UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di
Economia e Management

Corso di Laurea Magistrale in
Statistica, Scienze Attuariali e Finanziarie

KPI monitoring and anomaly detection in a production process: a practical application in the professional amplifier industry

Relatore
Prof. Fabrizio Cipollini

Candidato
Giovanni Papini

Anno Accademico 2018/2019

Abstract

Industrial production companies that aim to increase quality, and efficiency, cannot neglect the importance of process supervision. Statistical Process Control (SPC) became a standard tool in this century, in particular in the last 40 years. In this context, I collaborated with Powersoft S.p.A. company to improve the quality of its monitoring system, developing three projects: the reimplementation of the scheduled reporting system for production Key Performance Indices (KPI); an automated anomaly detection system for the same performance metrics; a scheduled reporting system for the daily testing device checks. I analyzed the current situation, explored alternative ways to tackle the problems, discussed them with the stakeholders, implemented and automated end-to-end procedures, from the Extraction Trasformation Load (ETL) of data to the dispatch of synthetic dashboards and files to the respective area managers.

Based on standard methods of statistical quality and process control, we created a highly customized and flexible system with ad hoc adaptations to the specific context in which we applied these tools. We integrated classical process engineering tools with statistical tools such as Generalized Linear Mixed Models (GLMM) and Bayesian estimation algorithms. We adopted methods from scientific literature [Alwan and Roberts, 1988; Lu and Reynolds, 1999], integrating control charts with Auto-Regressive Moving Average (ARMA) models, to treat data serially correlated. Results received positive feedback by the stakeholders, and the tools that I developed for three projects have been tested and implemented in the production.

Sommario

Le aziende di produzione industriale che mirano ad aumentare la qualità e l'efficienza non possono trascurare l'importanza della supervisione dei processi. Il controllo statistico dei processi (SPC) è diventato uno strumento standard in questo secolo, in particolare negli ultimi 40 anni. In questo contesto, ho collaborato con la società Powersoft S.p.A.per migliorare la qualità del suo sistema di monitoraggio, con lo sviluppo di tre progetti: la reimplementazione del sistema di reporting programmato per gli indici di prestazione chiave (KPI) della produzione; un sistema di rilevamento automatico delle anomalie per le stesse metriche di prestazione; un sistema di report programmato per i controlli quotidiani dei dispositivi di test. Ho analizzato la situazione attuale, esplorato metodi alternativi per affrontare i problemi, discusso con gli stakeholder, implementato e automatizzato le procedure end-to-end, dall'estrazione, trasformazione e caricamento (ETL) dei dati all'invio di dashboard e file sintetici ai rispettivi gestori di area.

Basandoci su metodi standard di controllo statistico della qualità e dei processi, abbiamo creato un sistema altamente personalizzato e flessibile con accorgimenti ad hoc per il contesto specifico in cui abbiamo applicato questi strumenti. Abbiamo integrato strumenti di ingegneria di processo classici con strumenti statistici come i modelli lineari generalizzati a effetti misti (GLMM) e gli algoritmi di stima bayesiana. Abbiamo adottato metodi dalla letteratura scientifica [Alwan and Roberts, 1988; Lu and Reynolds, 1999], integrando i control chart con i modelli ARMA (Auto-Regressive Moving-Average), per trattare i dati auto-correlati. I risultati hanno ricevuto feedback positivi dalle parti interessate e gli strumenti che ho sviluppato per i tre progetti sono stati testati e messi in produzione.

Link to the digital version of the document.

PREFACE

> In theory, there is no difference
> between theory and practice.
> But, in practice, there is.
>
> Yogi Berra

During my studies, I have always been fascinated by how the abstraction of mathematical language helps in understanding complex real-world phenomena. Studying Statistics, I became aware that every step toward the powerful simplification (or complication) of abstract models is a step farther from the “dusty” and full-of-shades reality. The hard and methodic building of such models needs the collaboration of a team with a wide range of skills and different points of view, not to lose the connection between the abstract model and the processes it represents.

This awareness was brighter during my bachelor thesis project, where an important part of my efforts was spent understanding the phenomena I was modeling, and the business process around it. Keeping this in mind during my master’s degree, I did not hesitate to accept the challenge of writing my master thesis on optimizing the quality control monitoring system of a manufacturer company in Florence.

I thank the Powersoft company that hosted me during the project. It offered me the opportunity to know a dynamic and vibrant firm and give my small contribution to its improvement. In particular, I thank Marco and Enrico, that supported, supervised, and helped me in the planning and technical implementations. I’m also grateful to Luca, Lorenzo, and Stefano, for the insightful talks on their respective areas of expertise.

I thank professor Leonardo Grilli, for proposing this collaboration. I’m genuinely grateful to my supervisor, professor Fabrizio Cipollini for his readiness to help me, and even more for its precious suggestions and cues.

I owe a particular acknowledge to my friend Lorenzo, that actively contributed this thesis with insightful technical details.

I dedicate this dissertation, conclusion of my course of studies, to my family: to my father, my mother, and my brother, for providing me their unconditional support through these years. Thank you.

Giovanni Papini

Florence, April 2020

PREFAZIONE

> In teoria non c'è differenza tra
> teoria e pratica.
> In practica c'è.
>
> Yogi Berra

Durante i miei studi, sono sempre stato affascinato da come l'astrazione del linguaggio matematico aiuti a comprendere fenomeni complessi del mondo reale. Studiando Statistica, mi sono reso conto che ogni passo verso la potente semplificazione (o complicazione) dei modelli astratti è un passo più lontano da una realtà "sporca" e piena di sfumature. La dura e metodica costruzione di tali modelli richiede la collaborazione di un team con una vasta gamma di competenze punti di vista diversi, per non perdere la connessione tra il modello astratto e i processi che esso rappresenta.

Questa consapevolezza si è intensificata durante il mio progetto di tesi di laurea triennale, in cui una parte importante dei miei sforzi è stata spesa per comprendere i fenomeni che stavo modellando e il processo aziendale attorno ad esso. Tenendo presente questo aspetto durante il mio corso di laurea magistrale, non ho esitato ad accettare la sfida di scrivere la mia tesi di laurea sull'ottimizzazione del sistema di monitoraggio del controllo qualità di un'azienda manifatturiera di Firenze.

Ringrazio l'azienda Powersoft che mi ha ospitato durante il progetto. Mi ha offerto l'opportunità di conoscere una realtà dinamica e vibrante e di dare il mio piccolo contributo al suo miglioramento. In particolare, ringrazio Marco ed Enrico, che mi hanno supportato, supervisionato e aiutato nella pianificazione e nelle implementazioni tecniche. Sono inoltre grato a Luca, Lorenzo e Stefano, per le discussioni formative sulle rispettive aree di competenza.

Ringrazio il professor Leonardo Grilli, per aver proposto questa collaborazione. Sono sinceramente grato al mio supervisore, il professor Fabrizio Cipollini per la sua disponibilità ad aiutarmi, e ancora di più per i suoi preziosi suggerimenti e consigli.

Devo un riconoscimento particolare al mio amico Lorenzo, che ha contribuito attivamente a questa tesi con approfonditi dettagli tecnici.

Dedico questa tesi, conclusione del mio percorso di studi, alla mia famiglia: a mio padre, mia madre e mio fratello, per avermi dato sostegno incondizionato in questi anni. Grazie.

Giovanni Papini

Firenze, Aprile 2020

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACF Autocorrelation Function

ACWV Alternating Current Withstanding Voltage

AP Audio Precision Test

ARIMA Autoregressive Integrated Moving-average

ARMA Autoregressive Moving-average

AUC Area Under Curve

BIC Bayesian Information Criterion

CDF Cumulative Distribution Function

CFP Final Extended Test

COF Functional Test

DGP Data Generating Process

DUT Device Under Test

ETL Extraction Trasformation Load

FPY First Pass Yield

GBON Ground Bonding Resistance

GLM Generalized Linear Models

GLMM Generalized Linear Mixed Models

HV Dielectric Withstand Test

KPI Key Performance Indicator

LCL Lower Control Limit

MCMC Monte Carlo Markov Chain

PACF Partial Autocorrelation Function

PF Finished Product code

PKG Packaging Step

PRG Firmware Programming Test

PTE Production Test Equipment

RTY Rolled Throughput Yield

SPC Statistical Process Control

TPY Throughput Pass Yield

UCL Upper Control Limit

INTRODUCTION

The collaboration with Powersoft S.p.A. began in September 2019 and ended in November 2019. During these three intense months, my direct supervisor was the quality control and certification manager, Marco Cati, while the test engineer Enrico Provinciali assisted me in the technical aspects. The company hosted me in its offices, in the production site of Scandicci, where I shared continuous feedback with my stakeholders. This dissertation reports the activities pursued in this period and provides references for those wishing to deepen the theoretical aspects behind them.

Understanding the topics treated in the text requires some level of technical background: general notions about statistics and quality control are essential requirements. Despite my efforts to make the dissertation accessible to non-technical readers, the methods we used rely on a nontrivial knowledge of mathematics and probability topics. So, I enriched the work with several appendices and bibliographic references: not only to the scientific literature but also to manuals and books for more general arguments (Control Charts in Appendix A and Time Series Analysis in Appendix C).

The dissertation is structured as follows: the first chapter overviews Powersoft's history, business and products, the infrastructures, and the data storage schemes. The second chapter is composed of three sections: the analysis and transformation of testing flow data, the creation of automatic reports about process flows, the implementation of automatic alarms for process anomaly detection. The third chapter treats the analysis of daily testing device checks and the creation of an automated reporting system on those data.

Note that, for privacy reasons, the data presented were partially or entirely anonymized.

CHAPTER

ONE

CONTEXT

1.1 Powersoft

Powersoft S.p.A. is an example of how a mix of creativity, technology, and vision can lead a small start-up in a suburb of Florence to an international venture. The firm is founded in 1995 by two brothers, Luca and Claudio Lastrucci, and their friend Antonio Peruch. In the following years, it conquered its market share in the domain of professional amplifiers thanks to cutting-edge international patents and a massive investment in research and development.

Ideas, electronic projects, mechanical projects, signal processing algorithms, and software are developed entirely under the supervision of the head office. The same office manages the tests, both for certification purposes and to assure the proper functioning of the products in all conditions.

Since 1995 the organization of the company evolved to satisfy the growing demand. From a starting group of few people, now the company has more than 100 employees. The production distributes in the plants of Scandicci (Florence), San Giovanni in Persiceto (Bologna), Cortona (Arezzo), and Ronchi dei Legionari. The whole production and assembling processes locates in Italy. However, thanks to a widespread network, the company sells products in more than eighty nations.

The company develops and delivers products covering the entire branch of professional acoustic amplification, as well as a few innovative devices for professional entertainment: it provides both modules for third-part manufacturers and ready-to-use amplifiers for live events. It also designs and installs entire amplification systems for facilities such as hospitals, stadiums, and parks.

1.2 Testing infrastructure

Developing and producing professional amplifiers needs a meticulous organization. The high price of the devices and their use in professional situations require them to be powerful, sophisticated, and yet highly reliable in any condition. Even if Powersoft provides after-sales assistance, it is inconceivable that an amplifier does not work properly (or even it does not work at all) during an exhibition. As a consequence, the testing phase is fundamental: every single piece passes through several test steps that should intercept every possible anomaly and guarantee a number of defects tending to zero. All test results are archived in the database using an automatic recording system. This data allows the quality control office to carry out statistical and failure analysis at any level of aggregation.

Strictly speaking, the production sites of San Giovanni in Persiceto, Cortona, and Ronchi dei Legionari are not under the direct supervision of Powersoft SpA. They are owned and controlled by third parties, which perceive a remuneration proportionally to the number of pieces supplied. Powersoft provides them the testing equipment and the operators' training. The procedures must respect Powersoft quality control directives, and any information about the testing process returns to Powersoft in real-time.

The testing procedure is standardized following the guidelines issued by the quality control headquarter. Despite that, there is some tendency of different plants to adopt ad hoc adaptations. So, aggregated analyses on the different manufacture centers must take into account the "between group" variability.

1.3 Testing flow

Each product has an ad hoc test plan that technicians define in the phases of design, product, and process engineering. In general, not all products need the same testing sequence. Moreover, each site has different production criteria and, often, assemble different products (e.g., Powersoft never produces semi-finished components, like audio cards). Nevertheless, all testing flows are composed of a sequence of these steps:

Dielectric Withstand Test (HV) [Acronym for *High Voltage*.] It is performed on a component or product to determine the effectiveness of its insulation.

Audio Precision Test (AP) [Acronym for *Audio Precision*.] It checks the audio performance of the tool, and products undergo it when they get their serial number, MAC address and tag.

Functional Test (COF) [Acronym for *COllaudo Funzionale*.] It is performed only on semi-finished components (thus, it is made only by external providers). It is the analog of the AP test on the semi-finished products.

(a) Standard testing sequence in Powersoft internal production.

(b) Standard testing sequence on an automated line in Gorizia.

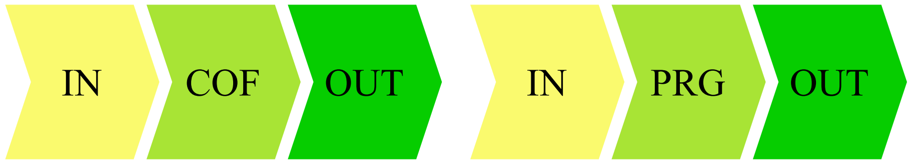
Figure 1.1: Most common testing step sequences (part 1).

Firmware Programming Test (PRG) Test performed at the end of the line to update the firmware version to the latest release. Notice that pieces have a standard firmware version to undergo the previous tests.

Final Extended Test (CFP) [Acronym for *Collaudo Finale Prolungato*.] It is usually not performed in the Scandicci site (in fact the AP test already integrates it). However, some third-party suppliers execute it on amplifiers, making them operate for some minutes at average load.

Packaging Step (PKG) It is not a test: the system generates a report to indicate that an item has reached the end of processing, and the operators pack it up.

Figure 1.1 and 1.2 depicts some of the most common testing sequences. The operators calibrate time to time each step is on the specific product processed. It can conclude with only two possible outputs: *pass* or *fail*. In the case of *fail*, the testing device shows an error code (possibly enriched with numerical measures) and saves the output on the system. The quality control protocol wants the codes encoded following the standardized protocol IRIS (International Repair Information System).



(a) Testing sequence where the HV and COF are summarized in just one report.

(b) Testing sequence for semi-finished components where the HV is not performed.

Figure 1.2: Most common testing step sequences (part 2).

1.4 Database structure

The automated system stores data in a local server, on a Microsoft SQL Server database, available for the statistical analysis of the quality control officers. The extraction of data during this project was performed querying the DB through a direct R connection, using the `DBI` package interface [R Special Interest Group on Databases (R-SIG-DB) et al., 2018] and the in `odbc` package implementation [Hester and Wickham, 2018].

The database has a relational structure and is updated loading new records from the XML reports generated by the proprietary software Powersoft Production Test Equipment (PTE). At every test, the device saves a file containing information about: measures, pass/fail status, error details, duration in seconds, start-time, finish-time, single measurement steps, specification limits, and many further details specified in the test plan. The final output is a set of XML files, with a hierarchical structure that represents the objects created by the programming logic. Every table described in the next paragraph is associated with a nested level of classes represented in the XML report.

The master table is `Session`, in which every ID associates with a unique report generated by PTE. Each record corresponds to one test execution, with an output status *pass* or *fail*. `Session` table is related to the `TestPlan` table through a one-to-one relation. Passing by this relation, it is possible to reconstruct the entire measurement sequence, accessing to the `Measurement` and `MeasurementValue` tables. Also, in the `Session` table one can associate each test to its plant and type, through the `CFPSystem` table, where the server stores data about testing devices. Lastly, in the `IRISSessionMessage` table you gather information about error details (e.g. area of the item, symptoms). Figure 1.3 on the following page schematizes the database structure and the relations between tables.

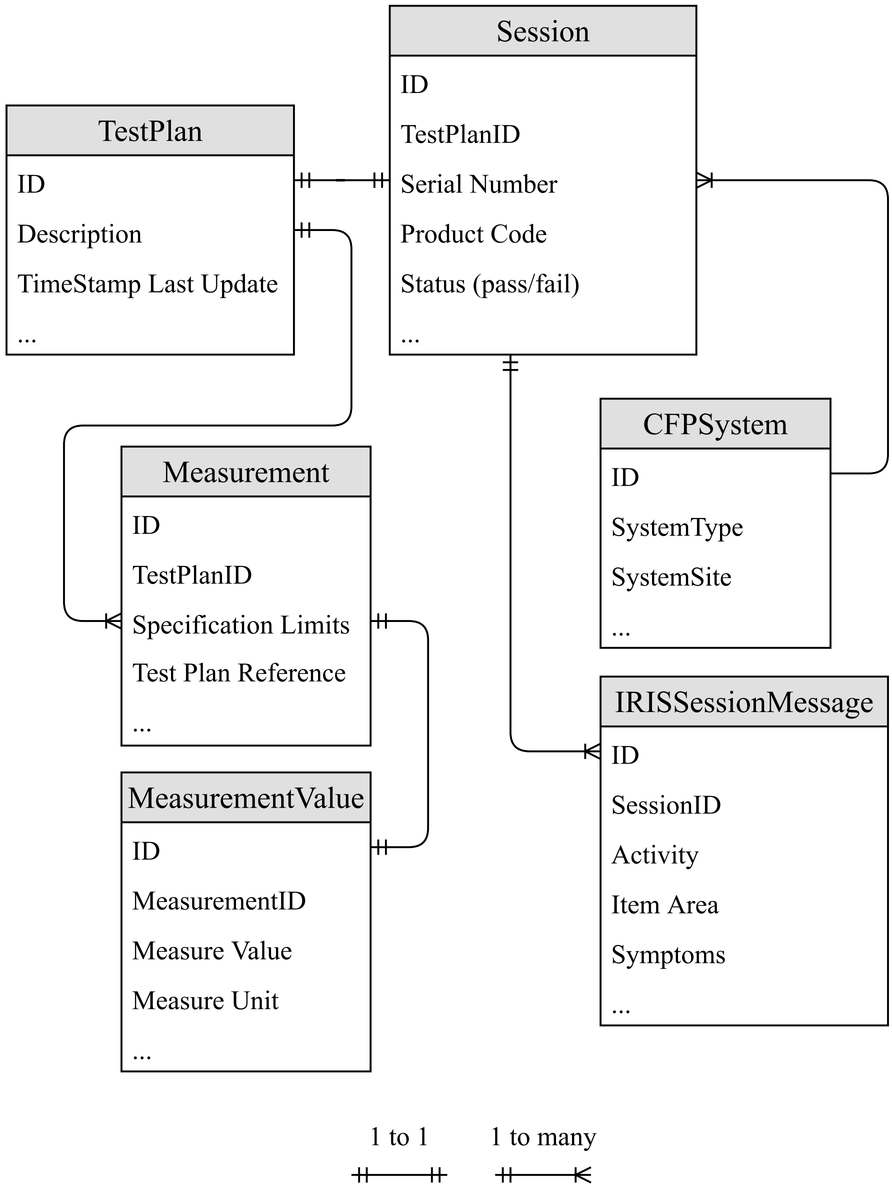


Figure 1.3: Main structure of Powersoft's database system for online testing data.

CHAPTER

TWO

PRODUCTION KPI MONITORING

Constant training of the factory operators and monitoring of the production are fundamental parts of the daily work in a company that produces highly technological products.

On the same day, the number of assembled products can be high if compared to the number of pieces: daily production plans contribute to this heterogeneity. In fact, on the same day, some of the products might undergo the testing procedure to be immediately commercialized, others could be residuals that did not pass the tests in the previous days or they can be prototypes in early development. The most specific classification of products is the Finished Product code (PF): at the moment there are more than 150 PFs, and each of them is part of a family of models with its characteristics, testing plans, and policies. In 2019, the average number of unique PFs per day was over 20, corresponding to almost 10 different families.

2.1 Tested product classification

Some heterogeneity is also present in terms outcome: in different contexts, the same item can be tested several times before passing the test or can be put immediately aside for further checks or reparations. The raw daily number of passed or failed tests per PF is, thus, a poor indicator of the production performance: it is necessary to track the whole "history" of an item. It is possible to see those histories at two levels of focus: the product testing in a single step, or the entire process (from the assembling to the packaging). Both these perspectives are critical for the decision about production politics and lead to different classifications.

2.1.1 Classification over single tests

Single-test classification requires thinking about the possible outcomes of a step in those listed in section 1.3: the test can result in *pass*, or it can *fail*; in the latter case it can be immediately re-tested, or taken by an operator that tries to fix it. Given a sequence of passed or failed tests, a piece on a single step can be classified as:

`1st PASS` A piece that passed the test at the first attempt: in an ideal scenario, all pieces would be in this category;

`2nd PASS` A piece that failed at least once, was re-tested (one or more times) and passed it without any fix. Usually, it results from a bad linkage or collateral errors of the testing device;

`REPAIRED` A piece that failed some test, was repaired by an operator and then passed the test. The operator that changes or repairs any part in the product marks it as "repaired" to distinguish it from `2nd PASS`es;

`1st FAIL` A piece that failed the test at the first try: the subsequent treating depends on the characteristic of the error, the test type, and the site policies. It can be re-tested or given to a specialized operator to fix it;

`2nd FAIL` A piece that failed the test for the second (or higher) time;

`CHECK` A piece that was re-tested even though it already passed. Such cases play a marginal role.

These outcomes are associated to a specific combination of serial number and test (one of AP, HV, CFP, COF, PKG). Figure 2.1 on the next page represents the algorithm to attribute the correct status.

In many low-cost productions, it is common that defective pieces (`1st`/`2nd FAIL`s) exit the workflow as trash. In Powersoft, wastes are a negligible percentage: test-failing products get re-tested or repaired until they pass. So, an amplifier classified as `2nd FAIL` in Day 1 could be re-classified as `2nd PASS` or `REPAIRED` in Day 2, and so on. Figure 2.2 on page 10 represents how the outcomes of a single serial could evolve through its testing until it passes.

