Tester Capacity Control at Qimonda Portugal

Frederico Maria Mourão Glória

Projecto de Dissertação do MIEIG 2007/2008

Orientador na FEUP: Prof. João Falcão e Cunha

Orientador na Qimonda Portugal: Eng. Jorge Sousa

Faculdade de Engenharia da Universidade do Porto

Mestrado Integrado em Engenharia Industrial e Gestão

2008-02-26

À minha família e amigos.

Resumo

O presente documento descreve o projecto que decorreu na Qimonda Portugal, empresa na área dos semicondutores. O projecto teve como título "*Tester Capacity Control*" e consistia na definição de uma ferramenta de controlo da capacidade na área de Teste.

Este projecto está inserido no âmbito do tema Controlo de Produção, sendo descritas no final do documento as vantagens que este projecto, enquanto ferramenta de monitorização e controlo de produção, pode trazer na área de semicondutores.

A área de Teste representa cerca de 50% do investimento total da empresa, apresentando actualmente algumas ineficiências relativamente ao controlo de capacidade da área produtiva. No presente, as empresas focam-se essencialmente no controlo de produção relacionado com a logística e controlo do fluxo de materiais, não acentuando a importância que o controlo de produção a nível de capacidade das máquinas tem no aumento de produtividade. A Qimonda, enquanto empresa de alta tecnologia e de rigorosos padrões de qualidade e prazos de entrega, reconhece a necessidade de melhorar a sua produção através do controlo de capacidade da sua área produtiva.

Actualmente existem algumas ferramentas disponíveis na área de Teste que permitem controlar a capacidade através da detecção das causas de baixo desempenho, mas apresentam algumas ineficiências a nível de flexibilidade e tempo de pesquisa. O projecto desenvolvido teve como objectivo melhorar a visibilidade dos problemas de capacidade na área de Teste, através do desenho e especificação de uma ferramenta flexível e que permitisse uma rápida e eficaz detecção dos problemas.

A ferramenta a desenvolver deveria conter informações sobre o plano de produção semanal, bem como da produção actual de maneira a gerar alertas nos casos de baixo desempenho. Adicionalmente, e para permitir uma melhor e mais correcta análise da situação, a ferramenta devia apresentar os indicadores chave que permitiriam uma avaliação correcta e aprofundada da situação, de maneira a detectar as causas da redução de eficiência na área. Deste modo, a ferramenta iria permitir aumentar o desempenho geral da área, reduzindo os custos inerentes às paragens e à redução de capacidade.

Para a execução deste projecto foi definida uma metodologia que permitiu uma correcta abordagem ao problema e o desenvolvimento de uma ferramenta que superasse as ineficiências sentidas pelos responsáveis da área de Gestão da Produção. Esta metodologia permitiu identificar os problemas e as necessidades dos futuros utilizadores, os quais foram fundamentais para a especificação de tal ferramenta.

No presente relatório está contida toda a metodologia seguida, explicando cada uma das fases detalhadamente, bem como a descrição do protótipo criado que irá dar suporte e substituir a ferramenta durante o seu desenvolvimento. No final do documento estão descritas as vantagens que o programa, enquanto ferramenta de controlo de capacidade, irá proporcionar à área de Teste.

Tester Capacity Control at Qimonda Portugal

Abstract

This report describes the project developed during the internship taken in an enterprise that acts in the area of semiconductors, named Qimonda Portugal. The project with the title "Tester Capacity Control" consisted in the definition of a tool to control the capacity of the Test area.

This project is related with the theme *Production Control*, and in the end of the paper it can be found its advantages in the area of semiconductors, while tool of monitoring and production control.

The Test area represents about 50% of total investment of the company, and presents some inefficiency related with the capacity control of the productive area. Actually, the enterprises focus in the control of the production related with the logistics and the control of the flow of materials, and do not enhance the importance that controlling the production by observing the capacity of the machines have, in increasing the productivity. Qimonda, as a high-tech company with high and rigorous quality standards and delivery deadlines, recognizes the necessity of improving its production by controlling the capacity of its productive area.

Nowadays, there are some available tools which allow controlling the capacity by detecting the causes of low performance; however they present some inefficiency of flexibility and research time. The project that was developed had the objective of improving the visibility of the capacity problems in the Test area, through the definition of a flexible tool that will allow a faster and efficient detection of the problems.

The tool should contain information about the weekly production plan, as well as the current production for alerting in case of low performance. In addition, and to allow a better analysis of the situation, the tool should present key-indicators which should permit a correct and deep evaluation, by detecting causes that reduce the efficiency of the area. In this way, the tool would allow to increase the general performance of the area, reducing the inherent costs of the stops and the reduction of the capacity.

To perform the project it was defined a methodology which allowed a correct approach of the problem and the development of a tool that surpass the inefficiencies identified by the responsible of the Production Management area and their future necessities as users.

In this document the methodology is shown, describing in detail each phase and the prototype which will be the support for the conception of a new tool and also replace the current one. In the end of the report it is described the advantages of the program, while capacity controller tool, to the Test area.

Agradecimentos

Gostaria de agradecer ao Eng. Jorge Sousa e ao Sr. Prof. João Falcão e Cunha por todo o apoio, motivação e disponibilidade oferecidas durante o projecto de estágio.

Agradeço à Qimonda Portugal e à Faculdade de Engenharia da Universidade do Porto a oportunidade de realizar este projecto.

Agradeço também ao Eng. Jorge Rodrigues, Eng. André Bento e Eng. Joel Geraldes pela disponibilidade, apoio e confiança que me deram ao longo destes seis meses.

A toda a equipa do Departamento de *Test Operations* que foi fundamental na minha integração e aprendizagem e que contribuíram de forma decisiva para a realização deste projecto.

Quero também agradecer à minha família, à Luísa e a todos os amigos que me ajudaram e incentivaram durante o estágio.

A vivência durante estes seis meses contribuiu de uma forma muito positiva para o meu desenvolvimento pessoal e integração no mundo de trabalho.

A todos, muito obrigado.

Table of Contents

Glossary

Terms	Definition
BO	Business Objects program
BOC	Board on Chip
CTF	Continuous Test Flow
DUT	Device Under Test
EPFTES	Tester reference
HR	Human Resources
IT	Information Technology Department
KPI	Key Performance Indicators
MSP	Mark, Scan and Pack area
NC	Non Conformant Quality
PL	Planning & Logistics Department
QC	Quality Conformant
QimTa	Qimonda Monitoring Test Area
QPT	Qimonda Portugal
RTC	Real Time Clock
TAKPI	Test Area Key Performance Indicators
TBL	Time Between Lots
TFBGA	Thin Fine Pitch Ball Grid Array
TSMON	Test System Monitoring
TSOP	Thin Small Outline Package
U.R.	User Requirement
UML	Unified Model Language
UPH	Units Processed per Hour
WIP	Work in Process

1 Introduction

1.1 Scope and Objectives of the Project

The Test Area represents over 50% of the Company's investment, therefore the Equipment Utilization is a critical performance indicator that in a certain way will measure the efficiency of the area. The aim of this project is to improve equipment's utilization and hence to optimize the production in the Test Area, increasing capacity and reducing time wasted in order to improve the efficiency of the process.

To achieve such aim, a monitoring tool for the Test Area shall be develop in order to improve the process capacity by giving more visibility of the performance problems. In other words, the objective is to develop a tool to online monitor and benchmark equipment capacity to identify low performers improving equipments' capacity and hence the general productivity. The other issue is to give a perspective of the Weekly Plan, in terms of Volume, Type of Machines, their utilization and capacity planned and then compare with the actual status to identify low performers and the deviation from the targets defined.

To summarize, the aim is to develop a tool to improve Test Area capacity and productivity, giving the alert when there is some deviation against the target planned, in order to identify low performers and correct the situation.

1.2 Project Phases

Figure 1 shows the milestones which were proposed for the project and the respective duration.

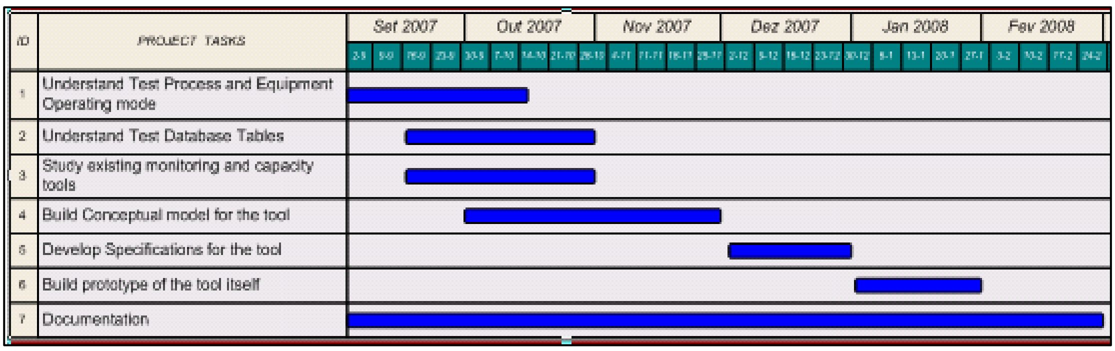

Figure 1 –Project Milestones

In the first phase, a training plan was created to give the knowledge about all the Test Area. The idea of this training was to deeply understand the Test Area Logistics (by studying the machines, tools and knowing the people responsible for all Test Area layout), the Operational Procedures (how they work to process the different materials, which factors the operators and line controllers need to have in account, how they follow the weekly plan, and so on), and all the information about the products' processes and flows.

In the second phase, there was a training plan to explain the different databases, their structure and relevant information with a special focus on the ones which were related with production indicators.

In the third phase it was done a study of the existing programs which were the basis for the model's creation. It was possible to define several factors that should be considered when the model was built. There are many programs to monitor the Test Area's production and the objective was to build a new one that could cover most of the information which is provided by those programs, but in a singular one, in a way to reduce the time spent in the identification of low performers responsible for the decrease in capacity and productivity.

In the fourth phase, it was created a Conceptual Model based on the study done in the previous steps. It was defined how the model should work, which inputs (Data Bases) and outputs should be given by the model and the relation between them.

The fifth phase was scheduled to create the Specifications for the program, which means that it was identified the way it will work, who will be the users and their permissions and restrictions related to the program's usage. This part was extremely important, once it was the basis to develop the program.

In the sixth phase, and based on all the previous five steps, a prototype was created with the objective of showing the functionalities of the program, the users' interface and to test the model in an initial stage by proceeding with the identification of errors or some redundancy that could be generated by the model.

During all those phases, the documentation with the requirements about all the stages was written since the beginning of the project.

1.3 The company – Qimonda

Qimonda AG

"The way we do business is just as important as the business we do"- Loh Kin Wah, Qimonda's CEO.

Qimonda is a leading global memory supplier with a broad diversified DRAM products portfolio. It was founded on May 1st, 2006 as a split out of Infineon Technologies AG, to form the third largest DRAM Company worldwide. Qimonda AG has it headquarter in Munich, Germany. The CEO-designate of Qimonda AG is Kin Wah Loh, who has been head of the Memory Products business group since 2005 and a member of the Management Board of Infineon Technologies AG since the end of 2004.

The name and brand identity of Qimonda express the philosophy and personality of the company, illustrating the vision and values that focus it. "Qimonda" has universal qualities that work across the globe: "Qi" stands for breathing and flowing energy. In the West, where languages are largely based on Latin and have been widely influenced by English, the interpretation as "Key to the World" (key-monda) is intuitive. While purple, the primary colour of the logo stands for leadership, the secondary colours, the cursive typeface, the round and organic shape of the logo and its impulsively spreading shape all emphasize Qimonda's core values: being ***creative***, ***passionate*** and ***fast***. [1], [2]

Figure 2 – Qimonda Logotype

Qimonda is a leader in 300mm manufacturing, and is one of the top suppliers of DRAM products for the PC and server markets. Infineon still controls a 77.47% stake.

Qimonda employs approximately 13.500 people worldwide, including 2.506 in R&D with access to five 300 mm manufacturing sites on three continents. It operates in five major R&D facilities, including its lead R&D centre in Dresden. (According to September 2007 facts)

Qimonda AG is listed on the NYSE with the ticker symbol **QI** and is based in Munich, Germany. The company has issued 342 million ADS shares, each representing one ordinary share of Qimonda.

According to June 2007 facts, Qimonda presents a 13.9% of market share in the DRAM's market, being only exceeded by Samsung (24%) and Hynix (21.7%) , as one can see in Figure 3.

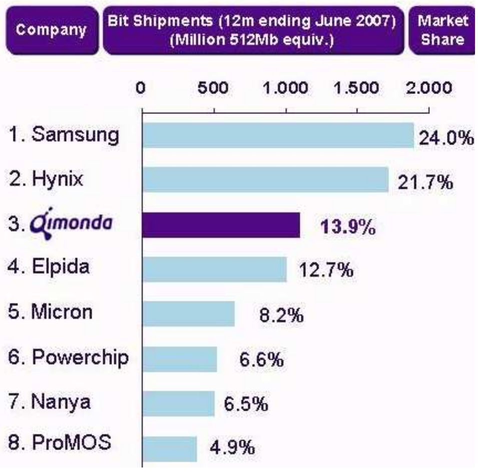

Figure 3 – Qimonda's Market share

Qimonda has a vast wallet of customers with a big dimension that support the idea of a company with high requirement standards and great future perspectives. Some of the companies are: *Hewlett Packard*, *Microsoft*, *Sony*, *Nintendo*, *Dell*, *Asus*, *LG*, *Toshiba*, *Intel*, *Sandisk*, among others.

Qimonda is all around the world in several countries, being the production sites situated in Suzhou-China, Malacca-Malaysia, Dresden-Germany, Porto-Portugal, and Richmond-USA. The company also works with the join ventures Inotera and Winbond, both in Taiwan. The different locations can be seen in Figure 4.

Figure 4 - Qimonda locations

Qimonda Portugal

Founded in 1996, Qimonda Portugal is the biggest Back-end[1] manufacturing company of the group. The company's activity is built in a flexible production and in a customer's necessity-based logistics, making Qimonda PT one of the most advanced Back-end sites in the whole world. Nowadays, Qimonda is the biggest export company in Portugal (all its products are exported) and the biggest company in the Electric and Electronic Industry sector in the country.

It has around 2080 employees and according to December 2007 facts its cumulative investment is about **€720 millions**. It has an implementation area of around **220.000 m^2**, which **17.000 m^2** correspond to production area.

QPT represents 7% of the memories' production in the whole world; this fact describes by itself the importance of the Portuguese site and its dimension overseas.

Figure 5 -Porto site

[1] Factory responsible for assembly and test the components.

With a historical emphasis on PC and server products, the company is now focusing on products for graphics, mobile and consumer applications using its power-saving trench technology.

There are two different types of components distinguished by the package type: the TSOP (Thin Small Outline Package) and the TFBGA (Thin Fine Pitch Ball Grid Array) also referred as BOC (Board on Chip). Both expressions, BOC e TFBGA, are used for the same type of components.

Figure 6 – TSOP (left side), BOC (center) components and their relative sizes (right side)

TFBGA is more recent than TSOP and it is expected to replace it in few years. Nowadays TFBGA is almost 80% of the production volume at Qimonda and TSOP only 20%.

TFBGA has many advantages when compared to TSOP: it is smaller, it has the same capacity and lower energy consumption; and it has even better performance. Although TFBGA and TSOP have the same basic functions, some processes and raw-materials are different.

Nowadays the final products where the company is focused are changing, following the technology improvement and the demand. Qimonda is changing its memories from products such as Desktops and Servers, to Laptops, Wireless and Wire line infrastructures and more recently, and probably the only focus products in the future, the consumer electronics such as Mobile phones, digital TV, PDAs, videogames, flash based MP3 players and GPS. This strategy relies on the asset of producing SDRAM memories to high add-value products, that obviously bring higher margins for the business thus, higher profit for the company.

Qimonda Portugal is a Back-end manufacturing company. Therefore it receives the wafer that is produced in a Qimonda Front-end site and then after a flow of different processes, transforms the input (wafer) in the final product for the client (the component) that later will be assembled in modules, in other Qimonda site.

Despite of being a Back-end site, the production process at Qimonda Portugal S. A. includes two Front-end operations: WLA/RDL (**W**afer **L**evel **A**ssembly/**R**e**D**istribution **L**ayer) and Wafer Test. On the other hand, *Back-end Operations* are divided in two areas: Assembly Operations and Test Operations.

Figure 7 allows the reader to understand better the company's layout.

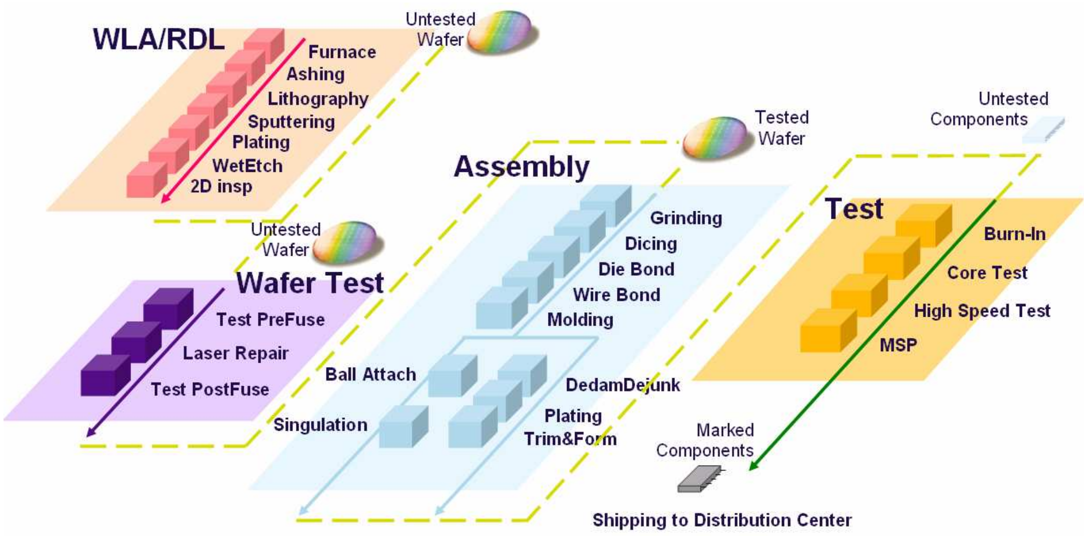

Figure 7 - Factory Layout

Since 2007, Qimonda Portugal stopped its production of Modules that had been transferred to Malacca; in the inverse way the production lines of components were transferred to the Portuguese site to increase the efficiency in Qimonda worldwide. Therefore, nowadays, QPT produces only components that after will be assembled into modules in another site.

In the Portuguese site, Qimonda has four main areas: WLP/RDL, Wafer Test, Assembly and Test Operations. The different areas will be briefly described bellow, although this project is inserted only in the Test Operations, more precisely in the "Core test" and "High Speed test" (see Figure 7).

a) Front-end Operations

WLP/RDL: The **R**e-**D**istribution **L**ayer is a metal layer that "redistributes" the central bond pads to the edge of the die. It has the objective to shape the wafers for a specific kind of components used in multi-chip packages. The RDL allows:

- Different configurations of pad layout
- staking of dies

Wafer Test: the Wafer testing is the final Front-end operation and seeks to test the wafers that come from different Qimonda sites before the production (or assembly) of the components that will be the final product of Qimonda. Although it is not a key operation in QPT, it is very important to ensure that each wafer (and each die that it contains) is according to the established Quality control parameters to be transformed in components later on.

b) Back-end Operations

Assembly Operations: this stage is divided in three parts: **Pre Assembly**, **Assembly** (Front of Line) and **End of Line.** There is a fourth one **MSP** (that before was part of Test Operations) that is the last operation in QPT, performed after the Test Operations.

The *Pre assembly's* purpose is to grind the wafer and to cut the individual chips into the final chip dimensions. In this operation the wafer is cut, although it stays with the same form because the cut wafer is still attached to a carrier and a sticky foil to prevent the chips from falling down. (See Figure 8)

Figure 8 – Wafer in Pre-Assembly

The *Assembly* process (also called Front of Line) is divided into *Die Bond* and *Wire Bond* Operations.

In the *Die Bound* operation (Figure 9) the individual chip is attached into the substrate electrical carrier, and is done in two different stages: firstly the chip is removed from the wafer and attaches it into the substrate and, after that, an adhesive cure is performed. In the *Wire Bond* (Figure 10), the die is connected to the substrate/lead frame through gold wire so that the electric contact is established between the die and the exterior.

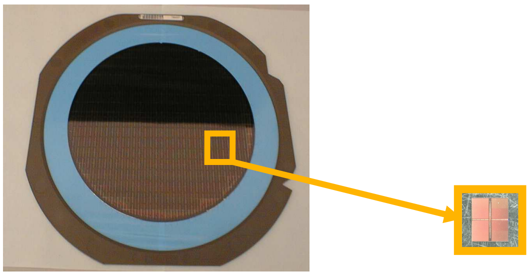

Figure 9 – Wafer in the Die-Bond

Figure 10 – Wire-Bond Operation

In the *End of Line* is important to distinguish between the flow of a TSOP and a BOC component. At this stage of the process, TSOP's and BOC's packages will perform one common operation and then they will diverge into different processes.

The common stage will be the *Molding* where the chips are encapsulated and bond channels with Epoxy Mold Compound to prevent damages against mechanical and chemical hazards.

The following operations will configure both types of components; BOC's will go through two other stages, *Ball Attach* and *Singulation*, where the "balls" will be attached in the component and after will be separated creating an individual chip.

The process for TSOP's is totally different, as they go through other operations such as: *DeDam/DeJunk Plating* and *Trim & Form*. In the last one, the final aspect will be different from BOC, being this configured with "legs" instead of "balls", as Figure 11 illustrates.

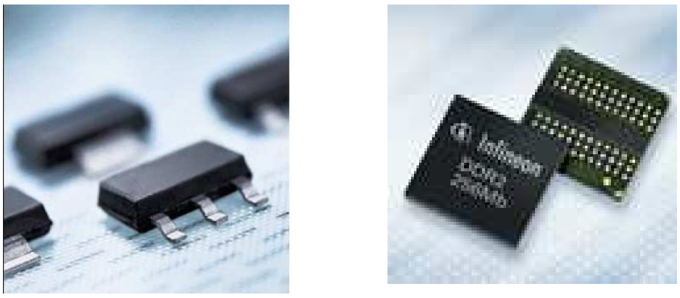

Figure 11 – TSOP (left side) and BOC (right side) final aspect

The last Assembly Operation is known as Mark, Scan and Pack (MSP), that like the name states is where the chip is marked, scanned for marking errors and dimensions, and is finally packed. The chips can be packed in reels or trays. Each reel has 1500 or 2000 chips depending on the specification and if packed in trays will have 1500 chips.

Figure 12 - A reel after the final package

Before being considered as an Assembly Operation, MSP was included in Test Operations because was performed just after the last test operation, but due to its nature now is considered as part of Assembly Operations.

The focus of this project is on Test Operations, more specifically in Core and High Speed Tests. Bellow, it will be explained each of these stages.

Test Operations are divided in three different stages:

- Burn In
- Core Test
- High Speed Test

Burn In is a previous stage that is done to reduce the infant mortality fails of the finished devices. The *Burn In* test has the objective to age the chip and consequently decrease the number of futures fails. After Assembly, the components are loaded (*Load operation*) in boards that go inside special ovens where they receive electrical and temperature stress. After, the components are removed from Burn-In boards (*Unload operation*) and the devices which failed the tests are segregated.

In the **Test** (**Core** and **Speed Test**) electrical tests are made to ensure that the components that will be sold respect the specifications. The tests' results are used to continuously monitor the Front-end and Back-end process allowing understanding the reasons behind the reduction of performance, being capable of improving the product and its yield.

In this stage there are several tests (also called production steps) which will be different from product to product. The main flow (named Core Test) includes low temperature tests (-10ºC and -25ºC) and high temperature tests (85ºC and 95ºC). There are some products that after the Core Test go to a Speed Test and others go directly to the MSP stage.

The next section contains more explanations about the Test area, as it is the area where this project is inserted.

1.4 Test Area

Test Area (also called *Test*) is performed right after the *Burn-In* process. In *Test* the electrical tests are carry out in each unit (memory) previously produced, in order to verify if they mach with the quality specifications. After this process, the good ones (called *pass units*) flow to the MSP stage. [10]

In the *Test* there are several operations (also called steps) that vary from product to product. The main flow (*Core test*) includes first low temperature tests and then high temperature tests. There are some products that after the C*ore test* go to a *Speed test* and other go directly to the MSP.

In the core and speed tests the products are processed in *Test Cells* (Figure 13) that include:

- *1 Tester*
- *2 Handlers*
- *2 Test Heads*.

The ***Tester*** is the "brain" of the cell, and is constituted by a set of electrical boards that generate electrical signals with adequate tension levels in order to stimulate the unit and analyse its reaction.

