

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová Práce

Nástroje na tvorbu digitálních dvojčat

Bc. Luděk Dvořák

Pod vedením: Ing. Michal Stočes, Ph.D.

© 2023 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Luděk Dvořák

Informatika

Název práce

Nástroje na tvorbu digitálních dvojčat

Název anglicky

Creating digital twins tools

Cíle práce

Práce je tematicky zaměřena na problematiku nástrojů ke tvorbě digitálních dvojčat. Hlavní cíl práce je z zhodnocení nástrojů na tvorbu digitálních dvojčat.

Dílní cíle jsou:

- Charakteristika nástrojů na tvorbu digitálních dvojčat.
- Výběr vhodných hodnotících kritérií.
- Zhodnocení vybraných softwarů.
- Formulace doporučení.

Metodika

Metodika teoretické části práce řešené problematiky diplomové práce je založena na studiu a analýza odborných informačních zdrojů, podle kterých bude následně v praktické části provedena zhodnocení nástrojů pro tvorbu digitálních dvojčat. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60–80 stran

Klíčová slova

Digitální dvojče, Nástroje, Průmysl 4.0, Simulace, Modelování

Doporučené zdroje informací

Digital Twin Driven Smart Manufacturing – Fei Tao, Meng Zhang, A.Y.C. Nee – ISBN: 9780128176306

Digital Twin, Kevin Chen, ROYAL COLLINS PUB CO, 2021, ISBN: 1487804865

Digital twin technology, Taylor & Francis Ltd, 2021, ISBN: 9781000455878

E-learning – Učení (se) s digitálními technologiemi: Wolters Kluwer ČR, Praha 2021, ISBN:
978-80-7676-175-9

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. Michal Stočes, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 14. 7. 2022

doc. Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 28. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 11. 03. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Nástroje na tvorbu digitálních dvojčat" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.03.2023

Poděkování

Chtěl bych mnohokrát poděkovat Panu Ing. Michal Stočes, Ph.D. za odborné vedení práce, že mi byl vstřícný ve volbě diplomové práce a za veškerý čas, který se mnou do práce vložil.

Nástroje na tvorbu digitálních dvojčat

Abstrakt

Diplomová práce je tematicky zaměřena na novější problematiku nástrojů na tvorbu digitálních dvojčat. Na začátku teoretické části jsou východiska zaměřovány na technologie a nástroje digitálních dvojčat, které se v průběhu problematiky podrobně popisují. Následně se rozebírají jednotlivé nástroje, s kterými dnes digitální dvojčata vytváříme a snažíme se je v praktické části komparovat mezi sebou různými metodami jako konkrétně vícekriteriální analýzou variant a to Saatyho, kde podle jednotlivých kritérií určíme nejlepší variantu. Finální část práce tedy dostáváme nástroj, který je dle určených okolností ideální a popisujeme jednotlivé možnosti implementace.

Klíčová slova: Digitální dvojče, Nástroje, Průmysl 4.0, Simulace, Modelování

Development of mobile application

Abstract

The diploma thesis is thematically focused on newer issue of tools for creating digital twins. At the beginning of the theoretical part, the starting points are focused on the technologies and tools of digital twins, which are further described in detail in the course of this issue. Subsequently, the individual tools with which we create digital twins today are described, in the practical part. We try to compare them with each other using different methods, specifically multi-criteria analysis of variants, namely Saaty, where we determine the best variant according to individual criteria. In the final part of the work, we receive a tool that is ideal according to the determined circumstances, and we describe individual implementation options.

Key words: Digital Twin, Tools, Industry 4.0, Simulation, Modeling

Obsah

1	Úvod	13
2	Cíl práce a Metodika	14
2.1	Cíl práce	14
2.2	Metodika	14
3	Teoretická východiska.....	15
3.1	Digitální dvojče.....	15
3.1.1	Vývoj	19
3.1.2	Rozdělení	19
3.1.3	Vrstvy digitálních dvojčat chytré infrastruktury města	21
3.1.4	Jednotlivé využití v praxi.....	24
3.1.5	Proces Obrábění.....	26
3.1.6	FMS (Flexible manufacturing system)	27
3.2	IoT.....	28
3.2.1	Rozdíl mezi IoT vs digitální dvojče.....	29
3.3	Umělá Inteligence	30
3.3.1	Využití UI u digitálních dvojčat	30
3.4	Průmysl 4.0	31
3.5	Postupy vývojového procesu.....	32
3.5.1	Vodopádový model.....	32
3.5.2	Prototypový přístup.....	34
3.5.3	Spirální model.....	35
3.5.4	RAD přístup	36
3.5.5	Extrémní programování	37
3.5.6	Shrnutí.....	39
3.6	Vývojové nástroje	39

3.6.1	Siemens NX	40
3.6.2	Unity	42
3.6.3	Unreal Engine	43
3.6.4	Azure.....	46
3.6.5	IBM.....	47
3.6.6	ArcGIS	48
3.6.7	Shrnutí.....	49
3.7	Modely vícekritériální analýzy variant	50
3.7.1	Bodovací metoda	52
3.7.2	Saatyho metoda.....	52
3.7.3	Shrnutí.....	53
4	Praktická část.....	54
4.1	Konzistence trendu v odborné studii.....	54
4.1.1	Shrnutí.....	56
4.2	Selekce vývojových nástrojů.....	56
4.3	Výběr a ohodnocení kritérií	59
4.3.1	Použitelnost.....	60
4.3.2	Hardwarové požadavky	61
4.3.3	Doba učení	63
4.3.4	Náročnost UI.....	64
4.3.5	Cena licence	65
4.4	Vícekritériální analýza variant – Stanovení kompromisní varianty.....	66
4.4.1	Shrnutí.....	67
4.5	Implementace	68
4.5.1	Požadavky	68
4.5.2	Návrh	68

4.5.3	Vývoj	70
4.5.4	Verifikace.....	71
4.5.5	Shrnutí.....	76
5	Závěr	77
6	Seznam použité literatury	78
7	Seznam příloh.....	85
7.1	Příloha 1 – Stanovení vah kritérií – Saatyho metoda párových porovnání.....	85
7.2	Příloha 2 – Nastavení Unity Analytics.....	86
7.3	Příloha 3 – Unity zapínání a vypínání světel	87

Seznam obrázků, tabulek a grafů

Obrázky

Obrázek - 1 - Metoda CFD v praxi ^[6]	16
Obrázek - 2 - zátěžový test Ojnice ^[9]	17
Obrázek - 3 - ohybné rameno u kola a turbína ^[10]	17
Obrázek - 4 - Rozdělení podle integrace ^[14]	20
Obrázek - 5 – tvorba OSM v softwaru QGIS ^[15]	22
Obrázek - 6 – Vrstvy ^[16]	24
Obrázek - 7 - Digitální dvojče větrné turbíny ^[29]	25
Obrázek - 8 - Digitální dvojče města ^[29]	26
Obrázek - 9 - Simulace deformace lopatek ^[17]	27
Obrázek - 10 - Digitální dvojče a průmysl 4.0 ^[24]	32
Obrázek - 11 – Vodopádový model ^[56]	33
Obrázek - 12 - Prototypový model ^[57]	34
Obrázek - 13 – Iterativní přístup ^[61]	35

Obrázek - 14 - RAD model ^[59]	36
Obrázek - 15 - Extrémní programování ^[60]	38
Obrázek - 16 – Siemens tvorba digitálních dvojčat ^[30]	41
Obrázek - 17 – Unity tvorba digitálních dvojčat ^[35]	43
Obrázek - 18 – Digitální dvojče v Unreal Engine ^[44]	45
Obrázek - 19 - Microsoft Azure ^[43]	46
Obrázek - 20 - Návrh modelu	69
Obrázek - 21 - Reálný návrh modelu	69
Obrázek - 22 - Přesun dat	70
Obrázek - 23 - Design modelu	70
Obrázek - 24 - Doba od zapnutí světel	71
Obrázek - 25 – Počet stisknutí vypínače.....	72
Obrázek - 26 - Unity Analytics dashboard	73
Obrázek - 27 - Export dat z Unity Analytics	73
Obrázek - 28 - Dashboard Power BI.....	74

Tabulky

Tabulka 1 - Počet IoT zařízení v průběhu let ^[26]	29
Tabulka 2 - Siemens NX ^{[65][66]}	41
Tabulka 3 - Unity	43
Tabulka 4 - Unreal Engine.....	46
Tabulka 5 - Azure	47
Tabulka 6 - IBM Exchange.....	48
Tabulka 7 - ArcGis	49
Tabulka 8 - Shrnutí nástrojů	50
Tabulka 9 - Informace o preferencích mezi kritérii ^[62]	52
Tabulka 10 - Použitelnost	60