Some examples are reported in Tables 2.1 and 2.2:

- Amplifier `SN0001` is tested, in the context of continuous production, in the Scandicci plant (PWSPRD):
  - it passes the HV test at the first try (`1st PASS` for this system type);
  - it is tested at AP, where it fails, because some measures were close to specification limits. It is then immediately re-tested with success, so it is classified as `2nd PASS`;

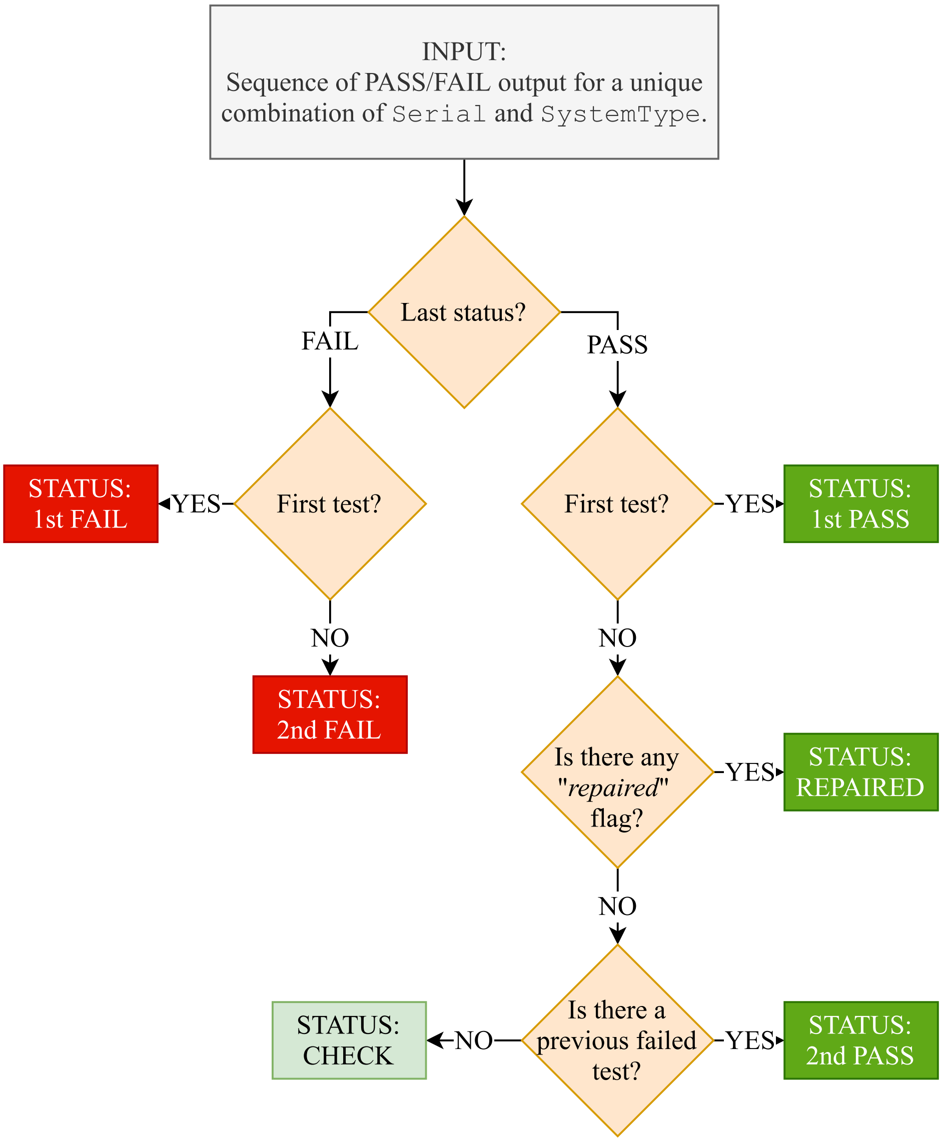


Figure 2.1: Algorithm to assign the status to a serial tested on a single step.

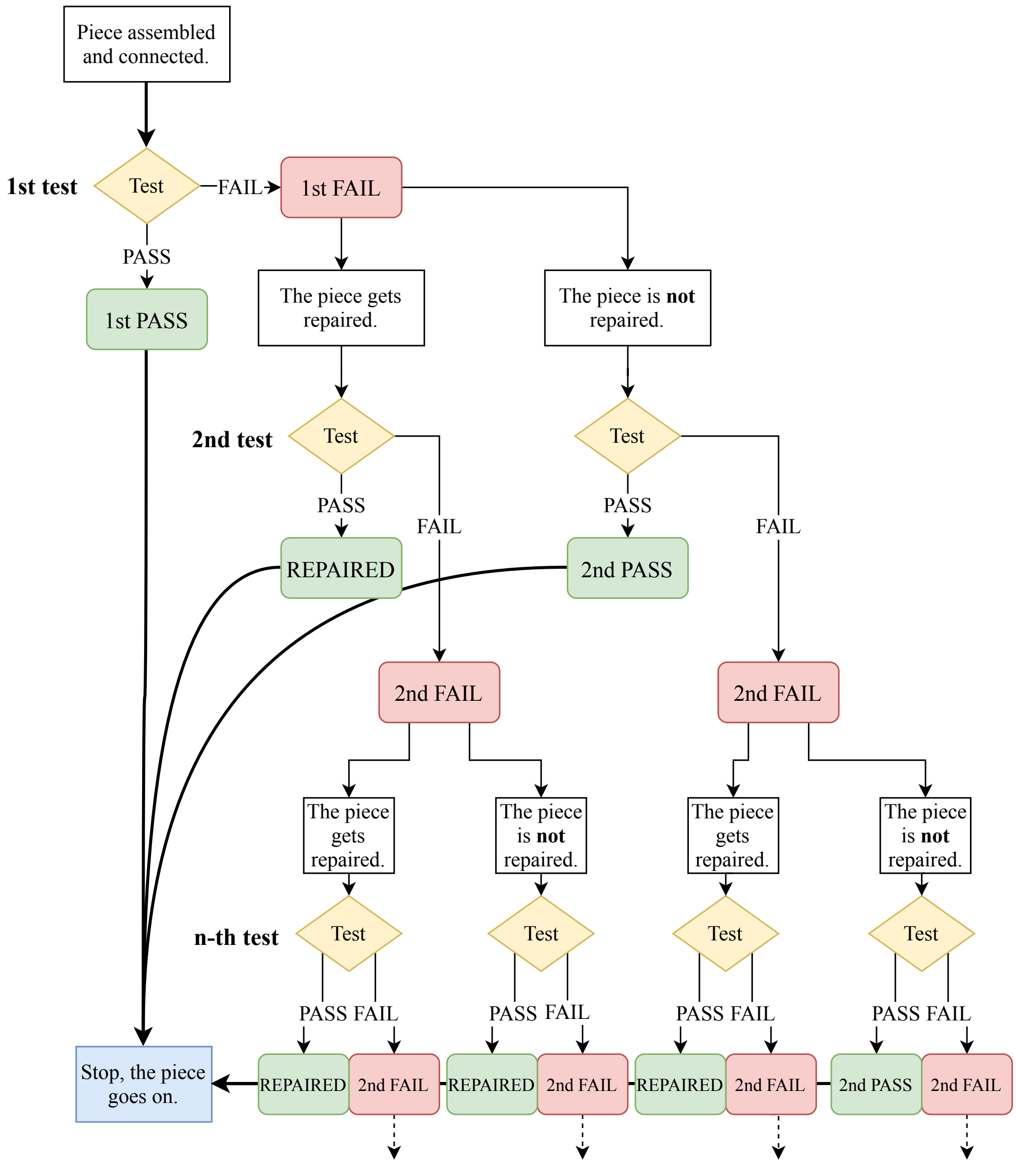


Figure 2.2: Flow of the possible sequential classifications of a piece in a *single test* (see the section 1.3 for a list). When a piece reaches a green cell, it has passed the test. In this case, it can go on in the testing cycle. When it reaches a red cell, it interrupts the regular testing flow and can be submitted to a direct re-test ("The piece is **not** repaired"), or it can be fixed ("The piece gets repaired"). There are cases in which the item does not end the processing flow and exits as waste. Which path to choose is determined by a set of contextual factors like PF, type, and site.

  - it passes the PRG step at the first attempt (classified as `1st PASS`) and it is finally ready to be packaged.

- Amplifier `SN0103` is tested, in the context of a continuous production, in the Ronchi dei Legionari plant (RDL):
  - at the COF step, it does not pass the test, so it reaches the repairer operator that changes some internal components. It is flagged as *repaired* and passes the second test. It can prosecute the test cycle, classified as `REPAIRED` in this test type;
  - it is tested at HV, where it passes at the first attempt (classified as `1st PASS`).

- Audio card `SN1101` is tested, in the context of continuous production, in the Cortona plant (CRT):
  - it fails the AP test, but since it is late, the item is not re-tested immediately: it is left apart for the next morning. In day 1, it figures as `1st FAIL`. The next day it is re-tested and passes, resulting in a `2nd PASS`;
  - the card now is ready to be installed on an amplifier.

- Amplifier `SN1010` is tested, in the context of low volume production, in the Scandicci plant (PWSPRD):
  - it passes the HV test at the first attempt;
  - similarly it passes the AP test step at the first attempt;
  - it fails at the PRG step. Following specific policies, it is immediately re-tested a second and a third time, without success. While one operator analyses the error codes, the piece is re-tested both in HV and AP steps, to check the functional capability, and it passes both. Checking the blocking error at the PRG step, it results that there is a problem in a cable connection: once having replaced it, the operator re-test the item obtaining a final *pass*.

  Overall serial `SN1010` results as a `CHECK` for both HV and AP, and as a `2nd PASS` for PRG.

These are only a few examples: during the creation of the new reporting system, we examined lots of other pieces case-by-case. During the stay at Powersoft, we validated the data stored in the SQL database system over an entire day of production, annotating all serial numbers, event times, passes, and fails. In this way, we compared the annotations with the database records, to check data quality and reliability of the system.

DateTime	Serial	PF	Type	Site	Status
Day 1 - 09:09:57	SN0001	PFXYZ	HV	PWSPRD	PASS
Day 1 - 09:11:43	SN0001	PFXYZ	AP	PWSPRD	FAIL
Day 1 - 09:14:45	SN0001	PFXYZ	AP	PWSPRD	PASS
Day 1 - 09:16:44	SN0001	PFXYZ	PRG	PWSPRD	PASS
Day 1 - 09:24:11	SN0001	PFXYZ	PKG	PWSPRD	PASS
Day 1 - 10:15:11	SN0103	PFLMN	COF	RDL	FAIL
Day 1 - 10:25:11	SN0103	PFLMN	COF	RDL	REPAIRED
Day 1 - 10:26:36	SN0103	PFLMN	COF	RDL	PASS
Day 1 - 10:28:25	SN0103	PFLMN	HV	RDL	PASS
Day 1 - 10:38:08	SN0103	PFLMN	PKG	RDL	PASS
Day 1 - 17:10:16	SN1101	SMDEF	AP	CRT	FAIL
Day 2 - 07:45:17	SN1101	SMDEF	AP	CRT	PASS
Day 2 - 14:50:02	SN1101	SMDEF	PKG	CRT	PASS
Day 2 - 09:51:11	SN1010	PFZYX	HV	PWSPRD	PASS
Day 2 - 09:54:32	SN1010	PFZYX	AP	PWSPRD	PASS
Day 2 - 10:05:21	SN1010	PFZYX	PRG	PWSPRD	FAIL
Day 2 - 10:06:57	SN1010	PFZYX	PRG	PWSPRD	FAIL
Day 2 - 10:08:21	SN1010	PFZYX	PRG	PWSPRD	FAIL
Day 2 - 10:10:52	SN1010	PFZYX	HV	PWSPRD	PASS
Day 2 - 10:13:52	SN1010	PFZYX	AP	PWSPRD	PASS
Day 2 - 10:05:33	SN1010	PFZYX	PRG	PWSPRD	PASS
Day 2 - 10:10:13	SN1010	PFZYX	PKG	PWSPRD	PASS

Table 2.1: Toy example of query output for the creation of a process monitoring dashboard. Each record corresponds to a unique combination of Date-Time and Serial, and describes the output of a unique test.

Date	Serial	PF	Type	Site	Status
Day 1	SN0001	PFXYZ	HV	PWSPRD	1st PASS
Day 1	SN0001	PFXYZ	AP	PWSPRD	2nd PASS
Day 1	SN0001	PFXYZ	PRG	PWSPRD	1st PASS
Day 1	SN0103	PFLMN	COF	RDL	REPAIRED
Day 1	SN0103	PFLMN	HV	RDL	1st PASS
Day 1	SN1101	SMDEF	AP	CRT	1st FAIL
Day 2	SN1101	SMDEF	AP	CRT	2nd PASS
Day 2	SN1010	PFZYX	HV	PWSPRD	CHECK
Day 2	SN1010	PFZYX	AP	PWSPRD	CHECK
Day 2	SN1010	PFZYX	PRG	PWSPRD	2nd PASS

Table 2.2: Classification of serials in Table 2.1 with single-test focus. Every record corresponds to a unique combination of Date, Serial and Type, describing the whole history of a single item for a single test type.

2.1.2 Classification over the entire process

Another classification algorithm has been developed (see Figure 2.3 on the next page) to classify each piece on the base of its whole history:

`All passed` A piece that has concluded the process, never failing a test (all `1st PASS` or `CHECK`);

`Hitch` A piece that failed at least one test (any of `2nd PASS`, `REPAIRED`, `1st`/`2nd FAIL`);

`In progress` A piece that passed all previous tests (`1st PASS` or `CHECK`) but has not yet concluded the process.

See Figure 2.4 on page 15 for the representation of the possible outcomes for a serial number.

Note that all pieces in Table 2.1 fall in the category `Hitch` because all pieces fail at least one test once.

2.2 Descriptive KPI reporting and automation

The quality control reporting system at the beginning of the collaboration was implemented in Minitab. It consisted of a procedure transforming the data from the structure in Table 2.1 (directly queried from the SQL database) to the structure in Table 2.2, using in some cases loopholes to circumvent Minitab lack of flexibility. We reverse-engineered the routines, to reimplement it using the R programming language [R Core Team, 2019]. We implemented the classification

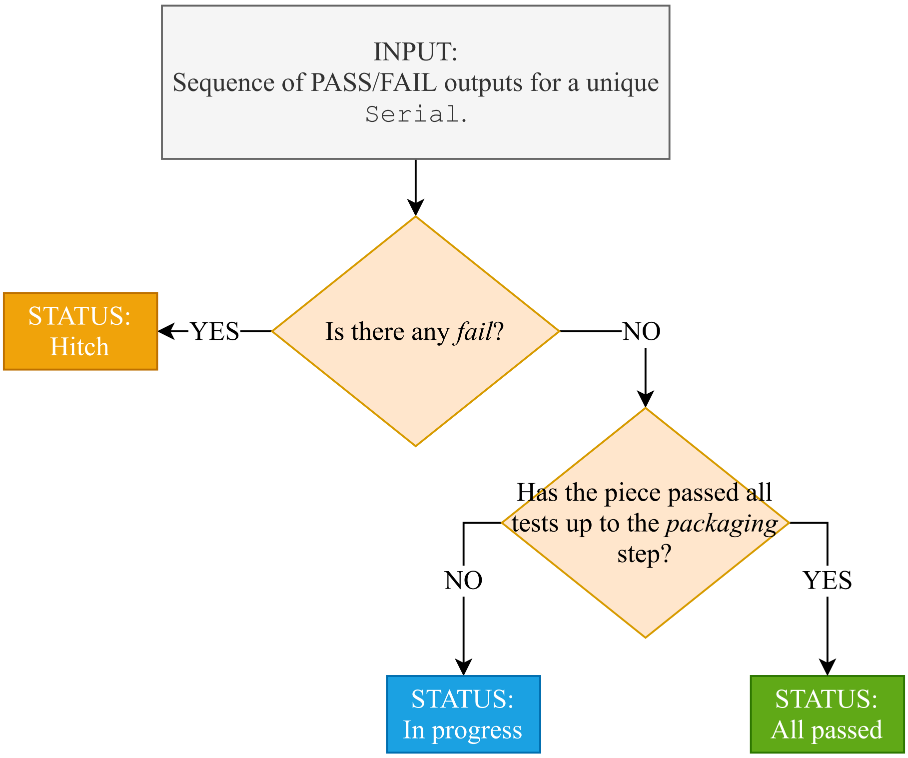


Figure 2.3: Algorithm to assign the status to a tested serial over the entire process.

algorithms in Figures 2.1 and 2.3 into functions that receive in input the sequence of statuses of a piece over the processing time and returns in output its final status.

The entire ETL flow was refactored using the `tidyverse` framework [Wickham et al., 2019] to guarantee maintainable code, the possibility of reusing scripts and if necessary easily change the procedures. The new reporting system generates synthetic dashboards and CSV files on daily, weekly, and monthly basis that a program automatically sends to different stakeholders with details about the respective production site. The Powersoft quality control head office, as well as the internal production managers, have the complete overview of the situation; third part producers obtain the daily report concerning their site.

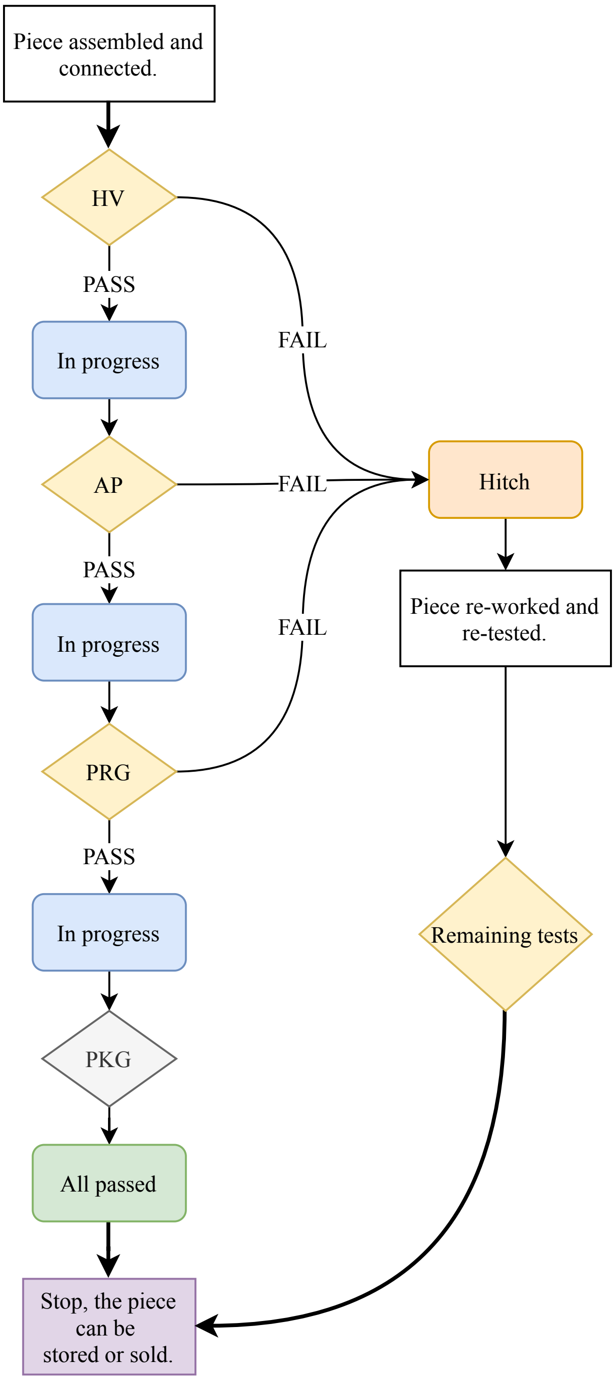


Figure 2.4: Flow of the possible sequential classifications of a piece with *complete process focus*, in case of Scandicci standard testing sequence (see Figure 1.1a on page 4). In other cases the flow is the same, the only difference can be in the number of intermediate steps (HV, AP, PRG) that pieces must pass before concluding the testing cycle.

2.2.1 Process Performance Metrics

Factory operators and production line managers are the employees with the clearest and most immediate view of the situation. However, sometimes they focus on the details of the production and lose the "full-picture." The higher we go in the company hierarchy, the more we need synthetic indicators of the process as a whole, from which we can discover problems and inefficiencies, besides identifying possible process improvements.

Since years, Powersoft has adopted practices of the Lean-Six Sigma methodology (a combination of Lean Manufacturing and Six Sigma techniques), to improve performance by systematically removing waste and reducing the variation of the process metrics [Allen, 2006; Daneman, 2014].

First Pass Yield (FPY) In a Lean-Six Sigma production, the primary Key Performance Indicator (KPI) is the FPY, also known as Throughput Pass Yield (TPY). This metric, calculated for single tests, consists of the ratio between the number of items that pass it without needing a re-working and the total number of processed items:

$$\text{FPY} = \frac{\text{number of } \texttt{1st PASS}}{\text{total number of pieces in the process}} \tag{2.1}$$

It ranges between 0 and 1, where 1 (0) indicates that all the items did (not) passed the test at the first attempt.

In an ideal context, the production manager would like a daily FPY of 1 or almost 1. This goal is not always possible, because the process tends to have a part of `2nd PASS` pieces. In particular, FPY is likely to have lower values at the release of new products, or when the production re-starts after a long time (operators need to re-acquire familiarity with the test device, technicians have to calibrate the tests). The situation is worse and needs the process analysts to better investigate if the FPY decrease for a high number of `REPAIRED` or `1st`/`2nd FAIL` pieces. So, FPY can give only first-level warnings: if it drops to low values, it is necessary to acquire background info, such as how many pieces have entered the process, and which error codes have the devices generated.

Rolled Throughput Yield (RTY) The second (and for many experts most important) metric to monitor is the RTY. This index is the ratio between the number of items that passed all the tests without any *fail* and the total number of processed items:

$$\text{RTY} = \frac{\text{number of items arrived at the packaging with } \textbf{only } \texttt{PASS}}{\text{total number of pieces in the process}} \tag{2.2}$$

As FPY, it ranges between 0 and 1. It gives a more generic overview of the process performance: when it is equal to 1 it indicates that all the pieces that started the process have concluded without ever stopping at any step. In the opposite case where it is equal to 0, it indicates that all the pieces that started the process are still being worked or failed at least once.

2.2.2 Daily reports

The first periodic report to implement was a synthesis of the production day-by-day. Every morning a dashboard presenting an overview of the entire process of the previous day is generated and sent by e-mail. The dashboard contains two graphic representations:

FPY part On the left side there is a focus on single test steps (see Figure 2.5 on page 19). A matrix orders the processes into rows and columns: rows refer to the same PF, columns to the same test type. Inside each cell there is a bar plot representing the relative distribution of statuses (listed in subsection 2.1.1 on page 8). At the end of each bar, a numeric label shows the exact count of serials in the bar, both as absolute and relative frequency. The main performance metric (FPY) is the percentage of `1st PASS`es in the upper green bar.

RTY part On the right side there is an overview of the entire production (see Figure 2.6 on page 20). This representation contains only one column, referring to the entire site, and the same number of rows of the FPY part, referring to the same PFs (the ordering is coherent in the two parts). Inside each cell there is a bar plot representing the relative distribution of statuses (listed in subsection 2.1.2 on page 13). Even here at the end of each bar, a numeric label indicates the exact frequency, both in absolute and relative terms. The main performance metric (RTY) is the percentage of `All passed` in the upper green bar.

In each cell, the dashboard exhibits a vertical line at the level of the previous week's KPI for that specific combination. The KPI value is the FPY or the RTY calculated in the period from Monday to Friday of the previous week, for each combination of PF, type, and site in the dashboard matrix. In cases where the combination is not present in previous week data, the vertical line is omitted. The idea is that, in the future, the vertical line will define a target, to constantly increase production quality. At the top of the dashboard, a title indicates the production company and site (see Figure 2.5), together with the time of the first and last records found in the dataset at that date.

While reading the two parts of the dashboard, note that they tell the same "story" (they represent the same period and site data) but have a different scope. Figure 2.6 describes how the production is going and which is the most problematic product if any. Figure 2.5 shows where the process is failing and if the problem is *test*-related (many `2nd PASS`) or *product*-related (many `REPAIRED` and `FAIL`s).

Together with the graphic dashboard, there is a set of CSV files attached to the e-mail:

`DailyFPY.csv` A frequency table of the data in the FPY dashboard, with absolute counts, the daily FPY and previous-week's one;

`DailyRTY.csv` A frequency table of the data in the RTY dashboard, with absolute counts, the daily RTY and previous-week's one;

`FailureAnalysis_2ndPASS.csv` A detailed list of errors relative to `2nd PASS` pieces (see Table 2.3a);

`FailureAnalysis_REPAIRED.csv` A detailed list of errors and their reparations relative to `REPAIRED` pieces (see Table 2.3b);

`FailureAnalysis_FAIL.csv` A detailed list of errors relative to `1st`/`2nd FAIL` pieces (see Table 2.3c).

