

ŠKODA AUTO VYSOKÁ ŠKOLA o.p.s.

Studijní program: N0413A050001 Ekonomika a management

Studijní obor/specializace: Specializace Řízení mezinárodních dodavatelských řetězců

Předávání projektových informací a optimalizace řízení projektu s využitím VBA v MS Excel Diplomová práce

Bc. David Hloušek

Vedoucí práce: doc. Ing. Jan Fábry, Ph.D.

V tištěné verzi závěrečné práce tento list vyjměte a nahradte zadáním závěrečné práce. V elektronické verzi práce zde vložte oskenované zadání se všemi podpisy.

Prohlašuji, že jsem závěrečnou práci vypracoval(a) samostatně a použité zdroje uvádím v seznamu literatury. Prohlašuji, že jsem se při vypracování řídil(a) vnitřním předpisem ŠKODA AUTO VYSOKÉ ŠKOLY o.p.s. (dále jen ŠAVŠ) směrnici OS.17.10 Vypracování závěrečné práce.

Jsem si vědom(a), že se na tuto závěrečnou práci vztahuje zákon č. 121/2000 Sb., autorský zákon, že se jedná ve smyslu § 60 o školní dílo a že podle § 35 odst. 3 je ŠAVŠ oprávněna mou práci využít k výuce nebo k vlastní vnitřní potřebě. Souhlasím, aby moje práce byla zveřejněna podle § 47b zákona č. 111/1998 Sb., o vysokých školách.

Beru na vědomí, že ŠAVŠ má právo na uzavření licenční smlouvy k této práci za obvyklých podmínek. Užiji-li tuto práci, nebo poskytnu-li licenci k jejímu využití, mám povinnost o této skutečnosti informovat ŠAVŠ. V takovém případě má ŠAVŠ právo ode mne požadovat příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to až do jejich skutečné výše.

V Mladé Boleslavi dne 16.10.2021

Vlastnoruční podpis

Děkuji doc. Ing. Janu Fábrymu, PhD. za odborné vedení diplomové práce, poskytování rad a trpělivosti, kterou se mnou měl při zpracování diplomové práce. Nesmím opomenout také poděkování za poskytnutí odborných informací od kolegů z oddělení vývoje automatických převodovek EPG/3 ze ŠKODA AUTO a.s.

Obsah

Úvod.....	7
1 Projektový management.....	9
1.1 Projektový manažer.....	9
1.2 Historie projektového managementu.....	11
1.3 Projekt.....	13
1.4 Různé formy projektového managementu.....	16
1.5 Organizační struktura projektu.....	19
1.6 Tradiční způsob vedení projektu.....	21
1.7 Agilní způsob vedení projektu.....	23
1.8 Porovnání tradičního a agilního způsobu vedení projektu.....	29
1.9 Vliv pandemických situací na projektové řízení.....	31
2 Visual Basic for Application.....	33
2.1 Všeobecné využití VBA.....	34
2.2 Výhody a nevýhody VBA.....	34
3 Představení oddělení EPG/3 ve společnosti ŠKODA AUTO a.s.....	36
4 Aplikace a systémy používané pro předávání projektových informací.....	39
4.1 Situace před zavedením změn.....	39
4.2 Přiřazovací tabulka SW.....	40
4.3 Situace po zavedení změn.....	61
4.4 Dodržovaná pravidla při tvorbě VBA.....	62
4.5 Porovnání agilního způsobu řízení s/bez pomoci VBA for Excel.....	65
Závěr.....	67
Seznam literatury.....	69
Seznam obrázků a tabulek.....	72
Seznam příloh.....	74

Seznam použitých zkratk a symbolů

BASIC	Beginner's All-purpose Symbolic Instruction Code
EOP	End of production
FTP	File Transfer Protocol
MS	Meilenstein – milník
MS4-LF	Launchfreigabe – uvolnění pro náběh
PMBok	Project Management Body of Knowledge
PMI	Project Management Institute
PRINCE2	Projects in Controlled Enviroments
PVS	Produktionsversuchsserie – zkušební výrobní série
Serie	Sériový software pro výrobní nultou sérii vozů
SOP	Start of production
SW	Software
ŠA	ŠKODA AUTO
VBA	Visual Basic for Application
VBE	Visual Basic Editor
VFF	Vorserienfreigabe Fahrzeuge – předsériové uvolnění vozu
Vorserie	Předsériový software pro prototypové vozy
VW	Volkswagen

Úvod

Tato práce se zaměřuje na předávání projektových informací a optimalizaci řízení projektu s využitím prostředí Visual Basic For Application (dále jen VBA). Jedná se o konkrétní oddělení ve společnosti ŠKODA AUTO a.s. (dále jen ŠA), které se zabývá vývojem automatických převodovek. Oddělení se zabývá konkrétně především kalibrací softwaru, a tedy i jeho správou, která skýtá stovky kombinací převodovek a motorizací, jež mají vlastní variantu softwaru.

Cílem práce je nalézt optimalizační model pro předávání informací pomocí agilních metod, zajistit stabilní a bezproblémové fungování navrhovaného modelu, který bude vytvořen pomocí naprogramované aplikace ve VBA for Excel. Původním cílem bylo vytvořit pouze jistou formu tabulky, která se při plánování změnila v komplexní tabulku s maximálním možným podílem automatizace a kontrolování určitých náležitostí jednotlivých projektů pro usnadnění jejich správy.

Hlavním důvodem výběru tohoto tématu bylo vymyšlení vlastního řešení pro zavedení agilních metod při zajišťování dostatečné informovanosti o jednotlivých projektech bez nutnosti dotazovat se jiných členů týmu. Na základě nabytých zkušeností během dvou let působení na daném oddělení bylo možné definovat úzká místa, a tak identifikovat vhodná řešení.

Problematika bude nejprve řešena konzultacemi s jednotlivými členy daného oddělení, kde cílem bude získání dostatečných informací. Na základě těchto konzultací bude navrženo vhodné zlepšení komunikace a celkové informovanosti celého týmu automatizovanou tabulkou v prostředí VBA for Excel, která bude sloužit jako nástroj pro snazší správu projektů.

Teoretická část se bude věnovat projektového managementu. Konkrétně historie, co si představit pod pojmem projekt a jak takový projekt funguje. Důležitá bude organizační struktura projektu a seznámení se s tradičním a agilním způsobem vedení projektu. Tyto dva způsoby budou porovnány mezi sebou. Ve druhé kapitole bude definován pojem VBA, především jeho hlavní přednosti a výhody oproti klasickému používání pouhých vzorců a podmíněného formátování.

Praktická část se již bude zabývat v krátkosti představením společnosti ŠA a oddělením EPG/3. Následovat bude kapitola o situaci před zavedením návrhu a poté obeznámení s nalezenými úzkými místy. V návaznosti na úzká místa bude

uveden rozbor návrhu tabulky, která by měla posloužit nejen jako pouhý seznam projektů, ale bude také schopna provádět jistou formu jejich správy a sledování změn. Poté se vyhodnotí situace po zavedení daného návrhu.

V této práci je nutné brát ohled na citlivost používaných údajů společnosti ŠA, a z toho důvodu některé informace, které nejsou přímo nutné k vyobrazení, budou vyjmuty nebo skryty, případně nahrazeny informacemi obecnějšího charakteru.

1 Projektový management

Projektový management se zabývá organizováním, plánováním, integrací zdrojů a úkolů pro dosažení cílů organizace. V každém projektovém managementu existuje osoba, které se říká projektový manažer a má zodpovědnost za vedení projektu, ať už se jedná o liniové manažery, režiséra filmu nebo projektového manažera, mají tu stejnou roli (Nicholas a Steyn, 2017).

Čím dál tím větší množství podniků využívá projektový management. Podniky totiž přecházejí na tuto možnost z běžného operativního řízení, jelikož je to pro ně v mnoha případech vhodnější. Na rozdíl od operativního řízení totiž musí mít projekt vždy nějaký cíl a jakmile je cíle dosaženo, projekt končí. Všechny pracovní, finanční nebo technologické zdroje, které zbydou po ukončení projektu, jsou převedeny nebo spotřebovány pro účely jiného projektu. Operativní řízení funguje odlišně, tedy pokud je dosaženo cíle, jsou stanoveny nové a vše pokračuje (Svozilová, 2016).

1.1 Projektový manažer

Funkce manažera je možné rozdělit do pěti funkcí, které by měl plnit pro správné fungování a zajištění dosažení cíle projektu.

Manažer plánuje průběh projektu, co je nutné provést za činnosti a stanovuje cíle tak, aby korespondovaly s cíli organizace na vyšší úrovni. Plánuje, jakým způsobem budou přidělovány zdroje a jaká jsou omezení.

Další v pořadí je organizační funkce, která úzce souvisí s plánováním, protože obnáší zajištění práce (tedy jednoho ze zdrojů projektu), která musí být provedena. Manažer má na starosti zajišťovat, koučovat, a především shlukovat lidi do týmu. Měl by do týmu dostávat takové lidi, na které bude spolehnout, a hlavně budou schopni mít za určité činnosti zodpovědnost. Krom práce musí také zajistit potřebné stroje a zařízení, patřičný kapitál a materiál potřebný pro dané činnosti projektu. V rámci organizační funkce je nutné také vytvořit organizační strukturu týmu, ať jsou jasně rozdělené zodpovědnosti a role v týmu (Nicholas a Steyn, 2017). Součástí této struktury je nutné také vytvořit určitá pravidla, zásady, způsob komunikace a postupy.

Pro dosažení cílů je nutná také motivace a řízení lidí k dosažení stanovených cílů. Právě tyto činnosti spadají pod vedoucí funkci. Manažer nemůže pouze nečinně

přihlížet na proces, ale musí se aktivně zapojovat, a to právě tím, že bude povzbuzovat členy týmu ke stálému výbornému pracovnímu výkonu a ovlivňovat tak jejich pracovní nasazení a chování. Nebude se zaměřovat pouze na jeho přímé podřízené, ale na všechny pracovníky, kteří jsou součástí týmu, potažmo projektu.

Kontrolní funkce je také nezbytnou součástí funkcí manažera. Krom motivace lidí k dosažení cílů, je také nutné sledovat, zda se pracovní výkon využívá stále k tomu původně určenému stanovenému cíli projektu. V případě, že tomu tak není, musí přijmout patřičná opatření. Pro zjednodušení kontroly využívá manažer kritéria cílů, plány a především náklady, které v průběhu projektu vznikají. Při dobře nastavené organizační struktuře dokáže manažer snadno zjišťovat, kde může problém vznikat (Svozilová, 2016).

Poslední, neméně důležitou funkcí, je hodnocení těch předchozích čtyř funkcí, které se zaměřují na plnění cílů a jak těchto cílů dosáhnout. Hodnocení tedy dává zpětnou vazbu, kde je nutná změna k co nejefektivnějšímu dosažení cíle projektu jako takovému.

Charakteristiky projektového managementu

Základní charakteristikou projektového managementu je vedení jedním člověkem, tedy projektovým manažerem, jak již bylo popsáno výše. Řídí organizaci projektu a pracuje nezávisle na obvyklém velení. Organizací projekt představuje cílově orientovanou a dočasnou jednotku.

Projektový manažer zodpovídá za sestavení týmu, kde členové mohou být z různých funkčních oblastí nebo dokonce být externími dodavateli služeb. Jeho úkolem je sestavit tým takovým způsobem, aby zajistil co nejproduktivnější skupinu lidí pro efektivní dosahování cílů projektu. Jeho úkolem je tyto lidi zakomponovat do organizační struktury, aby každý pracoval na tom, co mu půjde nejlépe.

Mezitím, co se projektový manažer zaměřuje na dodání určitého produktu nebo služby ve stanoveném datu a s určitými náklady, funkční manažeři zodpovídají za svou skupinu pracovníků a zdrojů, které jim byly přiděleny pro jejich danou oblast. Může tedy docházet k problémům mezi funkčními a projektovými manažery právě kvůli přerozdělování zdrojů v rámci projektu.

V rámci projektu se může pracovník zodpovídat dvěma nadřízeným – vedoucímu projektu, tedy projektovému manažerovi a zároveň funkčnímu manažerovi, tedy jeho přímému nadřízenému. Může kvůli tomu vzniknout problém s prioritizací úkolu, protože projektový manažer je výše postavený (Nicholas a Steyn, 2017).

Veškeré možné odměny, zodpovědnosti a rozhodování, jsou sdíleny napříč všemi zúčastněnými v projektu. Jakmile se dosáhne cíle, případně cílů projektu, jsou veškeré alokované zdroje pro projekt umístěny zpět na svá původní funkční místa, jelikož po jeho ukončení je organizace projektu rozpuštěna.

1.2 Historie projektového managementu

Z dob stavby egyptských pyramid se datuje evoluce projektového managementu, který se později zlepšoval a modifikoval pro použití v dnešní době.

Začátkem dvacátého století se ukázalo, že projektový management lze používat i na jiné druhy práce, nejenom pro stavby budov, ale také pro designování a testování nových produktů. Prakticky je možné jej používat na jakoukoliv činnost, která bude trvat více než několik dní a zahrnuje několik dílčích úkolů. Velký posun ve vývoji projektového managementu nastal v době první světové války, kdy se používané techniky zlepšily a vznikl Ganttův diagram. V návaznosti na něj se po zhruba 30 letech objevil diagram zvaný „process flow diagram“ (Nicholas a Steyn, 2017), tedy vývojový diagram procesu.

V 50. letech 19. století začaly projekty nabírat na komplexnosti a současné techniky nebyly dostatečně vyvinuté na to, aby umožnily správné fungování organizace a řízení projektu. Vznikaly problémy s nedodržováním termínů a překračováním stanovených nákladů. Na základě těchto problémů se objevily dvě nové metody – CPM a PERT (Nicholas a Steyn, 2017), které sloužily právě pro plánování, kontrolu a plánování velkých projektů s mnoha provázanými aktivitami. Později se zkombinovaly i s počítačovými simulacemi.

V návaznosti na metody CPM a PERT se začalo více využívat počítačových simulací, které znatelně zlepšily možnost řízení informačních toků pro efektivní kontrolování projektu. Mezi první instituce využívající tyto metody v širším měřítku patřila například NASA.

V posledních 40 letech se značně zvýšilo využívání počítačových simulací pro projektový management. Dříve totiž programy a počítače, které byly schopny toto zvládnout, stály desítky i statisíce dolarů. Nyní je situace však jiná, a to právě díky tomu, že jsou již používané programy hojně rozšířeny, a proto se ceny pohybují v řádech tisíců dolarů. Programy také jsou více flexibilní ke změnám, tudíž je snazší je přizpůsobovat pro určité podniky a jejich projekty. Zvládají řešit plánování, analyzovat výkonnost a efektivnost, předpovědi, náklady a je jedno, o jak velký projekt se jedná.

Jako tomu bylo i na začátku, tak je tomu i nyní. S druhou světovou válkou byl další velký posun v projektovém managementu, a to právě z toho důvodu, že bylo nutné rychle a efektivně přesouvat velké množství materiálu, jídla a jednotek (vojáků). V roce 1961 firma IBM byla jednou z prvních společností, která definovala role projektových manažerů. Měli zodpovědnost za dohled nad celkovým vývojem projektu a nad jednotlivými funkčními liniemi.

Při diskuzi ve spojitosti s vývojem projektového managementu v roce 1962 byly představeny čtyři různé typy organizování projektového managementu, které se mohou v průběhu mezi těmi čtyřmi typy měnit. Změna probíhala na základě toho, jak se problematika a komplexita v průběhu projektu měnila. Tyto čtyři typy jsou následující (Nicholas a Steyn, 2017):

- „Project expeditors“ - jejich cílem je urychlit práci, jsou komunikačními partnery pro firmu a projekt, účelem je dosáhnout jednotné formy komunikace a předkládat technické koncepty formou byznysového konceptu jako jsou náklady, plánování a marketing.
- „Project coordinators“ - jejich cílem je dosažení jednotné kontroly nad projektovými aktivitami, mohou kontrolovat rozpočet a vyplácet z něj potřebné prostředky.
- „Matrix managers“ - jejich cílem je jednotné řízení, mohou oproti předchozím typům navíc plánovat, řídit a kontrolovat, jak probíhají projektové práce.
- „Pure project managers“ - jejich cílem je dosažení jednoty velení nad lidmi, kteří jim přímo podléhají, především se jedná o integrátory a všeobecné odborníky, musí řešit vyvážení harmonogramů, nákladů, zdrojů a lidské faktory.

1.3 Projekt

Projekt je nejdůležitějším prvkem projektového managementu. Existuje mnoho různých definic, které říkají, co je to vlastně projekt. Definice se liší v určitých podrobnostech, ale ve směr všechny říkají tu stejnou základní myšlenku, kterou je, že projekt má jasně definovaný cíl, účel (tedy důvod existence) a jaké jsou očekávané výsledky nebo výstupy. Jednou z dalších základních charakteristických znaků je dočasnost. Každý projekt vzniká na určitou dočasnou dobu. Tato doba z pravidla vzniká na základě nějakého harmonogramu, kde na jeho konci bude výsledný produkt, tedy splnění zadaného cíle.

Důležité je také, že projekt je jedinečný, což platí z vícero důvodů. Vyžaduje se totiž udělat něco, co tu ještě nebylo, něco nového, odlišného. Jedná se o jednorázovou aktivitu, která se už úplně stejná opakovat nebude. Jedinečnost, nebo také unikátnost, vzniká hlavně kvůli tomu, že je potřeba se přizpůsobovat podmínkám a také specifickým potřebám a cílům, které si firmy stanovily. Z tohoto důvodu má každý projekt specifické vlastnosti a rozsah aplikovaných zdrojů (Svozilová, 2016).

Jedinečnost sebou ale také nese rizika a neznalosti, protože každý projekt je unikát a výsledek je tedy nejistý. Organizace, pod kterou se projekt provádí, má tedy obavy z toho, že selhání může ohrozit nejen výsledek projektu, ale také organizaci samotnou nebo její cíle.

Projekt v jeho životnosti prochází cyklem, jako je tomu například u produktu. V průběhu přecházení mezi jednotlivými fázemi se často mění zdroje, úkoly, členové projektu a také organizace (Nicholas a Steyn, 2017).

Cílem každého projektu je tedy vytvoření nějakého unikátního produktu, služby nebo jejich kombinace, jenž naplní očekávání zadavatele projektu a pomůže se posunout kupředu k dosažení strategického nebo taktického cíle.

Každý projekt má vždy stanoveno, jaký je pro něj dostupný rozpočet a jakým způsobem se bude čerpat, což vychází z plánovaného využívání technologií, materiálu a cenu práce. Tímto je možné také snadněji sledovat kontrolu postupu projektových prací.

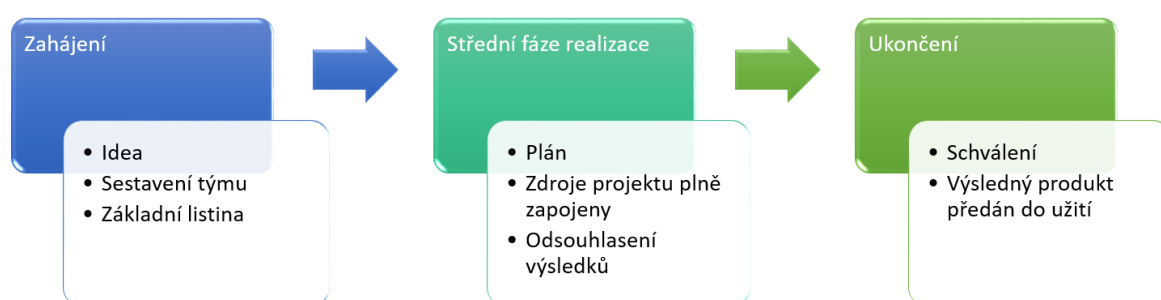
Při formování plánů a realizace projektu je také nutné brát zřetel na okolí projektu, protože jej může pozitivně, ale také negativně ovlivňovat. Mimo jiné se může jednat například o mezinárodní a politickou situaci, což může znamenat třeba jiné počty

dnů dovolené nebo svátků (Svozilová, 2016). Hospodářské a tržní prostředí může také značně ovlivňovat projekt nejen případnou inflací, úrovní platů, ale také konkurenční útoky a vstupy nových technologií na trh. V neposlední řadě se také může jednat o omezení kvůli geografickému nebo geologickému umístění, ochraně životního prostředí nebo například podnebí s čímž souvisí i počasí.

1.3.1 Životní cyklus projektu

Projekt stejně tak jako jiné procesy má životní cyklus. V průběhu se mění a přechází do postupně dalších fází tohoto cyklu. Existuje mnoho teorií a definic, jak takový životní cyklus projektu vypadá. Každá z nich má trochu jiný pohled na věc, ale opět se v základní myšlence shodují. Cílem rozdělení projektu do jednotlivých fází, které na sebe navazují, je usnadnění orientace pro všechny členy v různých stádiích projektu, čímž se může zvýšit šance na jeho úspěch, tedy naplnění cílů. Také to umožní snadnější provádění kontroly jednotlivých procesů (Svozilová 2016).

Jednotlivé fáze projektu definují, co se v daném úseku bude vykonávat za práce, jakým způsobem se bude postupovat, co má být konkrétními výstupy a jak se budou kontrolovat. Zároveň se tím také zlepší přehlednost o tom, kdo by se měl a v jakém okamžiku zapojit do různých aktivit projektu. Na obrázku 1 jsou tyto základní fáze, včetně dílčích konkrétních výstupů, vyobrazeny.



