

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
STROJNÍ**



**BAKALÁŘSKÁ
PRÁCE**

2018

**JAN
SLABÝ**

České vysoké učení technické v Praze

Fakulta strojní

Ústav přístrojové a řídicí techniky

Monitoring skartu výrobního a balicího stroje ve firmě Philip

Morris International

Monitoring of processing and wrapping machine shredding in

Philip Morris International company

BAKALÁŘSKÁ PRÁCE

Vypracoval: Jan Slabý

Vedoucí práce: Ing. Stanislav Vrána, Ph.D.

Rok: 2018

Čestné prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně s použitím literárních zdrojů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů.

Datum:

.....

Podpis

Poděkování

Tímto bych rád poděkoval panu Ing. Stanislavu Vránovi, Ph.D. za ochotné odborné vedení mé bakalářské práce, veškerý jeho čas, trpělivost a užitečné rady, které mi pomohly při zpracování této práce. Dále děkuji společnosti Philip Morris za poskytnutí všech podkladů potřebných k vypracování bakalářské práce. V neposlední řadě bych chtěl poděkovat své rodině za podporu během celého studia.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Slabý** Jméno: **Jan** Osobní číslo: **457548**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Monitoring skartu výrobního a balícího stroje ve firmě Philip Morris International

Název bakalářské práce anglicky:

Monitoring of processing and wrapping machine shredding in Philip Morris International company

Pokyny pro vypracování:

- 1) Popište výrobního a balícího stroje a vysvětlete důvody vzniku nevyhovujících výrobků a jejich detekci,
- 2) Vybte vhodné úložiště pro data o monitorovaných výrobcích
- 3) Vytvořte aplikaci pro monitorování výrobků a důvodů jejich skartu
- 4) Navrhněte opatření pro snížení skartu

Seznam doporučené literatury:

Martinásková Marie, Šmejkal Ladislav, PLC a automatizace / Základní pojmy, úvod do programování, Praha: Ben - technická literatura, 1999. 223 stran. ISBN 80-86056-58-9
Martinásková Marie, Šmejkal Ladislav, PLC a automatizace / Sekvenční logické systémy a základy fuzzy logiky, Praha: Ben - technická literatura, 2005. 207 stran. ISBN 80-7300-087-3
David M. Kroenke, David J. Auer, Databáze, Brno : Computer Press, 2015. 496 stran. ISBN 978-80-251-4352-0

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Stanislav Vrána, Ph.D., U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

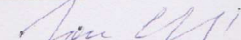
Datum zadání bakalářské práce: **18.04.2018**

Termín odevzdání bakalářské práce: **15.06.2018**

Platnost zadání bakalářské práce: _____



Ing. Stanislav Vrána, Ph.D.
podpis vedoucí(ho) práce



podpis vedoucí(ho) ústavu/katedry

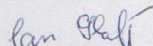


prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání



Podpis studenta

Anotační list

Jméno autora:	Jan Slabý
Název BP:	Monitoring skartu výrobního a balícího stroje ve firmě Philip Morris International
Anglický název:	Monitoring of processing and wrapping machine shredding in Philip Morris International company
Rok:	2018
Obor:	Teoretický základ strojního inženýrství
Ústav:	Ústav přístrojové a řídicí techniky
Vedoucí práce:	Ing. Stanislav Vrána, Ph.D.
Bibliografické údaje:	Počet stran 46 Počet obrázků 31 Počet tabulek 2 Počet příloh 15
Klíčová slova:	Skart, monitoring, výrobní linka, balicí stroj, výrobní stroj, relační databáze, NoSQL databáze, SSRS, Visual Studio
Keywords:	Rejected production, monitoring, production line, wrapping machine, processing machine, relational database, NoSQL database, SSRS, Visual Studio

Anotace

Tato bakalářská práce se zabývá tématem skartu na výrobní lince ve společnosti Philip Morris. V teoretické části je popsána vybraná výrobní linka, a to se zaměřením na skart. Následuje teoretický úvod do tématu relačních a NoSQL databází. Praktická část práce je pak věnována výběru úložiště dat skartu, a to vhodnějšího, než je ve společnosti aktuálně využíváno. Poté je ověřena správnost zápisu dat do databáze. V závěru praktické části jsou vytvořeny dotazy do databáze a monitorovací nástroj. Cílem práce je poskytnout skrze monitorovací nástroj validní informace vedoucím pracovníkům, kteří na ně budou v co nejkratším možném čase schopni vhodně zareagovat. Práce je úvodní částí rozsáhlejšího projektu, který bude v případě kladných výsledků dále realizován prostřednictvím rozšíření nástroje na další výrobní linky.

Abstract

This bachelor's thesis focuses on the theme of rejected pieces on the production line in the company Philip Morris. The theoretical part contains a description of the chosen production line, with special focus on rejection, which is followed by an introduction to relational and NoSQL databases. The practical part then tackles the choice of a new reject data storage method – a better suited one than that currently used by the company. The following chapter focuses on the verification of data input into the database. Finally, database queries are created, as well as the monitoring tool. The aim of the thesis is to provide valid information to supervising employees via the monitoring tool, which will enable them to react in the shortest possible amount of time. In effect, the thesis provides the basis for a larger-scale project which will, in the case of positive results, be further implemented on other production lines.

Obsah

1	Úvod	10
2	Výrobní linka.....	11
2.1	Výrobní stroj Protos 100	11
2.1.1	Rozdělovač VE	12
2.1.2	Provazcová jednotka SE.....	12
2.1.3	Část na nasazení filtrů MAX.....	13
2.1.4	Kontrolní systém	14
2.2	Výrobní stroj - skartovací místa a důvody.....	15
2.3	Řízení stroje a tok dat.....	15
2.3.1	PLC.....	15
2.3.2	Řízení a komunikace	16
2.4	Balicí stroj.....	17
2.4.1	První část - X3	17
2.4.2	Druhá a třetí část PB15,C600	19
2.5	Čtvrtá část SBNE	20
2.6	Balicí stroj - skartovací místa a důvody	21
3	Databáze.....	22
3.1	Relační databáze	22
3.2	NoSQL dokumentové databáze.....	23
4	Výběr úložiště dat.....	25
4.1	Návrh relační databáze	26
4.1.1	Popis vybraných úseků procedury LoadDataM.....	28
4.2	NoSQL Dokumentové databáze	29
4.3	Shrnutí	29
5	Tvorba monitorovacího nástroje.....	30
5.1	Validace dat.....	30

5.1.1	Validace balicího stroje	30
5.1.2	Validace výrobního stroje.....	34
5.2	Monitorovací nástroj – tvorba	35
5.2.1	Dotazy.....	36
5.2.2	SSRS	37
6	Využití monitorovacího nástroje	39
7	Závěr	41
8	Reference	43
	Seznam obrázků	44
	Seznam tabulek	45
	Seznam příloh.....	45

1 Úvod

Tato bakalářská práce se zabývá tvorbou monitorovacího nástroje skartované produkce na cigaretové výrobní lince v kutnohorské továrně společnosti Philip Morris. Historie tabákového průmyslu je s Kutnou Horou spjata už od roku 1812, kdy se zde vyráběl dýmkový a šňupací tabák. V roce 1844 byla zahájena výroba doutníků a následně v roce 1882 zde byly vyrobeny první cigarety. Šlo však pouze o ruční výrobu s roční kapacitou pět milionů kusů. V roce 1896 byla zavedena strojní výroba, která je stále modernizována a v dnešní době továrna disponuje kapacitou kolem 40 miliard kusů cigaret ročně. V roce 1992 se stala společnost Philip Morris International většinovým podílníkem. [1] [2]

V současné době je kladen stále větší tlak na efektivitu výrobních procesů, a to ve všech průmyslových odvětvích. Této vyšší efektivitě lze dosáhnout nejen pořízením výkonnějších zařízení, ale například i pomocí správného nastavení strojů a rychlou indikací vyskytujících se problémů a jejich následným odstraněním. V této bakalářské práci se zabývám právě indikací problémů a monitorováním stroje obecně. Přesněji se zaměřuji na ukládání dat o skartu a následném využití těchto dat k indikaci problémů a tím pádem k zefektivnění výroby.

Tato práce by měla v první řadě sloužit k získání základního přehledu o výrobní lince, možnostech ukládání dat a seznámení se s databází obsahující data o skartu. Tyto nabyté znalosti následně využiji při návrhu vhodnějšího úložiště dat o skartu a při tvorbě nástroje na sledování skartu. Tento nástroj by měl sloužit vedoucím pracovníkům ve výrobě jako indikátor nastávajících problémů, na které budou díky tomuto nástroji schopni v krátkém čase zareagovat a odstranit jejich příčiny. Dále by měl poskytovat přehledné informace o možných problémech, které by se mohly vyskytnout v budoucnu a tím pádem umožnit jejich předejití.

V teoretické části této práce představím zvolenou výrobní linku se zaměřením na skartovací místa, způsob napojení stroje na databázi a databázi samotnou. Dále porovnáím možnosti využití relačních a NoSQL dokumentových databází. V praktické části navrhuji možné způsoby ukládání dat a zhodnotím, jaké úložiště by pro data o skartu bylo nejvhodnější. Dále připravím sadu dotazů do databáze, které budou využity při tvorbě monitorovacího nástroje, jehož tvorba a popis funkce budou poslední částí této práce.

2 Výrobní linka

Tato část práce slouží k vysvětlení základních funkcí cigaretové výrobní linky se zaměřením na skartovací místa a důvody skartu. Skartovací místa jsou části linky, ve kterých je po detekci vady výrobku daný výrobek automaticky vyveden ze stroje. Manuální skartaci se v této práci nebudou nevěnovat, ale zahrnu ji do celkového procenta skartu ve výpočtu v následujících kapitolách.

Daná výrobní linka se skládá z výrobního a balícího stroje, které jsou dále rozděleny na menší celky a v následujících kapitolách popsány. Tuto konkrétní výrobní linku jsem vybral po diskuzi s Lean Improvement Inženýrem zaměřeným na skart. Prvním kritériem bylo vyšší využití výrobní linky. Druhým kritériem byl vyšší skart na výrobní lince. Posledním kritériem bylo vybrat linku bez možnosti křížové výroby (kombinace dvou výrobních linek), a to z důvodu snazšího přiřazení výrobních zakázek.

2.1 Výrobní stroj Protos 100

Pro tuto práci jsem zvolil výrobní linku s výrobním strojem od německé společnosti Hauni, která spadá do holdingu Körber AG. Společnost Hauni se zabývá jak výrobou zařízení pro produkci cigaret, tak i strojů na zpracování tabáku a zařízení na výrobu filtrů. [3]

Tento výrobní stroj je schopen vyrobit až 10 000 cigaret za minutu. [4] Je možné ho rozdělit na tři základní části: rozdělovač VE, provazcová jednotka SE a část na nasazení filtrů MAX. [5]



Obrázek 1: Výrobní stroj (ilustrační), zdroj: <https://www.hauni.com/en/nc/service/services/modernization-and-tip/filter-cigarette-maker/detail/product/protos-100-1.html>

2.1.1 Rozdělovač VE

V této části stroje nedochází ke skartaci výrobků, z tohoto důvodu tuto část popisují jen stručně pro vytvoření představy o funkci celého zařízení.

První část výrobního stroje slouží k separaci nevhodného tabáku a dávkování správného množství pro následující část. Do této části je dopravním potrubím přiveden řezaný tabák k třídícímu bubnu. Množství přiváděného řezaného tabáku se reguluje pomocí podtlaku, který je tvořen sacím potrubím napojeným také na třídící buben. Z třídícího bubnu se tabák dostává do tabákového zásobníku přes turniketový uzávěr, který plní funkci usměrnění a dávkování. V zásobníku tabáku pod turniketovým uzávěrem je umístěn optický senzor, který k sacímu potrubí pomocí programovatelného automatu (více v kapitole 2.3.1) vysílá informaci o přítomnosti tabáku. [6]

Optickým senzorem je v tomto případě myšlen optický reflexní senzor. Tyto senzory patří mezi nejpoužívanější snímače pro zjištění přítomnosti objektů. Pracují na principu detekce existence nebo měření intenzity paprsku světla dopadajícího na přijímací část senzoru. U reflexních senzorů jsou zdroj paprsku světla i detektor, tedy přijímací část, umístěny v jednom pouzdru. Senzor detekuje množství odraženého světla a porovnává se s požadovanou hodnotou. Jako přijímač může sloužit například fotodioda nebo fototranzistor. [7]

Ze zásobníku se tabák přepravuje po pásu šikmého dopravníku do násypné šachty. Případná cizí tělesa jsou odstraněna magnetickou lištou. Plnicí válec odebírá tabák z násypné šachty a vyčesávací válec vyčesává tabák z plnicího válce do spádové šachty. Ze spádové šachty je tabák se správnou hmotností směřován k tryskovému žlábkem profukovacím vzduchem a těžší částičky propadávají do nádoby na odpadový tabák. Řezaný tabák je veden tryskovým žlábkem k následující části výrobního stroje. [6]

2.1.2 Provazcová jednotka SE

V této části stroje, stejně jako v té předchozí, nedochází ke skartaci výrobků, z tohoto důvodu tuto část popisují jen stručně pro vytvoření představy o funkci celého zařízení.

Z tryskového žlábkem předřazeného stroje je tabák přiveden na přísávací dopravník. Přísávací dopravník přísaje tabákový provazec pomocí sání vzduchu k přísávacímu pásu. Ten přivádí tabákový provazec přes dva oddělovače do formátovací soustavy. Oddělovač se skládá ze tří kotoučů, které oddělují část tabáku z provazce tak, aby se do formátovací soustavy dostalo jen potřebné a rovnoměrně rozložené množství tabáku. Oddělená část provazce je dopravována zpět do zásobníku tabáku z kapitoly 2.1.1. Ve formátovací soustavě se tabákový provazec obaluje cigaretovým papírem, následně se potře lepidlem a slepí se. Cigaretový papír je do formátovací soustavy dopravován pomocí několika válečků, které zajišťují optimální rychlost odvíjení a zároveň umožňují vystředění vůči razítku, které nanáší potisk na cigaretový papír. Lepidlo je do formátovací soustavy dopravováno čerpadlem ze zásobníku a následně je tryskou nanášeno na okraj cigaretového papíru. Po nanesení lepidla se část papíru s naneseným lepidlem přehne, cigaretový provazec se uzavře a vzniká nekonečná cigaretová tyčinka. Již zformovaný cigaretový provazec postupuje dále do žehličky švu, která má za úkol usušit nanášené lepidlo. Po průchodu žehličkou švu je nekonečná tabáková tyčinka řezána na dvojnásobně dlouhé

tabákové tyčinky, které jsou předávacími rameny se sacími žlábkami předávány na přejímací buben následující části stroje. [5], [6]

2.1.3 Část na nasazení filtrů MAX

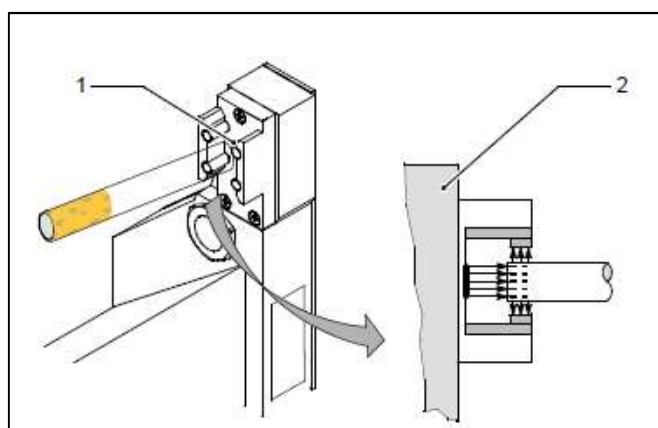
V této části výrobního stroje se k již zformovaným cigaretovým tyčinkám přidává cigaretový filtr a kontroluje se kvalita vyrobených cigaret.

Přejímací buben předává tabákové tyčinky řezacímu bubnu, na kterém dojde k rozříznutí tyčinek na dvě stejně dlouhé části. Rozříznuté tabákové tyčinky putují na separační buben, který tyčinky roztáhne tak, aby mezi ně bylo možné vložit dvojitý filtr. Ze separačního bubnu se tyčinky dostávají na přiváděcí buben, který již nese dvojitý filtr. Filtr se na přiváděcí buben dostane ze zásobníku filtrů a v případě absence filtru, která je detekována optickým senzorem, je dvojitá tabáková tyčinka sfouknuta k sobě a na následujícím bubnu je vyhozena vyhozovacím ventilem do skartovací nádoby. Zmíněný optický senzor je stejného typu jako již zmíněný senzor v části 2.1.1. Na přiváděcím bubnu se také skartují tabákové tyčinky s chybnou hmotností, která byla zjištěna již v provazcové části stroje. [6] Hmotnost tabákového provazce je zjišťována kontrolním systémem MP5. Tento systém je poměrně komplikovaný, ale pro zjednodušení vysvětlení uvažují jeho složení pouze ze tří částí. Jedním PLC (více k PLC v kapitole 2.3.1) a dvěma snímači, jedním teplotním a druhým mikrovlnným. Teplotní snímač je předřazený a získává informaci o teplotě tabákového provazce, tato informace je nutná k správné funkci mikrovlnného snímače. Mikrovlnný snímač vysílá skrze proudící tabákový provazec signál k přijímači a na základě změny rychlosti signálu je měřena hustota. Tyto informace jsou zpracovávány PLC, které ovládá rychlost tabákového provazce. Jelikož se neměří jedna konkrétní tabáková tyčinka, ale měření probíhá kontinuálně, je v případě nižší nebo vyšší hustoty/váhy vyskartováno větší množství cigaret. [8] Z přiváděcího bubnu je dvojitý filtr s tabákovými tyčinkami veden na ovinovací buben, na kterém se filtr a tabákové tyčinky slepí obalovým papírem. Po slepení se do skartovací nádoby vyfukují špatně slepené dvojité cigarety. Následuje další předávací buben, který předá dvojité cigarety řezacímu bubnu, na kterém jsou rozděleny na jednotlivé. Následuje buben předávající jednotlivé cigarety obracecímu bubnu. Obracecí buben otočí každou druhou cigaretu, tím pádem mají všechny cigarety stejnou polohu pro následné zpracování. Následuje skupina čtyř bubnů, která dopraví cigarety na pás vedoucí k následnému stroji. Na těchto bubnech jsou skartovány chybné cigarety, které identifikuje kontrolní systém CIS (Cigarette Inspection System). Kontroluje se celková ventilace, hustota tabáku, tvar a chybějící filtr. Tuto část podrobněji popíší v následující kapitole. Dále je optickým senzorem kontrolován počet vyrobených cigaret. [5], [6]

2.1.4 Kontrolní systém

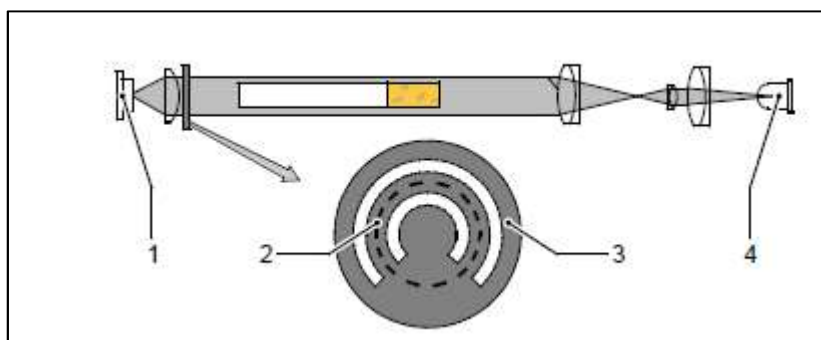
Úloha kontrolního systému je vyhazování těch cigaret, které neodpovídají dané kvalitativní normě.