Failure analysis files have a very low-level focus and give very detailed info to process engineers in order to discover product defects and process inefficiencies.

2.2.3 Weekly and monthly reports

To have a mid- and long-term view of how the process is performing, we also created weekly and monthly dashboards, similar to the daily reports.

The example of a daily report in Figure 2.5 shows that the high number of PFs can comport a large image. For more extended periods, we expected even a higher number of rows to display. Exploring a wide-area matrix, with much info condensed in a little space, could make anomalies less visible, hardening to have a clear overview of the situation. To improve readability, we aggregated counts for long-period reports by *family* instead of PF. The dashboard still lists PF codes under row names on the right column (see Figure 2.7 on page 22), to keep the info on which products belong to the families.

Even in these dashboards, a vertical line indicates the level of the previous period KPI, which is the previous week for weekly reports, the previous month for the monthly ones. In each cell, the KPI is relative to that specific combination of family and test type.

The script generates and sends weekly (monthly) reports every first day of the week (month) in the morning, together with failure analysis CSV files in the same form of Figure 2.3 on page 21.

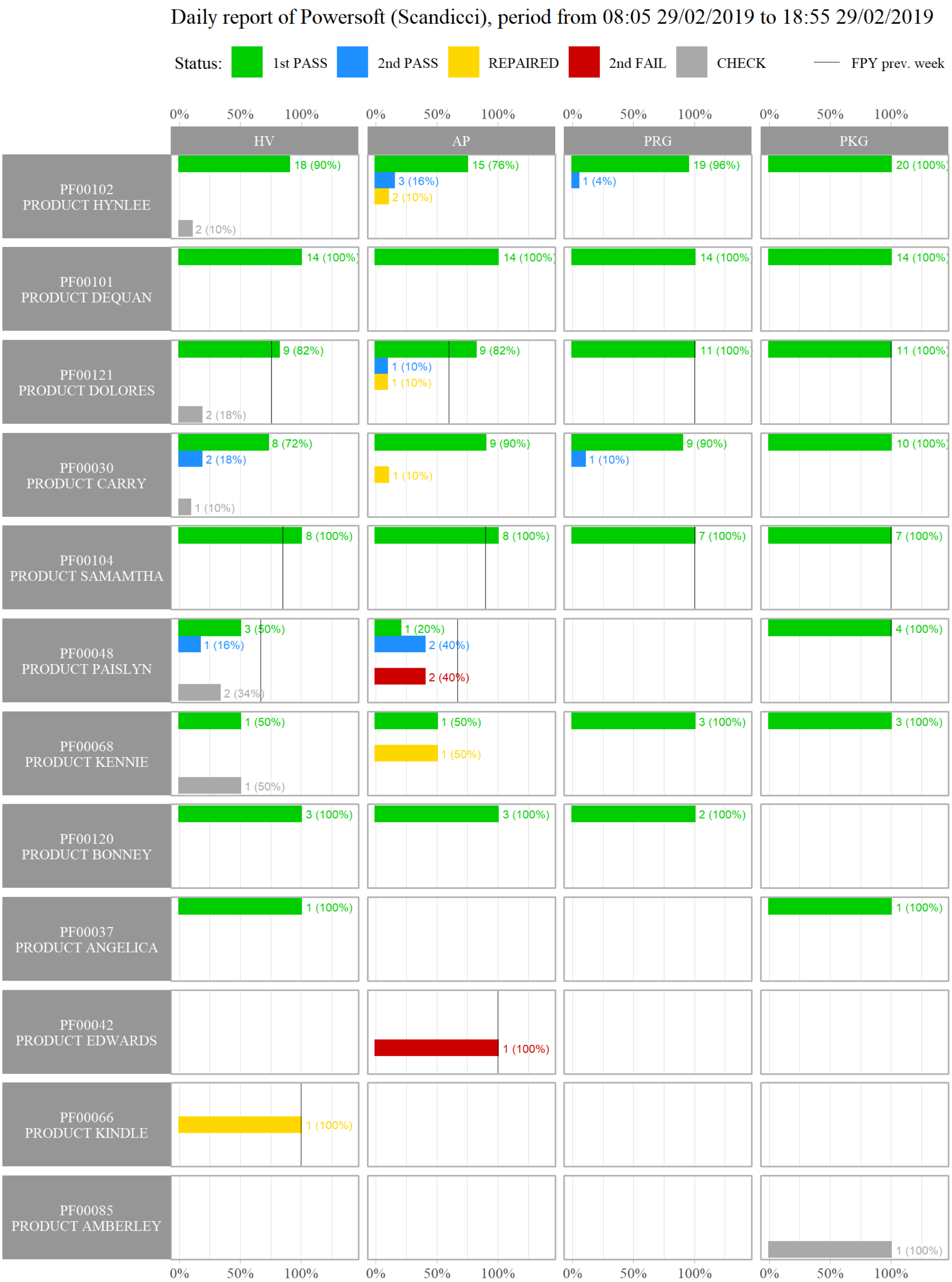


Figure 2.5: Anonymized example of dashboard visualization for daily FPY report. This dashboard is generated for every production site (Scandicci, San Giovanni in Persiceto, Cortona, Ronchi dei Legionari).

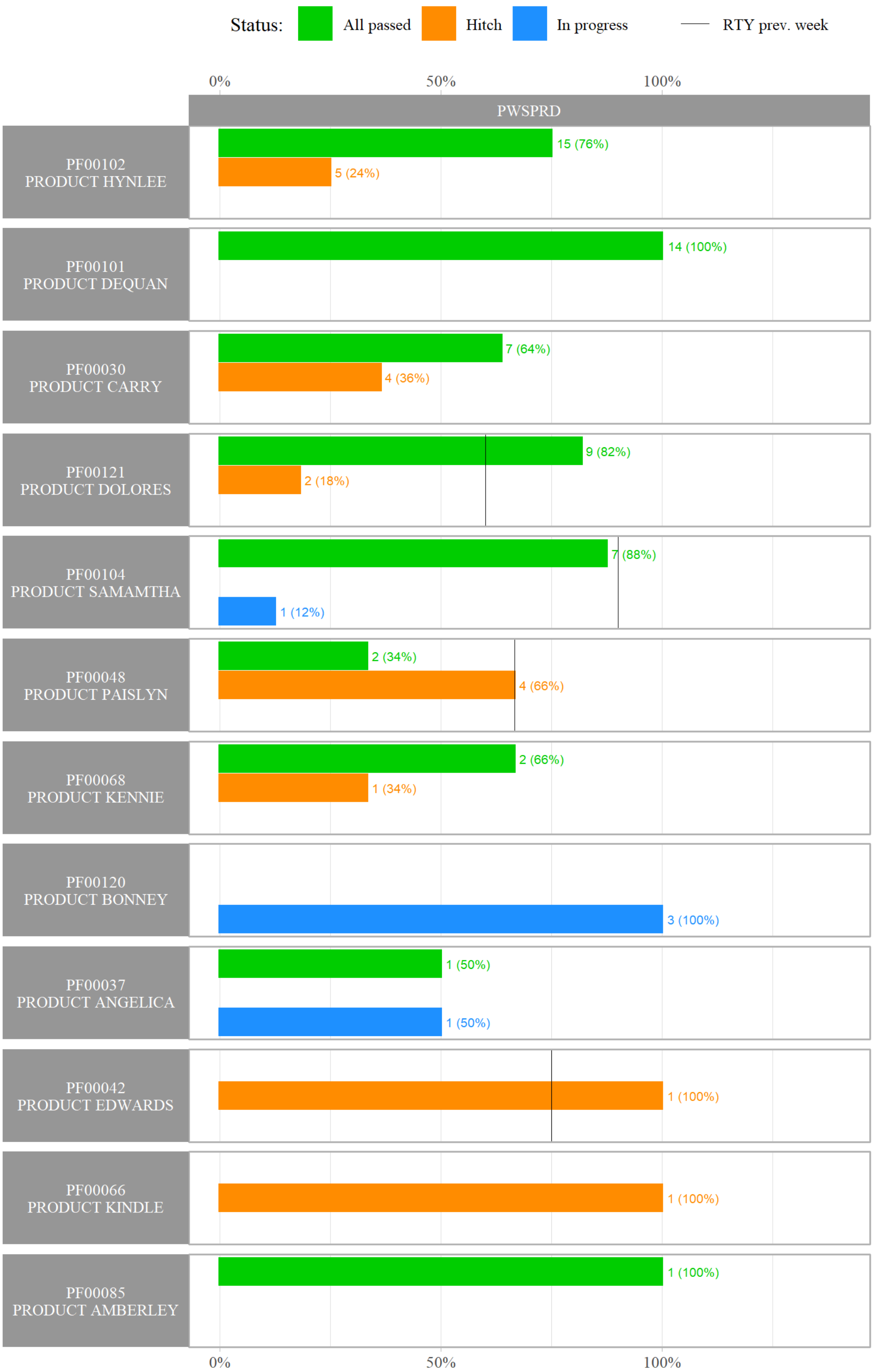


Figure 2.6: Anonymized example of dashboard visualization for daily RTY report. This dashboard is generated for every production site (Scandicci, San Giovanni in Persiceto, Cortona, Ronchi dei Legionari).

Serial	DateTime	Status	FinalStatus	PF	Site	ActivityDescription	FailureDescription
SN0001	Day 1 - 11:22:49	FAIL	2nd PASS	PFX001	RDL		JLink WFS Amplifier program Timeout ERROR
SN0001	Day 1 - 11:24:39	FAIL	2nd PASS	PFX001	RDL		Securing AMP 4211 error: correct target not found!
SN0003	Day 1 - 15:56:34	FAIL	2nd PASS	SM00X2	RDL		4ohm LOAD CH1-16 Noise Level ; Noise (RMS) ; Noise Level
SN0003	Day 1 - 15:58:43	FAIL	2nd PASS	SM00X2	RDL		WFS FW Upgrade brOut error: correct target not found!
SN0123	Day 1 - 16:01:02	FAIL	2nd PASS	SM00X2	RDL		WFS FW Upgrade error!

(a) `FailureAnalysis_2ndPASS.csv`: Failure analysis of `2nd PASS` pieces in Ronchi dei Legionari (RDL).

Serial	DateTime	Status	FinalStatus	PF	Site	ActivityDescription	FailureDescription
SN0101	Day 1 - 12:06:31	FAIL	REPAIRED	PFZ301	PWSPRD		230VAC 8ohm CH5-6; Frequency Response [...]
SN0101	Day 1 - 12:21:41	FAIL	REPAIRED	PFZ301	PWSPRD		230VAC 8ohm CH5-6; Frequency Response [...]
SN0101	Day 1 - 14:49:17	REPAIRED	REPAIRED	PFZ301	PWSPRD		230VAC 8ohm CH5-6; Frequency Response [...]
SN0909	Day 1 - 14:05:10	FAIL	REPAIRED	PF0987	PWSPRD	First Power ON 240V	
SN0909	Day 1 - 14:48:30	REPAIRED	REPAIRED	PF0987	PWSPRD	First Power ON 240V	

(b) `FailureAnalysis_REPAIRED.csv`: Failure analysis of `REPAIRED` pieces in Scandicci (PWSPRD).

Serial	DateTime	Status	FinalStatus	PF	Site	ActivityDescription	FailureDescription
SNX202	Day 3 - 08:35:11	FAIL	2nd FAIL	PFXY01	SGP		230VAC BURNIN 8ohm ; Measurement Recorder ; Peak Level
SNX202	Day 3 - 09:26:51	FAIL	2nd FAIL	PFXY01	SGP		34P Signals CH1-2 ; +/-12VDC -20dB ; DC Level
SNX207	Day 3 - 10:42:13	FAIL	1st FAIL	PFXY01	SGP		
SNX210	Day 3 - 14:46:19	FAIL	1st FAIL	PFXY01	SGP		34P Signals CH1-2 ; +/-12VDC -20dB ; DC Level

(c) `FailureAnalysis_FAIL.csv`: Failure analysis of `FAIL` pieces in San Giovanni in Persiceto (SGP).

Table 2.3: Example of failure analysis CSV files.

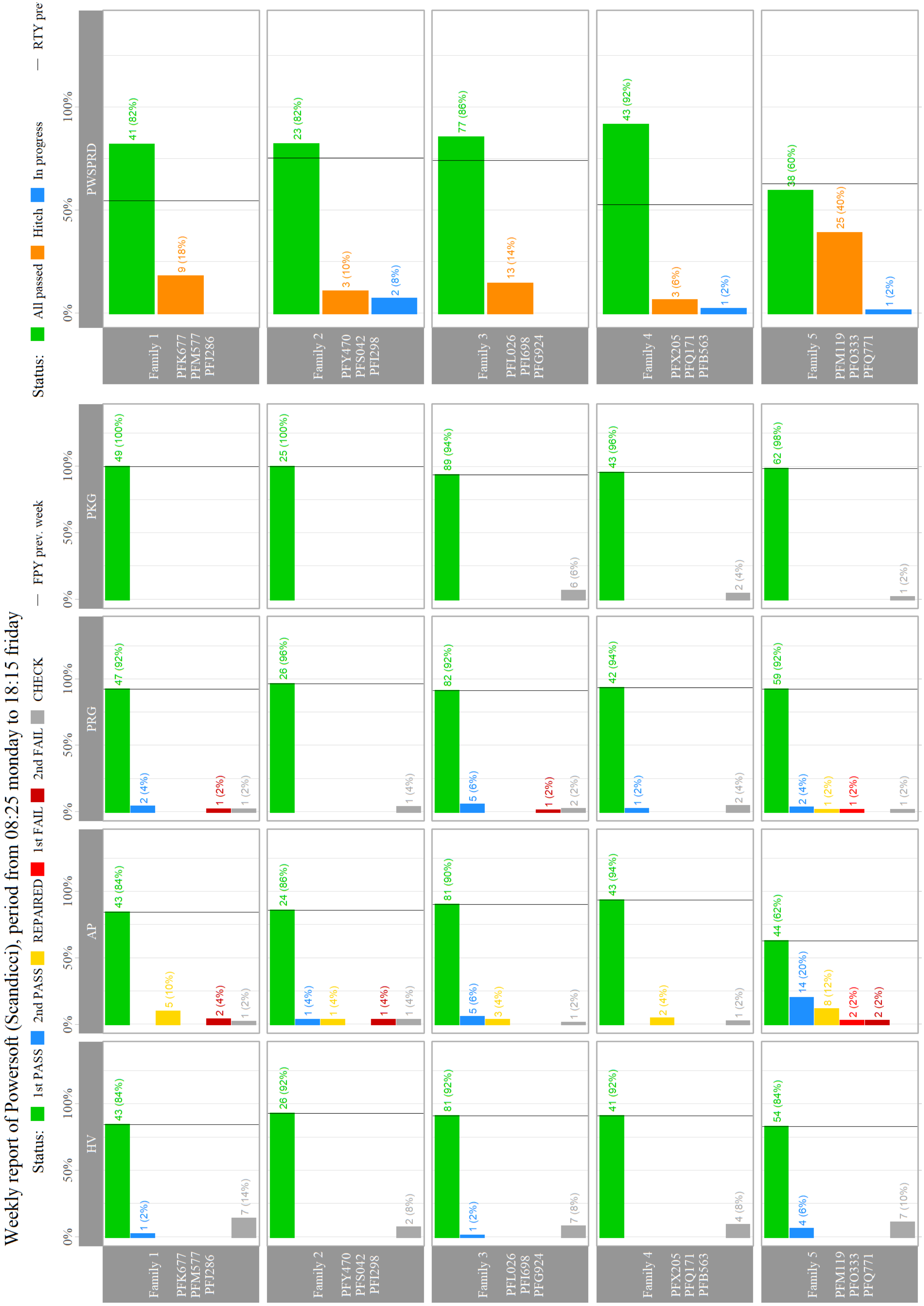


Figure 2.7: Example of dashboard visualization for weekly or monthly report. This dashboard is generated for every production site (Scandicci, Cortona, San Giovanni in Persiceto, Ronchi dei Legionari).

2.2.4 Production real-time report

All previous reports are sent by e-mail to the quality control officer and production managers with a delay of at least one day. We also implemented a real-time monitoring script, that shows on a tv screen the distribution of serial number's outcomes, in the same form of daily reports (Figure 2.5, next to Figure 2.6). The visualization is updated hourly and shows the counted serials in the site of Scandicci up to the current time.

In the case of empty query output, an automatic e-mail alerts the chief quality control officer.

2.2.5 Monthly and quarterly time series

A final analysis with a long-term perspective is the visualization of process performance metrics in time. This one is generated together with the monthly report and contains a synthetic line plot of the main KPIs for each family of products, test type, and production site. As well as daily, weekly, and monthly reports, time series visualizations describe both single-test and whole-production through different plots.

For the single-test focus, the main KPI is the FPY. An additional secondary metric was created, to separate product-related and test-related anomalies:

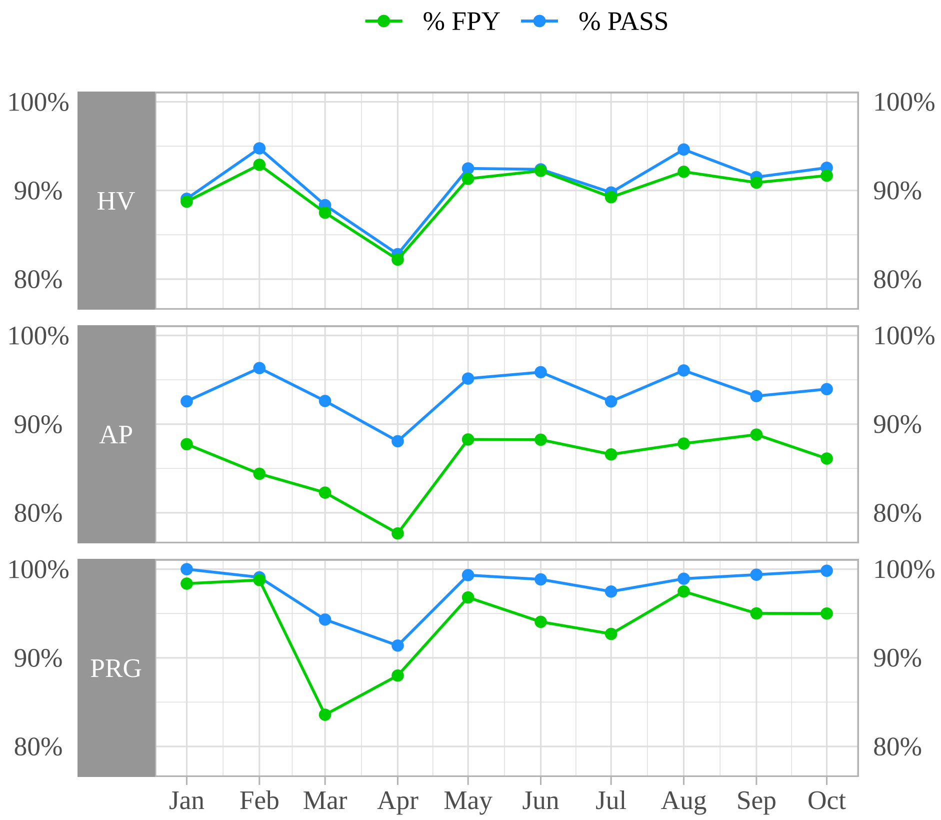
$$\text{PASS} = \frac{\text{num. of } \texttt{1st PASS} + \text{num. of } \texttt{2nd PASS}}{\text{total number of pieces put into process}} \tag{2.3}$$

For the whole-test focus, the only RTY is present in the line plot.

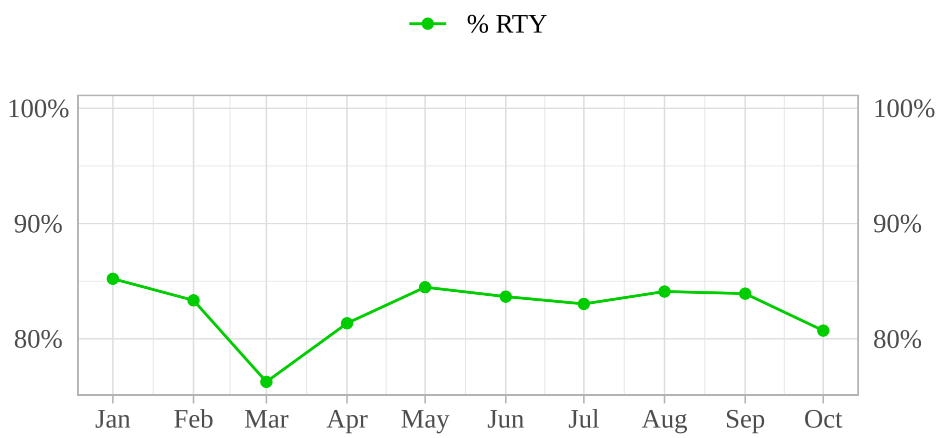
Observing and comparing the two KPI's trend, the analyst should be capable of detecting events such as

1. an improvement of the entire process (increasing FPY, PASS, and RTY);
2. a worsening of the entire process (decreasing FPY, PASS, and RTY);
3. an improvement in testing devices and procedures (increasing FPY and RTY, constant PASS);
4. an improvement in products quality but worsening of testing procedures (increasing PASS, constant FPY, and RTY);
5. a worsening in the testing devices (decreasing FPY, constant PASS).

Figures 2.8 on the next page and 2.9 on page 25 show exampling dashboards produced by this routine.

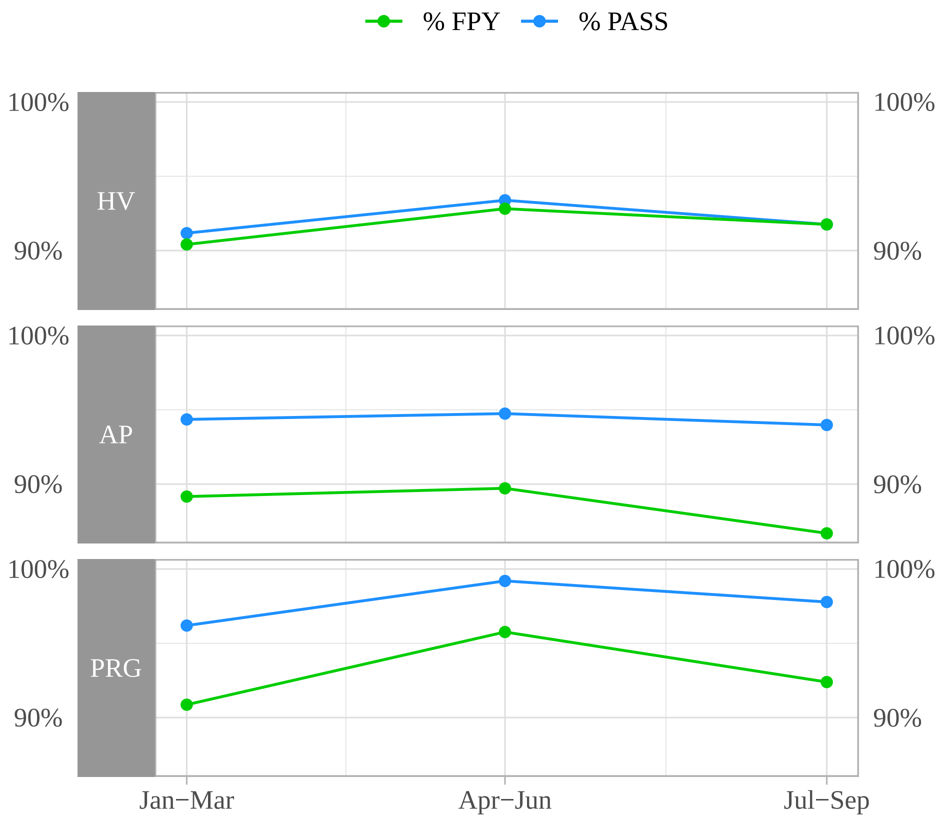


(a) Single-test focus.