Zdroj: (Svozilová, 2016)

Obr. 1 Zobrazení fází projektu

Přechod mezi fázemi nenastává samovolně, ale je nutné provádět schvalování o, že je možné postoupit do další fáze projektu. O tom, kdy je možné přistoupit k přesunu na další úroveň se rozhoduje dle již dříve definovaného stavu projektu nebo dílčích naplánovaných výsledků, kterých musí být dosaženo. Fáze projektu jsou na sebe navázány jako sekvence, přičemž každá má stanoven časový rámeček.

V prvotních fázích, především při zahájení, se jedná převážně o implementaci základní myšlenky, založení projektu, tvorby projektového týmu. V následujících fázích již začíná postupné zapojování zdrojů nejen ideových, ale už i fyzických a materiálních prostředků. S každou další fází jsou již generované výstupy projektu, které ještě nejsou tím hlavním cílem, ale představují dílčí výsledky výkonu řízení a samozřejmě i produktu.

Při přechodu mezi různými fázemi životního cyklu se mění charakteristiky projektu, jimiž jsou čerpání nákladů, jenž se ve střední fázi dostávají na maximální využití až po ukončení projektu, dále se jedná také o čerpání lidských zdrojů, které začínají už při zahájení projektu a s dokončováním jejich potřeba klesá. S tím, jak se projekt postupně dostává do svého cíle, tedy k dokončení produktu, ztrácí odolnost vůči dodatečným změnám, jelikož už by v pozdější fázi znamenaly nutnost provádět některé činnosti znovu (Svozilová 2016). Oproti problému s dodatečnými změnami klesá riziko neurčitosti, jelikož se postupně odstraňují důvody jejich vzniku.

Životní cyklus projektu se od této definice pochopitelně odchyluje dle potřeb jednotlivých projektů, protože některé složitější projekty jsou definovány ve více sekvencích nebo se jednotlivé fáze rozdělují ještě na nějaké dílčí.

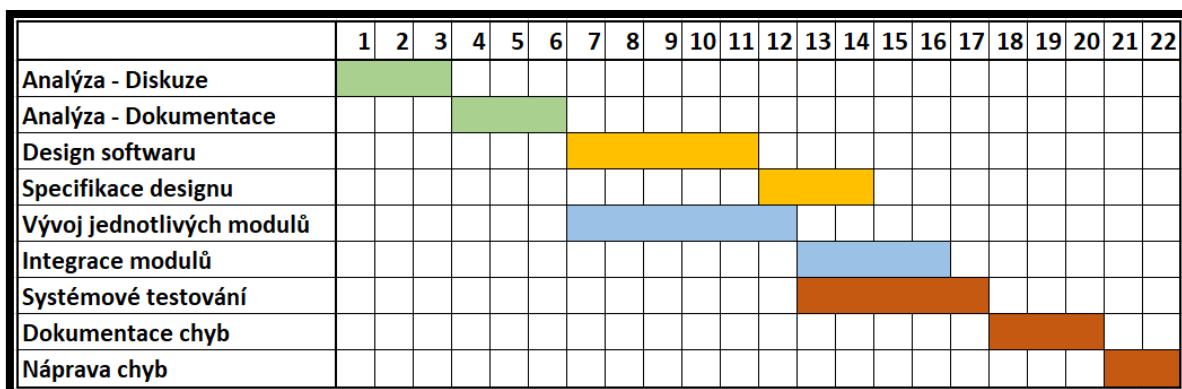
1.3.2 Ganttův diagram

Ganttův diagram je jedním z nejpoužívanějších formátů pro organizaci a plánování projektu. Zjednodušeně řečeno Ganttův diagram je sloupcový graf, který vizuálně zobrazuje úkoly naplánované v čase. Velmi snadno z něj lze vidět od kdy do kdy jaký úkol bude trvat, které úkoly mají rezervy (tedy mohou mít zpoždění) a jaké úkoly jsou těmi, co mohou celý projekt pozdržet.

V Ganttově diagramu tedy lze snadno vidět datum začátku projektu, jaké úkoly v projektu jsou, kdo na jakém úkolu pracuje, kdy začne a skončí jaký úkol (a také jak dlouho každý úkol bude trvat), jakým způsobem se jednotlivé úkoly překrývají a kdy bude projekt dokončen (projectmanager.com, 2021; Projektově.CZ, 2021).

Díky Ganttově diagramu lze snadno určit úzká místa projektu, tedy kritickou cestu a jaké úkoly jsou součástí této kritické cesty. Snadno lze z diagramu také vyčíst, jak je která funkční jednotka vytížená.

Na obrázku 2 je ukázka Ganttova diagramu vytvořeného v Microsoft Excelu, kde jsou vidět čtyři různé funkční jednotky, které mají několik úkolů. Některé funkční jednotky se se svými úkoly mohou překrývat, tedy probíhat současně. Zároveň je také vidět, že projekt bude trvat 22 týdnů, než bude dokončen.



Obr. 2 Ukázka Ganttova diagramu

1.4 Různé formy projektového managementu

Projektový management může mít mnoho forem, které se však dají rozdělit do vyšších základních celků. Všechny typy forem však sdílí dvě věci – projektový tým nebo projektová organizace vznikne za účelem dosažení specifického cíle a každý projekt musí mít svého projektového manažera, který dohlíží na projekt, plnění jeho cílů a zodpovídá za něj.

Nejběžnější forma projektového managementu je označována jako „základní“. Projektový a funkční manažer jsou na stejné úrovni, tudíž se oba zodpovídají stejné osobě na vyšší úrovni. Projektový manažer má na starosti plánování, organizaci, řízení a kontrolování projektu od začátku do konce. Pro dosažení cílů může pracovat s jakoukoliv úrovní a funkční oblastí organizace. Průběžně informuje o stavu projektu generálního ředitele nebo vlastníka. Základní řízení projektu se rozděluje na dvě nejvíce používané formy – čistě jen projekt nebo matice. Když se jedná čistě o projekt, vytvoří se nová organizace a potřebné zdroje patří pouze danému projektu. V případě maticového řešení se projekt vytváří ze zdrojů přidělených, respektive vypůjčených, funkčních jednotek (Nicholas a Steyn, 2017), které se musí sdílet mezi dalšími souběžně běžícími projekty. Tyto formy se často používají ve stavebním a technologickém průmyslu, přesto Barndt Adams a Martin

Adams uvádějí příklady, kde lze také toto základní projektové řízení použít (Nicholas a Steyn, 2017):

- Zdravotnictví, vzdělávání a sociální péče – práce se vykonává na základě přidělovaných grantů státních a místních agentur a s každým grantem jsou spojeny rozdílné požadavky na čas, výkon a náklady. Mohlo by se tedy říci, že se s každým grantem jedná o nový projekt, na který lze aplikovat koncept projektového řízení.
- Propagační kampaně – reklamní společnosti běžně provádějí propagační kampaně, kde se využívá podpora účetnictví, grafiky, marketingového výzkumu a dalšího. Probíhá několik projektů (kampaní) současně v různých životních cyklech.
- Rozvoj vzdělání – lze považovat za projektovou činnost, protože značná část práce je financována z grantů s určitými cíli a nákladovými nebo časovými omezeními. Potřebná je také koordinace jednotlivých pedagogů a výzkumných pracovníků.

Programové řízení se může považovat za nadřazenou verzi předcházejícímu základnímu typu. V tomto případě se totiž jedná o několik projektů, které jsou součástí většího celku, tedy programu. Program je definován stejně jako projekt v případě stanovování cílů a čeho má být dosaženo, vyžadují se plány, rozpočty a jakým způsobem se splní zadaný cíl. Hlavním rozdílem je časový horizont. Program se skládá z několika paralelních nebo na sebe navazujících projektů, které jsou řízeny tak, aby byl splněn programový cíl. Jednotlivé projekty, které jsou součástí programu, jsou na sobě vzájemně závislé, sdílejí společné cíle a zdroje. Například se může jednat o program určitého produktu, kdy jedním projektem bude samotný produkt, který se má vyvinout. Za provoz a servis výsledného vyvinutého výrobku už je zodpovědný někdo jiný. V případě programu se zodpovědnost přesouvá na správu programu, která má zajistit integraci do jiných systémů a udržovat jej funkční tak dlouho, jak to bude nutné. Správa programu dohlíží na všechno, aby bylo dosaženo jeho cíle. Programový manažer dohlíží a koordinuje jednotlivé projekty, které jsou součástí programu. Je nezbytné mít vytvořenou programovou strukturu pro jejich koordinaci. Důležitým poznatkem je obvyklá

změna programového manažera v průběhu trvání programu, jelikož jeho cyklus může být velmi dlouhý.

Dalším typem formy projektového managementu je nové řízení rizik. Jedná se o druh řízení používaný ve spotřebitelských firmách pro zajišťování nových produktů nebo trhů. Vytvoří se tým, který vyhledává nové produkty nebo trhy odpovídající schopnostem a prostředkům organizace. Poté, co se produkt definuje, pokračuje se na další fázi, tedy k návrhu, vývoji, jakým způsobem bude probíhat výroba, prodej a distribuce. Řízení rizik má pouze dočasný charakter stejně jako typický projekt. Jakmile nový rizikový tým dokončí svůj úkol, členové se vrátí zpět do svých původních oddělení nebo do jiných rizikových skupin, případně se přesunou do nové divize nebo společnosti, která vznikne za účelem výroby nového produktu (Nicholas a Steyn, 2017).

Posledním typem je produktový management, který označuje osobu zodpovědnou za dohled nad všemi aspekty plánování výroby, distribuce, zásob a prodeje produktu. Produktový manažer koordinuje a urychluje celý proces související s výrobou a distribucí produktu. Komunikuje stejně tak jako projektový manažer s funkcemi uvnitř i vně organizace (Svozilová, 2016). Produktový manažer má na starosti plynulý a bezproblémový průběh, což znamená aktivní řešení konfliktů a problémů, které mohou produkt ohrozit.

1.4.1 Výhody a nevýhody projektového managementu

Projektový management může přinášet mnoho potenciálních výhod. Mezi ně patří hlavně plánování, jelikož celý projekt musí být již na začátku naplánován až do samého konce. Každá aktivita má přiřazenou roli a odpovědnost a není tudíž závislá na změnách realizačního personálu. Pro realizaci je jasně stanoven nákladový a časový rámec, v průběhu realizace projektu je celkem snadné sledovat plnění plánu a provádět případné korekce v případě definování odchylek. Díky určení odpovědností a rolí v projektu není nutný pravidelný nadměrný dohled ze strany sponzora/zákazníka projektu, protože jsou jasně stanoveny postupy při provádění eskalace nějakého problému. Při překročení plánovaného cíle je nutný souhlas, není možné bez něj pokračovat, tudíž se nic neděje bez vědomí zadavatele/zákazníka projektu. Další projekty mohou využívat informace získané

při realizace daného projektu, aby tak bylo umožněno efektivnější a méně rizikovější plnění cílů.

Samozřejmě projektový management má i své nevýhody, jelikož se může objevit mnoho nepředvídatelných situací, které musí projektový manažer řešit a na takové situace by měl být připraven. V takových situacích se ukáže, jak je manažer schopný, talentovaný a zkušený. Nejčastější komplikací může být samotný zákazník projektu, často se jeho specifické požadavky projevují až v průběhu samotné realizace, což může být velmi nákladné (Svozilová, 2016). Při plánování a oceňování nákladů projektu nelze předvídat všechny vlivy, které mohou nastat a jak se budou měnit ceny různých zdrojů používaných při dosahování cílů projektu.

1.5 Organizační struktura projektu

Každý projekt musí mít vytvořenou organizační strukturu i když se jedná jen o určitý časový úsek, dokud se nenaplní stanovené cíle. Důvodem je závislost na lidech, a to nejen na jejich individuálním výkonu, ale na výkonu projektového týmu jako celku. Velmi důležité je stanovit strukturu rolí, jaké vztahy budou mezi jednotlivými rolemi a jaké odpovědnosti budou mít nejen za splnění dílčích úkolů, ale také za komunikaci mezi jednotlivými členy projektu a plnění celkového cíle projektu.

Organizační struktura je velmi důležitým prostředím, protože největší množství interakcí probíhá právě mezi jednotlivými členy projektu, kdy se řeší koordinace, monitorování, řízení projektových prací, kontroly a další projektové komunikace využívané pro dostatečné předávání informací. Každá struktura musí splňovat jisté faktory, aby fungovala vhodně a správně určovala postavení jednotlivých členů projektu. Jedná se totiž o velmi dynamickou strukturu, a proto musí mít jasně stanovené principy řízení, odpovědností a rozdělení autorit (Svozilová 2016).

Jak již bylo řečeno, každý projekt je jedinečný, a proto také nelze vždy přesně předvídat všechny situace, které mohou nastat a je nutné se dopracovat k optimálnímu řešení v případě nějaké neočekávané situace. Pro úspěšné řízení projektu je nutné rozložit zájmy, rozhodovací schopnosti a autority rovnoměrně. To je mapováno popisem zájmových skupin projektu, tedy těch, kteří jsou účastníky nebo součástí projektového dění a mají své individuální nebo kolektivní cíle, jichž chtějí dosáhnout. Když se zajistí soulad těchto zájmů s těmi projektovými, může se tím podpořit celková úspěšnost projektu. Vedle zájmu skupin projektu se také řeší

právě zmíněná organizační struktura, která představuje definované vztahy, mezi kterými probíhá komunikace a formální rozložení odpovědností a autority projektu. Jedná se o neustálé vyjednávání mezi zájmovými stranami a subjekty.

Autorita je důležitým prvkem řízení, jelikož s jejím užitím jsou koordinovány dílčí úseky výkonu práce a při sledování výsledků kontrolních měření, jak se daří postupovat v plnění cílů projektu, se rozhoduje o náležitých korektivních opatřeních. Jednotlivým účastníkům jsou přiřazeny role a odpovědnosti podle pracovních povinností, které korespondují s jejich důvodem pro obsazení v projektu. Nejvyšší autoritu by měl mít samozřejmě manažer projektu, protože se jedná o centrální bod všech účastníků projektu a také zájmových skupin, které se nějakým způsobem podílejí na projektu. Má na starosti, kromě koordinace aktivit členů a standardního řízení projektu, také řešení neočekávaných situací, které nebylo možné naplánovat a očekávat. Pokud by neměl manažer projektu dostatečnou autoritu, může to vést k problémům naplnění cílů projektu, protože nebude mít patřičný rozhodovací vliv na všechny zájmové skupiny a bude tak obtížné v nich udržet celistvou linii. V případě, že bude nejasně určená autorita, může vzniknout rivalita o to, kdo bude mít rozhodovací vliv, což může vést k časovému skluzu a v důsledku toho k překročení stanoveného rozpočtu projektu. To může také následně vést k problémům s kvalitou a řízením rizik projektu, jelikož aktivity jinak využívané pro řízení projektu budou řešit problémy s autoritou (Svozilová, 2016).

Problém s autoritou se v praxi stává často a nastává z důvodu delegování až příliš mnoho autority a odpovědnosti, čímž se prosazení názoru manažera snižuje. Tento problém se však dá řešit pomocí přidělování odměn nebo udělování trestů, což může být ku příkladu osobní ohodnocení, bonus k výplatě nebo přesun do jiného projektu (atraktivnějšího nebo horšího). Další metodou může být možnost povýšení, čímž se dostává k možnosti nabýt nové znalosti a zkušenosti, které pak může sdílet s ostatními členy týmu.

1.5.1 Zájmové skupiny a jejich vztahy

Zájmové skupiny jsou všechny interní nebo externí účastníci se vztahem k projektu. Identifikace těchto skupin je důležitým úkolem při přípravě a plánování projektu. Jednotlivé skupiny mají různou úroveň rozhodovací autority a odpovědnosti.

Zákazníkem projektu je někdo mající zájem na realizaci a může být jeho zadavatelem nebo investorem. V případě zadavatele se jedná o někoho, kdo bude budoucím uživatelem výstupu projektu. Investora zajímá realizace projektu kvůli potenciálnímu zvýšení úspěšnosti na trhu, kde působí, pomocí nové služby nebo produktu, jenž bude výstupem projektu.

Dodavatelem nebo realizátorem je společnost nebo její část, jež se přímo účastní kontraktu, ze kterého plyne odpovědnost za realizaci projektu. Jejím zájmem je dodržení podmínek kontraktu, za což získá svou odměnu. Dodavatelem projektu se může stát externí společnost, jiná jednotka přímo od zadavatele nebo organizační jednotka, které je zároveň zadavatelem (Svozilová, 2016).

Rozklíčovat všechny zájmové skupiny projektu je důležité pro úspěšné řízení projektu. Spektrum těchto skupin může být široké, tudíž ať už se jedná o osobu, která občas pomůže finančně nebo politicky nebo ti, co dokážou ovlivnit sponzorování projektu. Všichni z jakékoliv zájmové skupiny mohou jistým způsobem pozitivně, ale také negativně ovlivnit průběh nebo celkové dosažení cíle projektu. Právě z těchto důvodů je potřebné identifikovat všechny zájmové skupiny, zjistit jejich autority a zodpovědnosti, rozpoznat jejich očekávání a požadavky, definovat jaké jsou jejich individuální cíle a rizika s tím spojená.

1.6 Tradiční způsob vedení projektu

Výše zmíněné informace o projektovém managementu se v posledních letech v některých publikacích označují za tradiční, protože již existují dlouho a používají se již pravidelně. Ve většině odvětvích, kde jsou jasně definovány cíle už na začátku, je možné úspěšně využít tradiční způsob (Layton, 2017; KnowledgeHut.com, 2021).

Mezi typické charakteristiky pro tradiční způsob vedení projektu patří především implementace životního cyklu projektu a jeho fáze mají jasně stanovené úkoly a výsledky. Jakmile se těchto cílů fáze dosáhne, přechází se po schválení do další, a prakticky vše z předchozí fáze (nebo fází) se již považuje za uzavřené a pevně dané. Tradiční přístupy kladou velký důraz na dodržování předem stanoveného procesu a předpokládají stabilitu, čímž je myšleno, že prostředí, požadavky, návrhy a analýza zůstanou stále stejné od začátku projektu až po jeho dokončení.

V návaznosti na stabilitu se tradiční způsob zaměřuje na předvídatelnost projektu. Pokud se vyvine dostatečné úsilí pro přesné předvídání rizik a výsledků projektu, bude možné udržet projekt pod kontrolou, tedy v souladu s procesem. Kvůli předvídatelnosti se také řeší rozdělení úkolů. Členění úkolů umožní prohlubování detailů a tím také provádět přesnější předpovědi (Depaire, 2019).

Metodologie PMBoK

Metodologie PMBoK (Project Management Body of Knowledge) je kolekce procesů, terminologií, pokynů a osvědčených postupů, které se považují za standardy. Umožňuje společností snáz standardizovat postupy napříč odděleními, adaptovat procesy konkrétním potřebám a zamezovat selhání projektu.

Projekt využívající metodologii PMBoK má popisný charakter, tudíž neurčuje, jak má člověk řídit projekt, ale představuje sadu procesů, jak projekt správně řídit. Každý z procesů musí být přizpůsoben pro konkrétní projekt, jedná se například o to, kdo má provádět jaké procesy a jak důsledně.

Žádné postupy a procesy nezůstávají stejné nebo se přidávají nějaké další, a proto existuje institut Project Management Institute (dále PMI), který dohlíží na to, aby se nově nalezené znalosti, které odborníci objevují a tím aktualizují a rozšiřují metodologii PMBoK, dostávali mezi širokou veřejnost (wrike.com, 2021; Depaire, 2019).

Metoda PRINCE2

Metoda PRINCE2 (Projects in Controlled Environments) se používá po celém světě. Jedná se o procesní metodu pro efektivní vedení projektů a poskytuje základní dovednosti o tom, jak se stát úspěšným manažerem (prince2.com, 2021).

Metoda PRINCE2 je založena na procesech, konkrétně jich definuje 7 a skládají se ze strukturované sady aktivit, které slouží pro dosažení konkrétních cílů. Procesy, které představuje projekt řízený metodou PRINCE2, jsou definovány na vyšší úrovni než metodologie PMBoK. To, co definuje, jsou aktivity prováděné dle metody PRINCE2.

Procesy se velmi podobají fázím životního cyklu projektu a předpokládají, že projekt je rozčleněn (dle principů tradičního způsobu vedení projektu) do jednotlivých samostatných fází (Depaire, 2019).

K metodě PRINCE2 nepatří jen 7 definovaných procesů, ale také 7 principů. Tyto principy nelze přizpůsobovat, jsou pevně určeny a správně by se měly promítnout do celé implementace samotného řízení projektu. Jedním z principů je zdůvodnění pokračování projektu. Měl by existovat oprávněný důvod k tomu, zda projekt je ještě životaschopný a má tedy význam, aby byl projekt stále provozován a spravován, v opačném případě by mohlo dojít k předčasnému ukončení. Učit se ze zkušeností, to je princip, který nabádá ke hledání a čerpání poznatků z předchozí práce, poučit se z chyb a využít je pro zlepšení současně běžících projektů.

Správné definování rolí a odpovědností je velmi důležité. Stanovit si jasnou organizační strukturu, kde bude jednoznačné, kdo spadá, pod kterou osobu a do kterých úkolů se má kdo zapojit.

Správoání projektu by mělo probíhat po etapách, stejně jako jsou jednotlivé fáze. S každou novou etapou by měl zahrnovat aktualizaci rizik, celkového plánu, obchodního případu a podrobné plánování další fáze s přihlédnutím na nové poznatky.

Každý člen projektového týmu by měl mít nastavenou takovou autoritu a odpovědnost, aby byla jeho práce co nejvíce efektivní. Při překročení těchto pravomocí se musí postoupit na další úroveň, aby bylo možné rozhodnout, jak postupovat.