Kontrolují se následující kvalitativní znaky: měření hustoty v hlavě cigarety, rozpoznání chyb tvaru cigaret, měření celkové ventilace. Kontrolní systém sestává z počítače kontrolního systému, který přijímá signály od kontrolních snímačů a následně komunikuje s programovatelným automatem a centrálním ovládacím přístrojem MLP, kterým je i ovládán (více o MLP v kapitole 2.3.2). Podrobněji popisují pouze kontrolu hustoty tabáku a tvaru cigaret. Kontrola hustoty tabáku v zapalovaném konci cigarety LES (Loose Ends) je prováděna optickým snímačem. Infračervené světlo senzoru svítí na čelní stranu cigarety, čtyři senzory uspořádané do čtverce okolo procházející cigarety akumulují světlo, které proniká kolmo skrz cigaretový papír. Množství světla, které se odráží od hlavy cigarety se přemění na odpovídající elektrické napětí. Pokud je hodnota elektrického napětí nižší než přednastavená spodní hodnota, počítač kontrolního systému vyhodnotí cigaretu jako vadnou.



Obrázek 2: Měření hustoty v zapalovaném konci, vysvětlivky: 1 = senzor LES, 2 = přijímače, zdroj: Interní materiál 4

Kontrola tvaru cigarety OTIS (Optical Tipped Cigarette Inspection System) je prováděna světelným snímačem, který se skládá z vysílače a přijímače. Světelný paprsek vysílače je pomocí optiky paralelně veden nad cigaretou a skrze nepřesnosti přechází k přijímači. Tento senzor pracuje na stejném principu jako optický senzor popsáný v kapitole 2.1.1., pouze s tím rozdílem, že přijímač je umístěn proti vysílači. Čím je ztmavení větší, tím větší je tvarová chyba. Množství světla dopadajícího na přijímač se v měřicí hlavě přemění na odpovídající napětí a stejně jako v předchozím případě je systémem vyhodnoceno buď jako vyhovující nebo nevhovující.



Obrázek 3: Kontrola tvaru cigarety, vysvětlivky: 1 = přijímač, 2 = požadovaný průměr cigarety, 3 = clona s otvory, 4 = vysílač,
zdroj: Interní materiál 4

Množství světla dopadajícího na přijímač je ovlivněno zakrytím otvorů na cloně. [9]

2.2 Výrobní stroj - skartovací místa a důvody

Tato kapitola slouží k shrnutí všech skartovacích míst a důvodů skartu na výrobním stroji. Výrobní stroj disponuje třemi skartovacími místy. Prvním místem je přiváděcí buben, na kterém se skartuje na základě hmotnosti. Dalším skartovacím místem je ovinovací buben, na kterém se skartují chybně slepené cigarety a cigarety s chybějícím filtrem. Posledním je vyhadzovací buben, kde se skartují cigarety vyhodnocené kontrolním systémem jako chybné. Kontrolní systém vyhodnocuje tvar cigarety, hustotu tabáku a celkovou ventilaci cigarety. Dále je na vyhadzovacím bubnu znovu kontrolována přítomnost filtru. [6]

2.3 Řízení stroje a tok dat

V této části textu popisují komunikaci jednotlivých kontrolních systémů, PLC a dalších částí stroje se zaměřením pouze na skart.

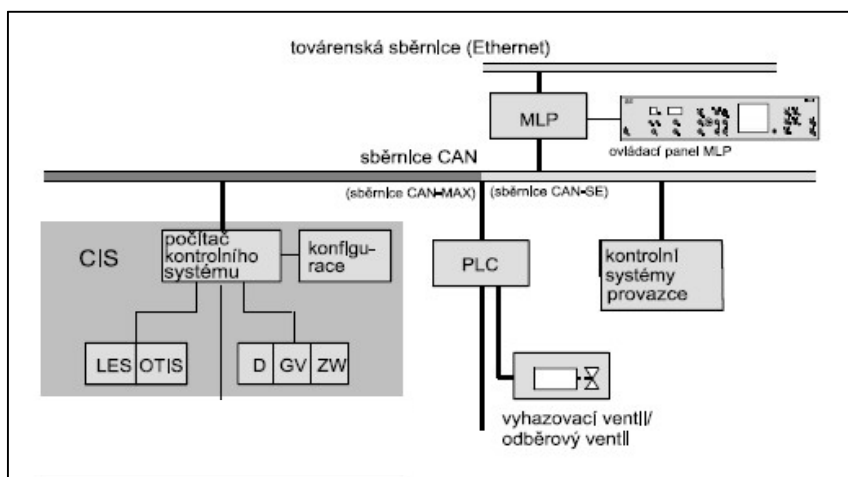
2.3.1 PLC

PLC (Programmable Logic Controller) je „uživatelsky programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů“. [10] Použitím programovatelného automatu v řídicím systému může být docíleno automatizace buď částí nebo celého procesu. Řídicím systémem je zde myšlen komplexní celek celé výrobní linky. Tento automat je schopen přijímat a zpracovávat informace, například z výše zmíněných optických senzorů, a na jejich základě, dle programu/nastavení automatu vysílat příkazy k dalším částem linky (akčním členům), například motorům jednotlivých pásových dopravníků. Reakce na vstupy, respektive na změny vstupů je v PLC vyhodnocována, do jeho paměti vloženým programem. Rozlišují se dva základní typy řízení. Dopředným řízením působí PLC na daný objekt jednosměrně. Příkladem dopředného řízení může být spojení optického senzoru, PLC a

vyhazovacího ventilu. V případě, že optický senzor detekuje vadný výrobek, je vyslán signál do PLC, které je naprogramované tak, aby v tomto případě vyslalo signál vyhazovacímu ventilu a ten daný výrobek vyfoukl ven z linky. Při zpětnovazebním řízení je navíc sledován i předchozí stav řízeného objektu. Příkladem může být část stroje, ve které je kontrolována hustota přiváděného tabákového provazce. Pokud PLC dostane informaci o nižší hustotě, zvýší rychlost přiváděného provazce, ale to na základě aktuální rychlosti. Uvedený příklad dopředného řízení je na sledované lince realizován řízením zpětnovazebním, a to z důvodu počítání kusů vyskartovaných vadných výrobků. V případě, že je vadných výrobků v určeném časovém intervalu více než je dovoleno, PLC vyšle signál k zastavení stroje. [10], [11]

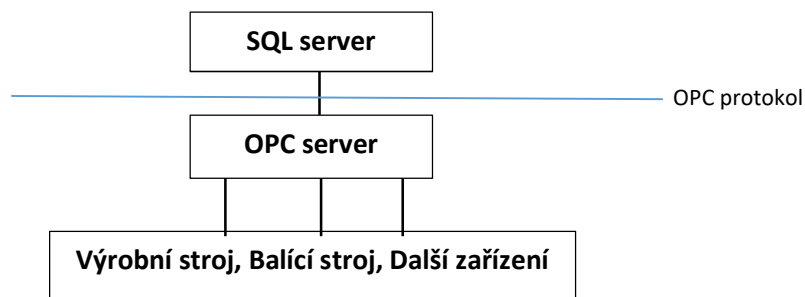
2.3.2 Řízení a komunikace

Jednotlivé snímače vysílají signály do PLC buď přímo, nebo skrze počítač kontrolního systému CIS, popřípadě PM5. PLC vysílá signály k vyhazovacím ventilům, motorům a dalším akčním členům. Tyto informace jsou také přijímány MLP (Making Line Procesor). MLP je centrální ovládací a indikační přístroj s vizualizací procesů. Je nadřazeným ovládacím přístrojem pro všechna účastnická zařízení. Jeho úkolem je například obsluha kontrolních systémů a zobrazování poruch. MLP umožňuje skrze uživatelské rozhraní ovládnutí stroje, PLC i kontrolních systémů. Po továrenské sběrnici může MLP komunikovat s nadřazeným centrálním počítačem. [12]



Obrázek 4: Struktura systému, zdroj: Interní materiál 5

Vybraná data jsou dále z MLP přes OPC server ukládána do databáze. OPC server umožňuje ukládání dat z různých zařízení, například PLC do jedné databáze. Jen na výše popsané lince jsou data o výrobním stroji získávána z MLP a data o balícím stroji přímo z jeho PLC. OPC server převádí data získaná z jednotlivých zařízení do formátu OPC a následně je v tomto standardizovaném formátu poskytuje SQL serveru. Tím odpadá problém různých komunikačních protokolů. [13]



Obrázek 5: Schéma, zdroj: vlastní

2.4 Balicí stroj

U sledované výrobní linky je balicí stroj tvořen ze čtyř částí. Z toho první tři části jsou výrobky firmy G.D a poslední část od společnosti Sensani. Společnost G.D je předním světovým dodavatelem pro cigaretový průmysl. Společnost Sensani se zabývá výrobou balicích strojů pro potravinářský průmysl. [14],[15]

V následujících odstavcích stručně popisují jednotlivé části a jejich skartovací místa. Podrobněji však rozeberu pouze část, ve které se cigarety skládají do krabiček, a to z důvodu nejvyššího podílu na skartované produkci. Rychlosti jednotlivých částí stroje jsou shodné s rychlostí výrobního stroje, to zaručuje kontinuální chod bez větších zastávek. Optické detektory použité k detekci vadných výrobků jsou typově shodné s těmi popsány v části výrobního stroje. Z tohoto důvodu je již v této části nebudu samostatně popisovat. [14]

2.4.1 První část - X3

V této části výrobní linky se cigarety balí do krabiček. V závislosti na konfiguraci stroje mohou být krabičky naplněny různým počtem cigaret o různých délkách.

Cigarety jsou do balicího stroje dopravovány pomocí pásového dopravníku, který propojuje výrobní a balicí stroj. Výrobní linka je také vybavena zásobníkem cigaret, který je spojen s dopravníkem mezi výrobním a balicím strojem. Do zásobníku cigaret jsou cigarety dopravovány z výrobního stroje ve dvou případech. Prvním případem je zastavení balicího stroje například z důvodu poruchy. Druhým případem je vyšší rychlost výrobního stroje proti rychlosti balicího stroje. Cigarety jsou ze zásobníku dopravovány do balicího stroje v opačných případech. Jelikož se zde nedetekuje žádná vadná produkce, tak tento zásobník nebudu dále popisovat. Tok cigaret je dopravován do násypky stroje, ze které jsou pomocí vyrážeců dopravovány do kapes kapsového dopravníku. Zde je skupinou optických senzorů kontrolována přítomnost cigaret. V případě chybějící cigarety je vyskartována celá kapsa do první skartovací nádoby. Kapsový dopravník přenáší kapsy s cigaretami na první kolo. Během transportu jsou kapsy kontrolovány skupinou snímačů. První kontrolní místo zjišťuje správné narovnání

cigaret v kapsách, a to z boku a ze spodu kapsy. Na druhém kontrolním místě se zjišťuje hustota tabáku v cigaretách. Na třetím kontrolním místě se kontroluje přítomnost cigaretového filtru. Na konci kapsového dopravníku se ještě jednou zkontroluje přítomnost všech cigaret v kapsách. Pokud je detekována alespoň jedna z výše zmíněných výrobních nepřesností, tak jsou cigarety z kapsy vyskartovány ještě před vstupem do prvního kola. Vyskartované cigarety putují pásovým dopravníkem do první skartovací nádoby. Po průchodu kapsovým dopravníkem jsou cigarety přemístěny vyrážecím do prvního kola. Prvním kolem jsou cigarety dopraveny k druhému kolu. K druhému kolu je zároveň s cigaretami přiváděna alufólie, která je odvíjena z vřetena a prochází přes přelepovací jednotku, vodící válce a skupinu nožů. V případě skartace kapsy cigaret na kapsovém dopravníku je fólie též vyskartována a pásovým dopravníkem putuje do první skartovací nádoby. Během otáčení druhého kola je fólie kolem cigaret zformována zahýbači. Zformovaný balíček je pomocí vyrážeců dopraven ke třetímu kolu. Optickým snímačem je zde kontrolováno správné složení fólie kolem cigaret. V případě zjištěného problému je balíček vyskartován a pásovým dopravníkem putuje do první skartovací nádoby. Třetí kolo přepravuje balíček cigaret ke kolu čtvrtému a optická kontrola sleduje tvar balíčku cigaret a přítomnost odtrhovací části fólie. V případě neshody je balíček vyskartován a pásovým dopravníkem veden do první skartovací nádoby. Na čtvrtém kole je k balíčku cigaret přidáván krček, který má za úkol držet víčko krabičky zavřené. Krček je do čtvrtého kola veden skupinou válečků a finální tvar krčku je zajištěn tvarovými noži. Nad čtvrtým kolem je umístěna optická kontrola, která v případě neshody vyšle signál do řídicí jednotky, která vydá příkaz ke skartaci na šestém kole. Veškerá vyřazená skartace na šestém kole je skluzavkou směřována do druhé skartovací nádoby. Čtvrté kolo je dále vybaveno dvojicí optických detektorů, které kontrolují přítomnost a kvalitu balíčku. Stejně jako v předchozím případě je po detekci závady vyslán signál ke skartaci balíčku na šestém kole. K pátému kolu je spolu s balíčkem cigaret přiváděn výsek krabičky, který je zde formován kolem balíčku cigaret. Sloupce výseků jsou do stroje dopravovány pomocí ramena s přísavkami. Následně jsou jednotlivé výseky dopravovány k pátému kolu pomocí dopravníků. Cestou k pátému kolu je na výseky nanášena slabá vrstva lepidla a dále je cesta kontrolována optickými detektory, které sledují přítomnost výseků. V případě nepřítomnosti výseku je balíček cigaret vyskartován na šestém kole. Na pátém kole je optickými detektory kontrolována přítomnost balíčku, kontrola celistvosti balení, kontrola přítomnosti krčku. Šesté kolo slouží k dopravě již zformovaných balíčků cigaret do dosoušecího dopravníku a ke skartaci vadných výrobků. Dosoušecí dopravník umožňuje dokonalé dosušení lepených ploch krabičky a dále posílá krabičky do dopravního kanálu, který nese krabičky k následné části balicího stroje. [17], [18]



Obrázek 6: X3 (ilustrační), zdroj: <http://www.gidi.it/en/solutions/product/x3>

2.4.2 Druhá a třetí část PB15, C600

Druhou částí stroje je zásobník cigaretových krabiček, který je spojen s dopravníkem krabiček vyvedeným z první části. Jeho plnění a vyprazdňování pracuje na podobném principu jako zásobník cigaret zmíněný v části 2.4.1 a při průchodu krabiček tímto zásobníkem nevzniká ani není skartována žádná vadná produkce. Z tohoto důvodu nebudu zásobník krabiček dále popisovat.

Do třetí části stroje jsou krabičky dopravovány buď z první nebo druhé části stroje skupinou pásových dopravníků. Na krabičku je zde nanesen kolek, který je kontrolován optickým senzorem a v případě detekované chyby je krabička vyskartována. Dále je krabička obalována do fólie a následně jsou skupiny krabiček obalovány do kartonové fólie. Správnost složení těchto fólií je kontrolována optickými senzory a v případě detekce chyby je výrobek skartován. Krabičková a následně i kartonová fólie je zatavena v zatavovací stanici a karton je pásovým dopravníkem dopravován k následnému stroji.

Tuto část jsem popsal pouze stručně z toho důvodu, že skartovanou produkci v této části stroje dále nesleduji, přestože je zde produkována. To je zapříčiněno tím, že je možné vyskartované produkty, a to jak krabičky bez kolků, krabičky nesprávně obalené fólií i kartony nepřesně složené, vrátit zpět do stroje a znovu zpracovat. Skartovanou produkci na této části stroje zahrnu v praktické části této práce pouze jako rozdíl mezi dobrou produkcí předřadného stroje a dobrou produkcí této části. [19]



Obrázek 7: C600 (ilustrační), zdroj: <http://www.gidi.it/en/solutions/product/c600-pack>

2.5 Čtvrtá část SBNE

V této části jsou kartony vyprodukované předchozí částí baleny do krabic. Z předřazené části se kartony dostávají pásovým dopravníkem do stohovače kartonů, ve kterém se kartony rovnají na sebe dle předem nastaveného počtu. Poté jsou kartony vsunuty do krabice, která je ramenem s přísavkami odebrána ze zásobníku krabic a rozložena. Po naplnění je krabice mechanicky vertikálně vedena k výstupnímu dopravníku. Cestou se postupně skládají horní a spodní víka krabice, která jsou na konci transportu přelepena lepicí páskou. Transportním dopravníkem jsou krabice vedeny do skladu. Při rovnání kartonů do krabic jsou obsazená místa v krabici kontrolována optickými snímači. Z této části stroje nejsou v současné době informace o skartu posílány dále do databáze. Proto v následujících částech práce nebudu tuto část stroje dále uvažovat. Za finální produkci bude považována produkce předřazené části výrobní linky. [20]



Obrázek 8: SBNE (ilustrační), zdroj: <http://www.senzani.com/en/sector/tobacco/bag>

2.6 Balicí stroj - skartovací místa a důvody

Tato kapitola slouží pro shrnutí skartovacích míst a důvodů skartu, která jsou sledována na balicím stroji. V dalších částech této práce budu pracovat pouze s nevyhovující produkcí první části balicího stroje. Skartovanou produkci ostatních částí budu počítat pouze rozdílem vyhovující produkce jednotlivých částí. Z tohoto důvodu v této kapitole uvedu souhrn skartovacích míst a důvodů pouze pro první část stroje.

Balicí stroj disponuje čtyřmi skartovacími místy. První skartovací místo se nachází pod násypkou cigaret. Zde se kontroluje přítomnost všech cigaret přepravovaných na kapsový dopravník. Druhé skartovací místo se nachází na konci kapsového dopravníku a jsou zde vyřazovány chybné kapsy detekované senzory na dopravníku. Sleduje se zde správné narovnání cigaret v kapsách, hustota tabáku v cigaretách (prázdné konce) a přítomnost cigaretového filtru. Dalším skartovacím místem je třetí kolo, kde se vyřazuje alufólie v případě již dříve vyskartované kapsy, správné složení fólie kolem cigaret, přítomnost odtrhací části fólie a správný tvar balíčku. Dále je zde kontrolována zácpa nebo uvíznutí materiálu mezi druhým a třetím kolem. Vyřazená produkce z těchto stanovišť je dopravována

do první skartovací nádoby. Čtvrtým skartovacím místem je šesté kolo, na kterém se skartuje z důvodů chybějícího nebo špatně položeného krčku, chybějícího výseku krabičky, špatně umístěné alufólie a tvaru balíčku. Dále je zde kontrolována zácpa mezi pátým a šestým kolem. V případě zastavení stroje na delší než dovolenou dobu jsou při následném rozjezdu vyskartovány balíčky se zaschlým lepidlem a v případě, že se dostane motor lepení mimo fázi, jsou vyskartovány všechny balíčky do doby, než se dostane zpět do fáze. [17], [18]

Princip čtení a zápisu je realizován obdobně jako u výrobního stroje pouze s tím rozdílem, že OPC server čte přímo z PLC.

