

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

**Analýza společnosti OV-Media s. r. o. a možnosti zefektivnění
jejího výrobního procesu**

Bakalářská práce

Autor: Michal, Zaccal
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Tereza Otčenášková, BA
Odborný konzultant: Alexandr Paš, programátor, OV-Media s. r. o.

Hradec Králové

Duben 2017

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 26.4.2017

Michal Zecpal

Poděkování:

Děkuji vedoucí mé bakalářské práce Ing. Tereza Otčenášková, BA za metodické vedení práce, cenné rady, připomínky a odborný dohled, které mi pomohly k vypracování bakalářské práce.

Poděkovat bych chtěl také společnosti OV-Media s. r. o. za příjemnou spolupráci a poskytnutí důležitých informací potřebných k vypracování bakalářské práce. V neposlední řadě velmi děkuji své rodině za psychickou podporu, důvěru a poskytnutí možnosti studovat vysokou školu.

Anotace

Téma: Analýza společnosti OV-Media s. r. o. a možnosti zefektivnění jejího výrobního procesu

Bakalářská práce se zabývá analýzou společnosti OV-Media s. r. o. a problematikou zefektivnění jejích procesů, konkrétně v oblasti návrhu tiskařských forem. Bakalářská práce se skládá ze dvou částí, z teoretické a praktické. V teoretické části je kladen důraz na otázku, proč je důležité věnovat se zvyšování efektivity společností a na otázku problému řezání papíru gilotinovými řezy. V této části je dále uvedeno představení dostupných nástrojů pro vývoj aplikace. Praktická část je zaměřena na analýzu stávajících procesů s návrhem a tvorbou aplikace pro zefektivnění procesů v analyzované společnosti. Aplikace zprostředkuje výpočet minimálního počtu řezů potřebných pro rozřezání archu papíru na určitý počet dílů. V závěru práce je shrnuto, jaký má aplikace přínos pro firmu OV-Media s. r. o. Stejně tak je poskytnuto porovnání nového řešení se stávajícím procesem zavedeným ve společnosti.

Annotation

Title: Analysis of the company OV-Media s. r. o. and options of improving its manufacturing processes

The focus of the Bachelor Thesis is the analysis of the OV-Media s. r. o. company and the efficiency of its operating processes, namely of the printing forms design. The Thesis is composed of two parts, theoretical and practical. The theoretical part highlights why it is important to pay attention to the efficiency increase of operating processes and explains the problem of paper cutting using gilotine cuts. It further introduces the available tools for the application development. The practical part is focused on the analysis of current operating processes linked with the design and development of an application for efficiency increase in the company. The application calculates the minimal amount of paper cuts needed to split a paper sheet into the specified number of parts. The contribution of the application for OV-Media s. r. o. is summarized in the conclusion. Furthermore, the developed solution is compared to the current process employed by the company.

Obsah

1	Úvod.....	1
2	Cíl a metodika bakalářské práce.....	2
2.1	Cíl práce.....	2
2.2	Metodika práce.....	2
3	Teoretická východiska	3
3.1	Analýza	3
3.2	Analytické techniky	3
3.2.1	Brainstorming.....	4
3.2.2	SWOT analýza	4
3.2.3	Benchmarking.....	5
3.2.4	Change Management.....	5
3.2.5	Lewin's Three-step Change Theory	6
3.2.6	Appreciative Inquiry	6
3.3	Řízení procesu inovace.....	8
3.4	Inovace	9
3.4.1	Typy inovací.....	10
3.4.2	Analýza inovované oblasti	11
3.4.3	Monitoring.....	11
3.4.4	Plánování	12
3.5	Řezání papíru.....	12
3.5.1	Two-dimensional Cutting Problem (TDC)	13
3.5.2	Two-dimensional cutting stock problem	15
3.6	Jazyk C# a související prostředí .NET	16
3.6.1	Porovnání C# s jinými jazyky	17
3.6.2	Omezení jazyka C#	19

4	Návrh a tvorba aplikace.....	20
4.1	Společnost OV-Media s. r. o.	20
4.1.1	Funkční model společnosti.....	21
4.1.2	Analýza výrobních procesů společnosti OV-Media s. r. o.....	21
4.2	Specifikace aplikace.....	24
4.2.1	Účel aplikace.....	24
4.2.2	Nasazení aplikace.....	24
4.2.3	Analýza vstupních dat.....	24
4.2.4	Analýza problematiky návrhu aplikace.....	29
4.2.5	Analýza C# projektu.....	35
4.3	Návrh aplikace.....	36
4.3.1	Kontrola XML dokumentu.....	37
4.3.2	Zpracování dat.....	37
4.3.3	Kolize objektů na formě.....	39
4.3.4	Separace formy.....	40
4.3.5	Nejmenší počet řezů.....	44
4.3.6	Jazykové mutace.....	46
5	Shrnutí výsledků a jejich diskuze.....	47
5.1	Zpětná vazba společnosti OV-Media s. r. o.	47
5.2	Testování Aplikace.....	48
5.3	Srovnání nového řešení s dosavadním procesem.....	48
5.3.1	Dosavadní proces.....	48
5.3.2	Nové řešení.....	49
6	Závěr.....	49
7	Seznam použité literatury.....	51
8	Přílohy.....	55

Seznam obrázků

Obrázek 1- Optimální návrh řezacího vzoru.....	15
Obrázek 2 - (a) Forma pro gilotinový řez. (b) Forma, která nelze rozřezat pomocí gilotinového řezu.	16
Obrázek 3 - Funkční model firmy OV-Media s. r. o.	21
Obrázek 4 - Varianty skládání papíru.	23
Obrázek 5 - Obal na blu-ray disk.....	26
Obrázek 6 - Výstupní forma procesu návrhu.	27
Obrázek 7 - Stav kdy se objekty na formě se překrývají.....	30
Obrázek 8 - Stav kdy neexistuje žádný možný řez k separaci všech objektů	30
Obrázek 9 - Dva objekty bez vzájemného vztahu pozice.....	32
Obrázek 10 - Skupina objektů ve vzájemném vztahu.	32
Obrázek 11 - Levá forma se špatně rozmístěnými objekty. Pravá forma s ideálně rozmístěnými objekty.	34
Obrázek 12 - Výpočet kolize dvou objektů.	39
Obrázek 13 - Forma s vyznačenými hlavními skupinami na ose X a Y.	40
Obrázek 14 - Základní skupina objektů.	43
Obrázek 15 - Rozdělení formy do základních skupin.	44

1 Úvod

Již od pradávna se člověk snaží zaznamenávat informace, které pro něj mají určitou hodnotu. Tato lidská potřeba začala u jeskynních maleb, pokračovala přes tesání znaků do hliněných destiček a malování na látku až k vynalezení papíru v podobě, jak ho dnes známe, k čemuž došlo přibližně kolem roku 105 n. l. v Číně.

I když je dnes papír považován za samozřejmost, jen málo z nás by si bez něj dokázalo představit život. Proto jsme se ho naučili zpracovávat a používat jej pro přenos informací. Během času bylo toto umění zdokonaleno natolik, že vedlo k vynalezení tisku.

Dnes je po celém světě nespočet společností specializujících se na tisk a zpracování papíru. S neustále rostoucí poptávkou roste i náročnost výroby. Máme tedy možnost zamýšlet se a vynalézat způsoby, jak do práce nejlépe investovat své prostředky tak, abychom maximalizovali kvalitu a případně i kvantitu požadovaného produktu, spolu s uspokojením zákaznickových požadavků.

Jedna ze společností, podnikající v oboru tisku, je firma OV-Media s. r. o. sídlící v jihočeském městě Vimperk. Firma se zabývá tiskem a následným zpracováním vytištěného díla pro zábavní průmysl. Právě díky vřelé spolupráci OV-Media s. r. o. mohla tato bakalářská práce vzniknout.

První část bakalářské práce se zabývá otázkou důležitosti zvyšování efektivity společností v dnešním vysoce konkurenčním prostředí. V návaznosti na analyzovanou společnost je nastíněna problematika řezání papíru při kontinuálních procesech. Dále je zde k nalezení analýza stávajících procesů, spolu se srovnáním dostupných nástrojů a návrhem řešení problému. Praktická část je zaměřena na návrh a tvorbu aplikace pro zefektivnění procesů. Aplikace vypočítá minimální počet řezů potřebných k rozřezání papíru na předem určený počet dílů, který zajistí zvýšení efektivity chodu celé společnosti.

Tato práce je přínosnou především pro společnost OV-Media s. r. o. Přidanou hodnotou by měla být především praktická část bakalářské práce, která by měla napomoci zefektivnění práce v oddělení řezání papíru. Také by měla pomoci vedení společnosti k plánování výroby.

2 Cíl a metodika bakalářské práce

Tato kapitola obsahuje cíl práce a metodiku, která je použita v bakalářské práci.

2.1 Cíl práce

Hlavním cílem bakalářské práce je navrhnout a vytvořit aplikaci sloužící k zefektivnění práce ve společnosti OV-Media s. r. o. Aplikace je schopna výpočtu minimálního možného počtu řezů, který musí řezač provést k rozřezání archu papíru na určité části. Nedílnou součástí úspěšného návrhu je analýza procesů firmy a srovnání použitelnosti dostupných nástrojů. Dílčím cílem je i podrobněji představit programovací jazyk C#, který je využit při tvorbě aplikace.

2.2 Metodika práce

Bakalářská práce, jejímž cílem je analýza společnosti OV-Media s. r. o. a zefektivnění jejího výrobního procesu je rozdělena do dvou hlavních částí. První část je pojmenována jako Teoretická východiska, druhá část jako Návrh a tvorba aplikace. Obě tyto části jsou dále děleny na dílčí podkapitoly.

Teoretická východiska popisují metody, které jsou spojeny s analýzou firemních procesů a inovacemi. Tyto informace vycházejí z relevantní literatury (knih a odborných časopisů) a udávají směr, který je potřebný pro pochopení a úspěšné provedení analýzy ve společnosti OV-Media s. r. o. Dále je zde uveden popis problémů vznikajících při řezání papíru na gilotinových řezacích strojích, který umožňuje lépe si uvědomit problémy pojící se s návrhem samotné aplikace. Část teoretických východisek uzavírá představení jazyka C#. Aplikace pro minimální počet řezů, je psána v tomto jazyce. Seznámení se s tímto jazykem je přínosné

pro práci ve vývojovém prostředí.

V části Návrh a tvorba aplikace je provedena analýza společnosti OV-Media s. r. o., která odkrývá a předchází možným problémům, které by se při návrhu mohly vyskytnout. Dále se zabývá samotným návrhem aplikace, kde jsou popsány a vysvětleny všechny postupy tvorby aplikace. Na samotném konci práce je uvedeno srovnání dosavadního řešení s novým a shrnutí výsledků, které byly získány testy finální verze aplikace.

3 Teoretická východiska

Tato kapitola vychází z literárních zdrojů, které napomáhají pochopit a nalézt řešení návrhu a tvorby aplikace určené ke zvýšení efektivity firemních procesů.

3.1 Analýza

Během posledních čtyřiceti let se vývoj informačních technologií posunul prudce kupředu. S rychlým rozvojem počítačového hardwaru, zvýšením rychlosti zpracování dat a velikosti dostupné paměti, došlo k prudkému rozvoji metod analýzy dat, které dříve nebylo možné využít v praxi vzhledem k požadovanému výpočetnímu výkonu (Agnieszka a spol., 2015).

3.2 Analytické techniky

„Analytická technika (analytická metoda) je postup či způsob provedení rozboru nějakého problému, stavu či skutečnosti.“ (ManagementMania, 2017) Bývá pravidlem, že analytické metody bývají dosti omezeny jak z časové, tak úkolové stránky. To znamená, že jsou využívány pro hledání problému „narázově“. Metody řízení naopak upřednostňují pozvolné nasazení postupu do firemního prostředí. V praxi často bývá zcela normální využívání jednoduchých technik pro analýzu, které slouží analytikům k vykonávání jejich všední práce. Někdy si nemusí uvědomovat to, že by analytické techniky používali, a bývají tedy často bez jakéhokoliv pojmenování. K provedení těchto jednoduchých technik často stačí využití systému „tužka – papír“. Tyto pomůcky jsou k nalezení v každé kanceláři. Tento systém je založen především na zkušenostech lidí, kteří je využívají. Existuje však celá řada specializovaných analytických technik, které nejsou založeny jen na náhodném procesu analýzy, ale jsou naopak z velké většiny postaveny na určitém matematickém modelu nebo technice požadující specifické nástroje.(ManagementMania, 2017). Mezi základní a v praxi nejpoužívanější analytické techniky patří například SWOT analýza, brainstorming a benchmarking, jejichž využití je relevantní jak pro management obecně, tak pro oblast programování a tvorby aplikací.

3.2.1 Brainstorming

Brainstorming jinými slovy znamená skupinovou spolupráci vedoucí k vytváření kreativních nápadů. Za otce brainstormingu je považován Alex Faickney Osborn, který v roce 1939 přišel s touto metodou za účelem kreativního řešení problémů. (Scientific & Academic Publishing Co., 2017) Přestože pozdější vědecké studie některé z Osbornových myšlenek zdiskreditovaly, skupinová tvorba nových nápadů je i v současné době velice oblíbená. Tato metoda se začala používat jak ve školách, tak firmách po celém světě. Brainstorming si velmi oblíbili především studenti, kteří jej využívají ke zdokonalování tvůrčích nápadů (Dziak, 2016). V odvětví informačních technologií je brainstorming s oblibou využíván při návrhu aplikací, kdy se celý vývojářský tým sejde a spontánním vymýšlením návrhu se dopracují k hrubému návrhu aplikace.

3.2.2 SWOT analýza

SWOT analýza je takzvaný brainstorming a plánovací nástroj v jednom, který pomáhá lidem hodnotit projekty či podnikání obecně. Název SWOT je zkratka pro silné a slabé stránky, příležitosti a hrozby. Firmy používají tuto metodu k pochopení vnitřního a vnějšího prostředí podniku. Tato analýza může poskytnout vedení podniku lepší představu o tom, jak zlepšit firemní procesy. Také dokáže být velice přínosnou pro stanovení nových obchodních plánů. Vývojářům poskytuje možnost zjistit nedostatky aplikací, které se mohou skrývat jak v samotném zabezpečení aplikace, tak v uživatelské přívětivosti, která je klíčovou pro spokojenost zákazníka.

Vnitřní faktory, to znamená silné a slabé stránky, zahrnují jak pozitivní, tak negativní aspekty spojené s firmou. Dále poukazují na věci, které v podniku dobře pracují, ale také odhalují ty, které je potřeba opravit, nebo změnit. Každý tento faktor lze v podniku změnit či opravit.

Poté, co jsou zjištěny vnitřní faktory, by se firma měla zaměřit na vnější faktory, tedy příležitosti a hrozby. Každá firma je jimi ovlivněna buď přímo, nebo nepřímo prostřednictvím vnějších prvků, které sama nemůže ovlivnit (Harmon, 2016).