Projekty řídicí se metodou PRINCE2 se zaměřují na definici produktu, jeho dodání a požadavky na kvalitu. V neposlední řadě se také snaží o přizpůsobení prostředí projektu a to jeho složitostí, důležitostmi, riziku, schopnosti a také především velikosti (Depaire, 2019; Axelos.com, 2021).

1.7 Agilní způsob vedení projektu

Pro často se měnící požadavky nebo při očekávání velkých změn nemusí být tradiční způsob vedení vhodný a je vhodnější využít spíše moderní přístup, kterým je právě agilní způsob.

V dnešní zrychlené době je velmi vysoký počet neznámých a nejistot, které neoddělitelně patří ke každému projektu. Většina projektových manažerů by se shodla na tom, že je prakticky nemožné při plánování projektu definovat všechny požadavky. Po zahájení projektu a odstartování vývoje produktu si zákazník začne

uvědomovat, že jeho potřeby se změnily a svůj původní požadavek chce změnit. Náklady však mohou výrazně vzrůst právě kvůli takovýmto nedbalým rozhodnutím, z čehož plyne překročení plánovaných nákladů a nedodržení původního stanoveného data dokončení. Právě z těchto důvodů je potřebné používat takový přístup k řízení projektu, který dokáže na tyto neustálé změny reagovat vhodněji nebo který zahrnuje učení a objevování během celého životního cyklu projektu a takto právě funguje agilní způsob vedení projektu (KnowledgeHut.com, 2021; White, 2008).

Agilní přístup je založen na empirické kontrolní metodě, tedy procesu rozhodování na základě pozorovaných realit v projektu. Vhodný může být i v případě projektů zaměřujících se na zdokonalení nebo zlepšení nějakého produktu, např. softwaru. Pro úspěšné používání empirické kontrolní metody je důležité dodržovat tři základní pravidla. Každý, kdo se účastní agilního projektu, musí vědět, co se právě děje a jak projekt postupuje – tedy transparentnost projektu. Dodržovat časté kontroly, kdy se bude hodnotit produkt a proces lidmi, kteří do daného produktu nebo procesu investují. Třetím pravidlem je okamžité přizpůsobení, čímž je myšleno rychlé provádění úprav, aby bylo možné minimalizovat problémy, a s tím také souvisí nutnost provádět okamžité změny, pokud se při kontrole ukáže jejich nutnost.

Pro umožnění dodržování těchto tří základních pravidel, je nutné změnit způsob, jak projekty fungují. Agilní způsob funguje na tzv. iteracích, což se může zjednodušeně popsat jako menší segmenty celkového projektu. Stejně jako je tomu v tradičním způsobu vedení projektu, tak i zde se zahrnuje stejný typ práce do formy vodopádu, což je obdobné k životnímu cyklu projektu. Nejprve se vytvářejí požadavky a plány, poté probíhá vývoj produktu, dokumentuje se a pokud je to nutné, integruje se produkt s jinými produkty. Po otestování produktu se opraví zjištěné problémy a uvede se do používání. V případě agilního způsobu se namísto čekání dokončení těchto kroků najednou, projekt rozdělí na iterace, nebo také tzv. sprinty (Layton, 2017).

Již za běhu projektu se provádí zlepšování, využívá se Demingův cyklus (kanbanize.com, 2021), metoda PDCA (Plan, Do, Check, Act), která v češtině znamená plánuj, dělej, kontroluj a jednej. Jedná se o jeden z nejdůležitějších atributů agilního způsobu vedení projektu. Tým se pravidelně zapojuje do častých cyklů učení společně s vývojem projektu, což nahrazuje pouze jedno velké

„ponaučení“ na jeho konci. Tím se zajišťuje, že dojde k podstatným zlepšením procesu, zatímco projekt stále pokračuje, což pozitivně přispívá k úspěšnému dodání finálního řešení zákazníkům. Jelikož je práce rozdělena do několika malých výstupů, které jsou průběžně předávány zákazníkům za účelem jejich přezkoumání a získání zpětné vazby, je možné takto přispívat k neustálému zdokonalování produktu nebo služby s ohledem na výsledek, aby byl produkt vhodný pro zákazníka (kanbanize.com, 2021).

V průběhu vývoje agilního způsobu vedení projektu se neustále přidávaly nějaké principy a pravidla. Postupem času se ze všech vytríbily tři nejdůležitější principy, které vznikly na základě zkušeností od týmů přecházejících na agilní způsob. Zkušenosti se získávaly přímo v terénu v různě velkých organizacích po celém světě a takto vznikly Platinové principy (Layton, 2017).

Platinové principy

Prvním principem je odolávání formálnosti. Tento princip poukazuje na problém, který vzniká kvůli dodržování určitých zbytečných standardů, jako může být krásně vypracovaná prezentace, na které musí někdo trávit několik hodin, přitom z informativního hlediska by byla dokončena během chvíle. Dalším příkladem jsou třeba hodinové schůzky, kde se řeší leckdy jednoduché problémy, které je možné vyřešit i jinak nebo rychleji v pár větách, než svolávat celý projektový tým a tím zdržovat několik desítek lidí delší dobu. Celkově je apel na udržování nejnižší úrovně kvantity, složitosti dokumentace a četnosti schůzek, čímž je možné ušetřit čas a dělat místo toho něco ku prospěchu dosažení cíle projektu.

Dalším principem je myslet a jednat jako tým. Všichni členové týmu by měli být zaměřeni na to, jak udělat tým produktivnější, čímž se může docílit vyrovnanosti výkonů jednotlivých členů týmu. Všichni by měli být sladění ve svém závazku k cíli, odpovědnosti za rozsah práce a v uznání času, který je potřebný pro dosažení cíle. Podporovat rozvoj ve dvojicích s častou obměnou partnerů, stínování, čímž je myšlena zastupitelnost, aby nebyl pouze jeden člověk, co se v dané problematice vyzná. Nahrazením jednotlivých pracovních titulů za nějaký jednotný, aby se nikdo necítil méněcenný nebo nadřazený vůči ostatním. Mezi vývojové aktivity se zahrnují všechny úkoly potřebné ke splnění požadavků, tedy zakomponovat do seznamu nejen hlavní, jasně dané úkoly, ale také ty, které vyvstanou při jejich plnění,

například v případě psaní kódu je nutné brát v potaz také rozpad úkolu na programování, testování a tvorbu dokumentace. V neposlední řadě je také vhodné místo zkoumání produktivity jednotlivých členů týmu zkoumat spíše produktivitu týmu jako celku.

Posledním principem je upřednostňování vizualizace před psáním. Používáním různých typů vizualizace, jako jsou například různé diagramy, počítačové modelování, nebo různé obrázky, tak dokážou mít vyšší vypovídací schopnost, než nějaký text. Říká se, že obrázek má hodnotu tisíce slov. Schopnost definovat nějaké vlastnosti systému se může několika násobně zvýšit, když se zvyšuje interakce s navrhovaným řešením (Layton, 2017). Příkladem může být grafické znázornění výsledného produktu než strohý popis, jak by takový produkt mohl vypadat. Pokud se produkt bude ukazovat v nějaké zmenšené fyzické podobě, tak si každý dokáže lépe představit, co by na tom bylo nutné změnit.

Metoda SCRUM

Mezi nejpoblárnější metodu patří SCRUM. Jedná se o časově ohraničené intervaly a rozdělují se tak projekty na pevná období, tzv. sprinty. Hlavním účelem je podpora týmů při produktivním a kreativním doručování produktů s co nejvyšší možnou hodnotou. Každý sprint trvá zhruba jeden až tři týdny, kdy týmy pracují na úkolech stanovených na schůzce, kde se daný sprint plánoval. Princip je takový, že sprinty se opakují do té doby, než bude finální produkt připraven. Když je sprint u konce, zkontroluje se produkt, zjistí se, co funguje a nefunguje, provedou se potřebné úpravy a zahájí se další sprint za účelem vylepšení produktu nebo služby (Layton, 2017).

Součástí metody SCRUM jsou také tzv. Stand-up schůzky. Jedná se o krátké desetiminutové schůzky, které se také mohou označovat jako denní SCRUM schůzky. Účelem těchto schůzek je ujistění se, že každý je na správné cestě a je informován o všem. Název Stand-up vznikl z toho důvodu, že jeden z členů týmu si „stoupne“ a mluví v řádech několika sekund, maximálně minuty o tom, co daný den dělal, co bude dělat nadcházející dny a seznam věcí, které čekají na dokončení jiného člena týmu. Ostatní členi týmu pouze poslouchají. Jakmile se prostřídají všichni zúčastnění dané schůzky, je schůzka u konce (workfront.com, 2021; kanbanize.com, 2021).

Metoda Kanban

Další metoda, která se spojuje s agilními metodami, je kanban, někdy také nazývaná jako SCRUMBAN. Kanban je součástí štíhlé výroby pocházející z Toyoty Production System. Jedná se o systém určený k eliminaci plýtvání a zlepšení toku a propustnosti systému. Metodu kanban je možné použít v jakékoliv fázi projektu, protože je snadné jej aplikovat na již probíhající procesy. Největší výhodou je vizualizace, která byla již zmíněna jako jedna ze tří platinových principů. Vizualizace je prvním krokem k identifikaci potenciálního plýtvání. Už v průběhu vizualizace toku práce, někde na tabuli, je možné určit, kde se produktivita začíná vytrácet a lze snadno analyzovat hlavní příčinu a zjistit, jak odstranit daný problém. I po nalezení problému je potřeba toto provádět opakovaně, protože při odstranění jednoho, se může objevit jiné úzké místo, které může celý projekt brzdit a snižovat celkovou produktivitu projektu. Kanbanová tabule většinou visí, ať už fyzicky nebo virtuálně, někde na viditelném místě pro všechny, aby ji mohl každý vidět. Jasně ukazuje, které předměty nebo úkoly je nutné zpracovat jako další. Každý úkol lze přesunout do kategorie dle toho, zda se na ní pracuje, je dokončená, nebo je prozatím nepodstatná. Na obrázku 3 je ukázka, jak taková zjednodušená tabule může vypadat. V prvním sloupci jsou úkoly, které je potřeba zpracovat, v prostředním jsou úkoly, které si již někdo přiřadil a zpracovává je. V posledním sloupci jsou dokončené úkoly.

Seznam úkolů	V procesu zpracování	Dokončeno
Opravit brzdy	Kalibrace ESP	Vyměnit čelní okno
Vyměnit olej v motoru		
Kalibrace couvací kamery		

Obr. 3 Ukázka kanbanové tabule

Ke každému úkolu se také může přiřadit určitý pracovník, čímž dá najevo ostatním, že na daném úkolu již pracuje a ostatní si tak mohou vybrat „volné“ úkoly. Díky této tabuli je také snadné zjistit pokrok a řízení toku práce, lze také snadno identifikovat, kdo na jakých úkolech pracuje a kolik jich již dokončil, čímž je možné také zjistit efektivitu daného pracovníka.

Omezování nedokončené práce je jedním z východisek používání kanbanové tabule. Tým by se měl soustředit na to, aby rozpracované úkoly byly dokončeny dříve, než se někdo pustí na další. Práce na několika činnostech zároveň neznamena, že budou dokončeny rychleji, ale naopak. Soustředěním se nejdříve na nedokončení úkoly umožní dokončovat úkoly rychleji, což zrychlí i tempo zpracování položek čekajících ve frontě na dokončení.

Korektní správa toku může také znatelně urychlit zpracovávání úkolů. Příkladem je třeba dopravní zácpa, která může vzniknout kvůli nedostatku jízdnic pruhů, což způsobí, že všechny automobily musí jet pomaleji a tím pádem je to negativní pro všechny účastníky. Pro vyřešení je nutné nějakým způsobem regulovat vjezd vozidel do provozu nebo zvýšit počet jízdnic pruhů v místech, kde se tvoří největší dopravní zácpy. Stejně tak jako v případě dopravní zácpy mohou dopadnout úkoly, když se budou pokoušet dělat je všechny najednou. Práce na jednom úkolu současně, identifikaci a odstranění omezení zvyšuje tok všech položek v systému. Měření doby zpracovávání lze sledovat správu toku. Tým určuje dobu realizace sledování doby, po kterou přejde požadavek ze stavu ke zpracování, přes zpracovávání až po dokončení. Díky tomu tým zná dobu cyklu sledování času od začátku práce do jejího zpracování. Agilní tým pak může optimalizovat tok identifikováním úzkých míst, které brání zkrácení doby vypracování úkolů (Layton, 2017).

1.7.1 Výhody a nevýhody agilního způsobu oproti tradičnímu způsobu

Mezi výhody patří rychlé nasazení řešení, větší flexibilita, efektivnější využívání zdrojů, rychlejší detekce problémů a přizpůsobivost měnícím se potřebám. To umožní lepší spolupráci se zákazníky, aby produkt vyhovoval jejich potřebám.

Nevýhodou agilního způsobu je to, že není vhodný pro všechny projekty. Není vhodný pro organizace, které mají tendenci analyzovat problémy po delší dobu před přijetím opatření, nebo pro ty, které přijímají rozhodnutí vyšší úrovně řízení. Existují i další potenciální nevýhody, mezi které patří nedostatek dokumentace, což je právě způsobeno pokusem o co největší množství vizualizace, vynechávání popisů a také jsou méně předvídatelné výsledky projektu (Torode, 2020).

1.8 Porovnání tradičního a agilního způsobu vedení projektu

Výše bylo popsáno, jak funguje tradiční a agilní způsob vedení projektu. Každý způsob má své výhody a nevýhody. Agilní způsob není vhodné používat u projektů, které musí být jednoznačně dokončovány na určité fáze nebo pokud zákazníka nezajímá průběh projektu, ale zajímá se jen o konečný výsledek. Tradiční způsob zase není vhodné využívat u projektů, které jsou složité a mohou podléhat častým změnám v průběhu jeho vývoje, protože není postaven na tom, aby byl flexibilní (vivifyscrum.com, 2019). Lze si i pro porovnání vzít jednotlivé vlastnosti a dle toho si určit, který bude pro určitý projekt vhodnější.

Organizační struktura

Agilní přístup má iterativní organizační strukturu, to znamená, že se využívají opakující se sprinty, o kterých bylo psáno v předchozí kapitole. Tradiční přístup však sází na lineární strukturu, ve které se musí kompletně dokončit jedna fáze a pak se přechází na další. Při přechodu do další fáze již není možné se vracet k tomu, co se uzavřelo. U sprintů však se právě vrací k tomu, co nebylo dokonalé a zlepšuje se to, dokud to nebude vyhovovat všem (Carr, 2020; Kashyap, 2021).

Rozsah projektu a vývojový model

Agilní způsob se nejvíce hodí pro malé nebo středně velké projekty, které mohou být i velmi složité. Mezi důvody mohou patřit například denní Stand-up schůzky, kde každý má svou minutu. Když se bude jednat o velké projekty, s velkým počtem lidí, tak tyto Stand-up schůzky budou příliš dlouhé a budou neefektivní. Pro velké projekty je vhodný tradiční způsob.

S rozsahem projektu také souvisí vývojový model. Tradiční přístup se řídí obdobou modelu životního cyklu, který je fixní, kdežto agilní se vydal cestou evoluční dodávky, a to opět z důvodů sprintů, kdy výsledkem jednotlivých sprintů je produkt, který se hodnotí a postupně vylepšuje a upravuje dle požadavků, je tedy možné jej snadno upravit (Carr, 2020; Kashyap, 2021).

Zapojení zákazníků a náklady na změny

Agilita umožňuje vysoké zapojení zákazníků, tedy výsledných uživatelů produktu. S ohledem na časté iterace a častým měněním požadavků zákazníka v průběhu probíhajícího projektu, dokáže agilní přístup vhodně reagovat, protože po každém sprintu se provádí hodnocení, analýza a zpětná vazba, čímž je možné upravovat produkt dle požadavků zákazníka a náklady na změny nebudou nikterak vysoké, protože změny se mohou provádět na pravidelné bázi. V případě tradičního přístupu zákazník před zahájením projektu definuje, co má být výstupem a případné úpravy jsou složité a velmi nákladné, protože projekt postupuje mezi fázemi a jen mezi nimi je možné provádět změny (Carr, 2020; Kashyap, 2021; vivifyscrum.com, 2019).

Preference modelu a řízení eskalace

Agilní týmy pracují jako jedna skupina, tedy v případě nějakého problému se to řeší v daném týmu a snaží se jej společně vyřešit. Tradiční týmy musí problém eskalovat na manažery, kteří pak dané problémy musí řešit, případně je eskalovat na ještě vyšší místa. Tyto eskalace mohou celý proces značně zdržet.

V souvislosti s eskalací se také může mluvit o preferovaném modelu, tedy ochoty se přizpůsobovat. Tradiční model se řídí očekáváním, co by mohlo mít úspěch, kdežto agilní model upřednostňuje adaptaci. Na začátku se určí nějaký postup, jaký by mohl být výstup, ale v průběhu se mění dle potřeb (Carr, 2020; Kashyap, 2021).

Tvorba dokumentace a zaměření se na produkt nebo proces

Agilní způsob se snaží formalitu snižovat na minimum, protože není nutné mít dokonalé prezentace a dokumentace, ale vědět, zda se to bude zamlouvat a k tomu stačí obyčejná heslovitá prezentace s obrázky nebo praktickými ukázkami. Nevýhodou tohoto přístupu je nedostatečná dokumentace procesů a postupů, jak se dospělo k určitým výsledkům. Po každém sprintu jsou plánovány testy.

Tradiční způsob se zajímá hlavně o procesy, aby vše probíhalo, jak má, všechno bylo patřičně zdokumentováno. Dokumentace bývá komplexní a jednoznačná. Plánují se pravidelné komplexní testy po každé fázi (Carr, 2020; Kashyap, 2021).

Přezkoumání a schvalování

Agilní přístup provádí kontroly po každé iteraci, tedy každém sprintu a schvalování probíhá na schůzce, která se koná vždy po konci jednoho sprintu. V případě tradičního přístupu se provádí nadměrné kontroly v průběhu každé fáze a schvalování na konci fáze (Carr, 2020; Kashyap, 2021).

Proč je agilní přístup preferovanější

V posledních letech se agilní přístup implementuje častěji, což má několik důvodů. Jedním z nich je více flexibility, což již bylo zmíněno. Při nutnosti v průběhu něco změnit, není to vůbec problém.

Dalším důvodem je transparentnost, jelikož v agilním přístupu se všichni podílejí na všech částech projektu a umožňuje jim sledovat pokrok od začátku až do konce, což hraje významnou roli při vytváření zdravého pracovního prostředí.

Zpětná vazba se může vhodnějším způsobem přijmout a zpracovat, protože se provádí na konci každé iterace a díky tomu je možné dodat kvalitní produkt nebo službu v dodací lhůtě.

Mezi další důvody patří nižší náklady, protože se zákazník může zapojovat do změn již v průběhu, a rychlejší doba uvedení na trh, což je v dnešní rychle se vyvíjející společnosti důležité. Umožňuje také větší týmovou spolupráci a jejich zapojení do vývoje (Carr, 2020; Kashyap, 2021).

1.9 Vliv pandemických situací na projektové řízení

PMI prováděla průzkum, jaký dopad má pandemická situace na podniky respondentů a jak se jim povedlo v takové situaci adaptovat. Průzkumu se zúčastnilo celkem 251 respondentů. 60 % respondentů bylo z Evropy (většina z Německa) a členů týmu bylo mezi 6 až 25. Rozsah projektů byl v rozmezí 100 000 až 10 mil. USD. Pouze 20 % respondentů uvedlo, že pandemická situace je neovlivňuje.

Dopad byl pro podniky velmi různorodý. Pro některé se jedná o masivní narušení, zjistili, jakou mají zranitelnost podnikání, boj o zákazníky se zřetelně zvýšil anebo čelili nedostatku zásob.

Průzkumem bylo zjištěno, že výkonnost projektu je méně ovlivněna než obchodní stránka. Ve velké míře se jednalo o společnosti z IT průmyslu a aktivně začali budovat infrastrukturu pro umožnění vzdálené práce a současně udržovat zabezpečení a soukromí. Tento krok v digitalizaci jejich služeb jim umožnil provozovat všechny, nebo alespoň většinu služeb i nadále.

Podniky zjistily, že práce na dálku nemusí vždy znamenat snížení efektivity, ale může mít i pozitivní dopad, protože zaměstnancům nevznikají cestovní náklady při cestě do práce a z práce, digitální prostředí umožňuje mít kancelář prakticky kdekoliv. V úspoře nákladů došlo také v případě služebních cest, protože se toho více řeší pouze virtuálně. Mezi negativní dopady patří ale také problém s tím, že zákazník očekává vyšší dostupnost, jelikož nelze snadno oddělit pracovní a mimopracovní dobu. Dalším problémem mohou být také děti, o které je nutné se starat, když mají školu zavřenou, čímž se produktivita může znatelně snížit, jelikož není možné se soustředit soustavně na práci (pmi.org, 2020).

PMI se snaží vypomoci a připravuje digitální produkty, které by měly pomoci. Některé z nich budou ze začátku zdarma, ale poté budou zpoplatněny. Podnikům to tak umožní vyzkoušet si, zda jim bude daný produkt vyhovovat a případně si jej poté budou platit. Bude se jednat o produkty umožňující snazší spojení se členy týmu a různá zábavná cvičení projektového řízení.

2 Visual Basic for Application

Microsoft Office aplikace od společnosti Microsoft zná jistě každý a někdy je již používal. Mezi ty nejpoužívanější patří především Word, Excel, PowerPoint a Outlook. Ačkoliv všechny tyto aplikace mají mnoho možností, co se s nimi dá dělat, tak při hlubším zkoumání lze zjistit, že vzorci v Excelu a animacemi v PowerPointu to nekončí.