3 Databáze

V této části budu popisovat výhody a nevýhody relačních a NoSQL databázových uložišť. Zaměřím se pouze na relační databázi Microsoft SQL serveru a dokumentové NoSQL databáze MongoDB. V následující kapitole zhodnotím jejich možné využití k ukládání dat o skartu z výše popsané výrobní linky.

3.1 Relační databáze

Relační databáze sestává z více tabulek, které se skládají z atributů (sloupců) a informací o entitách na řádcích. Jako entitu si lze představit například stroj a jeho atributy čas skartu, místo skartu, počet skartovaných výrobků. Aby mohla být tabulka považována za relační, musí splňovat určité podmínky, například všechny sloupce musí mít jedinečný název, buňky musí obsahovat jedinou hodnotu nebo v jednom sloupci musí být použit pouze jeden datový typ. Jednotlivé relační tabulky jsou propojeny klíči. Tyto klíče umožňují spojení relačních tabulek. Rozlišujeme klíče primární a cizí. Primární klíč musí být jedinečný. To znamená, že se v daném sloupci může vyskytnout pouze jednou. Cizí klíč je součástí druhé relační tabulky. Hodnoty cizího klíče již nemusí být jedinečné, ale lze například vynutit možnost zadávání cizího klíče na základě hodnot primárního klíče v druhé tabulce, a to následovně: pokud sloupec primárního klíče obsahuje hodnoty 1-10, nelze jako cizí klíč uvést hodnotu 11. Pomocí těchto klíčů je možné tabulky propojit. Použití klíčů bude demonstrováno v následujících částech. Se systémem řízení dat uživatel komunikuje pomocí jazyka SQL. Tento jazyk má spoustu různých variant v závislosti na jednotlivých systémech řízení dat. Například Microsoft SQL server využívá jazyk T-SQL. Tyto různé obdoby jazyka SQL mají stejný základ, liší se pouze až u pokročilejších příkazů či funkcí. To umožňuje celkem hladký přechod mezi jednotlivými systémy relačních databází. Tabulky v relačních databázích jsou tvořeny jako co nejmenší celky, které lze propojovat klíči. To znamená, že různé informace jsou v různých tabulkách, ale klíče přesně definují, které hodnoty patří k sobě, což umožňuje jejich snadné propojení. [21]

Jako výhody relačních databází vnímám jejich rozšířenost a množství literatury k nim sepsané. Například NoSQL databáze nemají takovou tradici, a z tohoto důvodu neexistuje tolik materiálů ke studiu jako u relačních databází. Tento problém je však stále menší, protože NoSQL databáze se v posledních letech využívají stále častěji a již nyní existuje početná komunita uživatelů. Další výhodou relačních databází je právě jazyk SQL, který je velice jednoduchý a lze ho použít u všech nejčastěji používaných systémů řízení dat. Jako poslední výhodu zmíním přehlednost relačních databází, kde každá tabulka nese pouze určitou část informací a je velice snadné jednotlivé tabulky propojit dle požadovaných informací. To však může být i nevýhodou například při potřebě rozšířit tabulku o další atributy. To vede často k nutnosti změny schématu celé nebo alespoň části databáze. Další nevýhodu spatřuji ve zpracovávání většího množství dat, kdy například NoSQL databáze dokáží pracovat rychleji.

3.2 NoSQL dokumentové databáze

V současné době se rozvíjejí nové databázové systémy nazývané NoSQL (nebo také Not only SQL), které podporují efektivní zpracování Big Data. Databázové systémy s NoSQL patří často mezi nerelační databázové systémy nazývané strukturované úložiště. Zatímco relační databáze nabízejí přístup pomocí SQL, NoSQL databáze využívají rozdílné jazyky. Tyto jazyky jsou mnohdy komplikovanější než jazyk SQL. NoSQL databáze využívají distribuované datové zpracování. Distribuovaná databáze je rozdělena mezi více serverů, to se používá ze dvou důvodů - výkonu a kontroly. Tímto způsobem lze rozdělit pracovní zátěž mezi tyto servery. NoSQL systémy lze rozdělit do čtyř základních typů. Databáze typu klíč-hodnota, dokumentové databáze, sloupcové databáze a grafové databáze. Typů NoSQL databází existuje více, tyto jsou však nejrozšířenější. Podrobněji popíši pouze dokumentové databáze systému MongoDB, nejpoužívanějšího databázového systému tohoto druhu. Dokumentové databáze využívají k ukládání dat dokumenty, které jsou také využívány například ke komunikaci s aplikacemi. Dokumenty mohou obsahovat různé hodnoty různých typů, proti relační databázi odpadá potřeba přidávání dalších sloupců a změna schématu databáze v případě výskytu nových atributů.

Na obrázku 9 jsou dva dokumenty dokumentové databáze, které dále obsahují vnořené dokumenty adres a profile. Pokud by bylo uložení těchto záznamů realizováno v relační databázi, bylo by třeba vytvořit více sloupců, z nichž by některé obsahovaly prázdné části řádků.


```

{
  "login": "honza",
  "firstname": "Jan",
  "surname": "Novák",
  "address": {
    "city": "Praha",
    "street": "Krásná 5",
    "zip": "111 00"
  }
}

{
  "login": "janicka",
  "firstname": "Jana",
  "surname": "Novotná",
  "web": "http://janicka.novotna.cz/",
  "profile": {
    "colorschema": "green",
    "design": "simple"
  }
}

```

Obrázek 9: Dokumenty, Zdroj: Big Data a NoSQL databáze

Alternativou k vnořeným dokumentům je ukládání dokumentů odděleně a jejich provázání pomocí odkazů, které fungují na podobném principu jako klíče v relační databázi. Toto schéma je využíváno v případech, ve kterých by vnořené dokumenty vedly k duplicitnímu ukládání dat.

```

kolekce "users"
{
  "_id": <ObjectId>,
  "login": "honza",
  "name": "Jan Novák"
}

kolekce "addresses"
{
  "_id": <ObjectId2>,
  "user_id": <ObjectId1>,
  "city": "Praha",
  "street": "Krásná 5",
  "zip": "111 00"
}

kolekce "profiles"
{
  "_id": <ObjectId3>,
  "user_id": <ObjectId1>,
  "colorschema": "green",
  "design": "simple"
}

```

Obrázek 10: Dokumenty s odkazy, Zdroj: Big Data a NoSQL databáze

Komunikace s databází funguje na podobném principu jako u SQL databází. Se systémem MongoDB (například přes jeho konzoli) se komunikuje pomocí jazyka odvozeného z formátu JSON (pro účely této

práce není nutné znát podrobnosti o tomto jazyku/formátu, z tohoto důvodu je nebudu dále uvádět). Dotazování databáze je v dokumentových databázích snazší z pohledu rozsahu dotazu. Na druhou stranu je nevýhodou komplikovanější dotazovací jazyk. Další výhodou dokumentové databáze je rychlost zobrazování dotazovaných dat, ale to pouze v případě správného návrhu dané databáze a správného použití indexů. Zpravidla platí, že tvorba dokumentové databáze je daleko pracnější než té relační, ale výsledkem může být následně snazší a rychlejší vytěžování dat. [21], [22]

Nejzásadnější výhodou NoSQL databází je jejich horizontální škálovatelnost. V případě relačních databází je možná pouze vertikální škálovatelnost, kterou je myšleno například navýšení výkonu fyzického serveru přidáním paměti či výměnou celého serveru za výkonnější. To však může být finančně neúnosné. NoSQL databáze mohou využít i horizontální škálovatelnosti, a to přidáním dalších serverů, mezi které se data rozdělí. Tyto servery mezi sebou komunikují, a tím pádem je lze dotazovat najednou a získat jeden výsledek. Tato varianta může být finančně méně náročná a lze ji aplikovat téměř bez omezení. Horizontální škálovatelnost přináší i výhodu v případě výpadku jednoho serveru, kdy lze data z ostatních serverů stále vytěžovat. Nevýhodou NoSQL databází může být nižší konzistence dat, ale to není pravidlem. Další nevýhodou je, že pro NoSQL databáze neexistuje jeden standartní jazyk jako u relačních SQL databází. Nevýhodu spatřuji i v podpoře od poskytovatele, která je například u Microsoft SQL serveru veliká, kdežto u open source NoSQL databází nedosahuje stejné kvality. Z důvodu distribuce mezi více serverů je také komplikovanější instalace a údržba. Největším nedostatkem je však podle mého názoru nedostatek lidí, kteří dokáží s NoSQL databázemi v současné době pracovat.[22] [23]

4 Výběr úložiště dat

V této části navrhnu vhodné úložiště pro data z výrobní linky, která budu v následujících kapitolách sledovat a zpracovávat vytvořeným monitorovacím nástrojem. Při výběru je nutné uvažovat i to, že data jsou vytěžována z více výrobních linek a také to, že se do vybrané databáze nebudou ukládat pouze data o skartu, ale i další produkční data.

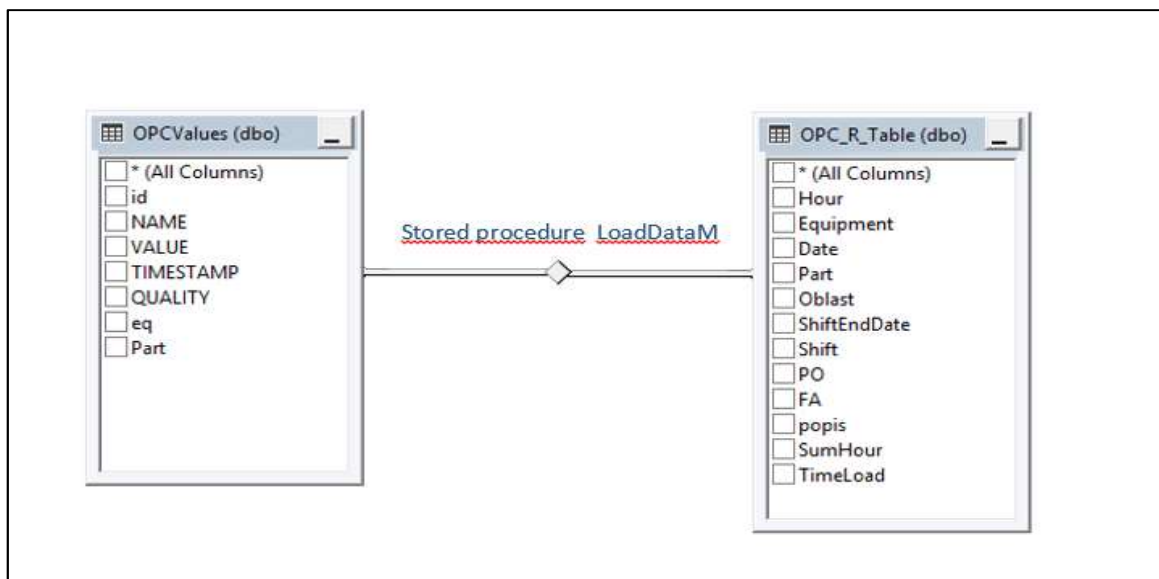
Jelikož jsou data již ukládána do relační databáze Microsoft SQL, tak tato kapitola slouží pouze k zamyšlení nad další možností uchování dat. Výběr úložiště jsem zúžil na již využívanou relační databázi a dokumentovou NoSQL databázi MongoDB. Výhodnost použití určím dle níže popsaných kritérií.

Ukládaná data se skládají z produkčních dat odesílaných každých deset vteřin a dále z informací k daným datům, jako je číslo stroje, typ stroje, část stroje, číslo zakázky, popis dané události anglicky a česky. Dále je nutné zaznamenávat i čas dané události.

4.1 Návrh relační databáze

Jako první uvedu již využívané rozložení databáze a následně navrhnu jiné možné rozložení. Dále je třeba zmínit, že databáze obsahuje více tabulek. Jelikož se v této práci zaměřuji pouze na skart, budu pracovat pouze s částí, ve které jsou uloženy tyto informace.

Informace o skartu jsou v databázi uchovávány pouze ve dvou tabulkách. První tabulka OPCValues je plněna přes OPC server a systémem řízení dat po deseti vteřinách různými produkčními daty. Do druhé tabulky OPC_R_Table se zapisují po hodinách agregovaná data z první tabulky a také informace o zakázkách a vyráběných produktech z jiné databáze nalinkovaného serveru. Toto je zajištěno procedurou LoadDataM, které se automaticky spouští každou hodinu. Tato procedura je stručně popsána v kapitole 4.1.1.



Obrázek 11: Tabulky databáze, zdroj: vlastní

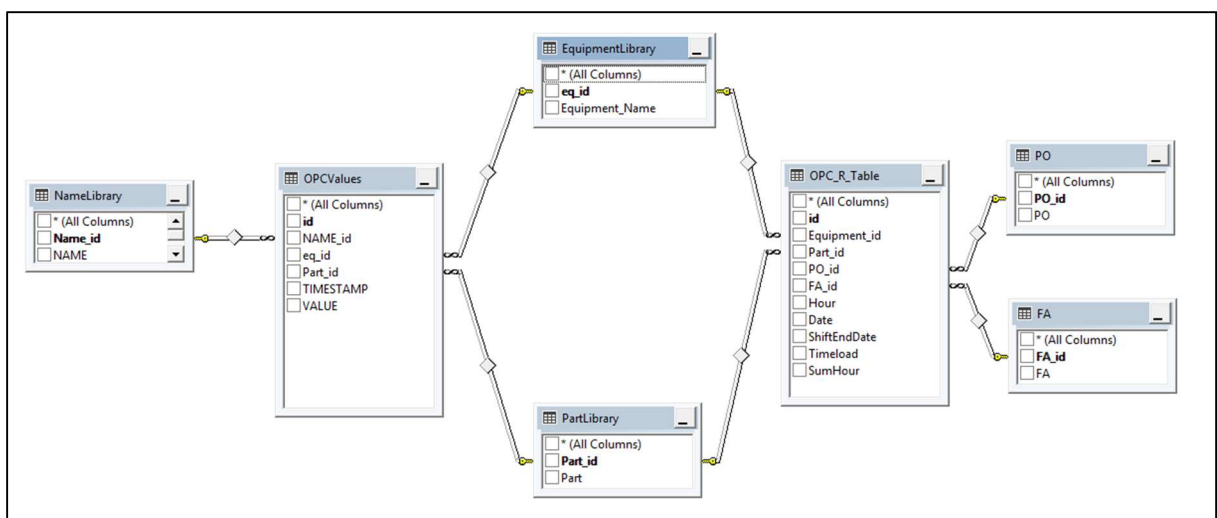
Popis sloupců v tabulce OPCValues	
id	id (číslo zápisu)
NAME	Název stroje a části
VALUE	hodnota (například počet výhozů)
TIMESTAMP	čas podle UTC zóny
QUALITY	informace o chodu stroje (ano/ne)
Eq	stroj
Part	část stroje/důvod skartu

Popis sloupců v tabulce OPC_R_Table	
Hour	Hodina
Equipment	stroj
Date	lokální datum
Part	část stroje/důvod skartu
Oblast	NULL
ShiftEndDate	datum konce směny
Shift	směna
PO	číslo zakázky
FA	Druh produktu
popis	popis produktu
SumHour	Součet skartu za hodinu
TimeLoad	čas konce měřeného úseku

Obrázek 12: Popis sloupců, zdroj: vlastní

Jedná se o relační databázi, a to i přestože jde pouze o dvě faktové tabulky, které nedisponují kombinací primárního a cizího klíče z důvodu splnění podmínek pro relační tabulky. Nevýhodou tohoto uspořádání databáze je potřeba většího uložení dat, v tomto případě databázového serveru. Potřebu většího uložení budu demonstrovat na tabulce OPCValues. Sloupec NAME obsahuje informaci, která je kombinací stroje, části stroje a důvodu skartu (příklad: PAXX.X3.No Blank On 5th Wheel). Tento zápis je prováděn každých deset vteřin pro několik linek skládajících se z několika částí, které mají více skartovacích důvodů. To představuje veliké množství zapisovaných dat. Sloupec NAME je datového typu varchar a na disku/serveru zabírá velikost v závislosti na počtu znaků + dva baity. V uvedeném příkladu jde tedy o 31 bajtů. Vytvořením dvou tabulek a použitím primárního a cizího klíče s datovým typem small int (dva baity) by byl snížen objem dat ve sloupci Name více než desetkrát. [24] Naopak výhodou tohoto modelu je právě to, že jsou všechna data v jedné tabulce, a jsou tedy přehlednější. V tabulce se lépe čte a při dotazování databáze není třeba psát složitější dotazy pro propojení tabulek.

Navrhované možné rozložení databáze by bylo zpracované dle schématu Hvězda. Toto schéma sestává z jedné faktové (v uváděném případě dvou) tabulky a více tabulek dimenzí propojených klíči. Musel bych navrhnout změnu symbolických názvů, které se využívají ke komunikaci OPC serveru a stroje, na čísla v rozsahu datového typu small int a následně vytvořit tabulku dimenze se sloupcem s primárním klíčem, který odpovídá symbolickému názvu a se sloupcem se skutečným názvem. Tímto způsobem by se sloupec NAME oddělil od tabulky OPCValues a byl by nahrazen cizím klíčem. Tabulku bych dále rozdělil na více celků. Vzniklé tabulky dimenzí by byly napojené jak na tabulku OPCValues, tak na tabulku OPC_R_Table. Tabulka OPC_R_Table by byla dále rozdělena dle obrázku 13. K dosažení tohoto rozdělení by dále bylo nutné přepsat proceduru LoadDataM tak, aby se z nalinkovaného serveru plnila nejen tabulka OPC_R_Table, ale i tabulky dimenzí.



Obrázek 13: Relační databáze-návrh, Zdroj: Vlastní

4.1.1 Popis vybraných úseků procedury LoadDataM

Z důvodu rozsahu dané procedury vysvětlím pouze některé její části.

```
, DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), ov.TIMESTAMP) AS TimeStampInLocalTime
```

Obrázek 14: LoadDataM 1, zdroj: vlastní

Tento příkaz nalezneme rozdíl v minutách mezi časovým pásmem UTC a lokálním časem a tento rozdíl přičte ke každému řádku ve sloupci TIMESTAMP (datový typ datetime), následně je sloupec pojmenován TimeStampInLocalTime. V některém z dalších kroků je sloupec přejmenován na TimeLoad.

```
, case when datepart(hh,(DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), ov.TIMESTAMP)))  
between 6 and 17 then 'G' else 'H' end as shift
```

Obrázek 15: LoadDataM 2, zdroj: vlastní

V tomto příkazu se nejprve opět převede sloupec TIMESTAMP na lokální čas a následně je z již upraveného data zjištěna hodina, podle které se vygeneruje označení ranní nebo noční směny G/H.

```
from cte2 left join XXX on Starteq = XXX.Name and UTC_TimeStamp between XXX.RealStart and  
isnull(XXX.RealEnd,getUTCdate())
```

Obrázek 16: LoadDataM 3, zdroj: vlastní

Název připojené databáze byl z důvodu uchování interních informací změněn. Tento příkaz napojí zleva tabulku z nalinkované databáze na virtuální tabulku odvozenou z tabulky OPCValues pomocí časů, kdy hodnoty ve sloupci UTC_TimeStamp musí být v rozmezí hodnot ve sloupci RealStart a sloupci RealEnd. Funkcí isnull a getUTCdate je doplněn sloupec o aktuální datum v případě hodnoty null. Na základě tohoto propojení tabulek jsou data z tabulky OPCValues doplněna o sloupce číslo zakázky (PO), popis produktu (popis), druh produktu (FA).