3.2.3 Benchmarking

Benchmarking je metoda, která využívá procesu vyhledávání a následné implementace osvědčených postupů. Nejobtížnější částí této metody je identifikace a definice bodu, podle kterého bude prováděno srovnávání. S touto metodou je spojována společnost Xerox, která se v 80. letech minulého století snažila minimalizovat své výrobní náklady použitím právě benchmarkingu a povedlo se jí tuto metodu rozšířit do různých nejen průmyslových odvětví (Ettorchi-Tardy a spol., 2012). Obecně platí, že srovnání je systematický a trvalý proces, který funguje až po sérii určitých kroků. Tyto kroky vykonává člověk, který je nazýván jako benchmarker. Benchmarker musí v prvním kroku zvolit typ porovnávání, vybrat produkt nebo proces porovnání a identifikovat klíčové ukazatele výkonnosti. Dále je nutné posbírat všechna potřebná data, provést analýzu získaných dat a identifikovat oblasti pro zlepšení. Jakmile je tato fáze hotova, benchmarker podá zprávu o svých zjištěních a sdílí osvědčené postupy s ostatními kolegy. Pokud je to vhodné, vytvoří strategický plán s cílem přizpůsobit a zavést osvědčené postupy benchmarkingu do analyzovaných oblastí. Tyto postupy zavádí podle svých potřeb a podmínek pro zlepšení jednotlivých oblastí. Ve finále by tyto plány měly být realizovány (Mercadal, 2016). Společnosti specializující se na vývoj software touto cestou mohou získat například výhodu inspiraci u konkurence, která vyvíjí podobné aplikace, ale s využitím jiných principů.

3.2.4 Change Management

Jinými slovy řízení změn je oblastí, jejímž hlavním cílem je zaměřit se na změny, na jejich zavádění a prosazování do života organizace. Pochopení problematiky řízení změn v organizačním prostředí napomáhá ke zlepšení organizační strategie a procesů podniku. Rychle se měnící podnikatelské prostředí jednadvacátého století, a to zejména v důsledku globalizace a zvýšeného konkurenčního prostředí, mělo za následek výraznou potřebu firem vedoucí k tomu, aby byly více otevřené organizačním změnám v podniku. Velké změny prostředí se dají velmi snadno pozorovat i u informačních technologií, kde vývoj a vznik nových technologií v poslední době rapidně vzrostl. Pro prosperitu společnosti je zapotřebí s tímto

trendem neustále držet krok. Firmy po celém světě však také musely pochopit, jak proces změny řídit. Existuje několik teorií změny a všechny mají nemálo společného. Některé teorie změny se lidem do podvědomí zapsaly silněji než ostatní. Mezi populární teorie implementace změn patří například Kurt Lewin's tThree-step Change Theory (Lewinův třífázový model změn), Appreciative Inquiry (vnímavé dotazování) a Eight Step Change Model (Model osmi kroků změny) Johna P. Kottera (Benson, 2014).

3.2.5 Lewin's Three-step Change Theory

Kurt Lewin je považován za otce organizačních změn podniku. Právě jeho model změn patří mezi nejstarší a nejznámější modely změn v organizaci. Podle Lewina změna probíhá ve třech fázích:

1. Rozmrazení

Kvůli prvnímu kroku procesu musí být skupina procházející změnou nejprve destabilizována, aby bylo možné ve skupině změnu provést.

2. Změna

Druhý krok umožňuje změnit činitele různých procesů nebo požadované chování do té míry, aby to bylo pro firmu co nejvíce žádoucí.

3. Zamrazení

Jakmile je úspěšně dosaženo předchozího kroku, třetí stupeň změnu zmrazí a stabilizuje ji. Změna se v této fázi stává konzistentní.

(French a spol., 2005)

3.2.6 Appreciative Inquiry

Appreciative Inquiry (vnímavé dotazování) je jeden z modelů změn, který stále získává na popularitě. Tento model změn se soustředí nikoli na selhání jednotlivých sektorů, ale výhradně na úspěchy, ze kterých pak vyvozuje, co je potřeba změnit. Pozitivní přístup tuto metodu činí atraktivní i pro ty, kteří jsou obecně známí svým odporem ke změnám. Jsou tím míněni především zaměstnanci, jejichž pozitivní pohled na pracovní proces přispívá k prosperitě celé firmy (Benson, 2014).

Vnímavé dotazování je přístup, který je jedinečný a vhodný pro organizace, které se snaží spolupracovat jak s vedením, tak se zaměstnanci. Vnímavé dotazování předpokládá skutečnou tendenci společnosti starat se nejen o všechny uvnitř organizace a také o ty, kteří s ní spolupracují (Johnson, 2001). V oblasti organizačního rozvoje, kde probíhá nejvíce transformací, je vnímavé dotazování v současné době nejčastější a nejvíce preferovaná metoda změny. Metoda vnímavého dotazování neslouží pouze jako katalyzátor změn, ale je také nejlépe přijímanou transformační metodou pro ty, kteří se obvykle změnám brání. Pro ředitele softwarové společnosti může být obtížné přistoupit k obměně starých technologií a postupů. Úkolem vnímavého dotazování je tento problém do značné míry vyřešit. Je doporučována konzervativním společnostem, které se drží starých zásad a odmítají jakékoliv inovace. Proces vnímavého dotazování zaručuje různé perspektivy pohledu na problém díky svému pozitivnímu přístupu (Berson, 2014).

3.2.6.1 Model Osmi kroků

John Kotter, profesor Harvard University, je dobře známý jako lídr v oblasti teorie řízení změn. Jeho nejslavnější dílo je jeho Model osmi kroků změny (Kotter, 1998). Implementace změny v jeho podání zahrnuje postupné provedení následujících osmi kroků:

1. Vyvolání vědomí naléhavosti
2. Sestavení koalice schopné prosadit a realizovat změny
3. Vytvoření vize a strategie
4. Komunikace transformační vize
5. Delegování v širokém měřítku
6. Vytváření krátkodobých vítězství
7. Využití výsledků a podpora dalších změn
8. Zakotvení nových přístupů do firemní kultury

Tento systém prozkoumá trh a konkurenční prostředí, identifikuje potencionální krize, vytvoří skupinu dostatečně silnou řídit změnu, přiměje tuto skupinu spolupracovat, vytvoří vizi, která pomůže proces změny řídit a vyvine strategii k dosažení této vize. Dále využije všech potřebných prostředků k nepřetržité komunikaci o nových strategiích, odstraní veškeré překážky, změní systémy

bránící transformaci a podpoří netradiční myšlenky, nápady a postupy. Filozofií této metody, která klade důraz na pozitivní myšlení je také systém vzdělávání a odměn cílící na motivaci zaměstnanců firmy. Ve finále tedy viditelně ocení a odmění osoby, které se na celém procesu podílely, a vzdělává, najme, či povýší osoby, které mají schopnosti transformační vize i nadále realizovat. Takováto firma poté dosahuje lepších výsledků prostřednictvím chování orientovaného na zákazníky a zvyšuje tak svou produktivitu. V neposlední řadě je důraz kladen na výběr vhodných nástupců (ManagementMania, 2017).

Motivace zaměstnanců v měnícím se prostředí není snadný úkol. Lídr v oblasti informačně založené organizace dvacátého prvního století se potýká s množstvím problémů (například udržení vysoké produktivity). Vůdce musí udržet krok s dobou a zapadnout do stylu vedení, který je nejvíce efektivní v neustále se měnícím nestabilním prostředí (Berson, 2014).

3.3 Řízení procesu inovace

Proces inovace je nedílnou součástí obchodní strategie a organizačního profitu každé firmy. Pro správné řízení inovace je dobré uvědomit si vzájemný vztah těchto složek. Inovace, která se týká použití nového produktu, služby, nebo metody v podnikové praxi bezprostředně následující po jeho objevu, ovlivňuje hospodářský úspěch a podíl na silně konkurenčním trhu. Tento proces firmám zvyšuje šanci na zvýšení profitu. V reakci na nové technologie řízené světovými trhy podniky zvýšily využití moderních technologií i své inovační úsilí (Zahra, 1993). V sektoru informačních technologií je inovace nutností. Společnosti, které inovaci, zejména technologickou, odmítnou, nezůstanou dlouhodobě konkurenceschopné. Inovace je spojena s konkurenční výhodou a na rozdíl od většiny jiných obchodních praktik, může změnit konkurenční rovnováhu ve vyspělých trzích (Brown, 1992).

Koncepce a praxe inovace se začala spojovat s ekonomickým ziskem a konkurenční výhodou v třicátých letech dvacátého století. V roce 1930 ekonom Joseph Schumpeter (1883-1950) vytvořil teorii hospodářského rozvoje na základě pěti typů ekonomických inovací:

1. nastavení nebo objevení nového produktu

2. nastavení nebo objevení nového výrobního procesu
3. otevření nového trhu
4. využití nového zdroje
5. vytvoření nových marketingových struktur (Leteneyei, 2001).

Současná teorie tvrdí, že obchodní společnosti musí soutěžit, zachovat, nebo získat podíl na trhu, aby maximalizovaly svůj zisk a staly se konkurenceschopnými. Inovace je považována za klíč k vytvoření konkurenční výhody (Stalk, 2006).

3.4 Inovace

Inovace může být nejjednodušším způsobem definována jako nová myšlenka. Nová myšlenka pro proces obnovy myšlení, nebo činnosti. Proto by právě ona měla být znakem prosperující a stále rostoucí firmy. Správně aplikovaná inovace tedy nepochybně zvyšuje kvalitu produktů. Příkladem může být inovace zdrojového kódu aplikace. Inovací lze dosáhnout čitelnější a funkčnější alternativy stejného řešení s použitím novějších technologií.

Schumpeter ve svých studiích o Teorii systému inovací, dokládá, že inovace, definovaná jako aplikace nových prostředků, kterými mohou být nové technologické postupy, technické prostředky, výrobní procesy a podobně, je strategickým podnětem pro firemní rozvoj (Jáč a spol., 2005).

Převod nových prostředků z oblasti tvorby do oblasti praktického užití je nedílnou součástí inovačního procesu. Tím je myšlen především transfer technologií, který je jednou z hlavních myšlenek této práce. „Transfer technologií je procesem cílevědomého časově ohraničeného převádění poznatků umožňujících inovovat výrobky, výrobní, pracovní a zkušební metody a služby.“ (Kraj Vysočina, 2017).

3.4.1 Typy inovací

V organizacích se vyskytují četné druhy inovací, jako jsou například marketingové inovace, technologické inovace, organizační inovace, inovace produktu, inovace služeb, a procesní inovace.

1. Marketingové inovace

Marketingové inovace se týkají způsobu, při kterém se lidé postupně seznámí s novými myšlenkami a poté je i přijmou. Marketing inovace je proces sociálního učení, který má za následek, že spotřebitelé pomalu mění své postoje a hodnoty. Někteří jedinci mají vyšší sklon vyzkoušet inovativní produkty, než ostatní a rychlost v přijímání takové inovace se může lišit od jednoho podniku k jinému (Brown, 1992).

2. Technologická inovace

Technologická inovace je proces, při kterém průmysl vytváří nové a lepší výrobky nebo výrobní procesy. Zahrnuje činnosti od vytvoření nápadu, výzkumu, vývoje, až k šíření těchto inovovaných produktů v ekonomice. Dále také vyžaduje využití nových technologií a vědeckých poznatků ke zlepšení výkonnosti produktů či výrobních procesů. Selhání vedení technologických inovací má často za následek organizační neefektivnost a frustraci (Bigwood, 2004).

3. Organizační inovace

Organizační inovace může být definována jako proces změny podniku zavedením různých způsobů výroby nebo administrativy. Tato inovace zahrnuje přijetí nápadů z vnějšího prostředí organizace a generování nápadů vlastních. Do této oblasti spadá plánování inovace, výběr a její realizace (Spender a Kessler, 1995).

4. Inovace produktů a výrobních procesů

- Inovace produktu zahrnuje zavedení zboží nebo služby, které jsou nové nebo výrazně zlepšily svou původní podstatu.
- Inovace výrobního procesu zahrnuje zavedení nového nebo podstatně zlepšeného způsobu výroby, anebo přináší takový

proces, jehož způsob aplikace do firemních podmínek není tak náročný (Flynn, 2015).

3.4.2 Analýza inovované oblasti

Analýza je rozborem a zkoumáním spleťtých faktů, které jsou následně rozloženy na skutečnosti elementární. Cílem analýzy je poznání podstaty a zákonitosti těchto skutečností.

Mnohdy k řešení nestačí pojmenovat povahu problému – například říci, že firma má duplicitní procesy nebo že největší pokles obratu nastává ihned při prvním kontaktu se zákazníkem. Je třeba najít místo, kde tyto potíže vznikají a tam je řešit. Tento úkol je mnohdy mnohem těžší, než se na první pohled může zdát. Je třeba mít hodnotné informace o problémové oblasti. Jestliže je možný přístup k informacím o infrastruktuře oblasti, vztazích jednotlivých sektorů, aktuálních problémech a tak dále, naděje pro správnou identifikaci epicentra problému významně stoupají (Plamínek, 2008).

Funkční modely firmy vytvořené na základě těchto informací jsou dalším klíčem k úspěšnému porozumění problémové oblasti. Takovýto model graficky znázorňuje funkce a procesy firmy v definovaném rozsahu. Jestliže je model správně namodelován, napomůže porozumět dané oblasti i lidem, kteří o ní nejsou dostatečně informováni.

3.4.3 Monitoring

„Klíčové nástroje projektového managementu, jako je například záznam o rizicích a problémech, mohou okorat stejně rychle jako chleba. Pokud nejsou pravidelně obnovována, zatuhnou.“ (Barker a Cole, 2007, 44)

Sběr dat není jednorázovým procesem. Sběr dat je neustálým monitorováním zkoumané oblasti spolu s aktualizací dosavadních dat. Udržení aktuálnosti dat přináší mnoho výhod především při plánovacích procesech. Stávají se tak mnohem efektivnějšími. Proces sběru dat s sebou nese rizika spojená například s jejich hodnověrností a relevancí. Ověřovat důvěryhodnost dat a informací si řada lidí – alespoň v profesním životě – zvykla. Kritéria jsou pro tyto aktivity poměrně jasná a známá. U relevance je řada lidí méně ostražitá. Často zahlcujeme sebe

i okolí daty malé nebo nejasné užitečnosti. „Poměrně často posuzují příčiny nefunkčnosti firem a musím říci, že neschopnost filtrovat data a distribuovat na správná místa pouze ta, která zde skutečně potřebují, bývá standardní slabinou komunikace ve firmách téměř pravidelně“ (Plamínek, 2008, 40).

3.4.4 Plánování

Dnešní prostředí různě velkých firem působících v různých oblastech je v literatuře často popisováno jako stále dynamičtější, bouřlivější a nepředvídatelnější, než tomu bylo v dřívějších dobách. Zejména v takovýchto podmínkách začíná být existence dlouhodobých cílů a přijetí nových postupů nezanedbatelnou nutností pro úspěšné podnikání. Hovoříme zde o strategii plánování spolu se strategickými postupy. Bez přijetí těchto požadavků firmy na nabídky pouze reagují, namísto toho, aby je efektivně vytvářely (Drucker, 2008).

Strategické plánování může být definováno jako proces, který slouží společnosti k nalezení strategie přispívající ke zvýšení výkonu a řízení podniku s ohledem na jeho vizi budoucnosti (Tapinos a spol., 2005).

3.5 Řezání papíru

I když se tato problematika může zdát zprvu dosti přímočarou, skrývá mnohem více neznámých. Dost výrazně ovlivňuje výrobní proces některých firem, a to jak po finanční stránce, tak po stránce časové. Nastolení neefektivních, ba i chybných řezacích technik, může vést k masivním finančním ztrátám, nebo dokonce k bankrotu celé firmy.

