Figure 6. Deviation of Fault Correction Times CT (Actual) & PCT (Predicted) from Goal Profile (GCT)**




## 15. Summary

We introduced the concepts of fault *correction profile*, *goal fault correction profile*, *fault correction process instability*, and the attributes of instability. The development of these concepts was motivated by the need to improve the fault correction process, with the objective of improving product reliability. We applied these concepts to the NASA Goddard Space Flight Center (GSFC) fault correction process and its data and we demonstrated the feasibility of identifying *possible* problems in the fault correction process. Although other application domains may yield *fault correction profile* functions that are different than the ones we identified using the GSFC data, the *principles* of our approach would remain the same.

## 16. References

[KEL97] Ted Keller and Norman F. Schneidewind, "Successful Application of Software Reliability Engineering for the NASA Space Shuttle", Software Reliability Engineering Case Studies, International Symposium on Software Reliability Engineering, November 3, Albuquerque, New Mexico, November 4, 1997, pp. 71-82.

[MUS99] John D. Musa, Software Reliability Engineering: More Reliable Software, Faster Development and Testing, McGraw-Hill, New York, 1999.

[SCH99] Norman F. Schneidewind, "Measuring and Evaluating Maintenance Process Using Reliability, Risk, and Test Metrics", IEEE Transactions on Software Engineering, Vol. 25, No. 6, November/December 1999, pp. 768-781.

[SCH01] Norman F. Schneidewind; "Modeling the Fault Correction Process", Proceedings of The Twelfth International Symposium on Software Reliability Engineering, IEEE Computer Society, Hong Kong, 27-30 November, 2001, pp. 185-190.

[SCH03] Norman F. Schneidewind, "Fault Correction Profiles", Proceedings of the Fourteenth International Symposium on Software Reliability Engineering, Denver, CO, 17-20 November, 2003, pp. 257-267.

[XIE92] Min Xie and M. Zhao, "The Schneidewind Software Reliability Model Revisited", Proceedings of the Third International Symposium on Software Reliability Engineering, IEEE Computer Society Press, Los Alamitos, CA, Research Triangle Park, NC, October 7-10, 1992, pp 184-192.