The ***Handler*** is responsible to place the units in one *Hifix* that allows the physical-electrical interconnection between the units' bolts and the *Test Head.*

The ***Test Head*** makes the connection between the Tester and the Handler. The Tester is linked with the Test Head by electrical cables and the connection between the *Test Head* and the *Handler* is done through a board called *Hifix*. The *Hifix* contains a set of *Sockets* (*devices*) that make the physical connection with the memories. In other words, the memories are placed by the *Handler* in the *Hifix* that is connected with the *Test Head*. The *Test Head* is responsible to send the information from the *Tester* to the *Hifix* where they will be tested and after send the results back to the *Tester*.

The tests are performed sequentially due to the pre-defined test program, and in the end, according to the results, a Bin Sort is made by the Handler that places the units into the respective *Stocker*[2] in order to separate the units according to their speed.

The Tester capacity and the number of available channels in each one defines the number of units the *Test Cell* can perform at the same time (in parallel). As an example an organization like "x16" needs more channels per unit that one "x8" or "x4" and some product are more rigorous that others in terms of precision and Tester speed.

The *Sockets* are the devices placed in the *Hifix* that make the contact with the memories that will be tested. One *Hifix* can contain between 16 and 256 *Sockets* according to its configuration. The Sockets can also be names as *DUT's* to refer to the position where it is placed in the board. For example, when one refers to *DUT* Usage, it is referring to the number of *DUT's* (or *Sockets*) that are being used to perform the test. The moment when the memories are tested through an electrical stimulation in the *Hifix* is called *Touchdown*.

The total processing time is divided in three parts: the *Process Time* (duration of the effective test), the *Overhead* (time that the memories are inside the Handler, but they are not being tested), and the *Time between lots* (portion of time since a lot is finished to the beginning of the next one).

The units from the same product type are grouped in *Lots* that have different sizes, i.e., depending on the number of units that are planned to be produced and the machines available, always trying to reduce setups, and hence increasing the efficiency of the productive area. After each operation, the units are separated and grouped into other lots according to their speed sort, in order to have lots that contain units with the same characteristics.

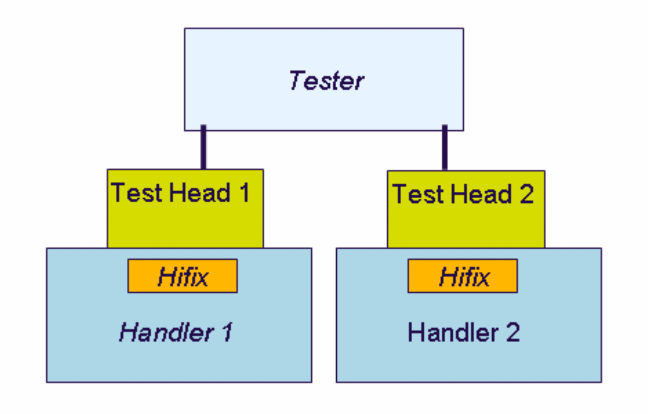

Figure 13- Representation of a *Test Cell* and its parts

More explanations about the Test Area can be found in Appendix A.

[2] *Stockers* - parts of the *Handler* used to separate the units according to the test results. The action of split the units in different stockers is called *Bin Sort*.

1.5 Structure of the Report

The report is organized in a way that the reader can find an explanation of the most important issues of the project in the main body of the document.

In the first chapter, the scope and the objective of the project are presented as well as a description of the organization and the project phases.

In the second chapter the project overview pretends to explain the reader what is the context of the project, the process and tools used nowadays for monitoring and planning the production in the Test Area, and summarize the issues of the project concerning these aspects.

In the third one, there is the Program Description, starting with the *User Requirements*, the *Use case model* (that explains which actors are involved in the environment of the program), the *Conceptual model* (an essential tool for the key developers) and finally, the *Program Specifications* that gives an idea of the high complexity of the program and its development.

In the fourth chapter, the Prototype (and its functionalities) is presented.

The fifth chapter describes the program that will be primarily implemented referring the differences comparing to the proposed model.

In the sixth chapter the author explains the key advantages of such program that will promote a large improvement in the area.

Last but not least, in the last chapter the author will make a conclusion of the project, summarizing the work developed and propose some future improvements.

Notice that some information used during the project is not included in the report for reasons of confidentiality.

2 Project Overview

This project is inserted in a critical area within the company, the Test Area. In order to better understand the project's problem, is important to know what the Test Area is and how the weekly plan and the monitoring process are done, in a way to understand what are the current problems and what shall be improved.

As it was explained before (in the Introduction chapter), Test Area is an area where the components, after performing many stages of conception and assembly, are tested. There are several types of tests depending on the type of products and for that reason there are different operations and machines associated. This complexity makes the Test Area one of the most critical and complex areas within the company.

This section is intended to give an overview of the project's context and is divided in four parts. The first is a brief description of the importance of the *Production Control*. The following part explains how the planning and monitoring process is done today, helping the reader to understand the need of improvements in this area. The third one is a complement of the previous section, explaining the actual tools that are used so that can be possible to identify low performers. Finally, the problems and general issues of the project will be described.

2.1 Production Control

Nowadays there is a general concern around the theme *Production Control*. This occurs because the control of the production is a very important area that can lead to huge improvements in the production area and the potential benefits are considerable and known. In a competitive market where a small improvement can lead to several cost savings, the need for improvement in the *Production Control* area is then a key aspect to achieve the production excellence.

The aim of this project is to try to improve the process of control the production in Test Area, increasing the machines' capacity by improving the problem's visibility and alert. As a consequence there will be an increase in the Volume production and even a Cost reduction, reinforcing the potential benefits that arise from this project.

To be capable of achieving this improvement it is urgent to analyse, in a first stage, the actual control/monitoring process to be able to find potential improvements that can lead to a change in the current processes.

The following sections will explain how the process is done actually, the tools used in the whole process and define some of the problems by identifying some potential improvements, explaining also how the process of planning the production is done, in order to contextualize the reader and understand some concepts that will be explored later on.

2.2 Actual Planning and Control process

Currently, the production plan for the Test Area is done in a *MS Excel* file (*Capacity Plan_Test.xls*) that traces the production plan for the following week having in account the product mix and its planned volume, the equipment available and their capacity. The Volume Plan and the product mix (parameters in the *Capacity Plan_Test.xls*) are previously set/ planned in the Planning & Logistics department that reports to the Test area.

It may seem that the production planning is an easy task and that is done in a fast and simple way, however the enormous number of machines, products and operations increase the system's complexity. Beyond this complexity there are, for each operation, several machines with different capacities, programs and tools that vary with the type of product. To give an idea of the complexity that exists, the Test Area works as a "single company" having more than 70 Testers and 140 Handlers, steps/operations with different temperatures, more than 50 combinations of tooling, around 2000 products references and 270 operators that imply a huge effort in terms of management skills and time spent. Moreover, it has a team with more than 40 engineers fully dedicated to this area.

Another point is that a Test Cell cost around 10 million € justifying the investment made by Qimonda PT in this area which represents more than 50% of the total investment

Today, all this planning is done in an *MS Excel* file, product by product, requiring dedication in terms of time spent that can increase the probability of human errors. This is a potential focus of improvement that will be discussed later on Appendix I.

After the weekly plan is done, the responsible for the production on the test area (production engineer) has to **monitor and evaluate** everyday all the products that are being processed. Therefore, if there is some deviation from the daily production targets, there is a need for evaluating the situation concerning the Key Performance Indicators (KPI's) in order to identify possible low performers that delay the production. In this case, some efficient corrective actions must be done in order to reduce the consequences and make the process as stable as possible.

Another aspect is the machine's capacity which varies with time, thanks to several factors such as, new programs, revisions, tooling and product variations. Therefore, there is a need to weekly update the capacity values in the production plan in order to have a more realistic scenario, consuming a large period of time of the production engineer once there are several combinations of machines, products and operations. Thus, another challenge was to create a mechanism that could make this updating automatically, reducing the time wasted for this function.

According to this, a tool shall be created with the objective of monitor the performance of the main indicators in the Test Area, reducing the time and effort needed for the evaluation of the area, making a user-friendly program that alerts possible deviations.

Moreover, some improvements for the actual production plan will be suggested in Appendix I, defining improvements for the production's calculation and management that, once implemented, will allow a significant reduction of the time spent when comparing with the actual process.

2.3 Planning and Control tools - Current situation

In this part the tools that are used in the production planning and monitoring procedures will be explained.

A) Planning tool – *Capacity Plan_Test.xls*

The *Capacity Plan_Test.xls* (also named Production plan) is the file where the production engineer plans the production for the following week. Despite of being total user dependent, which means it is done 100% manually by the production engineer, it is an essential tool that aims to plan and organize the machines' distribution in the production floor.

It will be followed an overview about the functionality of this tool, having in Appendix B the full description of this file.

The purpose of the *Capacity Plan_Test.xls* is to allocate the different mix of products to the different machines, having in account many conditions/restrictions. Once the production engineer's objective is to plan the production, the first thing is to consider the Volume Plan set by the *Planning & Logistics*. Generally, this volume planned, the number of machines available and their capacity are the starting points to plan the production.

Other important variables are the flow of operations that the product has to perform and the effective production time availability that corresponds to the number of hours available for production.

With all these variables it is possible to design a production plan as more realistic as possible. Having this plan on these terms it seems easy to plan the production, but in fact all of the variables are very complex. Going further in the analysis of the variables, for example, the machines' capacity varies with the type of product being processed and the type of operation that is performing. These facts affect the plan in a considerable way; if all the machines had the same capacity for all the production in the different steps, the plan would be easier, once all the machines could process all the products and the problems of setup-times wouldn't be a problem. Another scenario could be, for example, if the volume plan and the mix of products were constant. In that case, the plan could be always the same, with the machines processing the same products.

Once all this simplicity does not reflect the reality of the plan, the function of a production engineer and the production plan implicates hard work and knowledge about the whole area. Although this is not the focus of the project, some improvements will be suggested to improve it (see Appendix I).

B) Control tools

Nowadays, the control/monitoring process involves the usage of several tools with different functions to detect the low performers. This process can require much time spent and here is where this project is inserted, by reducing the time wasted in the production monitoring/control, allowing it to be more efficient and accurate in the analysis, leaving the time gained for other important tasks. The actual process is divided in different steps requiring different tools that are described in Figure 14.

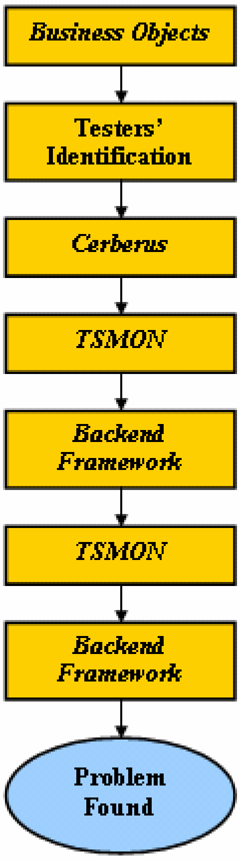

Figure 14 -Monitoring process roadmap (today's situation)

Figure 14 describes the complexity of the method that requires much time spent, so it has to be simplified.

Notice that the program is intended to monitor and find the causes that lead to problems of losses in production and define some actions that have to be implemented to solve the situation. The program will be useful to discover the factors that are causing the deviation in the machine's performance through the definition of some strategic KPI's used to evaluate the machines and their behaviour.

Therefore, the monitoring process ends when the problem is found and the person responsible for solving it is alerted.

As Figure 14 demonstrates, the process of monitoring the production requires several steps and programs that make it very slow, with a lack of flexibility.

The objective will be to reduce the number of programs (and hence the steps) that the production engineer as to analyse in order to detect the problems. As faster the process is, the more is the time gained to produce with the performance desired and the lower the time spent by the production engineer.

In the next section it will be made an approach of existing problems and possible solutions to improve the current situation.

2.4 The problem and general issues of the project

According to what was described above, there is an implicit need to improve the monitoring process in the Test Area. An effective control can lead to a better and faster problems' visibility that have significant potential savings.

One of the core aspects in a project is to identify the problem correctly and set the objectives that are associated with them. Bellow there are the main issues of this project and a brief discussion of them.

- Improve problem's visibility and alert (nowadays the problems are not so clear and due to the complexity of the analysis the people become relaxed and does not go so deeper in the analysis of the problems)
- Reduce the number of programs used in the production monitoring and control
- Reduce the time wasted on finding the root causes of the problems.
- Define the main KPI's (*Key Performance Indicators*).
- Set a more effective roadmap to monitor the Test Area.
- Increase the capacity and productivity in the Test Area
- Reduce the costs in the Test Area.

The issues pointed out above are connected and lead to a common objective: to increase the efficiency of the Test Area by evaluating and measure it through the definition of some KPI*'s*. *Key Performance Indicators* are metrics used to help an organization define and measure the progress toward organizational goals. They depend on the nature of the organization and the organization's strategy.[4] In the context of this project, the defined indicators must be align with the production strategy of the area, having a special focus on the ones related to Capacity and Tester utilization issues. By monitoring the Capacity, it is possible to know if the machine is testing according to the targets established and, if not, there is the need to detect the problems associated. On the other hand, the *T*ester's utilization will focus more in the process efficiency in terms of time spent in effective production.

To summarize, the definition of the correct KPI*'s* it is an essential part of the project once they will be the agents used to monitor the whole process by defining the main causes of the problems that will lead to the improvement of the process.

In order to make a correct approach of the problem and to develop a powerful tool, the users' feedback will be a key concern. The involvement of the users in the creation of a new program will be a precious task to define several features of the tool. The conception of a new tool is a complex matter, and it justifies the creation of a framework, illustrated in Figure 15, that will lead to a correct definition of a tool with visible benefits.

This framework was used as a roadmap to describe the different steps of the project which will be resumed in the next sections of this report. Notice that the last part (the Program) is the development of the tool itself which responsibility is from the IT Department.

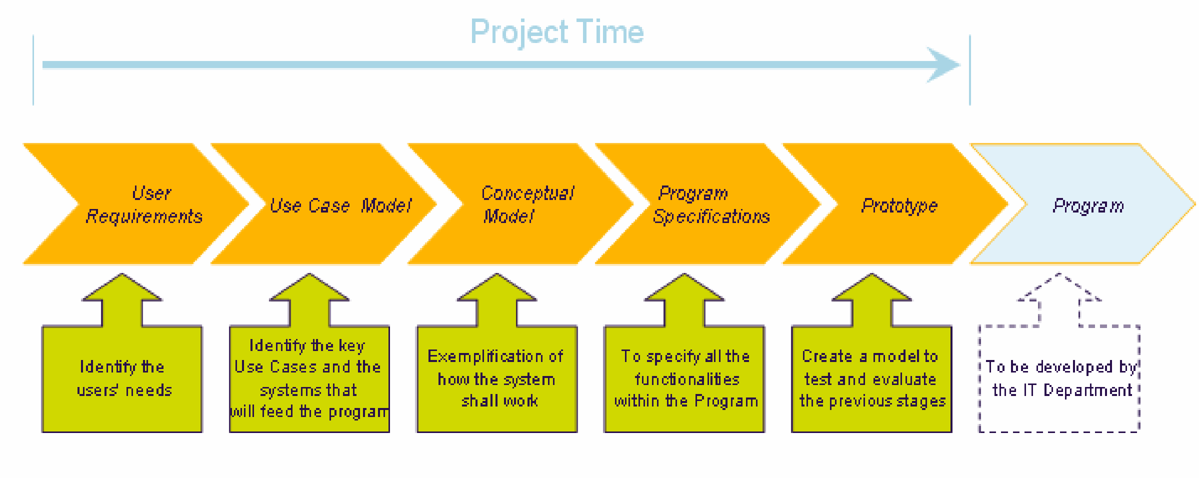

Figure 15 - Project phases

3 Program Description - *QimTa*

3.1 General Description of the Program

QimTa[3] is a program that was created to improve Test Area Capacity. This program will monitor the actual production performance and display the plan for a specific week, *Package*, *Product type* and *Configuration*[4] alerting in those cases where the volume produced is lower than planned and in the cases where the machines are having low performance, in order to try to detect the reasons of such situation.

It will also give the possibility to select a *Tester* and evaluate the main KPI*'s* in order to find the root causes from the lower performers allowing a fast reaction to the problems.

The program intended to reduce the need for the use of several programs, allowing a more efficient and flexible analysis, enhancing the problems in the area.

3.2 User Requirements

Now that the context was explained, is time to explain the user's requirements that the future program must have in order to improve the efficiency of the Test Area and answer all the user needs. *User Requirements* are defined as "the requirements governing the project's deliverable or product as expressed by the users". [5]

All the programs are made for the users and this is not an exception, so it was decided to talk with all future users (the actors of the system) to have a feedback of what shall be done in a positive way, to make a better analysis of the situation in the Test Area.

Based on the user suggestions, a list of *User Requirements* was made to fit all the user needs. The list of *User Requirements* is described as follow:

U.R.[5] ***QimTa* 1:** The user has a selection menu where is possible to select a filter for his analysis. He must choose the Fiscal week and Year that he wants to evaluate as well as the type of Product that wants to analyze.

---

[3] *QimTa* stands for **Qi**monda **m**onitoring **T**est **a**rea, enhancing the words **Qimonda** and **Test**, defining the area where the program is inserted.

[4] See "Appendix C : Granularity" to understand this concepts

[5] U.R. – User Requirement

U.R. QimTa 1.1: The user can differentiate his analysis. It can have an overview by Package Type, or can go further in his analysis and select the results by Product Type, or in the last case it is possible to specify the Configuration that will be object of analysis.

U.R. *QimTa* 2: The system shall provide relevant information about the Production Plan in order to later evaluate the possible deviation that can occur during the week and try to align the users with the targets for a specific week, decreasing the lack of information among the people in the Test Area.

U.R. QimTa 2.1: The system shall provide the weekly Volume Plan for the selection specified previously by the user.

U.R. QimTa 2.2: For each Test Operation that is indicated, it shall give information about the type of Testers that are processing that product in that step, which percentage of the machine's time has been allocated to that product and the capacity (Units per hour) that is planned for that machine. This functionality will be useful also to have an overview of the flow that the product has to perform and the type and number of machines that shall be allocated for each operation.

U.R. *QimTa* 3: The system shall allow the monitoring of Test Area, displaying the information about the actual performance of parameters related to production (specially the equipments), allowing the user to see the deviations from a given target.

***U.R. QimTa 3.1*:** The system must give information about the Volume that has already been tested in the different operations and the percentage of the planned volume that have already been processed. The program must give also the information about the Volume that have been processed but is not considered for the Volume Plan, once it have different specifications and hence is not considered as Volume out for production. The volume is defined as NC referring to components that are non conformant with the client's specifications.

***U.R. QimTa 3.2*:** The system must indicate what are the machines (Testers) that are producing the selected product in that specific week, giving information such as the Number of the machine (reference of the machine used to identify each machine in the line) and the type of *Tester* and the Operations that is performing it or have already performed. This is important because there are machines that can perform two types of operations (one of each time) and one shall know it in order to evaluate where the problems are.

U.R. QimTa 3.2.1: There shall be an alert when the *Tester* is working in an inefficient way so the user can find what the problem the product is facing. This alert shall be signed for each machine with lack of performance and displayed in the field of each machine.

U.R. *QimTa* 4: The user must have the possibility to select one of the machines that are processing the selected product (see **"U.R. *QimTa 3.2*")** and evaluate their performance. The indicators[6] will be shown in a different window and are: *UPH*, *Tester Utilization*, *Time between Lots, Lot size and Yield*.

***U.R. QimTa 4.1*:** About the *UPH* indicator, the value of Units per Hour for that machine when processing that product is showed and the user can compare the value planned with the actual value. If the deviation is pronounced, the system shall give the alert to inform the user of that situation. If the user wants to go further in his analysis, it is possible to select a function that will open a new window with other KPI*'s* that may found the root causes for such deviations. The new window contains charts representing the following KPI*'s*:

- **Contact Performance:** It measures the Tester's performance, more precisely the *Hifix* efficiency in terms of number of contact fails in the sockets**.**The higher the Contact Fails, the lower will be the Tester's performance that will be translated to a decrease in machine's capacity.
- **DUT Usage:** The purpose of this KPI is to report the *Hifix* performance and the utilization of Capacity. The DUT usage report is used to monitor the utilization of the DUT's available in the Testers. This indicator is also important to evaluate the capacity in the test because the units are tested in parallel, i.e., at the same time, and this indicator will measure the performance of each one individually.
- **CTF Usage:** It measures the % of volume (kpcs) processed in CTF[7] and Conventional mode.
- **Touchdowns:** this KPI gives an idea of the average number of touchdowns performed every 30 minutes, allowing the detection when there are some values out of control. In this way, this indicator can help to explain a reduction in capacity, once the lower the number of touchdowns, the lower the capacity, because a decrease in the value means that for the same period of time (usually 30 min), there were less units tested.
- **Handler Alarm Statistics:** This KPI intends to demonstrate the number of occurrences (alarms) in the system. It is a valuable parameter, once the occurrences can increase the *Overhead* and *Time between lots*, decreasing in this way the time of effective test. Thus, it leads to the reduction of units tested and decrease of the machines' capacity.

U.R. QimTa 4.2: The fields included in the *Tester Utilization* indicator are intended to give an overview of how the process is working in terms of time distribution, giving information about the percentage of *Processing Time* and *Overhead* [8] as well as the deviation from a predefined target. If the user wants to go further in his analysis, there is the possibility to select a function that will open a new window containing other

---

[6] The indicators and their importance in the program will be explained in the section 3.5) Program specifications

[7] *CTF* means *Continuous Test Flow* that is a processing mode that aims to produce the batches continuously, without stops, what makes the machines utilization more effective, allowing an increase in about 20% of Testers capacity.

[8] The Total Time is divided in Process Time (duration of the pure test), Overhead (the time the product is inside the machine but is not performing the test) and Time between Lots (TBL).

indicators that can detect the root causes that explain such deviations. The menu contains graphics with the following KPI*'s*:

- **Handler Alarm Statistics:** As it was explained before, it is useful for monitor the capacity of the machines, for the reasons that occurrences/alarms in the Handler are seen as low performers that reduce the capacity. This KPI is duplicated once it is important to evaluate both *UPH* and *Tester Utilization*.
- **Time Distribution:** This KPI measures the proportion of productive and non-productive time in the Testers. It is divided in three parameters: *Process time* (effective test time/pure test), *Overhead* (the index time, where the units are inside the handler but are not performing the pure test) and *Time between lots*.

U.R. QimTa 4.3: the *Time Between Lots (TBL)* pretends to monitor the evolution of the time between lots and its respective operational efficiency comparing it to a predefined target value. Ideally this value tends to zero.

U.R. QimTa 4.4: the Lot size indicator shall display the average value of the Lot sizes that have been processed in that machine. This indicator is very useful, once it can explain some deviations and values out of control, specially related with the *Time between Lots*.

U.R. QimTa 4.5: the *Yield* indicator is important to analyze the machines' performance. If this value is lower than the target value (predefined by the production) an alert shall be given.

U.R. *QimTa* 5: The system will actualize the value of *UPH* planned in the file used for plan the production, in a way to make a better approach to the actual performance of the machines, making a better forecast for the next week.

U.R. QimTa 5.1: The model will actualize the value of *UPH* planned based on a rule that evaluate if it is worded to change that value. This will be based on the following rule:

If the abs (UPH deviation >20%[9]) → UPH plan = UPH Actual

This value will be updated in the *Capacity Plan_Test.xls* file, but will be used only for the following week.

Unfortunately, not all the *User Requirements* were possible to include in the program, for several reasons. There were many users with the same requirements and that were priority for the developing of this tool.

The last User Requirement explained before ("U.R. *QimTa* 5") it will not be included in the first version of the program, once it would implicate several changes and hence the time to develop the tool would increase in a considerable way.

Although, this user requirement can be very useful, once it can reduce the time spent by the production engineers when the update of the fields related to the machines capacity.

Despite of this fact, some suggestions will be given in order to improve the actual Production Plan (see Appendix I).

[9] The calculations that justify this value can be found in Appendix L

In the Appendix D one can see all the user's suggestions fully described and have an idea of the short deviation between the user suggestions and the *User Requirements* that the model contains.

In the next section a *Use Case Model* will be explained in order to the reader understand the whole system, since the *Actors*, to the *Use Cases* and the relations between them. These have a strong linkage with the *User Requirements*, once the use cases were created having the *User requirements* as the starting point.

3.3 Use Case Model

In this section, it will be explained the *Use Case Model* that was created to make a first approach of how shall be the program, its functionalities and the actors involved in the whole system. The decision behind the creation of this model is related with the fact that it is a very useful representation of the system and its general functionalities. Moreover it was used as a mean to clarify and specify the program and the actors evolved.