Materiály jako je například papír, fólie, sklo a plasty jsou obvykle vyráběny v kontinuálním procesu ve větších formátech s tím, že budou později rozděleny na formáty menší. V konkrétním případě výroby papíru se papír vyrábí na větších strojích a poté jsou role papíru zpracovány na řezacích strojích, kde jsou rozřezány na menší rozměry o velikostech objednaných například tiskárnou (Correia a spol., 2012). V kontinuálním procesu výroby pak mohou tiskárny pokračovat. Rozřezané archy papíru výrobnou papíru, přebírají a tisknou na ně zakázky menších formátů. Posléze je papír dále dělen na menší formáty.

3.5.1 Two-dimensional Cutting Problem (TDC)

Problémy při provádění dvourozměrných řezů (dále jako TDC z anglického slova two-dimensional cutting) zahrnují určení řezacího vzoru, který minimalizuje odpad (nebo maximalizuje zisk) při separaci jednotlivých částí z velké obdélníkové desky. Každý takovýto malý obdélníkový kus má vlastní velikost a hodnotu, která přináší firmě profit. Řezací vzor je kombinací pozic všech kusů, které mají být řezány. TDC se dále dělí do skupin pro specifikaci problémů podle určitých kritérií, jejichž přehled je uveden níže.

- **Typ částí**

1. **Constrained TDC problem (CTDC)**

Constrained two-dimensional cutting problem (z anglického jazyka přeloženo jako omezené dvourozměrný řezací problém) omezuje četnosti typů (velikostí) obdélníkových částí určených k řezání.

2. **Unconstrained TDC problem (UTDC)**

Unconstrained two-dimensional cutting problem (z anglického jazyka přeloženo jako neomezené dvourozměrný řezací problém) nastává v té chvíli, kdy četnost typů (velikostí) jednotlivých obdélníkových částí není nijak omezen.

- **Profit částí**

1. **Weighted problem**

Weighted problem nastává tehdy, když díly na obdélníkové desce mají různou prioritu a obtížnost řezání. Nestejné hodnoty mají za následek to, že každý díl přináší pro společnost jiný zisk.

2. **Unweighted problem**

Unweighted problém nastává tehdy, kdy všechny díly na obdélníkové desce mají stejnou prioritu a obtížnost řezání. Každý díl přináší firmě stejný profit.

(Pro Výše uvedené problémy, neexistuje vhodný překlad do českého jazyka)

- **Počet řezů (nebo stupně) částí**

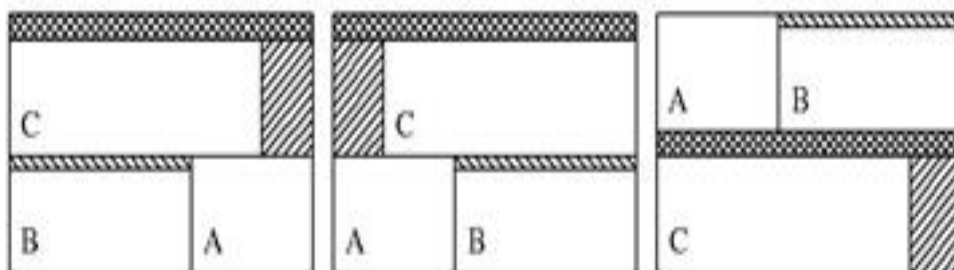
1. **Stages**

Problém je několika stupňový (překlad anglického slova stages), jestliže je omezen počtem řezů nebo stupňů. Čím je počet řezů, nebo stupňů, nižší, tím naopak efektivita výroby roste.

2. **Non-stages**

Problém lze prohlásit obecným, nebo nestupňovým (překlad anglického slova Non-Stages) pokud neexistují žádná omezení, týkající se počtu stupňů nebo řezů. Tento problém mívá za následek větší spotřebu materiálu (Correia a spol., 2012).

Pro lepší představu významu řezacího vzoru jsou na obrázku číslo 1 zobrazeny optimální řezací vzory.



Obrázek 1- Optimální návrhy řezacího vzoru (zdroj: Statistica Neerlandica, 2009, 474-489).

Tyto vzory jsou výsledkem řešení problému TCP. Z obrázku je zřejmé, že objekty jsou na formách naskládány tak, aby minimalizovaly odpad nepoužitého materiálu. Zatímco TCP se zabývá otázkou, jakým způsobem objekty na obdélníkovou desku naskládat, two-dimensional cutting stock problém se zabývá otázkou, jakým způsobem jednotlivé objekty od sebe separovat (Correia a spol., 2012).

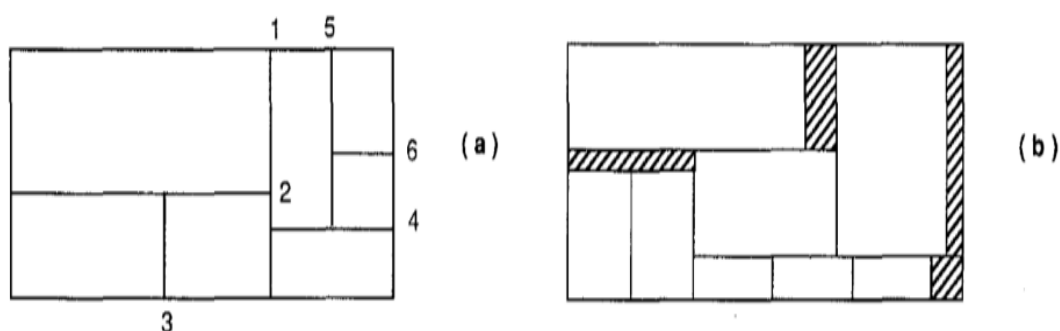
3.5.2 Two-dimensional cutting stock problem

TDC problém může být použit jako pomocný i při hledání řešení na two-dimensional cutting stock problém (dále dvourozměrný CSP) popisující proceduru řezání pro požadovaný počet malých pravoúhlých čtyřúhelníků z množiny velké desky s ohledem na minimální náklady výroby. Pro vyřešení dvourozměrného CSP musí být TDC několikrát řešen jako podproblém. Dvourozměrný CSP je tady sada řešení TDC problémů (Correia a spol., 2012).

O velké obdélníkové desce lze uvažovat jako o archu papíru. Cílem problematiky je maximalizovat množství nařezaných kusů, a to bez poškození jakéhokoliv z nich. Dalším předpokladem je minimalizovat množství odpadu (nepotištěného papíru), a tedy najít řešení na pro TDC problém.

Pravoúhlé předměty lze řezat ortogonálními nebo neortogonálními řezy. Při ortogonálním řezání se malé objekty umísťují tak, aby jejich hrany byly rovnoběžné s hranou velkého objektu, zatímco při neortogonálním řezáním je pozice objektů zanedbatelná. Problémy, ke kterým dochází při ortogonálním řezání, spočívají v tom, že řezací stroj dokáže řezat pouze spojité řezy z konce

jedné hrany do konce hrany protější. Tento druh řezu se nazývá gilotinový (Christofides a spol., 1995). Na obrázku číslo 2 lze porovnat dvě demonstrativní montáže forem. Forma na obrázku 2a znázorňuje formu, kterou lze rozřezat pomocí gilotinových řezů. Je zde také znázorněno pořadí a množství řezů potřebných k nařezání archu. Druhá forma obrázku číslo 2b zobrazuje formu, kterou není možno rozřezat pomocí gilotinových řezů bez zničení některého z objektů.



Obrázek 2 - (a) Forma pro gilotinový řez. (b) Forma, která nelze rozřezat pomocí gilotinového řezu (zdroj: Christofides, Hadjiconstantinou, 1995, 21-38).

3.6 Jazyk C# a související prostředí .NET

Souvislost mezi jazykem C# a prostředím .NET je významná. Objektově orientovaný programovací jazyk C# byl od počátku vyvíjen tak, aby spolupracoval především s touto technologií. Prostředí .NET bylo vyvinuto společností Microsoft. To má za následek to, že toto prostředí je určeno především pro vytváření téměř jakýchkoliv aplikací v systému Windows. Označení .NET má poukazovat na to, že společnost Microsoft se nyní cílí na distribuované aplikace, které pracují s výpočetním zpracováním rozděleným mezi klientem a serverem. (Model Klient – Server popisuje, jak server poskytuje prostředky a služby jednomu nebo více klientům. Serverem je myšlen například poštovní server. Klientem je myšlen například stolní počítač (TechTarget, 2017).)

Neznamená to však, že C# slouží pouze k vývoji síťových a internetových aplikací. Prostředí vývojářům poskytuje vyvíjet téměř jakýkoliv program, či komponentu určenou pro systém Windows. Vývojáři pomocí tohoto jazyka mohou vyvíjet

desktopové aplikace pro systém Windows, dynamické webové stránky, komponenty pro databázový přístup, nebo distribuované aplikace, inteligentní klientské aplikace podporující jak online, tak off-line režim, mobilní aplikace, webové služby XML a tak dále.

Vývojáři společnosti Microsoft byli před vznikem jazyka C# postaveni před velice těžké rozhodnutí. Byli si vědomi toho, že neustálé přidávání funkcí do dosavadních technologií má za následek postupné zvyšování komplikovanosti technologií. Tomu se snažili zabránit. Vizí bylo vytvořit takový jazyk spolu s prostředím a nástroji, který bude jednoduchý přesto pokročilý. Rozhodli se tedy vytvořit novou technologii .NET spolu s jazykem C#, který je navržen pro spolupráci s .NET technologií a současně si zachovává veškerý pokrok dosažený v posledních dvaceti letech ve vývojovém prostředí objektově orientovaného programování (Nagel a spol., 2008).

3.6.1 Porovnání C# s jinými jazyky

V této kapitole jsou uvedeny vybrané jazyky, které jsou srovnány v jazykem C#.

- **Visual Basic**

Visual Basic je jazykem, který není objektově orientovaný. Je tak dáno z toho důvodu, že jeho hlavní přednosti spočívají v jednoduchosti, a tedy skrývá před vývojáři podrobnosti o rozhraní Windows Api a infrastruktury komponent COM (Component Object Model Technologies – technologie, která umožňuje softwarovým komponentám v systému Windows mezi sebou komunikovat (Microsoft, 2017e).). Začátečnickům usnadňuje a napomáhá pochopit programátorské úkoly lépe, než je tomu ve složitých objektově orientovaných jazycích. Problém nastává při vývoji složitých aplikací, kdy je velice obtížné udržovat přehlednost napsaného kódu.

S příchodem prostředí .NET byl jazyk Visual Basic kompletně přepracován. Vznikl tak Visual Basic 2008. I když jsou některé základní prvky stejné jako v původním Visual Basic, princip je podstatně jiný.

- **C++**

Společnost Microsoft vyvinula svůj první kompilátor pro jazyk C++ ještě dříve, než byl oficiálně vydán standard mezinárodní organizací

pro normalizaci (ISO – International Organization for Standardization) jazyka C++. Tento krok měl za následek to, že jazyk C++ vychází z definice standardu ISO jazyka C++, nicméně nesplňuje ISO normu zcela. Jazyk C++ byl vyvíjen od počátku osmdesátých let. Dlouhá historie vývoje má negativní dopad na podporu dnešních moderních technologií jazykem C++. Také způsobuje v C++ setrvání některých zastaralých struktur, které byly využívány tehdejšími kompilátory. Jazyk je vyvíjen tak, aby byl určen především pro velké a náročné programy fungující v systému Windows. Pro splnění těchto zásad musí být pro jazyk C++ vyvíjeny nové knihovny spolu se specifickými klíčovými slovy. S každou aktualizací jazyk nabývá na náročnosti.

- **C#**

„Jednoduchý, moderní objektově orientovaný a typově bezpečný programovací jazyk, který je odvozen od jazyka C a C++.“ (Nagel a spol., 2008, 35) Tak zní oficiální definice jazyka C# společností Microsoft. Do výčtu jazyků, od kterých je C# odvozený, by mohla přibýt i Java. Lze tak konstatovat z toho důvodu, že tyto jazyky jsou tak podobné, že obsahují velké množství stejných klíčových slov. Jazyk C# přebírá od jazyků C++ a Java strukturu bloků, kde též používá složené závorky pro rozlišení bloků kódu. Příklad si také bere ze stylu ukončení příkazů středníky. Náročnost jazyka C# je podobná jako u jazyka Java. Oproti jazyku C++ je C# pro pochopení výrazně snazší. Snaží se nabídnout snadné použití jako je to u jazyka Visual Basic a zároveň poskytovat moderní vývojové prostředí spolu s možností využití vysokého výkonu a nízkoúrovňový přístup k paměti (přístup poskytující malou nebo žádnou abstrakci od programových konceptů. Je velmi blízko ke skutečnému zápisu strojových instrukcí).

3.6.2 Omezení jazyka C#

Nedostatky jazyka C# jsou znatelné při vývoji aplikací, která vyžadují extrémně zatěžovaný nebo časově náročný kód. Příklad může být například program, který bude využívat stovky či tisíce cyklů procesoru. Když nastane situace, že takovýto program bude potřebovat při běhu programu uvolnit během milisekundy ty prostředky, které již nejsou využívány, C# bude mít se zpracováním velké problémy, které mohou vést ke kolapsu. Pro tento druh aplikací je bezpochyby určen jazyk C++. Jazyk C# nadále postrádá funkce s modifikátorem inline (zjednodušeně jsou to funkce, které jsou rychlejší, než funkce klasické) a destruktory (funkce pro eliminaci objektů z důvodu uvolnění paměti, u kterých je zajištěno spouštění v definovaných bodech kódu) Tyto funkce však využívá jen hrstka aplikací, které potřebují k vykonání příkazů velký výkon.

Další část práce aplikuje zmíněné metody, principy a nástroje do praxe.

4 Návrh a tvorba aplikace

Pro návrh a tvorbu aplikace bylo nutné porozumět firmě OV-Media s. r. o. a proniknout do jejích procesů. Dále bylo nutné vytvořit takový návrh aplikace, který koresponduje se specifickými požadavky týkající se funkčnosti aplikace, které společnost na začátku zadala. Nahlédnutí do infrastruktury společnosti díky komunikaci s firmou přispělo ke vzniku návrhu aplikace a její tvorbě dle specifických požadavků. Komunikace probíhala především s programátorem firmy Alexandrem Pař. V prvotní fázi návrh s ním byla prodiskutována problematika a zadání požadavků aplikace, která má napomoci zefektivnění pracovního procesu společnosti. Alexander v průběhu vývoje velice napomohl především s modernizací zdrojových kódů. Zjevně byl redukován počet řádků kódu tak, aby funkcionality zůstala stejná. Docíleno tím bylo především díky LINQ (Language-Integrated Query – nástroje pro dotazování) dotazů, které kód zjednodušily a zjednodušily. S vedením firmy probíhala nejprve komunikace především z důvodu vysvětlení pracovních a výrobních procesů. Při vývoji aplikace firma poskytovala informace o interních procesech, potřebných ke správnému zpracování zakázky. Na konci projektu byla vedení představena výsledná aplikace a byla diskutována zpětná vazba ohledně využitelnosti výstupu práce. Aplikace pro výpočet minimálního počtu řezů včetně její dokumentace a zdrojových kódů je k dispozici na přiloženém kompaktním disku bakalářské práce.

