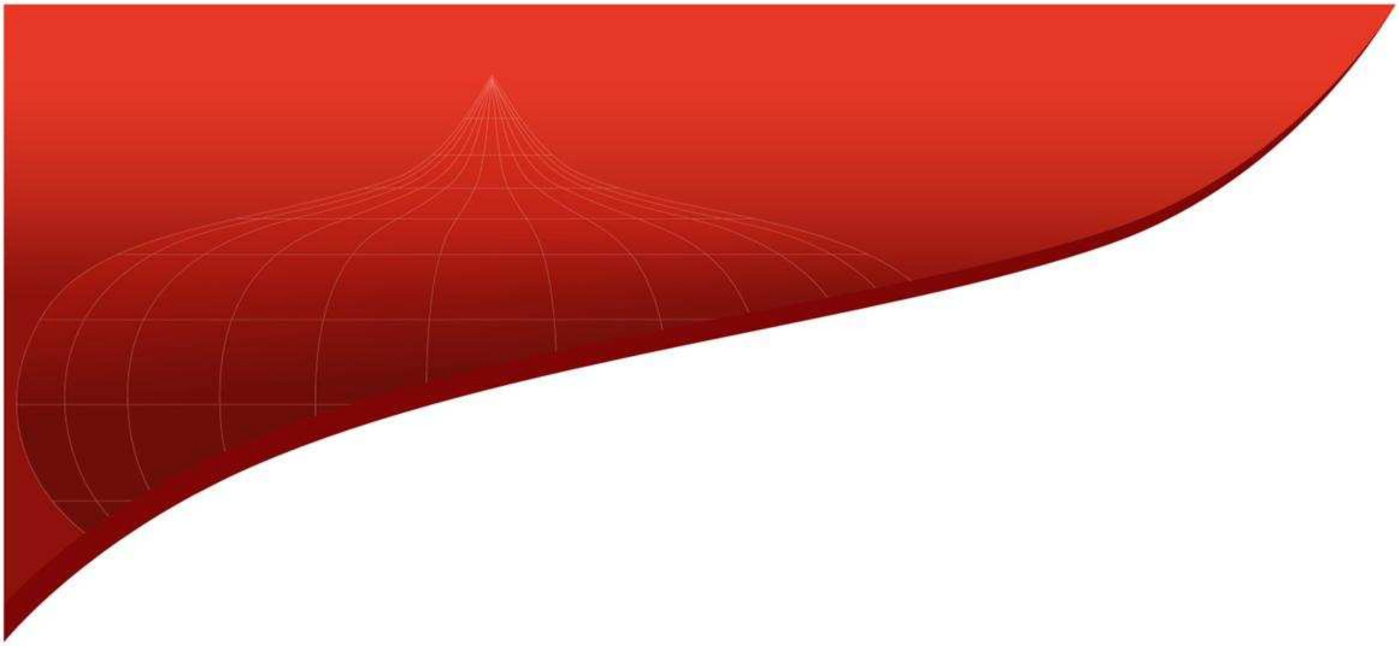


Testing expensive?
Not testing is more expensive!

Determine the test cost optimum with simple metrics

authors: Leo van der Aalst en Corné de Koning
based on the original publication in: Informatie

TESTING EXPENSIVE?
NOT TESTING IS MORE EXPENSIVE!

Determine the test cost optimum with simple metrics

authors: Leo van der Aalst en Corné de Koning
based on the original publication in: Informatie

The question "what do you think of the test process?", is frequently answered by "expensive!". However, the value of this answer is often limited, since it is given instinctively without supporting figures. "Testing is expensive". Yes, this is true if only the test costs and not the benefits are taken into consideration. Test costs are based among others on the costs of the test infrastructure, the number of hours spent by the testers and their rates. The quality of the system development process plays a role as well. If this quality is low, more defects are present. As a result, the test process should be more thorough with a higher coverage ratio in order to detect these defects. Defects occurring during production may lead to consequential damage and higher repair costs. The prevented consequential damage and repair costs form the test benefits. The dilemma that is related to this, is: do I need to continue testing until all defects have been removed, before I allow the application to be released? Apart from the fact that it is impossible to detect all defects in time, it is quite reasonable that detecting a defect in advance is more expensive than letting it occur during production. Thus, a well-considered decision should be made.