A *Use Case Model* is a technique used in software and systems engineering to capture the functional requirements of a system. It describes the interaction between a primary actor—the initiator of the interaction—and the system itself, represented as a sequence of simple steps. *Actors* are something or someone which exist outside the system under study, and who (or which) take part in a sequence of activities in a dialogue with the system, to achieve some goal: they may be end users, other systems, or hardware devices. [6]

The Use Case Diagram

In a way to the reader understand the whole system, since the use cases to the actors and relations between them, a *Use Case Diagram* was created so anyone can design the program based on the *User Requirements* and this diagram. A use case diagram is a type of behavioural diagram defined by the UML. "Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals - represented as uses cases – and any dependencies between those use cases."[7]. Above one can find the *Use Case Diagram* (Figure 16) and all a list of Use Case descriptions that can explain the intension and functionalities related to each Use Case mentioned in the Diagram (Table 1).

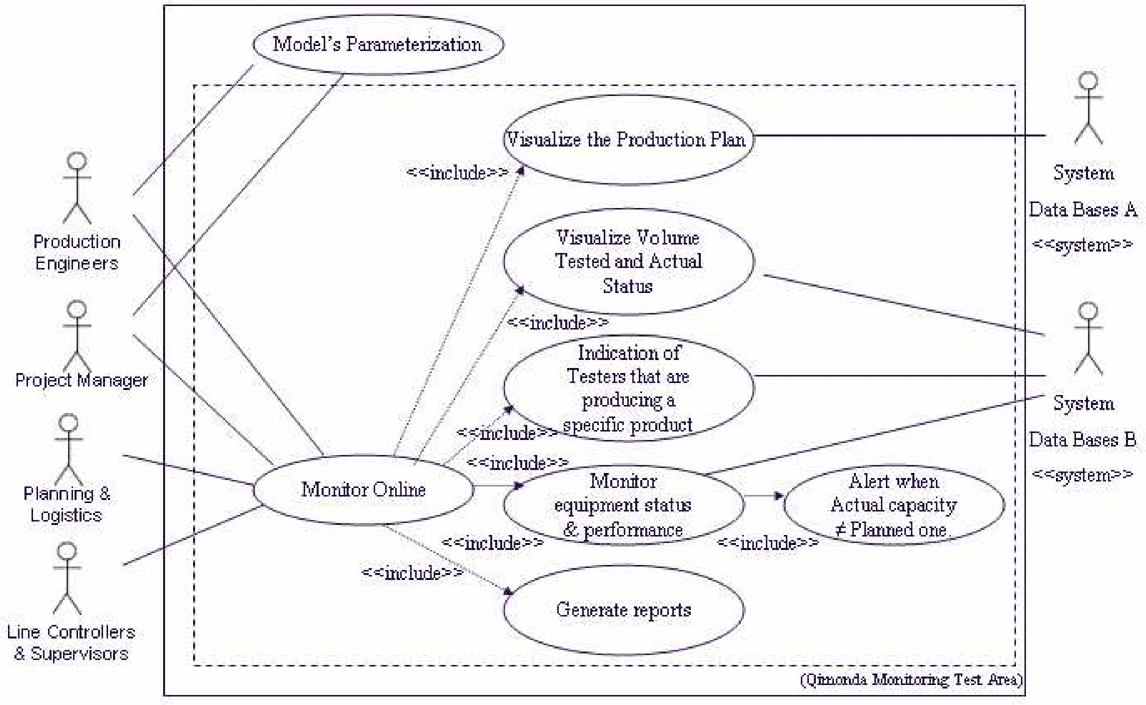

Figure 16 - Use Case Diagram

Table 1 - Use Cases' description

USE CASES	DESCRIPTION
Monitor Online	To monitor the activity and status of the production in the Test Area, comparing with the Production Plan, for a specified product in a specific date. It includes the next six Use Cases that are described above.
▪ Visualize the Production Plan	The program shows the Production Plan for a specified product in a specific date.
▪ Visualize Volume Tested and Actual Status	The program displays the Volume that has already been tested as well as the actual status, when comparing with the production plan.
▪ Indication of Testers that are producing a specific product	It also indicates the Testers (machines) that are producing a specific product in a specific date.
▪ Monitor Equipment Status & Performance	After having the information about the Testers that are processing the product, the Actors have the possibility to evaluate the equipment status and performance.
▪ Alert when actual capacity is different from the planned one.	It gives the alert when there is a reduction in the machine's performance, based on the deviation in capacity. (included in the previous one)
▪ Generate reports	It has the possibility to generate reports of the analysis made, in order to state the information provided by the program.
Model's Parameterization	It is an important function, once it gives the possibility to change the program's parameters in order to update the system with the actual specifications and targets.

The other features of the diagram are described as follow:

There are various **Actors**. The primary actors are the *Production Engineers*, *Project Manager*, the department of *"Planning & Logistics"* and the *Line controllers & Supervisors* (taken has one actor, once their intention concerning the model is the same, and they are located in the same area of the company).

As secondary (or supported) actors there are the *System Data Bases A and* the *System Data Bases B.* The first is responsible to "feed" and upload the model with data related to the Production Plan, more specifically, the volume plan, product flows, type of machines planned to process that product and their capacity. It is also responsible to give information about the planned percentage of capacity utilization for each test operation and tester type.

System Data Bases B is intended to give the feedback that allows the system to monitor the performance and behaviour of the equipment and their actual capacity.

The **Relationships** that are presented in the model are easily understandable. Beyond the ones there were explained before (the relationship between use cases were almost described inside the discussion of the Use cases definition), the relations between the Actors and the system are clear; the *Production Engineers*, *Project Manager* and the department of *"Planning & Logistics"* have access to all the system and can manipulate data and generate reports with all the information, being the Production Engineers the main responsible ones for the system and the analysis. The *Line controllers & Supervisors* only have access to the *Monitor* sub-system (and the use cases that are included on it), having the only need to analyze and evaluate the performance and status of the equipment. The relations between the supported actors (System Data Bases) and the Use cases were already explored in the previous paragraph when the role of the actors was explained.

About the **System Boundary Boxes** it is possible to define one principal boundary, ***QimTa,*** that involves one sub-system (marked with dashed square) the *Monitor Online*, and a Use Case outside of it, the Model's Parameterization that is a functionality related to the first but it shall be kept outside, once it is not an "Online Monitoring" functionality.

In the Appendix E one can explore these concepts in more detail.

3.4 Conceptual Model

Now that the model was briefly described, it is important to create a *Conceptual Model* that will be the basis to the program's development. The purpose of the model is to describe the relation between the different databases and the program, as well as the possible interaction made by the user when is running the program. A *Conceptual Model* is a model that describes the general functional relationships among components of a system.

The model was created based on the previous stages of the project (*User Requirements* and *Use Case Model*) in order to fulfil the user needs and try to make a relation between the databases that exists.

This model represents the *Qimonda Monitoring Test Area*; a program created to match all the user's requirements and pretends to be a power and user-friendly tool.

It is intended to be very easy to access and the functionalities are very intuitive and easy to learn how to work with.

In the Figure , there is a scheme that represents the Conceptual Model as the representation of the different parts of the program and their linkage with the “systems” that feed them in order to obtain all the data that is required.

It is important to explain how the system will work. Based on the Conceptual Model one can see that there are various relations between the different parts and those relations shall be fully explained.

This is the aim of this conceptual model, to show the linkage and relations of all the objects within the system and to make an overview of the different flows between the objects that will be explained in detail.

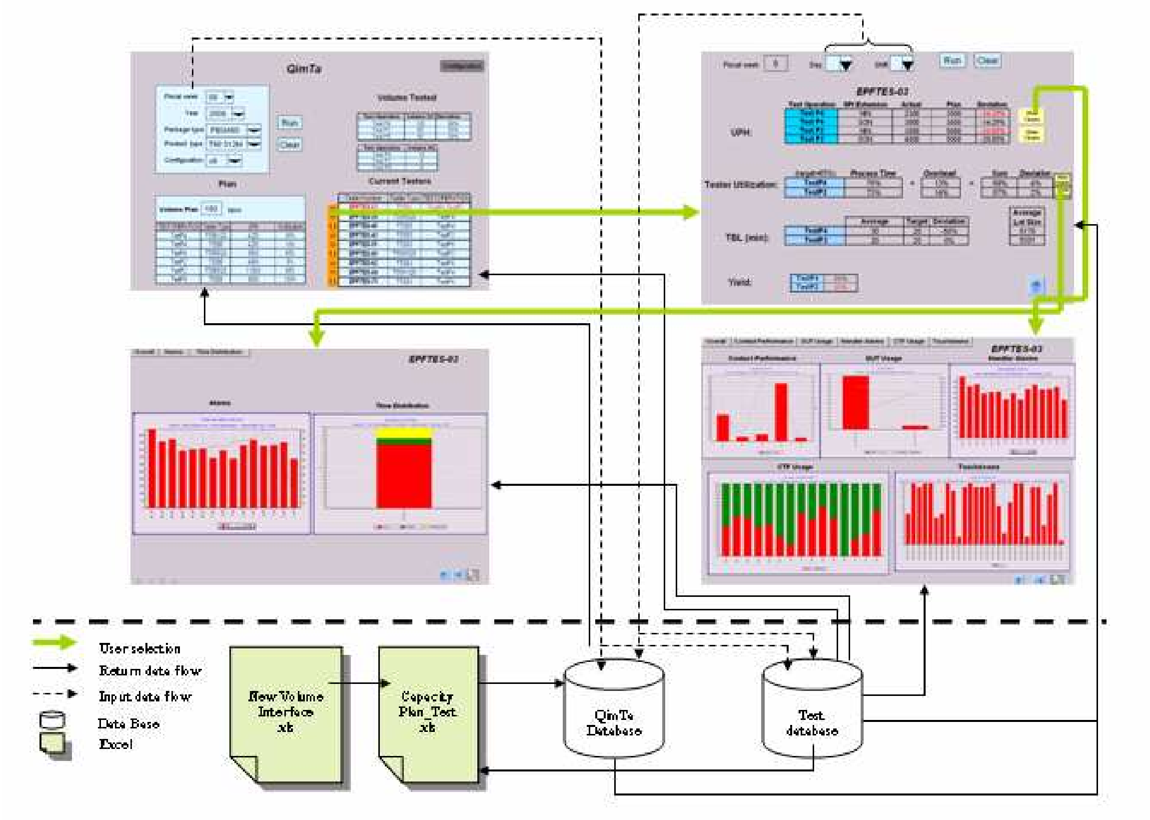

Figure 17 – Conceptual Model for *QimTa* program

First of all, in the bottom of the figure there are the databases and *MS Excel* worksheets that will feed the *QimTa*. The dashed line was used to show that the objects under that line are impossible to modify by the user. In other words, the user is limited by the output given by them, being these disabled for the user. Notice that some actors will have access to those objects (as referred in the *Use Case Model*, according to the Model’s Parameterization) but the description of those actions are not relevant for the *Conceptual Model*, once its only intention is to explain the actions performed to get the analysis and not its configuration.

The four objects that feed the program (under the dashed line) are essential parts within the system and they cannot be removed, otherwise the program doesn’t get the outputs expected. Here is possible to detect one of the few disadvantages of this tool, like almost all the tools: it depends on other objects and therefore if one of the databases crash, the system cannot return the outputs that were pretended.

The systems that will feed the tools have relations between them. The *Capacity Plan_Test.xls* used by the Production Engineer is feed by the *New Volume Interface.xls* file (more precisely, the field named "Volume Plan") which responsibility is from the *Planning & Logistics* department. The file is continuously updated hence the volume plan of the *QimTa* tool will vary at the same time. The relation is represented in the Conceptual Model by the arrow from the *New Volume Interface.xls* to the *Capacity Plan_Test.xls*. On the other hand, the *Capacity Plan_Test.xls* will feed an intermediary database that makes the bridge between the two worlds, i.e., to be accessible through the *QimTa* tool, the existence of this table is necessary to make the data more flexible and more easy to be manipulated by the program. This table is extremely important once it transforms the actual *Capacity Plan_Test.xls* done in *MS Excel* into database tables that will be responsible to return the values called the "Planned values".

More explanations about those two files can be found in the Appendix B.

To finish the relation between the systems, there are one important relation between the Test Database.mdb and the *Capacity Plan_Test.xls*. This aims to generate a function that evaluate the deviation between the UPH planned and the actual UPH, where deviations greater than 20%[10] will make the program update the value of the UPH planned to the actual UPH in order to have a more realistic scenario and save some time wasted on this follow up made by the production engineers (as referred in the U.R. *QimTa 5.1*). The functionality will be implemented in a second phase of the development.

There area also the action buttons "Run" and "Clear". Those are not more than buttons that give the order to the system to generate and clear all the values, respectively.

Finally, there shall be a Configuration window "behind" the system. This window will be used to specify the different parameters of the program and will be described later in this document, more precisely in the "Program Specifications" chapter. This configuration window will allow the Model's Parameterization described above in the *Use Case Model*.

Following the three different flow types represented in the diagram will be described.

The User Selection Flows

Focusing on these flows, represented with green arrows, the reader can easily see the possible user actions, i.e., what the user is able to do inside the program and what will be the information returned by each action.

The first one it happens when the user wants to analyze the performance of a specific machine that is shown in the Main Menu Window (after making a selection in the Selection Menu and the results are generated). This action occurs when the user presses one of the buttons closed to the machine that it refers. Therefore, if some of the buttons is pressed, a new window will appear; in this case it will be the General Performance window respecting to that selected machine.

Another possible flow occurs when the user presses the button close to the UPH indicator. Usually this button is pressed when a deviation higher than 20% occurs, but the user can press it no matter what is the value of the deviation. After the button is pressed, a new window will

[10] The calculations that justify this value can be found in Appendix L

show up, the UPH Indicators Window. Here the user will have the possibility to see many different charts that can help to explain the UPH performance.

The last relevant flow is similar to the one explain immediately before. The difference is that it will occur when the user presses the button close to the Tester Utilization indicator. A new window will be displayed with two relevant charts that can help to explain the reason behind such Tester Utilization performance.

It is important to refer that these are not the only user selection flows. In fact, these are the more important ones and the ones that shall be explained. However, there are other possible user selections. For example, in the various windows there will be a button called “Home” (represented with the symbol of a House) that, when pressed, it will make the program to go back to the Main Menu Window. Another possible action occurs when the user presses the button “Go back” (represented with and arrow) that allows the user to go to the previous window displayed. There will be also the possibility to “Print”, action that is related to a button represented by an arrow. The printing action is able in every window.

Last but not least, there are the “Run” and “Clear” buttons that shall be considered. Despite of they are not real flows, they are an important user selection once, they make the program run (in the case of the “Run” button) and are helpful for eliminating errors and make the selection faster (referring to the “Clear” button”).

Return Data Flows

As the reader can observe, there are several Return Data Flows represented with a black arrow. Some of them have already been explained, such the flow from *New Volume Interface.xls* to *Capacity Plan_Test.xls*, from *Capacity Plan_Test.xls* to *QimTa* Database and from *Test Database* to *Capacity Plan_Test.xls*.

What concerning the other flows, there are two that flow from the *QimTa* Database: one to The Main menu Window, more precisely returns the values to the fields that are included in the “Plan” part; and another from *QimTa* Database to the General Performance Window, more precisely to the fields related to the Plan.

There are also four more flows that derive from the *Test DB*: one flow for each window represented in the figure. The reason behind all this flows is because all the Data from the actual production will be fed by the *Test DB*, once there is information about the production in all the windows; therefore, a flow has to be included for each one.

Input Data Flow

The Input Data flows are conditioned by the user’s selection. Therefore, these flows will be generated from the selection menus until the Test DB and *QimTa* DB, that after will generate the “Return Data flows”. Thus, there are four different flows: one from the Main Menu window to the *Test DB*, another from the same source to the *QimTa* DB; in the same sequence there will be a flow that goes from the General Performance window to the *Test DB* and another from the same source (General Performance Window) to the *QimTa* DB.

3.5 Program Specifications

The next phase of the program's development was to define the *Program Specifications*. This is only possible after the previous stages have been complete, once they give feedback for developing the program. Therefore, the specifications were made based on the *User Requirements*, where it was possible to identify the needs and functionalities that the program shall have, the *Use Case Model* that was useful to know the *Actors* and understand the whole system, and finally the *Conceptual Model* that illustrated how the different parts of the system were related and what are the sources of information that feed the program.

In this stage, is important to have access to many points of view about the problem, in particular the descriptions from the experts about the problem, to all the reports that exist in the company and the system will have to produce and even maintain a strong interaction with the actual and future users, in order to obtain their opinion and criticize the model. For these reasons, they have to understand the model, and the specifications will lead to obtain this target.

In this chapter the specification for the tool will be described. This is an important part, once it will be the basis for the tool development by the IT Department.

This section is divided in four parts, each of which corresponds to different menus of the program, being the explanation order the way the user shall use the program in order to make a good analysis and try to find out the problems that are occurring in the production line.

A) Main Menu Window

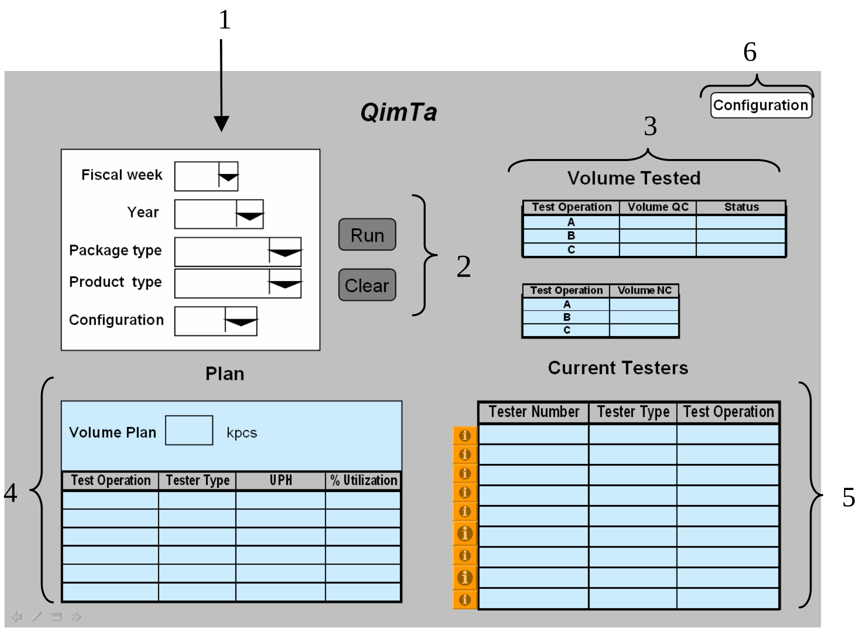

Figure18 - Main Menu Layout

Selection Menu (1):

Like the name states, this is the main menu (Figure 18), the first contact between the user and the program, where the filter selections are made, i.e., where the user introduces the inputs according to what he wants to analyse and what the system shall return.

This menu has five different fields: “Fiscal Week”, “Year”, “Package type”, “Product type” and “Configuration”, each of which will filter and influence in some way the results given by the program.

It is important to clarify that this menu is intended to filter in a way that cannot lead to mistakes. Therefore, the selection is intended to be made in a gradual way, having the user to select the different filters in different steps. In other words, the user has to choose first the “Year”, “Fiscal week”, then the “Package Type”. If he intends to go deeper in his analysis he can select the “Product Type” and then the “Configuration”, by this order[11].

To sum up, it is only possible to select “Package Type”, “Product Type” and “Configuration” if the previous fields have already been selected.

As a representation, the field selection sequence has to be:

Fiscal Week → Year → Package Type → Product Type → Configuration

This selection will influence the results once it will filter only the products with that specification. Therefore, is important to have in consideration that, from this point ahead, all the specification will have this in consideration.

Action Buttons (2):

These two buttons, “Run” and “Clear”, are used to execute two different actions. The first is used to run the program according to the selection previously made. The second clears all the program fields.

Volume Tested (3):

It is a group of fields that display the Volume that was tested until the moment of the menu selection, divided in terms of Grades QC (Volume with a better quality and which is planned every week) and Grades NC (Volume with lower quality that is not planned but needs to be considered, once it is produced).

In the case of the Volume QC, a field will be displayed with the proportion from the Volume planned (the target value) in order to understand if there are some delay when comparing to the production plan. It is a very important functionality once it alerts the user for the need of some change in the production in order to execute the weekly plan.

“Plan” Sub-Form (4):

This area of the program is linked with the “Transition table” that is linked with the *Capacity Plan_Test.xls* file. This means that, in some way, these fields are linked with the production plan.

The aim of these data is to return the values from the Production plan for a specific “Package”, “Product type”, “Configuration” in a specific date, depending on the selection

[11] The definitions of “Package Type”, “Product Type” and “Configuration” are explained in the Appendix C - Granularity

made by the user. This sub-form returns the weekly plan for a specific filter in terms of flow of operations, machines required and their capacity (in UPH).

"Current Testers" Sub-Form (5):

This area of the program is linked with the "Test Data Base" tables. The aim of these data is to return the values from the current status in the Test Area for a specific "Package type", "Product type", "Configuration" in a specific date, depending on the selection made by the user.

The importance of this functionality is to provide information about the machines that are (or had been) processing the specified product, the type of operations that are performing and even give the alert when they are processing with low performance. In this case, the row that corresponds to machines with lower performance appears with a red colour giving the idea that something shall be made to correct the situation.

Another important feature that the tool includes is a button that is closed to each row of the table. This button is a link to another window that aims to display some important KPI's that helps to understand the machines' performance.

Therefore, if a red colour is displayed in a specific row, the user can press that button and easily see the different KPI's and their deviation from the planed performance.

To illustrate a situation where a specific product in a certain week has been selected, the Figure 19 can exemplify what shall be the program's layout in the Main menu.

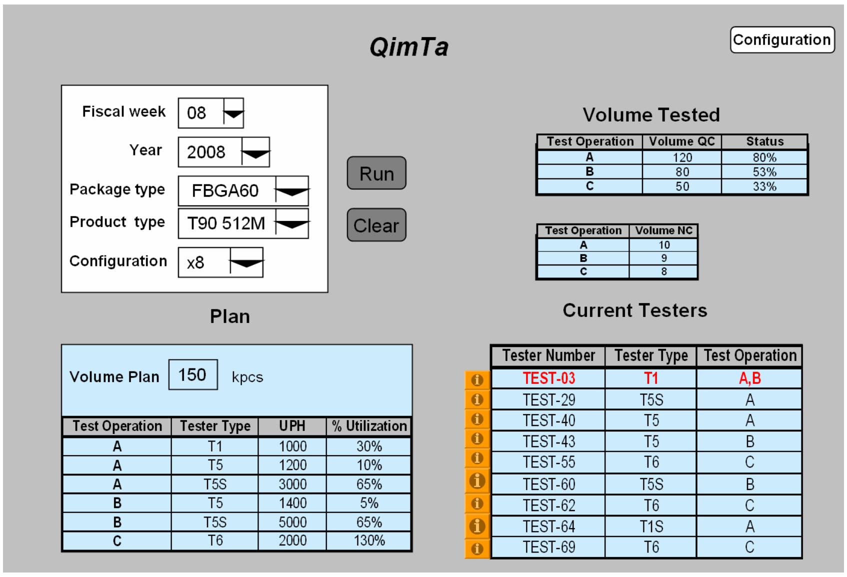

Figure 19 – Example of the results made for a specific selection.

As the previous picture states, this window can give an overview of the production status for a specific product in terms of volume tested vs. planned, the Testers that is producing it and even an alert when they are with a bad performance. As an example, in the figure above, for the product "FBGA60 T90 512M x8" in the Fiscal week 08 in the current year 2008, the Volume plan is 150 kpcs (150.000 units) and one of the nine machines that are producing the product (TEST-03) is with an alert in form of a red colour, indicating the lack of performance that probably is influencing the production of that component.

B) General Performance Window

As it was explained before, this window appears has a consequence of a selection made for a specific machine in the *Main Menu Window*, in order to analyse the more relevant indicators that can justify behaviour in terms of performance.

