1. Abstract

This paper presents a practical procedure for managing sampling in production testing of electronics. The focus is in functional testing which is often the most expensive and time-consuming test operation. Procedure utilizes Continuous Sampling Plan 1 (CSP-1) effective in an environment where you have to manage several tests and also keep the quality risks at selected level.

2. Introduction

The pressure of decreasing test costs exists all the time. Most companies have already tuned the number of tests in production to level where they cannot be reduced more without increased quality risks. Especially in high end electronics like telecom and aerospace tens or hundreds of various tests and measurements are run for every product to keep the quality of delivered products on high level.

One of the most potential next steps in fight against test costs is sampling. Usually it has been used widely for checking the quality of incoming material or doing the visual inspection in production and there are standards and practices available for that. Those practices are not the most suitable in high volume continuous production and in functional testing where you do not collect lots and want to manage sampling of several tests simultaneously.

In this approach initial coverage of tests is supposed to be adequate. The quality risk of the presented sampling procedure is compared to original situation: what is the maximum change in outgoing quality. We do not concentrate here in analyzing quality of original situation with 100% testing.

Paper presents first the standard Continuous Sampling Plan 1 (CSP-1) procedure (Dodge, 1943) which provides the basic mathematical rationale for the presented method. After that the background for interpreting fail -information when several tests are sampled is presented. Finally the rewritten CSP-1 procedure for functional testing is illustrated.

3. Continuous Sampling Plans

The first of the continuous sampling plans, CSP-1, was introduced by Dodge in 1943. After CSP-1 several other alternatives such as CSP-2, CSP-3, CSP-M and CSP-C has been developed. CSP-1 is the simplest and probably most used Continuous Sampling Plan. It is presented in numerous quality management textbooks and also in MIL-STD-1235C [2]. The procedure is following:

Use 100% inspection at the start. When i successive units are found to be acceptable and when there is assurance that the process is producing homogenous product, 100% inspection is discontinued and sampling is instituted to the extent of a fraction f of the units. The sampling is continued until a defective unit is found. One hundred percent inspection is then reinstated and the procedure is repeated [1].

Figure 1 presents the flow of the plan.

The parameters i and f are depending on selected AOQL, which characterizes the accepted risk. AOQL stands for Average Outgoing Quality Limit and it defines the worst, or “limit” of average quality of outgoing product [1]. Values for parameters can be found for from various sources [1, 2, 3].

The procedure where changing back to 100% inspection immediately after just one non-conforming sample is often seen too harmful from
inspection capacity point of view. When process quality or stability is not in good level the problem of oscillating between sampling and 100% inspection can be a problem, which affects to Average Fraction Inspected (AFI). Optimizing AFI with CSP-1 is often seen problematic and can force some users to more complicated plans [3]. Also short-run or low volume production can make AFI high and solutions such as improved short-run CSP-1 has been presented [4].

High AFI increases need for test capacity and decreases benefits of sampling. Therefore quality of products coming to sampled test operation as well as stability of manufacturing process has great impact on success of CSP-1 in practice. Most of the manufacturing defects should be screened and repaired before functional testing. This means using efficient process testing combined of automatic optical inspection (AOI), automatic XRAY inspection (AXI) and/or InCircuit testing. What technologies are used depends a lot of manufacturing process quality and complexity of the products. The process test operations have to be managed using statistical process control (SPC). This ensures that when manufacturing process goes to unstable state it is stopped and re-adjusted immediately.

Successful utilization of CSP-1 requires that there are tests, which do not fail often and are therefore good candidates for sampled testing. According to some studies detection of faults concentrates usually in certain tests and many tests actually fail very seldom [5].

Even though CSP-1 may not be the best plan for all cases, the simplicity makes it primary alternative. The straightforward procedure is great benefit in complex environment like this: managing sampling of several tests while completely controlling the quality risks.

To describe the problem when managing sampling more than one test we assume that AOQL level 0.05 % is selected. Based on CSP-1 plan we can identify 10 tests, which can be moved from 100% inspection to sampled testing. Each of the 10 tests has now AOQL of 0.05% but the maximum total change in outgoing quality is not 0.05 %. Because each of ten test represents possibility of 0.05% failed products we have total AOQL:

\[ 1 - (1 - 0.0005)^{10} = 0.005 = 0.5\% \]

This is maximum value and does not necessarily come true but it is the limit, which can be guaranteed.

The cumulative AOQL can be managed by setting AOQL of each test much lower. The disadvantage of that approach is that it gets high value, which causes long period of 100% testing before sampling can be started. More efficient is to use grouping of tests. Grouping makes CSP-1 more efficient in environment running multiple tests and it is described in chapter 5.

4. What tests to sample

4.1 Reasons to do testing

Reasons to make some certain test for a manufactured product are various: technical risk, customer requirements, previous experiences, and statistical results to name some of them. Sometimes argumentation is rather unjustified like “to be sure” or “this way it has always been done”.

When thinking the existing list of tests for a product and trying to understand if all the tests are actually needed, we can identify following reasons for running a test:

1. Test indicates if product is defected
2. Test is done for legal reasons or for customer’s purposes
3. Test includes obligatory tuning, setup or other necessary action
4. Test PASS/FAIL information supports faultfinding
5. Test result supports faultfinding
6. Test result gives statistical information of process or product quality

When analyzing the importance to run any tests based on reasons 1 to 6 above, we can notice:

*Reason 1* is important reason and has effect on outgoing quality.

*Reasons 2 and 3* are unavoidable, those test must be run 100%.

