IMPROVING THE PRODUCT CONTROL OF MECHATRONIC SYSTEMS USING KEY PERFORMANCE INDICATORS

B. Wohlers1, S. Dziwok2, C. Bremer1, D. Schmelter1, W. Lorenz2

1Fraunhofer IEM, Zukunftsmile 1, 33102 Paderborn, NRW, Germany
2Wincor Nixdorf Manufacturing GmbH, Diebold Nixdorf, Heinz-Nixdorf-Ring 1, 33106 Paderborn, NRW, Germany

Abstract
A product control process for complex mechatronic systems has to ensure high quality effectively and efficiently. However, due to the complexity, time-consuming post-assembly tests often lead to high quality costs. The contribution of this paper is a process and an assistance system for the product control of mechatronic systems, which significantly shorten the test duration. Hereto, we define key performance indicators (KPIs), which we use to analyze the product based on its sensor data. We implement our concepts at Diebold Nixdorf – a leading manufacturer of automated teller machines (ATMs) – and define seven ATM-specific KPIs. Moreover, we define a process for identifying and specifying KPIs for mechatronic systems. Finally, we report that our KPI-based product control at Diebold Nixdorf has the potential for a significant reduction in test duration and, therefore, quality costs while keeping at least the same quality level.

Keywords:
Product control, quality control, testing, mechatronic system, Key Performance Indicator (KPI), Six Sigma.

1 INTRODUCTION

Modern technical products are complex mechatronic systems, implying a profound integration of mechanics, electronics, and software. Such products are very demanding to a production system concerning its assembly and quality control. A modern product control process, which is part of the quality control, has to overcome two main challenges: (1) ensuring that the mechatronic system fulfills its functional and quality requirements and (2) being highly time efficient. However, due to the product’s complexity and the high quality demands, the product control often uses time-consuming post-assembly tests leading to high quality costs.

The contribution of this paper is a process and a corresponding assistance system for the product control of mechatronic systems, which significantly shortens the test duration and, therefore, the quality costs, while keeping at least the same quality level compared to existing processes. We realize this by defining key performance indicators (KPIs), which are primarily used to measure the performance of business processes [5, 1, 7]. In our case, we use this concept of KPIs to measure the quality of a mechatronic system while running. In particular, our KPIs are quantitative indicators that statistically measure the product’s core functionalities based on its sensor data. Hereby, we reuse concepts from Six Sigma [2, 3] like the calculation of the process capability index (Cpk) [4]. Based on the KPI results, we can determine automatically to what quality extent the product fulfills its functionalities. Moreover, we can enrich the feedback for the tester with possible root causes and counter measures, which reduce the overall quality control time even more. Noteworthy, we do not define general KPIs applicable for any mechatronic system as these systems may differ widely in their functionality. Instead, we define KPIs specific for a family of products, i.e., all available variants of a product.

We implement our concepts at Diebold Nixdorf (DN) – one of the world’s leading manufacturers of automated teller machines (ATMs). They use complex mechatronic components in their ATMs to satisfy the customers’ needs like correctness and reliability. Moreover, customers of DN have varying requirements for their ATMs, which leads to a wide set of variants per ATM family. Consequently, controlling the quality of an ATM and ensuring its correctness is a challenging task. Therefore, DN uses – among others – “final inspection and testing” [5] for each produced ATM. This leads to extensive testing times compared to a sample test approach. In order to reduce the necessary testing time despite testing all ATMs, we apply our KPI-based product control process at DN and define seven KPIs specific for ATMs. Moreover, we successfully implement a KPI-based assistance system at DN. It measures all KPIs for each ATM in an early phase of the test process, assesses the fulfillment of each KPI, and provides immediate and profound feedback to the tester.

We evaluate the potential of our concept on two ATM families by studying their test process at DN. In particular, we analyze test data of a small sample size of ATMs and discover a potential of ca. 50% test time reduction. Moreover, we report that our improved feedback reduces the time that the testers require to remove defects.