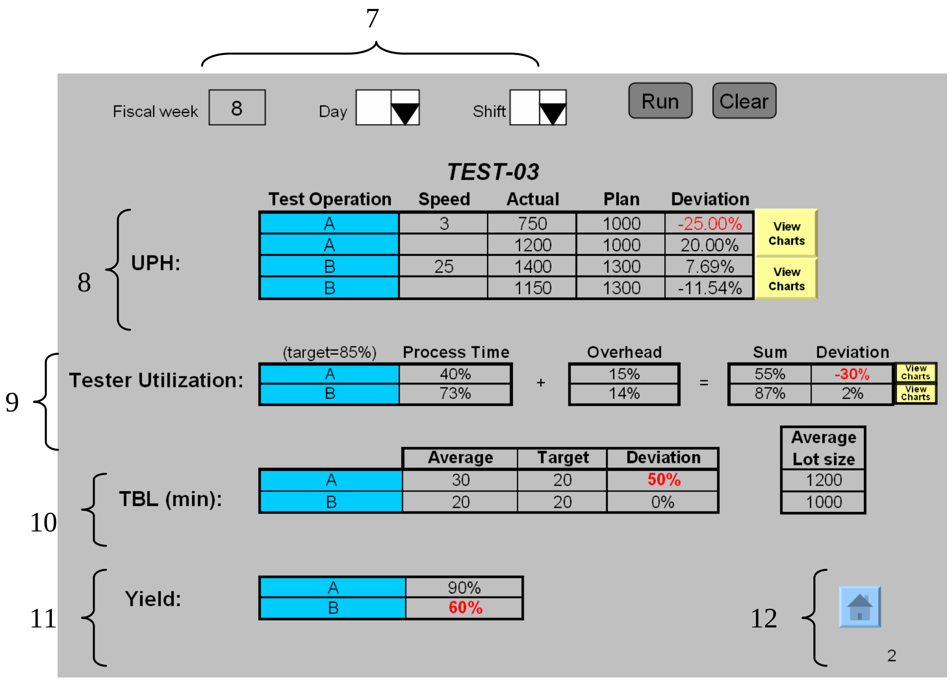

Figure 20- General Performance Window for the Tester TEST-03

Figure 20 is an example of a possible selection made by a user. In this case, the user have chosen the Tester "TEST-03" and the results (that in this report are fictitious) represent in some way the general performance of that machine, for a specific product previously selected in the first window.

"Shift/Day" Selection (7): there are three different fields in this selection menu: *Fiscal week*, *Day* and *Shift*. The value of "Fiscal week" field is always the same as the one that was selected before, being impossible to change it in this table, is "disable". Then, the user can make an analysis with more detail, selecting the day in that week and/or a shift[12]. The selections (day and shift), influence the results that the system will return, once their function are to filter the type of information that is wanted.

"UPH" Indicator (8): the values represented in these fields correspond to the Machine's capacity for a specific filter chosen before. There are three different types of values: the "Actual UPH", the "Planned UPH" (given by the *Capacity Plan_Test.xls* and the "Deviation" (difference in percentage between those fields). In the cases where this deviation is lower than -20%, an alert appears, being the value displayed with a red colour, like in the picture above. Notice that if the product is being processed for different customers (hence, different speeds),

[12] The production scheduled is divided in four shifts of twelve hours each. Therefore, in one day (24 hours of production) there are two shifts (from 7am to 7 pm – daily shift – and from 7 pm to 7 am – nocturne shift)

there will be a distinction between their values, represented by a different value in the "Speed" field in order to distinguish those cases.

"Tester Utilization" Indicator (9): The total time is divided in three major parts: the *Process Time*, *Overhead* and *Time Between lots* (the last one is not represented in the program). There are also a field that illustrates the difference between the actual values and the target specified by the Production Engineer.

"Average TBL" Indicator (10): This indicator is divided in four different column fields: "Average", "Target", "Deviation" and "Average Lot Size" that can explain a good or low performance of this indicator as they are correlated. Usually the lower the Lot size, the higher will be the value of the Time between Lots.

"Yield" Indicator (11): this KPI aims to measure the efficiency of the product selected. It measures the number of units "pass"[13] per volume tested. In other words, it calculates the final yield (i.e. including re-measurement[14]) for each products and measurements.

"Home" button (12): this button has the functionality to, when pressed, return to the Main Window. It is nothing more than a "linkage object".

As an example, Figure 20 represents a case where the tester "TEST-03" have been selected and the conclusions are clear. The Tester is having low capacity (UPH), there is an average TBL higher than expected and the machine's Yield in the Test Operation "B" is lower than pretended target value (set as 90%). For that reasons, a more effective analysis shall be done in order to find out the reasons that support such performance. Therefore, the next two windows contain a group of charts that illustrate the answers for that question.

C) UPH Indicators Window

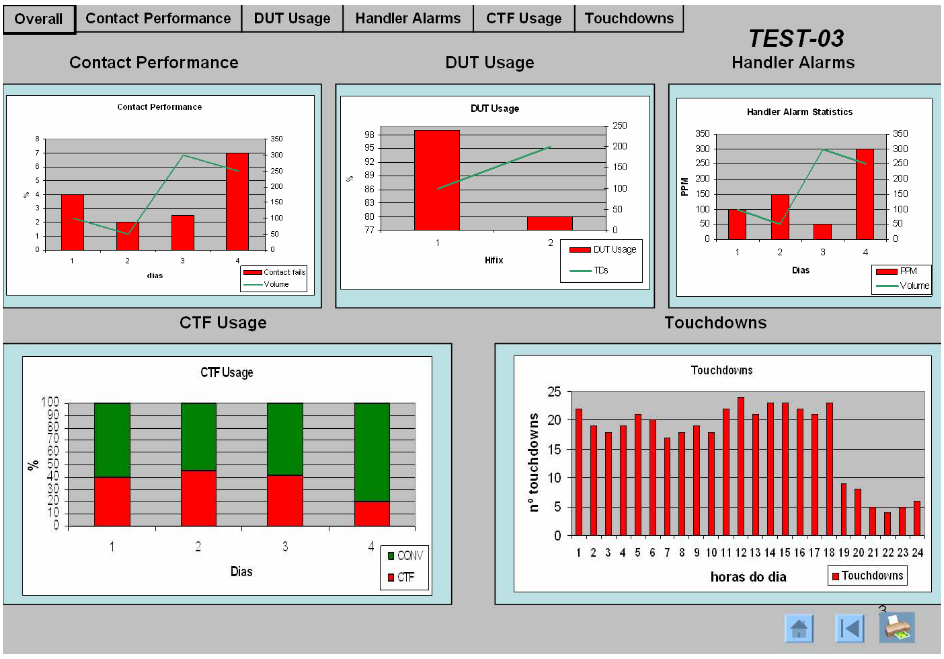

Figure 21 – UPH Indicators window

[13] After the test, the components are classified into "Pass" or "Fail" category, according to its performance when submitted to the tests. The yield measures the percentage of units considered as "Pass" in order to measure the efficiency of that machine and/or components.

[14] Re-measurement occurs when a product or group of products are not considered as good for sale ("Fails"), due to the programs' marginality, and returned to be processed in order to try to recover some of them.

As it was referred before, this window appears when one button of the UPH indicator in the *General Performance Window* is pressed.

The window represented contains many tabs, the first one with the overall performance, i.e., with all the five KPI's and than one tab for each KPI individually.

The need of a tab per KPI exists for a better comprehension of each chart, once has a bigger size and has some particular options included.

a) **Contact Performance:** In a generic way, it represents the number of Contact Fails per Volume Tested (%). The bad performance of a socket, usually manifests it self on devices failing contact tests. The socket performance can then be monitored by monitoring the % of fails on contact tests.

b) **DUT Usage:** The DUT usage report is used to monitor the utilization of the DUT's available in the testers. Ideally the DUT usage would be always 100%.

c) **Handler Alarm Statistics:** The aim of this indicator is to monitor the number of Jam's and Errors[15] that a machine is submitted.

d) **CTF Usage:** Measures the % of volume (kpcs) done with CTF and conventional flow. With this indicator the % of hardware that's being tested in CTF and Conventional mode can be followed.

e) **Touchdowns:** this indicator measures the number of touchdowns performed every 60 minutes.

An example of a possible analysis of these five indicators is not being given once it involves several technical concepts that are not the object of this report.

D) "Tester Utilization" Indicators Window

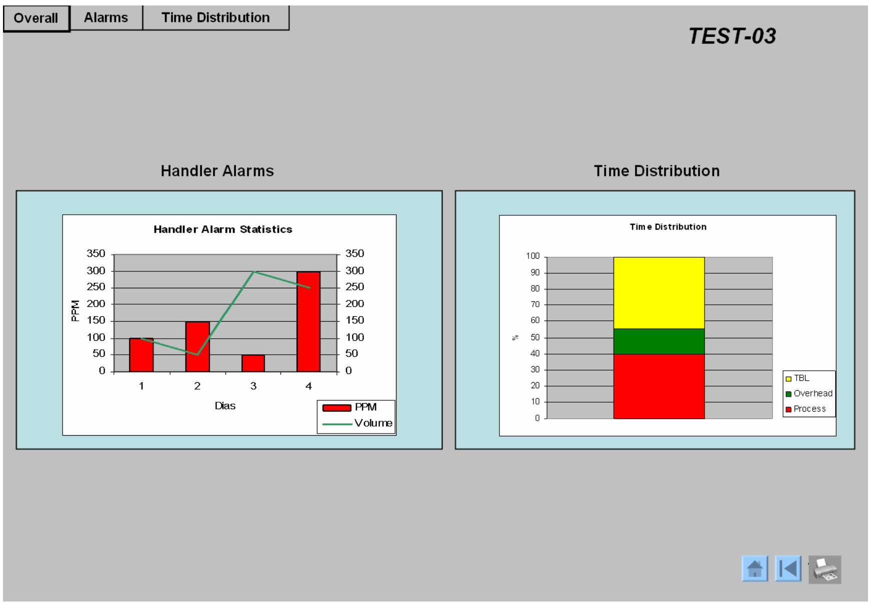

Figure 22 - Tester Utilization Indictors

[15] *Jam's* and *Errors* are frequent occurrences in these machines. Jam's occur more frequently and has easy and fast resolution. On the other hand, the Error's are not so frequent and in general take more time to detect and solve the problem. Errors are related with the motor's movement and Jam's with the trays used to transport the memories.[14]

Like was explained before, this window appears when one button of the *Tester Utilization indicator* in the *General Performance Window* is pressed. It contains three tabs, two with different KPI's and the first one with the overall performance, i.e., with both KPI's.

The need of a tab per KPI exists for a better comprehension of each indicator chart, once has a bigger size and has some particular options included.

a) **Handler Alarm Statistics:** The aim of this indicator is to monitor the number of Jam's and Errors that a machine is submitted. This chart is the same as the one explained previously in the UPH Indicators, but in this case is used to justify the values in Tester Utilization indicator.

b) **Time Distribution:** it shows how the test time is being used. Based on lot by lot data, the time between lots, the overhead and the process time can be calculated. In this case, the process time stands for effective test time, time where there are devices connected to the *Hifix* and the tester is performing measurements. The overhead is the time between the lot starts and lot ends, where the process time was removed. In other words, the overhead (or index time) is time spent in the trays[16] movement while they are not being tested. The time between lots, as the name implies, is the time between the end of a lot and the start of the next one.

[16] Trays are boards where the memories are carried.

4 *QimTa's* Prototype

The prototype was the last part of the project. "A prototype is a working model created to demonstrate crucial aspects of a program without creating a fully detailed program". [9] It was pretended to be a model that would describe the main functionalities of the tool, and also try to detect some inefficiencies that should be adjusted with the purpose of improving the version that will be implemented, increasing its flexibility and accuracy.

On the other hand, the prototype is useful to replace the original tool (future program) while is being developed in such a way that allows the detection of some errors. Besides, with the prototype an overview of the Test Area performance can be performed by now, alerting for some problems that are occurring while the other tool is not implemented. This will be an intermediary step while the program is not created.

However, it is important to keep in mind that the prototype is nothing more than a model, and for that reason it doesn't have the efficiency required in the original program. Because the full program was not possible to be developed for several reasons such as lack of time, permissions and programming skills, the prototype will represent the *Main Menu Window* and the *General Performance Window* (referred in the chapter 3), leaving the other two menus (*UPH Indicators* and *Tester Utilization Indictors*) and their functionalities for the original program.

In this chapter a general description will be made. In the Appendix J, a conceptual model is represented to give an idea of the system.

4.1 Prototype Description

The prototype created was developed with the *MS Access*, practical and user-friendly software which one of the main advantages is the possibility to access the Test Data Base in a very simple way. Moreover, it allows the creation of interfaces (forms) that are critical in the program's development.

The prototype's development implied the creation of tables with the data that would feed the program; this was the first challenge in this stage. The problem existed because there was a need to mach the information between the Test Database and the Production Plan (*Capacity Plan_Test.xls*), a difficult task once the products granularity was different.

As a consequence of this fact, there was a need for the creation of a transition table that in some way could make the relation between both "worlds" without loosing information.

Another challenge was the need for the creation of tables and queries with the restrictions and configurations associated, that represent the Model Parameterization referred before in the

Use Case Model. This part will be created as a Configuration window in the original program, according to the specification.

The hardest part was the creation of several *queries*[17]. There were created *queries* for each feature/functionality and the forms associated. Nevertheless, for most of all the features, there were created three different *queries* and sub-forms that correspond to the filter coupled. In other words, the user has the possibility to filter the results by “Package”, “Product type” or “Configuration”, translating the need for a *query* in each case. To have an idea of the prototype’s complexity, there were created sixty *queries*, twenty-eight forms and seven tables.

Adding to this, the prototype was feed by two tables from the Test Databases that are crucial for the monitoring of the production performance.

Like was pretended, the prototype was developed having in consideration the Specifications that had already been created, not forgetting that not all the features were possible to model in the prototype.

Bellow, there will be described the prototype developed during the project.

Main Menu Window

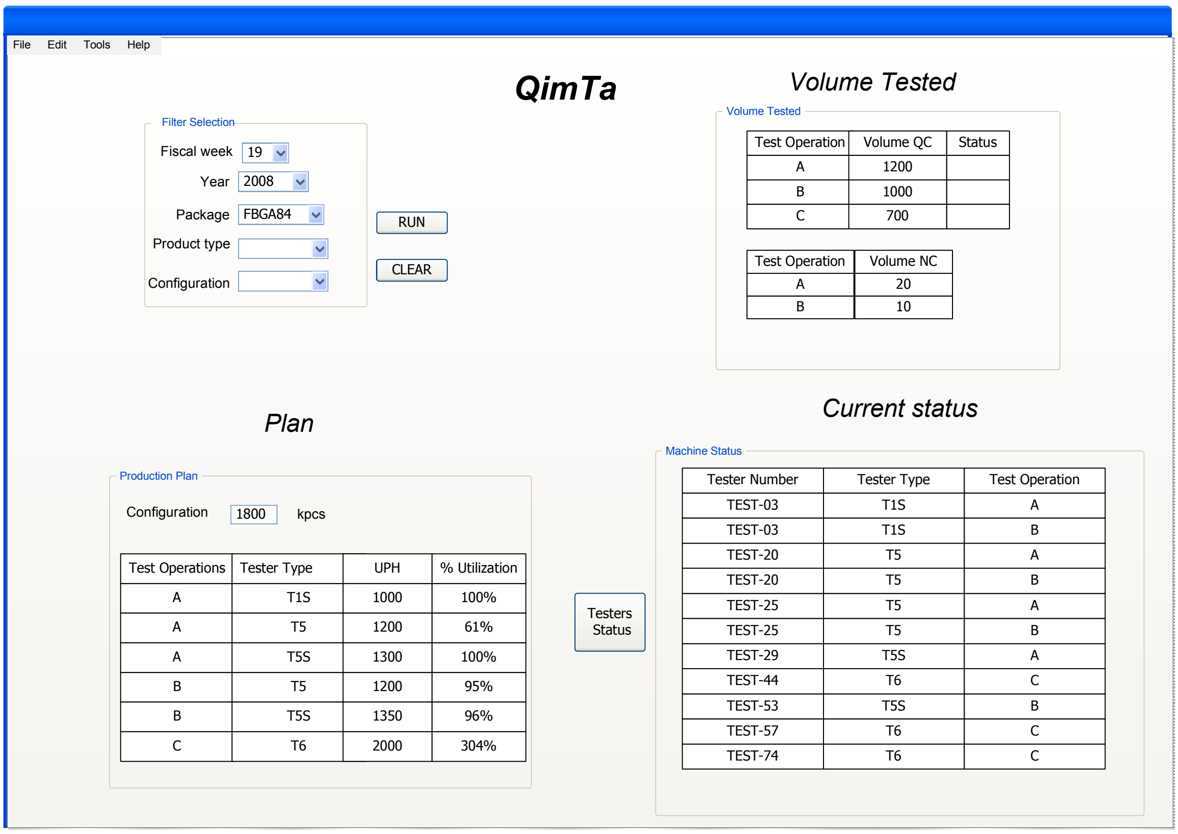

Figure 23 - Main Menu Window (Prototype's version)

[17] *Queries* are features in the MS Access software used to filter and analyse the data form tables in a more flexible way.

Figure 23 is an example of the window displayed when the user selects a specific product. In this case, and to exemplify how the system works, it was decided to filter only the Fiscal week; Year and Package type (in this case the Fiscal week 19, year 2008 and the Package type FBGA84). The results returned by the program are displayed in the picture above and some analysis can be made. First of all, the user can see what the weekly plan is for the referred week. As one can see it is planed to be produced 1800 kpcs of that specific Package until the end of the week 19.

The next analysis to be made is trying to understand the status in terms of volume produced until the moment. Therefore, and looking for the "Volume Tested" one can see that in each operation it was produced less 70 % of the expected volume. Once the analysis was made in the last part of the week (there was only one day to produce the quantity left) is important to understand why the volume produced is beneath the plan. Another analysis possible to make is that there were produced some material *NC* that wasted some productive time of the machine but it will not be considered as volume out in the production. Moreover, this can explain in part why the volume tested is not achieving the target until the moment.

Now is time to understand other reasons behind such performance. As a first approach, one can try to find if the number of machines processing that package type are less than planned. Looking to the left side of the window, in the "Plan" part of the Menu, one can see that for the core operations it shall be used 5 machines[18], and for the Speed operations (represented as Test Operation C) there shall be allocated 3 machines during the week. Now looking to the right side of the window (in the "Current Testers" fields) it is possible to observe that the number of machines processing that products correspond to the plan. Thus, the machines "TEST-03", "TEST-20", "TEST-25", "TEST-29" and "TEST-53" are performing the core operations and the other three ("TEST-44", "TEST-57" and "TEST-74") are processing the Speed Operations. With this analysis it was intended to see if the machines used were less than planned.

Until this point, the problem that is being faced is the volume tested under the planned value. Therefore, there is a need for a more effective analysis to understand the reasons that explain a lower performance than expected. Remember that the machines used in core operations can perform both A and B operations, that is the reason why there are three machines that are showed twice, each of which representing a different operation.

The main disadvantage of this machine is that in this menu (more precisely in the "Current Testers" fields) there is not implemented the functionality of an alert for the machines with lower performance. However, this functionality will be implemented in the final program, once it is seen as one of the main advantages of this tool that can save much time wasted.

The next step of the analysis is the evaluation of the Testers' performance, in order to find the root causes of such deviation from the plan, the aim of this program. There is a lack of volume produced that probably is related with the machine capacity; the objective of this monitoring tool will be to find the causes that are influencing the capacity, identifying the low performers, acting as fast as possible in a way of increasing the process efficiency.

This analysis is done in the General Performance Window explained bellow. The user just has to press the button "Testers Status" and a new window will be open.

---

[18] Notice that 100%of capacity utilization represents 1 machine totally dedicated during the whole week. If the capacity utilization is higher than 100% it means that there is a need for more than one machine.

General Performance Window

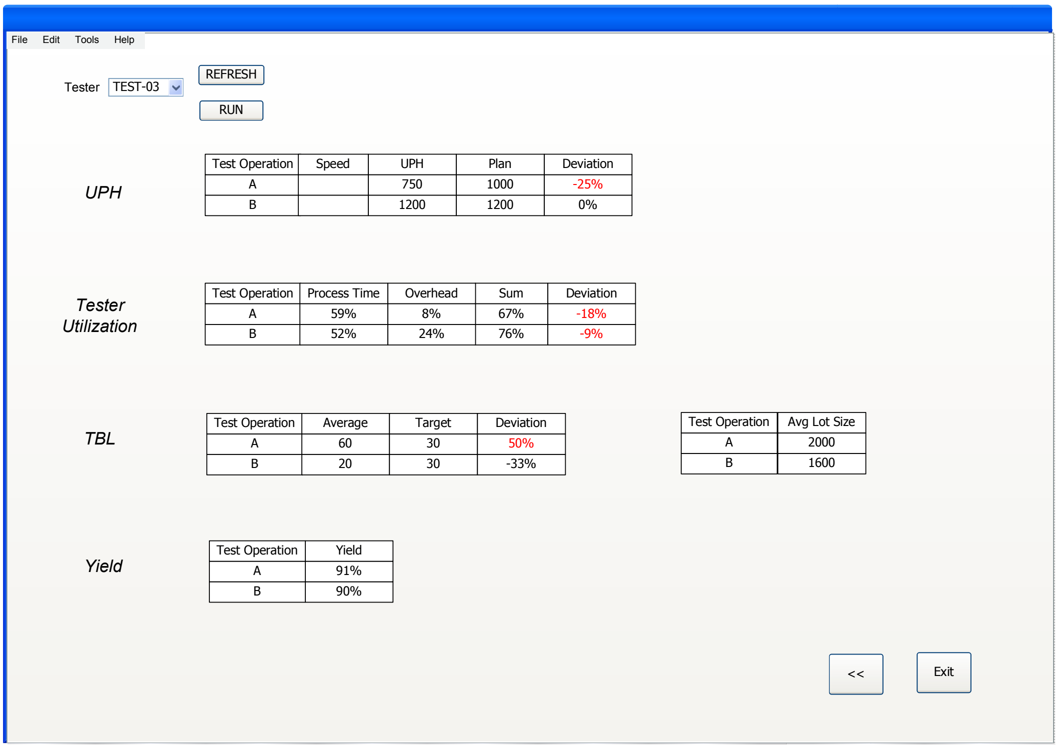

Figure 24 - General performance window (Prototype's version)

After the window been open, the first step is to press the “Refresh” button in order to update the “Tester” field with the “Current Testers” from the previous window. This action is important to avoid that for some reason the Testers loaded in that field correspond to a previous product that was analyse in an earlier analysis.

Following, the user just needs to choose the Tester he wants to analyse press the “Run” button and the program will return five main KPI’s related with the “Package type” and Tester earlier selected, that are the parameters that will filter the results. The window is similar to the one that will be displayed in the program to be developed and it gives powerful information as well.

As an example, the Tester “TEST-03” was chosen and the results are represented in the previous picture. In a first evaluation, this Tester is with low performance as the number of “red alerts” demonstrates. First, the UPH in the Test Operation A has a deviation of -25% from the target value, i.e., the machine is having a capacity twenty-five times percent less than expected; in this case, these machines are processing only 750 units per hour, in average, instead of the 1000 units that were planned. This means that the machines are testing in average around 250 units less. Therefore, a deeper analysis shall be done in order to identify the causes of such deviation. In this case, the user could use the program “Backend Framework” (for example) and analyse some important KPI’s related with the UPH.

Like was referred in the Chapter 3, the future program will have this functionality, returning some important KPI's, making the analysis more quickly and efficiently.

Concerning the *Tester Utilization indicator*, those fields display the "Deviation" value with a red colour illustrating the occurrence of some problems once the machine is having a sum of process time and overhead lower than expected. Like in the case of the UPH indicator a deeper analysis will be required using other tools with functionalities that will be included in the future program.

The last indicator that seems out of control is the "TBL" (Time between Lots) that in the case of the Tester Operation A is two times higher than the target value. This value can be explained, for example, for the number of setups ant the time required for them, the low size of the Lots, among other factors that shall be controlled.

4.2 Prototype's Inefficiencies

Like was referred before, the prototype was developed to get an overview of how the program shall be in a generic way. There will be pointed out some of the main inefficiencies comparing to the future program, not forgetting the fact that the prototype was thought since the beginning as a representation of just a part of the future program.

The inefficiencies are:

- The prototype only contains the two main menus (Main Menu Window and General Performance Window), not performing the graphics related to the UPH and Tester Utilization KPI's.
- The prototype takes more time to return the results (is slower than the original one, based on an estimative and in the fact that the queries that support the program delay the results).
- The user needs to choose which tester wants to analyse (in the real program there will be a link for the general performance menu and the calculations of the main KPI's will be calculated automatically).
- The data from the Production Plan has to be introduced manually, i.e., there is a need to insert a table with updated information of the production plan once a week. In the future program there will be a function that automatically will upload the results to the program's database (in this way doesn't waste the time spent for the upload).

All this inefficiencies make the prototype user-dependent and not as flexible as is desired in the original program. Nevertheless, it is considered as a very useful instrument for the developers understand the main objectives and concerns of the program and even to study the benefits of the new program will be referred later on this document (Chapter 6).

5 Implementation version – *Production Monitoring* tool

Until this point a first model was created based on the previous stages of the project in order to fulfil the user needs and try to make a relation between the databases that exist. However, some changes occurred once the proposed solution needed a considerable effort to be developed. Consequently, after some brainstorming sessions and negotiations with the IT Department[19] a different solution came up and hence the need of some modifications in the solution proposed. In the end, the main difference was related with the environment where the program shall be involved and the interfaces where the results were returned.

Notice that the first model and its specifications still remain in order to create the program in the future.

In the Appendix G there are more explanations about this version, including the Conceptual model that enhances the differences between this solution and the proposed one.