Celá procedura LoadDataM s přidáním stručnými vysvětlivkami je k nalezení v příloze č.1. Tato procedura se spouští každou hodinu, 21. minutu. To představuje určitý problém s doručováním informací v co nejkratším čase. Tento čas byl zvolen z důvodu rovnoměrného zatížení serveru. V případě, že bude monitorovací nástroj vyhodnocen jako užitečný a bude dále používán, navrhuji spouštění procedury v příhodnějších časech.

4.2 NoSQL Dokumentové databáze

Ukládání dat o skartu do dokumentové databáze bych navrhl následovně. Data by byla ukládána do dokumentů s informací o čase a informacemi o hodnotách jednotlivých skartovacích míst.

Příklad:

```
{
  "id": 1 -- generováno automaticky a slouží pouze jako unikátní hodnota dokumentu
  "Time": "2018-05-12 13:21:37.510" -- čas události
  "PAxx.X3.LooseEnds": "25" -- Na lince xx, části stroje X3 bylo vyskartováno 25 krabiček na prázdné konce
  "PAxx.X3.Pull Absence": "12"
  .
  .
  .
}
```

Název hodnot by byl odvozen dle symbolického názvu stejným způsobem jako u relačních databází. Problém nastává u možnosti připojení informací o zakázkách a výrobě, a to z důvodu jejich uložení v SQL databázi. Možností je například využití překladače Mongify. Tuto možnost uvádím pouze informativně, jelikož není možnost její implementace, a to z důvodu využívání pouze relačních databází. V současné době by nebyla využita ani výhoda škálovatelnosti, která by však do budoucna mohla mít své využití v případě sledování dalších oblastí výroby, jako je výroba filtrů nebo ostatních tabákových produktů (například plechovky plněné tabákem). V tomto případě by byla výhoda uložení dat na více serverů, kde v případě výpadku jednoho lze stále sledovat data na ostatních.

4.3 Shrnutí

Z předchozích kapitol vyplývá výhoda dokumentové NoSQL databáze především v následné práci s uloženými daty, ať už jde o škálovatelnost, rychlost dotazů nebo volné schéma databáze. Nevýhodou je zmíněný problém s připojením na jiné databázové systémy, například právě SQL server. Pokud by tedy nebylo nutné připojovat informace o zakázkách, tak by dokumentová databáze byla jako úložiště dat vhodnější. Jelikož je toto propojení nutné, bylo rozhodováno pouze mezi stávajícím schématem relační databáze a jeho navrhovanou modifikací. Současné rozložení relační databáze sestává pouze ze dvou faktových tabulek, což umožňuje lepší orientaci v datech a zároveň snazší dotazování. Nevýhodou je větší uložený objem dat z důvodu duplicitního ukládání. Jelikož v současné době není nutné ukládání historických dat po kratších intervalech než hodinách, jsou data v tabulce OPCValues udržována pouze dva dny a následně automaticky mazána. Historická data se udržují pouze v tabulce OPC_R_Table, ve které jsou již agregována po hodinách, a tím pádem nepředstavují takový problém s ohledem na

objem. Proto navrhuji zachovat stávající stav, dokud nebudou požadována také historická data z tabulky OPCValues. V takovém případě bude nutné jednotlivé tabulky rozdělit na faktové a tabulky dimenzí a vytvořit nové procedury pro jejich plnění z nalinkovaného serveru.

5 Tvorba monitorovacího nástroje

V této kapitole se věnuji popisu tvorby nástroje pro monitorování skartu na výše popsané výrobní lince. K tvorbě využívám část databáze popsané v kapitole 4.1. V následující části se věnuji popisu validace dat o skartu ukládaných do databáze, a nakonec již přípravě monitorovacího nástroje. Tento nástroj je pouze prvním krokem k monitorování dané výrobní linky. Po dokončení bude spolu s dalšími možnostmi představen koncovým uživatelům, kteří zhodnotí jeho využitelnost a bude diskutováno jeho finální provedení a přidání dalších produkčních dat, jakými jsou například informace o zastávkách stroje. Samotná výrobní linka je již vybavena vlastními monitorovacími nástroji, avšak tento by měl být zpracován tak, aby byl přístupný i vzdáleně a uzpůsobený k odhalování případných problémů v co nejkratší době. Dále by měl informovat o problémech, které se mohou vyskytnout v blízké budoucnosti.

5.1 Validace dat

Validaci dat jsem prováděl jednak za účelem zjištění správného čtení a následného zápisu dat do databáze, tak také z důvodu roztřídění daných skartovacích kategorií, které se z části překrývají. Validace byla rozdělena na dva úseky po dvou dnech. V prvním úseku byla validace provedena na balicím stroji, přesněji jeho části, ve které se cigarety balí do krabiček. V druhém úseku probíhala validace na výrobním stroji. Na ostatních strojích jsem validaci dat neuskutečnil. Celkem tedy tato činnost trvala čtyři dny, a to vždy přibližně tříhodinovým měřením a následnou analýzou. Validaci předcházela část přípravy, ve které jsem připravil dotazy do databáze (dotazy jsou k nalezení v přílohách č. 2 a č. 3), data z nich získaná byla následně zobrazena v pivot tabulce v Excelu, který byl na databázi napojen. Tato příprava mi umožnila poměrně efektivně, ale hlavně přehledně sledovat nové zápisy do databáze.

5.1.1 Validace balicího stroje

Validace probíhala pouze na části balicího stroje, která balí cigarety do krabiček. Jednotlivá místa a příčiny vzniku skartu byly popsány v kapitole 2.6. Je zde však nutné přiřadit jednotlivé názvy příčin ze sloupce part (názvy v databázi) jednomu ze čtyř skartovacích míst a v případě nejasného názvu také důvod.

Jednotlivé názvy jsou seřazeny sestupně podle průchodu strojem. Sloupec Část obsahuje informaci o skartovacím místě.

Part	Část	Popis názvů příčin
X3.Loose Ends	1	Prázdné konce cigaret, zjištěné kontrolou hustoty tabáku.
X3.Incomplete Cigaret Pockets	2	Neúplné balíčky cigaret.
X3.Flexible Wall Jam	2	Nesprávné narovnání cigaret v kapsách 1.
X3.Cigarette Jams In Pockets	2	Nesprávné narovnání cigaret v kapsách 2.
X3.Total Cigarette Pocket Rejections	2	Výhoz cig. kapes celkem. Součet všech předchozích kategorií.
X3.Pull Absence	3	Chybějící odtrhovací část alufólie.
X3.2nd-3rd Wheel Jam	3	Zácpa kapes mezi druhým a třetím kolem.
X3.Lower Foil Missing	3	Chybí spodní fólie.
X3.Foil Slips Out Of Position	3	Nepřesně nasazená fólie.
X3.Total 3rd Wheel Rejects	3	Výhozy na třetím kole celkem. Součet předchozích kategorií.
X3.Blank Gluer Out Of Phase	4	Motor lepení výseků mimo fázi.
X3.Packets With Dried Glue: Blank	4	Balíčky se zaschlým lepidlem. Po delší zastávce.
X3.No Inner-Frame On 4Th Wheel	4	Chybí krček.
X3.Innerframe With Centering Error	4	Krček je špatně umístěn.
X3.5th Wheel Rear Wrap Check	4	Kontrola balení na 5. kole.
X3.No Blank On 5th Wheel	4	Chybí výsek.
X3.Second 5th-6th Wheel Jam	4	Zácpa krabiček při předávání mezi 5. a 6. kolem.
X3.Rejected Production	//	Celkový součet vyskartované produkce.

Tabulka 1: Názvy příčin skartu výrobní stroj, zdroj: Vlastní

Tabulka 1 je až výsledkem procesu validace. Nyní je uvedena pouze pro přehlednost a porozumění následující části textu.

Dále popíši tvorbu dotazu do databáze a také zobrazení dat pomocí Excelu. Uvedl jsem pouze několik příkladů, které demonstrují postup při tvorbě dotazu. Dotaz byl tvořen pouze na tabulku OPCValues. Nejdříve bylo pro přehlednost nutné převést čas na lokální. Toho jsem dosáhl stejným způsobem jako v kapitole 4.1.1. Následně jsem funkcí convert změnil datový typ pro sloupec VALUE na int.

```
,convert(int,[VALUE]) as val
```

Obrázek 17:Validační dotaz 1.1, zdroj: vlastní

Toho jsem však mohl docílit až po vyfiltrování dat podmínkou where, jelikož například řádky uvádějící váhu cigaret obsahují znaky, které nelze převést. Po tomto kroku již bylo možné převést hodnoty řádků ze sloupce part na řádky. Toho jsem docílil funkcí PIVOT.

```
PIVOT(max([val]) for [part] in (  
[X3.Rejected Production]  
,[X3.Incomplete Cigaret Pockets]  
,[X3.Second 5th-6th Wheel Jam]  
,[X3.First 5th-6th Wheel Jam]  
,[X3.Packets With Dried Glue: Blank]  
,[X3.Flexible Wall Jam]  
,[X3.5th Wheel Rear Wrap Check]  
,[X3.Jam At Exit Drum]
```

Obrázek 18: Validační dotaz 1.2., zdroj: vlastní

Jelikož se v paměti PLC hodnoty skartu pro jednotlivé kategorie stále přičítají a nejsou mazány po každém odeslání do databáze, bylo nutné zjistit rozdíl následujících řádků. Toho jsem docílil funkcí row_number ve virtuální tabulce cte1.

```
row_number() over(order by timestamp) as seq
```

Obrázek 19: Validační dotaz 1.3., zdroj: vlastní

Rozdíl po sobě jdoucích hodnot jsem získal touto částí dotazu.

```
from cte1 a  
join cte1 b on a.seq = b.seq+1  
where a.[X3.Rejected Production]-b.[X3.Rejected Production] >= 0
```

Obrázek 20: Validační dotaz 1.4, zdroj: vlastní

Data získaná tímto dotazem byla zobrazena v Excelu, který podporuje možnost připojení na databázi. Poté jsem data rozložil v pivot tabulce tak, aby v nich byla možná snadná orientace. Na obrázku 21 je pivot tabulka použitá k validaci dat. Hodnoty však byly plošně změněny z důvodu interně stanovených pravidel.

DAY/HOUR/MIN		6															
PARTS		34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
5	Sum of TotalReject	0	2	6	4	7	1	3	8	4	4	1	6	4	0	5	1
	Sum of IncompleteCigaretPockets	0	1	1	0	0	1	0	0	1	1	1	4	1	0	1	0
	Sum of First5th_6thWheelJam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of PacketsWithDriedGlueBlank	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of FlexibleWallJam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of th5WheelRearWrapCheck	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	Sum of nd2_3rdWheelJam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of TotalCigarettePocketRejections	0	1	2	0	0	1	0	5	1	3	1	4	3	0	1	1
	Sum of LooseEnds	0	0	1	0	0	0	0	5	0	2	0	0	2	0	0	1
	Sum of Total3rdWheelRejects	0	1	3	3	3	0	2	1	2	1	0	0	0	0	0	0
	Sum of CigaretteJamsInPockets	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of NoBlankOn5thWheel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of PullAbsence	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of NoInner_FrameOn4ThWheel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of InnerframeWithCenteringError	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of FoilSlipsOutOfPosition	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sum of LowerFoilMissing	0	1	3	3	3	0	2	1	2	1	0	0	0	0	0	0
	Sum of BlankGluerOutOfPhase	0	0	1	1	4	0	1	1	1	0	1	1	0	0	4	0

Obrázek 21: Pivot tabulka - dotaz validace, zdroj: vlastní

Tímto způsobem bylo možné jednoduše zjistit počet vyskartovaných výrobků po minutách u každé kategorie. Další fáze validace probíhala již u výrobní linky. První jsem vyprázdnil obě skartovací nádoby a zaznamenal čas v předem připraveném Excelu. Po alespoň částečném naplnění skartovacích nádob jsem nádoby opět vyprázdnil a znovu zaznamenal čas. Následně jsem zjistil počet vyskartovaných alufólií z první skartovací nádoby a počet výseků z druhé skartovací nádoby. Jejich součtem byl zjištěn počet celkem vyskartovaných balíčků cigaret. Tyto operace se opakovaly celkem desetkrát.

V další části proběhlo porovnání naměřených hodnot s hodnotami v předem připravené pivot tabulce. Tabulky na obrázku 22 obsahují informace o výsledcích měření porovnané s daty uloženými v databázi.

První série měření			
pořadí	Měření celkem	Report celkem	Rozdíl v %
1	39	57	31,6%
2	78	78	0,0%
3	55	56	1,8%
4	22	24	8,3%
5	75	74	-1,4%
6	46	45	-2,2%
7	79	88	10,2%
8	71	72	1,4%
9	41	50	18,0%
10	66	67	1,5%

Druhá série měření			
pořadí	Měření celkem	Report celkem	Rozdíl v %
1	50	51	2,0%
2	57	66	13,6%
3	48	58	17,2%
4	75	75	0,0%
5	82	81	-1,2%
6	49	49	0,0%
7	88	90	2,2%
8	38	38	0,0%
9	48	66	27,3%
10	79	78	-1,3%

Obrázek 22: Výsledky měření X3, zdroj: Vlastní

Po první sérii měření byla zjištěna jen mírná nuance u šesti měření, jedno měření odpovídalo dokonale, ale u tří měření se výsledky razantně lišily. Nepřesnosti v řádu jedné nebo dvou krabiček jsem přiřadil dopadu krabičky nebo alufólie mimo skartovací nádobu. Měření s vyšším procentuálním rozdílem jsem komunikoval s mechaniky a byla zjištěna další příčina skartace. Balicí stroj automaticky vyskartuje nastavený počet cigaret při rozjezdu po určitých typech zastávek. Jelikož však skartuje pouze cigarety z výstupu násypky do pásového dopravníku, není skartován výsek ani alufólie. To způsobilo procentuální rozdíl ve výpočtu. Při druhé sérii měření jsem již tento problém bral v potaz a příčina byla potvrzena.

5.1.2 Validace výrobního stroje

Další část měření jsem prováděl na výrobním stroji. Validace probíhala obdobným způsobem, připravil jsem dotaz do databáze, kterým byla data v Excelu prezentována ve formě vhodné pro validaci. Jelikož byl postup obdobný, není třeba, abych jej podrobně popisoval. Dotaz do databáze je i s komentáři k nalezení v příloze č. 3, avšak je nutné upozornit na fakt, že z důvodu využití dotazu pouze k validaci nebyl dotaz optimalizován na co nejkratší dobu odpovědi. Jeho provedení se proto může jevit jako nevyhovující. V tabulce 2 jsou seřazeny jednotlivé skartovací názvy příčin ze sloupce Part.

Part	Popis názvů příčin
Maker.Reject_LightWeightCnt	Příliš lehké cigarety
Maker.Reject_HeavyWeightCnt	Příliš těžké cigarety
Maker.Reject_MissingFilterCnt	Chybějící filtr
Maker.Reject_SoftSpotsCnt	Cigarety s mezerami v tabákovém provazci
Maker.Reject_HardSpotsCnt	Cigarety se zhuštěným tabákovým provazcem
Maker.Reject_VentilationCnt	Chybná ventilace
Maker.Reject_AirTightnessCnt	Chybná těsnost
Maker.Reject_LightEndsCnt	Lehký konec cigarety
Maker.Reject_LESCnt	Nedostatečně naplněný konec cigarety
Maker.Reject_OTISCnt	Tvarové chyby
Maker.Reject_TotalRejects	Celkový počet skartovaných cigaret

Tabulka 2: Názvy příčin skartu 2, zdroj: Vlastní

Jelikož měření probíhalo během běžné produkce, bylo poměrně komplikované zaručit jeho správnost, a to z několika důvodů. Prvním důvodem je vyšší počet skartovacích nádob než na balicím stroji, které nebylo možné vyprázdnit všechny v jeden okamžik. Mezi vyprázdněním první a poslední nádoby mohlo být vyskartováno několik cigaret, které mohly měření zneprávnit. Z tohoto důvodu jsem měření prováděl po delší časový interval, aby tato odchylka byla procentuálně nižší. Další nepřesnost do měření vnáší cigarety, které vypadnou mimo skartovací nádoby.

První série měření			
pořadí	Měření celkem	Report celkem	Rozdíl v %
1	138	144	4,2%
2	220	226	2,7%
3	58	55	-5,5%
4	124	126	1,6%
5	65	67	3,0%

Druhá série měření			
pořadí	Měření celkem	Report celkem	Rozdíl v %
1	49	53	7,5%
2	36	41	12,2%
3	83	80	-3,8%
4	114	122	6,6%
5	68	75	9,3%

Obrázek 23: Výsledky měření MAX, Zdroj: Vlastní

Výsledky měření, až na jedno měření, se od údajů ukládaných do databáze lišily do 10 %. Tato odchylka byla připsána výše zmíněným důvodům a čtení dat jsem vyhodnotil jako správné. Dále byla v databázi zjištěna odchylka celkové skartované produkce od součtu jednotlivých skartovacích kategorií, to je způsobeno logikou výpočtu celkové skartované produkce. Ta je v MLP vypočtena jako rozdíl vyhovující produkce části SE a vyhovující produkce části MAX. Tento rozdíl by mohl být způsoben například při zastávce stroje způsobené poruchou, což by zapříčinilo odpadnutí cigaret z válců části MAX.

Validace dat na balicím stroji proběhla úspěšně a byla potvrzena správnost dat v databázi. Na výrobním stroji jsem zjistil odchylky v řádu jednotek %, jejichž příčinu jsem přiřadil výše zmíněným důvodům.

5.2 Monitorovací nástroj – tvorba

Poté, co jsem ověřil správnost dat uložených v databázi, bylo nutné zvážit, jakým způsobem data prezentovat. Prvním návrhem byla připojená databázová aplikace v C#, ale po konzultaci s IS oddělením byl vybrán nástroj SSRS (SQL Server Reporting Services), a to z několika důvodů. Jedním z důvodů je možnost použití již naprogramovaných částí, které tento software nabízí, jako jsou grafy nebo tabulky. Dalším důvodem je snadná kombinace několika zdrojů dat, což byl jeden z požadavků. Nejdůležitějším faktorem bylo však nepřidělení licence pro vývojové prostředí Microsoft Visual Studio, tato licence není pro SSRS vyžadována.