4.1 Společnost OV-Media s. r. o.

Společnost OV-Media s. r. o. sídlí v jihočeském městě Vimperk. Firma byla založena roku 2005. Nynější počet zaměstnanců se pohybuje okolo 130 členů. Firma se zabývá tiskem a následným zpracováním vytištěného díla pro zábavní průmysl. Disponuje dvěma zařízeními využívající Computer to Plate (Do češtiny přeloženo jako: Z počítače na plát) technologie, které slouží ke zhotovení moderních tiskařských forem, šesti vícebarvovými stroji a několika stroji pro finální zpracování díla.

4.1.2.1 Příprava tisku

Příprava tisku je klíčovým procesem. Právě tato chvíle je důležitá pro nastolení správné strategie výroby, vedoucí k úspěšnému splnění přijaté zakázky. Níže je popsáno vše, co se v této fázi musí zajistit a jaké pracovní procesy v ní probíhají.

1. Objednávka

Prvním krokem je přijetí objednávky a smlouení jejich podmínek se zákazníkem. Mezi ty patří množství produktu, datum dodání a cena. Tímto krokem je objednávka zařazena do výrobního plánu společnosti.

2. Zpracování dat

Data s požadavky, které zákazník zašle tiskárně, se musí dále zpracovat. Zajistí se grafické zpracování přijaté zakázky a určení správných vlastností papíru, jako je velikost papíru a jeho typ. V závěru této fáze je zakázce přiřazeno identifikační číslo spolu s údaji o ní.

3. Plánování výroby

Dílo je zde strategicky umístěno na tiskařskou formu tak, aby jeho výroba byla co nejefektivnější. Úspěšným provedením tohoto kroku přichází na řadu kalkulace a odhady času potřebného pro jednotlivé úkony výrobního procesu. Tyto kalkulace vycházejí z dat, která byla získána dřívějším monitoringem a zanalyzováním připravené formy. Jestliže je forma vyhovující, je zařazena do tiskařského plánu, jinak je forma poupravena a proces analýzy proběhne opětovně.

4.1.2.2 Tisk

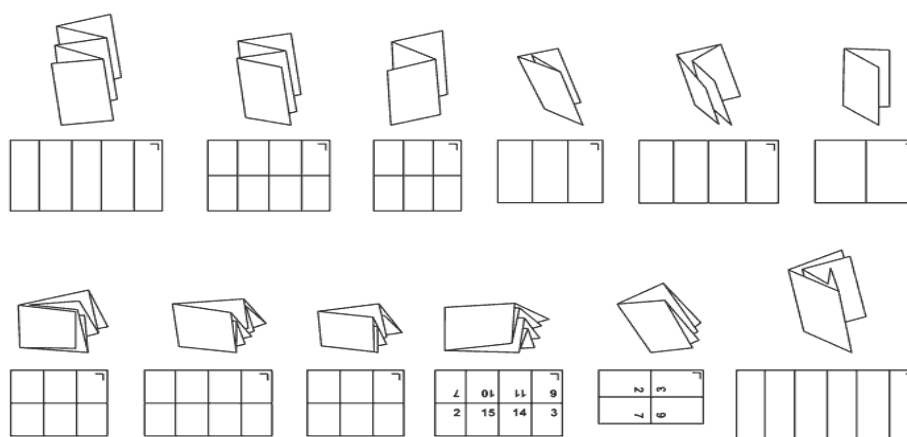
Tiskárna je založena na kontinuálním výrobním procesu. Tiskárna tiskne až na šesti více barvových strojích na velké archy papíru mnoho menších zakázek. Jakmile je vytištěn požadovaný počet archů, přechází se ke zpracování papíru, probíhajícím v dalším výrobním procesu. Seskupování jednotlivých zakázek na arch papíru velkých rozměrů přispívá k ušetření nemalého množství materiálu, času a financí. Způsob seskupování zakázek je popsán níže v podkapitole tvorba vstupních dat.

4.1.2.3 Knihárenské procesy

Po úspěšném dokončení tisku následují knihárenské procesy. Jde především o modifikaci papíru. V tomto procesu je zahrnuto:

1. Falcování papíru

Falcování papíru je jedna z knihařských technik, kdy je papír překládán, nebo skládán tak, aby dosáhl požadovaného formátu (například A5 nebo A6). Falcování se provádí na automatickém stroji, falcovače, do kterého je papír vkládán. Na obrázku 4 je vidět původní papír před vstupem do falcovačky a také finální produkt.



Obrázek 4 - Varianty skládání papíru (zdroj: Dobel tiskárna, 2016).

2. Vázání papíru

Vázání papíru je knihařská technika, která sváže několik papírů dohromady. Výsledkem jsou například brožurky.

3. Perforace papíru

Při tomto procesu je papír proděrován a následně lze odtrhnout jeho část. Perforaci je možné vidět například u vstupenek. Perforované části bývají u vstupenek odtrhávány.

4. Řezání papíru

Výsledkem rozřezáním archu papíru jsou menší části, se kterými se následně pracuje, nebo jsou již jako finální produkt expedovány. Řezací stroje, které používá společnost OV-Media s. r. o. , fungují na gilotinovém způsobu řezání, kdy řezací stroj dokáže řezat pouze spojitě řezy z konce

jedné hrany archu papíru do konce hrany protější. Porozumění tohoto způsobu řezání papíru vedlo k úspěšnému návrhu aplikace.

4.1.2.4 Expedice

Kombinací výše uvedených procesů vzniká konečný produkt. Jestliže zakázka úspěšně projde kontrolou a splňuje všechny zákaznickovy požadavky, je zabalena, zvážena a poslána na smlouvané místo.

4.2 Specifikace aplikace

Díky analýze společnosti OV-Media s. r. o. a jejích výrobních procesů byl určen účel aplikace spolu s místem jejího nasazení. Řízení procesu inovace zde bylo využito pro nalezení nového plánovacího procesu.

Návrh aplikace se potýkal s řadou situací, na které bylo zapotřebí najít řešení. Specifikace těchto problémů vedla k nalezení potřebných řešení. Jejich kompletace vyústila v tvorbu celkového návrhu.

4.2.1 Účel aplikace

Aplikace je schopna výpočtu minimálního možného počtu řezů, který musí řezač provést k rozřezání archu papíru na určité části. Výpočetní proces se skládá ze dvou částí. V první části aplikace vypočte, zda lze formu rozřezat tak, že ani jeden objekt na archu papíru nebude žádným řezem poškozen. Jestliže je tato podmínka splněna, aplikace přejde k druhému výpočtu, a to zjistit minimální počet řezů, který je nutný k separaci objektů na archu papíru.

4.2.2 Nasazení aplikace

Aplikace je nasazena v sektoru plánování a monitoringu společnosti. Sektor pro plánování využívá aplikaci k efektivnímu návrhu formy. V sektoru monitoringu je použita především pro možnost počítat a odhadovat časovou náročnost zakázek.

4.2.3 Analýza vstupních dat

Návrh pro tvorbu aplikace vyžadoval analýzu vstupních hodnot potřebných k výpočtu řešení. Tyto hodnoty jsou výstupem ze sektoru plánování výroby společnosti OV-Media s. r. o. Pracovníci v tomto sektoru u každé zakázky navrhují

vhodné rozložení jednotlivých stran na tiskařskou formu. Selekcce zakázek na jedné formě je určena pracovníkem, který se sám rozhodne podle svých zkušeností a firemních postupů. Klíčovým faktorem, který ovlivňuje jeho rozhodovací proces, je především velikost zakázek. Jednotlivé strany či skupiny stran reprezentují rozdílné zakázky, které však mohou být umístěny na formu pospolu. Identifikaci zakázek zajišťují identifikační čísla. Každá zakázka má své unikátní identifikační číslo, které se pojí k výrobnímu číslu.

Níže je popsán konkrétní příklad:

Na formě se mohou vyskytnout dvě vizuálně podobné zakázky. Pracovník dostane ke každé zakázce papír s informacemi o zakázce. Ty zahrnují kromě obecných údajů také unikátní identifikační číslo zakázky pojící se s výrobním číslem. Identifikační číslo se od výrobního čísla liší. Často nastává taková situace, kdy je identifikační číslo obsahem grafického návrhu zakázky. Takto je zajištěna správná identifikace jednotlivých zakázek

4.2.3.1 Typy zakázek

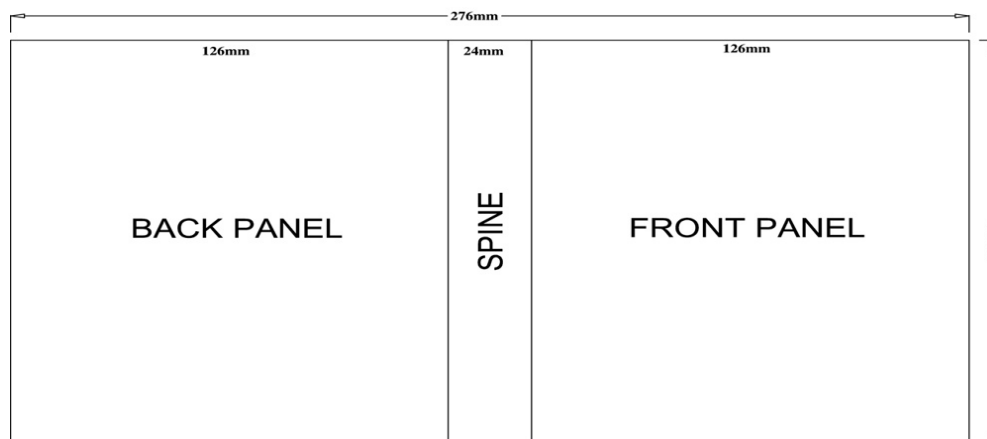
Zakázky, které jsou rozmisťovány na tiskařskou formu lze členit podle typu, nebo formátu. Jednotlivé typy a formáty jsou blíže popsány níže. Pro anglické termíny uvedeny níže neexistuje v českém jazyce výstižné synonymum a i v českém prostředí se využívají v anglickém originále. Proto nejsou přeloženy.

1. Typ zakázek

- a) Inlay Side – obal krabiček počítačových her
- b) Booklet – malé knížky s informacemi a návodem ke hře; vkládán do krabičky s počítačovou hrou
- c) Insert – jeden list papíru zpravidla obsahující bonusový obsah hry; vkládán do krabičky s hrou
- d) 4-Side – kombinace typu booklet s insert; jeden list papíru přeložený na půl, čímž vznikají 4 stany; vkládán do krabičky spolu s dalším obsahem
- e) Poster – plakát

2. Formát zakázek

- a) CD/DVD – přední a zadní strana obalu o rozměrech 150x118mm
- b) Blu-ray – na rozdíl od CD a DVD obalů jsou Blu-ray a další níže uvedené obaly tištěné jako jeden kus o rozměrech 276x148mm, jako je vidět na obrázku číslo 5



Obrázek 5 - Obal na blu-ray disk (zdroj: Upraveno, dostupné z: http://www.hirescovers.net/forum/uploads/monthly_09_2011/post-15084-1315619788.jpg).

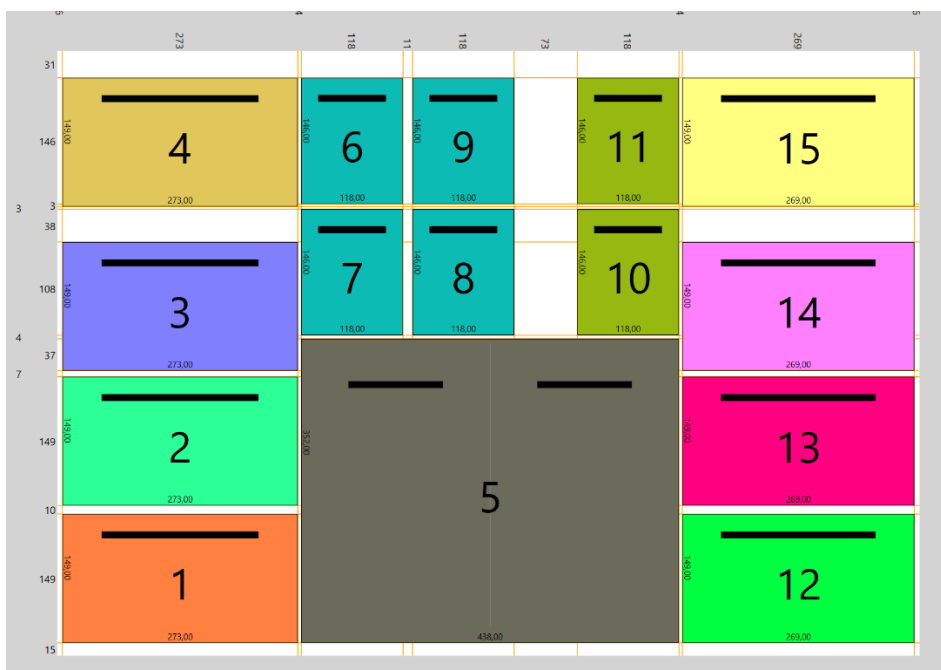
- c) PlayStation format – rozměry se liší podle typu konzole; krabičky pro PlayStation 3 mají jiný rozměr než krabičky pro PlayStation 4 a tak dále
- d) Nintendo format – rozměry se liší podle typu konzole
- e) X-Box format – rozměry se liší podle typu konzol

4.2.3.2 Reprezentace dat

Data jsou reprezentována dvěma způsoby. Prvním je vizuální reprezentace, kterou využívají pracovníci společnosti pro práci s návrhem formy v aplikaci. Druhá, datová reprezentace, je využívána aplikací, která tímto způsobem uchovává informace o zakázkách.

1. Vizuální reprezentace

Na obrázku číslo 6 je znázorněna výstupní forma procesu návrhu. Takto vypadá forma v uživatelském rozhraní programu. V reálné situaci představuje jeden velký list papíru, na který je umístěno několik menších částí rozdílných formátů například A4, A3 nebo A5. Bílá barva znázorňuje plochu, do které lze vkládat objekty. Objekty zde reprezentují strany označené čísly 1 až 15. Barvy stran označují, které objekty patří do stejné zakázky zadané zákazníkem. Na formě jsou též znázorněny velikosti jednotlivých objektů spolu se vzdálenostmi mezi objekty.



Obrázek 6 - Výstupní forma procesu návrhu (zdroj: Vlastní zpracování).

2. Datová reprezentace

Rozložení objektů na formě je reprezentováno pomocí .XML souboru, který je vytvořen při dokončení návrhu. XML soubor obsahuje informace, jako je jméno objektu, identifikační číslo objektu, velikost objektu, pozice

objektu, rotace objektu, počet objektů, sdružení objektů do skupin a tak dále. Pro správný návrh aplikace je důležité zaměřit se především na hodnoty tagů (Tag do češtiny přeloženo jako štítek, reprezentuje element v jazyce xml) reprezentujících velikost, pozici a rotaci.

Tagy XML jsou psané v německém jazyce. To je dáno německým původem společnosti. Programátor společnosti je též německého původu, a tedy napsal zdrojový kód v německém jazyce.