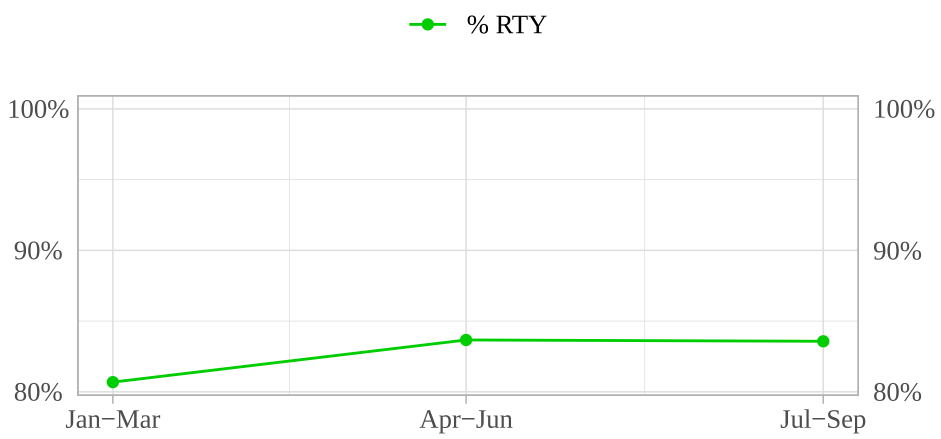


(b) Whole-process focus.

Figure 2.8: Example of monthly time series visualization on a family of products. *% FPY* is calculated using the definition in Equation (2.1), *% PASS* is calculated as in Equation (2.3). *% RTY* is calculated using the definition in Equation (2.2). The system generates these plots for each family and site, collected in one PDF document for each site.



(a) Single-test focus.



(b) Whole-process focus.

Figure 2.9: Example of quarterly time series visualization on a family of products. Same as the previous Figure.

2.3 Daily semi-automated process control

Daily monitoring of all the automatic e-mails is a routine that requires time and resources, in first place for the internal quality control management. Checking every single dashboard to look for anomalies in distributions implies to subtract time to other important activities such as certifications, test plan revisions, interactions with external suppliers. This cost of time is one of the main motivations for the automated alert system of significant FPY drops.

A broad range of possible events can cause FPY drops: a stock of defective components, multiple minor errors in the assembling procedures, problems in some testing device unit, cables, problematic test plans, and many more. The stakeholder chose FPY alarm as a first-level warning that once raised, should drive the receiver to investigate further the context that generated it.

The purpose of the project was to define and implementing a monitoring system that would run every morning to inspect the previous day production, and for each type and PF would detect eventual anomalies compared to the normal process.

2.3.1 Traditional process control solutions

Statistical Process Control (SPC) offers a set of tools for the supervision of discrete variables such as the number of defective pieces in a sampled subset of the process output [Montgomery, 2009]. For example, p-charts, np-charts, and u-charts are instances of the more general family of *Shewhart control charts*, initially developed by Walter Shewhart in the 1920s: researchers, engineers, and process analysts studied and applied them to a wide variety of contexts. Nowadays, they are a recognized standard for this kind of application. See Appendix A for an overview on control charts.

We started considering p-charts, one of the most used control charts for attribute data, typically used to monitor the number of nonconformities in production flow: the KPI that it tracks is

$$\hat{p}_t = \frac{D_t}{n_t} \tag{2.4}$$

where D_t is the number of defective units, and n_t is the number of total units in t-th sample. The expected value of this statistic is usually not known in practice. The analyst should estimate it from a reference period in which he is confident that the process was in-control, using the classical maximum likelihood estimator of $\bar{p}$:

$$\bar{p} = \frac{\sum_{i=1}^{m} D_i}{\sum_{i=1}^{m} n_i} \tag{2.5}$$

where $1, \ldots, m$ are the indices of in-control samples. Taking

$$\delta_t = L \cdot \sigma_{p_t} = L \cdot \sqrt{\frac{\bar{p}(1-\bar{p})}{n_t}} \tag{2.6}$$

where L is a parameter usually set equal to 3, δ, we can define the Upper and Lower Control Limits

$$\text{UCL} = \bar{p} + \delta_t \qquad \text{LCL} = \bar{p} - \delta_t \tag{2.7}$$

respectively.

Statistical Hypothesis Test

From a statistical perspective, this is comparable to a hypothesis test for Bernoulli data, where the null hypothesis is that the Data Generating Process (DGP) is

$$D_t \sim \text{Binomial}(\theta_t, n_t) \approx \text{Normal}\left(\theta_t, \frac{\theta_t(1-\theta_t)}{n_t}\right) \tag{2.8}$$

where the latter distribution denotes the Normal approximation (implicitly assuming a *sufficiently large* value $\theta_t(1-\theta_t)n_t$).

The nulla and alternative hypotheses are

$$\begin{cases} H_0 : \theta_t \leq \bar{p} \\ H_1 : \theta_t > \bar{p} \end{cases} \tag{2.9}$$

with significance level equal to

$$\alpha = 1 - \Phi(L)$$

where $\Phi(\bullet)$ is the Cumulative Distribution Function (CDF) of the standard Normal. For $L = 3$ (2), α is about 0.00135 (0.0228).

The test statistic $\hat{p}$ is compared with the critical value, equal to the Upper Control Limit (UCL), to chose if rejecting H_0 or not.

2.3.2 The case-specific application

There are relevant differences between the classical application of p-charts and the case in which we applied it:

1. Traditionally, control charts represent point statistics of item batches sampled from a production flow. In our case the sample is the whole production in a day;
2. Traditionally, control charts consider a statistic observed at regular time intervals. At Powersoft, the daily production is instead very heterogeneous: there are cases in which the production of a PF is limited to a short period (for example for a custom version that satisfied a partner's or customer's requirements); other PFs have inter-times between stock production of several weeks. These characteristics are also different among production plants;

3. Traditionally, the main application of p-charts is the observation of nonconformity percentage over the sample. In our case, the stakeholders consider the daily FPY as the statistic of interest:

$$\hat{p}_d = \mathrm{FPY}_d = \frac{\mathrm{NFP}_d}{\mathrm{NTOT}_d} = \frac{\text{number of daily } \texttt{1st PASS}}{\text{number of daily processed pieces}} \tag{2.10}$$

   so, the aim is not at keeping the KPI under the UCL, but at keeping it higher than the Lower Control Limit (LCL);

4. p-chart graphical representations allow the analyst to monitor the whole series of measures over time. Nonetheless, plotting a different series for every daily combination of PF, test type, and site would generate too many statistics. The solution we adopted is to limit the first Shewhart rule (see Appendix A) on each combination of PF and send an e-mail report only in case of violation of the LCL.

We could adopt several ad hoc approaches for each issue listed above, but finding and evaluating the right solution for every single PF or family would require much work, also with the perspective of maintaining the system. So, the decided to create a hybrid alarm system that would estimate reference FPY (and on the base of that an LCL), starting from the previous months' history, with also the possibility of setting a manual control limit, to manage particular cases.

2.3.3 Estimation of $\bar{p}$

Heterogeneity in Powersoft production is one of the major hurdles to the use of Equation (2.5) to estimate $\bar{p}$. In theory, it would be necessary to estimate a different $\bar{p}$ for each process, which is a unique combination of PF, test type, and site. One should calculate for each combination

$$\bar{p}_{g,t,s} = \frac{\sum_{d \in T} \mathrm{NFP}_{d,g,t,s}}{\sum_{d \in T} \mathrm{NTOT}_{d,g,t,s}} \tag{2.11}$$

where T is a time window, d is the day, g is the product PF, t is the test type and s is the site. In several cases, however, there is not sufficient data to create a defined picture of the in-control process at this level.

Training and validation sets

To evaluate the best approach to calculate $\bar{p}$ for each process, we reshaped the dataset in the form of Table 2.4 and selected only observations in May, June, and July 2019 with more than 15 items produced. This choice was lead by several aspects:

- the database cover several years, but in practice, there are so many differences (e.g., use of different labels, how the records where organized, when

Date	PF	Family	System Type	System Site	NFP	NTOT	FPY
2019-02-04	PF144	Family 19	PRG	PWSPRD	21	22	0.95
2019-05-10	PF155	Family 21	COF	SGP	1	7	0.14
2019-05-17	PF165	Family 17	AP	CRT	48	62	0.77
2019-08-07	PF109	Family 18	AP	PWSPRD	2	2	1.00
2019-09-06	PF044	Family 9	CFP	SGP	68	74	0.92
2019-09-09	PF068	Family 9	COF	SGP	49	56	0.88
2019-09-23	PF142	Family 17	PRG	CRT	45	45	1.00

Table 2.4: Fictitious sample of training set for estimation of expected FPY of process. NFP is the count of `1st PASS` items, NTOT is the total count of items that have started the testing process.

and how the operators should report their actions) that the only consistent data were those in the latest year;

- to evaluate the goodness of fit, it was necessary to use the estimated $\bar{p}$ on a period more recent than the training one. Considering months from May to July allowed to use remaining months as validation set;
- filtering only in-control processes from the entire list required manual work and much time with a process expert. As an alternative, we considered only those days in which the site produced at least 15 pieces. With this approach, we mitigated the effect of prototyping, calibration, and work on remaining defective pieces.

We applied then the estimates computed over May, June, and July to the data in August and September. The final size of the training set was 1496 rows, representing 173 different PFs belonging to 18 families.

Covariates

Calculating a $\bar{p}$ for each process requires to understand the possible sources of variability between them. They can be seen as categorization criteria, or as covariates, but in general consist of:

Site production site:

- Scandicci (PWSPRD),
- Cortona (CRT),
- Ronchi dei Legionari (RDL),
- San Giovanni in Persiceto (SGP);

Type test type: HV, AP, COF, CFP, PRG (see Section 1.3 on page 3);

Family product highest level classification;

PF product lowest level classification;

Time two variability levels:

- short-term variability (days);
- long-term variability (weeks, months).

Aggregation by family, type and site

The issue of high variability led to the first-step solution of aggregating groups by family instead of PF. This approach implicitly assumes that the FPY ($\bar{p}$) of each process is very similar between same-family products

$$\bar{p}_{g,t,s} = \frac{\sum_{d \in T} \mathrm{NFP}_{d,g,t,s}}{\sum_{d \in T} \mathrm{NTOT}_{d,g,t,s}} \tag{2.12}$$

where T is the training period, d is the day, g is the product family, t is the test type and s is the manufacturing site.

Another issue is that many combinations of family, type, and site in the validation set are missing from the training set. The most immediate solution was to estimate missing group FPY from the most similar training data:

$$\bar{p}_{g,t,s} = \begin{cases} \dfrac{\sum_{d \in T} \mathrm{NFP}_{d,g,t,s}}{\sum_{d \in T} \mathrm{NTOT}_{d,g,t,s}} & \text{if neither the family } (g) \text{ and the type } (t) \text{ are missing in the training set} \\[2ex] \dfrac{\sum_{d \in T} \sum_{g} \mathrm{NFP}_{d,g,t,s}}{\sum_{d \in T} \sum_{g} \mathrm{NTOT}_{d,g,t,s}} & \text{if only the family } (g) \text{ is missing in the training set} \\[2ex] \dfrac{\sum_{d \in T} \sum_{g} \sum_{t} \mathrm{NFP}_{d,g,t,s}}{\sum_{d \in T} \sum_{g} \sum_{t} \mathrm{NTOT}_{d,g,t,s}} & \text{if both the family } (g) \text{ and the type } (t) \text{ are missing in the training set} \end{cases} \tag{2.13}$$

For example:

- family ABC has been tested at the HV test type in the site of Ronchi Dei Legionari several times in the period covered by the training set: we can calculate the respective in-control FPY ($\bar{p}_{\mathrm{ABC,HV,RDL}}$) as the ratio between the number of `1st PASS` pieces and the total number of worked pieces, over the ABC family, at the HV test, at the Ronchi Dei Legionari plant;
- family XYZ has never undergone the AP test at the Scandicci plant in the period covered by the training set: the respective FPY ($\bar{p}_{\mathrm{XYZ,AP,PWS}}$) is equal to the ratio between the number of `1st PASS` pieces and the total number of worked pieces, at the AP test at the site of Scandicci, independently from the family;

- the Cortona plant never executed the AP test in the period covered by the training set: all the FPYs of the families tested at the AP test in Cortona ($\bar{p}_{\mathrm{g,AP,CRT}}$) are equal to the ratio between the number of `1st PASS` pieces and the total number of worked pieces, in the site of Cortona, independently from the family and test type.

Using these aggregations, we could estimate each $\bar{p}$ of the validation set. This approach faced both the problem of high variability for low-volume groups and the estimation for missing combinations in the training set. The effects on the $\bar{p}$ estimates are visible in Figure 2.10 on the following page. If estimates were equivalent for the two methods, all points would have lied on the bisector (x-coordinates would be equal to y-coordinates). Points that lie beyond the dashed line represent a combination of PF, test type and site for which the *estimation grouping by PF* (Equation 2.11) has a lower value than *estimation grouping by family* (Equation 2.13). Vice versa, points below the dashed line indicate a combination in which the former estimate is higher than the latter. The Figure shows that many estimations that with the by-PF aggregation had low values, rise with the by-family aggregation: many of them are estimates done on low-volume groups (represented by the point size). Besides, PFs missing from the training set (represented by the orange crosses) have been all estimated.

Fixed effect model

As an alternative to the raw calculation of rates for each subset, we estimated a logistic regression model.

Considering the variable of interest, in our case NFP_d (the number of `1st PASS` pieces of a single PF in a daily test), if the process is in-control we have that

$$\mathrm{NFP}_d \sim \mathrm{Binomial}(\bar{p},\, \mathrm{NTOT}_d) \tag{2.14}$$

where NTOT_d is the number of daily processed pieces. Given a set of k explanatory variables $X_1, \ldots X_k$ associated to each observation (in our case they could be *site*, *type* and the others at page 29), a set of parameter $\beta_1, \ldots \beta_k$ fixed for the entire model (from this *fixed effects*), and a constant parameter β_0, the logistic regression model assumes that

$$\bar{p} = f\left(\beta_0 + \sum_{i=1}^{k} \beta_k X_k\right) \quad \text{or equivalently} \quad f^{-1}(\bar{p}) = \beta_0 + \sum_{i=1}^{k} \beta_k X_k \tag{2.15}$$

where $f(\bullet)$ is

$$f(x) = \frac{e^x}{1+e^x} \quad \text{or equivalently} \quad f^{-1}(x) = \mathrm{logit}(x) = \log\left(\frac{x}{1-x}\right) \tag{2.16}$$

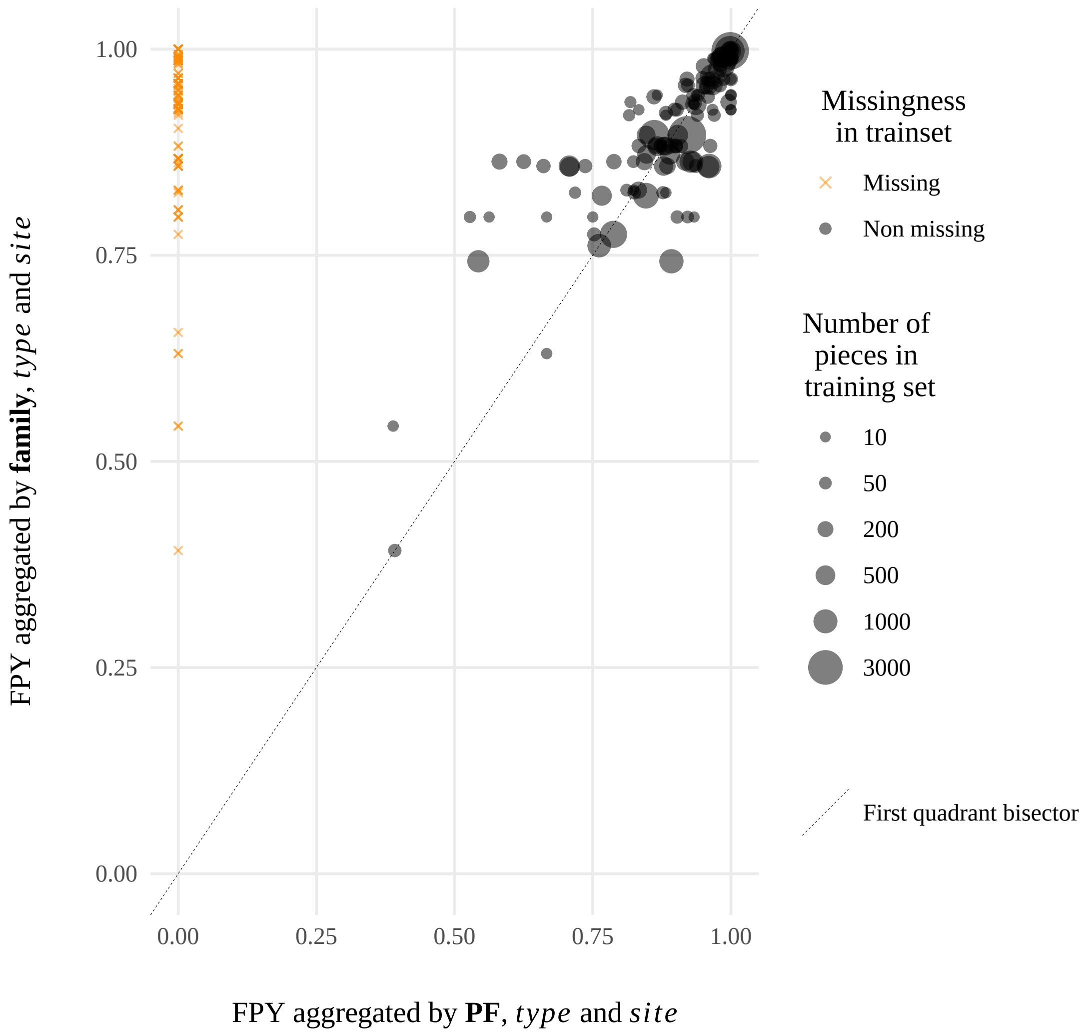


Figure 2.10: Graphical comparison of $\bar{p}$ estimated with PF aggregation versus family aggregation. Each point corresponds to a unique combination of PF, type and site in the validation set. $\bar{p}$ estimated using equation 2.11 on the x-axis, $\bar{p}$ estimated using equation 2.13 on the y-axis. Missing PFs in the training set are marked with a different symbol on the left axis.

X_i variables can be both continuous (e.g. temperature, height) or categorical (e.g. gender, site). For the latter case, particular mathematical tricks must be used for the estimation of parameter β_i [Fahrmeir et al., 2015, p. 31]. For sake of brevity, next formulas synthesize with β_{variable} to indicate a coefficient that "jumps" to the value of the variable modality. For example

$$\beta_{\text{site}} = \begin{cases} \beta_{\text{PWSPRD}} & \text{if } X_{\text{site}} = \text{PWSPRD} \\ \beta_{\text{CRT}} & \text{if } X_{\text{site}} = \text{CRT} \\ \beta_{\text{RDL}} & \text{if } X_{\text{site}} = \text{RDL} \\ \beta_{\text{SGP}} & \text{if } X_{\text{site}} = \text{SGP} \end{cases}$$

The number of parameters k in the model specification indicates how precisely the process is focused. The two extreme cases are:

Null model ***($k = 1$)*** The only parameter is the constant:

$$\text{logit}(\bar{p}) = \beta_0 \tag{2.17}$$

In this way the entire production is assumed to have the same behavior, without distinctions between sites and test types, that is obviously very unlikely, but in general it is used as a benchmark.

Estimating β_0 can be done easily as

$$\hat{\beta}_0 = \text{logit}\left(\frac{\sum_{t \in T} \text{NFP}_d}{\sum_{t \in T} \text{NTOT}_d}\right)$$

Saturated model ***($k = 2^{\textit{num. of variables}}$)*** The model is completely specified, dividing each possible subgroup in the finest way:

$$\text{logit}(\bar{p}) = \beta_0 + \beta_{\text{site}} + \beta_{\text{type}} + \beta_{\text{PF}} + \beta_{\text{site,type}} + \beta_{\text{site,PF}} + \dots \tag{2.18}$$

This model results in the same predictions of equation 2.11, so it is used as a benchmark as well.

Between these two extreme cases, several model specifications were tested: often the estimation algorithm did not converge because fitted probabilities were numerically equal to 0 or 1, preventing to reach a high number of covariates and interactions. The models that we managed to estimate and compared were:

$$\text{logit}(\bar{p}) = \beta_0 + \beta_{\text{site}} + \beta_{\text{type}} \tag{2.19}$$

$$\text{logit}(\bar{p}) = \beta_0 + \beta_{\text{site}} + \beta_{\text{type}} + \beta_{\text{family}} \tag{2.20}$$

$$\text{logit}(\bar{p}) = \beta_0 + \beta_{\text{site}} + \beta_{\text{type}} + \beta_{\text{family}} + \beta_{\text{month}} \tag{2.21}$$

$$\text{logit}(\bar{p}) = \beta_0 + \beta_{\text{site}} + \beta_{\text{type}} + \beta_{\text{family}} + \beta_{\text{month}} + \beta_{\text{site, type}} \tag{2.22}$$

Table 2.5 shows the results of a likelihood ratio test between the most relevant estimated models. The model number refers to the specifications above. We can see that all the tests lead to the rejection of the null hypothesis: this evidence the significant difference in terms of the FPY between the subgroups generated adding explanatory variables.

Model	Resid. Deg. Fr.	Resid. Dev	Δ Deg. Fr.	Deviance	Pr(>Chi)
(Null)	1495	12061.23			
(2.19)	1486	5376.75	9	6684.48	0.0000
(2.20)	1469	4990.62	17	386.13	0.0000
(2.21)	1467	4888.05	2	102.57	0.0000
(2.22)	1457	4517.98	10	370.07	0.0000

Table 2.5: ANOVA table for FPY models. They have then been compared with a likelihood ratio test, and the resulting p-value is represented on the last column.

The empirical evidence leads to the choice of the last specification (equation 2.22), also looking at the Bayesian Information Criterion (BIC) (see Table 2.6), that should trade-off the balance between the goodness of fit and complexity of the model.

Model	Deg. Fr.	BIC
(Null)	1	14189.50
(2.19)	10	7570.82
(2.20)	27	7308.96
(2.21)	29	7221.01
(2.22)	39	6924.05

Table 2.6: BIC comparison among FPY models.