5.1 Program Description

As explained before, the main difference is related with the environment where the program will be developed. As a way of saving costs and make the development in an easier and faster way, the program will be defined in a first phase as a new functionality of an existing program called "Backend Framework – Test Area Key Performance Indicators (TAKPI)"

TAKPI is used in the Test area as a tool to analyse different KPI's related with different matters such as "Tooling", "Equipment", "Product", "Test Time" and "Production". In the case of the new tool, it will be included in the functionalities related to the "Production" KPI's once its KPI's are mainly related with the production performance.

Again, the decision to implement this solution was based in some facts that make this option the more logical one in a near future. The advantages of this type of implementation will be enumerated later in this chapter.

An illustration of the future program will be described.

[19] IT Department – is the Information Technology department, responsible for the development and maintenance of all the programs and software within the company. Therefore, it has the last decision what concerns the creation of new programs including the one created in this Project.

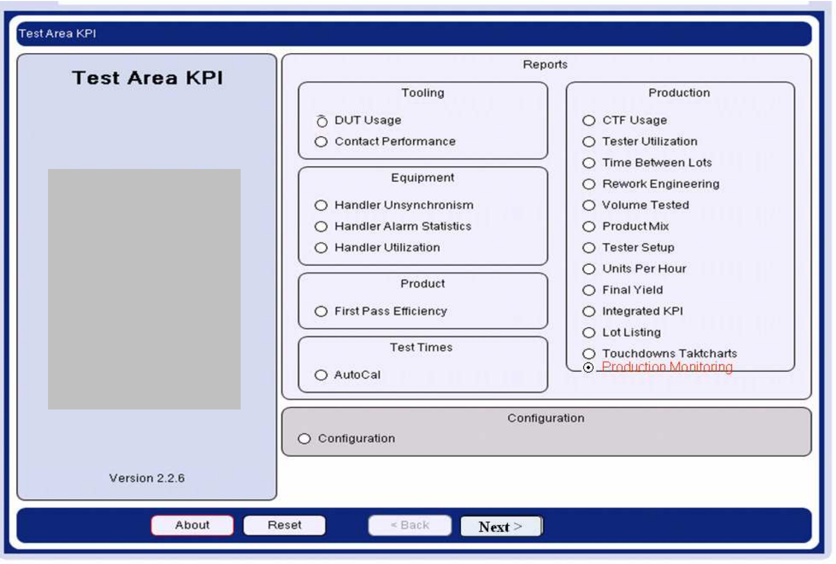

Figure 25 - New TAKPI Interface

The TAKPI interface will be changed, including another option in the Production reports named *Production Monitoring*, as the picture illustrates with a red colour.
When the user selects this option and presses the "Next" button, a new window appears as displayed in the following picture.

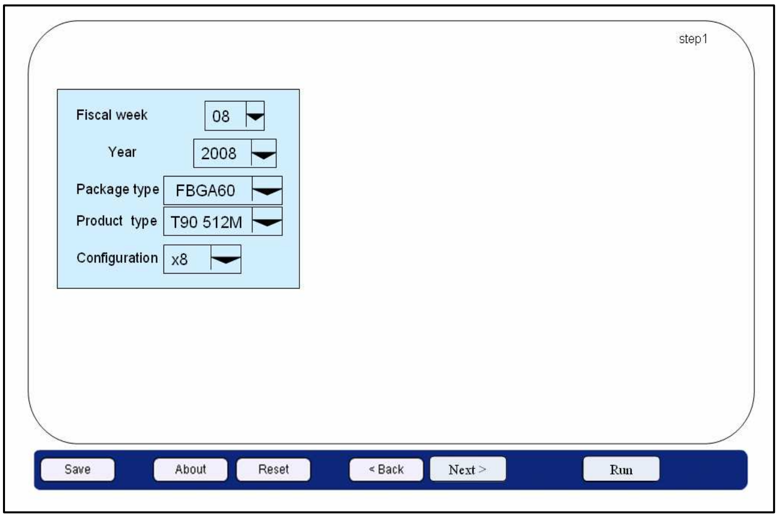

Figure 26 - Example of the first step window (*Production Monitoring version*)

Then, it has two options, or presses the button on the right ("Run" button) and the program generates a report with the relevant information about the filter selected (Figure 27) or if the button "Next" is pressed a new window appears and the user has the possibility to choose the Testers that he wants to evaluate (Figure 28).

The more natural approach is to first generate the report and, after analysing the report (and based on the Testers that are with a lack of performance), the user selects the pretended Testers like in the Figure 28.

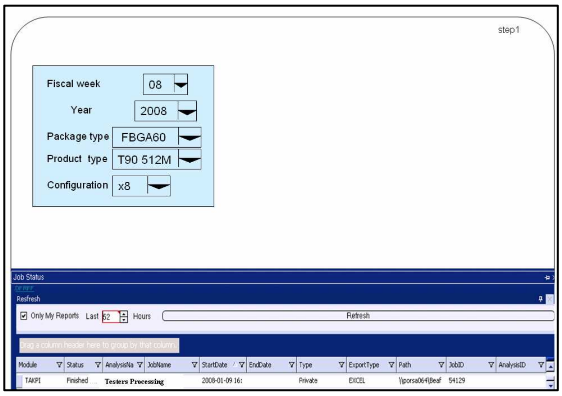

Figure 27 - Example of the window when the button "Run" is pressed

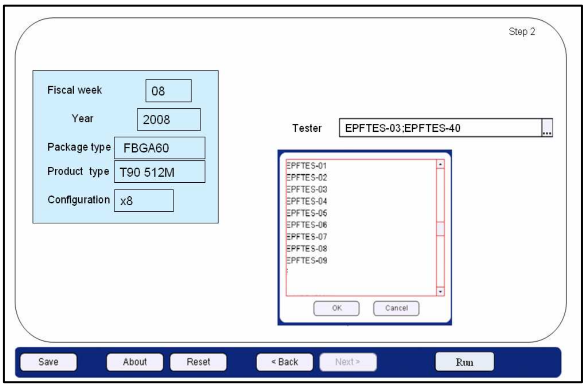

Figure 28 - Example of the window when the button "Next" is pressed

After the Tester has been selected, the program will generate another *MS Excel* report as the Figure 30 exemplifies. The menu/feature is displayed in the bottom of the window with the indication of the report that had been generated.

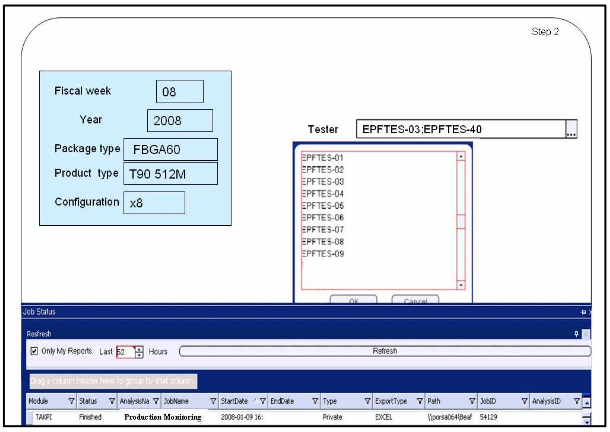

Figure 29 - Example of the feature showing the creation of a "*Production Monitoring*" report

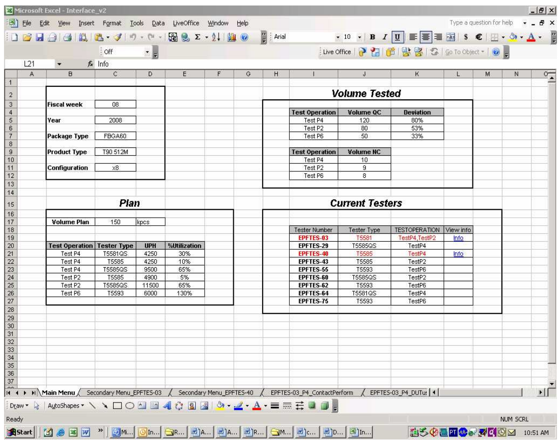

Figure 30 - Example of the Main Menu tab displaying the *MS Excel* report created

Selecting the report in the previous window (Figure 29), with the name *Production Monitoring*, an *MS Excel* file shows up with all the relevant information like in the Main Menu Window of the Proposed Solution (see Figure 30). This also have the alerts of the machines with lower performance and a link to the General Menu Window for the Testers selected as it is illustrated in Figure 31.

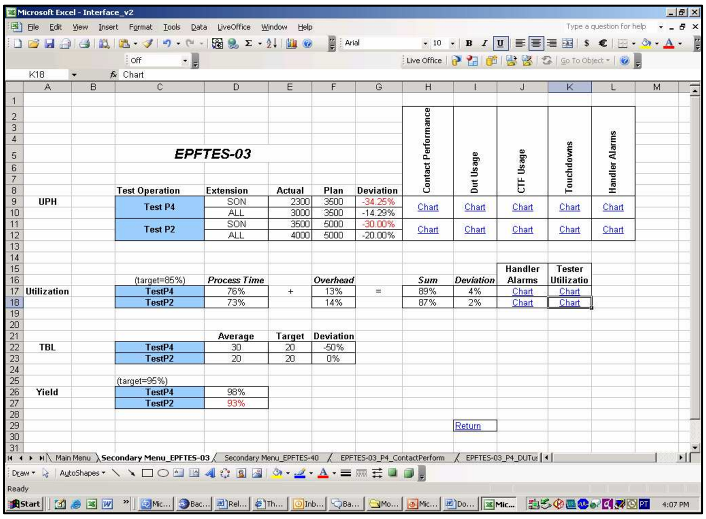

Figure 31 - General Performance window – *MS Excel* report (*Production Monitoring version*)

Through this report, it is possible to evaluate the Testers' performance in a generic way. However, if the user wants to go further in his evaluation, it has the possibility to analyse several charts like in the *QimTa* program. In this case, there are links to each chart that are contained in a specific tab/sheet in the *MS Excel* file.

In this way, this solution will be similar to the proposed one, having all the functionalities as the other, allowing a rigorous and efficient analysis of the situation in a flexible and simple way.

Finally, it was possible to design a new model, cheaper and with less work required according to the user needs/requirements.

In the next section, there will be pointed out the advantages and disadvantages of this implementation version (comparing to the proposed one) to make a parallelism between the two solutions discussed.

5.2 The Implementation version: Advantages and Disadvantages

As previously mentioned, this version came up as a more viable solution for several reasons. Unfortunately, the required modifications have also some disadvantages comparing to the proposed solution. The following topics point out the key advantages and disadvantages that are associated with the changes.

Advantages of the implementation version:

- Require less human effort to develop the tool;
- Required less time and associated costs to develop the tool;
- Uses most of all the specifications that already exist in the TAKPI functionalities;
- Required less time to learn how to work with the program once it is similar to the one that already exists;
- All the KPI's generated by the program already exist in the TAKPI, despite of the filters and parameterization being different, generating different results.

- Is more easily accepted by the other Qimonda sites, once it can be easily implemented and probably used by them in a near future.
- Related with this fact, it has the advantage of becoming the priority program to be created in the Test Area.

Disadvantages of the implementation version:

- It is not so user-friend as the other;
- It needs two different software tools to support it (the Backend Framework and the *MS Excel* to generate the reports).
- It doesn't have the option to display all the charts in an overall tab, one of the main advantages of the proposed solution;
- For users that are not used to the Backend Framework, this program is not so intuitive.
- It doesn't give the opportunity to configure the charts

6 Business Case – qualitative and quantitative advantages of this project

Now that the project was described the key benefits that this new program brings to the Test Area will be explained. In other words, in this section the reason behind this project and why it is so important for Qimonda will be described.

This chapter will be divided in three parts: the methodology used to estimate the quantitative benefits, then the quantitative benefits and last but not least, what are the qualitative benefits that *QimTa* tool will bring to the Test Area.

6.1 Quantitative benefits: The methodology used

This section intends to explain the methodology followed to estimate the benefits that *QimTa* will bring to the Test Area. This was not a simple task because the most of the advantages were not measurable, being most of them qualitative. However, it was decided to follow a framework that was at the same time consistent and with accurate results. It was a concern since the beginning to provide truthful information and for that reason the methodology needed to have a logical sense.

As the aim was to compare the new monitoring process with the actual one, some experiments were performed simulating real situations and comparing the two different methodologies, one with the actual process and tools associated (see Section 2.3))and another with the proposed tool and the new monitoring process. The new process follows the steps describe in Figure 32, enhancing the reduction of steps when comparing with the actual one.

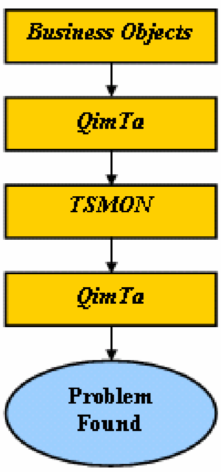

Figure 32 - Monitoring process roadmap (proposed situation)

Despite of the reduction of the number of steps, the logic remains the same, thanks to the versatility of the *QimTa* program that contain many functionalities (provided by several steps in the actual process), in only one program. The only steps that remain the same as actually are the first and the third from Figure 32 (functionalities not provided by the *QimTa* program). The other steps are all included in *QimTa* as it will be explained in section 6.3).

About the evaluation process, the methodology followed was:

1) Perform a monitoring process like is done actually and register the total time spent in that process; (Figure 14) The time then has to be estimated for one *Tester*, if the evaluation was made for more than one, then it has to be transformed into an estimative for a single machine.
2) Perform a monitoring process with the new tool and register the total time spent in that process; as in the previous step, the time has to be estimated for a single *Tester*.
3) Register the difference of time spent between the two methodologies, that is, the time saved by using the new methodology.
4) Register the difference of capacity, i.e., the average deviation of the actual *UPH* values when comparing to the planned ones.
5) After the experiments have finished is important to get a consistent value for the average time saving and for the average deviation of capacity.

The potential savings in terms of additional processed units (i.e., the increase of capacity) and the savings on time spent in the monitoring are the key benefits related with the implementation of the new program.

The Prototype was a fundamental tool to estimate the time spent in the future monitoring process. Despite of its inefficiencies, it was very useful to calculate the time spent in the analysis of the situation and in KPI's provided by the program. The analysis of the KPI's that are not included in the prototype it was done in the *Backend Framework* program.

In the next section, the results of the experiment will be described, as well as the estimated savings with this project.

6.2 Quantitative benefits: Results

After some evaluations of the time saving with the new solution, some conclusions were taken in terms of capacity gained, time saving and additional memories processed. These three factors were related, once the reduction of the monitoring time allows the faster detection of causes of variation, increasing in this way the capacity and hence increasing of productivity.

The time saving in the monitoring process was considerable. As the Figure 32 enhances, the number of steps to find the root causes of variations decreases, decreasing the time spent in this activity.

Table 2 - Difference of time spent (comparison between the two methodologies)

Time Saved	Experiments					Average
	1	2	3	4	5	
Seconds	2049	2204	2015	2165	2101	2089
Minutes	**34**	**37**	**34**	**36**	**35**	**35**

As Table 2 illustrates, the estimated value for the time saved is around 35 minutes/Tester, reducing the time spent in more than 75%, when comparing to the actual process.

Table 3 demonstrates the results after five experiments for the average capacity variation in machines with problems. As estimation, in average, the machines with low performance process 43% less than its planned capacity; therefore, the faster the detection of problems, the less time the machine will work with low capacity.

Table 3 - Average deviation (Comparison between the actual and planned value of *UPH*)

	Experiments					
***UPH* Deviation**	**1**	**2**	**3**	**4**	**5**	**Average**
	45%	35%	40%	45%	50%	43%

As a final result of the Business Case, Table 4 illustrates the values obtained for the additional capacity that results from the utilization of the new solution.

Table 4 - Weekly and annual values for the additional capacity in terms of units processed (for 5 and 14 occurrences)

Number of occurences/week	Addicional Capacity units/week	Addicional Capacity units/year
5 (minimum)	>2.500 units/week	>130.000 units/week
14 (average)	>7.000 units/week	>364.000 units/week

The new tool will allow an estimated increase of capacity. It is estimated to process more 2.500 units/week (considering 5 occurrences per week) and 7.000 units/week (considering the average of 14 occurrences per week). Therefore, in one year (52 weeks), the possible capacity gains are more than 130.000 units/year and 364.000 units/year, respectively. This achievement is due to the reduction of time in the detection of some low performers alerted by the program.

Last but not least, having as a reference the potential increase of capacity and the reference price of €1/memory unit, the annual increase in the sales value will be around €130.000/year (in the worst case) and €364.000 /year (in average).

Having in account that the cost of implementing this solution is around €5.000, the total benefits are considerable so this tool must be developed.

The results and the methodology used in the Business Case can be seen in more detail in the Appendix H.

6.3 Qualitative benefits

Like was referred before, the qualitative benefits were more simple to define. Bellow there is a list of aspects that match the issues described in the Section 2.4)

- Improve problem's visibility and alert
- Provide a better and faster analysis of the situation, preventing unnecessary time-wasted.
- Reduce the number of programs used to control the area (Figure 33)
- It gives the information about all the important KPI's at the same time instead of the need to select one by one.
- It allows the definition of a more effective roadmap to monitor the Test Area, through the reduction of the number of programs needed to control it. (Figure 33)
- It gives the alert in cases of high deviations between the Production Plan and the actual production status by giving information for the existence of some problems, finding the causes that will allow reacting more quickly to the situation. In this way, it will increase the capacity and hence the productivity in Test Area.
- It gives the information about the machines' spare capacity, through the indication of the planned capacity utilization in each machine.

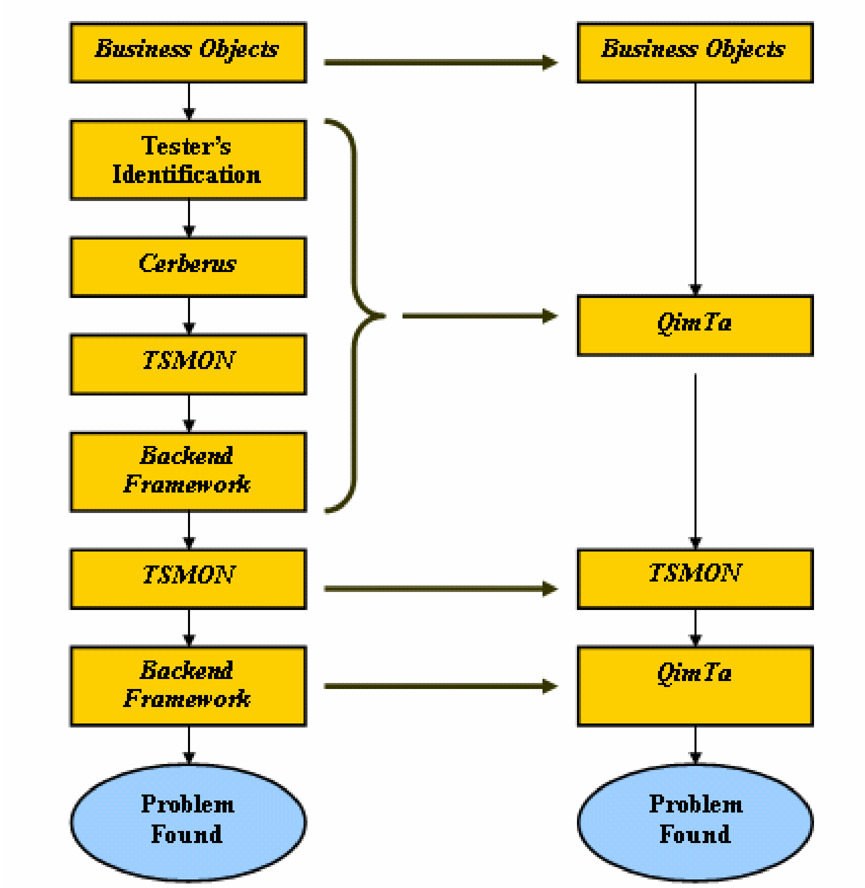

Figure 33 - Scheme illustrating the reduction of programs and steps needed for monitoring the production

7 Conclusions and Future Improvements

The aim of this project was to design a tool to monitor and control the Test Area of a semiconductor company, namely QPT. The designed tool will enable the controller to have an improved visibility of the problems, alert for special occurrences, and hence help increasing the testing capacity.

At the end of this project, all the specifications for the future tool were created through a methodology that involved the future users of the program. In addition, a fully working prototype was created to support the users and developers, and replace the final program, while this is being developed by the IT department.

A business case was performed to analyse the potential benefits that the implementation of this new tool and proposed change in the monitoring and control process can bring to the area. The major benefits are related with time saving in the process evaluation, the increase of capacity and the related increase in productivity.

The potential benefits carried out by this program were estimated. It was concluded that the new tool will increase the time spent in the monitoring process in more than 75%, leading to potential capacity increases in about 364.000 units/year and the related gains of about €364.000/year, in average.

The estimation will be calculated again, after the tool is implemented, in a way to obtain an even better approach of capacity gains with the new solution.

Indeed, the potential advantages of this tool reflect the need for improvements in the Production Control field within the company. It is suggested that this type of improvements be carried out in the other productive areas within the company as a way to improve the general production in QPT.

The project had several challenges. First of all, Qimonda plays in a fast industry and so words like “flexibility” and “dynamic” were mandatory. In this way there is an increasing level of complexity and need for continuously updating information. The second challenge was the program specification. The high complexity and the need to make a relation between the granularity from Test DB and the Production plan file; increase the difficulty of the project. Moreover, all this complexity was translated to the prototype. The third aspect is the several technical concepts related with this type of industry. For that reason, there was the need to study all the process and machines’ concepts in order to understand the whole area and design a program based on its inefficiencies. Last but not least, the business case performed was not easy to define. The fact that this tool is for monitoring and not for solving the problems, and is not already developed, carried some difficulties to define its potential benefits.

As future improvements the author suggests the implementation of a functionality that updates the UPH values in the production plan according to the specifications defined in this report, making the plan the more accurate as possible, and reduce the weekly time wasted in this activity.

Another proposal is to improve the production plan file, taking into account some suggestions that are mentioned in this report, in a way to reduce the user-dependency. Having as a starting point the conditions defined, it is possible to develop an algorithm to make the production plan in a faster way, reducing the possibility of human errors. This production plan could be modified allowing some flexibility in the decision making.

Finally, and in the context of *Production Control* where this project is inserted, it is suggested to create an auto-adjust capacity model for Test Area, to perform an analysis of the machines in more detail, adjusting the planned capacity with the current status of machines.

The final test monitoring tool solution proposed here will be reached after a two phase's implementation process. First, it will be partially developed as an additional functionality of an existing program (*Backend Framework - TAKPI*), for reasons such as lack of time, easiness of implementation and cost reduction. In a second phase, the proposed solution (*QimTa*) will be implemented as fully specified in this report.

Until the end of this report there was no defined date to start the implementation of the proposed solution, however it has already been accepted by the IT department and the forecast for its development is about two months. In the future it is indeed to be implemented in all Qimonda's Back-end sites.

8 References and Bibliography

[1] Wikipedia. “Qimonda”. October 2007. <http://en.wikipedia.org/wiki/Qimonda>. Accessed 2007 October 10.

[2] Qimonda AG. Core Values. <http://intranet.qimonda.com/Downloads/CF/C/Comp_Pres/375, 5, Our Core Values>. Accessed 2007 October 10.

[3] Google search for “Use Case Diagram”. October 2007. <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>. Accessed 2007 October 17.

[4] Wikipedia. “Key performance indicators”. October 2007. <http://en.wikipedia.org/wiki/Key_performance_indicators>. Accessed 2007 October 15.

[5] Project Auditors. “Definition of User Requirements”. October 2007. <http://www.projectauditors.com/Dictionary/U.html>. Accessed 2007 October 17.

[6] Wikipedia. “Use Case”. October 2007. <http://en.wikipedia.org/wiki/Use_case>. Accessed 2007 October 17.

[7] Wikipedia. “Use Case Diagram”. October 2007. <http://en.wikipedia.org/wiki/Use_case_diagram>. Accessed 2007 October 17.

[8] Google search for “Definition of Conceptual Model”. November 2007. <http://www.google.com/search?hl=en&q=define%3Agranularity&lr=lang_pt>. Accessed 2007 November 16.

[9] Google search for “Definition of Prototype”. January 2008. <http://www.google.com/search?hl=en&defl=en&q=define:prototype&sa=X&oi=glossary_definition&ct=title>. Accessed 2008 January 8.

[10] Qimonda Portugal. November 2004. “Process Description”. Components Test.

[11] Qimonda Portugal December 2006.“Logística da área de teste”. Process Instructions.

[12] Qimonda Portugal. March 2001. “Tecnologia de Teste”, Process Instructions

[13] Qimonda Portugal. July 2007. “Procedimentos de Operação na área de Teste”, Process Instructions.

[14] Qimonda, Portugal January 2005. “Preventive Maintenance Manual on Test components Handlers”. Maintenance Instructions.

[15] The Taxonomy guide. “Granularity”. November 2007. <http://plc.fis.utoronto.ca/tgdemo/Glossary.asp >. Accessed 2007 November 16.