4.2.3.3 Tvorba vstupních dat

Návrh a generování dat je prováděno v grafickém rozhraní programu, který využívá společnost OV-Media s. r. o. Pracovníci tato data tvoří dvěma způsoby. Prvním způsobem je použití předdefinovaných šablon, do kterých tažením ze zásobníku zakázek vloží ty, které odpovídají parametrům šablony. Nad zásobníkem zakázek probíhají filtry, definované kritériem specifickým pro vytvoření návrhu určitého produktu. Druhý způsob nevyužívá šablony. Obsluha aplikace umístí požadované strany na formu tažením ze zásobníku zakázek. Na formu je umísťuje podle svého rozvážení. Spoléháno je na zkušenosti pracovníka. Je tomu tak z důvodu velké složitosti potenciálního programu, který by tuto problematiku dokázal řešit. Program obsluze pomáhá strany zarovnávat. Také zde probíhá proces lícování, kdy jsou hledány strany, které k sobě mají vztah z hlediska rozměru. Tyto strany se sdružují do skupin, které pracovník takticky rozmísťuje do formy. Po dokončení procesu, jsou všechna data uložena do souboru XML. Pracovník má možnost po uložení, XML dokument opětovně nahrát do systému a vykonat na něm potřebné úpravy.

4.2.4 Analýza problematiky návrhu aplikace

Pro návrh aplikace bylo nutné vyřešit dva problémy. První problematika se zabývá otázkou, jestli lze formu rozřezat. Druhá problematika se týká zjištění minimálního počtu možných řezů, které jsou potřeba k separaci jednotlivých stran tak, aby žádná z nich nebyla poškozena.

4.2.4.1 Definice objektu

Velmi nápomocná byla při návrhu znalost vlastností objektů (stran) rozmístěných do formy. Všechny objekty jsou rovnoběžníky, které mají všechny vnitřní úhly pravé. Rotace objektů je možná pouze o 90° kterýmkoliv směrem. Není tedy možné, aby bylo s objekty rotováno jinak. Kdyby docházelo k jiným rotacím, než je uvedeno výše, nebylo by možné formu rozřezat pomocí gilotinových řezů, bez poškození objektů.

Každý objekt může mít jinou velikost a pozici ve formě. Žádný objekt nesmí být umístěn mimo formu.

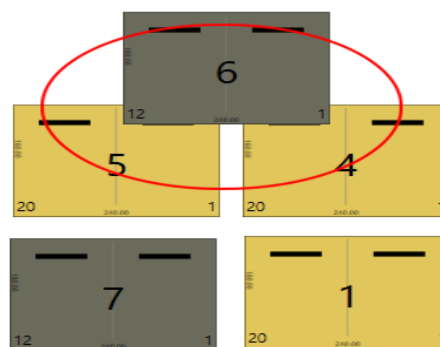
Objekty stejné velikosti, pozice na formě a jiného identifikačního čísla na sebe nemohou být skládány a poté společně řezány. Docházelo by k pomíchání jednotlivých zakázek a následné reklamaci. Je třeba uvažovat tak, že společnost nepracuje pouze s jedním velkým listem papíru, ale naopak s tisíci archů. Skládání stejných objektů na sebe, jejich řezání a následné třídění by bylo extrémně časově náročné a vznikalo by velké riziko způsobení chyby. Byť jeden odlišný kus (objekt) chybně vložený do nesprávné zakázky, obvykle znamená reklamaci celé zakázky obsahující například pět set tisíc kusů. Tyto chyby by mohly vést k nezanedbatelným finančním ztrátám a ke krachu celé společnosti.

4.2.4.2 Možnosti rozřezání formy

Formu lze rozřezat jen tehdy, když jsou objekty na formu rozmístěny podle několika pravidel. Porušením pravidel dojde ke špatnému návrhu formy a znemožnění bezpečné separace jednotlivých objektů. Níže jsou uvedena pravidla, která je nutné dodržovat.

1. Objekty se nesmí překrývat

Formu nelze rozřezat, jestliže se jeden nebo více objektů překrývají. Tato situace je znázorněna na obrázku číslo 7, kdy objekt číslo 6 překrývá objekty číslo 5 a 4. Stejně tak formu nelze rozřezat při rozložení objektů, kdy jedna hrana objektu leží přesně na hraně jiného objektu, a to i přes to, že relativně nedochází k poškození žádného objektu.

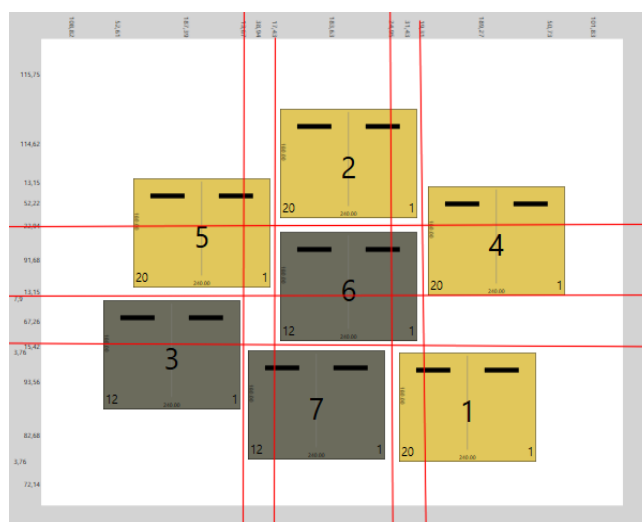


Obrázek 7 - Stav kdy se objekty na formě se překrývají (zdroj: Vlastní zpracování).

Objekty mezi sebou musí mít rozstup alespoň 0,1 mm. Děje se tak z důvodu tisku, kdy se hrany jednotlivých objektů při samotném tisku slévají. Vzniká tak zmetek, který musí být z výroby odstraněn. Ve výše uvedených případech musí být návrh formy přepracován.

2. Mezi objekty musí existovat vždy alespoň jeden možný proveditelný řez

Kvůli gilotinovému způsobu řezání, kdy řezací stroj dokáže řezat pouze spojitě řezy z konce jedné hrany formy do konce hrany protější, musí být brán zřetel na to, aby nebyl poškozen žádný objekt na formě. Jestliže nastane situace jako na obrázku



Obrázek 8 - Stav, kdy neexistuje žádný možný řez k separaci všech objektů (zdroj: Vlastní zpracování).

číslo 8, formu nelze rozřezat. V této situaci nelze provést ani jeden řez, který by nepoškodil žádný objekt. Když z příkladu na obrázku číslo 3 bude vyjmuta stránka číslo 5, formu lze rozřezat. Je to z toho důvodu, že lze provést řez mezi objekty 3 a 7. Tímto procesem vzniknou dvě již oddělené skupiny. V každé další skupině, vytvořené separací, musí neustále platit pravidlo alespoň jednoho možného řezu bez poškození jiného objektu. Separace skupin do podskupin probíhá až do té doby, dokud každá skupina neobsahuje pouze jeden objekt. Poté je možné formu rozřezat.

4.2.4.3 Minimální počet řezů

V další fázi je nutné identifikovat minimální počet řezů. Prvním předpokladem pro tento výpočet je možnost rozřezání formy. Jestliže je to možné, je prováděn výpočet pro nejmenší možný počet řezů. Jako u zkoumání možnosti rozřezání, i zde je několik pravidel, která vedou k optimálnímu výsledku. Na rozdíl od první problematiky, kde je předmětem zkoumání jen možnost rozložení objektů na formě, je v tomto případě brána v úvahu i manipulace a přesouvání separovaných skupin.

Skupiny objektů

Primárním stavem je skupina objektů, jejíž množinou jsou všechny objekty na formě. Postupně se tato skupina dělí vždy do dvou dalších skupin. Skupiny se rozmělní až do té fáze, kdy tyto skupiny můžeme přesouvat a vzájemně sdružovat tak, že k sobě mají vzájemný vztah. Vztah zde vyjadřuje symetričnost pozice na ose X nebo Y. Tyto skupiny jsou následně podrobeny dalšímu zkoumání.

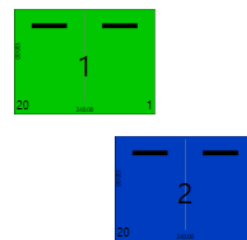
Pravidla pro výpočet řezů

Níže jsou uvedeny pravidla, která je nutno dodržovat pro úspěšné zjištění minimálního počtu řezů.

1. Ořezání objektu

Každý objekt musí být z každé strany jednou ořezán. Bude-li tedy na formě pouze jeden objekt, minimální možný počet řezů se rovná hodnotě čtyř. V dalším případě, kdy budou na formě rozmístěny dva objekty,

jako na obrázku číslo 9, je minimální počet řezů osm. Tyto objekty k sobě nesmí mít vzájemně žádný vztah co se pozice týče. Pokud ve vzájemném vztahu objekty nebudou, což znamená, že jejich pozice ani na jedné z os X nebo Y není stejná, pak se minimální počet řezů bude rovnat počtu objektů na formě vynásobený čtyřmi.



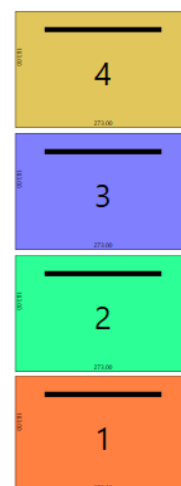
Obrázek 9 - Dva objekty bez vzájemného vztahu pozice (zdroj: Vlastní zpracování).

2. Vnitroskupinový vztah objektů

Chybnou úvahou, vedoucí k nesprávnému výpočtu je vynásobení počtu všech objektů čtyřmi, bez zkoumání vzájemných vztahů, ihned poté, co bylo zjištěno, zda lze formu rozřezat. Zaměřením se na vztahy objektů v rámci jedné skupiny lze ušetřit spoustu řezů, které by pracovník musel provést. Obrázek číslo 5 znázorňuje skupinu objektů, majících k sobě vzájemný vztah, který je definován vodorovnou pozicí, tedy pozicí na ose X. Všechny objekty mají tuto hodnotu stejnou.

Pro příklad: Jestliže pozice levého spodního rohu oranžového objektu číslo 1 na ose X, je rovna hodnotě p_x , mají i ostatní objekty pozici levého spodního rohu na ose X rovnou hodnotě p_x . Tato skutečnost se dá tvrdit i o pozici pravého spodního rohu.

Pro objekty rozložené na obrázku číslo 10 je minimální počet řezů v rámci jedné skupiny rovný číslu deset. Děje se tak



z důvodu ořezání více stran jedním řezem. V tomto případě jsou dvěma řezy ořezány všechny levé a pravé strany všech objektů. Je tedy ušetřeno šest řezů, oproti stavu, kdy by se každý objekt ořezával jednotlivě. Když na obrázku číslo 5 bude oranžový objekt číslo 1 posunut jakýmkoliv způsobem po ose X, počet řezů se zvýší o dva a skupina bude muset být rozdělena do dalších dvou skupin. První skupina bude obsahovat množinu objektů číslo 4, 3 a 2 a druhá skupina bude obsahovat pouze objekt číslo 1. Tímto způsobem by bylo zamezeno poškození objektu číslo 1 při řezání.

Obrázek 10 - Skupina objektů ve vzájemném vztahu (zdroj: Vlastní zpracování).

3. Meziskupinový vztah objektů

Vztah nemají objekty pouze v rámci jedné skupiny, ale je také možné zkoumat, jaký vztah mezi sebou mají objekty jedné skupiny k objektům skupiny druhé. Skupiny můžeme sdružovat do potencionálních skupin, které vytvoříme jejich přesouváním. Nastane-li na obrázku číslo 5 situace, že zelený objekt číslo 2 bude posunut jakýmkoliv směrem po ose X, skupina bude muset být, pro správné rozřezání, rozdělena do třech skupin. První skupina bude obsahovat množinu objektů čísla 4 a 3. Druhá skupina bude obsahovat pouze objekt číslo 2 a třetí skupina pouze objekt číslo 1. Když se se skupinami bude zacházet pouze tak, že v nich budou zkoumány jen vnitroskupinové vztahy objektů, povede to k chybnému výpočtu. Je tak dáno, protože objekty první skupiny mají s objektem třetí skupiny vztah týkající se pozice na ose X.

Pracovník u řezacího stroje v tomto případě musí postupovat následovně. Z první skupiny na obrázku číslo 5 vyjme zelený objekt číslo 2. Pro tento krok a současně pro ořezání celého zeleného objektu číslo 2 je potřeba provést čtyři řezy. Úplným ořezáním objektu dochází k jeho vyjmutí a dále se s ním už nepočítá. Tímto zaniká druhá skupina, protože žádná skupina nemůže být prázdná. Poté řezač seskupí první a třetí skupinu. Z důvodu stejné pozice levého a pravého spodního rohu na ose X všech objektů obou skupin, může pracovník provést dva řezy. Ty mají za následek ořezání jak levých, tak pravých stran všech objektů ve všech zbývajících skupinách. Finální počet řezů pro ořezání všech objektů je v tomto případě rovný dvanácti řezům.

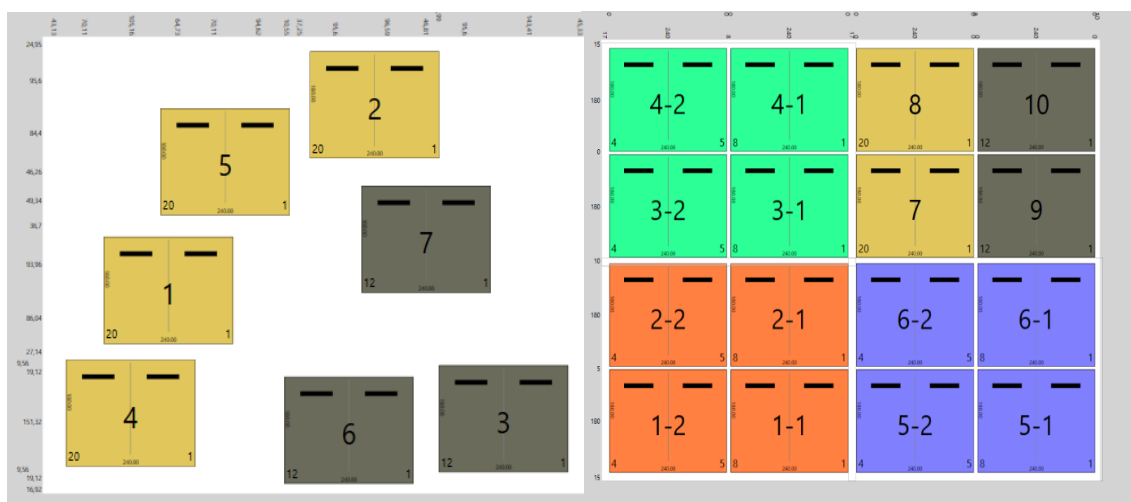
Špatný návrh formy versus Ideální návrh formy

Z výše uvedených pravidel vyplývá, že je opravdu důležité rozmístění objektů na formě. Pracovníci při návrhu a tvorbě formy musí s těmito pravidly počítat. Nevyhovující návrh formy, který je poslán do výroby, může mít za následek velké ztráty, které by byly zapříčiněné časovou náročností zpracování zakázek.

Na obrázku číslo 6 jsou dvě formy. Levá forma se vyznačuje chybně rozmístěnými objekty na rozdíl od pravé formy, kde jsou všechny objekty ideálně rozvržené. I

přes to, že na levé formě je pouze sedm objektů oproti pravé formě, kde je objektů šestnáct, pro rozřezání levé formy je potřeba o dvanáct řezů více. Zapříčiňuje to nulový výskyt jak meziskupinových, tak vnitroskupinových vztahů. Levá forma se tedy ve společnosti nemůže vyskytnout. Pracovník v oddělení přípravy tisku musí zamezit návrhu takovéto formy. Z toho důvodu navrhovaná aplikace není určena pro výpočet minimálního počtu řezů pro tento druh nahodilých forem. Aplikace tyto formy řeší jen okrajově. Předpokladem pro výpočet je, že objekty na formě k sobě mají aspoň nějaký vztah pozic. To znamená, že zaměstnanec objekty zarovná. Řešení pro levou formu by počítala aplikace, která hledá ideální rozmístění objektů do formy. V rámci zadání firmy byl účel aplikace vyspecifikován a byly jasně dány i výchozí podmínky, které jsou zaručeny díky zaměstnancům v přípravné fázi formy.