Vytvářený monitorovací nástroj by měl sloužit vedoucím pracovníkům jednotlivých směn zprvu jako přehled o skartu a jeho nejčastějších příčinách na právě probíhajících zakázkách a zároveň by je měl informovat o problémech, se kterými by se mohli setkat u zakázek následujících, a to bez nutnosti fyzické přítomnosti u dané linky. Proto tento sledovací nástroj rozdělím do čtyř bloků. První blok bude obsahovat informace o skartu na předchozí směně. Ve druhém bloku budou informace o aktuální směně rozděleny po hodinách, ve třetím bloku bude informace o nadcházejících zakázkách. Čtvrtý blok bude obsahovat informaci o skartu za poslední hodinu rozdělenou do šesti částí po deseti minutách.

V prvním kroku při tvorbě tohoto nástroje jsem připravil sadu dotazů do databází, přičemž většina dotazů směřuje k databázi popsané výše a pouze informace o nadcházejících zakázkách jsem vytěžil z datového skladu.

5.2.1 Dotazy

K tvorbě dotazů jsem použil literaturu Lacko, SQL – Hotová řešení. [25]

V této části práce vysvětlím význam jednotlivých dotazů, dotazy samotné budou z důvodu jejich rozsahu přiloženy do přílohy. Všechny dotazy jsem připravil ještě před začátkem tvorby nástroje, což jsem zpětně vyhodnotil jako nevýhodné, jelikož jsem následně část z nich musel upravit tak, aby je bylo možné snadno použít. Dotazy jsem tvořil v prostředí SQL Server Management Studia, což je softwarová aplikace pro správu Microsoft SQL Serveru. Vytvořil jsem celkem třináct dotazů. Data v níže uvedených výstřizích jsou oproti skutečnosti upravena z důvodu ochrany interních informací.

Prvním dotazem MA_LastShift_RR (v příloze č. 4) jsem z databáze získal informace o skartu a produkci cigaret za celou předchozí směnu.

	Equipment	ShiftEndDate	Shift	PO	FA	IDD	ProductionTH	RejectPC	RR	active
1	MA	2018-07-15	H	000101	FA05	1	1066	8910	0.35	0

Obrázek 24:Result1, Zdroj: Vlastní

Výsledek dotazu bude vysvětlen z levé strany. Sloupec Equipment značí, o jaké zařízení se jedná, v tomto případě jde o výrobní stroj (Maker). Druhý sloupec obsahuje informaci o datu konce směny, dále informace o směně Shift, číslo zakázky PO, kód výrobku FA, sloupec IDD určuje pořadí zakázek v případě, že na dané směně byla vyráběna více než jedna, sloupec ProductionTH nese informaci o vyrobených kusech cigaret v tisících, RejectPC jsou počty vyřazených cigaret v kusech, RR (Reject Rate) je procento skartu a sloupec active nese informaci 1 nebo 0 dle toho, zda směna právě probíhá.

Druhý dotaz X3_LastShift_RR (v příloze č. 5) obsahuje stejné informace, pouze o části balicího stroje X3. Další dva dotazy MA_LastShift_RR_Top3 a X3_LastShift_RR_Top 3 (v přílohách č. 6 a č. 7) vytěžují informace o třech nejčastějších důvodech skartu na daném zařízení. Na následujícím obrázku je výsledek dotazu pro zařízení X3.

	Equipment	ID	PART	ShiftEndDate	Shift	PO	FA	RejectPc	active
1	PA	1	Blank Gluer Out Of Phase	2018-07-15	H	000101	FA05	506	0
2	PA	2	Packets With Dried Glue: Blank	2018-07-15	H	000101	FA05	279	0
3	PA	3	Total Blank Rejections	2018-07-15	H	000101	FA05	93	0

Obrázek 25:Result2, Zdroj: Vlastní

Jména sloupců jsou popsána výše. Je však potřeba doplnit význam zkratky PA (Packer) a názvu sloupce ID (indikace pořadí příčin).

Dotazy MA_ActiveShift_RR a X3_ActiveShift_RR (přílohy č. 8 a č. 9) vytěžují informace o skartu a produkci na právě probíhající směně. Tyto informace jsou dále agregovány po hodinách.

	Hour	Equipment	Date	Shift	PO	FA	ProductionTH	RejectPC	RR	active
1	15	MA	2018-07-15	G	101	FA	219	1343	0.43	1
2	14	MA	2018-07-15	G	101	FA	245	1137	0.32	1
3	13	MA	2018-07-15	G	101	FA	312	912	0.20	1
4	12	MA	2018-07-15	G	101	FA	165	1649	0.70	1
5	11	MA	2018-07-15	G	101	FA	143	738	0.34	1
6	10	MA	2018-07-15	G	101	FA	262	853	0.22	1
7	9	MA	2018-07-15	G	101	FA	107	1236	0.80	1
8	8	MA	2018-07-15	G	101	FA	233	1334	0.40	1
9	7	MA	2018-07-15	G	101	FA	211	1050	0.34	1
10	6	MA	2018-07-15	G	101	FA	189	1127	0.41	1

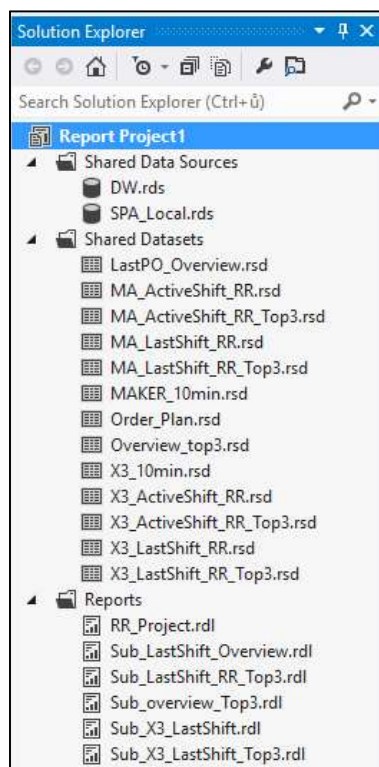
Obrázek 26: Result3, Zdroj: Vlastní

Obrázek 26 obsahuje informace z dotazu o aktuální směně pro výrobní stroj.

Další dva dotazy MA_ActiveShift_RR_Top3 a X3_ActiveShift_RR_Top3 (přílohy č. 10 a č. 11) nesou informace o třech nejčastějších příčinách skartu na dané směně, obdobně jako na obrázku 25. Dotaz Order_Plan (příloha č. 12) zajišťuje informaci o výrobním plánu aktivní a následující směny z datového skladu. Na tento dotaz jsou v SSRS pomocí parametrů napojeny dotazy LastPO_Overview a Overview_Top3 (přílohy č. 13 a č. 14). Tyto dva dotazy zjišťují, zda se v minulosti vyráběl stejný výrobek, jako je ve výrobním plánu a pokud ano, vypíše informace o skartu a produkci obdobně jako na obrázcích 24 a 25. Dále pomocí těchto dotazů zjišťují procento skartu na celé výrobní lince pomocí rozdílu dobré produkce výrobního stroje a dobré produkce stroje C600. Procento skartu celé výrobní linky zjišťují pouze u již ukončených zakázek z důvodu připojených zásobníků, které znemožňují tento výpočet na právě probíhajících zakázkách. Poslední dva dotazy MAKER_10min a X3_10min vytěžují informaci o skartu za poslední hodinu, rozdělenou do šesti částí po deseti minutách obdobně jako na obrázku 26.

5.2.2 SSRS

Nástroj jsem vytvořil pomocí SQL Server Reporting Services s využitím vývojového prostředí Visual Studio. Nejdříve jsem připojil projekt k databázi obsahující data o skartu a dále také do datového skladu. Následně jsem do projektu vložil všechny dotazy jako sdílené datasety. Sdílené datasety a zdroje dat jsem využil z důvodu tvorby více než jednoho reportu v daném projektu.



Obrázek 27: Solution Explorer, Zdroj: Vlastní

Následně jsem vytvořil kostru hlavního reportu, která již obsahovala pole pro vložení subreportů. Při tvorbě reportu jsem musel rozdělit jednotlivé části do bloků tak, aby v případě různého počtu prezentovaných dat bylo vše správně zobrazeno. Jelikož je počet řádků vytěžených dotazy proměnlivý na základě počtu zakázek vyráběných na jedné směně, mění se i rozložení velikosti jednotlivých polí při jejich prezentaci. Práce s SSRS ve Visual studiu je uživatelsky velice přívětivá a SSRS obsahuje spoustu předem naprogramovaných prvků, proto bylo nutné psát zdrojový kód ručně pouze v pár případech. Například při vkládání podmínek pro targety skartu nebo právě při návrhu již zmíněného rozložení.

Podmínky pro targety, dle kterých se části nástroje podbarvují zeleně, nebo červeně jsem nastavil na hodnotu půl procenta skartu u výrobního stroje a jednoho procenta u části X3. Tyto cíle budu dále diskutovat s uživateli a následně je upravím dle potřeby. Data jsou prezentována pomocí tabulek, grafů a bloků, kterým jsem přiřadil jednotlivé dotazy do databází. Dotazy do databází byly vytvářeny tak, aby tvorba nástroje pro prezentaci dat byla již co nejsnazší. Přesto bylo nutné některé dotazy mezi sebou propojit. To v SSRS umožňují parametry, které mi po správném nadefinování umožnily například zobrazení dat o skartu indikující možné problémy u nadcházejících zakázek. Dále jsem nastavil automatický refresh reportu. Tento refresh probíhá každých deset minut, a to tak, aby se v co nejkratším čase objevila všechna data získávaná výše popsanými dotazy. Po desetiminutových intervalech jsou získávány informace z dotazu X3_10min. Jelikož aktualizace trvá osm vteřin, což může být pro uživatele nepříjemné, nastavil jsem automatické zálohování. To umožňuje uložení dat z dotazů

do mezipaměti a následný refresh proběhne odsud. Tvorba nástroje na monitoring skartu probíhala necelý týden. Takto krátký čas tvorby byl umožněn předem připravenými dotazy do databáze, které byly zpracovány tak, aby nebyla nutná následná úprava nebo třídění dat v SSRS.

SSRS je uživatelsky přívětivé, ale bylo vyzorováno i několik nedostatků. Například rozmisťování objektů není vyladěné, a proto jsem tuto část provedl úpravou zdrojového kódu. Problém také nastal při využití vnořených parametrů, tedy parametrů odkazujících na parametry. To mělo vliv na rychlost reportu a z toho důvodu musely být předělány dotazy do databáze tak, aby nebylo nutné vnořené parametry kromě jediného případu využít.

Report je umístěn na firemní síti. Možnost přístupu skrze internetový prohlížeč mají zaměstnanci, kterým jsem udělil přístupová práva. Vzhledem k jeho značnému rozsahu je celý report s upravenými daty zobrazen v příloze č.15.

6 Využití monitorovacího nástroje

V následující části nastíním, jak daný nástroj efektivně využívat, což by mělo vést ke snížení skartu na výrobní lince. Tento nástroj je určen pro vedoucí pracovníky ve výrobě, kterým umožňuje sledovat množství skartu výrobní linky v aktuálním čase. Nástroj také poskytuje informace o možných problémech, které mohou nastat na budoucích zakázkách. Data zobrazená v následující části textu byla z důvodu zachování interních informací změněna a slouží pouze pro vysvětlení funkce.

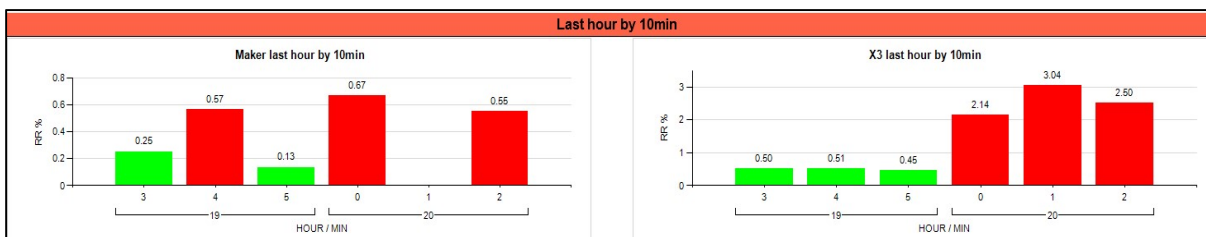
Na začátku směny by měla být v reportu zjištěna informace z bloku LAST SHIFT, ve které je stručný přehled o skartu vyprodukovaném na předchozí směně.

LAST SHIFT			
PO: 000101		FA: FA05	
MAKER		X3	
Production TH:	3,084	Production_Pack:	131,206
Rejected PC:	11,437	Reject_Pack:	1,652
RR%:	0.37	RR %:	1.26
TOP 3		TOP 3	
Part	Reject Pc	Part	Reject Pc
VentilationCnt	3,904	Blank Gluer Out Of Phase	544
AirTightnessCnt	1,580	Packets With Dried Glue: Blank	323
HardSpotsCnt	1,032	Total Blank Rejections	124

Obrázek 28: Last shift, Zdroj: Vlastní

Informace z této části by měl vedoucí pracovník zhodnotit a v případě zvýšeného množství skartu informovat operátory o jeho důvodu, popřípadě poslat k výrobní lince mechaniky, kteří by stroj seřídili.

V průběhu směny by měla být sledována část ACTUAL SHIFT a část Last hour by 10min. V těchto částech se nachází informace o stavu skartu během právě probíhající směny.



Obrázek 29: Last hour by 10min, Zdroj vlastní

Na obrázku 29 je zobrazena část monitorovacího nástroje, na které lze pozorovat vývoj skartu za poslední hodinu rozdělenou do šesti částí po deseti minutách. Z obrázku tedy vyplývá, že v posledních třiceti minutách se na části stroje X3 razantně zvýšila vyskartovaná produkce. Na takovou situaci by měl vedoucí pracovník v co nejkratší době zareagovat a zajistit odstranění problému.

X3 TOP 3				
ID	PO	FA	PART	Reject Pc
1	101 [REDACTED]	FA05 [REDACTED]	X3.loose ends	74
2	101 [REDACTED]	FA05 [REDACTED]	X3.Packets With Dried Glue: Blank	36
3	101 [REDACTED]	FA05 [REDACTED]	X3.Total Blank Rejections	15

Obrázek 30: Actual Shift, Zdroj: Vlastní

Obrázek 30 zobrazuje část informací z bloku ACTUAL SHIFT. Slouží k zjištění nejčastějších příčin skartu. Oba předchozí obrázky lze zhodnotit následovně: za posledních třicet minut bylo na části X3 balicího stroje zaznamenáno zásadní zvýšení skartu. Nejčastějším důvodem skartace byly prázdné konce cigaret. V tomto případě by měl být ke stroji vyslán mechanik, který by seřídil kontrolu prázdných konců cigaret na výrobním stroji. Pokud by cigarety s prázdnými konci byly vyskartovány již na výrobním stroji, znamenalo by to razantní snížení skartu, protože na balicím stroji jsou spolu s vadnými cigaretami vyskartovány i cigarety vyhovující, a to v jedné krabici.

V případě plánované změny výroby na jiný typ výrobků je možné sledovat poslední část reportu FOLLOWING SHIFT. Tato část nese informaci o typu produktu, který se bude vyrábět a pokud se již v minulosti vyráběl, zobrazí informace o této výrobě.

FOLLOWING SHIFT			
FA05 [REDACTED]			
OrderNo:	MAKER RR %	X3 RR %	TOTAL RR %
000101 [REDACTED]	0.43%	1.62%	4.25%
TOP 3			
Part		reject PC	
X3.Blank Gluer Out Of Phase		52,693	
X3.Packets With Dried Glue: Blank		26,450	
Maker.Reject_VentilationCnt		6,721	

Obrázek 31: Following Shift, Zdroj: Vlastní

Z obrázku 34 je patrné, že typ výrobku, který bude vyráběn na další zakázce, už jednou produkován byl. Nejčastější příčinou skartu byl motor lepení výseků mimo fázi. V tomto případě by měl vedoucí pracovník dohlédnout na správné nastavení této části stroje již při seřizování stroje na nový typ výroby.

Tento nástroj při jeho správném používání představuje vhodný návrh na možné snížení skartu.

7 Závěr

Cílem této bakalářské práce bylo seznámit se s výrobní linkou se zaměřením na skartovací místa a zorientovat se v databázi, která obsahuje data o skartu. Dalším cílem byla validace dat odesílaných do databáze, a to pomocí pravidelně prováděného měření. Následně jsem sestavil sadu dotazů do databáze, které jsem využil při tvorbě monitorovacího nástroje.

V teoretické části jsem stručně popsal fungování výrobní linky, dále byly popsány rozdíly mezi využitím relačních databází a NoSQL databází, což mi poskytlo teoretické ukotvení pro činnost v praktické části, kde jsem popsal současné schéma databáze a navrhl dvě další možná řešení. Prvním řešením byla pouze úprava stávajícího řešení optimalizovaná na snížení objemu ukládaných dat. Druhým návrhem bylo ukládání dat do NoSQL dokumentové databáze MongoDB.

Následně probíhala validace dat, kterou jsem na balicím stroji ověřil správnost ukládání dat, zatímco na výrobním stroji se vyskytly menší odchylky, které jsem však přiřadil určitým důvodům. Poté jsem vytvořil dotazy do databáze, z nichž některé musely být následně uzpůsobeny pro hladší chod monitorovacího nástroje.

Samotná tvorba monitorovacího nástroje probíhala víceméně bezproblémově, jelikož jsem pracoval s předem připravenými dotazy. Jediný nedostatek zpětně spatřuji v grafickém provedení, což bylo však částečně způsobeno omezenými grafickými možnostmi programu SSRS. Jedním ze současných nedostatků, které lze do budoucna eliminovat, je spouštění procedury LoadDataM až ve 21. minutě

každé hodiny, což způsobuje pozdní načítání dat. Tento systém byl zvolen z důvodu potřeby rovnoměrně rozložit zatížení serveru, což lze v budoucnu při používání nástroje změnit.

Vytvořený monitorovací nástroj bude v současné době využíván pouze na této jedné konkrétní lince, jeho přínos bude dále hodnocen a v případě pozitivních výsledků bude zváženo jeho rozšíření na další výrobní linky s možností přidání dalších produkčních dat, jako jsou např. informace o zastávkách stroje.