Our paper is structured as follows: Section 2 presents the foundations of this work. In Section 3, we describe the classical product control process at DN. Section 4 explains our concept for a KPI-based product control of mechatronic systems. We evaluate our concept in Section 5 and give a conclusion in Section 6.

2 FOUNDATIONS

As defined by Juran and Godfrey, the purpose of the quality control is to “decide whether or not the product conforms to the product quality goals […] after some amount of product has been produced” ([5] p. 106). Juran and Godfrey also define a quality control process that consists of four major steps: (1) The quality managers choose the control subjects for the quality control process. (2) They establish measurements for the control subjects. (3) Standards of performance (i.e., product goals and process goals) are defined. (4) A feedback loop is integrated. Within the feedback loop, the actual performance is measured and then compared to the defined standards of performance. If the standards are not fulfilled, an action is performed to meet the standards in the next iteration of the loop. Therefore, the quality control process establishes control mechanisms to keep the performance on a defined quality level. In general, the quality control process of Juran and Godfrey is applicable for our problem. However, they do not state how to realize such a process (especially...
for mechatronic systems) in detail.

In order to perform the actual measurement of the subjects, a production system requires product control equipment. The crucial part about this equipment is to test for the right aspects. Lukei et al. [6] show how to derive the requirements of product control equipment for mechatronic systems. However, they do not explain how to design the product control itself.

Six Sigma [2, 3] is an established implementation of quality control that controls processes within an organization. It provides a systematic way to improve customer satisfaction as well as an organization’s bottom line. Therefore, Six Sigma delivers an overall business plan and not only a way of quality control. It introduces key characteristics, often called key process output variables (KPOVs). KPOVs can evaluate process performance. Multiple KPOVs support the evaluation of an organization’s overall performance. KPOVs are evaluated using process capability indices (Cp). Besides KPOVs, there are key process input variables (KPIVs), which affect the KPOVs. The Six Sigma strategy is established to control the KPIVs, in order to reduce the KPOVs amount of variability within a process. Additionally, the process itself can be changed, to affect the KPOVs positively.

Key performance indicators (KPIs) [5, 1, 7] define targets for the operating performance of processes that are important for the success of the organization. Therefore, an organization can use them to compare the actual performance with the target defined by the KPIs. KPIs may be used in business areas like sales, human resources, etc. but also for the production processes. For example, Red Lion Controls Inc. [8] presents seven KPIs for production line monitoring like reject ratio and downtime. However, their KPIs are not applicable for the product control of mechatronic systems. Noteworthy, KPIs may also be considered as process performance indicators within Six Sigma [9].

3 CLASSICAL PRODUCT CONTROL AT DN

As the assembly of mechatronic systems is a complex task, assembly errors cannot be avoided completely. Therefore, DN uses a product control process to control the assembly of its produced mechatronic systems.

We depict the classical product control process of DN, which is among others used during final inspection and testing, in Figure 1. Its description is as follows: The tester connects an ATM to the test station (a PC) and then uses a DN-specific test program, which executes the defined tests on the connected ATM, e.g., a cash-out of 50 banknotes (short: notes) from each cassette of the ATM. During each test, the ATM produces log data for all its sensors. After each test, the test station resp. the test program presents to the tester whether the test was successful or not (e.g., a note might be stuck in the ATM). If any defect occurred, then the tester can manually have a look at the log data to identify the problem.

DN ensures the high quality level of its ATMs by a comprehensive test intensity. Each ATM has to transport a tremendous amount of notes to ensure the detection of all possible defects. The tremendous amount of transported notes within the ATM forces interaction with the tester, e.g., removal or addition of notes. This results in extensive test durations and high quality costs.

The product control process at DN reports about functional correctness but does not analyze the individual quality of each transaction. For example, the test program only validates whether all notes have been successfully transported from their cassette to the output tray and whether the exact amount of notes arrives. However, the process does not analyze whether the note transport was too fast or too slow, which is an indicator for a possible defect of the ATM.