"Testing expensive? Not testing is more expensive!" The title of this article suggests that a cost optimum could be found. [Juran, 1988] shows that an optimum in total quality costs exists. In this article we will demonstrate that this optimum can be calculated by using simple metrics, without the need for historical data. In contrast to Juran we do not display the costs against the quality level, but against time. This way, the exact moment can be determined on which continuation of testing is no longer (cost) effective. On this moment (the *cost optimum*), not all defects of the system have been detected, but from a cost viewpoint it is *the* moment to stop testing and start the production.[1]

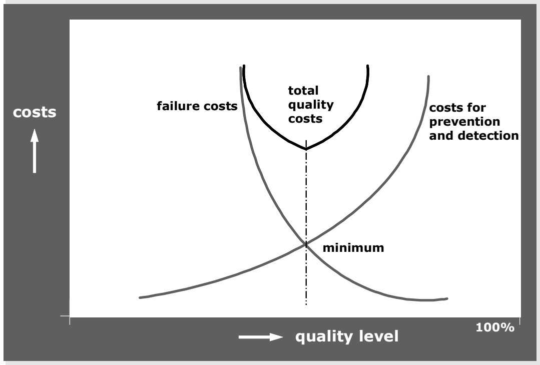
IN MORE DETAIL

Quality management costs

Taking measures in the scope of quality management costs money. The total of these different kinds of costs is called quality costs.
Quality costs consists of:

- Prevention costs
  costs for taking preventive measures
- Detection costs
  costs for taking detective measures
- Failure costs
  costs for taking corrective measures or costs as a result of insufficient quality (lost income, extra service costs, guarantee claims, damage claims)

The figure below displays the relationship between the different kinds of costs.



An important message Juran gives us is that pursuing perfect quality is not cost-effective, because eventually each small increase in quality requires a large investment.

DETERMINING THE TEST COST OPTIMUM

To determine the optimum, several data are necessary. These data are illustrated in graphs instead of number series, since graphs are more clarifying in practice.[2]
The figures are generated in three steps:
1. calculating test costs
2. calculating prevented damage
3. determining cost optimum.

Calculating test costs

Test costs are built up from several costs. Beside staff costs, among others costs for infrastructure, test tools, management and renting office space should be taken into consideration as well. It is not easy to calculate the test costs this way. In practice, multiplying the number of test hours spent with the hourly rate appears to be a good alternative. An additional advantage is that the legibility of the figures increases.
Expressed as a formula, the calculation of the test costs looks as follows:
Test costs = number of test hours spent x hourly rate

With the number of test hours spent are not only meant the test execution hours, but also hours spent during the other phases of the test process [Aalst, 2006]: planning, control, setting up and maintaining infrastructure, preparation, specification and completion.

In figure 1 “Test costs”, the test costs are illustrated. The above mentioned formula has been used, with the assumption that the test team is characterized by a constant occupation. This means that the line representing the test costs crosses the point of intersection of the cost and time axis and is drawn as a straight line.