Visual Basic for Application (dále VBA) je součástí Microsoft Office programů. „Basic“, který je součástí názvu VBA, je akronymem v překladu pro univerzální symbolický instrukční kód pro začátečníky, což naznačuje, že se nejedná o profesionální programovatelný jazyk.

Tento jazyk se poprvé objevil v roce 1960, kdy se využíval jako možnost výuky programovacích technik pro studenty. Basic se rychle uchytíl a rozšířil se do stovek dialektů pro mnoho typů počítačů.

V průběhu let se Basic stále měnil a vylepšoval. Dříve, v prvních implementacích, byl Basic pouze „tlumočený jazyk“, tedy vše muselo být podrobně vypisováno, což způsobovalo pomalé vykonávání příkazů. Nejmodernější verze už však umožňuje kód kompilovat a převádět do strojového kódu, jehož výsledkem je rychlejší a mnohem efektivnější vykonávání příkazů. VBA jako takové, které je vlastně postaveno na Basic se začalo využívat od roku 1991 jako součást aplikací od Microsoftu (Walkenbach, 2013).

Základem používání VBA k provázání různých Microsoft Office aplikací je pochopení modelu objektů. Jednoduše se jedná o manipulaci s objekty, kdy každá aplikace (např. Word, Excel a PowerPoint) má své vlastní unikátní objekty. Je možné tedy ku příkladu z Excelu ovládat Word a naopak.

VBA se skládá z několika základních částí. Kód jako takový, který vykonává určité příkazy napsané strojově nebo naprogramované člověkem. Modul je součástí každého souboru, který je možné editovat pomocí Visual Basic Editor (VBE), jedná se tedy o VBA modul složený z procedur. Procedury jsou část kódu, které vykonávají určité akce. Může se jednat o dílčí procedury nebo o funkční procedury. Poslední důležitou část tvoří objekty. Objekty jsou důležitou částí každé procedury. Jedná se o veškeré možné části aplikace, se kterými lze pomocí VBA manipulovat.

Objekty se dělí na kolekce (Collections), aktivní objekty (aktuálně zvolená buňka, zvolená tabulka apod.) a také různá nastavení objektů.

2.1 Všeobecné využití VBA

VBA ve zkratce slouží především k automatizaci, ulehčení práce, ke zdokonalování vzorců, případně i k tvorbě vlastních vzorců. Nejsnazším příkladem je hlavně Microsoft Excel. Skrze VBA je možné přidávat nové funkce, které lze poté používat normálně v jakékoliv buňce daného souboru, stejně jako se může použít funkce KDYŽ, PRŮMĚR apod. Tak si můžete vytvořit úplně vlastní funkci.

Často se využívá VBA pro automatizaci rutinních opakujících se procesů, které lze snadno naprogramovat, nebo pouze zaznamenat pomocí zabudovaného zaznamenávání makra. Uživatel poté provede daný postup a může jej opakovaně spouštět pouhým stisknutím navoleného tlačítka na klávesnici nebo vloženým virtuálním tlačítkem na daném listu dokumentu (Walkenbach, 2013).

VBA umožňuje také provázat jednotlivé Microsoft aplikace a tím automatizaci posunout na vyšší úroveň. Například z Microsoft Outlook při rozesílání e-mailů osobám v seznamu, který je umístěn v Microsoft Excel souboru.

Provázáním aplikací to ale nekončí. Skrze VBA je také možné ovládat základní funkce počítače, jako je například vkládání souborů do zipu, vyhledávání souborů napříč adresáři určitého disku, otevírat webové stránky, převádět dokumenty do PDF a mnoho dalšího. Ačkoliv se jedná o jeden z nejdéle fungujících programovacích jazyků, má velké možnosti. Záleží už jen na zkušenosti programátora.

2.2 Výhody a nevýhody VBA

VBA představuje nespočet možností, jak zautomatizovat cokoli, co v Excelu lze provádět. Stačí k tomu sepsat patřičný kód, který pak má Excel provést. Tyto úkony lze spouštět tlačítky nebo automaticky při určitých úkonech prováděných v samotném Excelu (Walkenbach, 2013).

Výhody

Mezi výhody patří jednoznačně to, že tak jak se kód napíše, tak přesně tak se i provede. Rychlost prováděných úkonů je mnohem rychlejší, než kdyby to měl

člověk dělat sám, závisí to pouze na výpočetním výkonu počítače. V Excelu je možné také používat zaznamenávání makra, což umožní i neznalcům programování nastavit jisté úkony automaticky, ale provede se to přesně tak, jak to provedl při tvorbě toho záznamu (Walkenbach, 2013).

Nevýhody

Kdokoliv bude potřebovat provádět pokročilejší úkony, než jsou pouze ty nahrané pomocí zaznamenávání makra, tak bez znalostí programování ve VBA není schopen nic vytvořit. Pro programování ve VBA je nutné mít zakoupený Microsoft Office Excel (Walkenbach, 2013).

3 Představení oddělení EPG/3 ve společnosti ŠKODA AUTO a.s.

Václav Laurin a Václav Klement založili v roce 1895 podnik, který stál u zrodu stoleté tradice výroby českých automobilů a od roku 1991 je součástí koncernu Volkswagen. Společnost se od této chvíle jmenuje ŠKODA AUTO a.s. a je největším výrobcem automobilů v České republice a čtvrtým nejstarším výrobcem automobilů světa. To lze velmi dobře vidět i z výroční zprávy roku 2019, ve které je udáván rekordní nárůst počtu kmenových a agenturních zaměstnanců, a to na celkový počet 41 800 ke dni 31. 12. 2019. Hlavním důvodem je konsolidace indických společností pod ŠKODA AUTO a také rozšíření odborných kompetencí v technickém vývoji, IT a dalších koncernových projektech. Po celém světě se prodalo celkem 1 243 tisíc automobilů a do technického vývoje firma vynaložila 25,2 mld. Kč, to činí nárůst o 2,7 mld. Kč oproti minulému roku.

Mise firmy: *„Touha vynalézat. Už téměř 125 let věnujeme svůj podnikatelský duch a vášně individuální mobilitě. A budeme v tom pokračovat i v budoucnu!“* (ŠKODA AUTO a.s., 2020, s. 14).

Vize firmy: *„Simply Clever společnost pro nejlepší řešení mobility. Pro rodiny, podnikatele, cestující nebo prostě znalce, kteří si chtějí užít radost z jízdy, je ŠKODA inteligentní volbou. Chytré nápady pro individuální mobilitu nás pohání téměř 125 let. Nyní nastal čas vynalézt nejlepší mobilní řešení pro budoucnost.“* (ŠKODA AUTO a.s., 2020, s. 14).

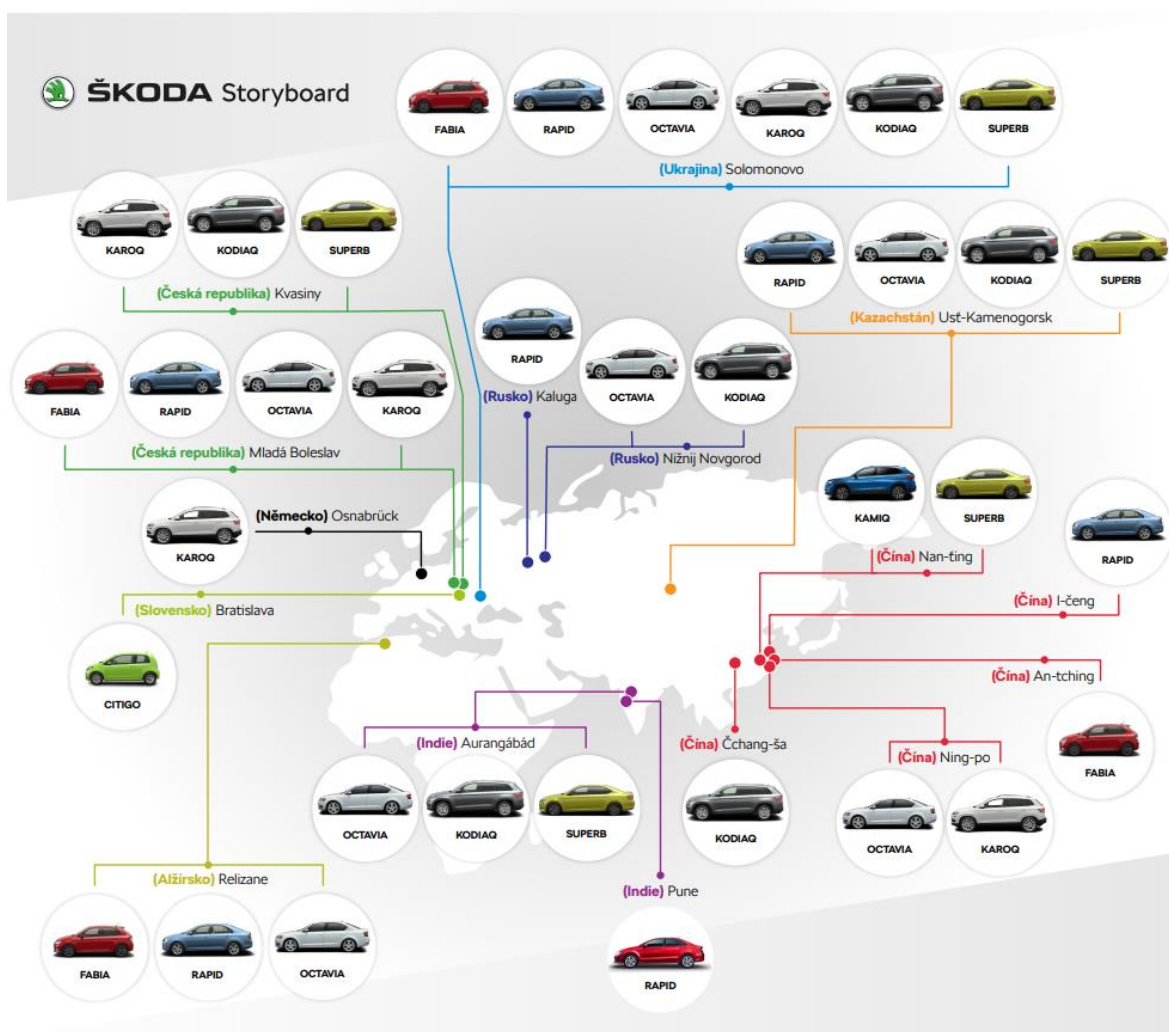
Aktuálně firma vyrábí ve svém hlavním závodě v Mladé Boleslavi na jedné výrobní lince modely Fabia, Scala, Kamiq a na druhé výrobní lince modely Octavia, Karoq a Enyaq.

V závodě Kvasiny, který je považován za nejmodernější závod firmy, se vyrábí vlajková loď Superb a také Kodiaq. Ve vedlejším závodě ve Vrchlabí se vyrábí převodovky DQ 200, které se vyrábí nejen pro firemní potřeby, ale také pro další koncernové značky spadající pod Volkswagen. Do výrobního závodu v Bratislavě se plánuje přesun výroby modelu Superb, čímž se uvolní kapacita linky a umožní se tím vyrábět nadcházející Fabii Combi.

Mimo těchto tří závodů na území České republiky a jednoho na Slovensku (viz Obr. 4) se Škoda vyrábí také v Německu, Rusku, Indii, Číně, Kazachstánu a na Ukrajině (ŠKODA AUTO a.s., 2020).



KDE SE VYRÁBÍ JEDNOTLIVÉ MODELY ŠKODA?



Zdroj: (ŠKODA AUTO a.s., 2020a)

Obr. 4 Výroba jednotlivých modelů ŠKODA ve světě

Oddělení EPG/3

Oddělení EPG/3 zajišťuje v ŠA aplikaci automatických převodovek, jejich zkoušky, vnější řazení a software pro převodovky typu DQ200. Tyto převodovky se aktuálně používají ve vozech s objemem motoru do 1,5 litru v případě benzínového agregátu, omezením je maximální možný točivý moment 250 Nm. Nejedná se o převodovky používané pouze ve vozech značky ŠA, ale také VW, AUDI a SEAT. Tým skládající se z 20 lidí pravidelně provádí jízdní zkoušky, úpravy a uvolňování softwaru pro výrobu. Mají pod svou správou přes 100 různých softwarů pro různé kombinace

převodovky, motorizace a trhu. Odpovídají mimo jiné za celý uvolňovací proces, homologace a řeší připomínky od zákazníků, kde v tomto případě se zákazníkem rozumí ŠA, VW, AUDI, SEAT a také samozřejmě koneční spotřebitelé.

Na oddělení pracuje 8 interních pracovníků a každý z nich je specialista na jinou oblast (rozjezdy, řazení, jízdní strategie, a další). Každý je zároveň zodpovědný za určitou motorizaci, trh a funkci. Někteří mají na starosti i správu dat jednotlivých projektů a také tvorbu různých automatizací rutinních úkolů, snazší správu dat a komunikaci v týmu.

Externí pracovníci jsou rozděleni do třech skupin. V jedné skupině jsou pracovníci, kteří stíní stejné funkce a oblasti, jako interní pracovníci, protože je nutné, aby měl zástupce v případě nepřítomnosti, měl s kým konzultovat a řešit případné problémy. Ve druhé skupině jsou dvě osoby, které zajišťují potřebnou dokumentaci, sledují stav projektů, tvoří podklady a schůzky k jednotlivým projektům. Ve třetí skupině jsou osoby zodpovědné za platformu, tzn. správa samotné „kastle“, tedy převodovky jako takové a pracují především s výkresy, komponenty a sledují zástavby v různých projektech.

4 Aplikace a systémy používané pro předávání projektových informací

Na oddělení vývoje automatických převodovek ve ŠKODA AUTO a.s. se využívá mnoho programů, systémů a aplikací, které obsahují důležité vývojové informace. V každém však jsou napsané jiné údaje a je nutné si je poté propojovat dohromady a vytvářet si z toho vlastní úplný seznam. Tyto informace se mění na pravidelné bázi a znamená to mnoho hodin práce při zjišťování daných informací, nehledě i na zdržování při otevírání těchto programů. Cílem tedy bylo vymyslet takový způsob předávání informací, kdy bude tým lidí starající se o vyhledávání těchto projektových informací a shlukování je na jedno místo, odkud bude pro ostatní snazší si je vyčítat ve srozumitelném a celistvém formátu. Na základě těchto požadavků vznikla v Microsoft Excel tabulka nazvaná „Přiřazovací tabulka SW“, ve které se hojně využívá VBA. Tato tabulka samozřejmě sama o sobě nemá takovou přidanou hodnotu, a proto k ní vzniklo několik dílčích procesů rovněž fungujících v Microsoft Excel zautomatizovaných pomocí VBA sloužící pro tvorbu reportů a porovnávání.

4.1 Situace před zavedením změn

Jak bylo nastíněno, tak původně situace byla taková, že informace o každém projektu bylo nutné dohledávat. Tato změna vznikla na popud kolegů, kteří se starají o vývoj a údržbu softwaru pro automatické převodovky, nazýváme je aplikátory. Pro dohledávání informací se využívá mnoho programů, které vyžadují speciální oprávnění, využívá se k tomu PKI čip v kartě a v některých případech trvá poměrně dlouho, než se načtou.

V případě, že aplikátor potřeboval zjistit nějakou informaci o voze, musel čekat dlouho, než se k tíženému údaji dostal. Často to znamenalo také dohledání v dalším programu, aby bylo možné zjistit si kompletní informaci o daném projektu. Někde jsou uvedeny SW, jinde zase označení převodovky, SOP a EOP a mnoho dalšího.

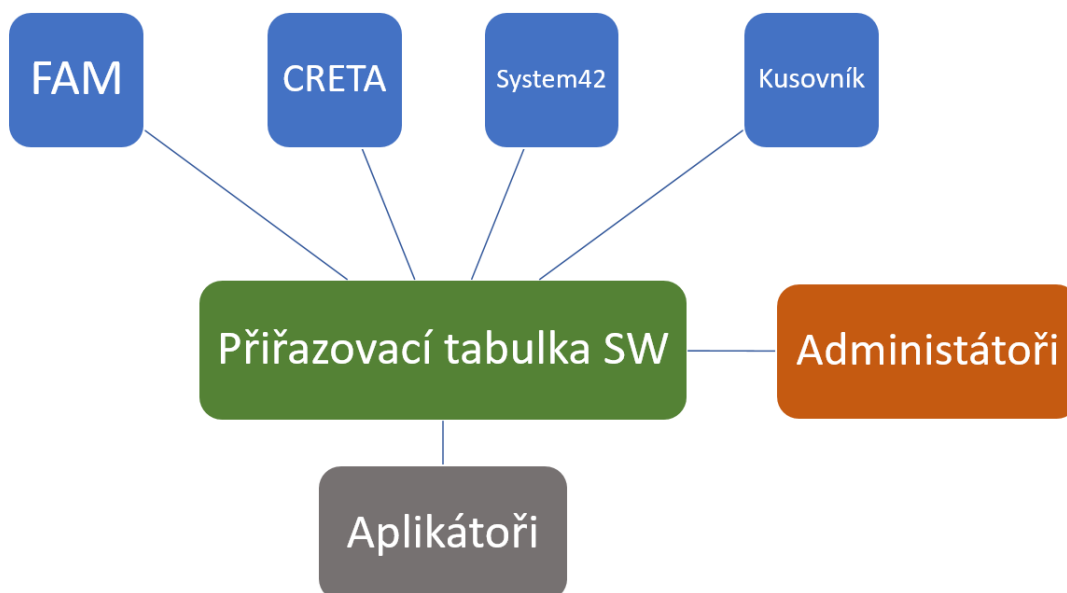
Pokud aplikátor potřeboval dané informace zjistit, bylo nutné k tomu vynaložit neúměrné množství času. I za předpokladu, že danou informaci hledal někdy dříve, stejně musel proces opakovat, jelikož se údaje nikam neukládaly pro pozdější dostupnost.

Aplikátoři se tak „okrádali“ o čas, kdy se mohli starat o samotnou aplikaci SW automatické převodovky. Informace byly roztržštěné v různých programech a tabulkách.

4.2 Přiřazovací tabulka SW

Po více jak dvou letech působení na oddělení začala postupně vznikat ve spolupráci s kolegyní tabulka, kterou měla na starosti poté plnit. Obsahovala informace o projektech shromážděných z několika různých programů a tabulek. Tato tabulka se začala postupně přeměňovat v „chytrou“ a využívalo se tam čím dál tím více VBA. Po více jak roce úprav a dodělávání změn se dostal vývoj přiřazovací tabulky SW tam, kde je nyní. Existuje tým lidí, kteří mají za úkol dohlížet na aktuálnost a plnění informací do tabulky, tzv. administrátoři.

Aplikátoři mají nyní co nejpřehlednější a nejrychlejší možný způsob, jak zjistit, co do kterého auta patří, jaký software převodovky, který je nejaktuálnější, kdy bude uveden do série a mnoho dalších informací. Všechny tyto informace se shlukují v tzv. Přiřazovací tabulce SW. Na obrázku 5 je vyobrazeno, jak proces kolem tabulky funguje, aby se udržovala stále aktuální a relevantní.



Obr. 5 Procesy prováděné na pozadí přiřazovací tabulky

Do tabulky vstupují data z několika různých aplikací. Nejvýznamnějšími zdroji jsou právě FAM, CRETA, System42 a kusovník. Každý projekt, který je uveden

v tabulce, obsahuje data právě z každého uvedeného zdroje a administrátoři jsou těmi, co tato data shlukují do jednoho zdroje – Přiřazovací tabulky SW, avšak proces je komplikovanější v tom, že každý zdroj má data strukturována jinak a není to tak snadný proces, proto je důležité, že tato tabulka umožňuje administrátorům data zpracovat pro všechny projekty a poté již udržovat jen jejich aktuálnost a doplňovat k nim nové informace. S daty z této tabulky poté pracují aplikátoři.

V případě vývoje softwaru se jedná o několik desítek projektů, které se vyvíjejí napříč značkami Škoda, VW, Audi a Seat. Pro všechny tyto vozy s točivým momentem 250 Nm včetně, se software automatické převodovky vyvíjí ve ŠKODA AUTO a.s., na oddělení vývoje automatických převodovek. Pro představu se jedná o více než 100 různých kombinací motoru, převodovky, výkonu a hledí se také na emisní normu a trh, kde se bude vůz prodávat. Každá tato kombinace má svůj vlastní software.

Název tabulky právě vznikl na základě toho, že se ke každému projektu přiřazují softwary automatických převodovek a mnoho dalších údajů. Ve výsledku tabulka umožňuje dohledávat vývoj softwaru zpětně, k danému datu a také do budoucnosti. Vše to však závisí na tom, jak tam pracovník, starající se o administraci dané tabulky, zanesse data.

Samotná tabulka zvládne několik operací pro usnadnění práce s hledáním, zadáváním a kontrolováním všech projektů v ní zanesených. Umí se i postarat o problémy, které mohou vzniknout kvůli překlepu, jelikož si ukládá i historii prováděných změn a vždy lze dohledat, kdo jakou změnu a jaký úkon v tabulce provedl.

Přiřazovací tabulka SW se může také svou funkcionalitou rovnat jisté formě databáze obsahující důležité informace a pomocí formuláře se z databáze vyvolávají potřebné položky k zobrazení, dle výběru uživatele. Tabulka tedy představuje databázi. Formulář, který se otevírá po kliknutí na zvolený projekt, je klientem, respektive uživatelem.

Statistiky Přiřazovací tabulky SW

Pro představu, jak hojně se využívá VBA v Přiřazovací tabulce a jak dlouho její vývoj probíhal v přepočtu na odpracovaný čas, je zde přiložena tabulka 1, ve které jsou tyto informace uvedeny. Její vývoj se samozřejmě nezastavil a stále pokračuje.