Tabulka 11 - Ohodnocení použitelnosti.....	61
Tabulka 12 - Hardwarové požadavky	62
Tabulka 13 - Ohodnocení hardwarových požadavků	63
Tabulka 14 - Ohodnocení doby učení.....	63
Tabulka 15 - Ohodnocení náročnosti UI.....	64
Tabulka 16 - Ohodnocení ceny licence.....	65
Tabulka 17 - Maximalizační tabulka ohodnocení.....	66
Tabulka 18 - Bodovací metoda s vahami.....	67

Grafy

Graf 1 – Oblasti použití digitálních dvojčat.....	55
Graf 2 – Využití digitálních dvojčat	55
Graf 3 - Využití digitálních dvojčat v různých oblastech použití.....	56
Graf 4 - Hodnocení vývojových nástrojů	58
Graf 5 - Počet recenzí vývojových nástrojů.....	58
Graf 6 - Posouzení dominancí variant	66
Graf 7 - Doba svícení.....	75
Graf 8 - Počet stisknutí tlačítka.....	75
Graf 9 - Cena energie.....	76

1 Úvod

V dnešní době se digitální dvojčata stávají více a více populární a firmy se předhánějí a zlepšují za cílem digitalizací jejich výrobních linek případně výrobků. Prvním a vlastně největším průkopníkem je nejznámější národní úřad pro letectví a vesmír NASA, která své modely musela zautomatizovat z důvodu dálkových oprav.

S podobným problémem se vyskytla i letecká společnost Airbus, která měla problém z hlediska zaučení, kdy si technici museli přes velice drahou cestu nastudovat veškeré modely letadel, aby věděli, jak daná letadla opravovat. Později tedy společnost Airbus přišla s řešením opravy, a to vytvořením digitální kopie. Tento technologický pokrok umožnil novým i stávajícím technikům simulovat testy a provádět opravy ve virtuální realitě v kratším časovém intervalu a tím reagovat rychleji na možné problémy, které se vyskytnou.

Velice zásadní věcí je, že firmy berou pořízení digitálního dvojčete jako nákladné, což častokrát není pravda. Většina firem již mnoho věcí pro vytvoření má v archívech, a to například plány, prostory, jednotlivé modely, manuály apod. Přehození do digitálního modelu je nesmírnou výhodou, protože paradoxně celý projekt na základě simulací, testů a vylepšení zlevní.

Hlavní pojmy, které tato práce popisuje, vychází z umělé inteligence a IoT (Internet věcí), jelikož tyto technologie jsou hlavním pilířem při vytváření digitálních dvojčat.

Nejde pouze o testování, případně vytváření replik ke zlepšení, ale o mnoho více. Příkladem lze jmenovat monitoring neboli sledování již funkčních letadel, kde pracujeme s ukazateli jako poruchovost, výkon nebo i opotřebení. ^[1]

2 Cíl práce a Metodika

2.1 Cíl práce

Práce je tematicky zaměřena na problematiku nástrojů ke tvorbě digitálních dvojčat. Hlavní cíl práce je zhodnocení nástrojů na tvorbu digitálních dvojčat.

Dílní cíle jsou:

- Charakteristika nástrojů na tvorbu digitálních dvojčat.
- Výběr vhodných hodnotících kritérií.
- Zhodnocení vybraných softwarů.
- Implementace modelu.
- Formulace doporučení.

2.2 Metodika

Metodika teoretické části práce řešené problematiky diplomové práce je založena na studiu a analýze odborných informačních zdrojů, podle kterých bude následně v praktické části provedeno zhodnocení nástrojů pro tvorbu digitálních dvojčat. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry diplomové práce.

3 Teoretická východiska

Úvodní kapitola bude popisovat teoretické části diplomové práce, jedná se konkrétněji o podrobné rozebrání simulací digitálních dvojčat a jejího významu. Objasnění jednotlivých částí, s kterými jsou spjata digitální dvojčata jako například průmysl 4.0 a později i samostatné softwary.

3.1 Digitální dvojče

Pod pojmem digitální dvojče si můžeme představit mnoho věcí. Jednou z nich je odraz reálného modelu ve virtuálním prostředí. Příkladem, lze uvést větrnou turbínu. Nejprve se postaví a na základě toho se reflektuje do modelu, kde je možné simulovat různé podmínky jako například: maximální výkon při ideálním počasí, výdrž v případě zemětřesení atd. [2]

Digitální dvojče lze použít nejen jako odraz reálného modelu, ale čehokoliv jiného. Je to virtuální replika modelu, produktu procesu nebo systému. Ke svému prospěchu využívá znalosti umělé inteligence, a to bez ohledu na lokalitu zařízení. Velikým plusem je, že změna nebo úprava ovlivní systém ještě před tím, než se realizuje v praxi, a tudíž sebou nenese žádná rizika a změna bude provedena bez chyb. Veškeré potřeby ke vzniku jsou na základě fyzických vlastností daného systému a musí obsahovat potřebné rysy, které jsou vytvořené za pomoci matematických modelů. Po vytvoření takového modelu dokážeme za pomoci snímacích senzorů zkoumat aktuální přehled systému a na základě toho upravovat činnost na produkci. Model lze dále rozvíjet a optimalizovat pro potřeby budoucího stavu. [2]

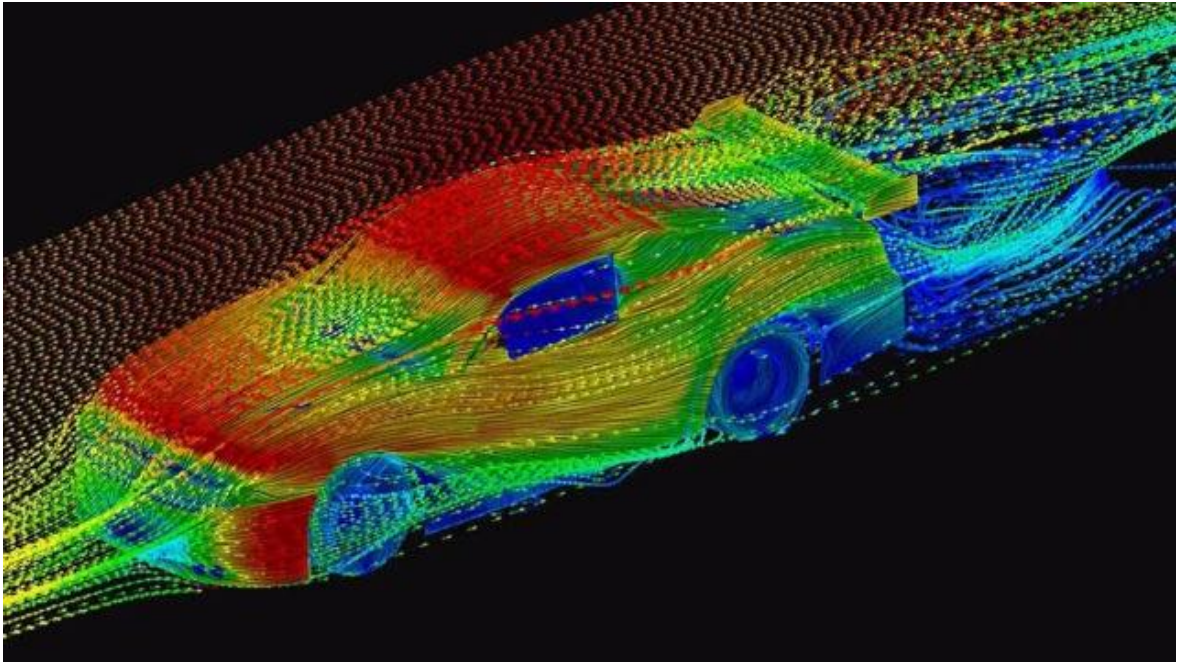
Rozdíl mezi simulací a digitálním dvojčetem?