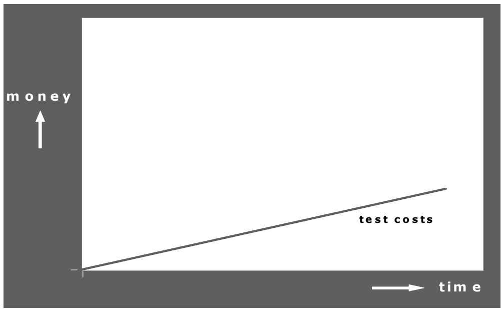


Figure 1: Test costs

Calculating prevented damage

The prevented damage is the sum of the potential consequential damage of all defects detected. Detecting a defect is not a problem: it is directly derived from the test process. Determining the prevented damage per defect, if this would occur in production, is harder. In practice, it appears that still a reasonably reliable answer can be given here: in a selected group, consisting of a developer (analyst), a user representative and a representative of the calculation centre, the defects are assessed on prevented damage one by one. Each representative argues from his own responsibility. The developer gives an estimation of the additional repair costs required if the defect would be detected during production instead of during the test process. The user indicates the expected damage resulting from lost income, compensation claims, waiting time users, et cetera. The calculating centre specifies among others the expected damage which is the result of performance loss and restore activities.

EXAMPLE

For a bank, a 'real-time' fraud detection system (FDS) is tested. This system determines, based on historical data, whether an executed transaction is a possible fraud case. During the test execution it is observed that the fraud score (the higher the score, the larger the chance that it concerns a fraud case) that corresponds to a certain transaction, is not removed before determining the fraud score of a next transaction, but is added up to it. The developer estimates the additional repair costs, for example the making of a conversion program, at 1000 euro. The user observes that too many transactions are labelled as fraud sensitive compared to reality. This implies extra effort for the users of the FDS, which is estimated at 5000 euro. The calculating centre estimates installation of the new program and running the conversion program at 500 euro. Thus, the total estimated consequential damage amounts 6500 euro.

To estimate the prevented damage for each defect like for the FDS has been done, is a labour-intensive method. We have chosen to make an estimation of the *average* prevented damage amount per defect. The consequential damage per defect may differ considerably: one leads to a simple cosmetic screen adaptation, whereas the other, none of the users has access to the system. To determine the average prevented damage amount, a weighting per defect has been used. The used weighing factors are:

- cause (test basis: 3, environment: 2, test object: 1)[3]
- severity (heavy: 8, moderate: 6, light: 2, wish: 1)
- quality attribute (continuity: 10, functionality: 8, security: 7, performance: 5, user-friendliness: 1, verifiability: 1).

When designating the weighing factors, the developer describes the cause of the defect, the user assesses its severity, and the developer, user and calculating centre together assign its quality attribute (the property of the information system related to the defect).

EXAMPLE

The developer describes that the defect described in the previous example has been caused by a defect in the test object. The user assesses its severity as moderate, and the developer, user and calculation centre together assign the defect to the quality attribute functionality. Thus, the total number of weighing points is 1 x 6 x 8 = 48.

For some of the defects it is determined as accurately as possible, what the consequential damage in money would be. This amount is then divided by the number of designated weighing points, as a result of which a damage amount per weighing point is determined. To calculate the consequential damage of the defects implies as from this moment that the total number of weighing points per defect is determined and then multiplied with the damage amount per weighing point.

EXAMPLE

The consequential damage of defect 1 is estimated at 6500 euro. The cause is the test object, severity is moderate and quality attribute is functionality.
Total number of weighing points is 1 x 6 x 8 = 48.
The consequential damage of defect 2 is estimated at 1000 euro. The cause is the test basis, severity is light and quality attribute is user-friendliness.
Total number of weighing points is 3 x 2 x 1 = 6.
The consequential damage of defect 3 is estimated at 12.000 euro. The cause is the test basis, severity is moderate and quality attribute is performance.
Total number of weighing points is 3 x 6 x 5 = 90.
The total amount of estimated consequential damage is: 19.500 euro. The total number of weighing points is: 144. Thus, the damage amount per weighing point is 135 euro.
If a new defect 'scores' 60 weighing points, then the damage amount is estimated at 8100 euro (60 x 135 euro).

In figure 2 "Prevented damage", the sum of the estimated consequential damage of all defects is displayed against the time at which these defects have been detected. Since these defects have been detected before the system is taken into production, we speak of prevented damage.