[16] Larry L. Constantine & Lucy A. D. Lockwood (2000), “Structure and Style in Use Cases for User Interface Design”, Object Modeling and User Interface Design.

[17] Cunha, João. May 2001. “Sistemas de Informação” – Modelação do Conhecimento e Bases de Dados".

[18] Ramos, Nogueira. “Desenhar Bases de Dados com UML”. 2006. Edições Sílabo.

[19] Wikipedia. “Software prototyping”. January 2008. <http://en.wikipedia.org/wiki/Software_prototyping>. Accessed 2008 January 8.

[20] Neves J.; Gomes L.; Correia M. . “Fundamentos do Access 2003 – 2ªEdição”. Edições FCA.

APPENDIX A: Test Area

Test area is responsible to perform the electrical tests into the memories previously produced. The results of the tests are used to continuously monitor the Back-end and Front-end processes in a way of improving their yield. Therefore, it is necessary to categorize the root causes of such fails to get better results in process performance.

In the Qimonda Portugal case, it is essential to characterize the Back-end process fails, in such a way that may be possible to improve the process and implement some correction and monitoring measures for the fails. In Qimonda PT the Front-end fails are also verified once they are the main causes of *yield losses*.

Another relevant issue of the *Test* is to verify and measure the level of performance and reliability of the components to avoid possible fails in the customer possession. In cases where those levels are bellow the parameters set by the Quality Department, the components suffer a *Downgrade*[20] and are sold to clients with lower requirement standards.

The main process flow includes tests at low temperatures and high temperatures, also called Test Operation A and Test Operation B, respectively. After the two steps, some components have to flow to another step, named Speed Test (according to their specifications).

Test Area is the area with worse yield, once it is the local where more tests are performed and where the main part of the fails are detected and separated. The fails can be characterized as: contact, functional, parametric or speed fails. This characterization is made by the Tester and it is extremely important to detect some causes of variations in the process performance.

The Figure A.1 shows the components of a *Test Cell*: the *Tester*, the *Handler* and the *Test Head*.

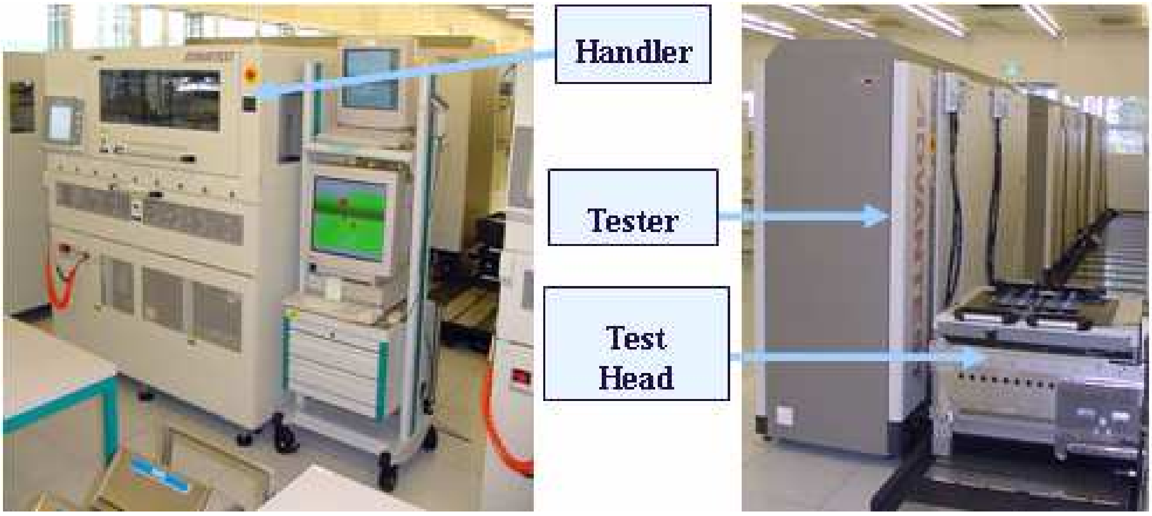

Figure A.18 - Test Cell

[20] Downgrade happens when the components are set with a lower quality due to the fails in some specific tests.

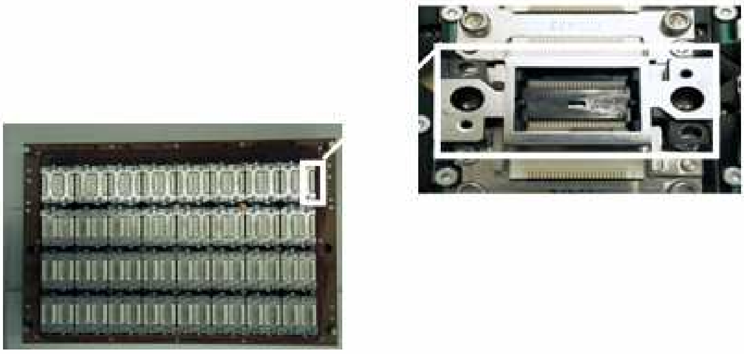

Figure A.2 - Hifix (on the left) and one socket (on the right)

In Figure A.1, one can see a *Test Cell*, having on the left the *Handler* and the two computers in each cell, and on the right, there are the *Tester* and the *Test Head*.

In the Figure A.2, it is represented one *Hifix* and one of the several sockets that are inserted.

Today, there are more than 70 Testers (divided in five types), more than 140 Handlers (of seven different types) and several number of *Hifix* (at least one per Handler), factor that increases the complexity of the Area.

APPENDIX B: Planning and Control tools

This part includes the Planning and Monitoring tools used in the Test Area to plan the production and monitor the performance of the process and equipment comparing to the target values.

The Monitoring tools are related with Figure 14 that describes the actual monitoring process in Chapter 2.

Planning tools:

- *New Volume Interface.xls*
- *Capacity Plan_Test.xls*

Monitoring Tools:

- *Business Objects*
- *Cerberus*
- *TSMON*
- *Backend Framework*

a) New Volume Interface.xls

New Volume Interface.xls is the file used and updated by the *Planning & Logistics* department that contains the Volume plan.

This is one of the most critical files in the planning process containing the production plan for the following 18 months; it is the working tool of the *Volume Planners* where the required volume pretended by the Munich site (where all the central planning is done) is inserted in a weekly basis on this file. The volume planned by the responsible in Munich is a Test-Out plan, i.e., it considers the memories in good conditions. However, the process has a certain Yield, and there are memories that are not in conditions to be sold.

Moreover, there is a need to make a plan having this in consideration and focusing in the process performance. This plan is done in Qimonda PT.

It is important to refer that this file is continuously updated and that the plan for the 18 months changes along the time according to several factors such as product mix, volume required, machines available, process yield, among others.

b) Capacity Plan_Test.xls

Capacity Plan_Test.xls is the most relevant file used in the production planning in the context of this project, because it will supply the program what concerning the weekly production plan. Since its importance is so clear, there will be a need for a deeper explanation about the features of this file.

- **PRODUCTS:**

  All the products that are (or can be) processed in the Porto site are considered in the model and they are grouped by type of Package.

The granularity described in each product contains:

Generation: Die memory size in Megabits

Shrink*:* Through the shrink it is possible to know what is the lithography process and what type of memory it is (SDRAM; DDR I; DDR II; DDR III)

Package*:* The Package is only defined after Pre-Assembly and determines whether it is a TSOP or a FBGA component and defines how many pins or solder balls the component has.

Organization: The organization is defined in the Wire-Bonding process.

SPI-Extension*:* reference to the final customer that will be influenced by the speed and quality.

As an example, Table B.1 describes the different attributes for the product “512M D11 PG-TSOPII-66 x8 ALL”.

Table B.1 - Example of the Product's attributes for the Product

“512M D11 PG-TSOPII-66 x8 ALL”

Product description	Generation	Shrink	Package	Organ.	SPI
512M D11 PG-TSOPII-66 x8 ALL	512M	D11	PG-TSOPII-66	x8	ALL

The products are grouped by package for the fact that products with the same package type can be processed in the same machine without changing the *tooling*[21], i.e., with a few setup times. For the production planner, this makes the planning more simple once it make it easier to allocate the machines into products from the same package type, always when possible. This is a strategy to optimize the production planning in the test area, because the setups are reduced.

- **UTILIZATION:**

The model contains a field with the capacity utilization rate. This value is calculated as an estimation of time that the machine is effectively processing, that is, it gives the percentage of productive time, excluding some non-productive times such as preventive maintenance and so on.

[21] Tooling refers to the Hifix that are different according to the Package type.

- **SHIFTS:**

  There is a field that contains the actual number of shifts per week. In the factory, more specifically, in the production floor, they are divided into two shifts per day, each one with twelve hours, one from 7 a.m. to 7 p.m. and the other from 7 p.m. to 7 a.m. completing the twenty four hours per day.

  There are four different teams with different schedules. The teams change between them having always one team in each shift per day. The number of teams is not taking into consideration in the model because this variable doesn't influence the plan as long as there is always a team in each shift thus allowing the continuous production in the factory.

  The factory works 24 hours per day, 7 days per week. Usually the number of shifts is equal to fourteen (2 shifts per day during the 7 days of the week). This value only changes in some special day such as Christmas, the New Year's Evening and Qimonda's day.

- **EFFECTIVE PRODUCTION TIME:**

  This field is related with the number of weekly available hours for production. The calculation is based on the fields *Shifts* and *Utilization* as it can be seen bellow:

  E.P.T = (Shifts x 12 hours) x Utilization

- **TESTERS' CAPACITY:**

  The model contains specific fields with the capacity of the different machines, especially for type of product and Operation (or Step).

  The value refers to UPH (Units per Hour), indicating how many units can be processed per hour in that machine for that product, in average.

  This value is not constant along the weeks and can vary for reasons such as: different Test programs, an increase in the number of setups, damages, etc. The production planner update (or should update) manually and based on the historical values of the previous weeks, the values for the capacities that had a great deviation from the value planned, having in account the reasons for that variation.

  This update is extremely important since variations in this factor can improve or make the plan worse. Thus, the production planner should be aware and make a scenario as real as possible.

  U.P.H._Plan = average from the UPH values in last five weeks

- **VOLUME PLAN**

  For each product there is a field that contains the Volume planned for the following week, and the Total Volume Planned calculated as the Sum of each volume plans for all products.

  These fields are linked with an MS Excel file ("*New Volume Interface.xls*") that contains the Volume plan of each product for the following weeks. *Planning &Logistics* department is the responsible for this file and its actualization has major impacts so it has to be done as fast as possible for the reason that the production plan is totally depend of the Volume plan values.

  Usually, in this model the value plan from the previous weeks are shown allowing the planner to make comparisons with the actual week.

- **TESTER AVAILABILITY**

  Contains:

  a) **Total nº of machines for reference type**: Usually is a fixed value that changes when there is damage in a machine during a long period of time such as purchasing of new machines, machines that are not more in use or that are not in use for a non defined period of time.

  b) **Number of machines specifically for production:** this value is calculated by removing the engineering hours (hours that are specified only for process engineering lots, qualifications, etc.) from the Total number of machines for reference type. The engineering lots are not part of the production plan, but as they have to be processed, it is important to have them in consideration when the plan is done.

$$Testers\ Available = Total\ n^{o}\ of\ machines\,(a) - \left(\frac{Engineer\ hours}{Effective\ Production\ Time}\right)$$

  c) **Number of Machines used:** is the sum of the capacity utilization of the products in a specific type of machine. It is equal to the sum of the totals of each machine in the different operations.

  d) **Nº of machines used per operation:** is equal to the sum of the utilization for a given machine in a specific operation. Because there are machines that can do more than one operation, there is a need to separate the plan for each operation.

- **CAPACITY UTILIZATION**

  Group of fields in which each cell corresponds to the capacity utilization related to a certain product, in one specific machine and operation.

  In cases where the Volume plan is null (when that product was not planned to be processed) the field returns the value zero. The values are introduced by the planner manually for each cell, having the value zero for defect.

  All the products have to flow in different process operations, depending on their own specifications; therefore all of them have to pass in the operations A and B. After these, some of the product has also to go into operation C or in some cases none of them.

  About the method for the calculation itself the planner usually introduces the value for Test Operation A in order to complete half of the production process and after introduces the value for Test Operation B to complete the value to process 100% of the products.

  This approach is based on the value of the Total Value that will be explained later, which aims to calculate an estimation of the volume planned based on the percentage utilization in the different steps. This assumption is used to make the average between the values of Test Operation A and B that is why the planner should introduce a percentage of capacity utilization for each step that produces "half" of the volume. This is the reason why the planner has to plan a capacity utilization that completes 50% for each step in order that the sum of both is 100%.

  When the decision of which machine to use is done, the production planner need to have in account that the products with the same package shall be processed in the same machines, reducing the need for setups, optimizing the equipment's utilization.

- **TOTAL VOLUME**

  Total Volume corresponds to the volume that is planned to be processed in the following week. Currently, this value comes from *New Volume Interface.xls*, file explained before. Therefore, since the responsibility of the Volume Plan is from the Volume planner, the fields in the MS Excel file are linked with that file, to ensure that all the information is accurate and precise.

c) Business Objects

Business Objects (*BO*) is a data reporting software that allows in a simple way the creation of consultation and data extraction models for every type of Data Bases.

For the planning and monitoring process in the Test Area, *BO* executes visualization and extraction of the production volumes and other ongoing production information that helps the monitoring of key aspects, allowing production performance follow-up.

As an example, with the *BO* software, it is possible to obtain data related to production such as volume produced per operation, machine, and production area with the desired temporal granularity – per day, per week, month or year.

It is also possible to follow the production history for a specific lot such as the starting date in a specific operation and the respective date of its end. Due to the easiness of use, *BO* is widely used in the consultation of *WIP*[22] in production, a basic activity in the weekly volume planning.

d) Cerberus

Cerberus is a software with a collection of applications developed for production reporting. As it was referred in the Chapter 2, *Cerberus* is used in the Test Area mainly with a function of machines' performance reporting, focusing on the time distribution in terms of productive and non-productive time, on a RTC[23] basis. This application, called *RTC Status*, is very powerful because it has an enormous variety of filters that allows the precision and reliability of the analysis. It is available in all Qimonda sites and the analysis can be performed for all the productive areas of the company.

The program returns the RTC status for the Testers choose, indicating the Time distribution for each one. Thus, for each Tester there is a bar with the status indicating, for example with a green colour, the percentage of Productive Time, with a yellow colour the *Standby*[24] and with red the *Unscheduled downtime*[25].

e) TSMON

TSMON (*Test System Monitoring Tool*) is a program used to online monitor all the Test systems. It provides general information about all the machines in the Test Area providing a general overview useful for controlling the production and even detailed information in terms of Tester and Handler names, test program, Lot Identification, Yield and Lot size.

It is used by the production engineers mainly to evaluate the performance in terms of capacity (UPH) and Handlers' Unsynchronism, indicators that explain in part the Testers' performance.

To summarize, TSMON is a powerful tool mostly by the functionality of taking a whole "picture" of the equipment and production in the Test Area.

[22] In production management, WIP stands for Work in Process

[23] RTC- Real Time Clock, meaning that the data is constantly actualized, being the monitoring performed online with an incremental delay.

[24] *Standby time* is the time the machines are not processing due to the lack of materials or operators, despite of being in conditions to work

[25] *Unscheduled downtime* is the time that machines are not processing due to non-planning events such as damage, power cuts, etc.

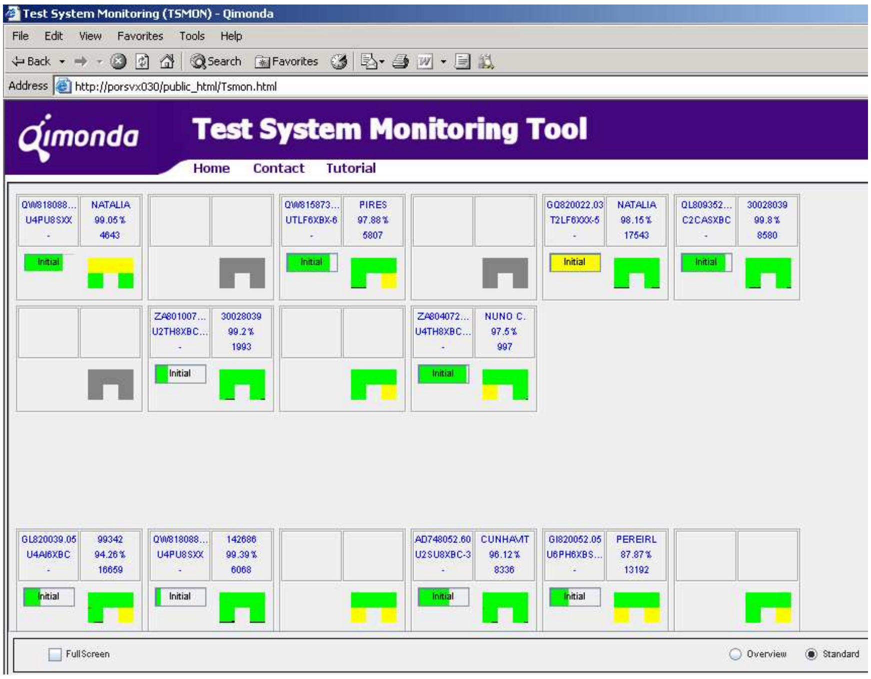

Figure B.1 – *Print Screen* of the Test System Monitoring tool (TSMON)

In the Figure B.1 there is an example of the system usage, showing an overview of some machines in the Test Area, displaying in a green colour the indication that the machine is testing, with yellow if the machines has stopped and with red colour (not showed in the picture) when exists some problems, usually related to the Sockets performance.

f) Backend Framework

The *Backend Framework* tool is an analysis tool that is used to analyze the performance of different measures in the various areas within the company.

The analysis pad used in the Test Area name's TAKPI, denomination for Test Area Key Performance Indicators. Despite of being a useful tool, it has many disadvantages that are pretended to be solved by the *QimTa* tool. If one wants to analyse the performance of a specific product, Tester, handler, week, and so on, the user needs first to specify first the indicator he wants to evaluate and after he needs to wait too much time to get its result not forgetting that he may not have the desired indicator.

Another aspect that makes this tool not so user friendly is the fact that it takes time to understand how to work with it and specially which type of indicator that are according to the user's evaluation.

What concerning the functionalities of the tool itself, there will be made a general description of the tool in order to the reader get a better comprehension of what are the general issues of this program.

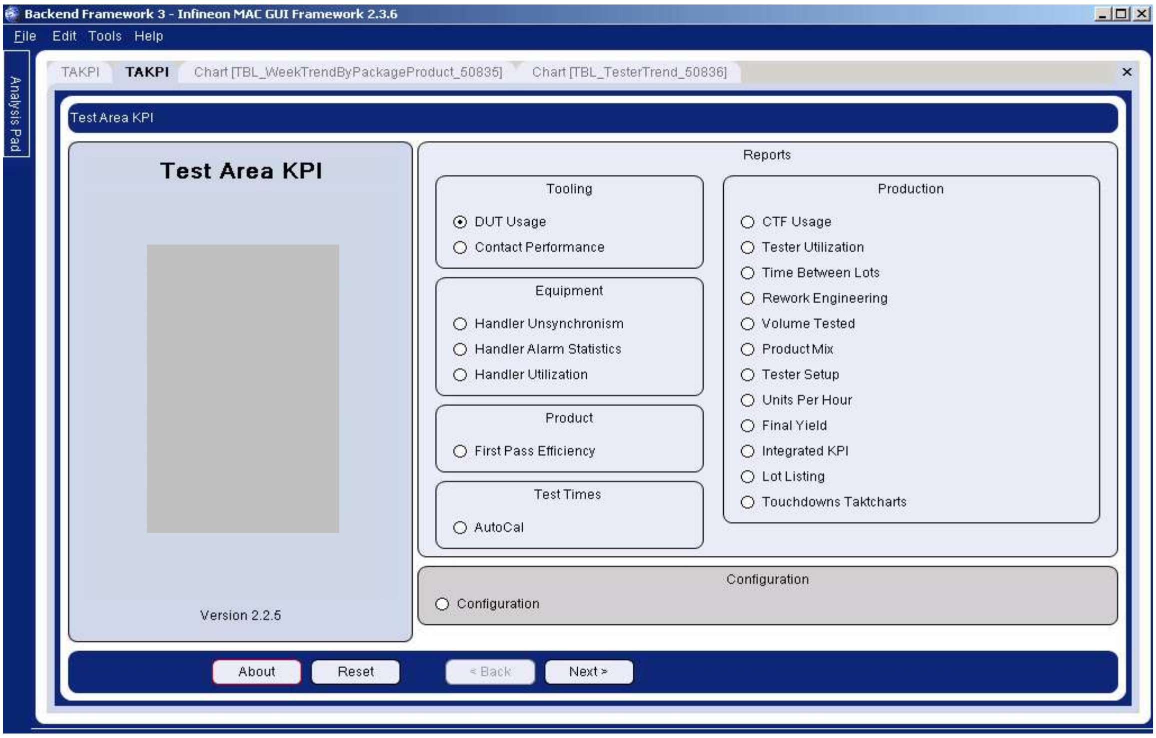

Figure B.2 – *Print screen* of the Backend Framework - TAKPI Layout

The indicators are divided by different areas of search: Tooling, Equipment, Product, Test Time and Production. The most relevant ones in the context of this project are related to the ones from the Production area. Those KPI's were the basis for developing the basis for developing the indicators of the QimTa tool

Inside each KPI, there is the possibility to choose the filter that shall be applied in the analysis. For example in the Tester Utilization parameter (included in the group of the production parameters), the user has the option to choose between "Week Trend", "Week Tester Pareto", "Week Product Pareto" and "Week Trend by Tester Type". Of course all of these options have their relative importance but it is important to understand that for a new user it might be complicated to choose between one of those. As it was said previously, this is a pitfall that must be improved.

Another program's feature is the option to execute some configurations in order to update the information and adapt with the current situation. To give an example, the users whom have permission have the possibility to select the machines that are in production and the ones that are intended to the Engineering processes like the qualifications, and so forth. This is a useful option since the user may choose if he wants to evaluate just the productive machines, the engineering ones or both. The type of filter that is selected for the analysis is extremely important because it has direct influence on the results.

An additional functionality is the possibility to choose how to get the data's format. If for one hand, one can choose just the graphical analysis, in another it is also possible to get an *MS Excel* file with all data used to perform the charts. Through this option the user can work, manipulate the data, and specify its own charts. As additional features, the user can save, print, copy, move, zoom and edit the chart data in order to perform the analysis in another way.

APPENDIX C: Granularity

This appendix will describe some concepts that are essential for the reader's comprehension of the report, especially in the section "Program Specification".

The main focus of this part is the concept of granularity. Granularity is "the level of specificity with which content is described. The more granular, the more specific." [15]

In general, the product's granularity is essential to plan the production for a correct visualization of the inventory that exists in the production line. It is affected by all the logistic IT[26] rules created for a better management of the production and the process engineering.

First, it is important to understand the product description in the Production Plan, more precisely, in the *Capacity Plan_Test.xls*. Figure C.1 illustrates the different levels of granularity to enhance the relation between product description and granularity.

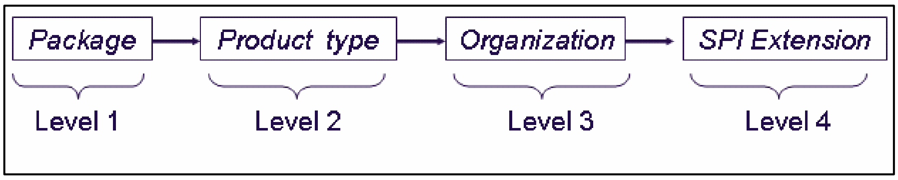

Figure C.1 - Granularity levels included in the product description

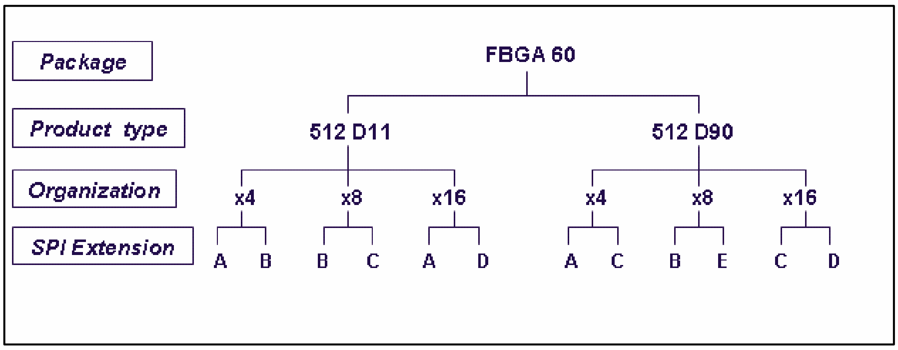

Figure C.2 - Scheme representing the granularity levels for the package FBGA60

Figure C.2 describes the products in the *Capacity Plan_Test.xls* are divided in four different levels according to the pretended granularity. The granularity is very useful for the production plan, once it allows grouping the products in different types according to its nature. Thus, for example, the products with the same Package type are grouped because they can be processed in the same machines without changing the tools, hence, trying to reduce the setups in the area. Therefore, for the production planner, this makes the planning simpler as it makes it easier to allocate the machines into products from the same package type, always when possible.