4 KPI-BASED PRODUCT CONTROL OF MECHATRONIC SYSTEMS

In this section, we present our concept for a KPI-based product control of mechatronic systems. We structure the section as follows: First, we provide an overview of our concept in Section 4.1 and introduce the ATM-specific KPI NoteTransportTime as a running example. Afterwards, in Section 4.2, we present additional KPIs for ATMs that we identified so far. Then, we define the structure of a KPI in Section 4.3. Next, we present a KPI-based product control process in Section 4.4 before explaining our KPI-based feedback for testers in Section 4.5. Finally, we present a process for identifying and specifying KPIs in Section 4.6.

4.1 Overview

The goal of our concept is to require less testing time than the classical product control at DN during final inspection and tests by measuring the quality of the mechatronic system using KPIs. In particular, we aim to reduce the necessary testing time of the product control at DN by providing seven KPIs for ATMs. These KPIs analyze the quality of each transaction within the ATM and, therefore, allow statements about the quality of the transactions. As a condition, we assume that all integrated sensors of the mechatronic system are tested before and are working correctly. At the considered assembly stage, the inspected products are mechatronic systems, consisting of mechanics, electronics and software. The concrete KPI values are generated based on already available data being captured within the inspected system. This means that no additional test equipment needs to be installed. The behavior of the system is captured by the system’s sensors, processed by the installed controllers and stored in test logs. Here, continuous data as well as discrete events can be found.

We define KPIs for the production control of mechatronic systems as indicators of quality for a mechatronic system under test. In particular, each KPI controls the quality of at least one core functionality of a mechatronic system. We define core functionalities as functionalities that are inevitable for the correct behavior of the system. For example, the note transport is a core functionality of an ATM because it is necessary for any kind of note transactions (e.g., cash-in, cash-out). Consequently, we define a KPI called NoteTransportTime. This KPI describes the transport time of notes between two points. Both, the distance between the points as well as the speed of notes, are pre-defined by construction. Therefore, the transport time between two points should not differ from the expected value. We are able to measure this KPI by rely on pre-existing light sensors that can detect a note (i.e., the

4.1 Overview

The goal of our concept is to require less testing time than the classical product control at DN during final inspection and tests by measuring the quality of the mechatronic system using KPIs. In particular, we aim to reduce the necessary testing time of the product control at DN by providing seven KPIs for ATMs. These KPIs analyze the quality of each transaction within the ATM and, therefore, allow statements about the quality of the transactions. As a condition, we assume that all integrated sensors of the mechatronic system are tested before and are working correctly. At the considered assembly stage, the inspected products are mechatronic systems, consisting of mechanics, electronics and software. The concrete KPI values are generated based on already available data being captured within the inspected system. This means that no additional test equipment needs to be installed. The behavior of the system is captured by the system’s sensors, processed by the installed controllers and stored in test logs. Here, continuous data as well as discrete events can be found.

We define KPIs for the production control of mechatronic systems as indicators of quality for a mechatronic system under test. In particular, each KPI controls the quality of at least one core functionality of a mechatronic system. We define core functionalities as functionalities that are inevitable for the correct behavior of the system. For example, the note transport is a core functionality of an ATM because it is necessary for any kind of note transactions (e.g., cash-in, cash-out). Consequently, we define a KPI called NoteTransportTime. This KPI describes the transport time of notes between two points. Both, the distance between the points as well as the speed of notes, are pre-defined by construction. Therefore, the transport time between two points should not differ from the expected value. We are able to measure this KPI by rely on pre-existing light sensors that can detect a note (i.e., the
A KPI for a mechatronic system provides one or more metrics that we compute during the execution of the mechatronic system. For example, for the KPI NoteTransportTime, we compute – among others – the arithmetic mean and standard deviation of multiple note transport times. Based on the quality control process of Juran and Godfrey (see Section 2), we define a general process for a KPI-based Product Control (see Figure 2). First, the tester starts the execution for the product under test (i.e., the mechatronic system). We expect that the product can provide a test log, which we can use as the input to our analysis of the KPIs. The analysis of the KPIs is fully automatic and calculates the metrics of the KPI, e.g., a mean of several measurements. Afterwards, we automatically evaluate the KPI results, i.e., we compare the values of the metrics with pre-defined limits. If the values respect the limits, then the test result is positive. However, if a limit of any KPI is violated, then the whole test result is negative and the tester has to take actions, e.g., repair the product.