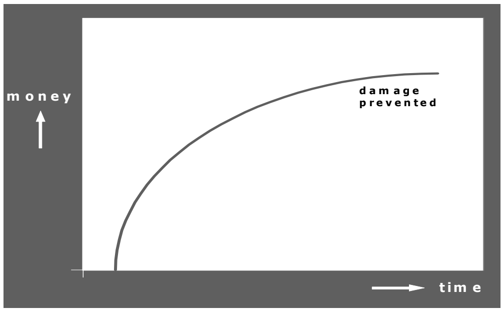


Figure 2: Damage prevented

The prevented damage curve in figure 2 does not start at time 0, since the test process does not result immediately in defects. In the Planning phase, no defects are detected. It is not until the Preparation and Specification phases that defects are detected, but relatively most of the defects are detected during the test execution. In figure 2 it is assumed that all defects are detected during the test execution. A well-executed risk analysis and test strategy development leads to a steep curve in the beginning. The aim of the test strategy is to detect as *early* as possible the *most important* defects at the *lowest* costs (after all, the most important defects cause the largest consequential damage and by detecting these during the test process the largest damage is prevented). The shape of the curve is also determined since at the beginning of the test process more defects are detected compared to when the process progresses.

Determining cost optimum

Determining the net profit of the test process (see figure 3) is now a relatively simple exercise. By deducting the test costs from the prevented damage, the net profit curve arises. The test cost optimum is indicated in the figure with a vertical dotted line.

The time (t) on which the optimum is reached, is also the time at which can be considered to stop the test process. After all, *testing should be continued as long as the costs of defect and repairing a defect are lower than the costs related to the occurrence of that defect during production.*

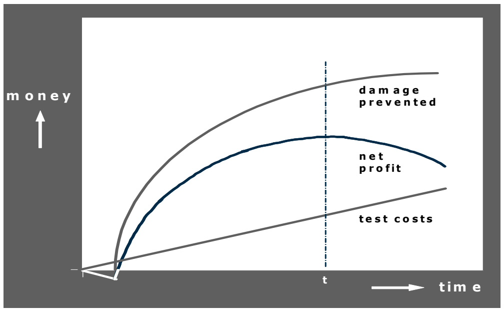


Figure 3: Net profit of the test process

In case testing is stopped before the optimum, it can be concluded that this is too early: the maximum net profit is not yet reached. On the other hand, if testing is continued after the optimum, the net profit decreases. It is nearly impossible to stop testing at the exact moment the optimum is reached. To stop testing too early can have large consequences. For that reason, it is recommended to stop testing as soon as it is clear that the net profit curve is actually decreasing. The loss suffered here compared to stop testing at the exact moment the optimum is reached, is preferred above the risks that are taken if testing is stopped too early.

POTENTIAL FAILURE COSTS

The prevented damage is derived directly from the defects. However, it is incorrect to assume that if the prevented damage curve starts to run horizontally – i.e. as no more defects are detected – all defects in the system have been detected. Perhaps during the strategy development it is wrongfully assumed that certain functions should be tested with a lower coverage ratio, or even not all. In that case, it is possible that still a lot of defects will be detected during production.

In fact, only a percentage of all existing defects are detected during the test process. The sum of the defects detected during the test process and the defects occurred during production is the total number of present defects of the system. Possible present defects that do not occur or are not able to occur during production are not taken into consideration as a defect here.

Why is it interesting to know how many defects are present in the system? If 'only' the prevented damage is known, no judgment can be made concerning the proportion of prevented damage in relation to potential failure costs. This is in fact a measure of the quality of the test process and strategy. The potential failure costs are calculated by estimating the damage for all defects detected during the test process (i.e. the maximum prevented damage). Subsequently the damage that occurred during the first three months production is added up. Thus, the potential failure costs are not known before *three months production*. This can be too late to modify the test strategy of the project, if necessary. However, these data are useful to estimate potential failure costs for future projects.