Ačkoliv se technologie mohou zdát stejné je to právě naopak. Digitální dvojčata obsahují daleko rozšířenější inženýrskou analýzu za pomoci softwaru neboli Computer aided engineering (CAE). Dalším rozdílem je Internet of Things (IoT), který nám otevírá možnosti za rychlejším přenosem dat a zároveň nám umožňuje vidět, jak se produkt chová v reálném čase, to u simulace nevidíme. [5]

Computer aided engineering (CAE)

Slouží k usnadnění inženýrské analýzy za použití softwaru. Používá metody, které zpřesňují analýzu jako **Výpočetní dynamiku tekutin (CFD)** neboli také dynamiku proudění, která pomáhá numerickému modelování. Dále pomáhá v mnoha směrech stanovit

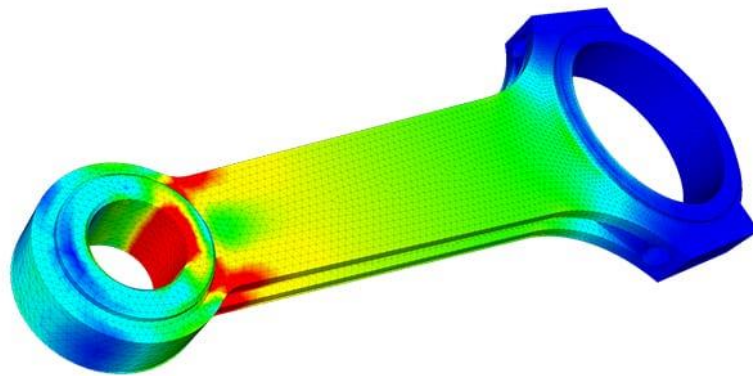
průběhy a rozložení rychlosti, tlaku, sil a dalších veličin. Na obrázku 1 je možné vidět zkouška aerodynamiky auta a celkové výdrže konstrukce.^{[6][7]}



Obrázek - 1 - Metoda CFD v praxi^[6]

Další analýzou je **Metoda konečných prvků (FEM)** – obdobně jako předešlá metoda je i tato numerická a zkoumá simulaci průběhu deformací, frekvencí případně napětí, ale také proudění tekutin, tepla nebo elektromagnetismu. Princip funguje na Galerkinově metodě, která řeší soustavy parciálních diferenciálních rovnic a říká nám konečný počet prvků.^[8]

Princip metody je založen na Lagrangeově principu, který je zobrazen na Obrázku 2. V krátkosti se vytvoří CAD model, který je v rovnováze, pouze pokud Celková potenciální energie deformace soustavy je minimální.^[8]

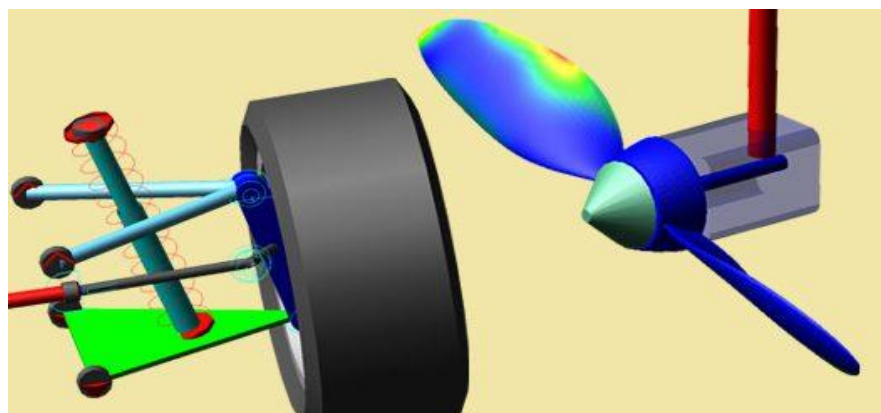


Obrázek - 2 - zátěžový test Ojnice ^[9]

Soustava mnoha těles (MBD) – je studie založená na dynamickém chování. Zkoumá chování modelu v praxi například fungování turbíny v běhu nebo rotace kola za jízdy a jeho ohyb ramene auta viz obrázek 3. Dále zkoumá také rotaci, flexibilitu, pohyb, propojenost atd. ^[10]

Za pomoci této analýzy, lze zlepšovat mechanický pohyb tělesa, a to v oborech:

- Robotika
- Biomechanika
- Dopravní odvětví
- Dynamické simulace
- Strojírenský průmysl ^[10]



Obrázek - 3 - ohybné rameno u kola a turbína ^[10]

Výhody

Hlavní výhodou této technologie je efektivita. Když se implementuje digitální dvojče kamkoliv, vždycky to má své výhody z hlediska efektivního využití času. Lze uvést příklady, kdy při stavbě budovy mohlo dojít k opomenutí trámu, který byl kritickou částí infrastruktury, aby nemohlo dojít ke zřícení budovy při silných bouřích. Nebo při stavbě turbíny, kde byly výpočty trošku nepřesné a turbína byla postavena na neefektivním místě. Právě díky simulacím, lze tyto problémy podchytit a vyvarovat se v co největší míře rizikům v reálném světě.

Dálkové monitorování je také jedna z výhod, pomocí které lze sledovat přes IoT senzory v reálném čase a následně byznys můžeme data analyzovat a vyřešit tím problémy v systému.

Údržba a bezpečnost je také nedílnou součástí digitálních dvojčat. Při problému lze vždy situaci nasimulovat a vyřešit stejně jako u zmiňovaného případu s NASA a jejich kyslíkovou bombou, která zachránila mnoho životů. V údržbě se pak dá přijít ještě na dodatečné detaily a to, že se dají nasimulovat nové situace, které ještě v průběhu návrhu a implementace nebyly známé.^[19]

Nevýhody

Jako tomu bylo dříve, když se lidé obávali dávat data na internet z důvodu jednodušší krádeže, tak je tomu i zde u digitálních dvojčat. V případě jednoduchého modelu, kde se dělá pouze replika budovy tak tam je to riziko menší, ale pokud vytváříme větší projekt, a to například monitorovací systém vodní elektrárny, tak v případě nějakého vnějšího kybernetického útoku je zde obrovské riziko, a to z vícero důvodů. Prvním a nejzávažnějším je ztráta dat. Pokud elektrárnu řídíme a monitorujeme online tak to riziko je fatální. Ke ztrátě dat se váže, ale mnoho víc věcí než to, že budeme muset obnovovat celý systém znova, je zde větší problém toho, že útočník ví o našem systému veškeré informace tím pádem je pravděpodobný únik dat na veřejnost, popřípadě prodej těchto dat, objevení slabých míst reálného modelu, možnost obměnit data atd. V případě vodní elektrárny to zas tak hrozné není, ale kdybychom si vzali příklad z možné budoucnosti ve zdravotnictví, kde bývalí GE CEO Bill Ruh předurčuje, že každý člověk po jeho narození bude mít svojí digitální kopii, která se bude aktualizovat a následně obměňovat tím pádem zde už to riziko útoku je o dost větší. V extrémním příkladu při pozměnění údajů na nějakou smrtelně závažnou nemoc by

pacient mohl podstoupit smrti nebezpečnou operaci, kterou ve skutečnosti nepotřeboval.^{[21][22]}

3.1.1 Vývoj

Použití dvojčete proběhlo již velice dávno ačkoliv to ještě název digitálního dvojčete neneslo, jelikož dříve se používali analogové koncepty dvojčete. První použití přinesl národní úřad pro letectví a vesmír NASA, která v průběhu sedmdesátých letch dvacátého století pracovala na projektu Apollo. V průběhu práce na projektu Apollo 13 došlo k nehodě, kdy explodovala kyslíková nádrž a ohrozila životy celé posádky tím, že poškodila servisní modul. Aby se takovéto neštěstí neopakovalo tým inženýru z NASA vytvořila repliku kosmické lodě s technickými detaily a nasimulovaly situaci, která se při nehodě stala. Přišli s řešením a následně pomocí hypotetických postupů dokázali opravit problém a tím nejspíše zachránit životy celé posádky

První zmínka byla napsána v knize „Mirror Worlds“ od David Gelernter v roce 1991 a první koncept modelu přišel až o 9 let později roku 2002 díky Michaelu Grieves, který prosazoval metodu do výrobního průmyslu.