### 4.2 Additionally identified KPIs for ATMs

Besides the KPI NoteTransportTime, we identified six additional KPIs for ATMs so far. We briefly introduce them in the following.

**ClutchTime** provides information about the quality of the note dispense process in a specific cassette of notes. Technically, it is defined as the time interval between the activating of a dispensing clutch and a change of state of the corresponding dispense light sensor.

**DispenseSkewness** provides – depending on the application area – information about dispensing quality, alignment functionality, or rather transport quality. Technically, it is defined as the obliquity of notes relative to the transport path.

**NoteTransportCurrent** describes the power consumption of different motors under load of the system. To produce a defined load condition, a defined number of notes is processed and moved through the system during the measurement.

**IdleCurrent** describes the power consumption of different motors at idle run of the system. When idling, no notes are moved through the system but the motors are running and, therefore, all transport belts are moved.

**FullClampTransportTime** describes the transport time of a full clamp (a clamp can hold and transport notes) between two pre-defined points. Both the distance between the points as well as the speed of the clamp are pre-defined by construction, so the time of several clamp transports for one path should not differ from the expected value.

**EmptyClampTransportTime** describes the transport time of an empty clamp between two pre-defined points. Both the distance between the points as well as the speed of the clamp are pre-defined by construction, so the time of several clamp transports for one path should not differ from the expected value.

### 4.3 Structure of KPIs for Mechatronic Systems

We show an excerpt of the general structure of a KPI in Figure 3. Technically, a KPI has a name and a description. It defines one or more parameters, which influence the analysis of the KPI, and defines at least three so-called SubKPIs, which provide calculated performance values. A SubKPI has an upper and/or lower limit and an expected value. In addition, a SubKPI also provides at least one possible cause and at least one possible solution for providing feedback when a limit is violated. In the following, we describe all these structural parts in detail.

For explaining the structure of a KPI, we use Figure 4 as a running example. It shows a simplified technical drawing of an ATM, which we call ATM1. The drawing only focuses on the mechatronic parts of the ATM that DN produces itself. Therefore, we leave out parts of an ATM like the keyboard, the monitor, and the credit card slot. ATM1 contains two cassettes for storing notes and one head in the variant front load that delivers the notes to the costumer and receives notes from him. An ATM has multiple sensors along the note transport path to detect a passing note. In our case, ATM1 defines the sensors s1 to s5. The first path is covered by the sensors s1, s2, and s5; the second path is covered by s3, s4, s2, and s5. As ATM1 supports cash-in and cash-out, notes may move in both directions of the paths.

For ATM1, the KPI NoteTransportTime has to analyze both of its paths. However, the KPI does not evaluate the complete path at once but analyzes each section between two sensors separately. For example, the note transport time from s1 to s5 could be in an acceptable timespan but the time from s1 to s2 may be too short and the time from s2 to s5 may be too long. Therefore, analyzing a complete path, which may consist of dozens of sensors (and, thus, dozens of sections), may be too coarse-grained.

### KPI Instances and Parameters

A KPI instance is a concrete manifestation of a KPI. We require KPI instances since the quality control of a mecha-
ronic system typically does not analyze each KPI once but several times. For example, when measuring the quality of ATM1, we analyze the KPI NoteTransportTime for each section twice (the cash-in and the cash-out direction). Therefore, we analyze eight instances of this KPI for ATM1 (each of the four sections in both directions).