In figures 4a and 4b, the relation between the prevented damage and potential failure costs is clarified.
In figure 4a can be seen that the prevented damage approaches the potential failure costs. In this case, it is not expected that there is still much more prevented damage present in the system. The still present damage amounts up to the difference between the potential failure costs and the prevented damage. In other words, the net profit is an accurate reflection of reality and it is a correct choice to stop at the exact moment the optimum is reached.
In figure 4b it is illustrated that the potential failure costs are considerably higher than the prevented damage. The difference between the potential failure costs and the

prevented damage is indicative for the damage that is still present in the system. With another test strategy it might be possible to approach the potential failure costs with the prevented damage. This means that the net profit may end up higher (as it is defined as the difference between the prevented damage and test costs).

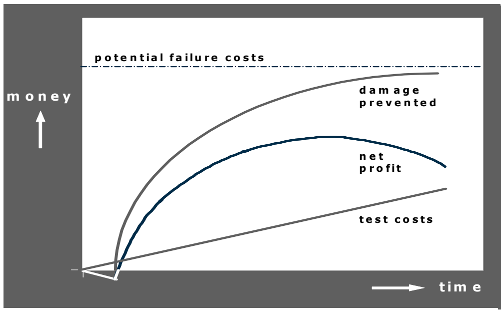


Figure 4a: Good quality test process

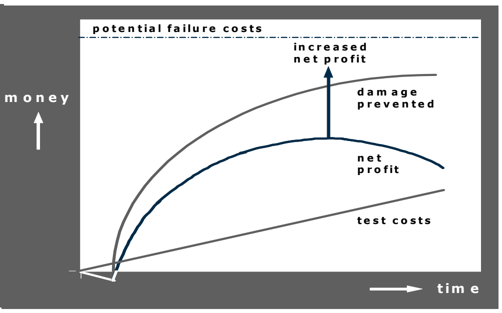


Figure 4b: Poor quality test process

Recommendations

The potential failure costs are not known in advance. So, the difference between the prevented damage and the potential failure costs is unknown as well. For this reason the choice of the moment on which testing should be stopped can not be based on the difference between data.

Our recommendation is: *do not stop testing if the net profit still shows a rising line. It is better to stop if the net profit decreases for some time.* This way, the chance that the system still contains many defects (consequential damage) has become smaller.

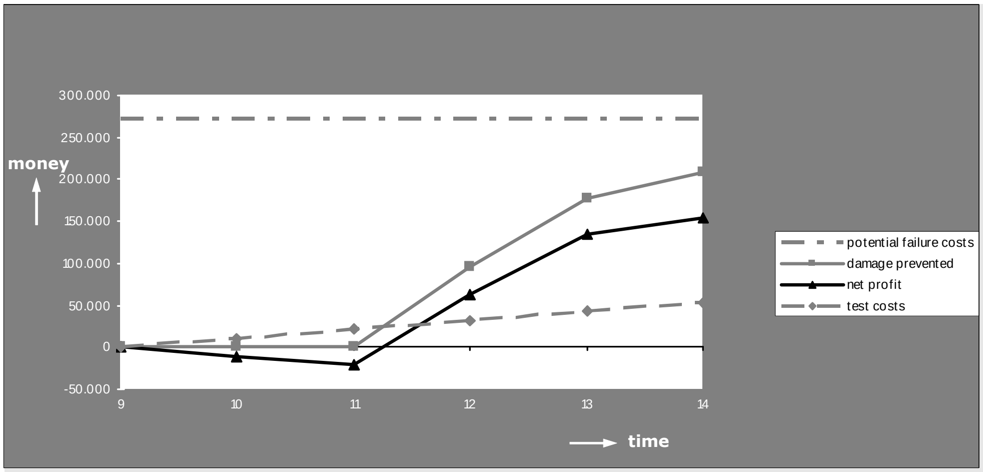
In the frameworks on the following pages we give some real-world examples whereby the method described in this article is used. In spite of the fact that only afterwards can be determined if the moment to stop testing has been correct (after three months production), this approach leads to satisfying results in practice.