Název digitálního dvojčete ovšem přišel až roku 2010, kdy John Vickers z NASA přišel s názvem u odevzdávání hlášení. Obsahoval koncept, který se skládal ze tří částí, a to fyzický produkt, digitální/virtuální produkt a poslední část byla o propojení mezi dvěma předešlými částmi.^{[12][13]}

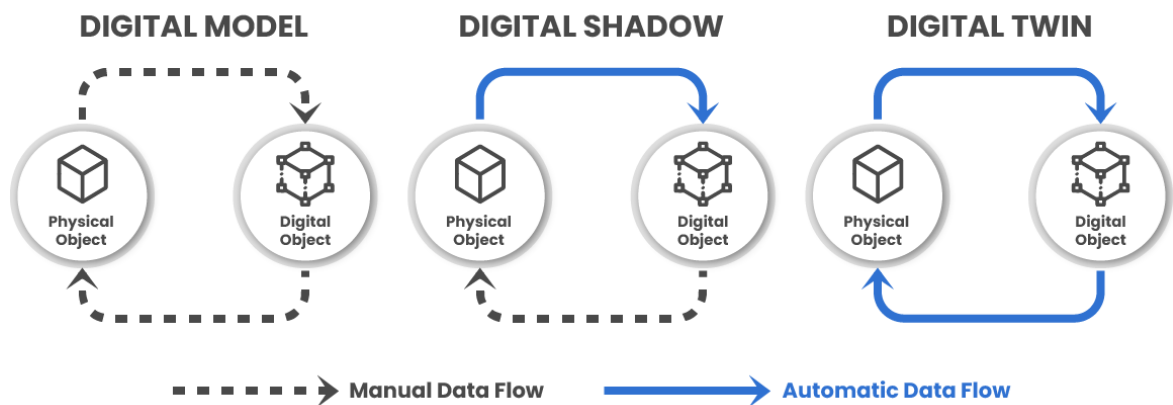
3.1.2 Rozdělení

Rozdělení podle vývoje

Digitální dvojče se rozděluje do třech základních typů: prototyp, instance a agregát. Prvním typ prototypu slouží ještě před tím, než je hotový fyzický model projektu. Využívá detailních analýz, návrhů a k vytvoření reálného modelu systému je možné připravit různé test case situace, které můžou nastat a dá se jim tak předejít. Digitální dvojče instance je vytvoření jednotlivých instancí systému, při tvorbě fyzického modelu. Posledním je agregát, který tvoří instance a přidávají se zde veškeré informace a data, která mohou být použita v jiných aplikacích, a tím pádem můžeme říci, že digitální dvojčata jsou logickými konstruktory.^[14]

Rozdělení podle Integrace

Dalším rozdělením jsou integrační formy, které pracují na jiných typech přenosu dat a informací. Digitální model, Digitální stín a Digitální dvojče, o kterém je tato práce. Digitální model je samotný návrh 3D modelu nebo designu CAD, který se vytvoří za pomoci 3d tiskárny, laseru nebo interpretací do programu. Digitální stín je dohledatelná aktivita, akce nebo komunikace na internetu případně na zařízení a je rozdělena do pasivní nebo aktivní. Kupříkladu na prohlížeči se taková aktivita ukládá do cookies, z kterých se dá dále číst. Poslední je digitální dvojče, které obsahuje obě tyto metody jak digitální návrh, tak i dohledatelnou komunikaci, která se používá mezi fyzickým a digitálním modelem zároveň tedy navíc obsahuje automatický dataflow, který je vidět na obrázku 4 níže. [14]



Obrázek - 4 - Rozdělení podle integrace [14]

Rozdělení podle produktu

- **Komponentové dvojčata**

Jsou základní jednotkou digitálních dvojčat a zároveň nejmenší. Fungují stejně a slouží k výrobě menších produktů. [49]

Asset dvojčata

Pokud dvě či více komponent spolu spolupracuje tvoří takzvaný „Asset“ neboli soustavu digitálních dvojčat navzájem komunikujících. [49]

Jednotková či systémová dvojčata

Jsou pokročilou úrovní „assetů“, která tvoří funkční systém. Již provádí základní mechanické práce a netvoří pouze jednoduchý proces. Integrací můžeme také dosáhnout, že

nám systémová dvojčata doporučí vhodný způsob zapojení ke zlepšení workflow a efektivnosti systému.^[49]

Procesní dvojčata

Je výsledkem celé operace, která nám odhalí, jak může systém spolu pracovat a vytvoří celé zařízení. Procesní dvojčata pracují na maximální efektivitě a spolupráci od, které se následně odvíjí její přesnost.^[49]

3.1.3 Vrstvy digitálních dvojčat chytré infrastruktury města

Chytrá infrastruktura města se obvykle dělí do 6 jednotlivých vrstev, kde prvních 5 je postavených na sobě z hlediska přidání informací jako například: jaký je terén, kde budou budovy, IoT zařízení, doprava atd. Poslední vrstva virtuální neboli digitální dvojče nám slouží jako ta nejdůležitější, protože si dokáže předávat informace s vrstvou digitální. Na základě dat, která se sesbírají z předešlých vrstev se následně uskuteční simulace jako například optimalizace rozmístění budov z hlediska terénu, využití větrných turbín z hlediska efektivity a třeba výkonnost slunečných panelů v čase. Na základě simulací se využijí informace, ze kterých se dále čerpá a využijí se v reálném světě.^[16]

Vrstva terénu

Je nultou vrstvou digitálního dvojčete chytrého města. Obsahuje veškeré základní informace o městě, a to na jakých částech města jsou pláže, zda protéká řeka městem, je na vyvýšeném místě nebo naopak v prohlubni atd. Veškeré podrobnější informace skládá technik a následně se implementují do simulace.^[16]

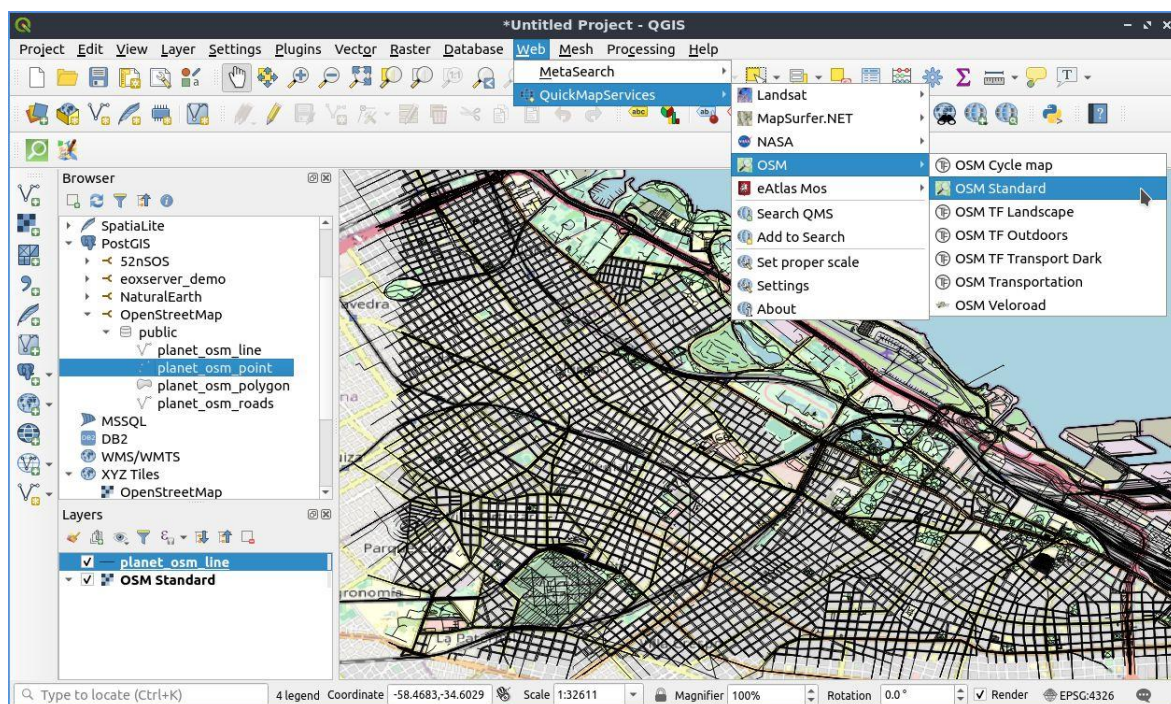
Vrstva budov

Přidává veškeré budovy do modelu. Zde je nutnost dbát na přesnost, aby implementace proběhla správně, také proto se používá známý model BIM (informační model budovy), který slouží ke všemu, a to od odstranění budovy až po jeho vzniku zkrátka má veškeré informace o budově.^[16]

Vrstva infrastruktury

Hlavním cílem v infrastruktuře je vytvoření organizačních budov, telekomunikací, turistických cest, elektráren atd. Nejčastěji se čerpá z OpenStreetMap (dále jen OSM), kde se tvorba modelu velice urychlí, jelikož čerpá z otevřené databáze. Jsou zde například

silnice, cesty, stezky apod. OSM je otevřeným softwarem, který využívá volně dostupná data a transformuje je do topografických map. [16]



Obrázek - 5 – tvorba OSM v softwaru QGIS [15]

Vrstva mobility

Důležitá vrstva, která přináší lidem velké výhody, a to dopravu a veškerý pohyb lidí, který je možno trasovat. Ve vrstvě se vymyslí nejefektivnější cesta dopravy pomocí zavádění nových tras a cest z hlediska úspory.

V programech se k tomu používá metoda „Traffic control“ neboli kontrola dopravy, která se snaží vytvářet lepší alternativy v trasách, které jsou obvykle vytížené například u silniční dopravy. Eventuální budoucnost je taková, že každý občan bude jako digitální dvojče. Který by se vytvořil do digitálního modelu města, dále se promítne do vrstvy mobility a toto umožní optimalizaci nejvytíženějších tras a zároveň odstranění nepoužívaných.