8 Reference

- [1] „Přehled | PMI - Czech Republic - CS“. [Online]. Dostupné z: <https://www.pmi.com/markets/czech-republic/cs/about-us/overview>. [cit.: 2018-07-19].
- [2] Č. televize, „Kutná Hora - Monte Carlo středověku a město tabáku“, *ČT24 - Nejdůvěryhodnější zpravodajský web v ČR - Česká televize*. [Online]. Dostupné z: <https://ct24.ceskatelevize.cz/relax/1136976-kutna-hora-monte-carlo-stredoveku-a-mesto-tabaku>. [Viděno: 19-čvc-2018].
- [3] „Hauni network - Hauni GmbH“. [Online]. Dostupné z: <https://www.hauni.com/en/company/hauni-network.html>. [cit.: 2018-07-07].
- [4] G. S. Cross a R. N. Proctor, *Packaged Pleasures: How Technology and Marketing Revolutionized Desire*. University of Chicago Press, 2014.
- [5] Hauni Maschinenbau, *Interní materiál 1: Service Manual Protos PM100*. 2004.
- [6] Hauni Maschinenbau, *Interní materiál 2: Operator's Manual*. 2004.
- [7] A. Vojáček, „Optické senzory přiblížení - obecný popis | Automatizace.HW.cz“, 2005. [Online]. Dostupné z: <https://automatizace.hw.cz/clanek/2005121901>. [cit.: 2018-07-13].
- [8] Hauni Maschinenbau, *Interní materiál 3: Software Package PM5*, roč. 2004. .
- [9] Hauni Maschinenbau, *Interní materiál 4: CIS Manual*. 2005.
- [10] M. Martinásková a L. Šmejkal, *PLC a automatizace / Základní pojmy, úvod do programování*. Praha: Ben - technická literatura, 1999.
- [11] M. Martinásková a L. Šmejkal, *PLC a automatizace / Sekvenční logické systémy a základy fuzzy logiky*. Praha: Ben - technická literatura, 2005.
- [12] Hauni Maschinenbau, *Interní materiál 5: Software Package MLP*. 2005.
- [13] Foxon, „Co je OPC? (OPC server, OPC klient)“, 2018. [Online]. Dostupné z: <https://www.foxon.cz/blog/10-odborne-clanky/100-co-je-opc-opc-server-opc-klient?search=Programov%C3%A1n%C3%AD+PLC>. [cit.: 2018-07-13].
- [14] „About us“, *G.D S.p.A*. [Online]. Dostupné z: www.gidi.it/en/about-us. [cit.: 2018-07-08].
- [15] „History, Mission & Vision | Senzani Brevetti S.p.A.“ [Online]. Dostupné z: <http://www.senzani.com/en/company>. [cit.: 2018-07-08].

- [16] „X3“. [Online]. Dostupné z: <http://www.gidi.it/en/solutions/product/x3>. [cit.: 2018-07-13].
- [17] G.D., *Interní materiál 6: Operator Manual X3*. 2018.
- [18] G.D., *Interní materiál 7: Mechanical Adjustments X3*. 2018.
- [19] G.D., *Interní materiál 8: Operator Manual C600*. 2018.
- [20] Senzani, *Interní materiál 9: Instruction Manual SBNE*. .
- [21] D. M. Kroenke a D. J. Auer, *Databáze*. Brno: Computer Press, 2015.
- [22] I. Holubová a kol., *Big Data a NoSQL databáze*. Praha: Grada Publishing, a.s., 2015.
- [23] T. Panyko, „NoSQL databáze“. Jihočeská Univerzita v Českých Budějovicích, 2013.
- [24] MikeRayMSFT, „Data types (Transact-SQL)“. [Online]. Dostupné z: <https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql>. [cit.: 2018-07-13].
- [25] L. Lacko, *SQL - Hotová řešení*. CPRESS, 2004.

Seznam obrázků

Obrázek 1: Výrobní stroj (ilustrační), zdroj:

<https://www.hauni.com/en/nc/service/services/modernization-and-tip/filter-cigarette-maker/detail/product/protos-100-1.html> 11

Obrázek 2: Měření hustoty v zapalovaném konci, vysvětlivky: 1 = senzor LES, 2 = přijímače, zdroj:

Interní materiál 4..... 14

Obrázek 3: Kontrola tvaru cigarety, vysvětlivky: 1 = přijímač, 2 = požadovaný průměr cigarety, 3 = clona s otvory, 4 = vysílač, zdroj: Interní materiál 4 15

Obrázek 4: Struktura systému, zdroj: Interní materiál 5 16

Obrázek 5: Schéma, zdroj: vlastní 17

Obrázek 6: X3 (ilustrační), zdroj: <http://www.gidi.it/en/solutions/product/x3> 19

Obrázek 7: C600 (ilustrační), zdroj: <http://www.gidi.it/en/solutions/product/c600-pack> 20

Obrázek 8: SBNE (ilustrační), zdroj: <http://www.senzani.com/en/sector/tobacco/bag> 21

Obrázek 9: Dokumenty, Zdroj: Big Data a NoSQL databáze 24

Obrázek 10: Dokumenty s odkazy, Zdroj:Big Data a NoSQL databáze	24
Obrázek 11: Tabulky databáze, zdroj: vlastní	26
Obrázek 12: Popis sloupců, zdroj: vlastní	26
Obrázek 13: Relační databáze-návrh, Zdroj: Vlastní	27
Obrázek 14: LoadDataM 1, zdroj: vlastní.....	28
Obrázek 15: LoadDataM 2, zdroj: vlastní.....	28
Obrázek 16: LoadDataM 3, zdroj: vlastní.....	28
Obrázek 17: Validační dotaz 1.1, zdroj: vlastní	31
Obrázek 18: Validační dotaz 1.2., zdroj: vlastní	32
Obrázek 19: Validační dotaz 1.3., zdroj: vlastní	32
Obrázek 20: Validační dotaz 1.4, zdroj: vlastní	32
Obrázek 21: Pivot tabulka - dotaz validace, zdroj: vlastní	33
Obrázek 22: Výsledky měření X3, zdroj: Vlastní	33
Obrázek 23: Výsledky měření MAX, Zdroj: Vlastní	35
Obrázek 24: Result1, Zdroj: Vlastní.....	36
Obrázek 25: Result2, Zdroj: Vlastní.....	36
Obrázek 26: Result3, Zdroj: Vlastní.....	37
Obrázek 27: Solution Explorer, Zdroj: Vlastní.....	38
Obrázek 28: Last shift, Zdroj: Vlastní	39
Obrázek 29: Last hour by 10min, Zdroj vlastní	40
Obrázek 30: Actual Shift, Zdroj: Vlastní	40
Obrázek 31: Following Shift, Zdroj: Vlastní	41

Seznam tabulek

Tabulka 1: Názvy příčin skartu výrobní stroj, zdroj: Vlastní.....	31
Tabulka 2: Názvy příčin skartu X3, zdroj: Vlastní.....	34

Seznam příloh

- Příloha č. 1:** Procedura LoadDataM
- Příloha č. 2:** Validace - X3
- Příloha č. 3:** Validace - výrobní stroj

Příloha č. 4: Maker_LastShift_RR

Příloha č. 5: X3_LastShift_RR

Příloha č. 6: MA_LastShift_RR_Top3

Příloha č. 7: X3_LastShift_RR_Top3

Příloha č. 8: MA_ActiveShift_RR

Příloha č. 9: X3_ActiveShift_RR

Příloha č. 10: MA_ActiveShift_RR_Top3

Příloha č. 11: X3_ActiveShift_RR_Top3

Příloha č. 12: Order_Plan

Příloha č. 13: LastPO_Overview

Příloha č. 14: Overview_Top3

Příloha č. 15: Monitorovací nástroj

Příloha č. 1: Procedura LoadDataM

```
USE [SPA_local] --použít tuto tabulku
GO
SET ANSI_NULLS ON --definuje práci s NULL
GO
SET QUOTED_IDENTIFIER ON --definuje práci s uvozovkami
GO
ALTER procedure [dbo].[LoadDataM]
as
set nocount on -- zabrání zasílání zpráv o počtu řádků
;
WITH tmp AS -- virtuální tabulka (CTE)
(
SELECT
[id] as radek
, ROW_NUMBER () OVER (Partition by ov.NAME ORDER BY ov.TIMESTAMP asc)
PoradiRadkuOblasti
--očísluje řádky dle hodnot ve sloupci
, ov.TIMESTAMP as UTC_TimeStamp
, DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), ov.TIMESTAMP) AS
TimeStampInLocalTime
--převede sloupec TIMESTAMP na lokální čas
, cast(DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), ov.TIMESTAMP) AS Date) AS
Date
--převede sloupec Date na lokální datum
, case when datepart(hh,(DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()),
ov.TIMESTAMP)))
between 6 and 17 then 'G' else 'H' end as shift
--vytvoří sloupec shift s indikátorem směny
, case when datepart(hh,(DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()),
ov.TIMESTAMP))) >= 18
then dateadd(dd,1,CAST(DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()),
ov.TIMESTAMP) as DATE))
else CAST(DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), ov.TIMESTAMP) as DATE)
end as ShiftEndDate
--vytvoří sloupec ShiftEndDate (datum konce směny)
, left([NAME], 4) eq --vytvoří sloupec eq odvozený ze sloupce NAME
, [NAME] as Oblast
, right([NAME],len([NAME])-5) Part -- vytvoří sloupec Part odvozený ze sloupce NAME
, Value as Value
, [QUALITY] as Quality

FROM [SPA_local].[dbo].[OPCValues] ov

where -- podmínka where vyfiltruje pouze požadované řádky
left([NAME], 4) in ('MA_XX', 'PA_XX', 'MA_XX', 'PA_XX', 'MA_XX'.....')
and [QUALITY] = 192 and [QUALITY] is not null
and right([NAME],len([NAME])-5) not in
('4350.CH.DowntimeReason', 'C600.DowntimeReason', 'C600.Speed'
, 'Maker.DowntimeReason', 'Maker.Reject_Procent', 'Maker.Speed', 'Maker.Status', 'X2.Downti
meReason'
, 'X3.DowntimeReason', 'X3.Speed', 'X3.Status', 'Maker.Reject_TotalRejectsPercent', 'CasePa
cker.DowntimeReason'
, 'CasePacker.Status')
and DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), ov.TIMESTAMP) >= dateadd(hour,-
2,getdate())
)
,
```

```

cte as ( -- virtuální tabulka (CTE)
SELECT
  Start.PoradiRadkuOblasti as StartPoradiRadkuOblasti, Finish.PoradiRadkuOblasti as
  FinishPoradiRadkuOblasti
  , Start.TimeStampInLocalTime as StartT, Finish.TimeStampInLocalTime as FinishT
  , Finish.Part, Start.eq as Starteq, Finish.DATE as FinishDate, Start.Oblast as StartOblast
  , Start.ShiftEndDate as StartShiftEndDate, Start.VALUE as StartValue, Finish.VALUE as
  FinishValue
  , Start.shift as Startshift, Finish.shift as Finishshift
  , case when (cast(Finish.VALUE as bigint) - cast(Start.VALUE as bigint)) > 0
    then (cast(Finish.VALUE as bigint) - cast(Start.VALUE as bigint)) else 0 end as
  DeltaValue
  -- zjistí rozdíl následujících řádků ve sloupci VALUE
  , Start.Quality as StartQuality , Finish.UTC_TimeStamp as UTC_TimeStamp
FROM tmp as Start right join tmp as Finish
  on Start.PoradiRadkuOblasti = Finish.PoradiRadkuOblasti-1 and Start.Oblast=
  Finish.Oblast
  -- napojí virtuální tabulku tmp na virtuální tabulku tmp posunutou o 1 řádek
where Start.Quality is not null and Finish.Quality is not null
)
, cte2 as -- virtuální tabulka (CTE)
(
SELECT
  datepart(hour, FinishT) as Hour , datepart(MINUTE, FinishT) as Minute
  -- zjistí minutu/hodinu ze sloupce FinishT
  , Starteq, FinishDate, StartOblast , StartShiftEndDate, StartValue, FinishValue
  , Startshift, Finishshift, StartQuality
  , sum(DeltaValue) as DeltaValue -- sečte hodnoty ze sloupce DeltaValue
  , UTC_TimeStamp, Part
from cte

group by -- určí sloupce dle kterých provede součet
  Starteq, FinishDate, StartOblast , StartShiftEndDate, StartValue, FinishValue
  , Startshift, Finishshift, StartQuality , datepart(hour, FinishT), datepart(MINUTE, FinishT)
  , UTC_TimeStamp, Part
),
comes as -- virtuální tabulka (CTE), zajistí informace o zakázkách pro PA a MA
(
SELECT
  KM.Name, PO.Number, PO.RealStart, PO.RealEnd, MD.Code, MD.Name AS Descr

FROM XXX.[data].[KappaProcessOrder] PO INNER JOIN XXX.[data].[KappaMachine] KM ON
PO.KappaMachineId = KM.p_id
  INNER JOIN
  XXX.[data].[KappaMaterialDefinition] MD ON PO.KappaMaterialDefinitionId = MD.p_id

WHERE (PO.RealStart > GETDATE() - 20) and left(KM.Name,2) = 'PA'
  --
union all -- k informacím pro PA přidá informace o MA

SELECT
  KM.Name, PO.Number, PO.RealStart, PO.RealEnd, MD.Code, MD.Name AS Descr

FROM XXX.[data].[KappaProcessOrder] PO INNER JOIN XXX.[data].[KappaMachine] KM
ON PO.KappaMachineId_ScondMachineId = KM.p_id
  INNER JOIN
  XXX.[data].[KappaMaterialDefinition] MD ON PO.KappaMaterialDefinitionId = MD.p_id

WHERE (PO.RealStart > GETDATE() - 20) and left(KM.Name,2) = 'MA'
)
, final as ( -- virtuální tabulka (CTE)

```



```

Select
    Hour, Minute, Starteq as Equipment, FinishDate as Date, Part, null as Oblast
, StartShiftEndDate as ShiftEndDate, StartValue, FinishValue , Finishshift as Shift
, StartQuality as Quality, DeltaValue as value, UTC_TimeStamp, comes.Number as PO
, comes.RealStart as Start, comes.RealEnd as Konec, comes.Code as FA, comes.Descr as
popis

from cte2 left join comes on Starteq = comes.Name and UTC_TimeStamp between
comes.RealStart
and isnull(comes.RealEnd, getUTCdate())
) -- propojí informace o skartu s informacemi o zakázkách
, konec as -- virtuální tabulka, která sečte hodnoty sloupce VALUE po hodinách
(
SELECT

final.[Hour],final.[Equipment],final.[Date],final.[Part],final.[Oblast],final.[ShiftEn
dDate]
,final.[Shift],final.[PO],final.[FA],final.[popis]
, sum(final.[value]) as SumHour, max(final.[UTC_TimeStamp]) as TimeLoad

FROM final

group by -- určí sloupce, dle kterých provede součet
    final.[Hour],final.[Equipment],final.[Date],final.[Part],final.[Oblast]
, final.[ShiftEndDate],final.[Shift],final.[PO],final.[FA],final.[popis]
)
insert into [dbo].[OPC_R_Table] ( -- vloží určené hodnoty do tabulky OPC_R_Table
R.[Hour],R.[Equipment],R.[Date],R.[Part],R.[Oblast],R.[ShiftEndDate],R.[Shift]
,R.[PO],R.[FA],R.[popis],R.[SumHour],R.[TimeLoad] )

select

Konec.[Hour],Konec.[Equipment],Konec.[Date],Konec.[Part],Konec.[Oblast],Konec.[ShiftEn
dDate]
,Konec.[Shift],isnull(Konec.[PO],1),Konec.[FA],Konec.[popis],Konec.SumHour,Konec.TimeL
oad

from Konec

left join [dbo].[OPC_R_Table] R on Konec.TimeLoad = R.[TimeLoad]
and Konec.Part = R.Part and Konec.Equipment = R.Equipment

where Konec.TimeLoad < dateadd(HH,DATEPART(HH,GETUTCDATE()),cast(cast(GETUTCDATE() as
date) as datetime))
and R.TimeLoad is null
order by Konec.TimeLoad desc

```

Příloha č. 2: Validace - X3

```
with cte as -- virtuální tabulka (CTE)
(
SELECT *
FROM
(SELECT
[TIMESTAMP]
,DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), TIMESTAMP) AS TimeStampInLocalTime
-- zjistí lokální čas
, [eq],[part]
,convert(int,[VALUE]) as val --převede sloupev Value na datatype int
FROM [SPA_local].[dbo].[OPCValues]

where [part] like 'x3%' and eq = 'PA_XX' --vybere pouze požadovanou část stroje
)
as sourceTable -- Převede požadované řádky sloupce part na
sloupce
PIVOT(max([val]) for [part] in (
[X3.Rejected Production],[X3.Incomplete Cigaret Pockets],[X3.Second 5th-6th Wheel
Jam]
,[X3.First 5th-6th Wheel Jam],[X3.Packets With Dried Glue: Blank],[X3.Flexible Wall
Jam]
,[X3.5th Wheel Rear Wrap Check],[X3.Jam At Exit Drum],[X3.2nd-3rd Wheel Jam]
,[X3.Total Cigarette Pocket Rejections],[X3.Loose Ends],[X3.Total Blank Rejections]
,[X3.Skart],[X3.Total 3rd Wheel Rejects],[X3.Cigarette Jams In Pockets]
,[X3.No Blank On 5th Wheel],[X3.Pull Absence],[X3.No Inner-Frame On 4Th Wheel]
,[X3.Packets With Dried Glue: Flap],[X3.Innerframe With Centering Error],[X3.Dry Glue
Rejections]
,[X3.Foil Slips Out Of Position],[X3.Lower Foil Missing],[X3.Blank Gluer Out Of Phase]
)) as p
)
,cte1 as --virtuální tabulka (CTE)
(
select
row_number() over(order by timestamp) as seq -- vytvoří sloupec ID (čísla řádků)
,timestamp,TimeStampInLocalTime,eq,[X3.Rejected Production],[X3.Incomplete Cigaret
Pockets]
,[X3.Second 5th-6th Wheel Jam],[X3.First 5th-6th Wheel Jam],[X3.Packets With Dried
Glue: Blank]
,[X3.Flexible Wall Jam],[X3.5th Wheel Rear Wrap Check],[X3.Jam At Exit Drum]
,[X3.2nd-3rd Wheel Jam],[X3.Total Cigarette Pocket Rejections],[X3.Loose Ends]
,[X3.Total Blank Rejections],[X3.Skart],[X3.Total 3rd Wheel Rejects]
,[X3.Cigarette Jams In Pockets],[X3.No Blank On 5th Wheel],[X3.Pull Absence]
,[X3.No Inner-Frame On 4Th Wheel],[X3.Packets With Dried Glue: Flap],[X3.Innerframe
With Centering Error]
,[X3.Dry Glue Rejections],[X3.Foil Slips Out Of Position],[X3.Lower Foil Missing]
,[X3.Blank Gluer Out Of Phase]

from cte -- virtuální tabulka, která zjistí rozdíl následujících řádků
)
select top (1000)
a.timestamp,a.TimeStampInLocalTime
,DATEPART(mi, a.TimeStampInLocalTime) as MIN ,DATEPART(hour, a.TimeStampInLocalTime)
AS HOUR
,day(a.TimeStampInLocalTime) as DAY -- zjistí den/hodinu/minutu ze sloupce TIMESTAMP
,a.eq,a.[X3.Rejected Production]-b.[X3.Rejected Production] as TotalReject
,a.[X3.Incomplete Cigaret Pockets]-b.[X3.Incomplete Cigaret Pockets]
as IncompleteCigaretPockets
,a.[X3.Second 5th-6th Wheel Jam]-b.[X3.Second 5th-6th Wheel Jam] as
Second5th_6thWheelJam
```

```

,a.[X3.First 5th-6th Wheel Jam]-b.[X3.First 5th-6th Wheel Jam] as First5th_6thWheelJam
,a.[X3.Packets With Dried Glue: Blank]-b.[X3.Packets With Dried Glue: Blank]
as PacketsWithDriedGlueBlank,a.[X3.Flexible Wall Jam]-a.[X3.Flexible Wall Jam] as
FlexibleWallJam
,a.[X3.5th Wheel Rear Wrap Check]-b.[X3.5th Wheel Rear Wrap Check] as
th5WheelRearWrapCheck
,a.[X3.Jam At Exit Drum]-b.[X3.Jam At Exit Drum] as JamAtExitDrum
,a.[X3.2nd-3rd Wheel Jam]-b.[X3.2nd-3rd Wheel Jam] as nd2_3rdWheelJam
,a.[X3.Total Cigarette Pocket Rejections]-b.[X3.Total Cigarette Pocket Rejections]
as TotalCigarettePocketRejections,a.[X3.Loose Ends]-b.[X3.Loose Ends] as LooseEnds
,a.[X3.Total Blank Rejections]-b.[X3.Total Blank Rejections] as TotalBlankRejections
,a.[X3.Skart]-b.[X3.Skart] as Skart
,a.[X3.Total 3rd Wheel Rejects]-b.[X3.Total 3rd Wheel Rejects] as Total3rdWheelRejects
,a.[X3.Cigarette Jams In Pockets]-b.[X3.Cigarette Jams In Pockets] as
CigaretteJamsInPockets
,a.[X3.No Blank On 5th Wheel]-b.[X3.No Blank On 5th Wheel] as NoBlankOn5thWheel
,a.[X3.Pull Absence]-b.[X3.Pull Absence] as PullAbsence
,a.[X3.No Inner-Frame On 4Th Wheel]-b.[X3.No Inner-Frame On 4Th Wheel] as
NoInner_FrameOn4ThWheel
,a.[X3.Packets With Dried Glue: Flap]-b.[X3.Packets With Dried Glue: Flap] as
PacketsWithDriedGlueFlap
,a.[X3.Innerframe With Centering Error]-b.[X3.Innerframe With Centering Error]
as InnerframeWithCenteringError,a.[X3.Dry Glue Rejections]-b.[X3.Dry Glue Rejections]
as DryGlueRejections
,a.[X3.Foil Slips Out Of Position]-b.[X3.Foil Slips Out Of Position] as
FoilSlipsOutOfPosition
,a.[X3.Lower Foil Missing]-b.[X3.Lower Foil Missing] as LowerFoilMissing
,a.[X3.Blank Gluer Out Of Phase]-b.[X3.Blank Gluer Out Of Phase] as
BlankGluerOutOfPhase

from cte1 a
join cte1 b on a.seq = b.seq+1 -- napojední virtuální tabulky cte1 na cirtuální
tapulku cte1 posunutou o 1 řádek

where a.[X3.Rejected Production]-b.[X3.Rejected Production] >= 0 --podmínka pro
případ nulování čítače

order by TimeStampInLocalTime desc -- srovná řádky podle času