We distinguish KPI instances via their parameters. For the KPI NoteTransportTime, the relevant parameters for defining the KPI instances are the two sensors specifying a section and the note direction (cash-in/out).

However, parameters are not only used to distinguish KPI instances in a concrete ATM but also influence the limits that this instance has to obey. For example, a parameter, which is contained by every KPI instance, is the system family of the ATM. Two ATM families might have sensors with the same name but placed on different locations within the ATM. Therefore, the distance between the sensors, and thus the length of the section might differ, which results in different transport times. Another example for such a parameter is the climate zone the ATM will be used in as it influences the functionality of the conveyor belts and the motors.

**SubKPIs**

In contrast to related work (e.g., [1, 7]), we define that a KPI does not provide one but multiple measurements, which we call SubKPIs. A SubKPI is calculated based on measurements provided by the sensors of the mechatronic systems. Each SubKPI has up to two limits, the lower and the upper limit, and an expected value.

A KPI defines at least the following three SubKPIs: (1) the arithmetic mean (μ), (2) the standard deviation (σ), and (3) the process capability index (Cpk). The latter one is an approved statistical measure within Six Sigma [4] for the ability of the production process to produce output within provided specification limits.

The calculation of SubKPI μ is as follows:

\[ \mu = \frac{1}{n} \sum_{i=1}^{n} x_i \]

The calculation of SubKPI σ is as follows:

\[ \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)^2} \]

Concerning KPI NoteTransportTime, μ is calculated based on the timespan \((x_i)\) of \(n\) note transports within a single test while σ is calculated based on \(x_i\) of \(n\) note transports and μ.

The calculation of SubKPI Cpk is as follows:

\[ C_{pk} = \min \left( \frac{\mu - LL}{3\sigma}, \frac{UL - \mu}{3\sigma} \right) \]

The provided specification limits for the calculation of the Cpk are the lower limit \((LL)\) and the upper limit \((UL)\) of SubKPI μ.

In contrast to Six Sigma, we define that the Cpk should not be the only statistical measure. One reason for this is that the Cpk alone is not sufficient for the root cause analysis if a defect occurs, as the reason for a bad Cpk might be a bad mean or a bad standard deviation.

A KPI may have additional SubKPIs to enrich the statement about the quality of the linked core functionality even more and to improve the cause analysis. For example, the KPI NoteTransportCurrent has a SubKPI, which we call MaxOC. MaxOC provides the maximum overcurrent of a specific motor that has been exceeded for at least \(x\) time units, where \(motor\) and \(x\) are parameters of the KPI. Neither \(\mu, \sigma\), nor the Cpk support the analysis of a too long overcurrent.

**Possible Root Causes and Solutions**

The calculation results of all SubKPIs of a KPI instance allows a profound feedback about the quality of the referenced core functionality by providing the limits, the expected value, and the measurement of each instance. Moreover, a KPI contains information about possible causes and possible solutions for a violated limit to facilitate the error detection process. We explain this concept in detail in Section 4.5.

### 4.4 KPI-based Quality Process

In the following, we introduce our KPI-based product control process and the corresponding assistance system. In general, we separate the assistance system in three parts:

1. The test station, where the tester can start the test program including the analysis of appropriate KPIs and receives the KPI-based feedback,
2. a command line tool named KPI-Analyzer (realized in Java) that enables to analyze all KPIs, and
3. a database for KPIs that provides all information necessary for the calculation of a KPI and for generating the feedback for the tester.