Aktuálně bohužel funguje na principu pluginu. V programu Unity se dá pořídit plugin kontroly dopravy a na základě jeho uvést počet aut, které městem projíždění lze tak jednoduše nasimulovat křižovatky, které byly špatně navrženy a dochází u nich ke větším kolonám, následně upravit intervaly semaforů a případně předělat křižovatku úplně. [16][25]

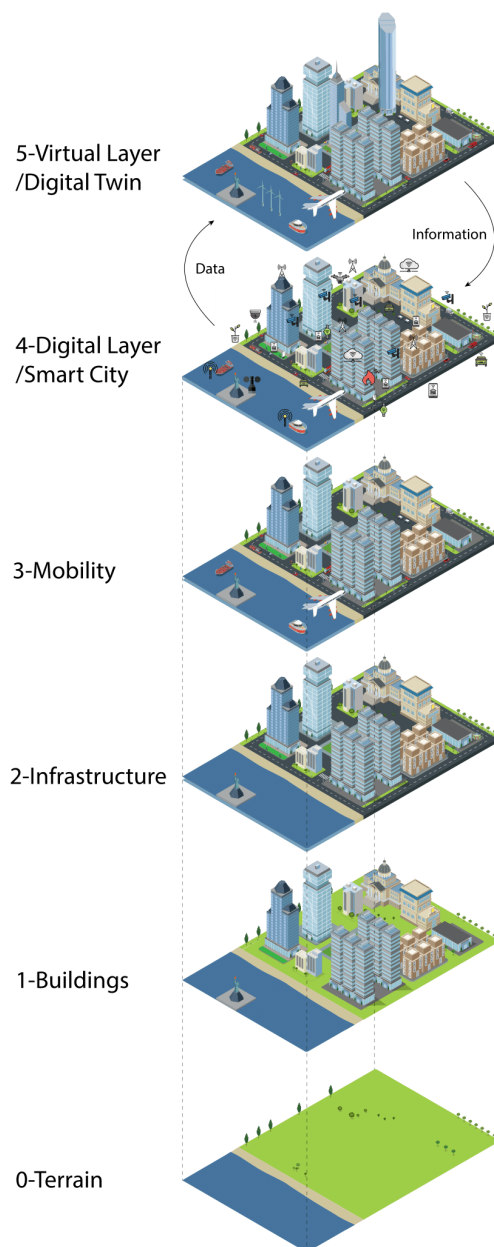
Vrstva digitální (chytré město)

Digitální vrstva odpovídá situaci dnešní doby, kdy už převážná část měst sbírá informace za pomoci IoT senzorů a je to velice populární řešení projektů. Město sbírá data a na základě nich dělá úsudky, které vedou ke změně dopravních procesů, dopravy anebo monitorování. Nejčastějšími příklady jsou: školy, elektrárny, Doprava neboli kamerový systém, který měří rychlost vozidel, nemocnice, knihovny a další.

Jak je ukázáno na obrázku 6 níže, digitální vrstva odesílá data do vrstvy virtuální, kde mohou občané posílat informace o dopravě. Například se takto data odesílají přes aplikaci Waze nebo mapy.cz a podle nich se pak aktualizuje mapa, kde se dozvíme, zda na daném místě je kolona. Chytré hodinky nebo mobily můžou přispět k různým opravám, kdy občané mohou odesílat zpětnou vazbu městu. Dalším příkladem je zalévání kyttek, který mají na sobě zalévací senzor a na základě toho se časově monitoruje doba, kdy přijde notifikace. Senzor na detekování kouře, který má následně přes IoT poslat notifikaci zavolat hasiče je taky další zařízení, které se v rámci toho implementují například do elektráren. O takových senzorech, které vracejí data a na základě toho na to reaguje druhá strana je nespočet. ^[16]

Vrstva virtuální (digitální dvojče)

Virtuální vrstva nebo také digitální dvojče se staví na základě dat, která dostane z vrstvy digitální a na základě toho odesílá informace, která můžeme vidět na obrázku 6 níže. Nepřijímá pouze nějaká data, ale veškerá, o kterých zde byla zmínka jako například terén, budovy, infrastruktura, mobilita a následně i o digitální vrstvě. V ukázce na obrázku 6 jsou použity vzdušné turbíny na výrobu zpětné energie pro celé město a zde se setkáváme se simulací. Na základě, které získáme, poté vytvoříme repliku a simulujeme, kde je potřeba, aby byly turbíny zapojeny na pobřeží. Data ovšem nepřijímáme všechny například pro tuto situaci vůbec nepotřebujeme informaci o infrastruktuře, mobilitě, terénu. Víme už, že na pobřeží je možnost zapojit nebo zapojení IoT zařízení, díky datům jako větrné podmínky zasílané pobřežními loděmi, zpětnou vazbou občanů nebo aktuální počasí posílané senzory. ^[16]



Obrázek - 6 – Vrstvy ^[16]

3.1.4 Jednotlivé využití v praxi

Letecký průmysl

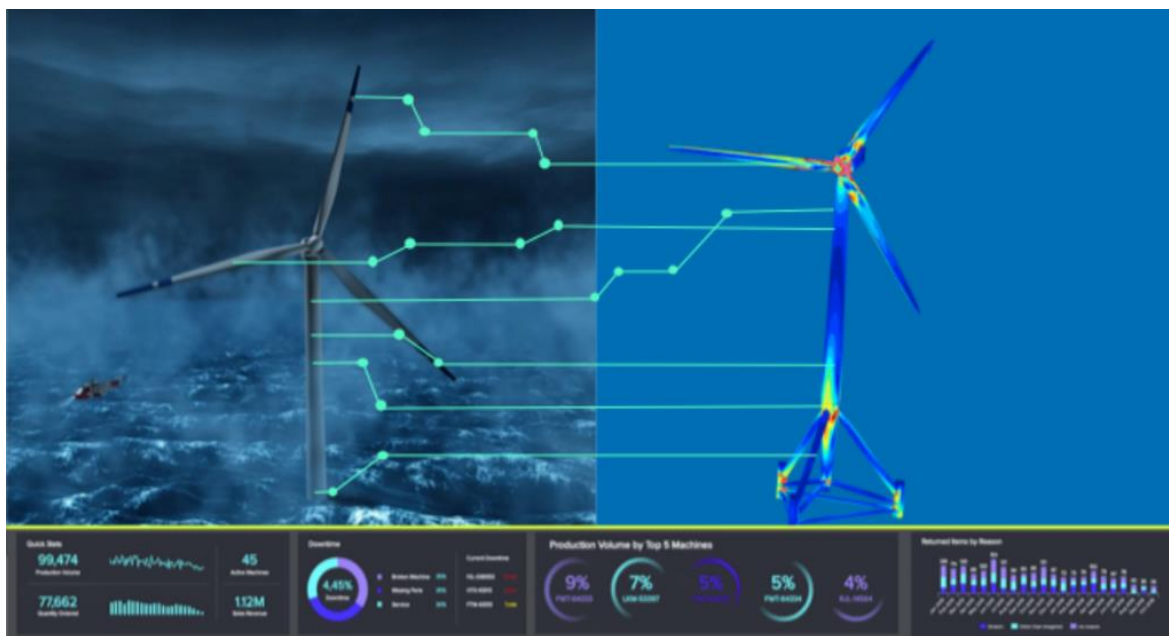
Digitální dvojčata se v leteckém průmyslu začali používat velice hojně, a to z jediného důvodu a to konzistence. Aby každé letadlo mohlo být vždycky stejně vytvořeno a monitorováno konkrétně po využití se systémy. Kvalita letadel Boeing zlepšila až o 40 % což je dramatický nárůst v tak důležité dopravě. Při použití programů lze simulovat veškerý

průběh letu od proudění vzduchu, vytváření tlaku až po změny věnostních podmínek na letadlo.^[30]

Větrné elektrárny

Zde byl vždycky problém a to ten, že nikdy nelze nasimulovat přesný počet vyrobené elektriky a to z důvodu velké inkonzistence podmínek, které zde působí. Těmi jsou vítr, počasí, opotřebení křídla turbíny a případně i tlak. Zkrátka je těžké vypočítat kolik bouřek se přežene případně zda bude vítr foukat správným směrem. Z toho důvodu se berou data za několik měsíců a pak lze dopočítat budoucí výrobu.^[30]

Digitální dvojčata zde umožňují vypočítávat opotřebení jednotlivých turbín na základě monitorování a následně dávat upozornění na potřebnou údržbu k jejich opravě. Lze také simulovat úhel, postavení případně i upravení větrné turbíny k lepší efektivitě.^[30]



Obrázek - 7 - Digitální dvojče větrné turbíny^[29]

Městská infrastruktura

Ideální městská infrastruktura by měla mít senzory na veškeré údaje, a to ať už na aktuální stav internetu, elektriky, plynu, vody atd., ale zároveň i na možnosti simulace v případě nedostatku daných příkladů. Dopad, a i následnou opravu digitální dvojčata vyhodnotí i v následku nějaké přírodní pohromy, která nastala. Infrastruktura by měla zaznamenat ohrožené části a využít data k zaslání řešení.^[30]

Za pomoci umělé inteligence lze vyhodnocovat dané opravy, které systém zaznamená a již se tak používá. Například ve velkých městech Boston, Singapore, Šanghaj nebo New York. Dále lze toto také aplikovat na dopravu, kterou můžeme simulovat a tím optimalizovat nejvytíženějších částí města. (více ve 3.1.4 Vrstvy digitálních dvojčat chytré infrastruktury města)

Příkladem vzorového monitoringu je obrázek 8. Ukazuje nám údaje o městě a lze se podle vrchního panelu přesouvat na jednotlivé odvětví města.