Since we decompose the effect of the factors by coefficients, we can combine them as needed. We are able to obtain predictions on arbitrary combinations of families, types, and sites, even if they miss from the training set: an improvement in comparison with the estimation of FPYs by clustering similar processes. Still, this approach does not permit to estimate $\bar{p}$ on families that are completely missing in the training time window. In theory, one could use a model with the *family* factor (Specification 2.22) to predict non-missing families, and a model without the *family* factor (Specification 2.19) to predict unknown families' FPY. It would be equivalent to making a sort of mean of the Specification (2.22) predictions by type and site. Though, this could lead to bias in the parameter estimation due to the unexplained heterogeneity between family groups.

Mixed effect model

To obtain a consistent estimation of the average effect, we had to recur to the framework of Generalized Linear Mixed Models (GLMM). GLMMs are useful when one aims at taking apart the effect of a variable *between* groups from the effect *within* groups. A complete overview of these models can be found in [Fahrmeir et al., 2015, ch. 7].

After the assessment of which could be considered a fixed effect and which random, and keeping in account the importance of having a parsimonious model in term of number of parameters, the final model specification that we tested was

$$\text{logit}(\bar{p}) = \beta_0 + \beta_{\text{site}} + \beta_{\text{type}} + \beta_{\text{type} \mid \text{family}} + \beta_{\text{month}} \tag{2.23}$$

Only $\beta_{\text{type} \mid \text{family}}$ is a random effect and it is distributed as

$$\beta_{\text{family} \mid \text{type}} \sim \text{Normal}(0, \sigma^2_{\beta_{\text{type}}})$$

while other coefficients are assumed fixed.

A problem immediately emerged in the estimation with the use of `lme4` R package [Bates et al., 2015]: often, the iterative approximation did not converge, returning a "matrix singularity" error. Removing particular families from the training set sometimes fixed the convergence issue. However, the final implementation, that update parameters monthly, could not risk failing with an error or even worse returning biased estimations, so we opted to try an alternative solution.

Bayesian mixed effect model

A Bayesian approach was then attempted, since it does not require iterative algorithms, but estimates the parameters through Monte Carlo Markov Chain (MCMC) algorithms. Bayesian inference is a branch of Statistics based on the assumption that probability is a representation of our information over the world: we start from some a priori knowledge and gradually update it based on the information available, using the Bayes formula. A complete overview of Bayesian GLMMs can be found in [Fahrmeir et al., 2015, section 7.4]. All the next models were estimated using the `rstanarm` package [Goodrich et al., 2018], an interface for the Stan probabilistic programming language [Carpenter et al., 2017].

Starting from the model specification (2.23), we tested several parametrizations to get a good combination between computational time and good convergence of the chain. The final parametrization consisted of

Priors

$$\beta_0 \sim \text{Normal}(2.197, 10.0) \quad \text{(intercept)}$$
$$\beta_i \sim \text{Laplace}(0.0, 10.0) \quad \text{where } i \neq 0$$

$$\beta_i \perp \beta_j \quad \text{when } i \neq j$$

MCMC parameters

num. of chains $= 4$
num. of iterations $= 2000$ (1000 warm-up $+$ 1000 regular)
no thinning

This same parameters will be used also in the future updating of the model. The choice for the intercept prior mean was calculated as the logit function of 0.9

$$\log\left(\frac{0.9}{1-0.9}\right) = 2.197$$

This prior mean is due to the analytic consideration for which the standard FPY for this kind of tests should be about 0.9. The prior parameters could be anytime changed on the base of new evidences.

In the end, we used the Bayesian estimated posteriors' expected values to define a predicted $\bar{p}$ for each combination of family, test type and site. For the *month* variable, we used only the most recent month's estimated coefficient for all the predictions. For families observed in the training dataset, we used the prediction given by the complete specification (2.23), while for families that we did not observe in the training we assumed $\beta_{\text{type} \mid \text{family}} = 0$ and so

$$\text{logit}\left(\bar{p}_{g,t,s}\right) = \begin{cases} \beta_0 + \beta_t + \beta_{t|g} + \beta_s + \beta_{\text{last month}} & \text{if the family } (g) \text{ is non missing in the training set} \\ \beta_0 + \beta_t + \beta_s + \beta_{\text{last month}} & \text{otherwise} \end{cases} \tag{2.24}$$

where g is the family, t is the test type and s is the site.

Estimated coefficients were analysed on their own (see Figures 2.11 for fixed effects, Figures 2.12 for random effects) and compared with the results of the raw aggregation by family, type and site (see Figure 2.13). Fixed effects show significant variability in terms of time dynamics (see the *month* variable coefficients, coherently with what emerged from the Generalized Linear Models (GLM) models) and different production sites. About test types, the difference is more evident between two groups: types 01, 02, 03, and types 04, 05.

About Figure 2.13, the Bayesian model predictions evidence the so-called shrinking effect: random effects of small groups tend to the means of bigger

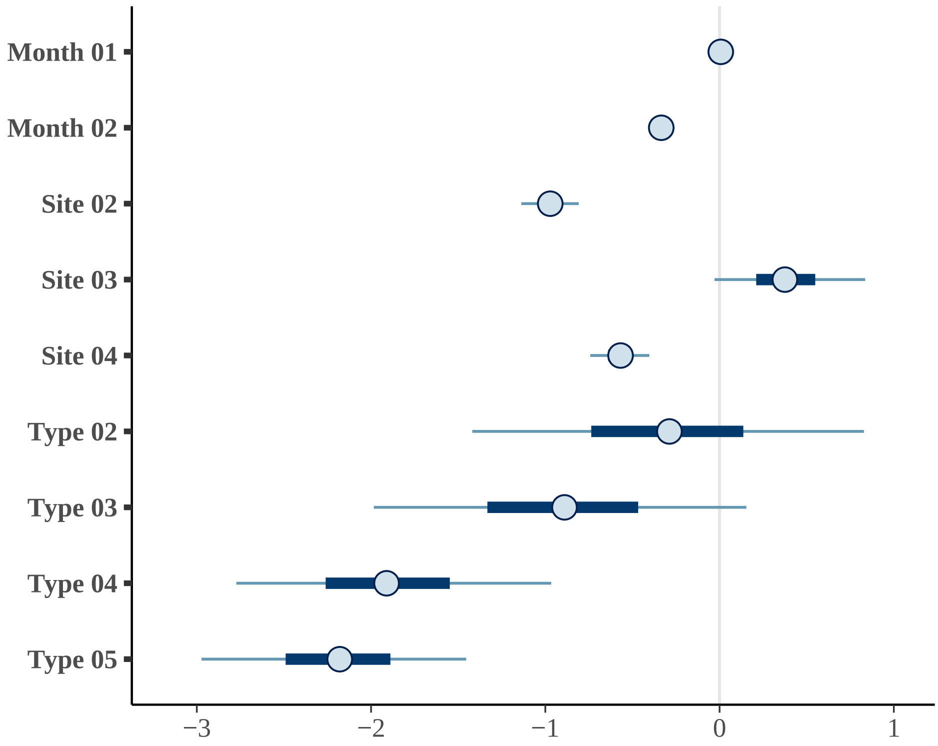


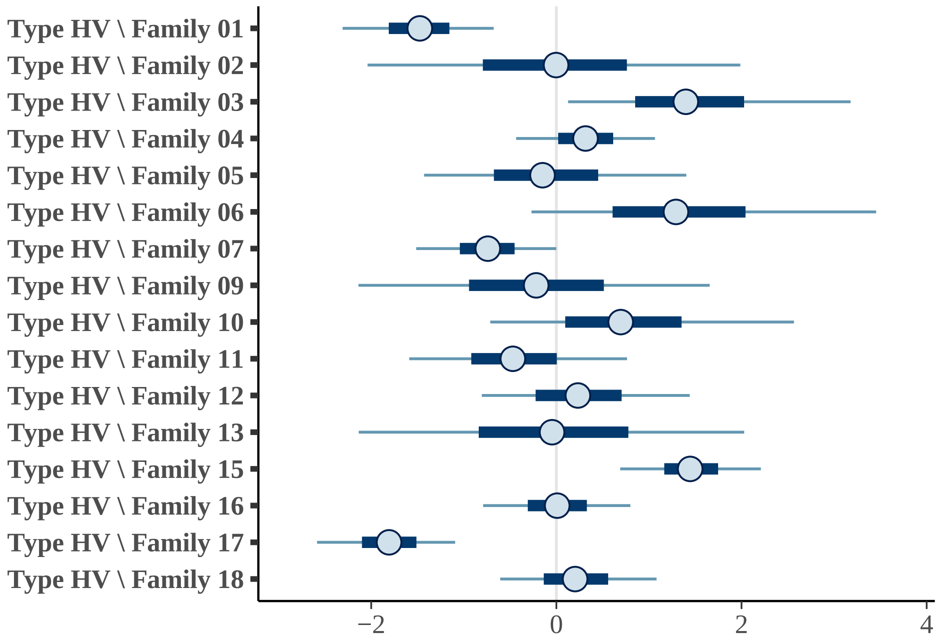
Figure 2.11: FPY mixed effects model fixed effects coefficients.

groups. A cluster of items shows lower FPY predicted values in the GLMM model, in comparison with the initial estimates: this is due to the particularity of the products' families, that in the training set were in a phase of prototyping.

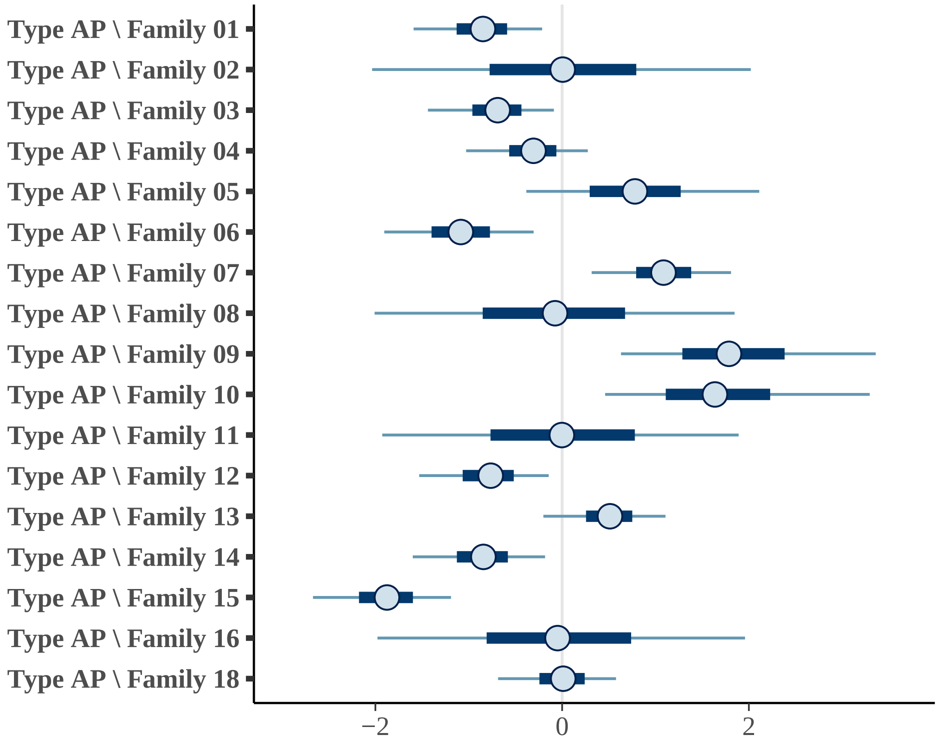
2.3.4 Comparison between control limits and the binomial test

Section 2.3.1 describes the parallelism between SPC and Statistical Hypothesis Test Theory, with particular focus on the control limits' definition for dichotomous data. The Normal approximation of $\hat{p}_d$ distribution is usually acceptable in traditional high-volume productions but is questionable in Powersoft, where single processes often generate a small number of outputs (less than 30 pieces). In practice, the difference between the two distributions depends on the relation between n and p: Figure 2.15 shows several combinations of n and p and the relative differences between the critical values for a constant-alpha hypothesis test ($\alpha = 0.05$).

Notice that exact the binomial test is more *conservative* than the control-limit test: every time that the observed statistic $\hat{p}_t$ bypasses the binomial critical value, it is certainly out of the control limits. It is a result of the fact that, when assumptions for the approximation are violated (totally or partially), the Normal distribution assigns a non-zero probability to values beyond the Binomial

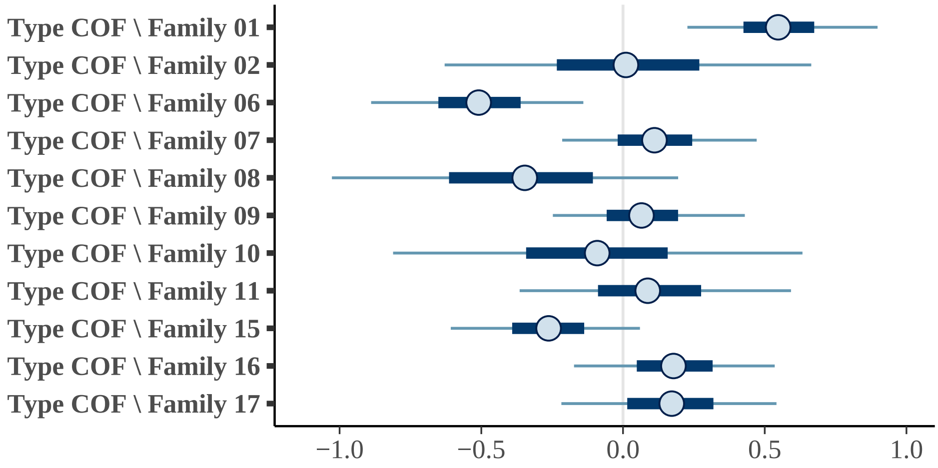


(a) Type HV.

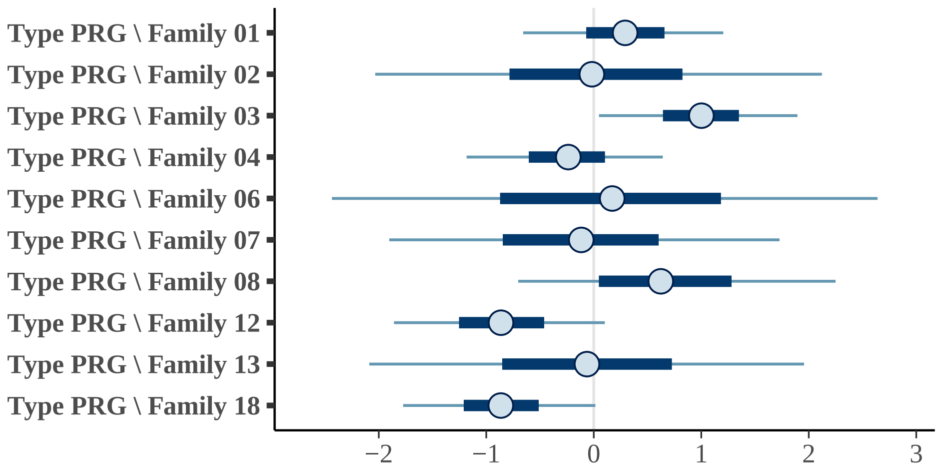


(b) Type AP.

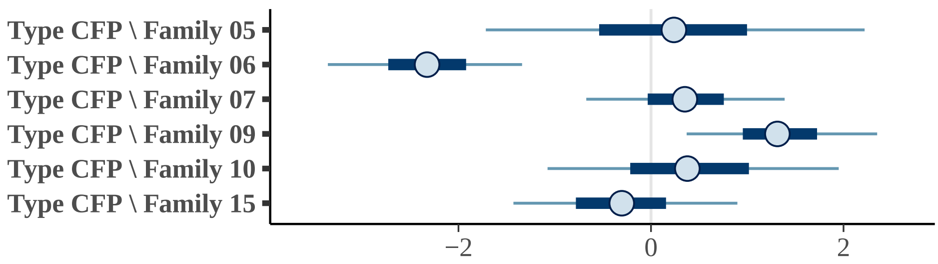
Figure 2.12: FPY model random effects coefficients (part 1).



(c) Type COF.



(d) Type PRG.



(e) Type CFP.

Figure 2.12: FPY model random effects coefficients (part 2).

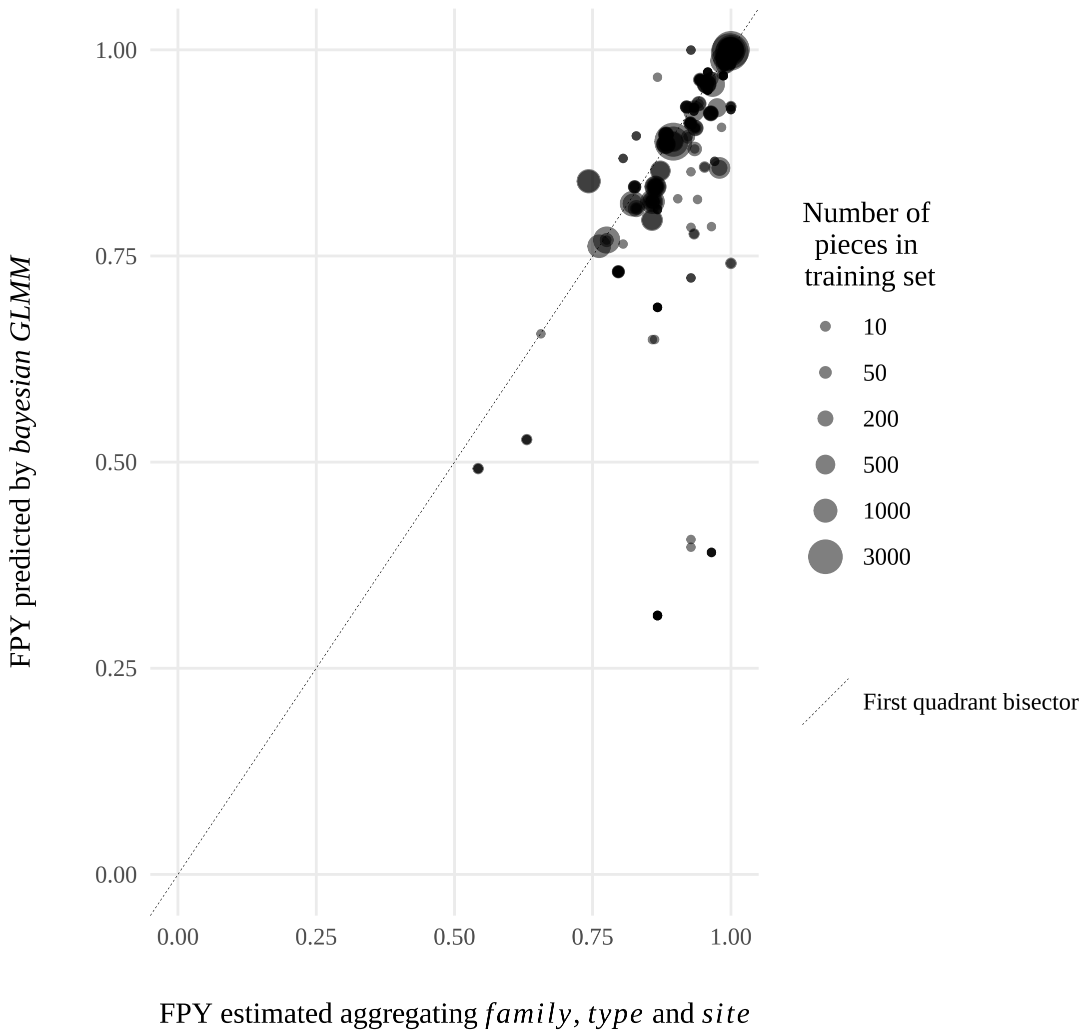


Figure 2.13: Comparison between FPY estimated grouping data by family, site and type and FPY predicted from the bayesian GLMM model.

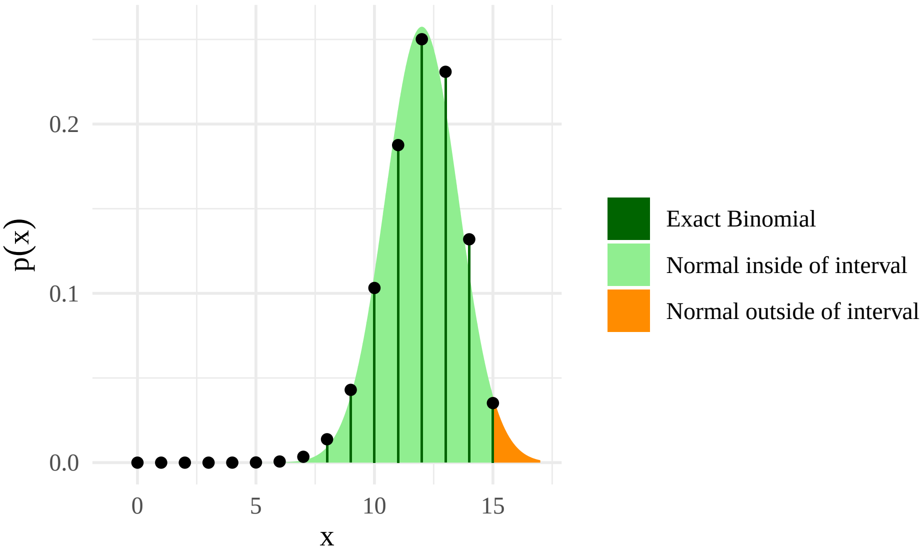


Figure 2.14: Exact binomial ($p = 0.8$; $n = 15$) versus Normal-approximating distribution. While the exact Binomial is a discrete distribution (its $p(x)$ is defined only for values $x \in \mathbb{N}$) the Normal distribution is continuous and its $p(x)$ has a value for each $x \in \mathbb{R}$.

distribution support (see Figure 2.14).

Given this, we performed an analysis of validation data to understand if an above kind of violation can bias the tests. Since it was not possible defining the "in-control" or "out-control" status manually for every single record in the validation set, we replace the number of `1st PASS` with a simulated number based on the assumption in equation 2.14. We defined a dummy variable to indicate an in-control (IC) or out-of-control ($\bar{\text{IC}}$) process, assigned randomly to observations with a proportion of 25% of $\bar{\text{IC}}$, and 75% of IC. Then, for every record the number of `1st PASS` pieces NFP_i was sampled as

$$\text{NFP}_i \sim \begin{cases} \text{Binomial}(\bar{p},\ \text{NTOT}_i) & \text{if IC} \\ \text{Binomial}(k \cdot \bar{p},\ \text{NTOT}_i) & \text{if } \bar{\text{IC}} \end{cases} \tag{2.25}$$

where k is a constant between 0 and 1. Several scenarios were observed, starting from the completely in-control process ($k = 1.0$) to a drop of the 50% of the production ($k = 0.5$).