Compared to the classical product control process of DN (see Section 3), we extend the test station by the KPI-Analyzer and the KPI database. Figure 5 illustrates the architecture of the implemented assistance system and describes the KPI-based test process at DN in detail. The test process consists of the following ten steps:

1. The tester uses the test station to start the test.
2. The test station initiates the test of the ATM.
3. The ATM executes the test. During the test, the ATM stores relevant data in a test log. These data contain measurements of the sensors (e.g., timestamps when a light sensor detects a note), commands to the actors, and metadata about the configuration of the ATM (e.g., the amount of cassettes). The latter one allows the determination of the matching KPI instances for the actual ATM variant under test.
4. The ATM provides the test log to the test station.
5. The test station delegates the provided ATM data to the KPI-Analyzer and instructs the KPI-Analyzer to start the analysis of the given test.
6. The KPI-Analyzer queries the KPI database, which KPIs are appropriate for this test.
7. The KPI database provides the appropriate KPIs including all relevant information (its parameters, SubKPIs, limits, possible root causes, and solutions).
8. The KPI-Analyzer analyzes the test logs and calculates the KPIs.
9. The KPI-Analyzer provides feedback to the test station.
10. The test station provides feedback to the tester.

![Figure 5. KPI-based Product Control Process at DN.](image-url)
Based on this information, the KPI-Analyzer can analyze the ATM’s metadata to identify the parameter values of all KPIs. Then, the KPI-Analyzer is able to instantiate the KPIs.

8. The KPI-Analyzer calculates all SubKPIs for each KPI instance and compares the results to the limits.

9. Based on the comparison results, the KPI-Analyzer provides the KPI-based feedback to the test station.

10. The feedback provided by the KPI-Analyzer is shown to the tester at the test station. If the ATM violates a limit of any SubKPI analyzed during the test, then the KPI-Analyzer provides additional information to the tester. We explain this in detail in Section 4.5.

The steps 1 – 4 of this process are exactly the same steps as in the classical product control process (see Figure 1). We add the steps 5 – 9 for analyzing the tests via KPIs. Step 10 is similar to step 5 of the classical process but we now provide more profound feedback to the tester.

Note that within one single test, it is possible to analyze multiple KPI instances at once. For example, during a cash-out of notes, we can analyze KPI instances of NoteTransportTime, ClutchTime, DispenseSkewness, Full-ClampTransportTime, NoteTransportCurrent. Therefore, the amount of KPIs does not necessarily increase the number of tests and, therefore, the required test time.

4.5 KPI-based Feedback

The test of an ATM is a complex process, which is challenging for the tester. If the test of an ATM fails, the tester has to identify and solve the problem respectively the root cause. Hereby, for each quality problem, e.g., a too slow note transport, there typically exists more than one possible root cause and for each root cause more than one possible solution. Thus, the tester needs a deep understanding of the ATM. Moreover, DN builds to order and produces just in sequence. Therefore, they do not produce in stock but only produce an ATM when a customer places an order, which results in a wide range of produced ATMs – even within a single production day. Consequently, the tester must be able to identify root causes and solutions for all system families (and, especially, all their variants). Obviously, this is a challenging task. We support the tester while doing this task by providing the following feedback after each test:

1. We list the KPIs that we evaluated and describe what they analyze. Additionally, we list each SubKPI per KPI that we measured and describe what they measure.

2. We provide for each SubKPI the measured value, the expected value, its limits, and the sample size. If applicable, we provide additional data like a graph of a motor’s current consumption.

3. If at least one limit is violated, we provide a list of possible root causes and possible solutions, sorted by likelihood of occurrence. Hereby, the list of root causes and solutions differs depending on whether a lower or an upper limit is violated.

By providing this detailed information, we enable the tester to decrease the necessary time for solving the problem: The first aspect helps him to understand what has been analyzed, the second aspect informs him about the results and if a problem exists, the third aspect provides a list that he can simply execute.

For example, consider the KPI NoteTransportTime and the SubKPI μ. If the upper limit is violated, then notes passed this transport section on average quicker than allowed. An excerpt of the possible root causes and solutions is as follows:

1. incorrect gear ratio → check whether gears are incor-
rect or missing

2. missing torque → check whether gears are missing

4.6 Identifying and Specifying KPIs

While applying our concepts for a KPI-based product control to the quality assurance of DN, we captured the process for identifying and specifying a new KPI including its SubKPIs, limits, and problems and solutions. We present this process in the following.