Objekty pravé formy na obrázku číslo 11 jsou na rozdíl od objektů formy levé rozmístěny ideálně. Tím, že se pracovník rozhodl na formu naskládat objekty stejné velikosti a všechny objekty zarovnal, usnadnil svému kolegovi u řezacího stroje nemálo času. Je potřeba pouhých šestnáct řezů k ořezání každého jednoho objektu. Počet řezů je tak nízký, díky tomu, že bylo využito jak vnitroskupinových, tak meziskupinových vztahů.



Obrázek 11 - Levá forma se špatně rozmístěnými objekty. Pravá forma s ideálně rozmístěnými objekty (zdroj: Vlastní zpracování).

Příklad řezání pravé formy:

V první fázi rezač provede všechny svislé řezy. Tím pádem všechny objekty formy mají ořezané levé a pravé hrany objektu. Byly zde využity vnitroskupinové vztahy,

díky kterým řezač mohl jedním řezem ořezat až čtyři objekty současně. Pro ořezání všech levých a pravých hran objektů na formě je zapotřebí osmi řezů. V druhé fázi řezač provede všechny vodorovné řezy. Meziskupinové vztahy jsou zde zajištěny pozicí a také velikostí objektů. Skupiny jsou reprezentovány obdélníkovými kusy papíru, které vznikly rozřezáním archu papíru svislými řezy. Jedním řezem lze ořezat až čtyři objekty jak na spodní straně, tak na horní straně objektu díky stejné velikosti. Pro ořezání všech spodních a horních hran objektů na formě je zapotřebí též osm řezů. Toto rozložení je nejefektivnějším možným k ořezání všech objektů.

4.2.5 Analýza C# projektu

Společnost OV – Media s. r. o. poskytla pro návrh aplikace projekt, do kterého měly být zaneseny funkcionality nově navrhované aplikace. Jedním z požadavků bylo, aby byl při tvorbě aplikace využit jazyk C#. Tento požadavek byl zadán z toho důvodu, že interní program, který společnost využívá, je psaný v jazyce C#. Aplikace pro výpočet nejmenšího možného počtu řezů, musel být kompatibilní s dosavadním programem a vývojovým prostředím společnosti. Bylo tomu tak z důvodu zajištění snadného importu aplikace do celkového systému. Zvolení jiného jazyka, například jazyka Java, by mohlo vést k problémům při propojení nové aplikace s již užívaným systémem.

Dalším benefitem, který přináší zvolení jazyka C#, jsou vlastní C# knihovny společnosti, které byly využity při kompletaci aplikace. Tyto knihovny zajišťují v nové aplikaci především práci s XML soubory. Bez jejich využití, by musela být celá funkcionality napsána znovu.

Poslední velkou výhodou použití jazyka C# je fakt, že programátor, se kterým probíhaly konzultace při tvorbě aplikace, jazyk velmi dobře ovládá. Klade důraz na udržení trendu ve světě jazyka C#, a proto autorovi napomohl ke zlepšení aplikace. Díky využití nových přístupů se podařilo po konzultaci zobecnit kód a ušetřit jeho počet řádků, které jsou potřeba k napsání funkcionality v programu, což vedlo k lepší čitelnosti kódu.

4.2.5.1 Projekt aplikace

Aplikace byla vytvořena v integrovaném vývojovém prostředí Microsoft Visual Studio s použitím jazyka C#. Na samém začátku vývoje společnost poskytla solution (Struktura pro organizaci projektů ve Visual Studio) obsahující jeden projekt (Prostor, který seskupuje všechny zdrojové soubory, datové soubory, obrázky a vše co je zapotřebí k sestavení spustitelné aplikace). Využívaným obsahem projektu byla třída MainWindow, která ztvárňuje grafické rozhraní aplikace. Grafické rozhraní je napsané v jazyce XML. Nicméně „backendová“ část (část softwaru sloužící ke zpracování dat) je stále v jazyce C#.

Třída MainWindow v počáteční fázi obsahovala plátno reprezentující arch papíru. Dále dvě tlačítka. Tlačítko pro načtení XML souboru a tlačítko pro zobrazení pozic objektů. Tlačítko pro načtení XML souboru načte XML soubor, přeloží ho a vloží do plátna objekty tak, jak je definováno XML souboru.

Projekt dále obsahoval složku TestData, ve které byly uloženy XML soubory. V nich byla definice objektů vkládaných do plátna. XML soubory sloužily k testovacím účelům aplikace.

V poslední řadě byla velmi nápomocná knihovna fromenlib.dll, která zajišťovala funkcionality plátna a pomáhala se zpracováním XML souboru. Do této knihovny však společnost neudělila přístup.

4.3 Návrh aplikace

Samotný návrh aplikace byl nejnáročnější částí celého projektu. Při jeho vymýšlení několikrát došlo k nefunkčnímu návrhu, který musel být přepracován. To bylo zčásti způsobeno metodou brainstormingu, která byla z počátku návrhu využita. Díky ní byl návrh vymyšlen, ale problém byl v tom, že se již nezaobíral tím, kde se nachází jeho slabiny. Chybou by však bylo tvrdit, že nefunkční návrh nevedl k ničemu jinému, než k jeho přepracování. K nalezení správného návrhu napomohly i návrhy chybné, jejichž analýzou byla zjištěna slabá místa, které vedla k efektivnějšímu pochopení a uchopení problému. Při analýze chybných návrhů byly využity prvky SWOT analýzy.

4.3.1 Kontrola XML dokumentu

Prvním krokem bylo zajištění správného obsahu XML dokumentu. V XML dokumentu musí být definované objekty. Jestli tomu tak není, program zahlásí uživateli, že vložený XML dokument není validní a nepustí ho k dalším operacím. Tento krok byl implementován zejména kvůli tomu, aby uživatel nemohl provést výpočty před tím, než programu pošle správná data. Předešlo se tak ke zkolabování programu ještě před spuštěním výpočtů.

4.3.2 Zpracování dat

Další problematikou následující po kontrole XML dokumentu bylo to, jakým způsobem je nutno uchopit a zpracovat data získaná z validního XML dokumentu. XML dokument potřebný ke zpracování obsahuje mnoho hodnot. Ne se všemi tagy bylo nutno pracovat. Níže je výčet problémů, pro který byl vymyšlen funkční návrh.

- Návrhem byla vyřazena možnost jakýmkoliv způsobem zasahovat do XML dokumentu. Rozhodnuto bylo především z důvodu bezproblémového nasazení aplikace do celkového systému. Bylo ovšem nutné vymyslet, jakým způsobem se bude zaznamenávat stav, který objekt nabývá při potencionálním řezu. Žádný tag v XML dokumentu neposkytuje možnost tyto hodnoty zaznamenávat.
- Dalším problémem, který se pojil s nemožností upravit XML dokument, byly hodnoty tagů, pozice a velikosti. Tyto hodnoty jsou ukládány pomocí datového typu float (Klíčové slovo float označuje jednoduchý typ, který ukládá 32bitové hodnoty s plovoucí desetinnou čárkou (Microsoft, 2017a.) s přesností třináct desetinných míst. Práce s daty s takovýmto rozsahem je velmi pracná při porovnávání hodnot objektů. Čísla by musela být při chodu programu neustále zaokrouhlována. Pro správný chod aplikace stačí přesnost dvou desetinných míst. To je dáno limity řezacího stroje.
- Poslední problematika, které byla věnována pozornost, byla absence hodnot určující pozici. XML dokument uchovává jen pozici levého spodního rohu. Tím pádem by bylo nezbytné neustále dopočítávat zbylé pozice objektu sčítáním hodnot pozice s hodnotami velikostí objektu.

Problémy vyřešilo navržení třídy Piece.cs, která reprezentuje nový objekt, a nese informace o něm. Název piece je z anglického slova kus.

Níže je uveden výčet výhod, který přináší návrh třídy Pieces.cs .

- Třída uchovává jen potřebné hodnoty. Tím pádem není nutné nepřetržitě uchovávat všechny hodnoty, které původní XML dokument obsahoval. Tento krok vedl k ušetření paměti počítače.
- Do nové třídy byly přidány hodnoty datového typu bool. (Klíčové slovo je alias pro System.Boolean. Používá se k ukládání booleovské hodnoty proměnné deklarovat true a false (Microsoft, 2017b).) Hodnoty tohoto typu jsou ve třídě Piece.cs celkem čtyři a každá reprezentuje jednu stranu objektu. Uchovávají informace o tom, jestli daná strana objektu byla řezána.
- Hodnoty pozice a velikosti byly převedeny na datový typ integer (Klíčové slovo označuje integrálního typu, který ukládá 32Bitové celé číslo se znaménkem (Microsoft, 2017c).) Před převedením byly hodnoty datového typu float zaokrouhleny, vynásobeny číslem reprezentující požadovanou přesnost a ihned poté přetypované na datový typ integer.

Konkrétní příklad:

Původní hodnota pozice levého spodního rohu objektu pro osu X datového typu float je 23.3794564888437. Toto číslo je nutné vynásobit tak, aby byla dodržena přesnost, s kterou řezací stroj dokáže provádět řez. Přesnost řezacího stroje jsou setiny milimetru, číslo reprezentující takovouto přesnost se rovná číslu 100. Původní hodnota je tedy nejdříve zaokrouhlena na dvě desetinná místa a vynásobena stem. Následovně je přetypována na datový typ integer. Hodnota po provedení této operace uchovává hodnotu 2337. Desetinné místo bylo ztraceno z důvodu definice integeru, který ukládá pouze celočíselné hodnoty.

- Byly vypočítány hodnoty pro pozice všech čtyř rohů. Každý bod vyskytující se v rovině je definovaný dvěma čísly, reprezentujícími souřadnice bodu v rovině, tedy hodnotami na ose X a Y. Výsledkem je osm hodnot. Výpočty probíhají sčítáním souřadnic levého spodního rohu objektu s hodnotami velikosti objektu (šířka a výška objektu).

4.3.2.1 Uložení dat

Všechny objekty vytvořené třídou Piece byly uloženy do kolekce (kolekce slouží k seskupení a správě objektů). Pro lepší správu byly objekty v kolekci seřazeny vzestupně podle pozice velikosti souřadnic levého spodního rohu. Objekty jsou v této fázi kompletně připraveny pro další výpočty.

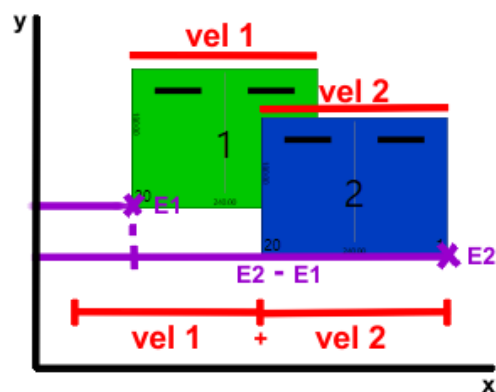
4.3.3 Kolize objektů na formě

První výpočty, které je nutno provést s každou formou, ještě před zahájením výpočtu, poskytnou odpověď na otázku, zda lze forma rozdělit. Tento krok zajistí, aby se objekty na formě vzájemně nepřekrývaly. Návrh přišel s řešením, které porovná všechny objekty v kolekci a ptá se na to, jestli se nepřekrývají. Jsou porovnávány vždy dva objekty. První objekt bude reprezentovat neznámá o_1 a druhý objekt o_2 . V prvním kroku jsou nalezeny extrémní pozice hran ve dvojici objektů. To znamená hrany vyskytující nejvíce vlevo, vpravo, nahoře a dole. Následuje sečtení stejně orientovaných velikostí hran o_1 a o_2 . Výsledná hodnota nesmí být větší, nebo rovna rozdílu vypočtených extrémů. To, jaké extrémní je nutno od sebe odečíst, závisí na tom, jaké velikosti hran byly sečteny.

Příklad pro zjištění kolize na ose X:

Je zjištěno, která strana objektu má nejmenší pozici na ose X a která má pozici naopak největší. Na obrázku číslo 12 jde o pozice pod názvy E1 a E2. Tyto hodnoty jsou uloženy. Následuje sečtení délek objektů a porovnání tohoto součtu s číslem získaným odečtením levého extrému od pravého. Délky jsou na obrázku číslo 7 popsány jako vel 1 a vel 2. Jestliže je součet

délek vel 1 a vel 2 větší než rozdíl pozic E2 a E1, objekty se překrývají stejně tak, jak je vidět na obrázku číslo 7. Nastane-li tato situace, program zastaví veškeré výpočty a zobrazí uživateli upozornění o tom, že se objekty překrývají. Uživatel



Obrázek 12 - Výpočet kolize dvou objektů (zdroj: Vlastní zpracování).

následně musí návrh upravit tak, aby se objekty nepřekrývaly, a dále musí znovu spustit výpočet.

Pro vyřešení problematiky kolize objektů na ose X je za potřebí tento postup aplikovat ještě pro kolize na ose Y.

4.3.4 Separace formy

Při řezání papíru vznikají každým provedeným řezem dvě skupiny. Stejným způsobem pracuje i aplikace. Návrh musí umět najít, analyzovat a zpracovat skupiny, které by mohly vzniknout provedením řezu.

4.3.4.1 Nalezení hlavní skupiny objektů

Nalezení řešení pro otázku, jakým způsobem hledat místo, kde je nutno zkoumat, jestli je možno provést řez, vedlo ke třem návrhům, z nichž se poslední stal funkčním a nejefektivnějším. Všechny návrhy byly založeny na filozofii nalezení hlavní skupiny objektů, pomocí níž lze zkoumat, zda se arch papíru dá dále dělit.

Hlavní skupina objektů

Hlavní skupina je množina objektů, které mají nejnižší pozici na formě. Při hledání hlavní skupiny objektů lze vždy nalézt dvě hlavní skupiny. První je orientovaná podle osy X, druhá podle osy Y. Na obrázku číslo 13 jsou znázorněny dvě hlavní skupiny. Je zřetelné, že v hlavní skupině X jsou jen ty objekty, které mají spodní hranu nejbližší k ose X. Stejně tak do hlavní skupiny Y patří jen ty objekty, jejichž levá hrana je



Obrázek 13 - Forma s vyznačenými hlavními skupinami na ose X a Y (zdroj: Vlastní zpracování).

nejbližší ose Y. Návrh pro hledání hlavních skupin musí ošetřit to, aby se do skupiny těchto objektů nevmíchal jiný objekt. Nalezené objekty jsou uloženy do nových kolekcí, které jsou předány k dalším výpočtům a analýzám.

Návrh funkcionality pro hledání hlavních skupin objektů

Jak již bylo výše zmíněno, byly navrženy tři návrhy, z nichž byl nakonec aplikován ten poslední. Níže jsou popsány všechny zvažované možnosti.