Real-world examples

The information from these examples were taken from a test process in an administrative department from a large financial institution.

Example 1: stopped too early?
De net profit still showed an increasing line at the time testing was stopped (week 14). This would imply that testing was stopped too early. During the first three monts production it turned out that the damage totalled 65.000 euro. In other words the net profit could have been a maximum of 42 percent higher (220.000 euro in stead of 155.000 euro). Not having taken into acount the testing costs which would have been necessary to find these production failures during the test process.

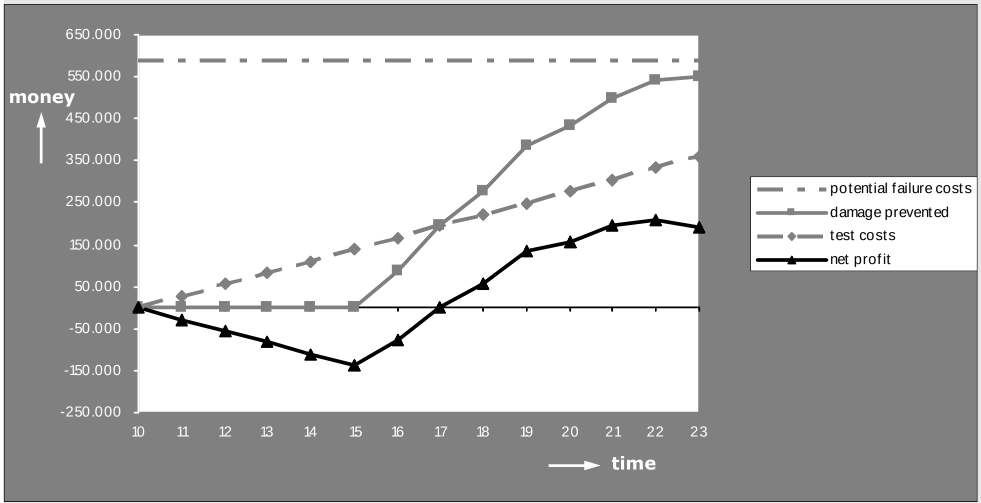
Conclusion:
First instance suggested testing was stopped too early. This was confirmed after three months production: the net profit could have been higher.



example 1

Example 2: stopped in time?
The net profit showed a slight decreasing line at the time (week 23) testing was stopped. This would imply that testing was stopped at the right time. During the three months production it turned out that the damage totalled 30.000 euro. In other words, the net profit could have been a maximum of 15 percent higher (220.000 euro in stead of 190.000 euro). Not having taken into acount the testing costs which would have been necessary to find these production failures during the test process.

Conclusion:
First instance suggested testing was stopped at the right time. This was confirmed after three months of production.



example 2

References

- [Juran, 1988]
  Juran, J.M. (1988), *Juran's Quality Control Handbook*, McGraw-Hill, ISBN 0-070-33176-6
- [Aalst, 2006]
  Aalst, L. van der, Broekman, B., Koomen, T., Vroon, M. (2006), *TMap Next, for result-driven testing*, 's-Hertogenbosch: Tutein Nolthenius Publishers, ISBN 90-72194-80-2

[1] Of course it would be nicer if the moment to stop testing could be determined before the project starts. However, for this, historical data is needed and several assumptions should be made. This means that creating such a 'forecast' is a complicated and environmental-dependent activity, as a result of which the application of this method is limited. In this article the registrative approach is described, the 'predictive' part has been disregarded.

[2] In order to get useful results, a good hour and defect registration has appeared essential. If this is inadequately arranged in your organisation or project, we dissuade you to base your decision-making with regard to the moment to stop testing, on the described figures.

[3] Defects in the test basis are defects which (if they were not detected) lead to defects in the software. Correcting such defects involve high repair costs. Defects in the test object are considered as 'normal'. Defects in the environment are frequently harder to solve, therefore they get the value 2.