Tab. 1 Statistiky Přiřazovací tabulky SW

Statistika	Hodnota
Odpracovaný čas	600+ hodin
Počet GLOBAL proměnných	59
Počet PUBLIC proměnných	39
Počet DIM proměnných	106
Řádků kódu	4894 řádků
Sub modulů	131

4.2.1 Monitorované údaje o každém projektu

U každého projektu je nutné sledovat velké množství dat, které musí být okamžitě a přehledně zobrazované pro uživatele tabulky. Mezi sledovanými údaji jsou data, kdy jaký projekt má SOP (začátek výroby), PVS (kdy vznikne první prototyp) a především MS (milníky – předem stanovená data v průběhu vývoje projektu). Každý MS představuje jiné důležité datum, ke kterému se musí splnit jistá činnost, proto je důležité je pečlivě hlídat a dodržovat.

U každého projektu se sledují čtyři oficiální stavy vývojového softwaru. Série, která už patří do vozů přímo na linku. Vorserie je pro vozy sloužící k testování vozů v různých podmínkách (kopce, jízda ve městě, mimo město, a další...), kde by se většina funkcí převodovky již měla doladit. VFF stav jsou ranné fáze testování, kde vůz je většinou tzv. Agregatträger, kdy se jedná jen o zástavbu některých prvků vývojového vozu. MS4-LF je software, který se zpravidla dělá tzv. „od stolu“ a vznikne na základě průvodní dokumentace projektu proto, aby vyrobený kus byl schopen jízdy.

Při vývoji softwaru se využívají tzv. pracovní sady, které představují mezistupeň mezi oficiálními stavy (Serie, Vorserie, VFF, MS4-LF) a mění se v závislosti na nutných změnách při testování softwaru. Vývoj obnáší také různé dokumentace,

kteřé většinou jsou uložené v dalších aplikacích/programech jako je například FAHST, DMS a odkazy na ně jsou u každého projektu.

U každého projektu se může stát, že v průběhu vývoje se budou měnit i oficiální stavy softwaru. K tomu slouží políčko „Vývoj SW“, kde jsou tyto změny zachyceny.

Vývoj vozu je navázán mimo jiné i na emisní normy, které rovněž ovlivňují vývoj softwaru převodovky. Každá emisní norma znamená jiný software, a proto je nutné sledovat i toto kritérium. Také je nezbytné k tomu dokládat dokumentaci o změnách softwaru, protože se může jednat o emisně relevantní změny.

4.2.2 K čemu slouží jednotlivé listy

Přiřazovací tabulka SW obsahuje celkem 4 listy. Každý z nich má samozřejmě jiný účel a jen některé vidí uživatelé.

„Aktuelle überzicht“ obsahuje seznam aktuálně běžících projektů, tedy ty, které ještě nemají po EOP. Tomuto listu sekunduje „Nach EOP Projekte“, který obsahuje projekty již neaktivní, ale informace k nim jsou stále důležité a relevantní. Oba tyto listy jsou viditelné všem uživatelům tabulky.

Další list se jmenuje „History“ a jak už název předpovídá, jedná se o seznam provedených změn v tabulce, kde je vidět, kdo, kdy a kde jakou změnu provedl. Tento list je viditelný pouze administrátorům tabulky.

„Základní údaje tabulky“ jsou posledním listem, který vznikl pro možnost úpravy některých prvků tabulky bez nutnosti upravovat VBA. Je zde tedy uloženo, kdo má dostat práva administrátora, jakým způsobem se vygeneruje nová modelová péče a jak se mají podbarvovat řádky dle emisních norem.

Všechny tyto listy budou probrány v následující kapitole s jednotlivými funkcemi, kterých se týkají.

4.2.3 Základní funkce přiřazovací tabulky

První a důležitou funkcí, kterou Přiřazovací tabulka SW skýtá, je rozlišování, zda se jedná o administrátora, nebo uživatele. VBA při otevření souboru provede patřičné kroky k úpravě toho, co daný uživatel vidí. Nejprve si tedy ověří, zda přihlášený uživatel je uveden v tabulce, kteří mají mít práva administrátora. Následně dle toho

zobrazí nebo skryje list s názvem „History“ a „Základní údaje tabulky“, režim přístupu se změní na „Jen pro čtení“.

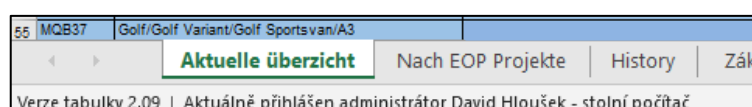
Každý, kdo si otevře hromadné úpravy, zobrazovací nebo vyhledávací okno, zjistí, kdo z administrátorů provedl poslední změnu v tabulce a kdy daná změna proběhla. Tyto informace jsou uvedeny v horní liště zobrazovacího okna (viz Obr. 6).



Obr. 6 Informace o poslední změně údajů

Může se jednat o důležitou informaci v případě, že si nejste jistí tím, kdy se zadávala aktuální data, nebo v případě, že se domníváte chyby zápisu dat, což mohlo nastat pouhým uklepnutím. Administrátor tabulky to však dokáže dohledat v historii změn hodnot a může chybu snadno, relativně okamžitě, napravit. Zjišťování této informace je poměrně snadné. Při každém kliknutí do buňky se uloží aktuální hodnota do proměnné „OldValue“ a v případě změny této hodnoty (porovnáním staré a nové), se vyvolá procedura, která zapíše na nový řádek do listu „History“ změnu. Zároveň se tato informace reflektuje právě do informace o poslední aktualizaci.

V pravé dolní části Excelovského okna, jak je vidět na obrázku 7, je napsáno, o jakou verzi tabulky se jedná. Tento údaj je především informativní, využívá se hlavně v případě nějakého problému. Tabulku je totiž možné používat také v omezeném režimu off-line, a tak je důležité vědět, jakou verzi uživatel používá. Také je zde napsáno, zda je přihlášen uživatel nebo administrátor tabulky a kdo konkrétně. Informace o aktuální verzi se získává z formuláře „Historie aktualizací“, kde to z poslední záložky vezme název verze (každá verze má vlastní záložku). Aktuálně přihlášený uživatel/administrátor je uložen v listu „Základní údaje tabulky“. Tato informace je také důležitá pro případ nahlášení nějaké chyby a snadno lze dohledat, jakou verzi daný uživatel používal. Tabulka totiž lze používat také off-line.



Obr. 7 Verze VBA a kdo je aktuálně přihlášen

Zobrazovací okno

Zobrazovací okno je tím nejdůležitějším, co tabulka umí. VBA kódování je zde nejrozsáhlejší a obsahuje mnoho kontrolních oblastí, zjišťování dostupnosti souborů na zvoleném FTP serveru a mnoho dalšího.

Okno se zobrazí při kliknutí na jakýkoliv projekt ve sloupci B (Fahrzeug) a zobrazí náhled projektu (viz Obr. 8), ve kterém jsou všechny základní informace o projektu. Uživatel zde má několik možností. Obrázek je proto rozdělen do šesti částí, kde každá část bude zvlášť popsána.

Náhled projektu - poslední aktualizaci přiřazovací tabulky provedl/a David Hloušek - stolní počítač dne 17.10.2020 v 21:51

Informace o zvoleném projektu pro modelovou péči: 48/20 Připojení ke sdílenému disku: i.O.

Platforma MQB37 Fahrzeug Tarek [redacted]
KPB VW316/SCS PKZ 2GG
Motor 1,4l 110kW TFSI China PR-Nr. 4BD Zusatz OPF
Verantwortlich LUKAS [redacted] Homologationstand [redacted]
Aktuální pracovní sada [redacted] [Nedostupné] [Nedostupné] [Nedostupné] **1**

Informace k modelové péči

	SOP	MS1	MS2	MS3	MS4	MS5	MS6	PVS	MS7
GKB	UWK	48/20	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Bemerkungen [redacted]
Komentář ve sloupci B [redacted]
Otevřít FAHST link https:// [redacted] **2**

Software

Výběr SW
Formát parametrové sady: .par .odx
Poslední aktualizace modelové péče: 10.07.2020
Serie FL_0CW [redacted] **3**
Vorserie FL_0CW [redacted]
VFF [redacted]
MS4-LF [redacted]

Možnosti pro zvolený SW
EF-S [redacted] Otevřít
Pracovní sada
U22_ [redacted]
Vývoj SW
FL_0CW300051_2201_UJLH
Otevřít DMS link
DMS Link https:// [redacted] **5**
Datum MSG 13.07.2020

Co si přejete s parametrovou sadou provést?
[Nedostupné] [Nedostupné] [Nedostupné]
Umístění souboru v případě stažení:
C:\Users\David Hloušek\Desktop **4**

Vyhledání parametrové sady Hromadné úpravy Zobrazit vysvětlivky emisních norem Historie aktualizací Zkontrolovat připojení
Uložit změny Režim úprav Přidat projekt Přesunout projekt do archivu **6**

Obr. 8 Zobrazovací okno – náhled projektu

V první části náhledu projektu jsou zobrazeny základní informace, které jsou s projektem pevně spjaty a nemění se v průběhu jeho životnosti. Tato část se jednoduše načítá ze sloupců na zvoleném řádku.

Uživatel má možnost stáhnout aktuální pracovní sadu pro daný projekt. Předpoklad je takový, že bude odpovídat nejaktuálnějšímu stavu, který je dostupný v programu CRETA. Pokud nějaká pracovní sada je uvedena, zkontroluje se, zda je soubor dostupný ke stažení a na základě toho se zpřístupní příslušná tlačítka (stažení, připojení jako příloha do E-mailu, otevření).

Uživatel má možnost v horní části volit modelové péče, ve kterých má projekt zapsaný alespoň jeden SW, což jinými slovy znamená, že pokud není ani jeden SW pro daný projekt v určité modelové péči zapsán, nezobrazí se ani v rozbalovacím okně. Tím má uživatel vyfiltrováno, ve které modelové péči projekt existuje. Při přepínání modelové péče provede VBA kompletní znovu načtení celého náhledu projektu. Každá modelová péče má „master“ sloupec, dle kterého se zjišťují další sloupce, ve kterých jsou potřebné informace k načtení.

Do druhé části spadají užitečné informace o datech jednotlivých milníků, včetně SOP a PVS. Zápis je formou KW/JJ (týden/rok), ale pro snazší orientaci se, po najetí myší na dané textové pole, zobrazí i informace o přibližném datu (zpravidla to má být k pondělku daného týdne). V příloze 1 je ukázka toho, jak se vypočítávají jednotlivé milníky. Proměnná *myDate* představuje datum ve formátu *DD.MM.RRRR* a zjišťuje se dle zadaného SOP. Formát KW/JJ se převede do formátu data a následně se od něj odečte pro každý milník příslušný počet týdnů a z nich se poté dle funkce *RecalculateMS* zjistí formát KW/JJ.

Každé políčko milníku může mít tři různá podbarvení (viz Obr. 9). Bílé podbarvení znamená, že se ještě neblíží čas daného milníku, nějaký bližší milník je ještě před ním. Zelené podbarvení nám oznamuje, že se blíží jeho čas. Šedé políčko znamená, že daný milník již proběhl. Zjišťování probíhá podobně jako u funkce *RecalculateMS*, kdy každé KW/JJ je převedeno na datum a porovnává se s aktuálním datem v době kontroly. Jakmile narazí na první se blížící/aktuální, tak ostatní ponechá bílé.

SOP	MS1	MS2	MS3	MS4	MS5	MS6	PVS	MS7
40/21	1/19	27/19	45/19	1/20	33/20	45/20	53/20	9/21
~ 01.10.2021								

Obr. 9 Zobrazení milníků projektů

Mezi další informace patří také GKB (označení dílu převodovky), poznámka, komentář ve sloupci B a odkaz k otevření projektu ve FAHST.

Ve třetí části zobrazovacího okna se nachází výběr SW. Uživatel si nejprve musí vybrat, zda ho zajímá formát PAR nebo ODX (jedná se o formát souboru používaný pro flashování do jednotky – PAR je pro vývojové účely a ODX se může využívat i pro sériovou výrobu). Poté si uživatel vybere, jaký oficiální stav SW ho zajímá. Má možnost mezi volbou SW Serie, Vorserie, VFF a MS4-LF.

Po výběru oficiálního stavu SW a formátu sady se zobrazí zbylé možnosti. V případě formátu PAR je možné soubor otevřít skrze poznámkový blok, připojit do mailu jako přílohu nebo stáhnout do předem zvoleného umístění (v základu se soubor stahuje na plochu). Formát ODX dovoluje to samé jako formát PAR, krom otevření souboru skrze poznámkový blok. Pro ukázkou v příloze 2 je zobrazena funkce otevírání souboru PAR v poznámkovém bloku. Počítá se i s tím, kdy náhodou něco selže. V ten moment se vyvolá funkce *ContactAdministrator*, která je popsána v další části podrobně.

Na pravé straně, v páté části jsou další možnosti, které se mění podle zvoleného oficiálního stavu SW. Zobrazuje se tam informace o názvu a případné možnosti otevření BMG/EF-S souboru v PDF (jedná se o průvodní dokumentaci k projektům a jejich SW), z jaké pracovní sady SW vyšel, jak se případně SW měnil a v neposlední řadě možnost otevřít DMS link (VW aplikace, ve které jsou uloženy důležité dokumentace) v prohlížeči a kdy bude/bylo datum milníkového rozhovoru.

Tabulka také dokáže zjistit, zda uživatel má aktuálně internetové připojení a na základě tohoto zjištění se přepne do online nebo off-line režimu, čímž se zablokuje možnost cokoliv stahovat a otevírat odkazy. Pro opětovné otestování připojení slouží tlačítko „Zkontrolovat připojení“ v pravé dolní části okna. VBA zde provádí zjišťování připojení na vzdálený FTP server (tedy kontroluje připojení

na internet), nebo na dřívější funkci sdíleného disku G, což byla kontrola připojení na intranet. Ukázka toho, jak funkce *CheckConnection* funguje je v příloze 3.

Šestá část obsahuje tlačítka, která usnadňují další správu projektů. Každý projekt se odlišuje nejenom podle kombinace motoru, převodovky a výkonu, ale také dle emisní normy, kterou musí splňovat. Emisních norem je několik desítek druhů. Prakticky nikdo si je všechny nepamatuje z hlavy podle označení, proto existuje seznam, který je přeložený pro naše snazší pochopení. Uprostřed v dolní části okna je tlačítko „Zobrazení vysvětlivky emisních norem“, které otevře postranní okno se seznamem emisních norem. Vysvětlivky se doplňují na základě informací v listu „Základní údaje tabulky“ včetně barvy, na základě které se bude podbarvovat řádek dle vyplněného názvu normy (viz Obr. 10). Na každém řádku je zkratka emisní normy, k ní také vysvětlivka.

Podbarvení řádků dle PR.Nr-Abgas		
Norma	Vysvětlivka	Barva
Zkratka	Vysvětlivka normy	

Obr. 10 Ukázka tabulky pro emisní normy

Mezi další tlačítka patří také „Historie aktualizací“, což jen otevře nové okno, ve kterém jsou popsány jednotlivé změny provedené ve fungování tabulky.

Tlačítko „Hromadné úpravy“ je provázání na jiné okno, které slouží pro další užitečné činnosti, které tabulka zvládne provádět. Více o hromadných úpravách v další podkapitole.

Posledním tlačítkem je „Vyhledání parametrové sady“, které otevře okno pro zapsání minimálně čtyř znakového klíče, co má vyhledat. Více o vyhledávání parametrových sad v další podkapitole.

Tlačítka „Uložit změny“, „Režim úprav“, „Přidat projekt“ a „Přesunout projekt do archivu“ jsou součástí vyšších práv, administrátorských, která jsou popsána v jiné kapitole, kde budou popsány všechny funkce administrátorů podrobně.

Hromadné úpravy

Hromadné úpravy slouží pro provádění obsáhlejších činností. Červeně vyznačená oblast platí pro všechny uživatele tabulky. Vzhled je vidět na obrázku 11.

The screenshot shows a software window titled 'Zpracování úkolu - poslední aktualizaci přiřazovací tabulky provedl/a David Hloušek - stolní počítač dne 17.10.2020 v 21:51'. The main area is titled 'Zpracovávání dat pro modelovou péči: 2021'. It contains several sections: 'Stážení modelové péče/Basis pracovní sady' with a 'Stáhnout pro 2021' button and file format options; 'Výpis SOP projektů' with a 'Vygenerovat výpis SOP' button and a file path; 'Aktualizace modelové péče' with a 'Potvrdit aktualizaci' button and a file path; 'Aktualizace EF-S PDF' with a 'Potvrdit aktualizaci' button and a file path; 'Ze ZIPu vztít jen pracovní sady z tabulky' with an 'Uložit pracovní sady z tabulky' button and a file path; 'Aktualizace barev' with an 'Aktualizovat dle emisních norem' button; and 'Nová modelová péče' with a 'Přidat modelovou péči' button. A red box highlights the top part of the window, including the title bar and the first two sections.

Obr. 11 Zpracování úkolu – hromadné úpravy

V případě, že je to potřeba, může si uživatel zvolit stažení kompletní zvolené modelové péče. Jedná se o stažení pouze vyfiltrovaných řádků. Uživatel si tak může předdefinovat, které motorizace ho zajímají a ty si nechat stáhnout.

Uživatel má ještě možnost si zvolit, která varianta ho zajímá – Serie, Vorserie, VFF nebo MS4-LF. Varianta Jahr je jako doplňková, v té je pro každý projekt dané modelové péče nastaven nejnovější SW, který je dostupný (např. když je dostupný někde software pro Serii, stáhne se právě ten. V případě že daný SW dostupný není, vloží se tam Vorserie, poté VFF nebo MS4-LF stav).

Užitečná funkce, hlavně pro administrátory, je možnost vytvoření výpisu SOP všech projektů. Opět to funguje s filtrem čili jen vyfiltrované řádky budou použity pro vytvoření výpisu. Report dat pak vypadá jako na obrázku 12. Opět, stejně jako pro náhled projektu, se zde podbarvují jednotlivá políčka dle toho, v jakém stavu je každý milník (proběhl, nejbližší nadcházející, ostatní).

SOP	MS1	MS2	MS3	MS4	MS5	MS6	PVS	MS7
25/21	38/18	12/19	30/19	38/19	18/20	30/20	38/20	47/20
	---	---	---	---	---	---	---	---

Obr. 12 Výpis SOP všech projektů

Samozřejmě je uvedeno také u každého projektu: Fahrzeug (název vozu), Konzern-Projekt (koncernové označení projektu), Motor (typ motorizace včetně kW) a PR.Nr-Abgas (tedy přesné označení emisní normy).

Vyhledávání parametrových sad

Vyhledávání parametrových sad funguje dvěma způsoby. V obou případech je nutné zadat alespoň 4 znaky pro zahájení vyhledávání. Vyhledávání probíhá na základě daného klíče v obou listech (aktuální i neaktuální projekty) a napříč všemi sloupci a řádky, které obsahují SW. Na obrázku 13 je vidět vzhled daného okna. Pokud vyhledávání je úspěšné, tak v rozbalovacím okně je vidět vždy list, ve kterém byla sada nalezena, kolikátá to je sada a kompletní název sady. Další zobrazené informace se načítají podobně jako v náhledu projektu. V tomto okně je ještě užitečnou informací, o jakou modelovou péči se jedná a jaký je to stav SW (na obrázku 13 se jedná o SW Vorserie). Uživatel si po vybrání jedné z nalezených sad může skrze tlačítko „Načíst výběr v zobrazovacím okně“ přenést načtenou sadu do náhledu projektu (zobrazovací okno, viz Obr. 8), kde s ní může pracovat dál. V případě, že bude nalezena pouze jedna sada, otevře se rovnou zobrazovací okno s vyhledanou sadou v příslušné modelové péči a se zvoleným SW.

Zvolte sadu ze seznamu

Vyhledat Sada k vyhledání: 1697

Nalezené sady: Aktuelle überzicht | 1_FL_0CW Načíst výběr v zobrazovacím okně

Informace o nalezené sadě v modelové péči 48/18 Zvolený řádek 78

Plattform MQB37 Fahrzeug Golf/Golf Variant/Golf Sportsvan/Octavia/Octavia Combi

KPB VW PKZ

Motor 1,5l 110kW EA211evo PR-Nr. Zusatz

Verantwortlich JAN Homologationstand

Aktuální pracovní sada

Jedná se o software Vorserie.

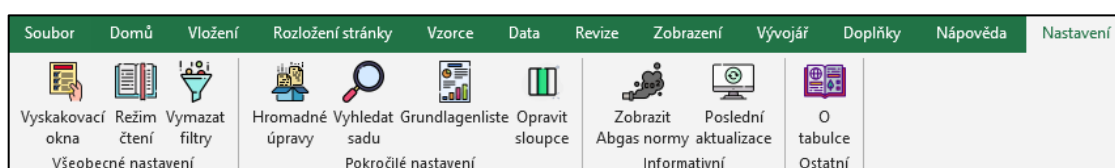
Obr. 13 Okno vyhledávání sady

Prohledávání funguje zprava doleva, a to z jednoduchého důvodu. Když totiž hledání probíhá, z každého řádku to vyhledávaný klíč zařadí do seznamu výsledků pouze jednou, protože jinak mohou vznikat duplicity kvůli zápisu SW do sloupce „Vývoj SW“ a případně také do sloupce s neaktuálnějším SW „Jahr“.

Vyhledávání je možné zobrazit také přímo tlačítkem v horní liště v záložce „Nastavení“. Začátek je trochu odlišný, a to v tom, že hned po kliknutí na „Vyhledat sadu“ se otevře vyhledávací okno, kam se zadá klíč, podle kterého se má sada vyhledávat. Postup je poté úplně stejný jako v předchozí metodě. V tomto okně je také vždy po zvolení napsáno, o jaký oficiální stav SW se jedná.