1. Návrh pracující pouze s pozicí levého spodního rohu objektu

Tento se zdál funkčním jen pro objekty rozložené na formě přesně tak, jak je to na obrázku číslo 8. Při testování tohoto návrhu byly do skupiny hlavních objektů chybně přiřazovány nesprávné objekty. Docházelo k tomu z toho důvodu, že navrhovaný algoritmus hledal a přiřazoval do hlavní skupiny X takové objekty, které měly nejnižší polohu levého spodního rohu na ose Y pro všechny levé spodní rohy objektů na ose X.

Jestliže byly objekty zarovnané tak, jak ilustruje obrázek číslo 7, vše fungovalo tak, jak mělo. Problém by však nastal, pokud byl například objekt číslo 8 posunut po ose X o pár milimetrů. Algoritmus by vyhodnotil nejnižší pozici levého spodního rohu objektu 8 na ose X i jako nejnižší hodnotu pozice na ose Y a objekt by chybně přiřadil do skupiny.

Tento problém nastával i při hledání hlavní skupiny Y.

2. Návrh pracující s hranou objektu

První návrh musel být přepracován a bylo nutné začít počítat s hranou objektu. Druhý návrh do množiny hlavní skupiny X zařazoval jen ty objekty, jejichž pozice spodní hrany se nejvíce blíží ose X. Vyhovující objekty hledal tím způsobem, že prohledával všechny body plátna. Při nalezení levé spodní hrany objektu hledal, jestli neexistuje nějaký jiný objekt s nižší hodnotou Y v rozmezí délky objektu. Když se tak nestalo, objekt přiřadil do hlavní skupiny objektů. Pokud by objekt s nižší hodnotou Y našel, na dosavadní objekt by zapomněl a testoval by to samé pro nově nalezený objekt. Tímto způsobem bylo zajištěno bezchybné nalezení hlavní skupiny. Nevýhodou však byla velká výpočetní náročnost procesu, způsobená procházením každého bodu plátna.

3. Návrh zefektivnění výpočtu

Pro zkrácení výpočetní doby bylo zapotřebí eliminovat procházení každého bodu plátna. Algoritmus nového návrhu byl vymyšlen tak, aby počítal pouze

s údaji v poskytnuté kolekci. Logika algoritmu je stejná jako v druhém návrhu s tím rozdílem, že se dotazuje, jestli je v okolí nejmenšího nalezeného objektu jiný objekt, který má pozici spodní hrany blížíící se ose X menší než první objekt. Volba hrany objektu, která bude porovnávána, s osou X nebo Y, opět závisí na typu hledané hlavní skupiny. Pro hlavní skupinu X je porovnávána spodní hrana objektu s osou X. Pro hlavní skupinu Y je porovnávána levá hrana objektu s osou Y.

Touto inovací byly ušetřeny bezmála čtyři sekundy výpočetního času.

4.3.4.2 Dělení kolekcí objektů do základních skupin

Samotné dělení kolekcí provádí komplexní algoritmus zajišťující, aby nebyl proveden žádný řez, který by poškodil jakýkoliv objekt v kolekci. Při prvním dělení, které aplikace provádí, je v kolekci množina všech objektů formy. Pro správné fungování návrhu je předpokladem využití funkcionality nalezení hlavní skupiny objektů.

Algoritmus pracuje s hlavní skupinou X nebo Y. Hledá, jestli je možné provést mezi objekty řez, který nepoškodí žádný objekt. Dochází k tomu tak, že z kolekce hlavní skupiny postupně zkoumá objekt po objektu, až do té doby, dokud nenalezne uskutečnitelný řez. Pro analýzu objektů hlavní skupiny X se algoritmus dotazuje na pozici pravého spodního rohu. Z této hodnoty získá jen souřadnice na ose X a přičte k hodnotě jeden milimetr. Poté se ptá, jestli v dělené kolekci, existuje objekt, který leží na souřadnici X. Tato podmínka může nabývat třech stavů.

1. Žádný objekt neleží na souřadnici

Jestliže žádný objekt neleží na souřadnici X, je zastaveno další prohledávání a je proveden na této souřadnici řez po ose Y, který rozdělí kolekci na další dvě skupiny. Tyto skupiny jsou dále zkoumány pro potřeby toho, zda se jedná o skupinu základní.

2. Alespoň jeden objekt leží na souřadnici

Jestliže alespoň jeden objekt leží na souřadnici X, řez nelze provést. Algoritmus přičte k souřadnici X další milimetr. V případě, že se na souřadnici X vyskytuje jiný objekt, proces je opakován až do té doby, než na souřadnici X neleží žádný objekt, nebo se zvýšená souřadnice X nerovná

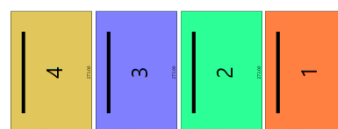
souřadnici levého spodního rohu objektu kolekce hlavní skupiny, který je další v pořadí. Pro tento objekt platí stejné postupy, jako to bylo u prvního objektu.

3. Nelze dělit skupinu po ose

Jestliže nastane situace, kdy mezi žádným objektem hlavní skupiny X není ani jeden proveditelný řez, je stejným způsobem zkoumána hlavní skupina, s tím rozdílem, že se řez provádí po ose X a sledovanou souřadnicí je souřadnice Y. Nastane-li situace, kdy je zjištěno, že nelze provést žádný z řezů orientovaný po ose X, celá forma je vyhodnocena jako nerozdělitelná.

Základní skupina objektů

Za základní skupinu objektů je prohlášena ta kolekce, ve které jsou objekty uspořádány buď do jednoho sloupce,



nebo do jednoho řádku. Tato skupina se dá prohlásit za stoprocentně dělitelnou. Na obrázku číslo 14 je vidět

Obrázek 14 - Základní skupina objektů (zdroj: Vlastní zpracování).

reprezentace základní skupiny objektů. Objekty nemusí být přesně zarovnané, nesmí se však dostat do toho stavu, že jeden objekt bude tvořit svou vlastní řadu, či sloupec. Kdyby nastala ta situace, že zelený objekt číslo 2 na obrázku číslo 14, by byl posunut o vzdálenost větší, než je hodnota výšky oranžového objektu číslo 1, vytvořil by tak další řádek a tím pádem, by tato skupina nebyla základní.

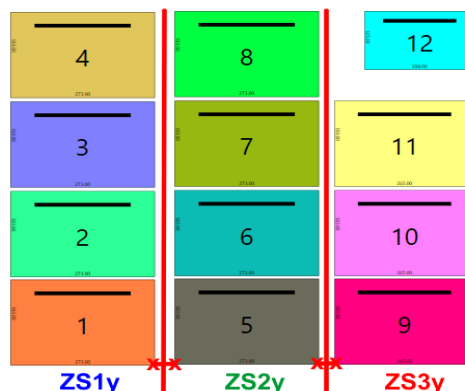
Je-li zkoumaná skupina skupinou základní, je zjištěna její orientace, podle toho, jestli jsou objekty zarovnané do sloupce či řádku. Informace spolu s kolekcí je uložena do třídy Dictionary (Přeložené do českého jazyka jako Slovník, který představuje kolekci klíčů a hodnot (Microsoft, 2017d).). Jako klíč je použita orientace skupiny. Hodnotu ke klíči zastává kolekce objektů základní skupiny. Takto je ošetřeno uchování a kontrola všech základních skupin.

Výsledek dělení skupin

Výše uvedený proces probíhá do té doby, než je výsledkem dělení skupin buď slovník naplněný základními skupinami, nebo upozornění, které sděluje uživateli, že forma nelze rozřezat.

- Slovník musí obsahovat tolik základních skupin, aby pro provedení celé operace každý objekt ze vstupní kolekce všech objektů patřil do jedné ze základních skupin. Pokud je tato podmínka platná, formu lze rozřezat a aplikace si uchovává rozdělené kolekce, které je možno bez problému identifikovat a pracovat s nimi.
- Upozornění o nemožnosti rozřezat formu je zobrazeno uživateli, pokud není možné vytvořit slovník skládající se jen ze základních skupin. Jestli je nějaká skupina nedělitelná a není zároveň základní, forma nelze rozřezat.

Obrázek číslo 15 znázorňuje, jak algoritmus rozdělí formu do základních skupin. ZS1 zastupuje název pro první základní skupinu. Písmeno y reprezentuje orientaci skupiny. Takto rozdělená forma je uložena do slovníku, který bude v tomto případě obsahovat tři základní skupiny.



Obrázek 15 - Rozdělení formy do základních skupin (zdroj: Vlastní zpracování).

4.3.5 Nejmenší počet řezů

Výpočet nejmenšího počtu řezů operuje se slovníkem, který byl vytvořen při předchozích výpočtech. Pro výpočet jsou již v této chvíli zajištěna všechna potřebná data. Nejproblematictější fází tvorby aplikace byla otázka, jakým způsobem návrh uchopit. Pro výpočet nejmenšího možného počtu řezů bylo zapotřebí vymyslet, jakým způsobem zajistit, to, aby jedním řezem bylo ořezáno více objektů. Velkým problémem byla rozmanitost vztahů, které k sobě objekty mohou vzájemně mít.

První návrh vedl k neúspěchu kvůli špatnému zamyšlení se nad problematikou, a to kvůli velkému počtu stavů, do kterých se objekty mohou dostat. Celý návrh byl zpočátku chápán z hlediska skupin, které jsou na formě jako celek. Oproštění se od tohoto konceptu vedlo autora k finálnímu návrhu, který se nezaměřil nad formou jako nad celkem. Vychází z toho poznatku, že ve slovníku jsou jen ty skupiny, které lze jistě rozřezat. Když bude řezač postupovat ve správném pořadí, nikdy se nemůže stát to, že by byl objekt řezem poškozen. Toto je zajištěno z toho důvodu, že se se skupinami, které jsou vytvořeny řezem, dá vždy manipulovat.

4.3.5.1 Algoritmus pro výpočet řezů

Každý objekt v sobě, díky třídě `Piece.cs`, uchovává informaci o tom, jestli byl z nějaké strany ořezán. Algoritmus poté s těmito informacemi pracuje. Řez je zde reprezentován změnou atributu, nesoucí tuto informaci. Výpočet probíhá tak, že ze slovníku základních skupin je vyňata první skupina a ta je testována. Aplikace se dotazuje na pozice prvního ze skupiny objektů a na orientaci skupiny.

Orientace skupin

Orientace skupiny je klíčová. Určuje, kterým směrem je vždy bezpečné rozřezat skupinu. Pro základní skupiny orientované podle osy Y, jsou bezpečné vodorovné řezy. Není možné, aby byl jakýkoliv objekt skupiny tímto řezem poškozen. Pro skupiny orientované podle osy X jsou svislé řezy těmi bezpečnými. S řezy, které skrývají potenciální riziko poškození jiného objektu, musí být prováděny další výpočty. Těmito řezy jsou myšleny řezy svislé pro skupiny orientované k ose Y a řezy vodorovné prováděné nad skupinami orientovanými k ose X.

Pozice objektů

Algoritmus nastaví všechny hodnoty objektu reprezentující ořezání na `true` (Do českého jazyka přeloženo jako slovo pravda). Tím zaznamenává to, že byl objekt kompletně ořezán. Současně nastaví všem objektům ve skupině, jejichž hrana leží na souřadnicích řezu, atributy příslušné hrany na `true`. Následně nastaví všem atributům u hran, nad kterými se dají provádět bezpečné řezy, hodnotu `true`. Každý řez je v této fázi připočítán k celkovému počtu minimálních řezů. V této situaci mohou mít všechny objekty své atributy nastavené na `true`. Znamenalo by to, že všechny objekty v této skupině jsou úspěšně ořezané.

Následuje nutnost zjistit a zaznamenat to, jak řezy provedené nad první skupinou ostatní objekty skupiny ovlivnily. Je za potřebí neustále uvažovat, že jeden řez je prováděn od jednoho rohu plátna k druhému. Lze jej tedy využít i k ořezání jiných objektů. Algoritmus si najde ty objekty, jejichž hrany ležely na pozici řezu. U těchto hran je nastaven atribut ořezání na `true`. Avšak tyto řezy jsou již do součtu všech řezů započítány.

V neposlední řadě program projde slovník hlavních skupin a vyřadí z nich ty objekty, které mají všechny hodnoty ořezání hran nastavené na hodnotu true.

Jestliže ve slovníku po této operaci nezůstane ani jeden objekt, aplikace uživateli zobrazí minimální počet řezů. Když tomu tak není, je proveden náhled do slovníku, výběr základní skupiny, obsahující aspoň jeden objekt a celý proces řezu opakován. Zobrazením výsledku veškeré výpočty končí a uživatel může zadat programu ke zpracování jiný dokument XML pro výpočet nejmenšího počtu řezů.

4.3.6 Jazykové mutace

Pro lepší práci s aplikací byl navržen přepínač jazykových mutací. Přepnutím jazyka uživatel dostává všechny informace a všechna upozornění programu ve zvoleném jazyce. Stejně tak jsou veškerá tlačítka přeložena do uživatelem zvoleného jazyka. Uživatel si sám v hlavním okně může zvolit ze tří jazyků: angličtina, němčina anebo čeština.

5 Shrnutí výsledků a jejich diskuze

V této kapitole je uvedena zpětná vazba společnosti OV-Media s. r. o., která vzešla ze schůzky, na které byla představena aplikace pro výpočet minimálního počtu řezů pro rozřezání archu papíru na potřebné části společnosti OV-Media s. r. o.

Dále kapitola obsahuje porovnání původního řešení s novým a shrnutí výsledků, které byly vytvořeny při testech finální verze aplikace.

5.1 Zpětná vazba společnosti OV-Media s. r. o.

Zpětnou vazbu poskytl Ing. Marek Veselý – ředitel společnosti OV-Media s. r. o. a níže je uvedeno jeho hodnocení.

V tiskárně zpracováváme denně velké množství zakázek, kdy je velký důraz kladen na rychlost a efektivitu zpracování tiskařských forem. Tyto formy umožňují efektivně využívat tiskovou kapacitu, zároveň ale často kladou velké nároky na řezání.

Technolog musí maximálně využít tiskový arch a rozmístit na něj co nejvíce produktů. Každá forma má pak skoré, které určuje, jak je daná forma efektivní z hlediska tisku. Hodnocení z hlediska řezání nebylo dosud využíváno.

Aplikace určující minimální počet řezů slouží technologovi jako vodítko pro posouzení, jak danou formu sestavil z hlediska náročnosti řezání. Počet řezů v praxi může být nižší, než určí aplikace. Je tomu tak proto, že při řezání částečně rozřezaných bloků může obsluha řezacího stroje v určitých případech, vytvořit efektivnější konstelaci bloků pro řezání. Toto je přínos zkušenosti obsluhy proti algoritmu. Tento problém by však řešila otázka nejefektivnějšího naskládání bloků do formy, která nebyla zadáním aplikace. Počet řezů dle aplikace je tedy teoreticky nejnižší počet řezů, které mohou nastat.

Pro úplné začlenění aplikace by bylo potřeba aplikaci dopracovat a více provázat s firemním workflow. V současnosti pracuje aplikace pouze s jednotlivými stranami, ale neumožňuje (vzhledem k dalšímu knihařskému zpracování) tyto strany zamykat do bloku. Lze tedy na arch připravit 16 stránkovou brožuru, kde na jednom archu je 8 samostatných stran, které se ale neoddělují, pouze se celý

blok ořeže po stranách. Tato funkcionalita však nebyla zadána ve specifikaci aplikace. Aplikace měla zpracovávat pouze výstupní informace z XML dokumentů, které výše popsané vlastnosti seskupování postrádají. Tyto změny zatím neplánujeme v brzké době realizovat. Nemůžeme tedy zaručit to, zda se autor práce bude podílet na dalším vývoji aplikace.