*Reasons 4 and 5* are good reasons to run a test but those tests can be done for defected products only

*Reason 6* is good reason to run a test but statistical information can be collected also from samples.

We make a decision that tests having reason 2 or 3 are not sampled and they are always tested 100%. From other tests we should run only tests that indicates whether product is defected. We just have to know if test is important for detecting defects. When a product fails in a tester there may be several tests giving FAIL result. This does not mean that all the tests are needed to indicate the defect: one of them is enough to record product’s non-conformance.
4.2 Experimental defect coverage

As a simple example we use a test operation with 5 tests in order a, b, c, d and e. Tests are functional tests checking the operation of the product by testing some values or outputs: power consumption, output signal level or clock frequency to name some examples.

When using sampling plan (CSP-1) we have to keep track of failed and passed tests. It is important that for every single defect only one of the tests is marked failed in this sense. The simplest way is to stop testing to first failing result or alternatively log first failing test in the queue faulty. For example when product is totally dead the only important test indicating this fault is the first failing test. As mentioned earlier, results of other tests may be useful for faultfinding and rework, but those can be run also separate for that defected unit.

As an example of this approach we may have statistics of 10,000 successive pieces of Product A (Table 1).

<table>
<thead>
<tr>
<th>TEST</th>
<th>FAILED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test a</td>
<td>298</td>
</tr>
<tr>
<td>Test b</td>
<td>0</td>
</tr>
<tr>
<td>Test c</td>
<td>150</td>
</tr>
<tr>
<td>Test d</td>
<td>5</td>
</tr>
<tr>
<td>Test e</td>
<td>107</td>
</tr>
</tbody>
</table>

*Table 1. Example of failed tests*

Defect coverage of all tests for Product A is something like presented in Figure 1: some tests cover more than others and there may be some overlapping. It is also possible that some of the defects are covered by none of the tests, but that problem is not in focus of this paper.

If we look again Test b in Table 1 and Figure 2, there is two possible reasons for zero defects in 10 000 products:

A  Test a which is before Test b covered same defects

B  There wasn’t defects that Test b only covers

The truth is probably a combination of the reasons but that is something we do not have to know. We can see that Test b is not very useful. Quality of outgoing products would not have decreased if we had done Test b on sample basis only. The mathematics behind sampling plan will take care of the amount of risks even though some new kind of defects appears.

5. Adjusting the plan

5.1 CSP-1 plan for grouped tests

Chapter 3 brought up the problem with AOQL level when several tests are sampled. Solution, which does not increase the amount of successive units needing 100% testing, comes from grouping the tests. In practice CSP-1 is applied a bit different than described earlier (Figure 1). The new procedure, which is presented in Figure 3, fulfills completely the statistical background of CSP-1.

The quality risk, AOQL, is selected first and in practice it is good to continue with selecting the target for sampling rate (fraction \( f \)). The CSP-1 tables gives then the third variable, \( i \). If we select AOQL of 0.05% and want to do the sampling with rate 10%, we must test 2200 consecutive units to start the sampling.

First we do the 100% testing for \( i \) units. Test not detecting defects (as described in chapter 4) in \( i \) successive units are identified as tests, which could be sampled: \( T_{S1}…T_{Sn} \). Now we make a group \( T_S \) of tests \( T_{S1}…T_{Sn} \). This group is a “virtual” test, which we are going to run sampled. \( T_S \) has passed \( i \) successive tests runs and therefore it can be sampled according to CSP-1 plan.

We do not know what is the quality risk of each of the sampled tests \( T_{S1}…T_{Sn} \) but we know that together they present no more than selected change in AOQL.

Now we can start sampling this group of tests (\( T_S \)). The group has to be kept together by synchronizing the sampled tests to same units. We cannot add tests to sampled group without starting procedure from the beginning because AOQL cannot be guaranteed to remain in selected level.
Returning to 100% testing

Just like in CSP-1 plan, one defected test causes changing back to 100% testing. Here this means changing only the test, which detected the defect. The failed test is dropped out of the sampled group $T_S$ and returned to 100% testing. This is possible because we can track back the whole previous history. We can notice that dropping that one test out of the $T_S$ group is not different from the situation that we didn’t take that test to $T_S$ group from the first beginning.

Returning the failed test back to sampled group is more complicated. If we want to keep the selected AOQL, the only guaranteed possibility is to do all test 100% for $i$ units again. You can also set a condition to control if sampling has changed too inefficient because of returned tests. Setting a limit $s$ can control this. When this limit has been exceeded, whole procedure starts from the beginning with 100% testing for $i$ units. All this is part of fine-tuning your sampling procedure to minimize amount of testing.

The whole procedure for multiple tests environment can be seen in Figure 3.

![diagram]

Figure 3. CSP-1 procedure for multiple tests

Rework

When a product needs some rework after a failed test (sampled test or not sampled test) it must be retested completely by running all tests. This is because the reworked unit does not present any more the original population that was used to define sampling practice. Due to rework the manufacturing process for this unit was different.

6. Conclusions

This procedure of using CSP-1 for sampling tests offers a quality proof method to decrease amount of testing. The method is based on a standardized sampling procedure and is excellent way to build a common understanding of optimizing testing among all parties in the company including design, quality and production.

Control of sampling is based on tracking the failures; therefore test data management procedures as well as rework data collection have to be managed properly and without gaps.

The defined procedure can be rather complicated to manage if you want to use it in large scale with many products each having tens or hundreds of tests. In modern networked manufacturing environment the control of sampling can be made automatic if procedure is taken into account when developing manufacturing execution system (MES), quality statistics and software in test systems.

7. References