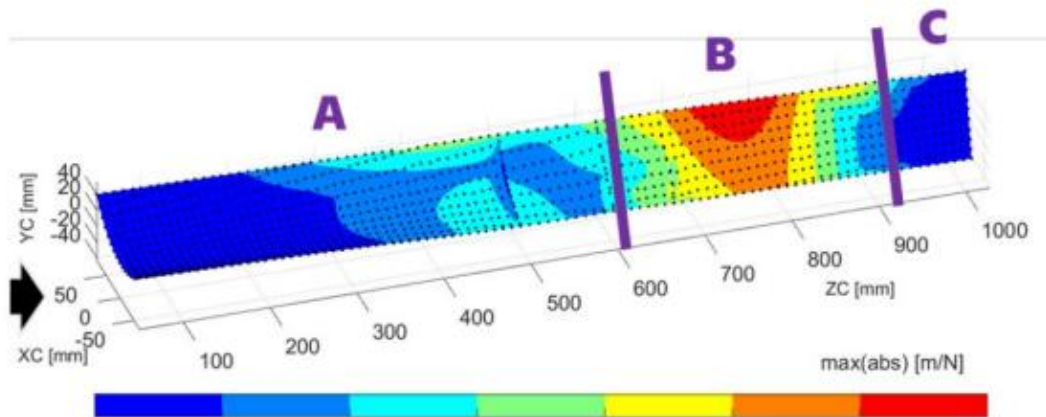
Obrázek - 8 - Digitální dvojče města [29]

3.1.5 Proces Obrábění

Obrábění je složitý technologický proces, kterým se dá vytvořit výrobek požadovaného tvaru. Digitální dvojče lze v této kategorii též využít na simulování obrábění náročnějších dílů, případně na zlepšení techniky nebo procesu samotné metody. V dnešní době se tomu již používá, a to i z důvodu, že digitální dvojče dbá na nastavení CNC neboli stroje řízené počítačem. Největší výhodou by bylo odstraňování chyb, který na základě simulací lze snadno odstranit a následně tak ověřit stroj bez nutnosti předešlých testů.^[17]

Ideálním příkladem byla optimalizace obrábění štíhlých poddajných lopatek u tvarově složitých dílců. Záměr byl vytvoření procesu výroby součástek, kde bylo vysoké riziko deformace z důvodu častých vibrací, ke kterým docházelo a to v nejkratším časovém intervalu a zároveň v co nejvyšší kvalitě. Proto při integrované simulace řezných sil bylo

dosaženo optimální varianty, kde se vzalo správné množství řízených otáček, aby nedošlo k porušení povrchu. Digitální dvojče rozdělilo lopatky do třech částí, které vidíme na obrázku 9 níže a následně u každého rozdělení určilo jak vibraci, tak i řezné parametry, aby předešlo poškození. ^[17]



Obrázek - 9 - Simulace deformace lopatek ^[17]

3.1.6 FMS (Flexible manufacturing system)

V překladi flexibilní výrobní systém, který slouží na okamžitou změnu. Systém může v mžiku sekundy reagovat na změnu situace, a to ať už předpovídanou nebo nepředvídanou. FMS dělíme na dvě kategorie, a to směřovací flexibilitu, která nám takzvaně ohýbá systém dokáže vytvářet nové typy produktů případně úplně změnit systém. Druhá je strojová flexibilita, která nám určuje, zda stroje dokáží měnit výrobu a to jak časově tak kapacitně nebo produkčně. ^[23]

Výhody

- Lepší kvalita
- Efektivita
- Lepší bezpečnost ^[23]

Nevýhody

- Drahý na vytvoření a přípravu
- Zapotřebí zkušenějších techniků ^[23]

3.2 IoT

IoT neboli internet věcí lze popsat více způsoby. V nejjednodušší formě se jedná o ekosystém chytrých zařízení a počítačů či strojů, které spolu navzájem komunikují a pracují bez asistence člověka. Dobrým příkladem jsou chytré hodinky, které se s postupem času začali zlepšovat a umožňovat více informací jako komunikaci mezi telefonem a samotnými hodinkami. Funkce jako měření tepu, sledování kvality spánku případné notifikace se staly běžným standardem. Výhoda ovšem nespočívá pouze pro uživatele, ale i pro vývojáře. Přes IoT lze odesílat zpětnou vazbu, jelikož zařízení dokáže komunikovat a odesílat informace do vnějšího světa. ^[10]

Cloud se stal také populárním řešením pro IoT produkty. Zařízení obsahují nadměrná množství dat a ty musejí být někam odesílána. Zde tedy přichází metoda M2M (machine to machine), která jak z názvu vypovídá přenáší informace ze zařízení na zařízení. Příkladem jsou naše hodinky. Ty mohou posílat informace serveru v rámci zpětné vazby. ^[11]

Digitální dvojčata jednoznačně vyžadují IoT. Kdybychom si vzali vymyšlenou ukázkou ze science-fiction filmů, kde dokáže systém bez pochyby zobrazit například planetu s detailní mapou a mít možnost přiblížení tak se jedná o vytvořené digitální dvojče planety za pomocí IoT, kde došlo nejspíše k nějaké podrobné sondě, která vrátila informaci systému a ten na základě toho implementoval digitální dvojče. ^{[10][11]}

Na základě statistické stránky můžeme vidět počet zařízení v biliónech během času, které vidíme v Tabulce 1 níže. Do srpna roku 2022 jsou to vymezené údaje, které byly zaznamenány a zbylé jsou již odhady na základě statistiky. ^[26]

Rok	Počet IoT zařízení (v biliónech)
2019	8,6
2020	9,7
2021	11,3
2022	13,1
2023	15,1
2024	17,1
2025	19,1
2030	29,4

Tabulka 1 - Počet IoT zařízení v průběhu let ^[26]

V průběhu sekundy se na internet připojí něco kolem 127 IoT zařízení, které nepotřebují lidskou asistenci, aby spolu mohli komunikovat. Zároveň se předpovídá, že k roku 2026 IoT zařízení bude používat 93 % společností. Cenový odhad takových zařízení by vycházel na 26 triliónu českých korun. ^{[27][28]}

3.2.1 Rozdíl mezi IoT vs digitální dvojče

Hlavním rozdílem mezi těmito dvěma technologiemi je, že digitální dvojčata musí být zapojená k internetu na rozdíl od IoT, která lze jednoduše připojit například přes Bluetooth případně jiný signál.

Druhým a zásadním faktem je, že digitální dvojčata vznikají na základě repliky, jak už z jejich názvu vyplývá. Dům tedy stavíme podle vybudovaného modelu v konkrétním softwaru na rozdíl od IoT, který to dvojče nepotřebují a jsou budovány separátně. Světla do domu se nejdříve implementují. Dále se udělají IoT čidla na Bluetooth a následně přes telefon je lze zapínat a vypínat. Nikoli naopak, kdy by se nejdříve udělat model světla, který by byl simulovaný, zda funguje správně případně jej lze vylepšit (jsou na to předepsané parametry, baterky, žárovky, které si obvykle stanovuje dodavatel) ^{[47][48]}

3.3 Umělá Inteligence

Samotná umělá inteligence (UI) je těžce zařaditelným oborem. Lze jí brát dvěma směry, buď jako technický obor anebo matematickou disciplínu s aplikacemi. Za velký zvrat bylo považováno automatizování v oblastech řízení, kde dochází k usnadňování činností za pomoci UI případně u průzkumu života nebezpečných míst, kde je vyloučena možná přítomnost člověka. ^[63]