V neposlední řadě aplikace nabízí technologovi kontrolu, zda nenastal při řezání konflikt a forma je zpracovatelná. Vzhledem k požadavkům a specifikaci aplikace, byl úkol splněn.

5.2 Testování Aplikace

Pro test aplikace byla vytvořena testovací třída s celkem pěti testovacími metodami. Testovány byly všechny XML dokumenty poskytnuté společností. Testovací metody spouští algoritmus aplikace pro každý XML dokument a testují, zda jsou všechny výpočty správné. Testy také měří čas, který je potřebný pro výpočet řezů.

Výsledky testu finální verze aplikace jsou následující – Testováno bylo celkem pět XML dokumentů. Všechny výpočty proběhly bezchybně. Průměrná rychlost veškerých výpočtů potřebných pro zjištění minimálního počtu řezů byla 0,118 sekundy, což je oproti prvním návrhům aplikace, kde se výpočet pohyboval okolo dvou sekund nad očekávání.

5.3 Srovnání nového řešení s dosavadním procesem

Zavedení inovace do společnosti pozmění dosavadní proces pro rozmístění objektů na tiskařskou formu. Umožní pracovníkům efektivnější návrh formy spolu s využitím funkcionality zabráňující vytvoření chyby, která by vedla ke ztrátám společnosti.

5.3.1 Dosavadní proces

Při tvorbě návrhu tiskařských forem bylo spoléháno na zkušenosti a znalosti pracovníka v daném oboru. Návrh tedy spočíval buďto v rozmístění objektů do předdefinovaných šablon, nebo na volném umístění objektů do formy. Šablony

jsou definované tak, aby minimalizovaly výrobní náklady a maximalizovaly profit společnosti. Ve většině případů nelze využít předdefinovaných šablon, a to kvůli rozmanitosti zakázek, které společnost přijímá. Každá zakázka může mít svůj vlastní formát objektu. Je nemožné vytvořit soubor šablon, zahrnující kombinaci všech druhů zakázek. Každý návrh tiskařské formy může být unikátní. Pracovníci tedy rozmisťují objekty (zakázky) do formy tak, jak uznají za vhodné. Zpětnou vazbu na návrhu formy, dostanou až od pracovníka u řezacího stroje, který s formou operuje. Neefektivní návrh tiskařské formy vede k prodloužení výrobního procesu. Chybný návrh tiskařské formy navíc může vést k nemožnosti formu rozřezat. Touto chybou se společnost dostává do velkého časového skluzu, způsobeného opakovaním celého výrobního procesu.

5.3.2 Nové řešení

Nové řešení umožňuje pracovníkům při vytváření návrhu zjistit, zda bude možné vytvářený návrh formy rozřezat a tím se vyvarovat velkým ztrátám. Dále zajišťuje to, aby pracovník nevytvořil takový návrh, na kterém se jednotlivé objekty překrývají. Pracovník má také možnost během vytváření návrhu formy, spouštět výpočty pro zjištění minimálního počtu řezů, které je nutno provést k rozřezání aktuální tiskařské formy. Aplikace napomůže k efektivnějšímu návrhu díky okamžité zpětné vazbě. Využitím těchto funkcionalit lze zabránit chybnému návrhu tiskařské formy, tím maximalizovat profit společnosti a minimalizovat výrobní náročnost výrobních procesů.

6 Závěr

Úkolem práce bylo navrhnout a vytvořit aplikaci, která zefektivní výrobní procesy ve společnosti OV-Media s. r. o. Aplikace zajišťuje výpočet minimálního počtu řezů potřebných pro rozřezání archu papíru na určitý počet dílů. K vyřešení této problematiky bylo zapotřebí nejdříve analyzovat společnost, a především sektor výroby, pro který je aplikace navržena. Následně bylo nutné zodpovědět na otázku, zda lze arch papíru rozřezat tak, aby ani jeden definovaný díl nebyl poškozen. Řešení pro výpočet nejmenšího počtu řezů potřebných k rozřezání archu papíru na určitý počet dílů definovaných uživatelem kompletuje a uzavírá celou práci.

Vývoj byl konzultován s vývojářským oddělením společnosti. Vývojové oddělení napomohlo udat směr k zefektivnění a inovaci zdrojových kódů. Byla vytvořena dokumentace, která zobrazuje detailní strukturu aplikace. Analýza spolu s návrhem a tvorbou aplikace proběhla úspěšně.

7 Seznam použité literatury

- [1] Agnieszka, S., Pietraszek, J. a Radek, N. (2015) *Parametric and Non-Parametric Methods of Data Analysis at Multiscale Modeling*. s. 4 Zurich.
- [2] Barker, S., Cole, R. (2009) *Projektový management pro praxi*. vyd. 1, s. 155, Praha, Grada Publishing. ISBN: 978-80-247-2838-4.
- [3] Benson, J. D. (2014) 'Motivation, Productivity and Change Management', *Managing Business Scenarios*. roč. 1, s. 1–6 [online] ISBN: 9781429835831. [Citováno dne: 27.3.2017].
- [4] Bigwood, M. (2004) 'Managing the new technology exploitation process'. *Research Technology Management*. roč. 47, č. 6, s. 38-42, [online], ISSN: 0895-6308. [Citováno dne: 27.3.2017].
- [5] Brown, R. (1992) 'Managing the 'S' curves of innovation.' *Journal of Consumer Marketing*. roč. 7, č. 2, s. 189-202, [online], ISSN: 0267-257X. [Citováno dne: 27.3.2017].
- [6] Dobel tiskárna (2016) [online]. Dostupné z: http://www.dobel.cz/obrazy/varianty_skladani.gif [Citováno dne: 14. 8. 2016]
- [7] Drucker, P. D. (2008) *Efektivní vedoucí*. vyd. 2, s. 205, Praha, Management Press. ISBN: 978-80-7261-189-8
- [8] Dziak, M. (2016) *Brainstorming*. Salem Press Encyclopedia, [online], Salem Press © 2016, [Citováno dne: 27. 3. 2017].
- [9] Ettorchi-Tardy, A., Levif, M. a Michel, P. (2012) 'Benchmarking: A Method for Continuous Quality Improvement in Health,' *Healthcare Policy*. roč. 7, č. 4, s. 101–119, [online]. [Citováno dne: 23. 4. 2017].
- [10] Flynn, S. I. (2015) 'Managing the Process of Innovation,' *Research Starters Business*. [online] Research Starters Business Online Edition. [Citováno dne: 27. 3. 2017]

- [11] French, W. L., Bell, C. H. a Zawacki, R. A. (2005). ‚Organization development and transformation‘: *Managing effective change*. roč. 6, s. 388 New York, ISBN: 0071112669.
- [12] Harmon, A. (2016) *Swot analysis*. Salem Press Encyclopedia, [online], Salem Press © 2016, [Citováno dne: 27. 3. 2017].
- [13] Christofides, N. a Hadjiconstantinou, N. (1995) ‚An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts,‘ *European Journal of Operational Research*. roč. 83, s. 21-38, [online], ISSN: 0377-2217. [Citováno dne: 14. 8. 2016].
- [14] Jáč, I., Rydlová, P. a Žižka, M. (2005) *Inovace v malém a středním podnikání*. vyd. 1, s. 169, Brno, Computer Press. ISBN: 80-251-0853-8
- [15] Johnson, G. a Leavitt, W. (2001) ‚Building on Success: Transforming Organizations Through an Appreciative Inquiry‘. *Public Personnel Management*. roč. 30, č. 1, s. 129-136, [online], ISSN: 0091-0260. [Citováno dne: 27.3.2017].
- [16] Kotter, J. P (1998) ‚Winning at change‘. *Leader to Leader*. roč. 10, s. 27-33, [online], ISSN: 1087-8149. [Citováno dne: 27.3.2017].
- [17] Kraj Vysočina (2017) *Předběžná analýza prostředí pro stanovení inovační strategie v kraji Vysočina*. [online], kr-vysocina.cz © 2002-2013, s. 3. [cit. 2017-03-28]. Dostupné z: [https://www.kr-vysocina.cz/VismoOnline ActionScripts/File.ashx?id org=450008&id dokumenty=4000931](https://www.kr-vysocina.cz/VismoOnline>ActionScripts/File.ashx?id org=450008&id dokumenty=4000931) [Citováno dne: 27.3.2017].
- [18] Letenyei, L. (2001) ‚Rural innovation chains.‘ *Review of Sociology*. vyd. 7, s. 85-100, [online], ISSN: 1588-2845. [Citováno dne: 27.3.2017].
- [19] ManagementMania. (2017) *Analytické techniky (Analytical techniques)*. [online], ManagementMania.com. © 2011-2017, Dostupné z: <https://managementmania.com/cs/analyzy-analyticke-techniky> [Citováno dne: 27.3.2017].
- [20] ManagementMania. (2017) *Osm kroků změny (Eight Step Change Model)*. [online], ManagementMania.com. © 2011-2017, Dostupné z: <https://managementmania.com/cs/osm-kroku-zmeny> [Citováno dne: 27.3.2017].

- [21] Mercadal, T. (2016) *Benchmarking*. Salem Press Encyclopedia, [online], Salem Press © 2016, [Citováno dne: 27. 3. 2017].
- [22] Microsoft. (2017a) float, *Referenční dokumentace jazyka C#*. [online], Microsoft © 2017, Dostupné z: <https://msdn.microsoft.com/cs-cz/library/b1e65aza.aspx> [Citováno dne: 25.3.2017].
- [23] Microsoft. (2017b) bool, *Referenční dokumentace jazyka C#*. [online], Microsoft © 2017, Dostupné z: <https://msdn.microsoft.com/cs-cz/library/c8f5xwh7.aspx> [Citováno dne: 25.3.2017].
- [24] Microsoft. (2017c) int, *Referenční dokumentace jazyka C#*. [online], Microsoft © 2017, Dostupné z: <https://msdn.microsoft.com/cs-cz/library/5kzh1b5w.aspx> [Citováno dne: 26.3.2017].
- [25] Microsoft. (2017d) Třída Dictionary(TKey, TValue), *System.Collections.Generic*. [online], Microsoft © 2017, Dostupné z: [https://msdn.microsoft.com/cs-cz/library/xfhwa508\(v=vs.110\).aspx](https://msdn.microsoft.com/cs-cz/library/xfhwa508(v=vs.110).aspx) [Citováno dne: 27.3.2017].
- [26] Microsoft. (2017e) COM: Component Object Model. [online], Microsoft Corporation © 2017, Dostupné z: <https://www.microsoft.com/com/default.aspx> [Citováno dne: 24.4.2017].
- [27] Nagel, C., Evjen, B., Glynn, J., Skinner, M. a Watson, K. (2009) *C# 2008 – Programujeme profesionálně*. vyd. 1, s. 1898, Brno, Computer Press. ISBN: 978-80-251-2401-7.
- [28] Oliveira, J. F. a Ferreira, J. S. (2012) 'Integrated resolution of assignment, sequencing and cutting problems in paper production planning.' *International Journal of Production Research*. roč. 50, č. 18, s. 5195-5212, [online], ISSN: 0020-7543. [Citováno dne: 14. 8. 2016].
- [29] Plamínek, J. (2008) *Řešení problému a rozhodování*. vyd. 1., s. 144, Praha, Grada Publishing. ISBN: 978-80-247-2437-9.
- [30] Scientific & Academic Publishing Co. (2017) *Brainstorming*. [online], Scientific & Academic Publishing Co. © 2011-2017, Dostupné z: <http://article.sapub.org/10.5923.j.mm.20120204.05.html> [Citováno dne: 24.4.2017].

- [31] Spender, J. a Kessler, E. (1995) 'Managing the uncertainties of innovation: Extending Thompson (1967).' *Human Relations*. roč. 48, s. 35-57, [online], ISSN: 0018-7267. [Citováno dne: 27.3.2017].
- [32] Stalk, Jr. G. (2006) 'Hardball innovation.' *Research Technology Management*. roč. 49, s. 20-28, [online], ISSN: 0898-6308. [Citováno dne: 27.3.2017].
- [33] Statistica Neerlandica (2009) *The two-dimensional cutting stock problem within the roller blind production proces*. roč. 63, č. 4, s. 474-489, [online]. [Citováno dne: 25. 4. 2017].
- [34] Tapinos, E., Dyson, R. G. a Meadows, M. (2005) 'The impact of performance measurement in strategic planning.' *International Journal of Productivity and Performance Management*. roč. 54, č. 6, s. 370-384, [online], ISSN: 1741-0401. [Citováno dne: 27.3.2017].
- [35] TechTarget. (2017) *Client/server*. [online], TechTarget © 2000-2017, Dostupné z: <http://searchnetworking.techtarget.com/definition/client-server> [Citováno dne: 24.4.2017].
- [36] Yoon, K., Ahn, S. a Kang, M. (2013) 'An improved best-first branch-and-bound algorithm for constrained two-dimensional guillotine cutting problems.' *International Journal Of Production Research*. roč. 51, č. 6, s. 1680-1693, [online], ISSN: 0020-7543. [Citováno dne: 14. 8. 2016].
- [37] Zahrah, S. a Covin, J. (1993). 'Business strategy, technology policy and firm performance.' *Strategic Management Journal*. roč. 14, s. 451-478, [online], ISSN: 0143-2095. [Citováno dne: 27.3.2017].

8 Přílohy

- 1) Kompaktní disk obsahující aplikaci pro minimální výpočet řezů, její dokumentace a zdrojové kódy aplikace.

Zadání bakalářské práce

Autor: Michal Zaccpal

Studium: I1301222

Studijní program: B1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název bakalářské práce: **Analýza společnosti OV Media s.r.o. a možnosti zefektivnění jejího výrobního procesu**

Název bakalářské práce AJ: Analysis of the company OV Media and options of improving its manufacturing processes

Cíl, metody, literatura, předpoklady:

Osnova: I. Úvod II. Cíl práce a metodologie III. Teoretická východiska IV. Návrh a tvorba aplikace V. Shrnutí výsledků a jejich diskuze VI. Závěry a doporučení VII. Seznam použité literatury Cíl práce: Cílem práce je navrhnout a vytvořit aplikaci, která umožní zvýšit efektivitu návrhových procesů. V teoretické části je kladen důraz na otázku zefektivnění návrhových procesů, spolu s představením a srovnáním dostupných nástrojů. V praktické části je řešen návrh a tvorba aplikace.

KADLEC, Václav. Učíme se programovat v jazyce C: základ pro programování v C++, C#, Javě, JavaScriptu, PHP a jiných jazycích. Vyd. 1. Praha: Computer Press, c2002, xiii, 277 s. Programování. ISBN 80-7226-715-9. SEDGEWICK, Robert. Algoritmy v C. Praha: Softpress, 2003, 688 s. ISBN 8086497569. PLAMÍNEK, Jiří. Řešení problémů a rozhodování: jak přinutit problémy, aby pracovaly ve váš prospěch. 1. vyd. Praha: Grada, 2008, 144 s. Manažer. ISBN 978-80-247-2437-9. BARKER, Stephen a Rob COLE. Projektový management pro praxi. 1. vyd. Praha: Grada, 2009, 155 s. Management (Grada). ISBN 978-80-247-2838-4.

Garantující pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: Ing. Tereza Otčenášková, BA

Datum zadání závěrečné práce: 15.3.2014