Horní záložka „Nastavení“

V tabulce je také speciálně vytvořená horní lišta, která se nazývá „Nastavení“. Obsahuje několik užitečných tlačítek, které se jednou za čas hodí každému uživateli. Na obrázku 14 je vidět, jak tato záložka vypadá.



Obr. 14 Záložka „Nastavení“

Tvorba a úprava záložky se může provést například skrze program *Office Ribbon X Editor* od vývojáře Fernando Andreu (Andreu, 2021). Samozřejmě je možné docílit toho samého i skrze VBA, ale výhodou tohoto řešení je okamžité zobrazení, jakmile se tabulka otevře. Není k tomu nutné povolovat VBA. Na obrázku 15 je vyobrazena skupina tlačítek „Všeobecné nastavení“.

```

customUI14.xml
1  <customUI xmlns="http://schemas.microsoft.com/office/2009/07/customui">
2  <ribbon startFromScratch="false">
3    <tabs>
4      <tab id="customTab" label="Nastavení">
5        <group id="basicSettings" label="Všeobecné nastavení">
6          <toggleButton id="openForms"
7            label="Vyskakovací okna"
8            size="large"
9            onAction="OpenFormsState"
10           image="openForms" />
11         <toggleButton id="browsingMode"
12           label="Režim čtení"
13           size="large"
14           onAction="ChangeBrowsingMode"
15           image="ReadMode" />
16         <button id="removeFilters"
17           label="Vymazat filtry"
18           size="large"
19           onAction="removeFilters"
20           image="removeFilters" />
21       </group>

```

Obr. 15 Ukázka XML kódu pro vlastní záložku „Nastavení“

Existuje mnoho možností, co lze skrze tento program provádět, ale neumí to nápodobit, tedy vyskakovací bublinu zobrazující se u tlačítka v záložce. Jedná se hodně o systém „pokus omyl“. U tlačítek se nastavuje, co se provede, když se stiskne a jak se vyvolá (ID tlačítka). Příklad VBA pro „Vyhledat sadu“ je na obrázku 16.

```
Sub SearchPAR(control As IRibbonControl)
    FindForm.Show
End Sub
```

Obr. 16 Ovládání vytvořeného tlačítka skrze Office Ribbon X Editor

V tabulce po kliknutí do jakéhokoliv políčka s textem ve sloupci B způsobí, že vyskočí zobrazovací okno. Pokud je tlačítko „Vyskakovací okna“ zašedlé, tak žádné okno vyskakovat nebude.

Dalším tlačítkem je „Režim čtení“ a jak už i název napovídá, po kliknutí se tabulka „přepne“ do tzv. režimu čtení. Nazývá se to tak kvůli tomu, že se většina ukotvených sloupců schová, ať je pravá část volnějši a lépe se v ní dá orientovat.

Občas se může stát, že je zapnuto několik filtrů a pak už není jisté, jaké filtry jsou zapnuté. Stačí pak kliknout na „Vymazat filtry“ a problém je vyřešen.

Dalšími tlačítky jsou „Hromadné úpravy“, které otevírá okno pro zpracování úkolů a tlačítko „Vyhledat sadu“, které otevírá okno pro vyhledávání, které již byly popsány.

„Grundlagenliste“ tlačítko využívá hlavně administrátor tabulky. Po stisknutí tlačítka se objeví okno (viz Obr. 17), kde si uživatel/administrátor navolí formu výstupu do nového sešitu, který bude formátován tak, jako Grundlagen listiny od německých kolegů. Jedná se tedy o export dat do určitého předem stanoveného formátu. Toto zpracování funguje hlavně na filtrování. Dle zvolené platformy se nastaví filtr projektů, dle varianty a modelové péče se zvolí příslušný sloupec, ze kterého se budou zjišťovat SW. Také je možnost nastavit, aby se doplňoval stav z předchozí modelové péče (pokud nějaká existuje a pokud v daném řádku pro ni existuje SW) a nastavit, zda se budou vypisovat i ty řádky, pro které ještě nejsou známy SW. Dle těchto kritérií se také nastaví „Název“ výsledného souboru. Zároveň, jak již bylo popsáno, ke každému SW existují pracovní sady, které se dělí na různé generace.

Při tvorbě tohoto reportu je nutné také nastavit, jaká je aktuální generace a ta se ve všech exportovaných projektech přepíše dle zadané proměnné hodnoty.

Zpracování Grundlage - poslední aktualizaci přiřazovací tabulky provedl/a Elias [redacted] dne 30.11.20... X

Zpracovávání dat pro modelovou péči: 2021

Vygenerovat export

Zvolit cestu k uložení exportu C:\Users\David Hloušek\Desktop

Název DQ200_MQB_Serie_MP2021

Volby pro tvorbu grundlage

Zvolte platformu

- PQ
- MQB
- MQB37w
- Všechny

Zvolte variantu

- Serie 2021
- Vorserie 2021
- VFF 2021
- MS4-LF 2021

Další nastavení

- Doplňovat stav z předchozí modelové péče
- Nevypisovat projekty bez softwarů

Úprava parametrové sady

U 22 LuJL 05 [redacted]

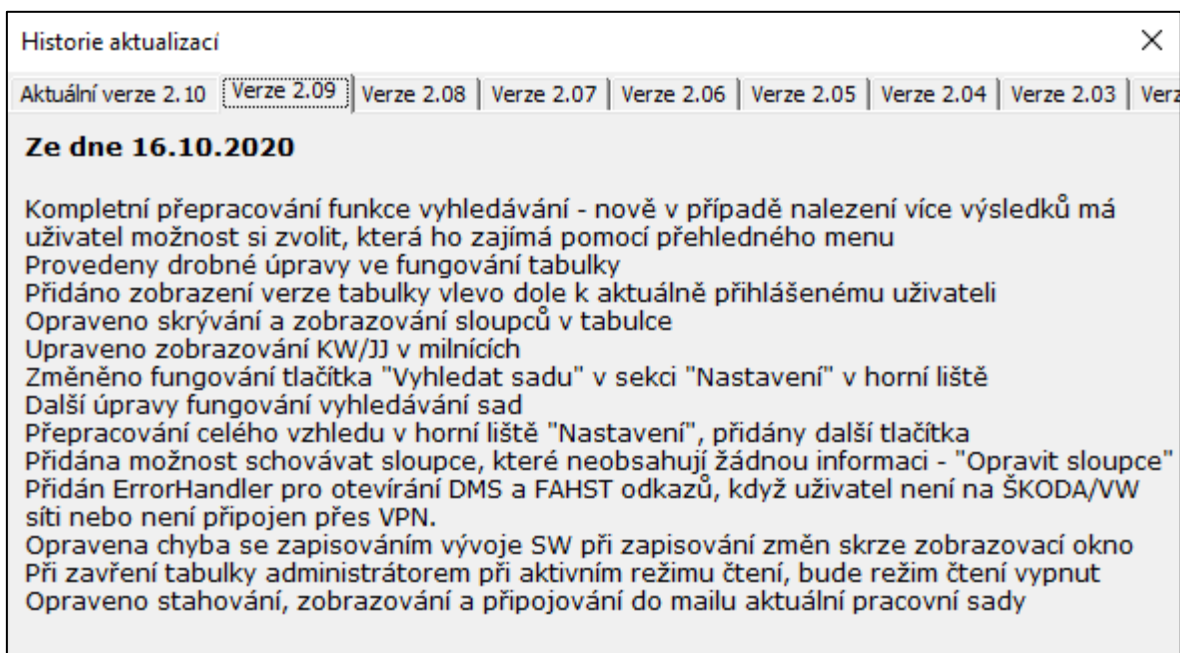
Náhled U22_LuJL_05_ [redacted]

Obr. 17 Zpracování Grundlage

Tabulka je vytvořena tak, že uživatel nepotřebuje vidět veškeré sloupce, které tabulka obsahuje. Víceméně postačí jen prvních deset sloupců, které jsou ukotveny. Ostatní sloupce jsou skryté. Jsou zde ale výjimky, které platí pro sloupce s názvy Serie, Vorserie, VFF, MS4-LF nebo Pracovní stav. Jakmile v daném sloupci je již vyplněn nějaký SW, tak sloupec by měl správně být odkrytý. Neděje se tomu tak ale automaticky. Po kliknutí tohoto tlačítka se všechny sloupce odkryjí a podle předem stanovených kritérií se opět postupně schovají. Celý proces trvá pár vteřin.

V tabulce v zobrazovacím okně je možné si zobrazit vysvětlivky emisních norem, které aktuálně jsou v některých z aktuálních projektů. Pokud by se však někdo chtěl podívat i na další emisní normy, které nejsou specificky vypsány pro některé z projektů, má možnost si je zobrazit pomocí tlačítka „Zobrazit Abgas normy“. Otevře se mu nový sešit, který je exportován z VW aplikace zvané TI-REX.

Předposlední tlačítko slouží k zobrazení okna „Historie aktualizací“ (viz Obr. 18), kde uživatel může zjistit, co se dělalo za poslední změny. Jedná se však čistě o informativní charakter.



Obr. 18 Historie aktualizací

Poslední tlačítko zobrazuje pouze dokument, který byl vytvořen jako návod na používání Přiřazovací tabulky SW. Jedná se hlavně o procesní návodku, jak pracovat s tabulkou nejen jako uživatel, ale také jako administrátor. Opět se jedná o jednu z výhod agilního vedení projektu, kdy není potřeba žádného pracovníka na zaučení, ale stačí mu pouze přečíst si danou návodku.

Speciální funkce administrátorů tabulky

Všechny výše uvedené funkce a možnosti tabulky jsou dostupné všem uživatelům tabulky. Existuje také několik funkcí, které slouží čistě administrátorům tabulky, jenž jim usnadňují práci.

V zobrazovacím okně je možné přidávat projekt stisknutím tlačítka „Přidat projekt“. Přidá se nový řádek v tabulce v listu „Aktuelle überzicht“ a je možné poté přidávat informace o něm skrze zobrazovací okno. Není tak vůbec potřeba se dívat do tabulky a dohledávat, kam jaká data zadat. Vše lze provést přímo v zobrazovacím okně.

Po otevření zobrazovacího okna, kliknutím na určitý projekt, je možné zvolit „Režim úprav“, čímž se umožní ve všech políčkách upravovat text.

Je nutné dbát na to, že v případě přidávání informací pro více oficiálních stavů (Serie, Vorserie, VFF, MS4-LF) je nutné vždy dát „Uložit změny“ mezi jednotlivými změnami oficiálního stavu.

Také je možné nastavit u pracovní sady, že se jedná o pracovní sadu „s upozorněním“. Zpravidla se zároveň přidává i informace o tom, proč se u ní dané upozornění aktivovalo, do políčka „Komentář ve sloupci B“, kde se tato informace uvede.

Administrátor má rovněž možnost v režimu editace projektu zvolit si jakoukoliv modelovou péči, která je aktuálně v přiřazovací tabulce vytvořena, není omezen pouze na ty, ve kterých daný projekt má již nějaký SW zadáný.

V případě, že nějaký projekt je již po EOP, je možné přes zobrazovací okno nechat aktuálně zobrazený projekt přesunout do listu „Nach EOP Projekte“, což je takový „archiv“ projektů. Funguje to také opačně, když se náhodou někdo přehlédne. Projekt může z archivu přesunout zpět mezi aktuální projekty. Pro zamezení náhodného kliknutí pro přesun, je vždy administrátor požádán pro potvrzení přesunutí aktuálního řádku do archivu/mezi aktuální projekty.

Možnost přidání modelové péče funguje pomocí listu „Základní údaje tabulky“, kde je oblast sloupců sloužících jako šablona pro vytvoření nové modelové péče. Administrátor pouze napíše, jak se má daná modelová péče jmenovat a podle toho se vytvoří do obou listů s projekty („Aktuelle überzicht“ a „Nach EOP Projekte“).

Aktualizace modelové péče provádí hned několik činností najednou. Pro začátek je možné si zvolit, zda se má při kontrole souborů provést i report, ve kterém je pro zvolenou modelovou péči vypsáno, zda pro každý projekt je k dispozici soubor PAR, ODX, EF-S/BMG, a to napříč všemi čtyřmi oficiálními stavy SW.

Proces aktualizace může trvat (v závislosti především na připojení) až 10 minut. Někdy však přes připojení skrze VPN to může trvat i více jak 30 minut. Tato doba trvání vzniká kvůli tomu, že pro každý projekt, který má nějaký z výše uvedených dokumentů zapsaný, se kontroluje v úložišti na týmových webech (FTP serveru), a to lze jen požadavkem na jejich „zobrazení“, samozřejmě pouze virtuálním požadavkem. Tudíž tento proces dotazování může být pro každý projekt a danou

modelovou péčí až 12krát, což při počtu více jak 100 projektů znamená 1200 dotazů na týmové weby (FTP server).

Při aktualizaci se tedy provádí kontrola souborů. V případě že v daném kontrolovaném poli není zapsán žádný text, proškrtně se. Zároveň se provádí zápis nejaktuálnějšího oficiálního stavu do sloupce modelové péče (master sloupce „Jahr“), který slouží jako identifikátor toho, že daný projekt je v dané modelové péči aktivní a také v případě stahování modelové péče při zvolení „Jahr“, který stahuje právě na základě tohoto sloupce.

V reportu vygenerovaném při aktualizaci modelové péče (viz Obr. 19), je vidět, že se pro každý projekt dohledávají informace pro všechny čtyři oficiální stavy SW. Pro každý SW se kontroluje PAR, ODX a EF-S/BMG dokumentace. Když v tabulce není doplněn žádný text, buňka se proškrtně a barva je oranžová. V případě, že je vyplněn název PAR a je dostupný ke stažení, bude zelený, jinak je červený. Pokud je vyplněn PAR, kontroluje se také ODX, ale jelikož by název byl stejný, vyplňuje se do daného políčka pouze „x“, když soubor není dostupný, nebo „i.O.“ (*in Ordnung*, z německého *v pořádku*). Kontrola EF-S/BMG souborů funguje stejně jako PAR. Cílem je, aby v reportu nebylo žádné políčko červené. Oranžová, z pohledu reportu, problém není.

Fahrzeug	Konzern-Projekt	Motor	PR.Nr. Abgas	MS4-LF			VFF			Vorserie			Serie		
				PAR	ODX	BMG	PAR	ODX	BMG	PAR	ODX	BMG	PAR	ODX	BMG
Fabia/Fabia Combi		1.0l 70kW EA211	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Polo/A1 Sportback/Rapid SB		1.5l 110kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Karoq/T-Roc/Ateca/Q2/T-roc C		1.5l 110kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Golf/Golf Variant/Golf Sportsvan		1.5l 110kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Karoq/Q2		1.0l 81kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Rapid SCALA SDO		1.0l 81kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Polo/biza/A1 Sportback		1.0l 81kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Arona/T-Cross/Rapid KAMIQ/C		1.0l 81kW EA211evo	4BI	-	-	-	FL_0CW	-	-	-	-	-	FL_0CW	-	x
Polo/A1 Sportback		1.0l 70kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW3	-	x
Passat/Passat Variant/Superb		1.5l 110kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW3	-	x
Arona/Rapid Crossover/City Ca		1.5l 110kW EA211evo	4BI	-	-	-	FL_0CW	-	-	-	-	-	FL_0CW3	-	x
Touran		1.5l 110kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW3	-	x
Passat/Passat Variant/Superb		2.0l TDI 90kW EA288evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW3	-	x
Karoq/Ateca/Q2		2.0l TDI 85kW EA288evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW3	-	x
T-Roc Cabrio		1.0l 81kW TFSI	4BI	-	-	-	-	-	-	-	-	-	FL_0CW3	-	x
Rapid SCALA SDO		1.0l 81kW TFSI	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Rapid SCALA SDO		1.0l 70kW TFSI	4BI	-	-	-	-	-	-	-	-	-	-	-	-
Golf/Golf Variant/A3 Sportback		1.5l 110kW EA211evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW	-	x
Golf/Golf Variant		1.5l 96kW	4BI	-	-	-	-	-	-	-	-	-	FL_0CW	-	x
Golf/Golf Variant/A3 Sportback		1.5l 96kW TGI EA211evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW	-	x
Golf/Golf Variant/A3 Sportback		2.0l 85kW TDI EA288evo	4BI	-	-	-	-	-	-	-	-	-	FL_0CW3	-	x
T-Cross LWB China		1.2l 65kW EA211 China	4BD	-	-	-	-	-	-	-	-	-	FL_0CW	-	x

Obr. 19 Ukázka reportu dostupnosti souborů pro určitou modelovou péči

EF-S/BMG dokumentace je možné aktualizovat tak, že administrátor zvolí složku, kterou má umístěnou, nejlépe, u sebe v počítači a podle ní se provede přiřazení těchto dokumentů k jednotlivým projektům – přiřazení se provádí na základě porovnání názvu oficiálního stavu SW a části názvu dokumentu.

Administrátor má také možnost nechat zaktualizovat komentář v buňce F1 na listu „Aktuelle überzicht“, který tam vypisuje informace o emisních normách podle tabulky v listu „Základní údaje tabulky“ (viz Obr. 10).

Podbarvení řádků se rozlišuje podle zapsané emisní normy. Barevné rozlišení se určuje v listu „Základní údaje tabulky“, podle které se pak provede podbarvení v obou listech („Aktuelle überzicht“ a „Nach EOP Projekte“).

List „History“, pro ukázkou vložen výstřížek (viz Obr. 20), slouží jen pro administrátory jako informace o tom, kdo kdy jakou hodnotu buňky změnil, kdy provedl aktualizaci modelové péče, nebo označil pracovní sadu na „s upozorněním“. Lze dle toho dohledat, kdo co změnil a případně jeho změnu napravit.

A	B	C	D	E	F
Datum a čas	Uživatel	List	Adres	Původní hodnota	Nová hodnota
16.10.2020 14:43	David Hloušek	Aktuelle überzicht	--	--	Provedena aktualizace modelové péče 2021
16.10.2020 15:05	David Hloušek	Aktuelle überzicht	---	---	Provedena aktualizace modelové péče 48/20
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	O108	27/20	22/17
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	P108	35/20	48/17
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	Q108	14/21	14/18
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	R108	26/21	22/18
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	S108	34/21	2/19
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	T108	43/21	14/19
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	U108		22/19
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	V108		31/19
26.11.2020 13:12	David Hloušek	Aktuelle überzicht	N108	9/20	09/20
26.11.2020 13:13	David Hloušek	Aktuelle überzicht	L108	22/22	
26.11.2020 13:13	David Hloušek	Aktuelle überzicht	M108	35/19	

Obr. 20 Ukázka listu „History“

V listu „Základní údaje tabulky“ je obsažena, jak již bylo zmíněno, šablona pro tvorbu nové modelové péče, podbarvení řádků dle emisních norem, ale mimo to také tabulka s uvedenými uživateli s administrátorskými právy, odkud se zjišťuje, kdo má a kdo nemá v tabulce vyšší práva.

Druhou tabulkou je jen uložení některých proměnných do fyzické hodnoty buňky, aby se nezatěžovala mezipaměť počítače. Takto se totiž hodnota zapíše a místo vyvolávání proměnné se jen načte hodnota z buňky. Na obrázku 21 je ukázka toho, jak list „Základní údaje tabulky“ vypadá. Vpravo je navíc také seznam sloupců s názvy, který se využívá v případě generování nové modelové péče (na obrázku 21 tyto sloupce nejsou vidět).

Obr. 22 Úprava názvů aktuálních pracovních sad

V příloze 4 je ukázáno, jak funguje přejmenovávání pracovních sad, aby se nic nepokazilo. Detekce správné platformy probíhá pomocí prvního sloupce, kde se dohledává, jestli v názvu platformy je „w“ nebo není. Poté se u sady pro MQB stav musí zjistit, zda se jedná o variantu s „UxxExx“ nebo pouze „Uxx“. Název se poté přepíše dle zadaných hodnot (FirstNumbers a SecondNumbers).

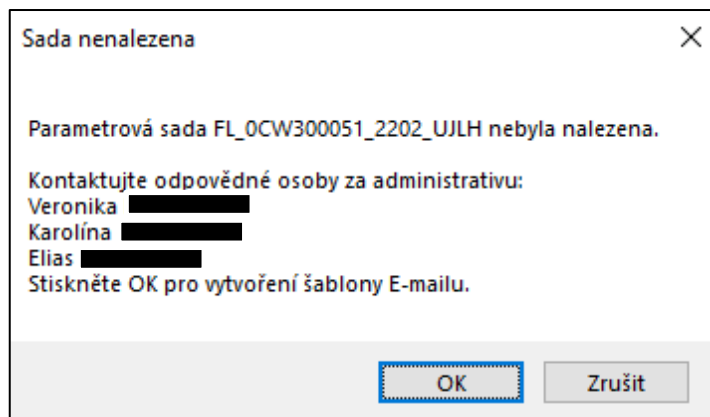
Existovala dříve ještě třetí varianta platformy (PQ), která již zaniká, ale i tak bylo nutné, aby kontrola s tímto názvem také počítala a vynechávala jej.

Další funkcionality tabulky

Jak již bylo zmíněno, tak každý projekt mimo oficiálního stavu SW má také pracovní sady. Tyto sady mají různé generace a je nutné je také stahovat. Tyto sady je možné stahovat jako balík až stovek souborů. Tento balík obsahuje i sady, které nejsou potřebné, a proto tabulka umí z tohoto balíku překopírovat do nové složky jen ty soubory, které jsou relevantní k projektům obsažených v tabulce. VBA toto zpracuje pomocí vyhledávání postupně každého souboru ve složce a porovnává, jestli existuje někde v tabulce. Samozřejmě je nutné změnit číslo generace na proměnnou, jelikož v tabulce nikdy nebude stejné označení generace, jako je tomu ve stažené složce. Pracovní sada vypadá například takto „U19_OJRC_05_VW370_15I_TFSI...“, ve které číslo „05“ označuje generaci.