Table 2.7 on page 43 reports simulations results, with a comparison of accuracy, specificity, sensitivity (with $\alpha = 1 - \Phi(3) = 0.0013$) and Area Under Curve (AUC). Sensitivity is defined as

$$\text{Sensitivity} = \frac{\text{num. of correctly predicted out-control proc.}}{\text{num. of real out-control proc.}} \tag{2.26}$$

that is, in statistical terms, the power of the test

$$\gamma = 1 - \beta$$

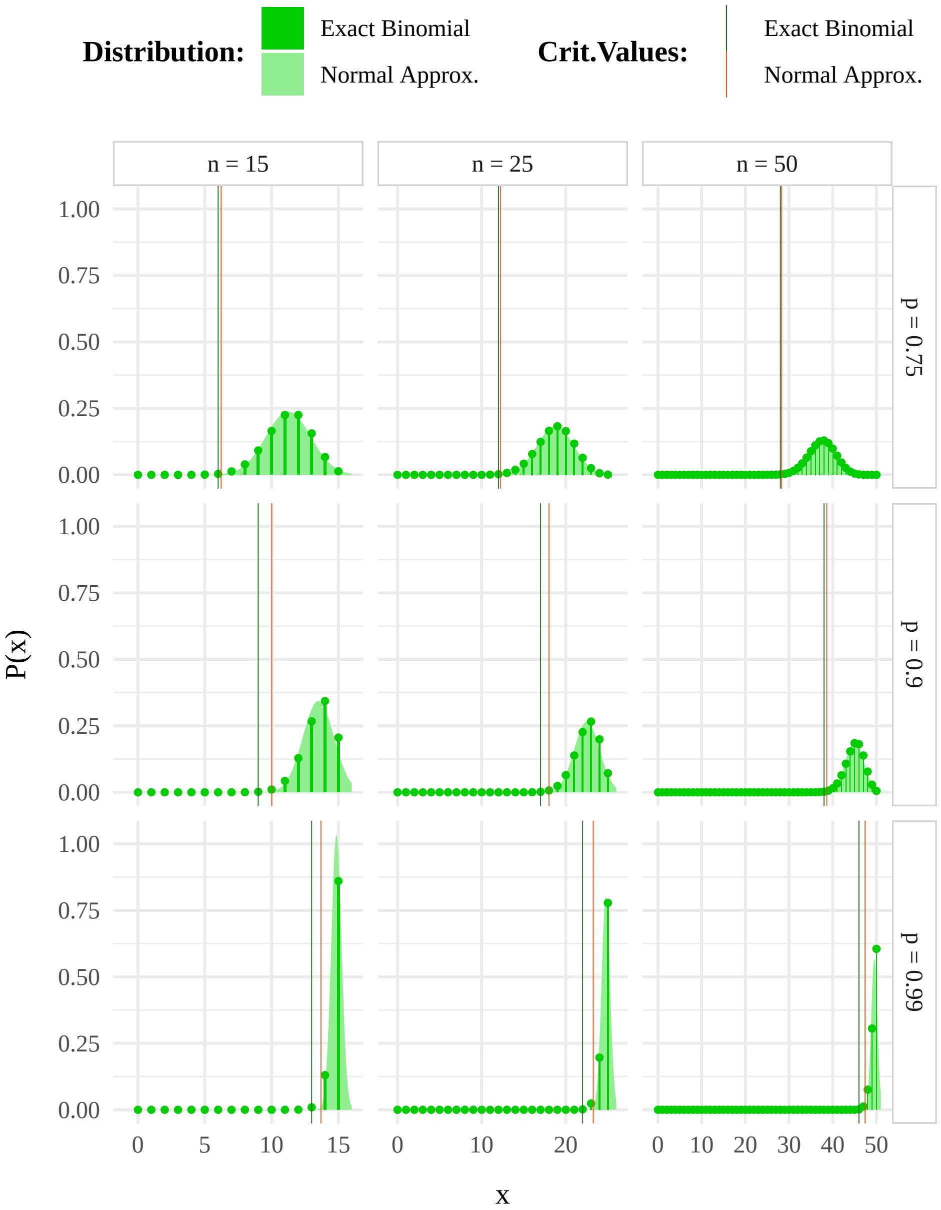


Figure 2.15: Comparison between control limit and exact binomial critical value. Binomial distribution parameters n and p change with columns and rows.

k	Control-Limit Test				Exact Binomial Test			
	Accur.	Sens.	Spec.	AUC	Accur.	Sens.	Spec.	AUC
1.0 (in-control)	0.995	0/0	0.995	//	1.000	0/0	1.000	//
0.9	0.87	0.42	0.99	0.85	0.85	0.30	1.00	0.86
0.8	0.91	0.63	0.99	0.95	0.90	0.53	1.00	0.95
0.7	0.95	0.78	0.99	0.97	0.94	0.73	1.00	0.98
0.5	0.95	0.78	0.99	0.97	0.94	0.73	1.00	0.98

Table 2.7: Results of simulations to compare control limits versus exact binomial tests.

where β is the probability of a Type II error (accepting H_0 when H_1 is correct). The higher specificity is, the more the test is efficient in detecting out-of-control processes. Specificity is defined as

$$\text{Specificity} = \frac{\text{num. of correctly predicted in-control proc.}}{\text{num. of real in-control proc.}} \tag{2.27}$$

that is, in statistical terms, the probability

$$1 - \alpha$$

where α is the probability of a Type I error (accepting H_1 when H_0 is correct). The higher the sensitivity, the less the test will result in false alarms. The usual indices that synthesize the goodness of prediction are *accuracy* (the percentage of correctly classified items on the total), and the AUC (a measure of how good the model classifies items in terms of both specificity and sensitivity when I let vary the classification threshold, that is the parameter α).

Looking at Table 2.7, we evince that the simulated results confirm theoretical considerations. Control-limit tests return a higher rate of alarms, both false and correct, as a consequence of the higher power and lower specificity of the control-limit test, throughout all the scenarios. In a "completely in-control" condition, the binomial test never provides false alarms, while the control-limit test has a rate of false alarms of 5 per 1000. When the k parameter decreases, the specificity remains almost constant at 99% for the control limits, and there are 0 false alarms for the binomial test. In terms of raw accuracy, control limits seem slightly better, vice versa, in terms of AUC, the binomial test always has marginally better results.

We arrived at the final choice driven by considerations on the final use of the tests: if the binomial test would have shown a notable improvement in terms of reliability, it could replace the traditional SPC Shewhart's rule. In a context where false alarms' cost was very high, a more in-depth analysis could lead to evaluate a switch in favor of this more mathematically correct approach. In this case, the cost of false alarms in relation to the modest improvement brought by the use of the exact binomial test is believed not sufficient to justify a methodology change; so, the final implementation adopted the traditional control limits.

2.3.5 Practical implementation

To manage parameters of the alarms with a high level of detail, we created a configuration file in CSV format. With the granularity level of single PF, site, and test type, it is possible to set:

- a flag for the activation (or deactivation) of the FPY check;
- a flag to indicate whether the scripts should calculate alarm limits automatically (using the estimation by grouping, or the GLMM model) or use a manual control limit;
- the manual limit, in case the previous flag indicated to use it.

An example configuration file can be seen in Table 2.8. Here the product PF00X1 is tested on the automatically calculated control limit, the product PF00Y2 has a manual limit (equal to 0.75), and we do not monitor the product PF00Z3 at all.

We developed the implementation in two parallel systems: a "basic" version adopting the FPY estimated aggregating family, type and site data, and an "advanced" version using the predictions of the GLMM model.

In-control FPYs are estimated monthly and stored in a CSV file. When the daily routine generates a complete overview of the previous day production, another script checks for possible anomalies. In case it finds some processes going outside the control limits, it sends an e-mail to the quality control manager with a focus on the failed test (see Figure 2.16 on the following page). All alerts are stored.

This system is born to simplify the supervision of production flows, not to replace an accurate and in-depth analysis of daily reports made by area experts. So, it was kept separated and autonomous from the daily reporting system described in Section 2.2.

PF	Family	Type	Site	Alarm	AutoAlarm	ManualLimit
PF00X1	Family XYZ	HV	PWSPRD	TRUE	TRUE	NA
PF00Y2	Family YZX	AP	CRT	TRUE	FALSE	0.75
PF00Z3	Family ZXY	COF	RDL	FALSE	TRUE	NA
...	...	...	...	...	...	...

Table 2.8: Partial example of configuration table for the daily FPY alarm system.

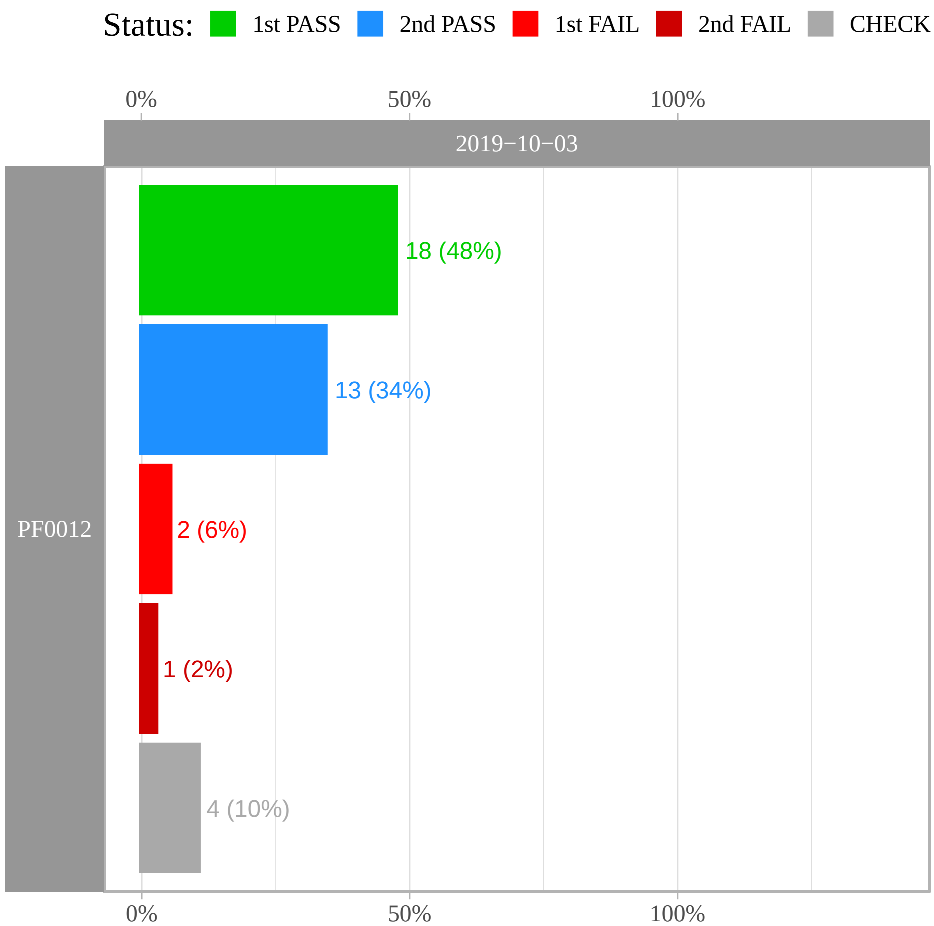


Figure 2.16: Example of FPY alarm report.

CHAPTER

THREE

TESTING DEVICE MONITORING

Testing procedures require efficient and fast devices, as well as trained operators. In cases where testing tools emit and register high voltage current flows, it is also necessary to guarantee the safety of both the internal operators and the final consumers. At Powersoft, these requirements are satisfied through a battery of testing tools. National and international norms define standards to align tests across the industries for the electrical safety of products in commerce. Often companies have the duty not only to perform a set of measures on their instruments but even to monitor them in time, to intercept possible anomalies.

3.1 HV test procedure

One of the tests adopted to guarantee the electrical safety of its testing devices and reliability of its results is the so-called `DAILYHV`. This test guarantees the good insulation of testing tools and operator safety in the case of short-circuit: Appendix B reports an in-depth description of the test. It consists of three measurements:

Alternating Current Withstanding Voltage (ACWV) A measure of current flow, in milliamperes.

Ground Bonding Resistance (GBON) Two measures of resistance (GBON1, GBON2), in milliohms.

Each of these values is measured each morning. The procedure is partially automated: it is sufficient to connect a little box to the testing device so that it calculates each of the three measures and compares them to the specification limits. If at least one on three lies beyond them, the test fails. This procedure must succeed before any other use of the equipment: operators have the directive to repeat it until it is a pass.

These tests are standard in the electronics industry, and many certification companies sell the instruments to perform them. Despite that, thanks to its expertise, Powersoft builds and calibrates its testing boxes to maintain efficiency and customization with low economic impact.

Technicians calculate the specification limits on the base of the structure of the box. Initially, there were few boxes for the whole plant, so that the operators used the same one across several Device Under Test (DUT). Later, the number of testing boxes increased, and currently, each there is a specific testing box to equip each DUT.

3.1.1 Sources of variability

As all measures in physics, also ACWV and GBON are affected by several internal and external sources of variability: some can be mitigated but never deleted at all. Among them:

Cables `DAILYHV` and usual HV tests employ the same cables to connect the amplifier to the device. They are constantly subjected to wear, and are often changed;

Internal temperature The internal components of the boxes for the test are small (in the order of millimeters), intending to reduce heat generated by the high voltage current. This fact implies a relevant deformation and changes in material properties due to temperature;

Ambient temperature Production sites' environment has not constant temperature: in summer, air conditioning mitigates heat marginally, but the inside of the buildings undergo seasonal temperature variations. In objects of millimetric size, this variation can have significant effects, in the modification of materials electrical properties;

Humidity Ambient humidity can have a significant effect on measures. In particular, for the ground bonding resistance, the presence of water changes the conductivity of the air between the internal circuits and the amplifier chassis.

3.1.2 `DAILYHV` data

A `DAILYHV` test passes if and only if all the three measured values are within limits specified in the test plan. ACWV is the first measurement, and if its value is outside the specification bounds, the instrument avoids GBON test steps. If one of the GBON steps fails, since they are in parallel, the other is usually measured as well.

The server archives the three test measurements in a long format, where each one corresponds to a different record (see Table 3.1 on page 49), enriched with information about:

- date and time of the measurement;
- production line (indicated by the PF code, e.g. `DAILYHVPWS4CH` is the line that produces *Quattrocanali* family in Scandicci, `DAILYHVPWS8CH` for *Ottocanali* in Scandicci, `DAILYHVCRT` is the box in Cortona etc.);
- status (*pass* or *fail*);
- specification limits in the test plan;
- measured value;
- measure unit.

Figure 3.1 on page 50 depicts a sample of the three `DAILYHV` measurements: this time series pattern shows a stationary behavior around a constant mean (respectively 0.380 milliamperes for ACWV and 148 and 212 milliohms for GBON1 and GBON2). GBON measures have a much higher standard deviation in absolute than ACWV, but in general, all these variables have a variation coefficient between 1% and 5% (see Table 3.2 on page 53). Since operators must repeat the test until it passes, there is a variable number of observations on the same day.

Autocorrelation

One of the important aspects of `DAILYHV` data is that they are a *sequence in time* of the same variables. Section 3.1.1 lists environmental factors, such as ambient temperature and humidity, as important sources of variability. The period in the year is determinant for the greatest part of these aspects, and two tests taken close in time will usually be more similar than one measure in August and one in February. Formally, the tendency of data to behave similarly in contiguous periods is quantified in form of *autocovariance* (γ) and *autocorrelation* (ρ). Appendix C explains details of these statistical properties and how to estimate it.

Figure 3.2 shows the Autocorrelation Function (ACF) relative to the series in Figure 3.1, while Figure 3.3 reports the Partial Autocorrelation Function (PACF): both confirm the presence of a significant serial correlation. ACF and PACF have been computed on all series having enough data, always confirming a relevant autocorrelation effect. Summary table 3.2 report autocorrelation at lag 1 and lag 2 for each time series.

3.2 Process control monitoring

Quality industrial standards and legal norms define the checks that testing devices must necessarily pass. The objective of Powersoft is to make a further step, trying to prevent anomalies and testing tool wear through a periodic review of measured values and their trend in time.

ID	DateTime	PF	Site	Status	Measurement	Lower Bound	Upper Bound	Value	Unit
001	Day 1 - 08:31:54	DAILYHVPWSX	PWSPRD	FAIL	ACWV	0.1	2.0	0.23	mA
001	Day 1 - 08:31:54	DAILYHVPWSX	PWSPRD	FAIL	GBON1	103	120	999.9	mohm
004	Day 3 - 08:57:30	DAILYHVPWSX	PWSPRD	PASS	ACWV	0.1	2.0	0.23	mA
004	Day 3 - 08:57:30	DAILYHVPWSX	PWSPRD	PASS	GBON1	103	120	113.7	mohm
004	Day 3 - 08:57:30	DAILYHVPWSX	PWSPRD	PASS	GBON2	197	216	214.0	mohm
089	Day 3 - 08:59:13	DAILYHVPWS4CH	PWSPRD	PASS	ACWV	0.1	2.0	0.30	mA
089	Day 3 - 08:59:13	DAILYHVPWS4CH	PWSPRD	PASS	GBON1	131.7	145.5	143.8	mohm
089	Day 3 - 08:59:13	DAILYHVPWS4CH	PWSPRD	PASS	GBON2	193.8	214.2	206.9	mohm
125	Day 12 - 16:14:47	DAILYHVCRT	CRT	PASS	ACWV	0.1	2.0	0.22	mA
125	Day 12 - 16:14:47	DAILYHVCRT	CRT	PASS	GBON1	154.4	170.7	158.2	mohm
125	Day 12 - 16:14:47	DAILYHVCRT	CRT	PASS	GBON2	102.8	131.1	114.9	mohm

Table 3.1: Anonymized sample of query extraction for the DAILYHV test monitoring.

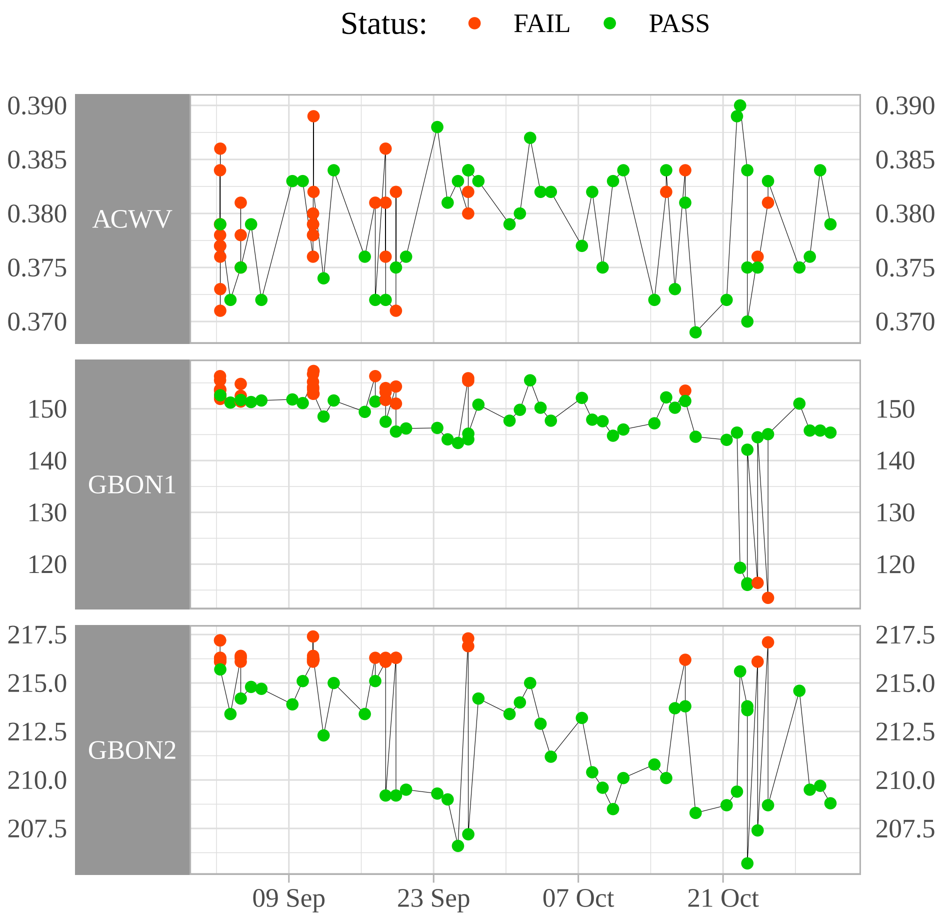
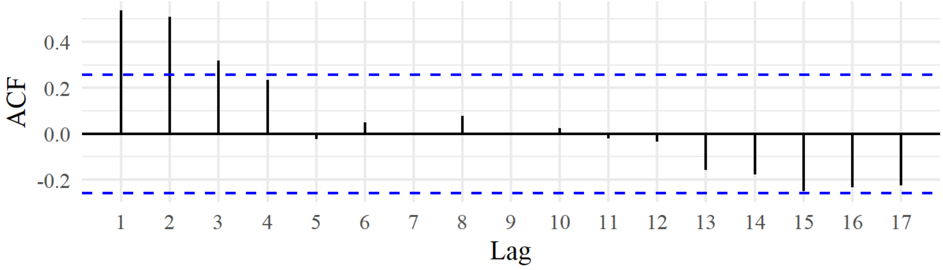
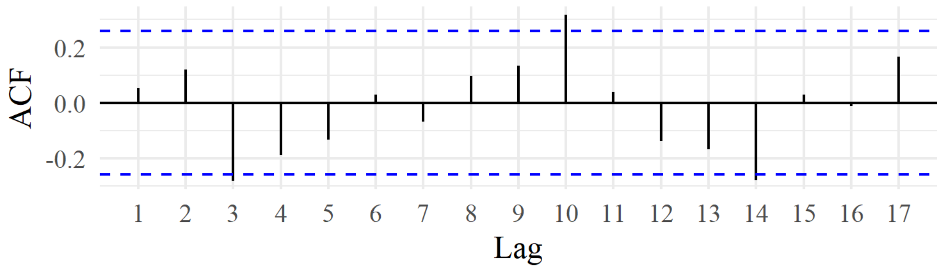


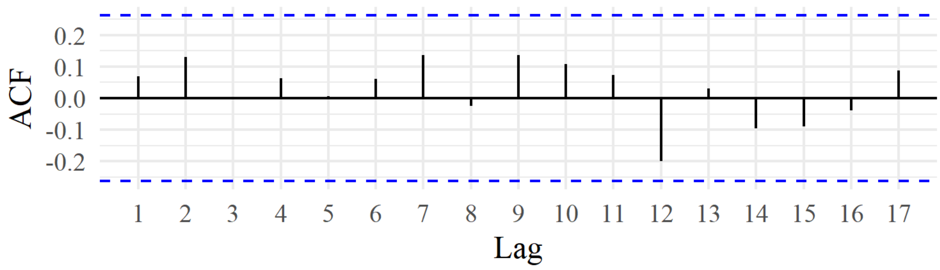
Figure 3.1: `DAILYHV` measurement series for a single testing device, from 1 September 2019 to 11 November 2019. Missing measurements have been omitted. The whole set of the measures performed during one test has just one status associated (*pass* or *fail*): if just one measure is out of the specification limits, all of them result as a *fail*.



(a) ACWV series.

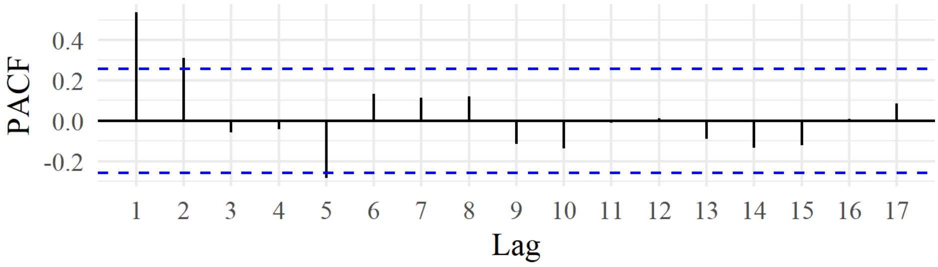


(b) GBON1 series.