Jako tomu je u digitálních dvojčat první větší použití využila NASA, k řízení a kontrole dálkových družic. Od té doby se více a více rozvíjela a stále není žádná jednotná definice, která by popsala UI, a to z toho důvodu, že se nachází ve více disciplínách jako: matematická logika, kybernetika, informatika, neurologie atd. Nic jako konečná fáze vývoje UI zatím není ani podána, je to stále souvisle rozvíjící se vědní disciplína. Základním prvkem UI je počítač. V současné době rozdělujeme UI na slabou – zde je počítač jako nástroj, který UI využívá a silnou – zde již UI přímo myslí a pomocí programu rozumí. V budoucnu se někdy můžeme dočkat ještě silnějšího rozdělení, kde by UI zvládalo myslet samo za sebe, ale kde nějaké scénáře by mohli dopadnout jako je v námětech ve filmech. ^[63]

Jedna z nejpřesnějších definicí UI, kterou v literatuře můžeme najít je od Marvin Minsky, který tvrdí, že „Umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu užívat takového postupu, který – kdyby ho dělal člověk – bychom považovali za projev jeho inteligence.“ (1967) ^[63]

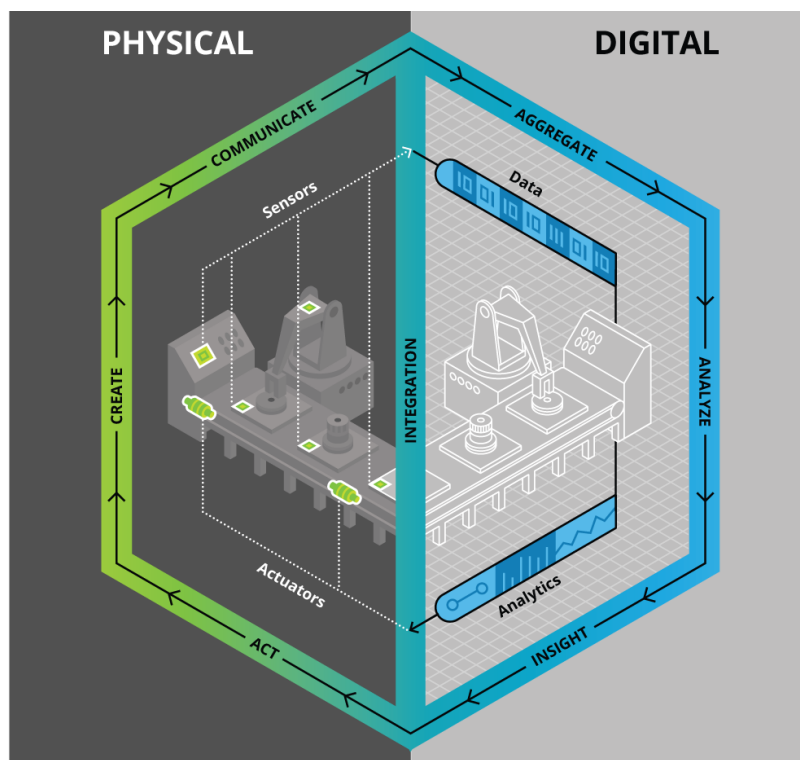
3.3.1 Využití UI u digitálních dvojčat

Zde je dopad opravdu veliký, protože bez UI by ve směs veškeré testování a možné simulační scénáře byly k ničemu. Umělá inteligence zpřesňuje tedy možné pohledy k tomu, co může dojít, a proto vylepšuje metody simulování. Využití se dá například uvést ve „Smart“ oblastech, kde například u „Smart Cities“ lze emulovat za pomoci UI dopad na životní prostředí, kde se na základě dat zprostředkují továrny, elektrárny, veškeré komíny, farmy atd. Po simulaci se následně musí udělat analýza, ze které už zjišťujeme, jak dané znečištění odstranit. ^[64]

3.4 Průmysl 4.0

Též jako čtvrtá průmyslová revoluce. Vyskytují se zde kyberneticko-fyzikální systémy a je to posun digitalizace zase o krok dopředu. Digitální dvojčata jsou úzce spjata s průmyslem 4.0, jelikož se zde hojně využívají. V příkladu můžeme uvést chytré elektrárny, které mezi sebou zároveň komunikují. Ovšem zde v takovém pokroku už dochází k velkému riziku a to, že tyto elektrárny, které byly doposud bezpečné, tak se pomocí takovéto míry digitalizace otevírají novým nebezpečím a to například v podobě hackerských útoků. Běžná elektrárna funguje za neustálého dozoru pracovníků na místě. Pokud například vypadne proud, lze postupovat pomocí krizových scénářů, jelikož pracovníci jsou on-premisově. U chytré elektrárny může dojít k útoku z vnější jako například výš zmiňovaný hackerský útok, který může zablokovat nástroje na obsluhu elektrárny a tím znemožní dodávky elektřiny. V případě takového dlouhodobého výpadku elektřiny to může mít veliký dopad na fungování závislých oblastí/měst/vesnic nebo jiných objektů. Zde se též otevírá otázka vyjednávání s útočníky, kteří mohou držet kritickou infrastrukturu bez elektřiny nebo jiné hrůzné scénáře. Bez vysoce kvalitních bezpečnostních systémů se zde neobejdeme.^[24]

Ve výsledku takový návrh může vypadat jako na obrázku 8 výše, kde jednotlivé senzory na fyzikální vrstvě posílají informace digitální a navzájem spolu komunikují. Zkrátka řečeno taková obsluha už mi u elektrárny vůbec nemusela být a mohlo by se vše řídit dálkově z pohodlí domova. Celý proces je tedy cyklus, kde na základě vstupních dat se naženu stroje a ty podávají výsledky systému, které se následně odešlou jako report a jsou dále zpracovávány.^[24]



Obrázek - 10 - Digitální dvojče a průmysl 4.0^[24]

3.5 Postupy vývojového procesu

Různé metody, přístupy a modely se dají použít pro vývoj procesu, proto je potřeba si jednotlivé postupy vysvětlit a na základě toho si vybrat ideální pro konkrétní využití. Každá úspěšná společnost se danými postupy řídí, a to z toho důvodu, že jsou efektivní a zamezují chybám. ^[52]

Základní kameny u vývoje digitálních dvojčat jsou tři. Sběr dat, modelování a zpětná vazba. Je potřeba si utvořit pojmy se zákazníkem o jeho nápadech a požadovaných specifikacích na vytvoření repliky. Následně jsou dvě možnosti. První, pokud se jedná o vývoj, který ještě nezná svůj reálný protějšek je potřeba vytvořit model, který slouží jako simulace modelu reálného. Následně se provádějí testy na základě, kterých poté vytvoříme dvojče již v reálném světě. Druhá možnost je ta jednodušší varianta a to, že již reálné dvojče existuje. Proto se řídíme daty, které již máme a snažíme se je využít pro vytvoření kopie. ^[52]

3.5.1 Vodopádový model

Je sekvenční vývojový proces, který funguje jako posloupnost na sobě navazujících etap bez žádných zpětných kroků. V ideálním světě by takhle měl fungovat

každý model, jelikož jak se vytvoří, tak se i dokončí. Paradoxně se říká, že takový model nelze v praxi využít, ale opak je pravdou. Velice známé firmy a společnosti jako NASA anebo Ministerstvo obrany Spojených států daný model využívá. Hlavním důvodem je zde velice důležitý každý krok, a to hlavně specificky ty první dva. Analýza a návrh, u kterých již od začátku musíme vědět finální vizi produktu.^[53]

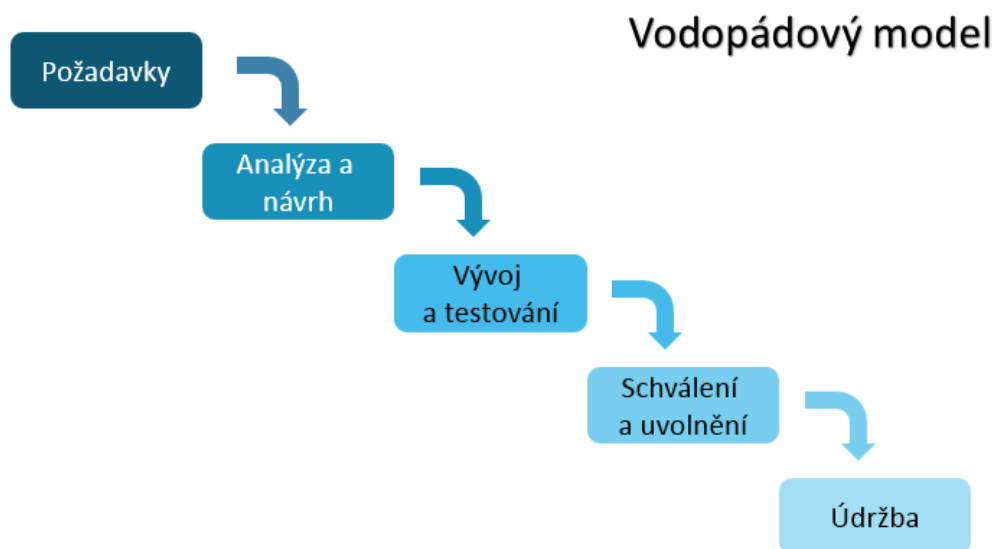
Prvním průkopníkem modelu byl Winston W. Royce, který roku 1970 publikoval článek v kterém model popisoval, a ačkoliv je považován za vynálezce tak „svůj“ model kritizoval a nazval ho jako „chybný“ a „nefunkční“. ^{[53] [54] [55]}