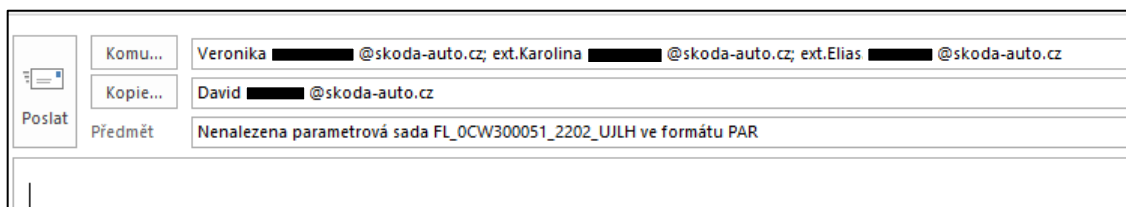
Předpokládá se, že VBA může mít nějakou chybu při stahování souborů. Pokud tedy nastane situace, že při otevření zobrazovacího okna s náhledem projektu se

vyhodnotí soubor jako dostupný a pak při pokusu o jeho stažení najednou žádost o stažení souboru pouze vrátí chybu, že soubor nebyl nalezen, tak v takovém případě se vyvolá funkce „ContactAdministrator“ (viz Obr. 23), kde se uživateli zobrazí okno s informací, koho má případně kontaktovat.



Obr. 23 Kontaktování administrátora v případě nenalezené sady

V případě, že zvolí možnost kontaktovat administrátora (stiskne „OK“), tak se mu předvyplní mail s kontaktem na administrátory a v předmětu mailu bude název nenalezené sady (viz Obr. 24). V kopii je vždy autor Přiřazovací tabulky SW, jakožto odpovědný za fungování tabulky, aby mohl prověřit, zda se nejedná o chybu na straně VBA.



Obr. 24 Ukázka mailu, který se vygeneruje pro kontaktování administrátorů

Uživatel má možnost využívat vkládání stahovaných souborů do zipu. Aktuálně se tato funkce využívá především u hromadného stahování oficiálních stavů SW a jedná se vlastně o využití zabudované aplikace Windows, kde tento VBA kód je v příloze 5 ukázán.

Mezi poslední funkce tabulky patří poněkud užitečná vlastnost. Některé procesy, u kterých jejich průběh trvá v řádech desítek minut (aktualizace nebo stahování modelové péče, ...), na konci informují o dokončení pomocí hlasového oznámení, které je dostupné pouze v anglickém jazyce.

4.3 Situace po zavedení změn

Se zavedením Přiřazovací tabulky SW a implementací všech výše uvedených funkcionalit, bylo umožněno efektivnější a agilní řízení projektu. Aplikátoři nyní dokážou pouhým otevřením jedné jediné tabulky a dohledáním projektu okamžitě zjistit všechny potřebné informace a v případě dokumentace nebo SW si je rovnou stahovat do počítače, připojovat do E-mailu nebo otevřít k prohlížení. Není tedy pro ně již nutné pravidelně pro získání informací otevírat několik programů. Otevřou jen jednu tabulku a tam najdou vše, žádné zdržování při otevírání jiných programů.

Vznikl také tým lidí, který se stará o doplňování informací do této tabulky (administrátoři) a proti nim jsou uživatelé tabulky (aplikátoři). Tabulku mohou používat nezávisle na tom, zda někdo z administrátorů je zrovna na příjmu, což je při této práci důležité, protože se může stát, že aplikátor bude na druhé straně zeměkoule zrovna potřebovat zjistit nějaké informace, ale v České republice budou všichni spát.

Nespornou výhodou jsou také funkcionality jako je hromadné zjištění chybějících souborů, což administrátorům umožňuje udržovat si přehled o tom, jaké soubory musí stáhnout a doplnit do jejich vlastní databáze souborů, odkud si je získává přiřazovací tabulka. Dále export do speciálních formátů používajících se při různých rozhovorech s vedoucími projektu, možnost vytvoření reportu stavu milníků a SOP a mnoho dalšího. Všechny tyto funkce usnadňují práci a zajišťují menší chybovost z důvodu lidského faktoru.

Přiřazovací tabulka SW jako taková má VBA postaveno na principu „Poka Yoke“, což znamená, že uživatel nebo administrátor nemůže v tabulce udělat nic špatně a všechna tlačítka a textová pole se odemykají a zamykají k editaci dle příslušných zvolených parametrů vyhledávání, prohlížení, a tak dále. Z pohledu zajištění co nejmenší chybovosti v programování VBA je využíváno především informativního zobrazování pomocí *MsgBox*. Pro ilustraci je na obrázku 25 představen princip, dle kterého je ihned detekován problém. Předpokladem je, že *i* může nabývat hodnot od 1 do 3.

<pre>Sub Detekce_chyby() If i = 1 Then 'Při kontrole se zjistí, že i = 1 ElseIf i = 2 Then 'Při kontrole se zjistí, že i = 2 Else 'Při kontrole se zjistí, že i není 1 nebo End If End Sub</pre>	<pre>Sub Detekce_chyby() If i = 1 Then 'Při kontrole se zjistí, že i = 1 ElseIf i = 2 Then 'Při kontrole se zjistí, že i = 2 ElseIf i = 3 Then 'Při kontrole se zjistí, že i = 3 Else 'Při kontrole se zjistí, že i není ani 1, 2 a 3 MsgBox "Nastal problém, i vyšlo mimo povolený rozsah", vbCritical End If End Sub</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Obr. 25 Princip detekce problému

V levé části je viditelné, že funkce dokáže pojmout jakoukoliv hodnotu *i*. V pravé části je vidět, že v případě, že *i* bude mimo povolený rozsah, objeví se varovné hlášení, které upozorní na tuto skutečnost.

Celé VBA je postaveno na tomto principu, aby tak bylo možné předejít co největšímu množství chyb, tedy nikde nemůže nastat chyba zpracování bez upozornění, že není něco v pořádku.

4.4 Dodržovaná pravidla při tvorbě VBA

Tato podkapitola je o pravidlech, které je dobré při programování dodržovat a také se dodržovaly při vývoji Přiřazovací tabulky SW. Díky dodržování těchto „pravidel“ bylo snazší a rychlejší provádění úprav, oprav a přidávání nových funkcí.

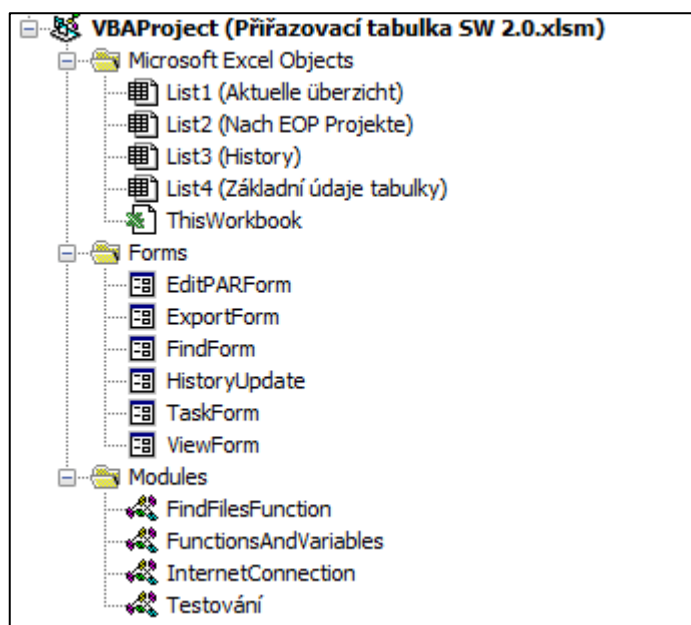
4.4.1 Formátování kódu

Jedním z nejzákladnějších pravidel, které je vhodné dodržovat, je formátování kódu, jinými slovy určit si, že každá funkce bude odsakována tabulátorem. V příloze 6 je příklad toho, jakým způsobem je formátován kód v přiřazovací tabulce. Každá funkce je odskočená tabulátorem a každá v ní vnořená dalším tabulátorem. Pokud se jedná pouze o řádek kódu, tak jen o jednu mezeru. Díky tomu je pak přehlednější prohlížení a je snadnější zpětně najít místo, kde je potřeba provést případnou změnu. Pomocí korektního a svědomitého formátování je pak snadno vidět, které části je možné sjednotit a použít případně *For – Next Loop*.

4.4.2 Jak efektivně organizovat VBA projekt

Organizace kódu je velmi důležitá, stejně tak jako formátování kódu. Nezáleží ani tak na rozsahu VBA, protože přidáváním a vylepšováním funkcionalit může nastat nutnost používat stejné série příkazů v různých procesech. Přesně z takového

důvodu je lepší všeobecně použitelné funkce přidávat do odděleného modulu. V případě přiřazovací tabulky se jedná o *FunctionsAndVariables* modul, kde jsou umístěny všechny globálně používané funkce. Touto metodou je také možné docílit přehlednějšího kódu a snadnější orientace v něm. Zobecňováním a vytvářením univerzálních funkcí tak může vzniknout snadno spravovatelný kód.



Obr. 26 Organizace VBA projektu

Na obrázku 26 je vidět rozvržení kódu v Přiřazovací tabulce SW. První část tvoří samotné objekty, kde se v tomto případě jedná pouze o jednotlivé listy souboru a jeden globální, který je vhodné využívat na procesy používané při otevření a zavření celého dokumentu. V druhé části jsou tzv. „Forms“, v překladu formuláře. Konkrétně se jedná u této tabulky o zobrazovací okno, okno hromadných úprav a další. Třetí část tvoří moduly, které se využívají především pro globálně používané procesy. Jsou rozděleny na samotný modul pro vyhledávání souborů (*FindFilesFunctions*), jelikož je velmi obsáhlý a není důvod s ním nijak manipulovat. Další částí je již zmiňovaný modul *FunctionsAndVariables* a třetí částí je funkce zjišťující, zda je zařízení připojené k internetu (*InternetConnection*). Poslední modul je určen pouze pro testování, tedy není to nijak formátovaný kód, ale jen sled několika možných příkazů.

Pro ukázkou odvolávání se na *FunctionsAndVariables* modul je v příloze 7 ukázána funkce pro stahování souborů, kde v případě chyby se odkazuje

na *ContactAdministrator*. Tato funkce se využívá v mnoha Sub procesech, a proto je umístěna do globálního modulu.

4.4.3 Efektivní používání proměnných

Využívání proměnných je jednoznačně součástí každého VBA. Existují tři základní typy – Dim, Public a Global. Dim není možné sdílet mezi různými Sub procesy, ani moduly. Znamená to, že pokud se proměnná definuje v jednom Sub procesu a v druhém pod úplně stejným názvem, tak navzájem nebudou nijak provázány. Nejvíce jsou tedy využívány v případě, že pod tímto názvem není nutné něco opakovaně používat.

Druhou možností proměnné je Public. Public znamená, že v rámci všech Sub procesů se již nedefinuje a hodnota proměnné zůstává použitelná ve všech Sub procesech, ale pouze v rámci jedné části VBA projektu.

Poslední možností je Global. Tato proměnná může nabýt hodnot v rámci různých částí VBA projektu a bude dostupná ze všech modulů a Sub procesů. Globální proměnné je možné zapisovat pouze do části VBA projektu „Modules“. Takto také vznikl název části *FunctionsAndVariables*, kde krom globálně používaných funkcí jsou také globální proměnné.

V rámci ušetření dalších několika řádků kódu a zajištění si větší přehlednosti, jsou desítky proměnných nastaveny jako globální. Má to také ale svá úskalí. Musí se brát v potaz, že pokud se takto definují proměnné a bude chtěné je použít úplně „nanovo“, je nutné nezapomenout je před samotným procesem „vyprázdnit“ nebo vynulovat.

V návaznosti na přílohu 7 (Odkazování na globálně používaný modul), kde se odkazuje funkce na jednu globální, je v příloze 8 vidět vyvolaná funkce *ContactAdministrator*, ve které se již samotná funkce provede a pak se vrací zpět do vyvolávajícího Sub procesu.

4.4.4 Přidávání popisných řádků a textů

Proč je dobré si přidávat popisné řádky? Zapomenutí, proč to tam je. Snadno lze najít část, kterou je potřeba vyhledat – určitá část kódu, která provádí nějakou činnost.

Ve VBA je možné si přidávat popisné řádky pouhým znakem „““, vše, co se vyskytuje za ním, se již nepovažuje za část aktivního kódu. Využívat popisné řádky je základ, protože ať už se jedná o takto rozsáhlý kód nebo nějaký zprvu vypadají jako nijak složitý a nerozvinutý proces, je to vždy vhodné, protože když je potřeba se k tomu vrátit a něco v daném procesu upravit nebo k němu přidat, tak orientace kde co vlastně je a k čemu to slouží, znatelně urychlí zpracování. Zároveň když je potřeba nějakou část, třeba jen jeden řádek, dočasně vypnout pro kontrolu, zjištění něčeho, tak právě stačí daný řádek přeměnit na popisný.

Jako velkou výhodou, která v těchto popisných řádcích vězí, je možnost to pak využívat jako názvy. Když se každá část kódu nazve specificky podle činnosti, kterou vykonává, tak pak je možné snadno pomocí vyhledávání určitého názvu prakticky okamžitě dohledat, kde se nachází právě ta část kódu provádějící daný úkon. V příloze 9 je část kódu, který se využívá pro provádění aktualizací modelové péče. V této části je konkrétně proces k aktualizaci sloupce s vývojem SW a ověřováním EF-S/BMG dokumentace.

V případě, že nebude fungovat nějaká část aktualizace, tak dle pouhého kouknutí se na kód určitě nelze snadno a rychle poznat, kam přesně se podívat. Díky popisným řádkům je zřetelné, kde každá část začíná a lze efektivněji danou část upravit. Zároveň tam je také vidět „zakomentovanou“ část kódu, která se pravděpodobně dříve používala a pak se zjistilo, že není nutná, ale v případě, že to bude nakonec nutné používat, stačí pouze odstranit uvozující znak, a to je celé.

4.5 Porovnání agilního způsobu řízení s/bez pomoci VBA for Excel

Před implementací Přiřazovací tabulky SW, ve které se hojně využívá VBA, musel uživatel jednotlivé informace o projektech zjišťovat sám z několika různých zdrojů a případně také volat/psát jiným, kteří mu budou schopni poradit nebo doplnit některé informace. Byl tedy vždy vázaný na někoho dalšího, kdo by mu byl schopen pomoci a musel mít přístup do několika programů.

Situace po implementaci změny

Přiřazovací tabulka umožňuje ušetřit několik hodin času při pravidelném dohledávání informací, shlukuje všechny projekty a informace o nich do jednoho dokumentu, dokáže s danými daty efektivně pracovat a hlídat určité termíny.

Uživatel, v tomto případě tedy aplikátor, ušetří značné množství času, jelikož nemusí informace dohledávat v různých programech, které se také značnou dobu zapínají a je v nich nutné složitě dohledávat projekt podle několika kritérií, a musel by také hlídat, zda do všech potřebných programů a systémů má patřičné přístupy.

Ačkoliv se zavedením přiřazovací tabulky bylo nutné vytvořit skupinu administrátorů, kteří se starají o zadávání dat do této tabulky, tak stále to přináší větší přínos, protože tito lidé mají na starosti „pouze“ tuto činnost a lépe se orientují kde jaké informace získat a v případě chybějících informací vědí, na koho se obrátit. Zároveň tyto informace využívají pro další proces spojený s řízením projektů a tabulka jim umožňuje provádět reporty, které pro tyto situace mohou využít, čímž se jim celý proces usnadní a zrychlí.

Aplikátor také ke zjišťování informací nepotřebuje mít administrátora k dispozici, protože vše lze vyřešit pomocí prostředníka, tedy tabulky, která pomocí VBA hlídá, aby aplikátor neudělal nějakou chybu a vždy mu ukázala relevantní informace, které ho v danou situaci zajímají. V případě, že nějaká chyba nastane, je velmi jednoduché kontaktovat E-mailem administrátory, kde tabulka sama připraví šablonu mailu, kterou stačí pouze dát odeslat a administrátoři budou hned vědět, na co se mají zaměřit.

Další kroky ke zlepšení předávání informací offline metodou

Přiřazovací tabulka SW stále nabízí prostory ke zlepšování. Vzhledem k tomu, že její vývoj trval několik měsíců a postupně v průběhu se stále vylepšovala, neuškodilo by přepracovat uživatelské rozhraní pro ještě snazší a intuitivnější ovládání. Jedním z dalších kroků bude určitě rozšíření o další nadstavby, čímž je myšleno především širší provázání se systémy, které se používají pro získávání dat, a to na takové úrovni, kdy tabulka bude automaticky data získávat a nebude nutný přístup administrátor.

Třetím důležitým krokem také bude lepší forma exportu dat. Na úplném začátku totiž tabulka obsahovala u každého projektu jen velmi omezený počet informací a tato data se dále využívala pro další komunikaci s partnery z různých oddělení (nejen z VW). Postupem času se těchto informací začalo získávat více, jednoduchý export by tak byl velmi vhodným pomocníkem pro běžnou práci.

Závěr

Hlavním cílem práce bylo nalézt úzká místa při předávání projektových informací v globálních projektech pomocí agilních metod a tato úzká místa eliminovat. Konkrétně se jedná o oddělení vývoje automatických převodovek v ŠA. Navržený způsob pro eliminaci úzkých míst byl následně porovnán se situací před zavedením a po zavedení návrhu.

Teoretická část byla věnována projektovému managementu. Byl představen tradiční a agilní způsob vedení projektu, definovány a porovnány jejich výhody a nevýhody a uvedeny typy projektů vhodné k jejich aplikaci. Druhá kapitola byla věnována programovacímu jazyku VBA a jeho možným aplikacím.

V praktické části byla krátce představena společnost ŠA a oddělení EPG/3, kterého se tato práce a navrhované řešení týká. Poté byla popsána situace před zavedením návrhu, kde bylo zjištěno, že úzkým místem je složitost získávání informací o jednotlivých projektech a nutnost zjišťovat je od dalších členů týmu, včetně složitého dohledávání v několika dalších systémech, které celý proces zpracování dat prodlužují.

Návrh na zlepšení zahrnuje tabulku, která byla posléze nazvána Přiřazovací tabulka SW. Jedná se o podrobný seznam s informacemi o jednotlivých projektech s využitím komplexní implementace VBA pro automatizaci a snazší správu projektů a na ně navázané informace. Uživatel nově nepotřebuje dohledávat informace na několika různých místech, ale využije zmíněnou tabulku, v níž nalezne veškeré informace. Při práci s tabulkou se zároveň automaticky hlídají termíny jednotlivých milníků, lze tvořit reporty, z nichž je zcela jasné, zda jsou ke každému projektu dostupné všechny informace a nechybí žádné dokumenty.

Se zavedením tohoto návrhu bylo nutné definovat administrátora, který zodpovídá za doplňování informací k jednotlivým projektům. Navržená aplikace mu zároveň slouží jako nástroj pro snazší tvorbu další dokumentace, kterou je potřeba provádět. Administrátor může navíc generovat reporty, které mu výrazně usnadní kontrolu projektů.

Vyhodnocení aplikace bylo provedeno na základě zpětné vazby od uživatelů, kteří se vyjádřili k provedeným změnám. Dotazováním bylo zjištěno, že největším přínosem aplikace je dostupnost dat v rámci jediného zdroje bez nutnosti zjišťovat

informace na několika místech. Další předností je možnost pracovat s daty i v režimu off-line, tedy bez přístupu k internetu. Uživatelé tak budou mít stále dostupné všechny informace, což dříve nebylo možné.

V průběhu navrhování probíhalo několik změn, které se flexibilně řešily až při fázi implementace a programování. Každý návrh je možné stále zlepšovat, proto je nutné neustále vyhodnocovat, analyzovat a navrhovat zdokonalení aplikace.

Seznam literatury

ANDREU, Fernando, 2021. *An overhauled fork of the original Custom UI Editor for Microsoft Office, built with WPF* [online] [vid. 2021-03-27]. Dostupné z: <https://github.com/fernandreu/office-ribbonx-editor>

AXELOS.COM, 2021. *What is PRINCE2? | Methodology & Qualification | AXELOS* [online] [vid. 2021-03-11]. Dostupné z: <https://www.axelos.com/best-practice-solutions/prince2/what-is-prince2>

CARR, Kira, 2020. *Agile Project Management Vs. Traditional Project Management* [online] [vid. 2021-03-12]. Dostupné z: <https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management>

DEPAIRE, B., 2019. *Lecture Notes for Project Management* [online] [vid. 2021-03-11]. Dostupné z: <https://bookdown.org/content/1785/traditional-project-management.html>

KANBANIZE.COM, 2021. *What Is Agile Project Management? A Comprehensive Guide* [online] [vid. 2021-03-11]. Dostupné z: <https://kanbanize.com/agile/project-management>

KASHYAP, Sadeep, 2021. *Traditional vs Agile: Which Methodology is Right for Your Projects?* [online] [vid. 2021-03-12]. Dostupné z: <https://www.proofhub.com/articles/traditional-vs-agile-project-management>

KNOWLEDGEHUT.COM, 2021. *Traditional Project Management* [online] [vid. 2021-03-11]. Dostupné z: <https://www.knowledgehut.com/tutorials/project-management/traditional-project-management>

LAYTON, Mark C., 2017. *Agile Project Management For Dummies, 2nd Edition*. ISBN 978-1-119-40569-6.