Notice that the Product type is a combination of two product characteristics (that can also be considered as single parameters) the *Generation* and *Shrink*.

[26] IT – Information Technology department, responsible for all the programs' implementation and maintenance within the company.

As an example, Table C.1 describes the different attributes for the product "512M D11 P-FBGA-60 x8 ALL", in order to understand the concept of granularity in the context of the product description.

Table C.1 – Example of the product's description and the attributes related with the granularity levels

Product description	Generation	Shrink	Package	Organiz.	SPI
512M D11 P-FBGA-60 x8 ALL	512M	D11	TFBGA-60	x8	ALL

In the production plan, there is a distinction between the Black (P) and Green (PG) types of packaging but as this parameter does not influence the planning process, it will not be considered in the program being developed. The two types of Package colour are grouped in the same Package type.

To make a parallelism between the *Capacity Plan_Test.xls* and the specified program that is specified, a brief explanation will be made. First of all, the program that is specified (*QimTa*) has only three types of selection parameters what concerns the products' description. Those are:

- *Package* (or Package type),
- *Product type*
- *Configuration* (or *Organization*).

In this way, it is possible to define the product with three levels of granularity (the first three in the Figure C.2), being the descriptions equal in the production plan. The SPI Extension is absorbed in the program, and the different types of extensions grouped as a single one called "ALL".

The major difficulty in the program specification and even in the creation of the prototype was related with the difference (in the Product descriptions) between the Production Plan and Test Database. Therefore, it was important to specify and parameterize the different variables in order to avoid errors and make the program reliable, flexible and understandable for all users.

As it was explained before, there are three levels of granularity, the *Package*, the *Product type* and the *Configuration*. All these parameters have a different description in the Test Database. Starting with the *Package*, the next table shows three different examples that compare both realities.

Table C.2 – Package type description - relation between the Production Plan and Test Database

Production Plan	Test Database
P-TSOPII-54	TSOP54
P-TSOPII-66	TSOP66
PG-TFBGA-136	FBGA136

In *QimTa*, the Package type specified is equal to the one used in the Test Database. This decision was made to give simplicity to the description when comparing to the Production Plan, reducing in this way the probability of errors in the parameterization.

The Product type was defined as described previously in the Production Plan, due to its simplicity. However, there is also the need to parameterize the product type in order to make the relation with the Test Database definition. The relation is done with the field called Design in the *LOTDATA* table. This parameterization is explained in more detail in the section 3.5).

Last but not least, the lower level of granularity, "Organization" (also called "Configuration") is defined in the same way as in the *Capacity Plan_Test.xls*, parameter that is obtained in the Test Database through the "Program" field, once the fifth character of the program name describes de configuration type of the product. For example "8" in the Program field, represents the configuration type "x8"; in the same way, "6", represents the configuration type "x16" and so on.

To summarize what was explained before, the concept of granularity is essential to understand the specification of the program. The different levels of granularity will influence the results returned by the program, and can be very useful to control the Test Area. As an example, if the user knows that a specific type of product is having a lack of performance for some reason, he has the possibility to go further in the analysis and go one level down in the granularity to try to identify precisely in which specific product type are the low performers. It has also the possibility to go to the third granularity level and evaluate which the Configuration from the specific product type that has a deviation from the target.

APPENDIX D: User Suggestions (Requirements)

This Appendix contains the description of the suggestions made by the future users of the program. As referred in the User Requirements (section 3.2)), not all the user's requirements were fulfilled but still, it was taken in consideration the ones which were more important for the good functionality of the tool.

These suggestions were obtained through different practices such as brainstorming and by direct contact with the users along the development phase. Thus, there was a continuous follow up during the different stages of the development. The suggestions are described bellow.

The program shall:

1) Return the values from the Production Plan, in terms of Volume, Test Operation and respective testers and the related capacity and utilization planned.

2) Return the Volume tested until the moment of the research and the deviation against the value given by the Production Plan (through the *Capacity Plan_Test.xls*).

3) Allow the consultation of machines' capacity (UPH), with the indication of the ones that have a great deviation from the plan. For example, based on the rule:

   Abs (UPH Deviation) > 20%

4) In the worst case, alert in the first menu which are the machines with a lack of performance.

5) Include a selection menu based on the Testers and not in the products.

6) Give indication of machines that perform setups (Low and High Temperature).

7) Be released in the production line in order to be available for all the operators (especially senior operators) in a way of acting more quickly.

8) Show the Testers that are being processed now and not the ones that have already been processed, like the *TSMON* tool where the machines that are online are shown.

9) Indicate the quantity produced (or better, tested) comparing it to the Production Plan, having in consideration the moment of the analysis. In other words, if the Planned Volume is 70k[27] (70.000 units), meaning that in average it must be produced around 10k, at the end of the third day of production there shall be processed at least 30k (10k x 3 days).

[27] Usually for Production Planning effects the volume is referred in terms of "k" units. One "k" unit is equivalent to 1000 units.

10) Give the possibility to take a general picture of the “production performance”, by selecting not the Package or Product Type, but yet the production as a whole.

11) Give the option to select, not only per week or day, but also per shift.

12) Specify the values in the volume in terms of “kpcs” (thousands of units/pieces) instead of “pcs” (pieces).

13) Indicate the average *Lot Size*, next to the *TBL* indicator, if possible.

14) Display the percentage deviation of the *UPH* indicator, close to the actual values of *UPH*. This deviation must be calculated as the difference between the actual value and the value planned.

15) Set targets for the *TBL* and *Tester Utilization* indicators.

16) Indicate the program associated to the *UPH* value. (This suggestion is due to the difference that exists between the Test Programs associated to the machines’ capacity).

Some of those suggestions were not possible o include in the tool.

First, the fifth suggestion was not totally followed once the user can select the machine pretended by referring to a specific product. Therefore, the results will be returned not for the overall performance of the Testers, but yet for the Tester’s performance which refers to a specific product previously selected. This is not so different, once the majority of the machines produce only one type of Package and hence if the user performs a selection by Package type, the results will describe the general performance of the machine.

Second, the suggestion given in point 6) was not totally fulfilled for a logical reason. Since the program gives the information about the Test Operation that the machine has been performed, it means that the Tester had to make a Setup in terms of operation and hence, the requirement is considered, but in another way.

Third, the proposed version should be available in the production area (suggestion 7)), but once the Solution primarily implemented will be part of an existing software, that will be not possible, because that software cannot run in that computers. Despite this, in a second phase, this suggestion will be taken into consideration.

About the eighth point, it shall be have in consideration that the program is for monitoring, hence, the main objective is to evaluate the performance comparing it to the plan and not to evaluate only the actual Testers’ performance with referring the plan.

Last but not least, the tenth requirement will not be included for the simple reason that this tool is intended to monitor the Test Area based on the comparison of the Production Plan with the actual status and the suggestion made is already available in other tools.

APPENDIX E: Use Case Model

A *Use Case Model* is usually represented by one or more use case diagrams and any supporting documentation such as use case specifications and actor definitions.

Within most use case models the use case specifications tend to be the primary artifact with use case diagrams filling a supporting role like the "glue" that keeps the requirements' model together. Use case models should be developed from the point of view of the user's requirements and not from the (often technical) point of view of developers.

A use case describes a sequence of actions which provides a measurable value to an actor. Usually the use case is drawn as a horizontal ellipse on a UML use case diagram that can be liked with other use cases and connected with the involved actors.

Despite of these definitions been simple and easily understandable, use cases are not well – defined. In fact, little attention has been given to the various styles for writing the narratives that define use cases and their consequences for user interface design and software usability. Common narrative styles are presented with examples and discussions of their relative advantages and disadvantages. [16]

Essential use cases, a variant employed within usage-centered design, are contrasted with conventional use cases and scenarios. For the most efficient support of user interface design and particularly for large, complex projects, a highly-structured form of use case has evolved. New narrative elements and relationships among use cases were introduced. These include means for expressing partial or flexible ordering of interaction, relationships with business rules, as well as a clarification of the often misunderstood concept of extension that recognizes two distinct forms: synchronous and asynchronous extensions. [16]

According to Bittner and Spence, "Use cases, stated simply, allow description of sequences of events that, taken together, lead to a system doing something useful."

Jacobson's original definition [Jacobson et al., 1992] is brief, broad, and barely descriptive: "A use case is a specific way of using the system by using some part of the functionality. It constitutes a complete course of interaction that takes place between an actor and the system." [16]

Despite of this undefined concept of Use Case Model, or better, lack of a generic definition for that concept, since their introduction in support of object-oriented software engineering, use cases have enjoyed a seemingly explosive growth to become ubiquitous in both development methods and development practices.

Part of this ubiquity can be attributed to their utility—use cases have proved to be versatile conceptual tools for many facets of design and development—but part may also be a consequence of a certain imprecision in definition. Most developers can say they are employing use cases because almost anything may be called a use case despite enormous variability in scope, detail, focus, format, structure, style, and content. Further muddying these already turbid waters, idiosyncratic terminology has been promulgated that obfuscates important distinctions, such as that between scenarios and use cases. [16]

As an effective bridge between usability engineering and user interface design on one hand and software design and development on the other, part of the promise that use cases offer is due precisely to their chameleon-like adaptability. For engineering requirements, *Use Cases* provide a concise medium for modeling *User Requirements*; in the hands of user interface designers, use cases can become a powerful task model for understanding user needs and guiding user interface design; for software engineers, use cases guide the design of communicating objects to satisfy functional requirements. Success in all these endeavors rests on the realization that user interface design is not software design. Models originally developed to support the design of software components and their interactions are not automatically and necessarily well-suited for organizing user interface components and the interaction between users and these components. [16]

Use Case Diagram

A use case diagram is “a diagram that shows the relationships among actors and use cases within a system.” Use case diagrams are often used to:

- Provide an overview of all or part of the usage requirements for a system or organization in the form of an essential model or a business model
- Communicate the scope of a development project. Therefore, this was very helpful not only to explain the users what is intended to do (related with the *User Requirements*), but also to do the developers understand what are the system and its context.

To build a Use Case Diagram is important to know how to identify potential use cases. Constantine and Lockwood (1999) suggest one way to identify essential use cases, or simply to identify use cases, that is to identify potential services by asking the stakeholders (future users) the following questions from the point of view of the actors:

- What are users in this role trying to accomplish? (related to *User Requirements*)
- To accomplish this role, what do users need to be able to do?
- What are the main tasks of users in this role?
- What information do users in this role need to examine, create, or change?
- What do users in this role need to be informed of by the system?
- What do users in this role need to inform the system about?

All of these questions were essential to build the diagram.

Now is important to define some concepts that are implicit in the *Use Case Diagram*:

Use Case – A use case describes a sequence of actions which provide a measurable value to an actor and is drawn as a horizontal ellipse on a UML use case diagram. [3]

Actor – An actor is a person, organization, or external system that plays a role in one or more interactions with the system (actors are typically drawn as stick figures on UML Use Case diagrams). [3] It is possible to distinguish between two types of actors:

- Primary Actors interact directly with a system to achieve their goals;
- Supporting Actors may be humans or systems called in to support the Primary Actor.

Relationships – Are depicted as lines connecting two modelling elements with an optional open-headed arrowhead on one end of the line indicating the direction of the initial invocation of the relationship.

System Boundary Boxes – the rectangle around the use cases is called the system boundary box and as the name suggests it indicates the scope of the system – the use cases inside the rectangle represent the functionality that are intended to implement.

APPENDIX F: Conceptual Model

Class Diagram consists on a group of classes and interfaces reflecting important entities of the business domain of the system being modelled, and the relationships between these classes and interfaces.

Blaha & Premerlani suggest some guidelines to create a Class Diagram: [17]

1. To Identify Classes (search for Proper names and common names)
2. To Identify Associations (search for transitive verbs and propositions)
3. To Add Attributes, detailing its Classes and Associations
4. To Use Generalizations to characterize similarities and differences between objects.
5. To Use access pads and information search.
6. Refine and frequent follow-up the model, adding, deleting or changing the detail or abstraction level.
7. To organize graphically the final model.

In the case of the tool that was created, the Conceptual Model was not a Classes' Diagram. The model developed was inspired in some issues usually related to the typical Conceptual Model of Classes, but the final result was not a model with several objects, attributes and relationships. The parallelism that exists is related to the existence of different parts of the system (not real actors) and different flows between them (not real relationships). However, if one focus on the part of the system that will feed the program (*Data Bases* and *MS Excel* files) there is a bigger approach, once there will be a true relation between them and the other part of the system.

Despite of the existent differences, the previous framework was very useful as a guideline to define the parts within the system and the flows and relations that shall exist, keeping in mind the constant follow-up and iteration during all the project duration.

Figure F.1 shows the Conceptual Model's diagram.

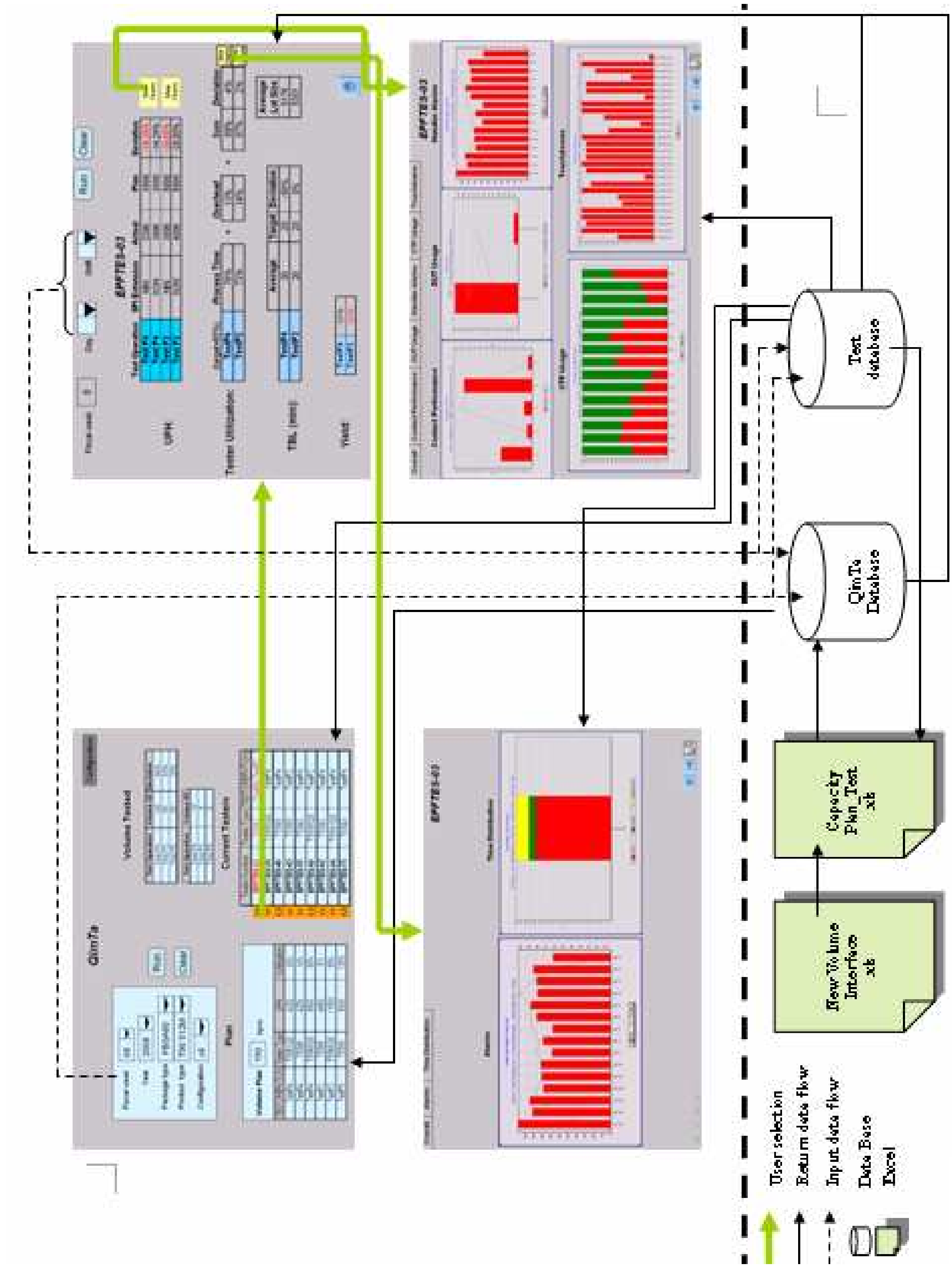

Figure F.1 – Conceptual Model for the proposed solution (*QimTa* tool)

APPENDIX G: *Production monitoring*

This Appendix pretends to explain the solution that will be implemented in a first phase. This solution is based on the proposed solution and the modifications are related with the environment where it will be created. In fact, the proposed solution (*QimTa*) is a better solution for the reasons described in the main body of this report, but yet, the one which was suggested and will be implemented in a near future (*Production monitoring* tool) it also provides many advantages for the Test Area.

First of all, it is important to explain the reason why this solution was suggested as a replacement of the other program (*QimTa*).

The tool will be developed by the IT Department; therefore their suggestions and feedback are extremely important for the execution of this program. After some discussions and brainstorming sessions, the solution that came up seems that can bring advantages comparing to the other, in a first stage. These advantages are due to the fact that the new tool (*Production Monitoring*) will be created as a functionality of an existent program (*Backend Framework – TAKPI*) that is used in the Test Area. It is logical that this fact can make the development become faster and cheaper once it is inserted in a program that have already been created, and hence, part of the hard work is already done. Another advantage is related to the fact that the new solution has many similarities with some functionalities of the *Backend Framework*, specially in terms of KPI's and therefore the effort in the creation of this suggested solution will be barely reduced.

The modifications that were suggested for the new tool will influence in the structure of the system to be created, thus there was a need for a different Conceptual model, described in the two following pictures. In fact, there are many similarities when comparing with the model created for *QimTa,* especially in the parts that will supply the program with the data. As one can notice, all the files and Databases which are connected with the program are the same as the ones in the *QimTa* program. The main difference is related with the interfaces where the information and results will be provided. In the *QimTa* tool, all the results are returned in different windows but inside the same program. In the *Production Monitoring* version, the results will be exported to *Ms Excel* files reducing, in part, the program's flexibility.

While the Figure G.1 is a conceptual model for the general functionality of the program, the second picture represents the model in more detail, complementing the first one. Figure describes how the system will work, concerning the reports in *Ms Excel,* pretending in this way to illustrate, in one hand the complexity behind the creation of this model and in another hand the flexibility that is intrinsic in the tool.

For an even better comprehension of the reports, the Figure G.3 shows an example of one type of graph exported by the program that can be accessed through both windows in the bottom of Figure G.2. Notice that next to the chart, on the right side, there is a menu containing links to the other charts and to the *General Performance window*, in order to improve the flexibility and usability of the tool, increasing the easiness of the analysis.

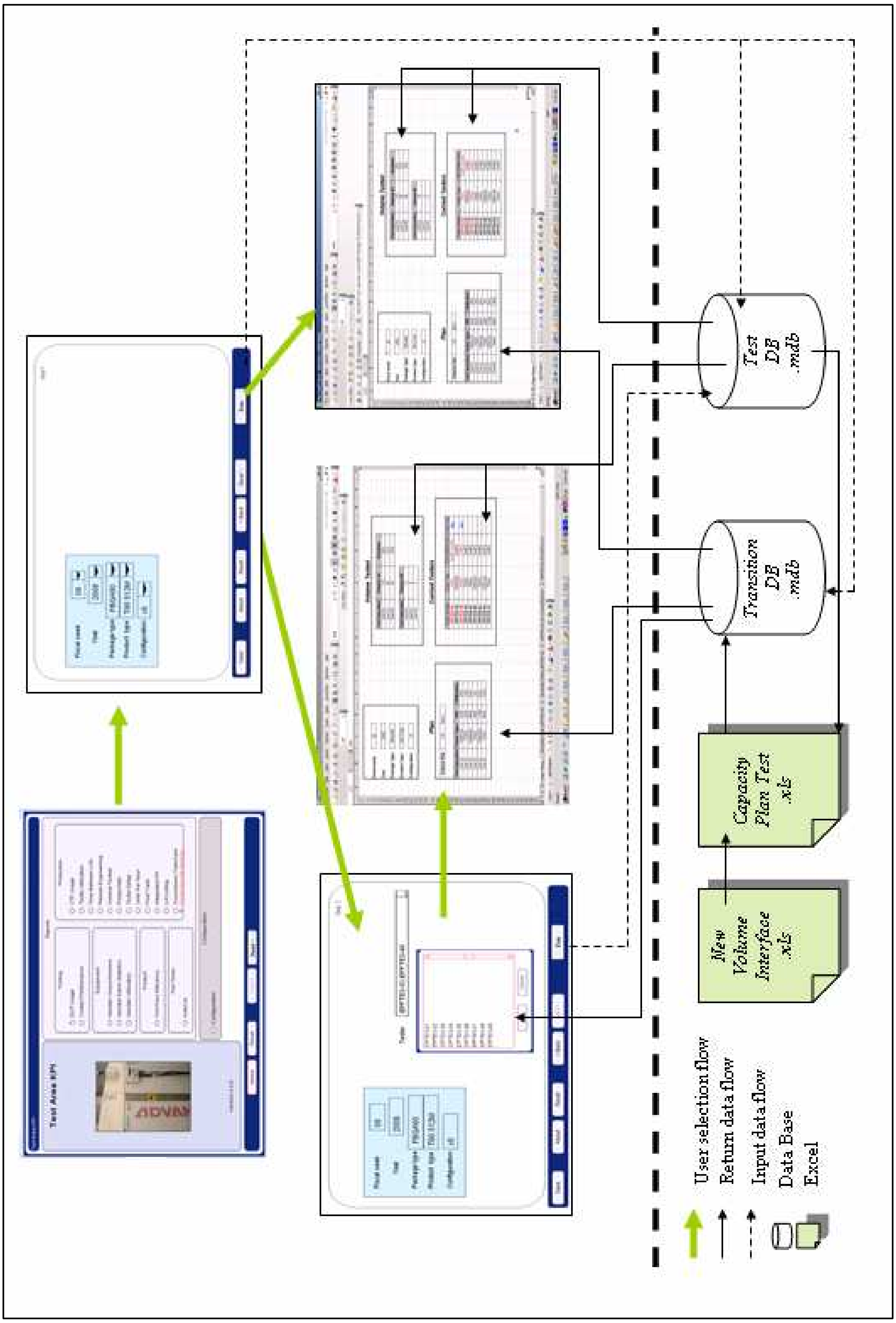

Figure G.1 - General Conceptual Model for the *Production Monitoring* version (Part A)

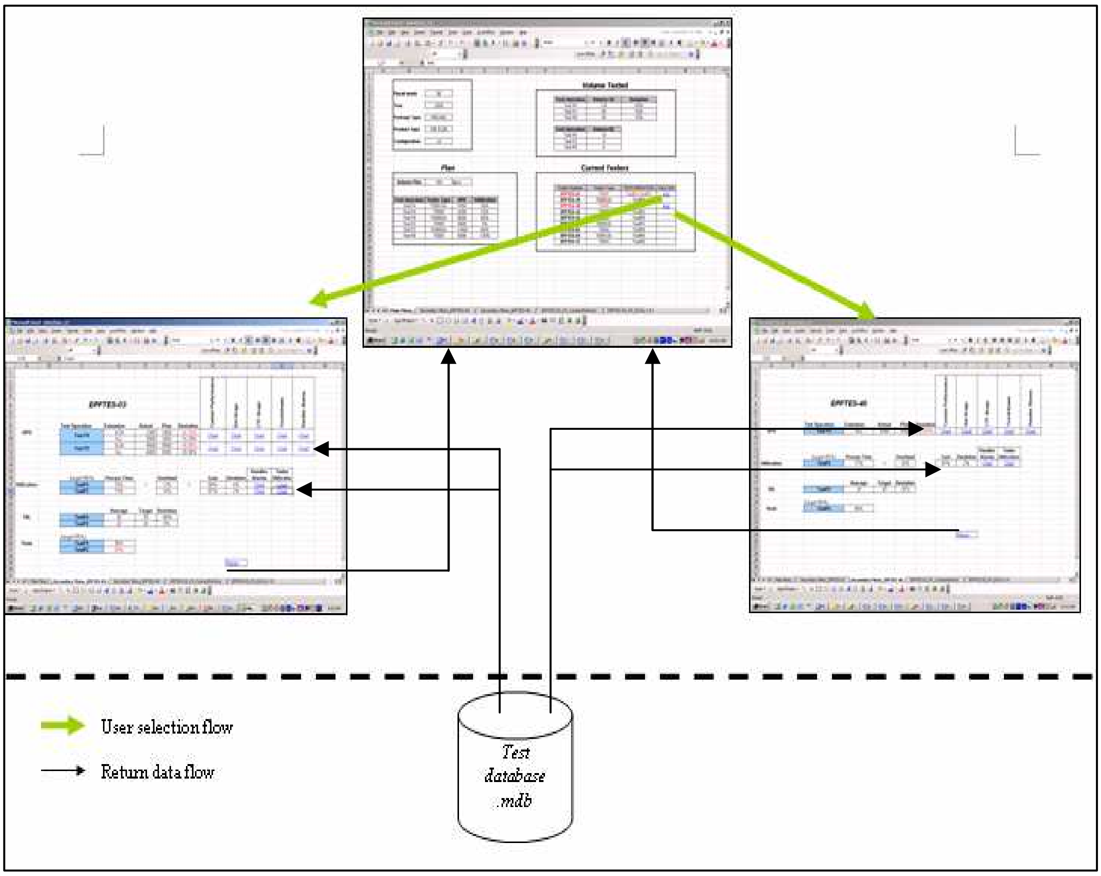

Figure G.2 - General Conceptual Model for the Production Monitoring version (Part B)

Each of the two windows in the bottom of Figure G.2 have links to other tabs in the same MS Excel sheet or tab that contain the KPI's charts for that specific Tester. For instance, the Figure G.3 is an example of one of those sheets containing a graph, not forgetting that, for each Tester, there are six different sheets with the different KPI's.