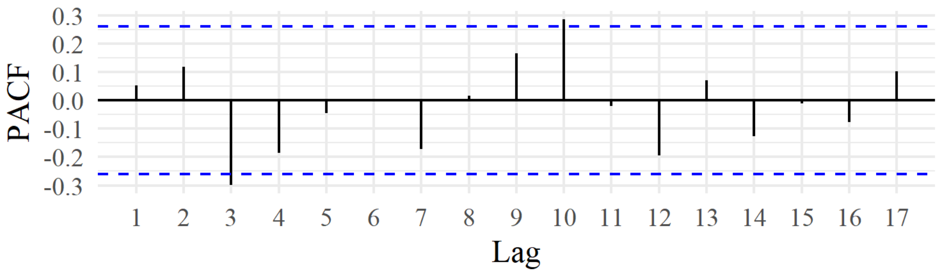


(c) GBON2 series.

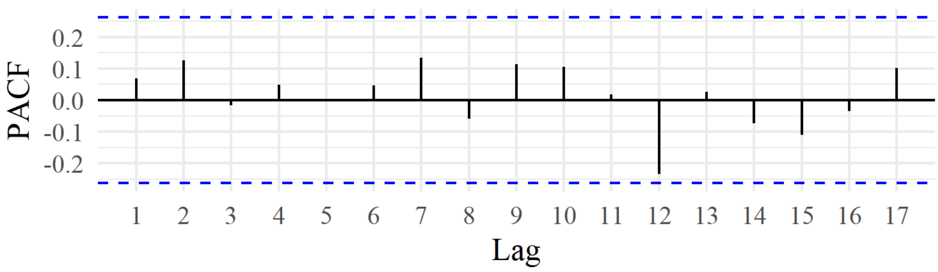
Figure 3.2: Estimated autocorrelation functions on `DAILYHV` measurement series. Estimates are relative to series in Figure 3.1 on the previous page. Dashed blue lines indicate critical value for the hypothesis of null autocorrelation at significance level $\alpha = 0.05$.



(a) ACWV series.



(b) GBON1 series.



(c) GBON2 series.

Figure 3.3: Estimated partial auto-correlation functions on `DAILYHV` measurement series. Estimates are relative to series in Figure 3.1 on page 50. Dashed blue lines indicate critical value for the hypothesis of null partial autocorrelation at significance level $\alpha = 0.05$.

TestStep	PF	N	Ndays	Min	Q1	Q2	Mean	Q3	Max	SD	IQR	VarCoef	Autocorr. Lag 1	Autocorr. Lag 2	Outlier
ACWV	DAILYHV01	22	22	0.22	0.22	0.22	0.22	0.22	0.22	0.00	0.00	0.00			7
ACWV	DAILYHV02	229	197	0.29	0.30	0.30	0.30	0.30	0.31	0.00	0.00	0.01	0.83	0.82	2
ACWV	DAILYHV03	69	35	0.23	0.23	0.23	0.23	0.23	0.24	0.00	0.00	0.01	0.65	0.71	40
ACWV	DAILYHV04	25	11	0.39	0.40	0.40	0.40	0.40	0.41	0.00	0.01	0.01	0.20	0.03	5
ACWV	DAILYHV05	258	193	0.35	0.37	0.37	0.37	0.38	0.39	0.01	0.01	0.02	0.57	0.52	0
ACWV	DAILYHV06	251	189	0.22	0.23	0.23	0.23	0.23	0.24	0.00	0.00	0.02	0.69	0.60	2
ACWV	DAILYHV07	39	34	0.23	0.23	0.23	0.24	0.24	0.25	0.01	0.01	0.03	0.65	0.56	0
ACWV	DAILYHV08	125	96	0.62	0.62	0.62	0.62	0.63	0.64	0.01	0.01	0.01	0.39	0.20	4
GBON1	DAILYHV01	24	21	154.50	156.73	158.40	172.00	164.55	228.80	27.06	7.82	0.16	0.98	0.85	2
GBON1	DAILYHV02	229	197	140.80	143.60	146.70	146.95	149.10	160.40	3.75	5.50	0.03	0.52	0.41	1
GBON1	DAILYHV03	68	29	73.30	76.57	81.20	81.74	86.40	94.10	5.95	9.83	0.07	0.86	0.78	1
GBON1	DAILYHV04	16	11	96.00	97.33	97.95	97.74	98.20	99.20	0.81	0.87	0.01	0.36	0.08	5
GBON1	DAILYHV05	253	193	113.50	143.40	147.20	147.06	151.70	158.30	6.85	8.30	0.05	0.66	0.63	5
GBON1	DAILYHV06	249	189	76.40	88.30	91.30	91.75	94.70	104.20	4.68	6.40	0.05	0.57	0.48	3
GBON1	DAILYHV07	37	34	82.80	92.40	94.10	94.25	96.40	102.60	3.32	4.00	0.04	0.33	0.01	2
GBON1	DAILYHV08	123	96	169.20	174.40	175.50	175.92	177.20	185.10	3.03	2.80	0.02	0.28	0.04	2
GBON2	DAILYHV01	19	18	111.30	114.15	115.20	116.29	118.40	122.30	3.32	4.25	0.03	-0.31	0.07	0
GBON2	DAILYHV02	226	197	204.40	206.90	208.75	209.28	211.38	217.60	2.87	4.47	0.01	0.49	0.46	2
GBON2	DAILYHV03	38	26	176.30	179.22	181.40	181.69	183.80	187.30	3.29	4.58	0.02	0.50	0.51	0
GBON2	DAILYHV04	15	11	196.30	197.20	197.60	197.48	197.85	198.20	0.53	0.65	0.00	0.24	-0.01	4
GBON2	DAILYHV05	247	193	204.40	207.40	210.80	211.15	215.05	217.40	3.92	7.65	0.02	0.73	0.70	5
GBON2	DAILYHV06	201	189	179.80	184.60	187.60	187.36	189.80	196.10	3.55	5.20	0.02	0.58	0.53	2
GBON2	DAILYHV07	35	33	186.20	189.00	190.60	190.23	191.45	194.70	2.01	2.45	0.01	-0.11	-0.01	0
GBON2	DAILYHV08	109	95	117.10	123.50	125.00	125.23	126.60	134.70	2.86	3.10	0.02	0.27	0.08	4

Table 3.2: `DAILYHV` data summary starting from January 2019 up to October 2019. Outlier values have been treated as missing.

3.2.1 Traditional process control solutions

SPC offers tools, such as x-bar charts and R charts (see Appendix A) to monitor measurements. In this case, the nature of data led us to consider starting from the traditional x-bar chart to monitor the process.

Usual x-bar charts consider the observed statistic

$$\hat{y}_t = \frac{1}{n_t} \sum_{i=1}^{n} y_{i,t} \tag{3.1}$$

where $y_{i,d}$ is the observed value of a the i-th measure in the t-th sample.

The use of the batch means, instead of the single value, guarantees a better convergence to the Normal distribution (thanks to the Central Limit Theorem) and a more consistent application of Shewhart's rules. The measures are usually divided into samples, assumed to be representative of the entire population.

The parameters for the x-bar chart are

$$\mu = \frac{1}{m} \sum_{d=1}^{m} \sum_{i=1}^{n} y_{i,d} \tag{3.2}$$

where $\{1, \ldots, m\}$ are indices of the period in which the process is considered in-control,

$$\mathrm{UCL} = \mu + \delta \qquad \mathrm{LCL} = \mu - \delta \tag{3.3}$$

$$\delta_t = L \cdot \sigma_y = L \cdot \sqrt{\frac{1}{m} \sum_{d=1}^{m} \sum_{i=1}^{n} (y_{i,d} - \mu)} \tag{3.4}$$

where L is usually set to 3.

3.2.2 The case-specific application

Several differences emerged immediately in the application of these traditional solutions:

1. Traditionally, data is examined by groups. The nature of this DGP does not allow to group data. This fact does not prevent per se to create an x-bar-chart (it was sufficient considering it a particular case in which $n = 1$, taken for granted the normality of data), but it makes impossible to monitor the process variability with R-charts or S-charts;

2. Traditional control charts rely on a fundamental assumption: the DGP under control is composed by a set of *independent* runs. Preliminary analyses on `DAILYHV` data (in Section 3.1.2 on page 47) showed however that assumption is not satisfied;

3. Traditionally, the control charts' parameters are estimated from the "gold standard" processing period. The continuous evolution of the `DAILYHV` testing process (e.g., adding new testing boxes, re-calibrations) makes it difficult to isolate a well-defined period to estimate in-control parameters.

As in FPY monitoring, we need to automatize the report generation but, at the same time, to permit to change parameters flexibly.

3.2.3 Time series modeling

Psarakis and Papaleonida [2007] collected a complete overview of methods from the scientific literature to use SPC tools in case of auto-correlated processes.

For stationary processes, Alwan and Roberts [1988] proposed the use of the standard Autoregressive Moving-average (ARMA) models in combination with control charts. In particular, they suggest the use of two control charts: Common Cause Chart and Special Cause Chart [Psarakis and Papaleonida, 2007, see p. 511]. The same suggestion comes from Lu and Reynolds [1999], which combines the time series modeling theory with advanced charts such as CSUM and EWMA to detect shifts in the process mean, net of the physiological jumping-mean nature of the process.

Paolella [2019] examines all the theoretic aspects of this statistical framework.

Each author proposes a different specification of ARMA or Autoregressive Integrated Moving-average (ARIMA) model. Also here, identifying the best model for the `DAILYHV` data needs to explore the several possibilities. We had to filter the autocorrelation-due part of the variability, without overkilling the model complexity or to incur in overfitting (including shifts in the model dynamics and not considering it an anomaly) and make the process monitoring useless.

With these objectives in mind, we fitted several model specifications to various subsets of the series, changing both the time window and testing device source. Parameters estimation is done using the `forecast` R package [Hyndman et al., 2019; Hyndman and Khandakar, 2008]. All the estimated models remained in the family of ARMA, with no integration component.

Table 3.3 on page 57 shows the best results in terms of BIC of ARMA models (we applied the models to various contiguous periods to test the fitting over different scenarios). In the choice of which AR and MA order use, we considered firstly the fact that letting the model adapt excessively to data, without the constant supervision of an expert to evaluate the model properties, could lead to dangerous overfitting of data time dynamics. So, instead of using an automatic selection algorithm for each different measurement series, we searched for the right compromise that fitted good enough in all the analyzed cases. Eventually, we opted for an ARMA(1, 1), to use indistinctly with all the time series. Besides, we left the option for the quality control manager to use the model evaluation or the raw series for the x-bar chart representation.

We also performed an analysis of residual normality, since x-bar chart control limits usually rely on this assumption. The graphical evaluation (see Figure 3.4 on page 58) provided positive results, except for some irregularity in the upper tails, such as in the GBON2 series (Figures 3.4c and 3.4d). We applied the Shapiro-Wilk test for the null hypothesis of normality (results are presented in Table 3.4 on the next page). The two outcomes with a p-value lower than the canonical 0.05 are likely due to two evident outliers on the tails of the distribution: Shapiro-Wilk test is particularly sensitive to this kind of values.

Outlier treatment

There are cases in which the numeric measurements are totally out of the normal range of plausible values: bad cable connections, dust, human errors of the operator, could result in a "near-infinite" resistance or a "near-zero" current flow. Defining outlier limits is a technical matter that depends on the kind of test we are performing and the physical properties of the test equipment.

The final choice was to use a configuration file and manually set the outlier's limits for each measure kind and replace them with missing values. This approach permits not to lose the information about the measurement since the presence of missing values is used for the time series modeling (replacing the observation with its conditional expected value).

3.2.4 Practical implementation

Similarly to the FPY monitoring, we created a CSV configuration file to set the script parameters at the level of each test step for each testing device (see Table 3.5 on page 61). It is possible manually set

- the *specification limits* of the measurement (by default it is read from the test plan);
- the *starting date*, from which to represent the sequence, first day of the reference period. Alternatively, it is possible to set the time window with a fixed number of days before the current date;
- the *reference date*, the last day of the reference period. In this period, we estimate the control limits and ARMA model parameters. As for the starting date, it is possible also to set a fixed number of days for the estimation time window;
- a flag to set or unset the use of the ARMA model residual (by default set to *on*): in case of flag *on* the x-bar chart y values are the model residuals. Otherwise, they are the raw series values;
- the outlier upper and lower limits;

Date min	Date max	PF	Step	AR Order	MA Order
2019-05-01	2019-08-01	DAILYHVPWS4CH	ACWV	1	1
2019-05-01	2019-08-01	DAILYHVPWS4CH	GBON1	1	0
2019-05-01	2019-08-01	DAILYHVPWS4CH	GBON2	1	1
2019-05-01	2019-08-01	DAILYHVPWST	ACWV	1	1
2019-05-01	2019-08-01	DAILYHVPWST	GBON1	1	1
2019-05-01	2019-08-01	DAILYHVPWST	GBON2	1	0
2019-05-01	2019-08-01	DAILYHVPWSX	ACWV	1	1
2019-05-01	2019-08-01	DAILYHVPWSX	GBON1	1	0
2019-05-01	2019-08-01	DAILYHVPWSX	GBON2	0	0
2019-09-01	2019-11-01	DAILYHVPWS4CH	ACWV	1	1
2019-09-01	2019-11-01	DAILYHVPWS4CH	GBON1	0	0
2019-09-01	2019-11-01	DAILYHVPWS4CH	GBON2	0	0
2019-09-01	2019-11-01	DAILYHVPWS8CH	ACWV	1	0
2019-09-01	2019-11-01	DAILYHVPWS8CH	GBON1	1	0
2019-09-01	2019-11-01	DAILYHVPWS8CH	GBON2	1	0
2019-09-01	2019-11-01	DAILYHVPWST	ACWV	0	0
2019-09-01	2019-11-01	DAILYHVPWST	GBON1	2	0
2019-09-01	2019-11-01	DAILYHVPWST	GBON2	1	1
2019-09-01	2019-11-01	DAILYHVPWSX	ACWV	0	0
2019-09-01	2019-11-01	DAILYHVPWSX	GBON1	1	0
2019-09-01	2019-11-01	DAILYHVPWSX	GBON2	0	0

Table 3.3: AR and MA orders of the best fitting model in terms of BIC.

PF	Step	P-value
DAILYHVPWS4CH	ACWV	0.67
DAILYHVPWS4CH	GBON1	0.39
DAILYHVPWS4CH	GBON2	0.48
DAILYHVPWS8CH	ACWV	0.77
DAILYHVPWS8CH	GBON1	0.02
DAILYHVPWS8CH	GBON2	0.91
DAILYHVPWST	ACWV	0.32
DAILYHVPWST	GBON1	0.33
DAILYHVPWST	GBON2	0.64
DAILYHVPWSX	ACWV	0.00
DAILYHVPWSX	GBON1	0.18
DAILYHVPWSX	GBON2	0.33

Table 3.4: Shapiro-Wilk normality test p-values for the ARMA(1, 1) model residuals on several devices' series.

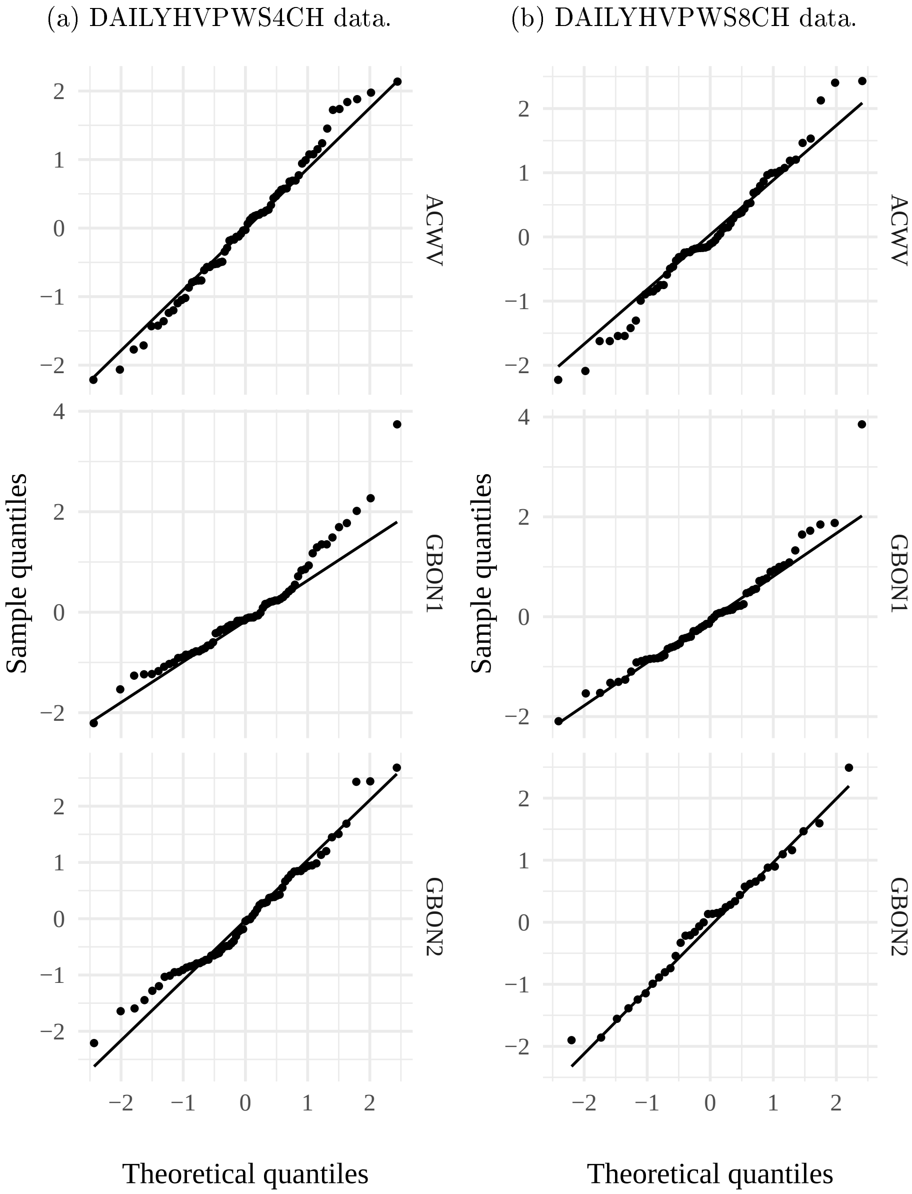


Figure 3.4: Standardized ARMA model residuals' qqplot (part 1).

(c) DAILYHVPWST data. (d) DAILYHVPWSX data.

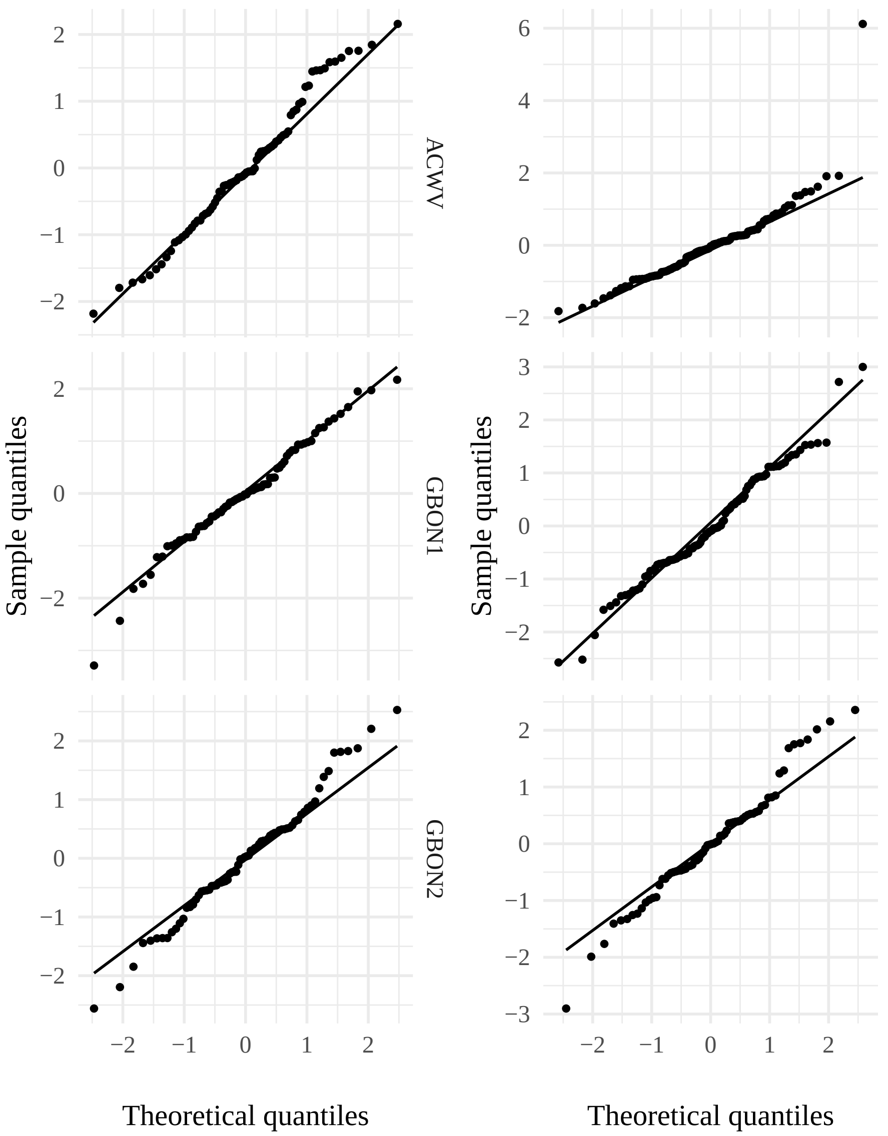


Figure 3.4: Standardized ARMA model residuals' qqplot (part 2).

- a flag to remove or not the outlier values (if removed, they become missing values);
- the dates to be removed for special causes (e.g., re-calibrations).

The program estimates control limits and model parameters from a reference period (from the *starting date* to the *reference date*). First, we estimate the AR and MA coefficients of an ARMA(1, 1) model, and the standard deviation (σ_r) of residuals; then we apply the classical x-bar chart on them. X-bar chart plots have been created using the R package `qcc` [Scrucca, 2004].