NICHOLAS, John M. a Herman STEYN, 2017. *Project Management for Engineering, Business and Technology* [online]. Dostupné z: doi:10.4324/9781315676319

PMI.ORG, 2020. *COVID-19 Crisis—Impact on Project Business | PMI* [online] [vid. 2021-03-12]. Dostupné z: <https://www.pmi.org/learning/library/covid-19-impact-project-business-12335>

PRINCE2.COM, 2021. *What Is PRINCE2 - PRINCE2 Qualification Explained | EUR* [online] [vid. 2021-03-11]. Dostupné z: <https://www.prince2.com/eur/what-is-prince2>

PROJECTMANAGER.COM, 2021. *Gantt Chart: The Ultimate Guide (with Examples) - ProjectManager.com* [online] [vid. 2021-03-07]. Dostupné z: <https://www.projectmanager.com/gantt-chart>

PROJEKTOVĚ.CZ, 2021. *Ganttův diagram | Projektově.CZ* [online] [vid. 2021-03-07]. Dostupné z: <https://www.projektove.cz/vlastnosti/ganttuv-diagram>

ŠKODA AUTO A.S., 2020a. *Kde všude je ŠKODA AUTO doma - ŠKODA Storyboard* [online] [vid. 2021-03-27]. Dostupné z: <https://www.skoda-storyboard.com/cs/modely/cs/kde-vsude-je-skoda-auto-doma/>

ŠKODA AUTO A.S., 2020b. *ŠKODA AUTO výroční zpráva 2019* [online]. [vid. 2020-12-05]. Dostupné z: https://cdn.skoda-storyboard.com/2020/06/SKODA_2019_CZE.pdf

SVOZILOVÁ, Alena, 2016. *Projektový management: systémový přístup k řízení projektů*. Praha: Expert (Grada). ISBN 978-80-271-0075-0.

TORODE, Christina, 2020. *Agile Project Management* [online] [vid. 2021-03-11]. Dostupné z: <https://searchcio.techtarget.com/definition/Agile-project-management>

VIVIFYSCRUM.COM, 2019. *Agile Project Management vs. Traditional Project Management | VivifyScrum* [online] [vid. 2021-03-12]. Dostupné z: <https://www.vivifyscrum.com/insights/agile-project-management-vs-traditional-project-management>

WALKENBACH, John, 2013. *Excel® 2013 Power Programming with VBA* [online]. Hoboken, NJ, USA: Wiley Publishing, Inc. ISBN 978-1-118-49039-6. Dostupné z: doi:10.1002/9781118257616

WHITE, Karen R. J., 2008. *Agile project management* [online] [vid. 2021-03-11]. Dostupné z: <https://www.pmi.org/learning/library/agile-project-management-mandate-changing-requirements-7043>

WORKFRONT.COM, 2021. *Agile Project Management - A Beginner's Guide | Workfront* [online] [vid. 2021-03-11]. Dostupné z: <https://www.workfront.com/project-management/methodologies/agile>

WRIKE.COM, 2021. *What is PMBOK in Project Management? Understanding PMBOK Methodology* [online] [vid. 2021-03-11]. Dostupné z: <https://www.wrike.com/project-management-guide/faq/what-is-pmbok-in-project-management/>

Seznam obrázků a tabulek

Seznam obrázků

Obr. 1 Zobrazení fází projektu.....	14
Obr. 2 Ukázka Ganttova diagramu	16
Obr. 3 Ukázka kanbanové tabule	27
Obr. 4 Výroba jednotlivých modelů ŠKODA ve světě.....	37
Obr. 5 Procesy prováděné na pozadí přiřazovací tabulky	40
Obr. 6 Informace o poslední změně údajů	44
Obr. 7 Verze VBA a kdo je aktuálně přihlášen	44
Obr. 8 Zobrazovací okno – náhled projektu.....	45
Obr. 9 Zobrazení milníků projektů	47
Obr. 10 Ukázka tabulky pro emisní normy.....	48
Obr. 11 Zpracování úkolu – hromadné úpravy	49
Obr. 12 Výpis SOP všech projektů	49
Obr. 13 Okno vyhledávání sady	50
Obr. 14 Záložka „Nastavení“	51
Obr. 15 Ukázka XML kódu pro vlastní záložku „Nastavení“	51
Obr. 16 Ovládání vytvořeného tlačítka skrze Office Ribbon X Editor	52
Obr. 17 Zpracování Grundlage.....	53
Obr. 18 Historie aktualizací	54
Obr. 19 Ukázka reportu dostupnosti souborů pro určitou modelovou péči	56
Obr. 20 Ukázka listu „History“.....	57
Obr. 21 Ukázka listu „Základní údaje tabulky“	58
Obr. 22 Úprava názvů aktuálních pracovních sad.....	59
Obr. 23 Kontaktování administrátora v případě nenalezené sady.....	60
Obr. 24 Ukázka mailu, který se vygeneruje pro kontaktování administrátorů.....	60

Obr. 25 Princip detekce problému	62
Obr. 26 Organizace VBA projektu	63

Seznam tabulek

Tab. 1 Statistika Přirázovací tabulky SW	42
------------------------------------------------	----

Seznam příloh

Příloha 1 Ukázka funkce RecalculateMS	75
Příloha 2 Ukázka funkce ShowButton_Click.....	76
Příloha 3 Ukázka funkce CheckConnection	77
Příloha 4 Ukázka funkce korektního přejmenování aktuálních pracovních sad	78
Příloha 5 Ukázka funkce zazipování složky pomocí VBA.....	79
Příloha 6 Formátování kódu	80
Příloha 7 Odkazování na globálně používaný modul	81
Příloha 8 Ukázka globálně používané funkce.....	82
Příloha 9 Popisné řádky	83

Příloha 1 Ukázka funkce RecalculateMS

```
Sub RecalculateMS()  
Dim ISOWEEK As Integer  
Dim Week1 As Integer  
Dim Week2 As Integer  
Dim ISOYEAR As Integer  
Dim Year1 As Integer  
Dim Year2 As Integer  
  
ISOWEEK = DatePart("ww", myDate, vbMonday, vbFirstFourDays)  
Week1 = DatePart("ww", myDate, vbMonday, vbFirstFourDays)  
Week2 = DatePart("ww", DateAdd("d", 7, myDate), vbMonday, vbFirstFourDays)  
ISOYEAR = DatePart("yyyy", myDate, vbMonday, vbFirstFourDays)  
Year1 = DatePart("yyyy", myDate, vbMonday, vbFirstFourDays)  
Year2 = DatePart("yyyy", DateAdd("d", 7, myDate), vbMonday, vbFirstFourDays)  
  
If ISOWEEK = 53 And DatePart("ww", DateAdd("d", 7, myDate), vbMonday, vbFirstFourDays) = 2 Then  
    ISOWEEK = 1  
End If  
If ISOWEEK = 1 And DatePart("yyyy", DateAdd("d", 7, myDate), vbMonday, vbFirstFourDays) > ISOYEAR Then  
    ISOYEAR = ISOYEAR + 1  
End If  
If Len(ISOWEEK) = 1 Then  
    ISOWEEK = "0" & ISOWEEK  
End If  
  
MSDate = ISOWEEK & "/" & Right(ISOYEAR, 2)  
  
End Sub
```

Příloha 2 Ukázka funkce ShowButton_Click

```
Sub ShowButton_Click()  
Dim FileToOpen  
ControlCount = 0  
lFilesNum = 0  
  
If InternetDownload = True Then  
ChosenFile = "Show"  
FolderType = "PAR"  
FormatType = ".par"  
PARName = SWName  
FunctionsAndVariables.DownloadFile  
If FileFound = False Then  
GoTo NotFound  
Else  
FileToOpen = Shell("C:\WINDOWS\notepad.exe " & OnlinePath, 1)  
Exit Sub  
End If  
Else  
FindedPAR = FindFiles(PARPath & "\PAR", strMyFiles, lFilesNum, "*" & SWName & "*" & ".par", False, True)  
If ControlCount = lFilesNum Or FindedPAR = False Then  
GoTo NotFound  
Else  
FileToOpen = Shell("C:\WINDOWS\notepad.exe " & strMyFiles(lFilesNum).Path & strMyFiles(lFilesNum).Name, 1)  
Exit Sub  
End If  
End If  
  
NotFound:  
FunctionsAndVariables.ContactAdministrator  
End Sub
```

Příloha 3 Ukázka funkce CheckConnection

```
Sub CheckConnection()
If InternetDownload = True Then
  If IsInternetConnected = False Then
    ConnectionStatus.BackColor = RGB(255, 0, 0)
    ConnectionStatus.Caption = "n.i.O."
    OfflineMode = True
    If PreStatusConnection = "" Then
      MsgBox "Byl aktivován offline mód, protože není možné se připojit ke ŠKODA AUTO síti."
    End If
  Else
    ConnectionStatus.BackColor = RGB(0, 204, 0)
    ConnectionStatus.Caption = "i.O."
  End If
Else
  If CreateObject("Scripting.FileSystemObject").Drives("G").IsReady = False Then
    ConnectionStatus.BackColor = RGB(255, 0, 0)
    ConnectionStatus.Caption = "n.i.O."
    OfflineMode = True
    If PreStatusConnection = "" Then
      MsgBox "Byl aktivován offline mód, protože není možné se připojit ke sdílenému disku."
    End If
  Else
    ConnectionStatus.BackColor = RGB(0, 204, 0)
    ConnectionStatus.Caption = "i.O."
  End If
End If
End Sub
```

Příloha 4 Ukázka funkce korektního přejmenovávání aktuálních pracovních sad

```
With ThisWorkbook.ActiveSheet
LastRow = .Cells(Rows.Count, 1).End(xlUp).Row
For i = 2 To LastRow
    If Not .Cells(i, 9).Value = "" Then
        If InStr(1, LCase(.Cells(i, 1).Value), "w") > 0 Then
            'Jedná se o MQBw
            .Cells(i, 9).Value = Left(.Cells(i, 9), 6) & FirstNumbers37w & "T" & SecondNumbers37w & Right(.Cells(i, 9), Len(.Cells(i, 9)) - 11)
        ElseIf InStr(1, LCase(.Cells(i, 1).Value), "p") > 0 Then
            'Jedná se o PQ
            PQFounded = True
            GoTo NextI
        Else
            'Jedná se o klasika
            If LCase(Mid(.Cells(i, 9), 4, 1)) = "e" Then
                DetectEXX = 7
            Else
                DetectEXX = 4
            End If
            .Cells(i, 9).Value = "U" & FirstNumbers & Mid(.Cells(i, 9), DetectEXX, 6) & SecondNumbers & Right(.Cells(i, 9), Len(.Cells(i, 9)) - DetectEXX - 7)
        End If
    End If
NextI:
Next i
End With
```

Příloha 5 Ukázka funkce zazipování složky pomocí VBA

```
Public Sub CreateZipFile(folderToZipPath As Variant, zippedFileFullName As Variant)
Dim ShellApp As Object

'Create an empty zip file
Open zippedFileFullName For Output As #1
Print #1, Chr$(80) & Chr$(75) & Chr$(5) & Chr$(6) & String(18, 0)
Close #1

'Copy the files & folders into the zip file
Set ShellApp = CreateObject("Shell.Application")
ShellApp.Namespace(zippedFileFullName).CopyHere ShellApp.Namespace(folderToZipPath).items

'Zipping the files may take a while, create loop to pause the macro until zipping has finished.
On Error Resume Next
Do Until ShellApp.Namespace(zippedFileFullName).items.Count = ShellApp.Namespace(folderToZipPath).items.Count
Application.Wait (Now + TimeValue("0:00:01"))
Loop
On Error GoTo 0

End Sub
```

Příloha 6 Formátování kódu

```
If Len(TextBoxSOP) = 5 And IsNumeric(Left(TextBoxSOP, 2)) = True And IsNumeric(Right(TextBoxSOP, 2)) = True And Not TextBoxMS1.text = "" Then
myDate = DateAdd("ww", Left(TextBoxSOP, 2) - 1, DateSerial(20 & Right(TextBoxSOP, 2), 1, -1))
TextBoxSOP.ControlTipText = "~ " & myDate + 2
For i = 1 To 7
    If Not Me.Controls("TextBoxMS" & i).Value = "" Then
        If Instr(1, Mid(Me.Controls("TextBoxMS" & i), 2, 1), "/") = 1 Then
            myDate = DateAdd("ww", Left(Me.Controls("TextBoxMS" & i), 1) - 1, DateSerial(20 & Right(Me.Controls("TextBoxMS" & i), 2), 1, -1))
        Else
            myDate = DateAdd("ww", Left(Me.Controls("TextBoxMS" & i), 2) - 1, DateSerial(20 & Right(Me.Controls("TextBoxMS" & i), 2), 1, -1))
        End If
        Me.Controls("TextBoxMS" & i).ControlTipText = "~ " & myDate + 2
    End If
Next i
If Not TextBoxPVS.Value = "" Then
    If Instr(1, Mid(TextBoxPVS, 2, 1), "/") = 1 Then
        myDate = DateAdd("ww", Left(TextBoxPVS, 1) - 1, DateSerial(20 & Right(TextBoxPVS, 2), 1, -1))
    Else
        myDate = DateAdd("ww", Left(TextBoxPVS, 2) - 1, DateSerial(20 & Right(TextBoxPVS, 2), 1, -1))
    End If
End If
End If
```


Příloha 7 Odkazování na globálně používaný modul

```
Sub DownloadButton_Click()
    ControlCount = 0
    lFilesNum = 0

    DownloadButton.Caption = "Stahování..."
    DownloadButton.ForeColor = &HFF&
    ViewForm.Repaint

    If FormatPAR.Value = True Then
        FolderType = "PAR"
        FormatType = ".par"
    Else
        If Left(Cells(ActiveCell.Row, 1), 1) = "E" Then
            FormatType = ".sgm"
        Else
            FormatType = ".odx"
        End If
        FolderType = "ODX"
    End If

    If InternetDownload = True Then
        ChosenFile = "Download"
        If Not FolderType = "ODX" Or InStr(1, Cells(ActiveCell.Row, 1), "w") > 0 Then
            PARName = SWName
        Else
            PARName = SWName & "_sw"
        End If
        FunctionsAndVariables.DownloadFile
        If FileFound = False Then
            GoTo NotFound
        Else
            MsgBox "Soubor " & PARName & " byl úspěšně stažen do umístění:" & vbNewLine & TextFilePath
            DownloadButton.Caption = "Stáhnout"
            DownloadButton.ForeColor = &H80000012
            Exit Sub
        End If
    Else
        FindedPAR = FindFiles(PARPath & "\" & FolderType, strMyFiles, lFilesNum, "*" & SWName & "*" & FormatType, False, True)
        If ControlCount = lFilesNum Or FindedPAR = False Then
            GoTo NotFound
        Else
            FileCopy strMyFiles(1).Path & strMyFiles(lFilesNum).Name, TextFilePath & "\" & strMyFiles(lFilesNum).Name
            MsgBox "Soubor " & strMyFiles(lFilesNum).Name & " byl úspěšně stažen do umístění:" & vbNewLine & TextFilePath
            DownloadButton.Caption = "Stáhnout"
            DownloadButton.ForeColor = &H80000012
            Exit Sub
        End If
    End If

    NotFound:
    FunctionsAndVariables.ContactAdministrator
End Sub
```

Příloha 8 Ukázka globálně používané funkce

```
Public Sub ContactAdministrator()  
Dim OutApp As Variant  
Dim OutMail As Variant  
Dim objNsp As Variant  
Dim colSync As Variant  
Dim objSync As Variant  
i = 0  
Administrators = ""  
AdministratorsMail = ""  
For i = 3 To 12  
With ThisWorkbook.Worksheets("Základní údaje tabulky")  
If Not .Range("B" & i).Value = "" And Not .Range("A" & i).Value = "Petr Večeř" And Not .Range("A" & i).Value Like "*David Hloušek*" Then  
Administrators = Administrators & "," & vbNewLine & .Range("A" & i).Value  
AdministratorsMail = AdministratorsMail & ";" & .Range("B" & i).Value  
End If  
End With  
Next i  
Administrators = Right(Administrators, Len(Administrators) - 1)  
If MsgBox("Parametrová sada " & SWName & " nebyla nalezena." & vbNewLine & vbNewLine & "Kontaktujte odpovědné osoby za administrativu:" & Administrators  
Set OutApp = GetObject(, "Outlook.Application")  
If OutApp Is Nothing Then  
Application.ActivateMicrosoftApp xlMicrosoftMail  
Set OutApp = CreateObject("Outlook.Application")  
End If  
If OutApp.inspectors.Count = 0 Then  
Set OutMail = OutApp.CreateItem(0)  
Else  
Set OutMail = OutApp.ActiveInspector.CurrentItem  
End If  
Set objNsp = OutApp.Application.GetNamespace("MAPI")  
Set colSync = objNsp.SyncObjects  
With OutMail  
.To = AdministratorsMail  
.cc = "David.Hlousek@skoda-auto.cz"  
.Subject = "Nenalezena parametrová sada " & SWName & " ve formátu PAR"  
.Display  
End With  
For i = 1 To colSync.Count  
Set objSync = colSync.Item(i)  
objSync.Start  
Next  
Set OutMail = Nothing  
Set objNsp = Nothing  
Set colSync = Nothing  
Set objSync = Nothing  
Set OutApp = Nothing  
Else  
Exit Sub  
End If  
End Sub
```

Příloha 9 Popisné řádky

```
'Update Var Vývoj column
If .Cells(i, Var).Value = "" Then
    .Cells(i, Var + 2).Borders(xlDiagonalUp).LineStyle = xlContinuous
ElseIf .Cells(i, Var + 2).Value = "" Then
    .Cells(i, Var + 2).Value = .Cells(i, Var).Value
    .Cells(i, Var + 2).Borders(xlDiagonalUp).LineStyle = xlNone
Else
    If Not Right(.Cells(i, Var + 2).Value, Len(.Cells(i, Var).Value)) = .Cells(i, Var).Value Then
        .Cells(i, Var + 2).Value = .Cells(i, Var + 2).Value & " " & Chr(62) & " " & .Cells(i, Var).Value
    End If
    .Cells(i, Var + 2).Borders(xlDiagonalUp).LineStyle = xlNone
End If
'Update Var BMG column
If Not .Cells(i, Var + 3).Value = "" Then
    If Not InStr(.Cells(i, Var + 3).Value, ";") = 0 Then
        PARName = Left(.Cells(i, Var + 3).Value, InStr(.Cells(i, Var + 3).Value, ";") - 1)
    Else
        PARName = .Cells(i, Var + 3).Value
    End If
    If InternetDownload = True Then
        ChosenFile = "Check"
        FolderType = "BMG_EF-S"
        FormatType = ".pdf"
        'PARName = .Cells(i, Var + 3).Value
        FunctionsAndVariables.DownloadFile
```

ANOTAČNÍ ZÁZNAM

AUTOR	Bc. David Hloušek		
STUDIJNÍ PROGRAM/OBOR/SPECIALIZACE	Specializace Řízení mezinárodních dodavatelských řetězců		
NÁZEV PRÁCE	Předávání projektových informací a optimalizace řízení projektu s využitím VBA v MS Excel		
VEDOUCÍ PRÁCE	doc. Ing. Jan Fábry, Ph.D.		
KATEDRA	KRVLK - Katedra řízení výroby, logistiky a kvality	ROK ODEVZDÁNÍ	2022
POČET STRAN	85		
POČET OBRÁZKŮ	26		
POČET TABULEK	1		
POČET PŘÍLOH	9		
STRUČNÝ POPIS	<p>Tato diplomová práce je zaměřena na předávání informací v globálních projektech a jakým způsobem toho vhodně docílit v prostředí Visual Basic For Application. Jedná se o konkrétní oddělení ve společnosti ŠKODA AUTO a.s. Cílem této diplomové práce je nalézt optimalizační model pro předávání informací pomocí agilních metod, zajistit stabilní a bezproblémové fungování navrhovaného modelu, který bude vytvořen pomocí naprogramované aplikace ve VBA for Excel.</p> <p>V teoretické části jsou shrnuty informace o projektovém managementu, projektu jako takovém, tradiční a agilní způsob vedení projektu včetně jejich porovnání a popis VBA. V praktické části je představena společnost ŠKODA AUTO a.s. včetně oddělení EPG/3, seznámení se situací před zavedením návrhu, popsán způsob zavedení změn a následné zhodnocení.</p>		
KLÍČOVÁ SLOVA	Logistika, VBA, Excel, programování, agilní, projekt		

ANNOTATION

AUTHOR	Bc. David Hloušek		
FIELD	Specialization International Supply Chain Management		
THESIS TITLE	Transfer of project information and optimization of project management using VBA in MS Excel		
SUPERVISOR	doc. Ing. Jan Fábry, Ph.D.		
DEPARTMENT	KRVLK - Department of Production, Logistics and Quality Management	YEAR	2022
NUMBER OF PAGES	85		
NUMBER OF PICTURES	26		
NUMBER OF TABLES	1		
NUMBER OF APPENDICES	9		
SUMMARY	<p>This diploma thesis is focused on the transfer of information in global projects and how to achieve this in a Visual Basic For Application environment. It is focused on specific department at ŠKODA AUTO a.s. The aim of this thesis is to find an optimization model for transmitting information using agile methods, to ensure stable and trouble-free operation of the proposed model, which will be created using a programmed application in VBA for Excel.</p> <p>The theoretical part summarizes information about project management, the project as such, the traditional and agile way of project management, including their comparison and a description of VBA. In the practical part, the company ŠKODA AUTO a.s. including the EPG/3 department is introduced, acquaintance with the situation before the implementation of the proposal, described the method of implementation of changes and subsequent evaluation.</p>		
KEY WORDS	Logistics, VBA, Excel, programming, agile, project		