1. Identify the (core) functionalities of the mechatronic system. Mainly, product managers can support the identification of the core functionalities. The natural example for a core functionality of an ATM is the note transport as it enables the cash-in and cash-out process at the ATM.

2. Identify for each core functionality at least one KPI. Typically, quality managers and developers can identify the appropriate KPIs. For example, all KPIs that we present evaluate the core functionality note transport.

3. Identify the sensors that provide measurements for the KPIs. For mechatronic systems, it is likely that not only one but multiple sensors within the product provide the necessary measurements for the analysis of a single KPI. Moreover, we have to define which measurements of the selected sensors we use. For example, for the KPI NoteTransportTime, we require the measurements of two light sensors. Both sensors provide two kinds of measurements: first, the time when a note begins to cover the sensor; second, the time when a note has passed the sensor. We defined that we only use the measurement when the note passed the sensor. In the worst case, it is possible that necessary measurements are currently missing. Often, a change of the sensor’s firmware might resolve the problem. For example, for the KPIs IdleCurrent and NoteTransportCurrent, we had to implement the functionality that we can optionally log the current of the motors within a specific period. If a firmware change is not sufficient, then the development team has to decide whether the KPI may be analyzed differently or whether sensors might be added.

4. Define the algorithm to calculate the KPI values based on the selected sensors. For example, the calculation for the KPI NoteTransportTime is basically the difference between the two timestamps of the selected light sensors. However, within the algorithm, we also have to consider all possible kinds of errors. For example, if one note directly follows a second one without a gap, then the sensor could report that only one note passed but the algorithm must be able to detect that in fact two notes passed and one measurement is missing.

5. Define the SubKPIs per KPI. Our experience shows that the SubKPIs μ, σ, and Cpk are mandatory. However, additional SubKPIs may exist. For example, as stated in Section 4.2, we defined the SubKPI MaxOC for the KPIs IdleCurrent and NoteTransportCurrent.

6. Identify the necessary parameters of each KPI that influence the limits of its SubKPIs. Our experience showed that for each KPI and each system, the list of parameters might vary. Furthermore, the list of possible parameter values should be collected as well. In Section 4.3, we give examples for parameters and their values for the KPI NoteTransportTime.

7. Define the expected values as well as lower and/or upper limit for each SubKPI. We identified three different approaches for defining these three values per SubKPI:

(a) Calculation: It might be possible that an expert can calculate some of the expected values and limits. For example, concerning KPI NoteTransportTime, both the
distance between the sensors as well as the speed of the notes are pre-defined. Therefore, we can calculate the expected value of the SubKPI $\mu$.

(b) Expert knowledge: It might be possible that an expert may provide some of the expected values and limits without the need of calculations. For example, we define for all KPIs that (1) the lower limit for the $C_v$ for all KPIs is 1.33, and (2) that the expected value and lower limit of SubKPI $\sigma$ is 0. As another example, we define for all instances of KPI DispenseSkewness that the expected value of the SubKPI $\mu$ is always 0.

(c) Statistics: If it is not possible to define the expected value and the limits based on calculations or expert knowledge, then they need to be gathered using a statistical analysis of test logs of products that have proven to be of high quality. In our case, we decided that for each expected value and each limit we require 30 test logs of different ATMs that executed identical tests. Quality experts of DN assured that they have a high quality. In particular, our procedure for defining the expected value and the limits of the SubKPIs $\mu$ and $\sigma$ based on statistics is as follows:

(I) For each of the 30 test logs, we calculate the SubKPIs $\mu$ and $\sigma$. For example, if a test transports 100 notes, then we calculate $\mu$ and $\sigma$ based on these notes. As a result, we have 30 values for $\mu$ and $\sigma$ for further calculations.

(II) We calculate the expected value of SubKPI $\mu$ as the arithmetic mean over all $30 \mu$.

(III) We calculate the expected value of SubKPI $\sigma$ as the arithmetic mean over all $30 \sigma$.