```

Příloha č. 3: Validace - výrobní stroj

```
with cte as -- virtuální tabulka (CTE)
(
SELECT *
FROM
(SELECT
[TIMESTAMP]
,DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), TIMESTAMP) AS TimeStampInLocalTime
-- zjistí lokální čas
,[eq] ,[part]
,convert(int,[VALUE]) as val --převede sloupec Value na datatype int
FROM [SPA_local].[dbo].[OPCValues]
where eq = 'MA_XX' and part not like '%Percent%'
--vybere pouze požadovanou část stroje/umožní convert
)
as sourceTable -- Převede požadované řádky sloupce part na sloupce
PIVOT(max([val]) for [part] in (
[Maker.Reject_HardSpotsCnt],[Maker.Reject_MissingFilterCnt],[Maker.Reject_VentilationCnt]
,[Maker.Reject_OTISCnt],[Maker.Reject_HeavyWeightCnt],[Maker.Reject_LESCnt]
,[Maker.Reject_LightWeightCnt],[Maker.Reject_AirTightnessCnt],[Maker.Reject_LightEndsCnt]
,[Maker.Reject_SoftSpotsCnt],[Maker.GoodProduction],[Maker.Reject_TotalRejects]
)) as p
)
,cte1 as -- virtuální tabulka (CTE)
(
select
row_number() over(order by timestamp) as seq -- vytvoří sloupec ID (čísla řádků)
,[timestamp],[TimeStampInLocalTime],[eq],[Maker.Reject_HardSpotsCnt]
,[Maker.Reject_MissingFilterCnt],[Maker.Reject_VentilationCnt],[Maker.Reject_OTISCnt]
,[Maker.Reject_HeavyWeightCnt],[Maker.Reject_LESCnt],[Maker.Reject_LightWeightCnt]
,[Maker.Reject_AirTightnessCnt],[Maker.Reject_LightEndsCnt],[Maker.Reject_SoftSpotsCnt]
]
,[Maker.GoodProduction],[Maker.Reject_TotalRejects]
]
)
from cte -- virtuální tabulka, která zjistí rozdíl mezi následujícími řádky
)
select top (2000)
a.timestamp,a.TimeStampInLocalTime
,DATEPART(mi, a.TimeStampInLocalTime) as MIN,DATEPART(hour, a.TimeStampInLocalTime) AS HOUR
,day(a.TimeStampInLocalTime) as DAY -- zjistí den/hodinu/minutu ze sloupce
TimeStampInLocalTime
,a.eq,a.[Maker.Reject_HardSpotsCnt]-b.[Maker.Reject_HardSpotsCnt] as HardSpots
,a.[Maker.Reject_MissingFilterCnt]-b.[Maker.Reject_MissingFilterCnt] as MissingFilter
,a.[Maker.Reject_VentilationCnt]-b.[Maker.Reject_VentilationCnt] as Ventilation
,a.[Maker.Reject_OTISCnt]-b.[Maker.Reject_OTISCnt] as OTIS
,a.[Maker.Reject_HeavyWeightCnt]-b.[Maker.Reject_HeavyWeightCnt] as HeavyWeight
,a.[Maker.Reject_LESCnt]-a.[Maker.Reject_LESCnt] as LES
,a.[Maker.Reject_LightWeightCnt]-b.[Maker.Reject_LightWeightCnt] as Lightweight
,a.[Maker.Reject_AirTightnessCnt]-b.[Maker.Reject_AirTightnessCnt] as AirThickness
,a.[Maker.Reject_LightEndsCnt]-b.[Maker.Reject_LightEndsCnt] as LightEnds
,a.[Maker.Reject_SoftSpotsCnt]-b.[Maker.Reject_SoftSpotsCnt] as SoftSpots
,a.[Maker.GoodProduction]-b.[Maker.GoodProduction] as Production
,a.[Maker.Reject_TotalRejects]-b.[Maker.Reject_TotalRejects] as TotalRejct
```

```

from cte1 a
join cte1 b on a.seq = b.seq+1    -- napojí virtuální tabulku cte1 na cte1 posunutě o
1

where a.[Maker.GoodProduction]-b.[Maker.GoodProduction] >= 0    --podmínka pro případ
nulování čítače

order by TimeStampInLocalTime desc    -- srovná řádky podle času

```

Příloha č. 4: Maker_LastShift_RR

```

with cte01 as -- virtuální tabulka (CTE)
(
SELECT TOP (1000) -- top redukuje počet řádků
ROW_NUMBER() OVER(partition by PO order by TimeLoad desc) as ID --očísluje řádky
,[Hour],[Equipment],[Date],[Part],[Oblast],[ShiftEndDate],[Shift]
,[PO],[FA],[popis],[SumHour],[TimeLoad]
FROM [SPA_local].[dbo].[OPC_R_Table]
Where Equipment = 'MA_XX' and part like '%tot%rej%' and TimeLoad > DATEADD(Day,-
2,GETDATE())
-- podmínka where vyfiltruje pouze požadované řádky
)
,cte02 as -- vyrtuální tabulka (CTE) zjistí čas začátku zakázky
(
SELECT *
FROM cte01
Where ID = 1
)
,cte03 as -- virtuální tabulka (CTE)
(
SELECT
a.[HOUR],a.[Equipment],a.[Date],a.[Part],a.[ShiftEndDate]
,a.[Shift],a.[PO],a.[FA],a.[popis],a.[SumHour]
,b.[timeLoad] --vypíše čas začátku výroby ke všem zápisů zakázky

FROM dbo.OPC_R_Table a
JOIN cte02 b on a.PO = b.PO -- spojení vyrtuální tabulky cte02 s OPC__R_Table
)
,cte as -- virtuální tabulka (CTE)
(
SELECT *
FROM
(SELECT
[Hour],[Equipment],[Date]
,DENSE_RANK() OVER(ORDER BY timeLoad) as IDD -- vytvoří skupiny orderů
,[ShiftEndDate],[Shift],[PO],[FA]
,DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), TimeLoad) AS TimeLoadInLocalTime
-- převede TimeLoad na lokální čas
,[Part],[SumHour]

FROM cte03

Where Equipment = 'MA_XX'
)
)

```

```

as sourceTable          -- převede řádky na sloupce (vybrané pouze 2 - Total
Production, Total reject)
Pivot(AVG([sumHour]) for [part] in (
  [Maker.GoodProduction] , [Maker.Reject_TotalRejects]
)) as p
)
,cte1 as -- virtuální tabulka (CTE)
(
SELECT
  [Equipment],[ShiftEndDate],[Shift],[PO],[FA]
, sum([Maker.GoodProduction]) as production -- součet sloupce GoodProduction po
hodinách
, sum([Maker.Reject_TotalRejects]) as Reject -- součet sloupce Total_Reject po hodinách
, IDD

FROM cte
GROUP BY -- určí sloupce podle kterých bude sčítat
  [Equipment],[ShiftEndDate],[Shift],[PO],[FA],[IDD]
)
SELECT TOP (5) -- vypíše maximálně 5 řádků (pro max 5 zakázek)
  [Equipment],[ShiftEndDate],[Shift],[PO],[FA]
, IDD -- slouží k zobrazení dle správného pořadí zakázek.
, ProductionTH = case when production = 0 then NULL else production/1000 end
-- změní 0 na NULL -umožní dělení nulou (převede produkci z kusů na 1000 kusů)
, reject as RejectPC
, cast(round((reject*100.0/(case when production = 0 then NULL else production+reject
end)),2)
as numeric(5,2)) as RR
-- zjistí procento skartu
, active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift]
then 1 else 0 end -- určí aktivní směnu
FROM cte1
Where ShiftEndDate >= convert(date,dateadd(hour,-6,getdate())) and
(Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H' when datepart(hour,getdate())
between 0 and 5 then 'H' else NULL end)=[Shift] then 1 else 0 end) = 0
-- filtruje pouze požadované řádky

ORDER BY IDD desc -- seřadí podle správného pořadí zakázek

```

Příloha č. 5: X3_LastShift_RR

```
with cte as -- virtuální tabulka (CTE)
(
Select *
From
(SELECT
[Hour],[Equipment],[Date],[ShiftEndDate],[Shift],[PO],[FA]
,DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), TimeLoad) AS TimeLoadinLocalTime
-- převede TimeLoad na lokální čas
,[Part],[SumHour]
FROM [SPA_local].[dbo].[OPC_R_Table]

where Equipment = 'PA_XX'
)
as sourceTable -- převede řádky na sloupce (vybrané pouze 2 - Total Production,
Total reject)
Pivot(AVG([sumHour]) for [part] in (
[X3.Prod] ,[X3.Rejected Production]
)) as p
)
,cte1 as -- virtuální tabulka
(
select
[Equipment],[ShiftEndDate],[Shift],[PO],[FA]
,sum([X3.Prod]) as production -- součet sloupce X3.Prod po hodinách
,sum([X3.Rejected Production]) as Reject -- součet sloupce X3.Rejected Production po
hodinách

from cte

group by -- vybere sloupce podle který sčítá
[Equipment],[ShiftEndDate],[Shift],[PO],[FA]
)
select top (13) -- Select maximálně 13 řádků
[Equipment],[ShiftEndDate],[Shift],[PO],[FA]
,Production_Pack = case when production = 0 then NULL else production end
-- změni 0 na NULL - umožní dělení nulou
,reject as Reject_Pack
,cast(round((reject*100.0/(case when production = 0 then NULL else production+reject
end)),2)
as numeric(5,2)) as RR -- zjistí procento skartu
,active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift] then 1
else 0 end
-- určí aktivní směnu

from cte1
where ShiftEndDate >= convert(date,dateadd(hour,-6,getdate())) and
(Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H' when datepart(hour,getdate())
between 0 and 5 then 'H' else NULL end)=[Shift] then 1 else 0 end) = 0
-- filtruje pouze požadovaná data
```

Příloha č. 6: MA_LastShift_RR_Top3

```
with cte as -- virtuální tabulka (CTE)
(
SELECT TOP (100) -- vybere top 100 řádků

    [Equipment]
    ,ROW_NUMBER() OVER (PARTITION BY FA ORDER BY sum([SumHour]) DESC) AS ID
    -- očísluje kritéria po FA
    ,RIGHT([Part],CHARINDEX('_',(REVERSE([Part]))) -1) as PART -- zkrátí Part o
    Maker.Reject_
    ,[Oblast],[ShiftEndDate],[Shift],[PO],[FA],[popis]
    ,sum([SumHour]) as RejectPc -- sečte SumHour za celou předchozí směnu po PO
    ,active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
    when datepart(hour,getdate()) between 18 and 23 then 'H'
    when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift] then 1
    else 0 end
    -- určí aktivní směnu
    ,CONVERT(decimal(4,1),DENSE_RANK() OVER(ORDER BY PO)) as IDD --převede IDD na datatype
    decimal

FROM [SPA_local].[dbo].[OPC_R_Table]

Where Equipment = 'MA_XX' and part not in
('Maker.GoodProduction','Maker.Reject_TotalRejects'
,'Maker.Reject_TotalWasteCnt') and
(Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift] then 1
else 0 end) =0
and ShiftEndDate >= convert(date,dateadd(hour,-6,getdate()))
--podmínka where vyfiltruje pouze požadované kategorie a předchozísměnu
group by --vybere sloupce podle kterých sčítá
[Equipment],[Part],[Oblast],[ShiftEndDate],[Shift],[PO],[FA],[popis]
order by ShiftEndDate desc, RejectPc desc -- seřadí podle počtu výhozů
)
select *
,IDDD = case when PO = 1 then IDD-1.5 else IDD end -- srovná zakázky ve správném
pořadí
--(IDDD v reportu nyní nevyužito, ale je možné použít k zjištění skartu během
přejezdu)
from cte

where id < 4 -- vybere 3 nejčastější kritéria pro každé FA

order by IDDD desc -- seřadí podle pořadí zakázek
```


Příloha č. 7: X3_LastShift_RR_Top3

```
with cte as -- virtuální tabulka
(
SELECT TOP (9) -- vybere maximálně 9 příčin (3*3FA) Pouze v případě více zakázek na
jedné směně
[Equipment]
,ROW_NUMBER() OVER (PARTITION BY FA ORDER BY sum([SumHour]) DESC) AS ID
-- očísluje kritéria po FA
,RIGHT([Part],CHARINDEX('.',(REVERSE([Part]))) -1) as PART
-- vybere pouze požadovanou část ze sloupce Part
,[Oblast],[ShiftEndDate],[Shift],[PO],[FA],[popis]
,sum([SumHour]) as RejectPc --součet po příčinách
,active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift] then 1
else 0 end
-- určí aktivní směnu

FROM [SPA_local].[dbo].[OPC_R_Table]

where Equipment = 'PA_XX' and part like 'X3%' and Part not in ('X3.Prod','X3.Rejected
Production'
,'X3.Reset','X3.Skart','X3.Total 3rd Wheel Rejects'
,'X3.Total Cigarette Pocket Rejections') and (Case when (case when
datepart(hour,getdate())
between 6 and 17 then 'G' when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift]
then 1 else 0 end) =0 and ShiftEndDate >= convert(date,dateadd(hour,-6,getdate()))
--podmínka where vyfiltruje pouze požadované kategorie a kativní směnu

group by --vybere sloupce podle kterých sčítá
[Equipment],[Part],[Oblast],[ShiftEndDate],[Shift],[PO],[FA],[popis]

order by ShiftEndDate desc ,RejectPc desc -- seřadí podle počtu výhozů
)
select * from cte
where id < 4 -- vybere 3 nejčastější kritéria pro každé FA
order by FA, ID -- seřadí podle FA a ID
```

Příloha č. 8: MA_ActiveShift_RR

```
with cte as -- virtuální tabulka (CTE)
(
Select *
From
(SELECT
[Hour],[Equipment],[Date],[ShiftEndDate],[Shift],RIGHT([PO],9) as PO,[FA]
,DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), TimeLoad) AS TimeLoadinLocalTime
-- převede TimeLoad na lokální čas
,[Part],[SumHour]
FROM [SPA_local].[dbo].[OPC_R_Table]

where Equipment = 'MA_XX'
)
as sourceTable -- převede řádky na sloupce (vybrané pouze 2 -Total Production, Total
reject)
Pivot(AVG([sumHour]) for [part] in (
[Maker.GoodProduction] ,[Maker.Reject_TotalRejects]
)) as p
)
,cte1 as
(
select
[Hour],[Equipment],[Date],[Shift],[PO],[FA]
,dateadd(hour, datepart(hour,[TimeLoadinLocalTime])
,CAST(convert(varchar(20),[TimeLoadinLocalTime],112) as datetime)) as cas
-- převede date time na celé nižší hodiny (Nutné pro sčítání zápisů po hodinách)
,sum([Maker.GoodProduction]) as production
,sum([Maker.Reject_TotalRejects]) as Reject -- součet Total Reject po hodinách

from cte

group by --vybere sloupce podle kterých sčítá
[Hour],[Equipment],[Date],[Shift],[PO],[FA]
,dateadd(hour, datepart(hour,[TimeLoadinLocalTime])
,CAST(convert(varchar(20),[TimeLoadinLocalTime],112) as datetime))
)
select top (13) -- Select maximálně 12 řádků
[Hour],[Equipment],[Date],[Shift],[PO],[FA],cas
,ProductionTH = case when production = 0 then NULL else production/1000 end
-- změni 0 na NULL - umožní dělení nulou (převede produkci z kusů na 1000 kusů)
,reject as RejectPC
,cast(round((reject*100.0/(case when production = 0 then NULL else production+Reject
end)),2)
as numeric(5,2)) as RR -- zjistí procento skartu
,active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift] then 1
else 0 end
-- určí aktivní směnu

from cte1
where cas >= (dateadd(hour,-13,getdate())) and (Case when (case when
datepart(hour,getdate())
between 6 and 17 then 'G' when datepart(hour,getdate())
between 18 and 23 then 'H' when datepart(hour,getdate()) between 0 and 5 then 'H' else
NULL end)=[Shift]
then 1 else 0 end) = 1
and po <> 1 -- filtruje pouze požadovaná data
order by cas desc, PO desc -- seřadí podle sloupce cas a PO
```