About the other specification of the tool, it will not be referred, once they are the same as the ones for *QimTa* program.

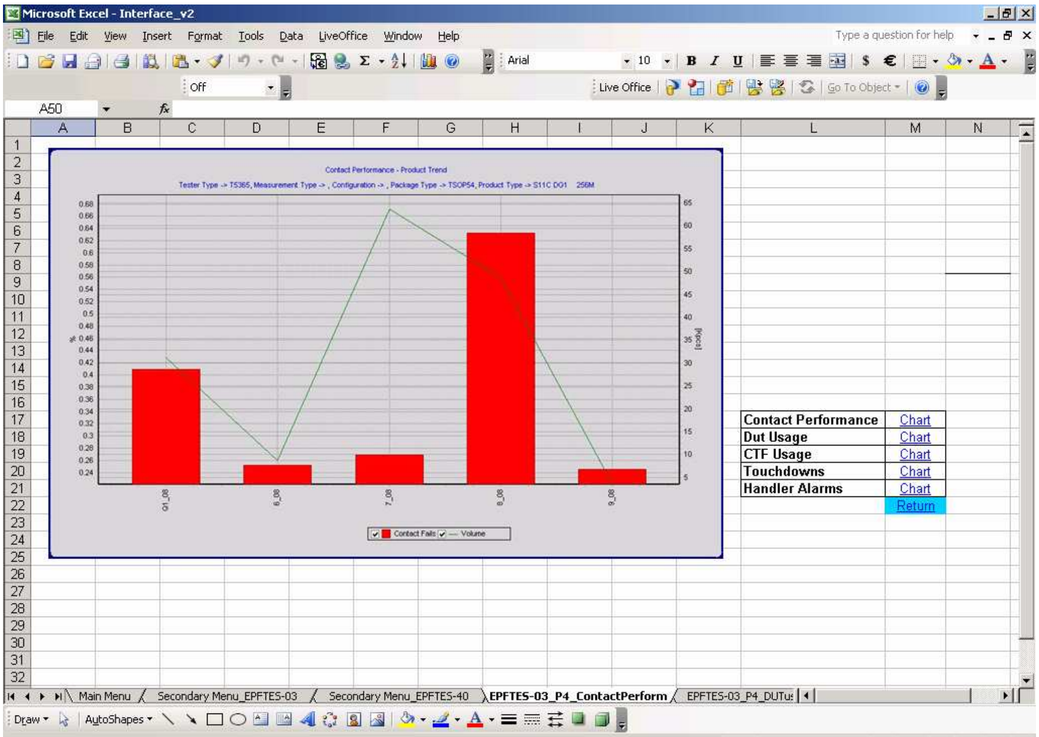

Figure G.3 - Example of a KPI chart contained in the report

APPENDIX H: Business Case - methodology and results

This Appendix explains the quantitative results of the Business Case and all the calculations done during the analysis.

The following steps describe the methodology used and the results achieved by the comparison between the two methodologies.

1. Perform a monitoring process like is done actually and register the total time spent in that process;

Step 1) Run the ***Business Objects*** program in order to detect the products where there are losses in terms of volume when comparing to the Production Plan.

Step 2) See which Tester(s) are processing that product through the consultation of a document containing every product and the respective Testers that are processing them.

Step 3) Run ***Cerberus*** in order to analyze the Testers that are processing that product, in terms of Time distribution.

Step 4) Run ***TSMON*** to get an overview of the Testers' behaviour (concerning the *Handlers synchronism*, *UPH*, etc.) and compare the actual UPH with the planned UPH used in the Production Plan.

Step 5) Run the ***Backend Framework - TAKPI*** program (the report named "Lot Listing").

Step 6) Based on the "Lot Listing" report (document in MS Excel file), it is necessary to calculate and evaluate the following KPI's with the data from this report.

- *UPH;*
- *Tester Utilization;*
- *Time Between Lots;*
- *Lot Size;*
- *Yield.*

Step 7) Execute ***TSMON*** to analyze if the problem is related with the tooling. It is possible that the Tester is having a different tooling than expected causing a variation in the machine's capacity.

Step 8) Go further in the analysis and run the charts for the other KPI*'s* (using the ***Backend Framework - TAKPI*** program) related to the *UPH* and *Tester Utilization*, from the operations with low performance. With this analysis it is possible to detect the problems in order to take the adequate corrective action.

Table H.1 shows the results obtained after five evaluations.

Table H.1 – Results from the experiments using the actual evaluation process (Expressed in seconds)

Steps	Experiments				
	1	2	3	4	5
1	34	34	30	33	32
2	17	15	16	18	18
3	63	150	119	170	70
4	150	411	250	303	156
5	507	309	242	438	486
6	558	1300	1412	1590	547
7	30	45	40	39	32
8	1500	3000	3000	4500	1500
Number of machines with problems	1	2	2	3	1
Total	2859	5264	5109	7091	2841
Eq. 1 Tester	2859	2834	2719	2716	2841

The analysis was based on a Package type, evaluating its potential causes of losses through the machines that were processing it. The number of machines with low performance in the different experiments was not the same, therefore it was decided to transform the values to a single machine, in order to be possible the comparison between the experiments. Notice that the calculation was made by dividing the Steps 3, 4, 6 and 8 by the number of machines evaluated, as these values depend on the number of machines. The time that the other steps take to monitor the machine's performance does not vary with the number of Testers.

Notice that the steps were similar to the ones explained in the section 2.3), being the only difference in the fifth step of the method, where it was decided to divide it in two (Steps 5 and 6 represented in the table). This decision was taken because it was important to distinguish between the time that the program takes to run and the analysis of the reports given by the program.

2. Perform a monitoring process with the new tool and register the total time spent in that process;

The methodology used to evaluate the time that it takes to monitor the production in Test Area with the new tool was:

Step 1) Run the ***Business Objects*** program in order to detect the products where there are losses in terms of volume when comparing to the Production Plan.

Step 2) **Run *QimTa*** for the product that is going to be evaluated.

Step 3) Evaluate the Tester(s) with low performance and run ***QimTa*** for that specific Tester(s).

Step 4) Evaluate the following five KPI*'s* in ***QimTa***.

- *UPH;*
- *Tester Utilization;*
- *Time Between Lots;*
- *Lot Size;*
- *Yield.*

Step 5) Execute ***TSMON*** to analyze if the problem is related with the tooling. It is possible that the Tester is having a different tooling than expected causing a variation in the machine's capacity.

Step 6) Go further in the analysis and run the charts for the other KPI*'s* related to the *UPH* and *Tester Utilization*, from the operations with low performance. With this analysis it is possible to detect the problems in order to take the adequate corrective action.

Table H.2 shows the results obtained after five evaluations.

TableH.2 – Results from the experiments using the proposed evaluation process

Steps	Experiments				
	1	2	3	4	5
1	34	34	30	31	31
2	27	25	31	28	25
3	38	82	90	95	35
4	23	54	58	61	23
5	30	54	60	72	28
6	658	898	1018	1102	598
Number of machines with problems	1	2	2	3	1
Total Time (Seconds)	810	1147	1287	1389	740
Eq. 1 Tester (Seconds)	**810**	**630**	**704**	**550**	**740**

Notice that the method used in the calculation was based on the proposed monitoring process roadmap (Figure) with some difference. Thus, the steps 2 to 4 in the Table H.2 corresponds to the second step in the monitoring process roadmap. This decision was taken in order to measure the different stages of evaluation inside *QimTa* tool.

3. Register the difference of time spent between the two methodologies, that is, the time saved by using the new methodology.

Table H.3 shows the results obtained by calculating the difference between the results from Table H.1 and Table H.2.

Table H.3 – Difference of time spent (comparison between the two methodologies)

Time Saved	Experiments					Average
	1	2	3	4	5	
Seconds	2049	2204	2015	2165	2101	2089
Minutes	**34**	**37**	**34**	**36**	**35**	**35**

4. Register the difference of capacity, i.e., the average deviation of the actual UPH values when comparing to the planned ones.

Table H.4 – Average deviation (Comparison between the actual and planned value of *UPH*)

	Experiments					
***UPH* Deviation**	**1**	**2**	**3**	**4**	**5**	**Average**
	45%	35%	40%	45%	50%	43%

The values in Table H.4 were obtained by the registration of the deviation between the actual and the planned values of *UPH*, given by the Prototype.

5. After the experiments have finished is important to get a consistent value for the average time saving and for the average deviation of capacity.

Results:

- Average Time saved: 35 minutes
- Average capacity (UPH) deviation : 43%
- Reference value of capacity (in equivalent units): 2100 UPH
- Number of occurrences per week: 5 (minimum) and 14 (average)
- Number of productive weeks per year: 52

The capacity for the reference product used in capacity estimations by the Planning & Logistics department is 2100 *UPH.*

The number of occurrences can vary. In a general way, each team shall perform the evaluation at least once per day. There are two teams (one per shift), so in one week there will be made 2x7=14 analysis. The minimum value of 5 represents the lowest number of machines which performance will be improved. Those values were obtained after some weekly evaluations about the number of occurrences in the machines and their detection.

The formula used to estimate the additional capacity was:

$$Additional\_capacity = ATS \times ACD \times RVC \times NO$$

With:

ATS – Average Time saved (hours)

ACD - Average capacity deviation

RVC – Reference value of capacity

NO - Number of occurrences

Therefore, the values for the additional capacity are:

Table H.5 – Weekly and annual values for the additional capacity in terms of units processed (for 5 and 14 occurrences)

Number of occurences/week	Addicional Capacity units/week	Addicional Capacity units/year
5 (minimum)	>2.500 units/week	>130.000 units/week
14 (average)	>7.000 units/week	>364.000 units/week

After the analysis being explained is time to analyse the results.

First of all, the benefits are clear. There are a reduction of time in the monitoring process in more than 75% that are translated to an additional production of more than 130.000 units/year (for the minimum number of occurrences) and about 364.000 units/year (for the average number of occurrences).

Second, it was possible to detect the critical steps when using the actual method. Steps number 6 and 8, were the ones that have a higher duration and hence a greater impact on results.

Third, in the proposed method, the critical step was the last one, as it was around 80% of the Total Time.

Finally, the higher the number of machines analysed in each experiment, the lower will be the equivalent time spent per machine. In other words, the evaluation methods are more efficient as the number of machines analysed increase, especially in the proposed evaluation process, using *QimTa* program.

APPENDIX I: Improvement suggestions in the Production Plan (*Capacity Plan_Test.xls*)

Currently the production planning/management is done based on a specific weekly plan elaborated by the *Planning & Logistics* department that, based on the weekly available capacity, makes a forecast for the Production Volume for the next weeks. Then the Production Engineer that already has the plan from the Test Area has the function of allocating and distributing those products in several machines available in the area according to the capacity restrictions.

After analysing the file that is used, on a weekly basis for managing the resources (machines and tooling), it was visible that there were some limitations, mainly related with the time needed to perform the plan, every week.

The most relevant limitations/inefficiencies of the actual model are described as follow:

- The need to make the planning allocation “manually”, product by product, step by step for the different Testers.

- The need to do manually the division between the two machines in the core operations (Test Operation A and Test Operation B). In other words, if the responsible person plan for a specific product to flow only in one operation (is not correct once the product has to perform both operations), the system will not alert for that fact, not restricting the user in his plan. This can lead to an incorrect planning that can have important consequences that must be avoided. Despite of this situation be very improbable, this fact shall be considered.

- The necessity to analyse permanently if there are Testers available to allocate the products. In this case, it could appear an indication when there is any available machine of that type in order to plan the allocation in a different way.

- There is inefficiency when there are several machines being allocated in just a small period of time instead of allocating just a few, but with higher capacity utilization. Thus, there is a need to create a function that optimizes the equipments’ utilization. Therefore, instead of using 20% of a machine for the product A and 30% in product B, it shall be allocated 50% of single machine.

- Sometimes there are some problems in the link for the file *New Volume Interface.xls* that can lead to some difference between the volume planned between the two files, and shall be avoidable. This can occur for several reasons and hence it is suggested to create a simple alert when the field “Total” from the *New Volume Interface.xls* is different from the one in the *Capacity Plan_Test.xls*.

- The great difficulty that it turns in the biggest challenge of the planning will be the machines’ allocation, i.e., how many machines shall be allocated to each product in order to make it in such a way that machines only test one type of Package by not having to make setups for exchange the “Change kits”.

After pointing out some inefficiencies of the program, there will be described some measures that can lead to an improvement in the process planning, as an optic of continuous improvement of the Test Area resources.

Conditions/ Restrictions to apply in the Production Plan (*Capacity Plan_Test.xls*):

In general, what is pretended is to do the production plan the more automatic as possible, with the minimal intervention required by the Production Engineer, through some definitions for the creation of an algorithm.

- The total number of machines cannot exceed the number of existing machines.

- The machines that had been used previously (in the previous week) to process one specific product, must be used to process that exact product (or product of the same Package) in the following week. It is a function that had to exist since the beginning, once it aims to reduce setup times and it allows optimizing the use of the *Testers*. Obviously, the Project Planner tries to follow this logic, but does not exist any mechanism that hinders it, to make in another way.

- To plan the same nº of machines that had been allocated in the previous week for the same type of "Package". This condition has some implications because that the product mix has a wide variability along the weeks and hence, the number of machines to process a specific Package can vary from week to week. Therefore, this function will need some maintenance by the Production Engineer.

- The number of *Testers* planned cannot exceed the number of available machines.

- To have a field where it can be defined the number of available machines of each type. This field will allow to plan according to the available capacity, letting the system knowing which machines can produce in a certain week.

- As already been referred, the system must allow some manual alterations, in one hand to make some alterations in the plan and in another hand to perform the maintenance of the system to become it most effective possible.

- The algorithm must prioritize the processing in CTF[28] mode, in order to reduce Setups, increasing the UPH in about 20%.

[28] *CTF* means *Continuous Test Flow* that is a processing mode that aims to produce the batches continuously, without stops, what makes the machines utilization more effective, allowing an increase in about 20% of Testers capacity.

- One of the entrance parameters in the system is the capacity of the machines in UPH, thus, the system will allow calculating the use of the different machines focusing on the respective capacity. The system will know from the beginning the flow from the different products, because when the UPH=0, for the several machines in a specific product, it means that the product does not have to perform that operation.

  From this moment, this question only matters in terms of "Speed Operations" (Test Operation C), once all the products have to flow in the other two operations (Test Operation A and Test Operation B). With this, the system also knows which machines will be able to process, that is, the machines that possess necessary tooling to process it.

- Define priorities for the different products. There are products that need to have an average stated period of deliveries sufficiently reduced, and hence, become priority for the production. This point is in account currently, although its allocation is made manually. A proposal for improvement would be the creation of a code of values (for example a scale of 0 to 2) where the products with level "2" would have maximum priority, the products with level "1"medium priority and products "0" were less urgent. Clearly that this scale could be widened according to the necessity, being able to increase with the increase of the mix of products, for example. This functionality needs maintenance from the part of the responsible people, once is inserted in a context of a volatile market, thus, the necessities of the customers are constantly modifying.

- Another suggestion of improvement would be the definition of a rule to prioritize utilization of the equipment with greater capacity. When the planning is been made, in case that the system was drifter between 2 machines in equality of circumstances, it would choose the one that had greater capacity, being able to release the other machines to activities of engineering, for example. Clearly that this rule would be applied only after the system have allocated the machines that had used in the previous week and only in the case it need to use only one of them, then would use this decision rule.

- Trying to use the remaining capacities of machines that are already placed, but not it 100%, in order to optimize the use of the machines. For example, a machine is planned to use only 40% of its capacity; the system must then allocate the other 60% of this machine (if possible) instead of using another machine for the effect.

APPENDIX J: Prototype’s Conceptual Model

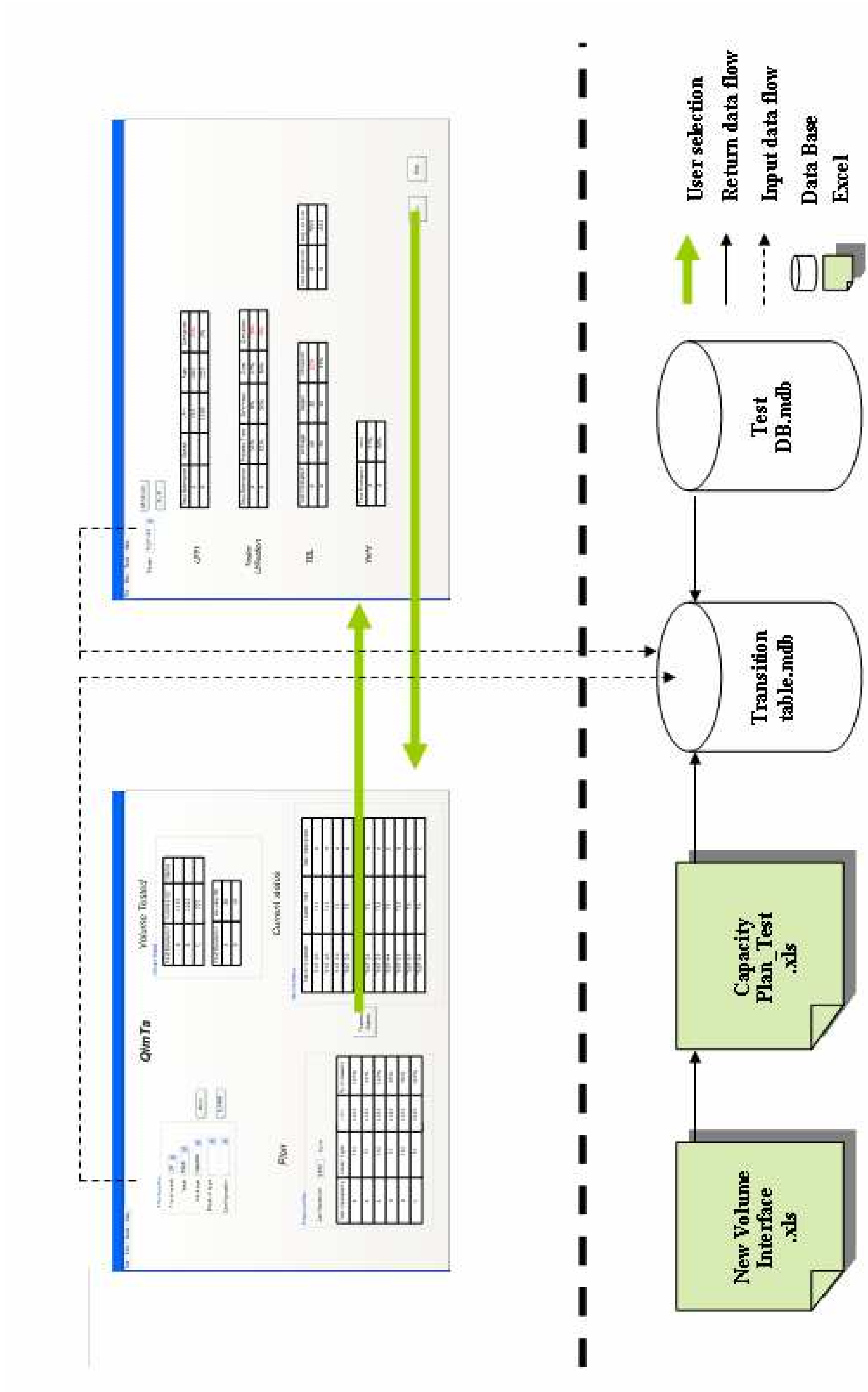

Figure J.1 - Prototype's conceptual model

APPENDIX K: Analysis of Capacity variation

This Appendix aims to explain the calculation that was done in order to study the capacity variation in the Testers.

The study had the objective to find an estimated value for the maximum variation allowed for Capacity (estimated in terms of *UPH*) giving some consistency in the functionality of the program. This value will be used in two situations:

1) To generate an alert in the program when the capacity variation is higher than the limit;

2) In a future situation, to upload the actual *UPH* values in the production to increase the efficiency and flexibility in the planning process.

The study implicates the analysis of the *UPH* absolute deviation between the actual and the planned value. To evaluate the behaviour of the *UPH* it was decided to get a sample that was statistically significant in order to obtain an accurate value. The sample size was 102 experiments.

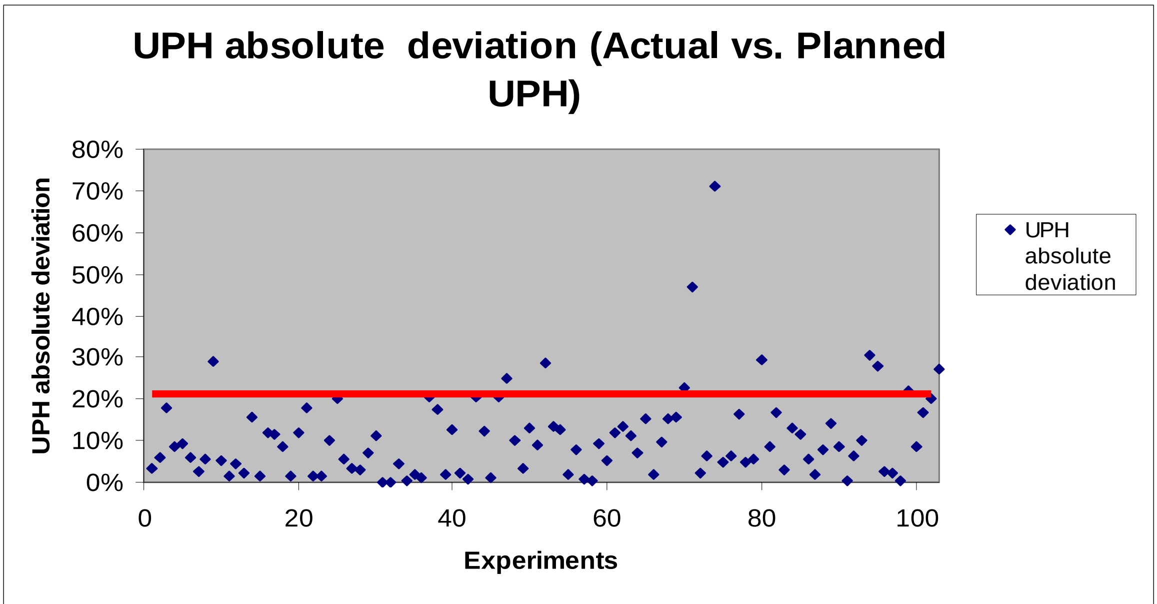

Figure K.1 –UPH absolute deviation (Actual vs. Planned UPH)

The points in Figure K.1 represent the absolute variation between the *UPH* planned and the actual UPH for all experiments. The size of the sample was enough to set a target, once the pattern of behaviour was clear.

After an evaluation of the results it was clear that there was an implicit variation in the capacity that can be explained by the complexity of the process. Therefore, it was needed to find a logical value for the Upper Control Limit (UCL) of deviation which could

define a rule for the program, always having in consideration the natural variation associated of this variable.
The value that was set as the limit for the absolute capacity variation was **20%**, as it represents 85% of the values in the sample.

During the experiment, a follow-up was made in order to evaluate the possible causes of variation. At the end, it was concluded that values lower than **20%** are usually related with the nature of the process; the majority of the values higher than this target related with external causes. Therefore, the value chosen is accurate enough to provide a correct analysis of the situation.

To summarize, the value considered as the maximum accepted value for the capacity variation is **20%** as it allows some stability in the process. The points that go over this value has to be seen as points “out-of-control”, and must to be analysed in order to detect possible problems in the process.