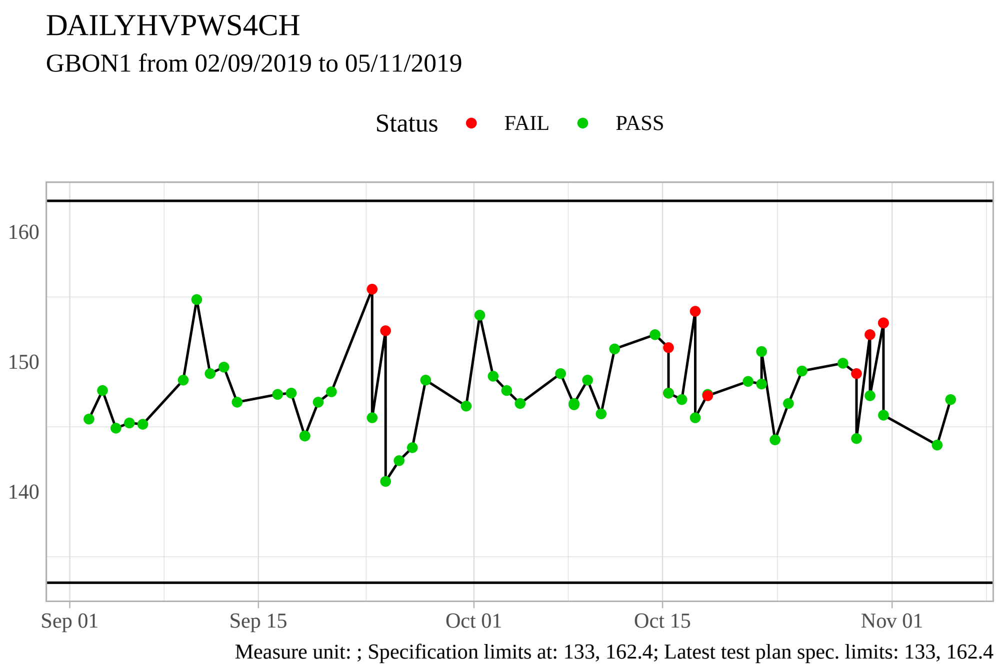
We implemented and scheduled a routine that produces two plots for each combination of testing device (e.g., QUATTROCANALI in Scandicci, OTTOCANALI in Scandicci, Cortona) and test step (ACW, GBON1, GBON2):

1. a representation of the entire sequence from the starting date without elaborations, and its specification limits;
2. the x-bar chart of the estimated ARMA model residuals. For the measurements between the starting date and the reference date, values are residuals from the fitted model. For the measurements after the reference date, values are residuals from the one-step-ahead predictions. Note that if the configuration file specifies not to use the ARMA modeling, the x-bar chart is created directly to the raw measurements.

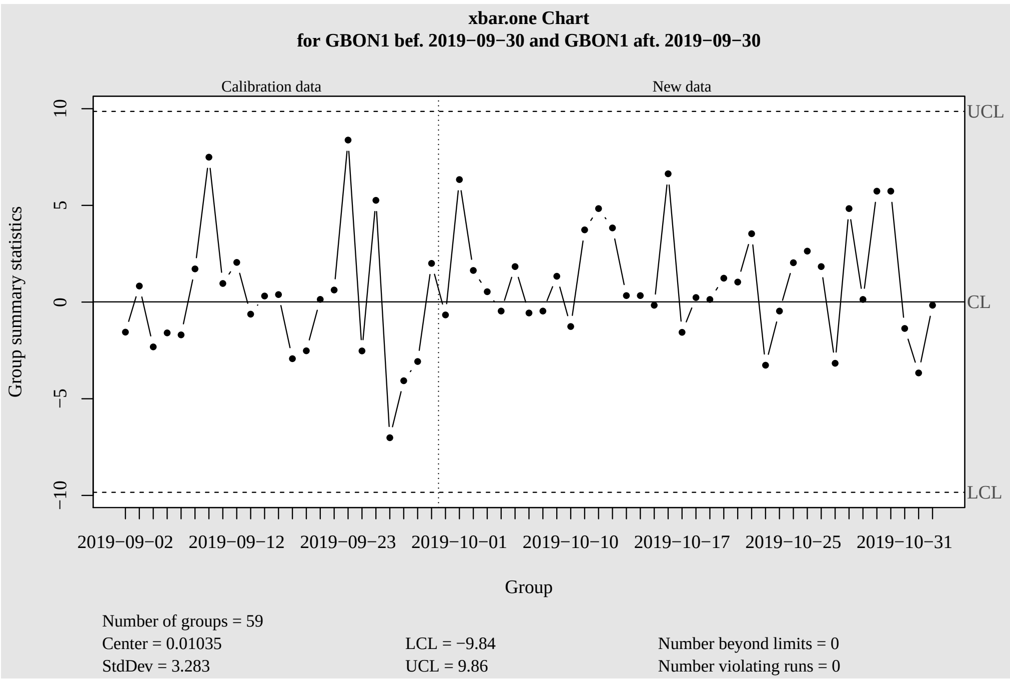
For each device, the production managers and the quality control officer receive a weekly PDF file summarizing the three tests. An additional CSV file, including raw data about the three test steps, is produced. Using the latter, process analysts can anytime perform ad hoc analysis on anomalies emerging from the Shewhart charts or the raw run-chart.

PF	Site	TestStep	Lower Bound	Upper Bound	Date Min	Date Ref	Model	Remove Outliers	Outlier Lower Bound	Outlier Upper Bound	Remove Dates
DAILYHVCRT	CRT	GBON2	102.8	131.1	01/07/2019	11/10/2019	TRUE	TRUE	20	500	
DAILYHVPWS8CH	PWSPRD	ACWV	0.209	0.255	01/07/2019	11/10/2019	TRUE	TRUE	0.5	100	
DAILYHVPWSMEZZO	PWSPRD	GBON1	86.0	106.0	20/10/2019	11/11/2019	FALSE	TRUE	20	550	25/10/2019, 26/10/2019

Table 3.5: Partial example of configuration file for the `DAILYHV` monitoring system.



(a) Raw data, filtered from outliers compared with specification limits.



(b) Model fitted values residuals and one-step-ahead prediction residuals compared with control limits. X-bar chart plots have been created using the R package `qcc` [Scrucca, 2004].

Figure 3.5: Example of `DAILYHV` report for the GBON1 test on the QUATTRO-CANALI production line.

CHAPTER

FOUR

CONCLUSIONS

The project had positive results in each of its parts. We tested daily, weekly, and monthly reports for two months. Once in production, they demonstrated to be reliable and capable of replacing the system AS-IS. SPC tools have been successfully adapted to the needs of Powersoft, giving birth to the FPY monitoring semi-automatic system to identify anomalies without the need for a daily screen of the processes. Also, this system has been tested for a short period and then officially adopted, using both the process FPY mean predicted with the aggregation by family, type and site, and those predicted by the Bayesian GLMM, with positive feedback from the users. We put on scheduling all these routines, and they update their parameters automatically, to keep the approach flexible, and do not allow these tools to become old with the possible evolution of the company.

We obtained many positive results in these few months: we reached them with the awareness that they are not an endpoint. In this sense, we expect that these programs will not crystallize in the form that we created them, but they will adapt to new necessities and objectives. Further aspects to be explored are an in-depth analysis of time dynamics in processes' FPYs, new KPIs to analyze processing durations, multivariate approaches for `DAILYHV` data (combining information from the three test steps and their interactions).

I finally want to remark as Powersoft Stakeholders returned positive feedback during our interaction as well as on the project results.

Appendices

APPENDIX

ONE

CONTROL CHARTS

Control charts are a general framework developed for the first time by Walter Shewhart in the 1920s, used to monitor an industrial or business process. They consist of a visualization of a KPI (y_t, a statistic of a batch of items) repeatedly recorded during the time, with regular intervals. Together with the observed statistics, the graph must report their expected value (μ) and UCL ($\mu + \delta$) and LCL ($\mu - \delta$) under hypothesis of *in-control process*.

In-control processes should have regularity and predictability properties to keep the defect rate under certain bounds. The in-control process is an ideal status, and the real parameters are unknown to the analysts. It is necessary to estimate them from a "phase 1" period when the process is supposed to have not anomalies. If it does contain any, they must be filtered out.

The most used control charts are:

X-bar chart for the monitoring of the process mean

$$y_t = \bar{x}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} x_{t,i}$$

where n_t is the size of the t-th sample and $x_{t,i}$ is the i-th value of the t-th sample.

R chart for the monitoring of the process variability (through the range)

$$y_t = R_t = x_{\max_t} - x_{\min_t}$$

S chart for the monitoring of the process variability (through the standard deviation)

$$y_t = s_t = \sqrt{\frac{\sum_{i=1}^{n_t} (x_{t,i} - \bar{x})^2}{n_i - 1}}$$

P chart for the monitoring of the proportion of nonconforming units

$$y_t = p_t = \frac{D_t}{n_t}$$

where D_t is the number of defective units in the t-th sample.

Each of these control chart has specific properties, but there are some common assumptions necessary to guarantee their power and broad applicability.

Assumptions

Control charts have been developed with a strong applied focus: manuals describe them with a high level of detail in a sort of "rule book" to use. Studying them with a statistical background, they show to have deep roots in hypothesis test theory, and their high reliability relies on some underlying assumption about data that is often true or almost true in practice. The first and most important assumption is about the independence of observations: an in-control process should output performance metrics that are not influenced by the previous measures. Another critical hypothesis that is dependent on the specific kind of metrics that we want to monitor is the distribution of it. In particular, for continuous data, we assume that they are distributed normally (or with a good approximation to the Normal) with constant mean and constant variance. If these assumptions are satisfied, control charts can are very efficient in detecting shifts and anomalies in a periodically monitored process flow with a low rate of false alarms (an Average Run Length of 340 observations before one false alarm).

Shewhart's rules

Shewhart proposed a list of several rules that should help process managers to detect not only evident anomalies, but also big and medium shifts from the control conditions. These rules are all based on the idea that under in-control process conditions, the observed statistics of the batches should be independent random variables, and regular patterns would appear with very low probability. Starting from that, in the course of the 20th century, scholars and associations established a list of general rules to determine when a process needs further investigation:

1. 1 point outside the control limits;
2. 8-9 points on the same size of the centerline;
3. 6 consecutive points are steadily increasing or decreasing;
4. 14 consecutive points are alternating up and down;
5. 2 out of 3 consecutive points are more than 2 sigmas from the centerline in the same direction;

6. 4 out 5 consecutive points are more than 1 sigma from the centerline in the same direction;

7. 15 consecutive points are within 1 sigma of the centerline;

8. 8 consecutive points on either side of the centerline with not within 1 sigma.

A complete overview of control charts and, in general, SPC can be found in Montgomery [2009].

APPENDIX
TWO

HV TEST

The HV test (or *hipot* test) has the general objective of verifying the insulation of a DUT with respect to the external metallic case. In theory, the device should be able to withstand voltage up to several kilovolts (e.g., sudden spikes in electrical supply network).

To schematize, we consider a DUT that consists of a simple circuit to switch on a light bulb (see Figure B.1a). A 3-terminal plug powers the device: *line* (L), *neutral* (N), and *ground* (GND, sometimes called *Protective Earth*). Line and neutral provide power to the internal circuit; the ground is a protective connection, providing a safe path for the current in the case and avoid electrical shocks. The tester is both a generator and a measuring instrument, that supplies high voltage direct (DC) or alternating (AC) current, up to 2000 V. It measures current flows in the range from 0.001 to 1 A. Both HV and Ground Bonding tests are part of the same testing protocol.

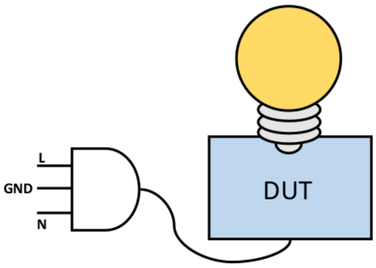
The DUT is a set of two components: the working circuit that performs the primary function (in the example, lighting the bulb), and the chassis, a metallic case enclosing the circuit, which can be touched by an external user (see Figure B.1b). The working circuit is separated from the chassis by an insulator (or *dielectric*): usually, it consists of a non-conductive foam (or any other material) which prevents electrical contact between the circuit and the metallic case. The chassis must connect to the ground: if the insulation between it and the circuit fails, the current must flow to the ground to avoid an electrical shock for the user.

Ground bonding test The operator connects the chassis to the GND plug of the tester, and the instrument performs one or more measures of the current flow. It then calculates resistance indirectly using Ohm's law

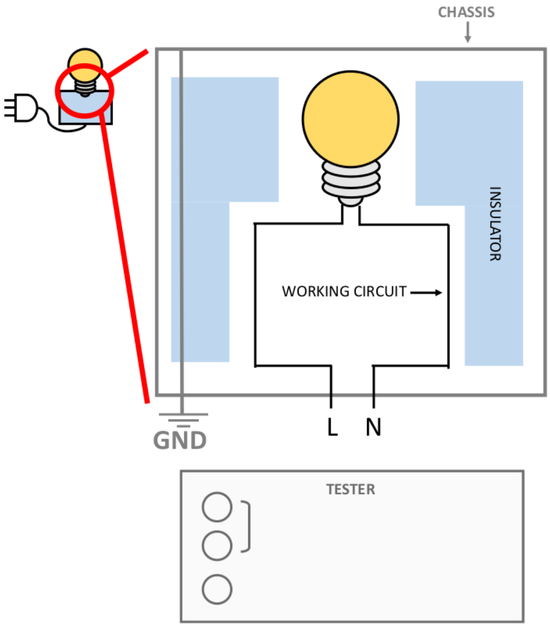
$$V = R \cdot I$$

that must be equal to a small fraction of Ohms (see Figure B.1c).

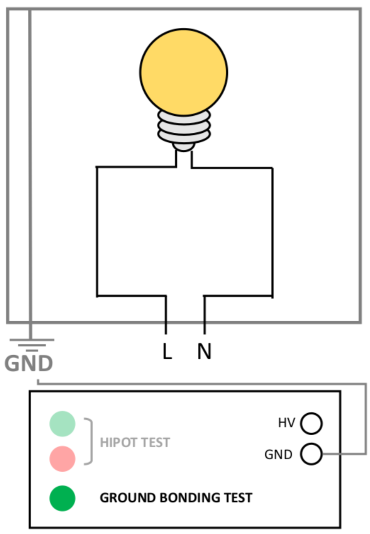
Hipot test Line and neutral are short-circuited together and connected to the high voltage (HV) terminal of the tester. Then the HV is switched on, and the working circuit and chassis become a capacitive system: two metallic components separated by an insulator. When voltage starts to increase between the two metallic plates, charges of opposite sign accumulate on the two sides (see Figure B.1d). If the insulator cannot withstand the cumulated electrical load, it will experience a dielectric breakdown: charges start to flow through the insulator material, through the chassis and will eventually reach the ground tester (see Figure B.1e). This current is called *leakage current*, and the tester measures it. If it overcomes a certain threshold, the DUT fails the test. If the insulator can withstand the high voltage, no risk of electrical shock takes place, and the DUT passes the test (see Figure B.1f).



(a)

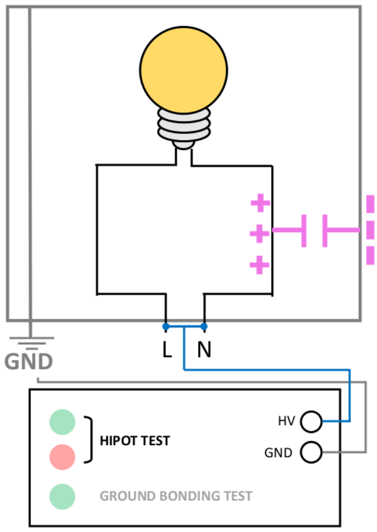


(b)

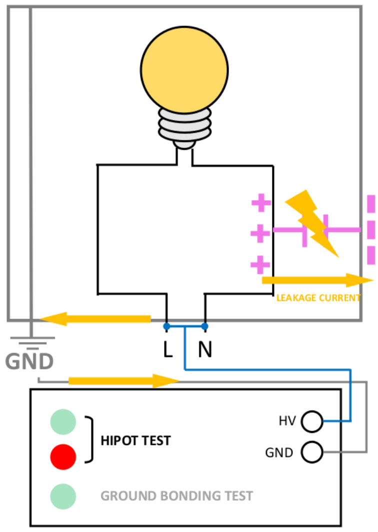


(c)

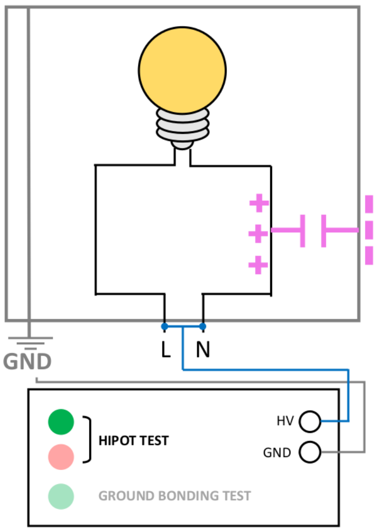
Figure B.1: HV test schematization (part 1).



(d)



(e)



(f)

Figure B.1: HV test schematization (part 2).

APPENDIX

THREE

TIME SERIES ANALYSIS

In Statistics, sequences of the same measures repeated over multiple instants of time can be modeled with the mathematical framework of *stochastic processes*. In formal terms, we can represent the sequences of measures as one of the possible realizations of the set of random variables

$$\boldsymbol{Y} = \{Y_1, Y_2, \ldots, Y_i, \ldots, Y_T\}$$

In general, they preserve the characteristics of random variables but can be analyzed with different approaches, taking into account the fact that they are naturally ordered. The order gives us precious info since we can extract information about how reciprocally near observations conditions each other and let us make predictions about next (in time) cases.

Autocovariance and autocorrelation

Given a sequence of stochastic variables ordered in time $Y_1, Y_2, \ldots, Y_t, \ldots, Y_T$ with constant mean μ and variance σ^2, where T is the total number of variables, the *autocovariance* is defined as the tendency of data to self-resemble at time lag d:

$$\gamma_d = \mathbb{E}[(Y_t - Y)(Y_{t-d} - \mu)] \tag{C.1}$$

where t is the time index, d the inter-time lag.

When γ_d is equal to 0, Y values at lag d are totally independent. The higher the covariance value γ_d, the more data will be similar at lag d. If γ_d has a value below 0, observations at lag d will be more distant than we expect on average.

To have a standardized value, easier for the interpretation, autocovariance can be divided by the total variance of Y, obtaining the *autocorrelation*:

$$\rho_d = \frac{\gamma_d}{\sigma^2} \tag{C.2}$$

where

$$\sigma^2 = \mathbb{E}[(Y - \mu)(Y - \mu)]$$

ρ_d ranges from -1 to +1 and can be interpreted in the same way of γ_d. When ρ_d has value equal to +1 (-1), the observation Y_t is exactly predictable from Y_{t-d}; in particular it will have a higher (lower) value. When ρ_d has values approximately to 0, the variable Y_t will be uncorrelated with the variable Y_{t-d}.

ρ_d can be estimated from the available data as

$$\hat{\rho}_d = \frac{\sum_{t=d+1}^{T}(y_t - \bar{y})(y_{t-d} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2} \tag{C.3}$$

where T is the last time index and $\bar{y}$ is the sample mean

$$\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

ARMA models

The information about the self-dependence of a stochastic process can be used to detect patterns in it. In particular, one of the most used models to predict auto-regressive series is the ARMA (Auto-Regressive Moving Average). It consists of the combination of two modeling components: the Auto-Regressive part and the Moving Average part.

The AR model states that the present value (Y_t) depends just on its past value (Y_{t-1}), with the addition of a random shock (ϵ_t): the most basic version is the AR(1) model

$$Y_t = \beta_0 + \beta_1 \cdot Y_{t-1} + \epsilon_t \tag{C.4}$$

where

$$\epsilon_t \sim \text{Normal}\left(0,\ \sigma_\epsilon^2\right) \qquad \text{with } \sigma_\epsilon^2 \text{ constant over } t \tag{C.5}$$

$$\epsilon_{t_1} \perp \epsilon_{t_2} \qquad \text{when } t_1 \neq t_2 \tag{C.6}$$

In general the AR(p) has p regressors backward in time

$$Y_t = \beta_0 + \beta_1 \cdot Y_{t-1} + \ldots + \beta_p \cdot Y_{t-p} + \epsilon_t \tag{C.7}$$

The MA model states that the present value (Y_t) depends just on the past random noise (ϵ_{t-1}), with the addition of a random shock (ϵ_t): the most basic version is the MA(1) model

$$Y_t = \theta_0 + \theta_1 \cdot \epsilon_{t-1} + \epsilon_t \tag{C.8}$$

with the same conditions in Equations (C.5) and (C.6). In general the MA(q) has q regressors backward in time

$$Y_t = \theta_0 + \theta_1 \cdot \epsilon_{t-1} + \ldots + \theta_q \cdot \epsilon_{t-q} + \epsilon_t \tag{C.9}$$

Combining together the two models we obtain the ARMA(p, q), where p and q indicate the respective number of regressors

$$Y_t = \mu + \beta_1 Y_{t-1} + \ldots + \beta_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \ldots + \epsilon_{t-q} + \epsilon_t \tag{C.10}$$

The use of ARMA models and their theoretical background is the subject of entire coursebooks. Kirchgässner and Wolters [2010] presents a complete overview of time series analysis and ARMA models.

BIBLIOGRAPHY

Layth C. Alwan and Harry V. Roberts. Time-series modeling for statistical process control. *Journal of Business and Economic Statistics*, 6(1):87–95, 1988. doi: 10.2307/1391421.

Chao-Wen Lu and Marion R. Reynolds. Ewma control charts for monitoring the mean of autocorrelated processes. *Journal of Quality Technology*, 31(2): 166–188, 1999. doi: 10.1080/00224065.1999.11979913.

R Special Interest Group on Databases (R-SIG-DB), Hadley Wickham, and Kirill Müller. *DBI: R Database Interface*, 2018. URL `https://CRAN.R-project.org/package=DBI`. R package version 1.0.0.

Jim Hester and Hadley Wickham. *odbc: Connect to ODBC Compatible Databases (using the DBI Interface)*, 2018. URL `https://CRAN.R-project.org/package=odbc`. R package version 1.1.6.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL `https://www.R-project.org/`.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686.

Theodore T. Allen. *Introduction to engineering statistics and six sigma: statistical quality control and design of experiments and systems*. Springer, 2006.

Matthew Daneman. Xerox cutting lean six sigma, jobs, Oct 2014. URL `https://eu.democratandchronicle.com/story/money/business/2014/10/13/xerox-cuts-popular-lean-six-sigma-program-jobs/17203841/`.

D.C. Montgomery. *Introduction to Statistical Quality Control*. Wiley, 2009. ISBN 9780470233979. URL `https://books.google.it/books?id=oG1xPgAACAAJ`.

L. Fahrmeir, Thomas Kneib, Stefan Lang, and Brian D. Marx. *Regression: models, methods and applications*. Springer, 2015.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi: 10.18637/jss.v067.i01.

Ben Goodrich, Jonah Gabry, Imad Ali, and Sam Brilleman. rstanarm: Bayesian applied regression modeling via Stan., 2018. URL `http://mc-stan.org/`. R package version 2.17.4.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

S. Psarakis and G. E. A. Papaleonida. Spc procedures for monitoring autocorrelated processes. *Quality Technology & Quantitative Management*, 4(4): 501–540, 2007. doi: 10.1080/16843703.2007.11673168. URL `https://doi.org/10.1080/16843703.2007.11673168`.

Marc S. Paolella. *Linear models and time-series analysis: regression, ANOVA, ARMA and GARCH*. John Wiley and Sons, Inc., 2019.

Rob Hyndman, George Athanasopoulos, Christoph Bergmeir, Gabriel Caceres, Leanne Chhay, Mitchell O'Hara-Wild, Fotios Petropoulos, Slava Razbash, Earo Wang, and Farah Yasmeen. *forecast: Forecasting functions for time series and linear models*, 2019. URL `http://pkg.robjhyndman.com/forecast`. R package version 8.10.

Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008. URL `http://www.jstatsoft.org/article/view/v027i03`.

Luca Scrucca. qcc: an r package for quality control charting and statistical process control. *R News*, 4/1:11–17, 2004. URL `https://cran.r-project.org/doc/Rnews/`.

Gebhard Kirchgässner and Jürgen Wolters. *Introduction to modern time series analysis*. Springer, 2010.