Příloha č. 9: X3_ActiveShift_RR

```
with cte as -- virtuální tabulka (CTE)
(
Select *
From
(SELECT
[Hour],[Equipment],[Date],[ShiftEndDate],[Shift],RIGHT([PO],9) as PO,[FA]
,DATEADD(mi, DATEDIFF(mi, GETUTCDATE(), GETDATE()), TimeLoad) AS TimeLoadinLocalTime
-- převede TimeLoad na lokální čas
,[Part],[SumHour]
FROM [SPA_local].[dbo].[OPC_R_Table]

where Equipment = 'PA_XX'
)
as sourceTable -- převede řádky na sloupce (vybrané pouze 2 - Total Production, Total
reject)
Pivot(AVG([sumHour]) for [part] in (
[X3.Prod] ,[X3.Rejected Production]
)) as p
)
,cte1 as -- virtuální tabulka (CTE)
(
select
[Hour],[Equipment],[Date],[Shift],[PO],[FA]
,dateadd(hour, datepart(hour,[TimeLoadinLocalTime]),CAST(convert(varchar(20)
,[TimeLoadinLocalTime],112) as datetime)) as cas
-- převede date time na celé nižší hodiny (Nutné pro sčítání zápisů po hodinách)
,sum([X3.Prod]) as production -- součet hodnot ve sloupci X3.Prod
,sum([X3.Rejected Production]) as Reject -- součet sloupce Total_Reject
from cte
group by -- vybere sloupce podle kterých sčítá
[Hour],[Equipment],[Date],[Shift],[PO],[FA]
,dateadd(hour, datepart(hour,[TimeLoadinLocalTime]),CAST(convert(varchar(20)
,[TimeLoadinLocalTime],112) as datetime))
)
select top (13) -- Select maximálně 12 řádků
[Hour],[Equipment],[Date],[Shift],[PO],[FA],cas
,Production_Pack = case when production = 0 then NULL else production end
-- změni 0 na NULL - umožní dělení nulou
,reject as Reject_Pack
,round((reject*100.0/(case when production = 0 then NULL else production+Reject
end)),2)
as RR -- zjistí procento skartu
,active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=[Shift] then 1
else 0 end
-- určí aktivní směnu

from cte1
where cas >= (dateadd(hour,-13,getdate())) and (Case when (case when
datepart(hour,getdate())
between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H' when datepart(hour,getdate())
between 0 and 5 then 'H' else NULL end)=[Shift]
then 1 else 0 end) = 1
and po<>1 -- filtruje pouze požadované řádky
order by cas desc, PO desc -- seřadí podle sloupců cas a PO
```

Příloha č. 10: MA_ActiveShift_RR_Top3

```
with cte01 as -- virtuální tabulka (CTE)
(
Select TOP(10000)
ROW_NUMBER() OVER(Partition by PO order by timeload desc) as IDD
-- očísluje jednotlivé řádky po PO
,Equipment,Part,Oblast,ShiftEndDate,Shift,PO,FA,Popis
,SumHour,TimeLoad
FROM dbo.OPC_R_Table

where Equipment = 'MA_XX' and part like '%tot%rej%' and TimeLoad > DATEADD(Day,-
2,GETDATE())
-- vybírá pouze z dat za poslední 2 dny, a konkrétní kategorii
)
,cte02 as
(
SELECT * from cte01 where IDD=1 -- nalezne pouze první řádek každé zakázky
)
,cte as -- virtuální tabulka (CTE)
(
SELECT TOP (9) -- vybere maximálně 9 příčin (3*3FA) Pouze v případě více zakázek na
jedné směně
a.[Equipment]
,ROW_NUMBER() OVER (PARTITION BY a.FA ORDER BY sum(a.[SumHour]) DESC) AS ID
-- očísluje kritéria po FA
,RIGHT(a.[Part],CHARINDEX('_',(REVERSE(a.[Part]))) -1) as PART
-- zkrátí Part o Maker.Reject_
,a.[Oblast],a.[ShiftEndDate],a.[Shift],RIGHT(a.[PO],9) as PO
,a.[FA],a.[popis]
,sum(a.[SumHour]) as RejectPc -- sečte sloupec SumHour
,active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=a.[Shift]
then 1 else 0 end -- určí aktivní směnu
,b.IDD -- čas začátku zakázky
,DENSE_RANK() OVER(order by b.TimeLoad) IDDD -- očísluje jednotlivé zakázky

FROM [SPA_local].[dbo].[OPC_R_Table] a
Join cte02 b on a.PO=b.PO -- spojí virtuální tabulky

where a.Equipment = 'ma_XX' and a.part not in
('Maker.GoodProduction','Maker.Reject_TotalRejects'
,'Maker.Reject_TotalWasteCnt') and
(Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G' when
datepart(hour,getdate())
between 18 and 23 then 'H'
when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL end)=a.[Shift]
then 1 else 0 end) =1 and a.ShiftEndDate >=
convert(date,dateadd(hour,+6,getdate())) -- zamezí zobrazení předchozí směny
--podmínka where vyfiltruje pouze požadované řádky
group by a.[Equipment],a.[Part],a.[Oblast]
,a.[ShiftEndDate],a.[Shift],a.[PO],a.[FA]
,a.[popis],b.IDD,b.TimeLoad
order by ShiftEndDate desc, RejectPc desc -- seřadí podle počtu výhozů
)
select *
from cte

where id < 4 -- vybere 3 nejčastější kritéria pro každé FA
Order by IDDD desc -- seřadí podle sloupce IDDD
```

Příloha č. 11: X3_ActiveShift_RR_Top3

```
with cte01 as -- virtuální tabulka
(
Select TOP(10000)
ROW_NUMBER() OVER(Partition by PO order by timeload desc) as IDD
-- očísluje jednotlivé řádky po PO
,Equipment,Part,Oblast,ShiftEndDate
,Shift,PO,FA,Popis,SumHour,TimeLoad
FROM dbo.OPC_R_Table

where Equipment = 'PA_XX' and part like '%tot%rej%' and TimeLoad > DATEADD(Day,-
2,GETDATE())
-- vybírá pouze z dat za poslední 2 dny, a konkrétní kategorii
)
,cte02 as --virtuální tabulka
(
SELECT * from cte01 where IDD=1 --nalezne pouze první řádek každé zakázky
)
,cte as
(
SELECT TOP (9)-- vybere maximálně 9 příčin (3*3FA) Pouze v případě více zakázek na
jedné směně
a.[Equipment]
,ROW_NUMBER() OVER (PARTITION BY a.po ORDER BY sum(a.[SumHour]) DESC) AS ID
-- očísluje kritéria po FA
,RIGHT(a.[Part],CHARINDEX('.',(REVERSE(a.[Part]))) -1) as PART -- zkrátí Part o
X3.Reject_
,a.[Oblast],a.[ShiftEndDate],a.[Shift]
,RIGHT(a.[PO],9) as PO,a.[FA],a.[popis]
,sum(a.[SumHour]) as RejectPc -- sečte sloupec SumHour
,active = Case when (case when datepart(hour,getdate()) between 6 and 17 then 'G'
when datepart(hour,getdate()) between 18 and 23
then 'H' when datepart(hour,getdate()) between 0 and 5 then 'H' else NULL
end)=a.[Shift]
then 1 else 0 end -- určí aktivní směnu
,b.IDD -- očísluje jednotlivé zakázky
,DENSE_RANK() OVER(Order by b.TimeLoad) IDDD

FROM [SPA_local].[dbo].[OPC_R_Table] a
Join cte02 b on a.PO=b.PO -- spojí virtuální tabulky

where a.Equipment = 'PA_XX' and a.part like 'X3%' and a.Part not in
('X3.Prod','X3.Rejected Production'
,'X3.Reset','X3.Skart',
'X3.Total 3rd Wheel Rejects','X3.Total Cigarette Pocket Rejections') and
(Case when (case when datepart(hour,getdate()) between 6 and 17
then 'G' when datepart(hour,getdate()) between 18 and 23 then 'H' when
datepart(hour,getdate())
between 0 and 5 then 'H' else NULL end)=a.[Shift]
then 1 else 0 end) =1 and a.ShiftEndDate >= convert(date,dateadd(hour,+6,getdate()))
-- zamezí zobrazení předchozí směny
--podmínka where vyfiltruje pouze požadované kategorie a kativní směnu
group by -- vybere sloupce podle kterých bude počítat
a.[Equipment],a.[Part],a.[Oblast],a.[ShiftEndDate]
,a.[Shift],a.[PO],a.[FA],a.[popis],b.IDD,b.TimeLoad
order by ShiftEndDate desc, RejectPc desc -- seřadí podle počtu výhozů
)
select * from cte
where id < 4 -- vybere 3 nejčastější kritéria pro každé FA
Order by IDDD desc -- Seřadí podle sloupce IDDD
```

Příloha č. 12: Order_Plan

```
with cte as
(
SELECT TOP (10) -- vypíše maximálně 10 řádků
    [SapOrderNo]
    , [Equipment]
    , [Product]
    , [ActQty]
    , [OrderStartDate]
    , [OrderVol]
    , [Status]
    , stav = case when [status] = 'Production' then 'probíhá' else 'plán' end
-- pojmenuje dle požadavku

FROM DW.[dbo].[DWfactStpPlan]

where Equipment like '%PA_XX%' and ShiftDate <= (GETDATE()+1)
and ShiftDate >=GETDATE()-1 and ActQty = 0
-- vyhledá pouze plán na aktuální dne+1

order by ShiftDate , OrderStartDate
)
Select distinct --vypíše každý řádek pouze jednou
*
from cte
```

Příloha č. 13: LastPO_Overview

```
with cte as -- virtuální tabulka (CTE)
(
SELECT ROW_NUMBER() over( order by timeload) as id -- číslování řádků
, [Hour], [Equipment], [Date], [Part], [Oblast]
, [ShiftEndDate], [Shift]
, [PO]= case when PO = 1 then NULL else PO end--změní neznámé processOrdery (1) na NULL
, [FA], [popis], [SumHour], [TimeLoad]
FROM [SPA_local].[dbo].[OPC_R_Table]

where Equipment in ('PA_XX', 'MA_XX') and part in ('x3.prod', 'X3.Rejected Production'
, 'Maker.goodproduction'
, 'Maker.Reject_TotalRejects', 'C600.real_prod', 'x3.rejectedproduction')
-- filtruje pouze požadované hodnoty
and date > '2018-03-07' and SumHour <>0 and PO <>000101798169
and PO <> 000101800179 and PO <> 000101801128
and PO <> 000101803452 and date not between '2018-05-02' and '2018-05-17'
-- vyřadí zakázky se špatně přiřazeným PO(lze smazat po opravě zápisů v databázi)
)
,cte1 as -- virtuální tabulka (CTE)
(
select
a.id, a. [Hour], a. [Equipment], a. [Date]
, a. [Part], a. [Oblast], a. [ShiftEndDate], a. [Shift]
, a. [PO], a. [FA], a. [popis], a. [SumHour], a. [TimeLoad]
, coalesce(a. FA, b. FA) as FACode
-- pro neznámé FA doplní následující (v případě skartu v době přejezdů-přiřadí k
předchozí zakázce)
, coalesce(a. popis, b. popis) as brand -- stejné
, coalesce(a. po, b. po) as OrderNo -- stejné
from cte a
```

```

left join cte b on a.id = b.id-15 -- posune ID o 15
)
,cte100 as --virtuální tabulka (CTE)
(
SELECT *
,ROW_NUMBER() OVER(partition by PO order by TimeLoad) as IDD -- očísluje zakázky
FROM cte1
)
,cte200 as -- zjistí čas začátku zakázky
(
select *
from cte100
where IDD = 1
)
,cte11 as -- virtuální tabulka (CTE)
(
SELECT *
FROM
(select
a.FAcode ,a.brand,a.OrderNo
,a.[SumHour],a.[Part] ,b.TimeLoad
,DENSE_RANK() OVER(order by b.timeLoad desc) as ID -- očísluje zakázky
from cte1 a
join cte200 b on a.OrderNo=b.PO -- propojí dvě virtuální tabulky
)
as sourceTable -- převede řádky na sloupce (vybrané pouze 2 - Total Production, Total
reject)
Pivot(SUM([sumHour]) for [part] in (
[Maker.GoodProduction] ,[Maker.Reject_TotalRejects]
,[C600.real_prod],[X3.Prod],[X3.Rejected Production]
)) as p
)
,cte12 as -- virtuální tabulka
(
select *
,Row_Number() OVER(partition by FAcode order by FAcode) as IDD -- očísluje FA
from cte11
where ID <> 1 -- zamezí zobrazení aktivní zakázky
)
select *
from cte12
where IDD<4 -- zobrazí maximálně 3 zakázky pro každé FA (optimalizace pro report)

```

Příloha č. 14: Overview_Top3

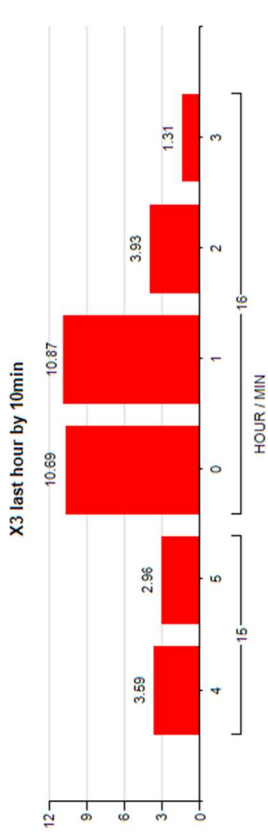
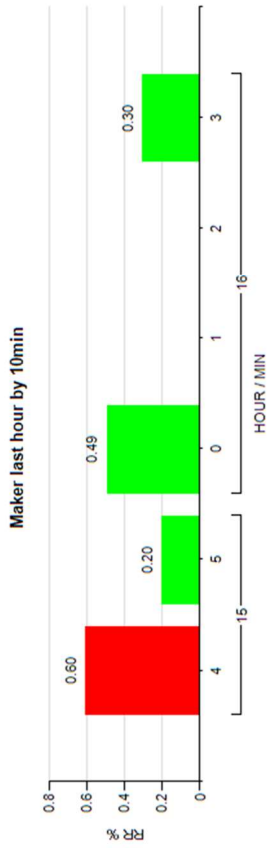
```
with cte as -- virtuální tabulka pro filtrování dat
(
SELECT ROW_NUMBER() over( order by timeload) as id -- číslování řádků
,[Hour],[Equipment],[Date],[Part]
,[Oblast],[ShiftEndDate],[Shift]
,[PO] = case when PO = 1 then NULL else PO end
-- změní nzenámé processOrdery (1) na NULL
,[FA],[popis],[SumHour],[TimeLoad]
FROM [SPA_local].[dbo].[OPC_R_Table]

where Equipment in ('PA_XX','MA_XX') and part not like 'C600%'
and part not in ('Maker.GoodProduction','Maker.Reject_TotalRejects'
,'Maker.Reject_TotalWasteCnt','X3.Prod','X3.Rejected Production','X3.Reset','X3.Skart'
,'X3.Total 3rd Wheel Rejects','X3.Total Cigarette Pocket Rejections')
-- filtruje pouze požadované hodnoty
and date > '2018-03-07' and SumHour <>0 and PO <>000101798169
and PO <> 000101800179 and PO <> 000101801128
and PO <> 000101803452 and date not between '2018-05-02' and '2018-05-17'
-- vyřadí zakázky se špatně přiřazeným PO(lze smazat po opravě zápisů v databázi)
)
,cte1 as -- virtuální tabulka
(
select
a.id,a.[Hour],a.[Equipment],a.[Date]
,a.[Part],a.[Oblast],a.[ShiftEndDate]
,a.[Shift],a.[PO],a.[FA]
,a.[popis],a.[SumHour],a.[TimeLoad]
,coalesce(a.FA,b.FA) as FAcode
-- pro neznámé FA doplní následující (v případě skartu v době přejezdu-přiřadí k
předchozí zakázce)
,coalesce(a.popis,b.popis) as brand -- stejné
,coalesce(a.po,b.po) as OrderNo -- stejné
from cte a
left join cte b on a.id = b.id-15 -- posune ID o 15
)
,cte2 as
(
SELECT
Part,OrderNo,FAcode
,rejectPC= case when part like 'Maker%' then sum(SumHour) else
sum([SumHour])*convert(int,right(brand,2))
end -- převede krabičky na počty cigaret
,brand
,ROW_NUMBER() OVER(partition by orderNo order by (case when part like 'Maker%' then
sum(SumHour)
else sum([SumHour])*convert(int,right(brand,2))
end) desc) as IDD -- očísluje podle počtu výhozů
from cte1

group by -- vybere sloupce podle kterých sčítá
Part,OrderNo,FAcode,brand
)
select * from cte2
where IDD< 4 -- vybere top 3 výhozy
order by OrderNo,rejectPC desc , IDD -- seřadí podle sloupce OrderNo, rejectPC a IDD
podle pořadí zakázek
```

Příloha č. 15: Monitorovací nástroj (zobrazená data byla upravena)

Last hour by 10min



LAST SHIFT

PO: 000101	FA: FA	FA: FA
MAKER		
Production TH: 2,605	X3	Production_Pack: 110,895
Rejected PC: 13,771		Reject_Pack: 2,905
RR%: 0.53		RR%: 2.62
TOP 3		TOP 3
Part	Reject Pc	Part
VentilationCnt	2,223	Incomplete Cigaret Pockets
HardSpotsCnt	802	Blank Gluer Out Of Phase
LESCnt	657	Packets With Dried Glue: Blank
		Reject Pc
		966
		736
		349

ACTUAL SHIFT

MAKER RR % BY HOUR/PO	
PO	FA
1	FA
2	FA
3	FA
1	FA
2	FA
3	FA
MAKER TOP 3	
ID	PO
1	FA
2	FA
3	FA
1	FA
2	FA
3	FA
Part	Reject Pc
VentilationCnt	330
LESCnt	111
HardSpotsCnt	82
VentilationCnt	1,089
HardSpotsCnt	628
AirtightnessCnt	387

X3 RR % BY HOUR/PO	
PO	FA
1	FA
2	FA
3	FA
1	FA
2	FA
3	FA
X3 TOP 3	
ID	PO
1	FA
2	FA
3	FA
1	FA
2	FA
3	FA
Part	Reject Pc
Blank Gluer Out Of Phase	98
Incomplete Cigaret Pockets	67
Packets With Dried Glue: Blank	41
Incomplete Cigaret Pockets	625
Blank Gluer Out Of Phase	383
Packets With Dried Glue: Blank	169

FOLLOWING SHIFT

OrderNo: 000101	MAKER RR %	X3 RR %	TOTAL RR %
	0.29%	1.32%	4.24%
TOP 3			
Part	reject PC		
X3.Loose Ends	15,962		
X3.Blank Gluer Out Of Phase	15,249		
X3.Incomplete Cigaret Pockets	6,578		
OrderNo: 000101	MAKER RR %	X3 RR %	TOTAL RR %
	0.47%	1.21%	3.91%
TOP 3			
Part	reject PC		
X3.Blank Gluer Out Of Phase	43,000		
X3.Packets With Dried Glue: Blank	16,175		
Maker.Reject_LESCnt	11,195		