(IV) For specifying the upper/lower limits of SubKPI $\mu$, we add/subtract three times the standard deviation over all $30 \mu$ from/to the calculated expected value of $\mu$.

(V) For specifying the upper/lower limits of SubKPI $\sigma$, we add/subtract three times the standard deviation over all $30 \sigma$ from/to the calculated expected value of $\sigma$.

8. **Define the list of possible root causes and possible solutions for this KPI respectively its KPI instances.**

5 EVALUATION

Our implementation of a KPI-based product control has been successfully deployed at DN’s factory in Paderborn and is already used productively. We currently support two families of ATMs: one family enables cash-in and cash-out while the other one enables cash-out only. Moreover, they also differ internally, e.g., while the first family relies on transport rollers and belts to move notes, the second family additionally uses a clamp to deliver the notes to the customer. The first family requires 5 KPIs, 5 SubKPIs, and 10 different parameters that result to 50 different KPI instances and 294 SubKPI instances; the second family requires all 7 KPIs, 5 SubKPIs, and 12 different parameters that result to 50 different KPI instances and 168 SubKPI instances.

For both ATM families, we only use 40% of the notes compared to the classical product control at DN. This amount of notes is sufficient for the calculation of all KPIs. Due to the reduced amount of notes, the KPI-based test time is reduced by ca. 50%. Note, that the note transports themselves consume much more time than the calculation of the KPI instances. In particular, the analysis of all possible KPI instances in all tests of one ATM is always below 3 seconds for each ATM family. First samples show that the KPI-based product control is able to ensure the same quality level, i.e., we find at least the same defects as the classical product control at DN. Nevertheless, DN decided to test each ATM with the remaining notes (the remaining 60%) for the next months. This enables to collect all test results and allows a statistical analysis. Our hypothesis is that the tests with the remaining notes do not find new defects and are, therefore, obsolete.

We validate the time saving concerning the identification of root causes, selecting the appropriate solution, and removing the defects as follows: We interview the testers of both families. They report a clear time saving due to the improved feedback. In particular, they describe that the provided problems and solutions are especially a great support for inexperienced testers. Additionally, they report that the provided feedback in form of KPIs (i.e., their descriptions) and SubKPIs (i.e., their description, limits, and expected values) supports this process a lot – even for experienced testers. Based on the feedback from the testers, we assume that we achieved a significant time saving due to our KPI-based feedback.

6 CONCLUSION

Product control for mechatronic systems is often an extensive task due to time-consuming post-assembly tests. In this paper, we present an assistance tool that can speed up the test process by utilizing quantitative KPIs for assessing the product quality. We elaborated the process of identifying as well as designing KPIs for mechatronic systems and explained how we use KPIs for the root cause analysis during the production process. Our evaluation by means of a production system for ATMs at Diebold Nixdorf shows the potential for a significant reduction of test time while keeping at least the same quality as before. Due to the excellent results, DN plans to adapt our concept for further ATM families.

For future work, we pursue several directions: First, we want to utilize our KPI-based approach for statistical process control [5] of mechatronic systems. This shall enable the quality manager to analyze the actual status of the production, to understand the production trends, and to execute appropriate countermeasures if necessary. Second, we want to apply our approach to further production systems of mechatronic systems, e.g., retail products like point of sales terminals. Consequently, we will be able to evaluate the transferability of our concept to other mechatronic systems – especially the process for identifying appropriate KPIs. Third, we want to extend our approach by an automatic cause analysis depending on the KPI results. For doing this, we plan to use additional information about the mechatronic system to detect and provide unknown correlations. For example, the implementation of updated components (e.g., a new motor) may influence multiple KPIs. Therefore, KPIs that adhered to their limits before could now violate them. If a KPI analysis fails, then our assistance system shall be able to check automatically databases like ERP (enterprise resource planning) systems whether a relevant component has changed, e.g., a new supplier produces the component. If this is the case, then we inform the tester about this possible root cause.

7 REFERENCES


328