Výhody

- Jednoduchý z hlediska řízení
- Ideální výsledek přesně dle specifikací

Nevýhody

- Uživatel nemusí vědět veškeré požadavky předem
- Verze, ve které by uživatel viděl chyby, které nejsou žádoucí, jsou dodány, až v příliš pozdějších krocích (fáze verifikace)
- Pokud se rozhodne o změně funkčnosti je projektová doba hodně prodloužena a zároveň se model začíná implementovat znova.
- Obtížný aplikovat v praxi ^[54]



Obrázek - 11 – Vodopádový model ^[56]

3.5.2 Prototypový přístup

Model se zakládá na komunikaci se zákazníkem a je velice hojně využíván v praxi. Primární soustředění je na sběr dat a analýzu, která vytváří tzv. „prototypy“. Ty jsou následně konzultovány a upravovány s uživatelem. Dle představ se prototypy upraví a až po odsouhlasení se zákazníkem se dává vývoj do produkce.^[52]

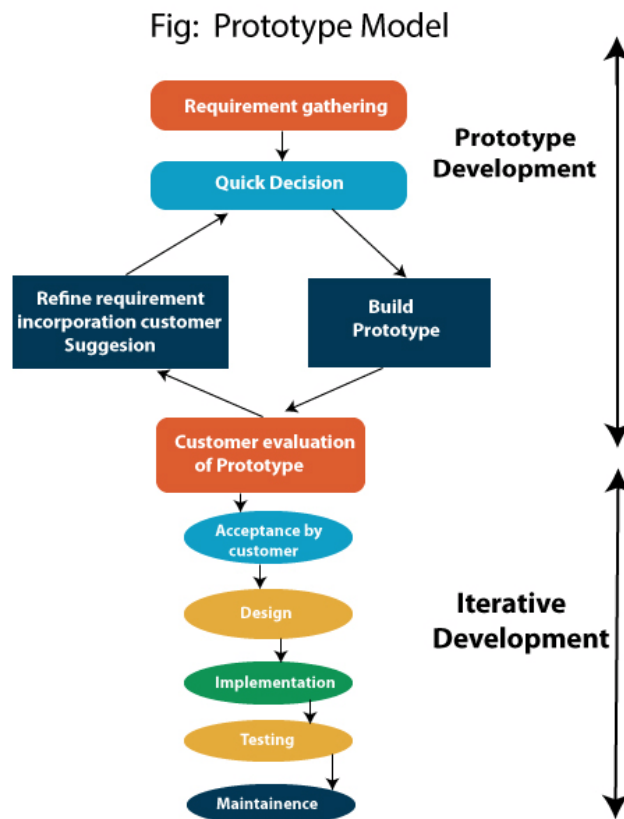
Funguje na principu vodopádového modelu, kde, ale dochází k zacyklení v analýze a návrhu tím pádem může doba projektu trvat déle, jelikož se projekt neustále dovytváří dle představ.^[54]

Výhody

- Pokud je zákazník nerozhodný a nevíme přesné zadání je tento model ideální
- Levný na vývoj (V případě změny se prototyp pouze upraví dle zadání zákazníka a nemusí se tak šahat do již funkční aplikace)

Nevýhody

- Časově náročný (prototyp se pořád upravuje a vytváří)



Obrázek - 12 - Prototypový model^[57]

3.5.3 Spirální model

Využívá ve svém modelu tzv. **Iterativní přístup**. Je založený na iteracích (opakování), které vypadají jako jedna velká smyčka (viz obrázek 13 níže), kde dojde v podrobném naplánování a rozvržení softwaru. Následně se vytvoří analýza, testy, úpravy v cyklu a až po veškerém upřesnění se započne s vývojem. [52]



Obrázek - 13 – Iterativní přístup [61]

Spirální model sám o sobě vychází z vodopádového modelu, ale rozlišuje dva prvky:

- Iterativní přístup
- Důsledná analýza rizik

Funguje tak, že při každém kroku se provede analýza rizik, kde se vezmou veškerá rizika, která mohou nastat a jejich hrozba se očísluje dle nebezpečnosti a pravděpodobnosti výskytu. Dle takové analýzy se následně odstraňuje velký výskyt chyb. [54]

Výhody

- Podrobnější analýza
- Méně chyb
- Možnost se vrátit v případě změny (interace)

Nevýhody

- Časově náročný

3.5.4 RAD přístup

Jako model spirálový tak též přístup RAD (rapid application development model) vychází z iterativního přístupu, který je cyklený o menší kus, kde chybí testování a implementace. Dalo by se říci, že je to kombinace spirálního modelu a zároveň modelu prototypového, kdy na základě iterací dochází k vytváření prototypů, které se konzultují se zákazníkem. ^[54]

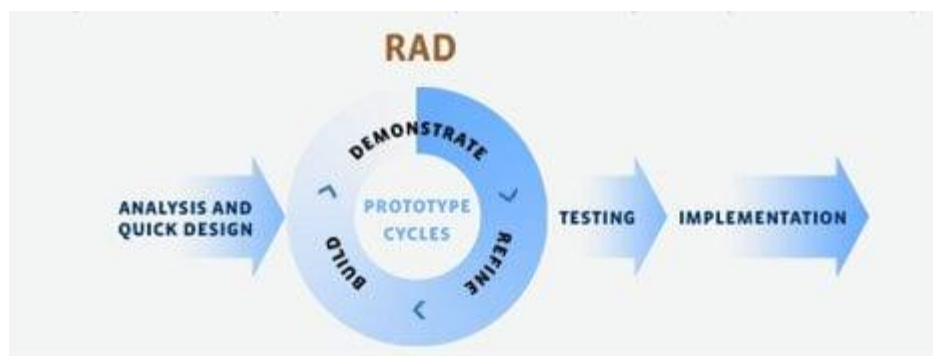
Postupy modelu můžeme najít na obrázku x a velkým rozdílem například od vodopádového je ten, že ke komunikaci se zákazníkem se dostáváme již při ranných začátcích aplikace neboli ve fázi prototypu. ^{[52][53]}

Výhody

- Jednoduchý
- Rychlý

Nevýhody

- Testovací a implementační fáze se necyklí
- Riziko chyby



Obrázek - 14 - RAD model ^[59]

3.5.5 Extrémní programování

Je metodika, která se nehodí pro velké vývojové týmy a je to z hlediska toho, jelikož se zadání často mění nebo není z prvního hlediska jasně definováno. Opírá se o několik nastavených principů:

- Dochází k úzké komunikaci mezi zákazníkem a programátorem
- Využívá iterační cyklení, kde při každé smyčce dochází k včasné, přesnější a neustálé zpětné vazbě.
- Návrh systému probíhá neustále a funguje jako tzv. „evoluční proces“.
- Přírůstkový přístup počítá s tím, že se plán může kdykoliv během vývoje změnit.

Důležitou vlastností této metodiky je též striktně dodržovat veškerá pravidla, aby se metoda mohla správně implementovat. V praxi je toto velice složité, jelikož se častokrát mění vývoj ze dne na den, a tak to může být i nemožné. ^[52]

Extrémní programování dostalo svůj název díky až extrémnímu množství činností, které využívá. K dosažení názvu je potřeba do písmene využívat s velkou pečlivostí všech dvanáct postupů:

- **Refaktorizace** – optimalizace kódu
- **Plánovací hra** – tvorba plánu skrz celý tým
- **Metafora** – nahrazuje termín „architektura“. Dochází k vyprávění konečného systému jako příběh.
- **Jednoduchý návrh**
- **Testování**
- **Párové programování** – kód se vyvíjí ve dvojici, kde dva programátoři pracují na jednom počítači.
- **Malé verze** – častější iterace verzí.
- **40hodinový pracovní týden** – dle metody se tento standard považuje za uvolněnější a lehčí pracovní dobu, kde programátor podává kvalitnější a přesnější výsledky, jelikož je více odpočatý.
- **Zákazník na pracovišti** – nelze pracovat z tzv. „home office“ neboli z domova. Používá se agilní metodika, která dbá i u programátorů na verbální komunikaci.

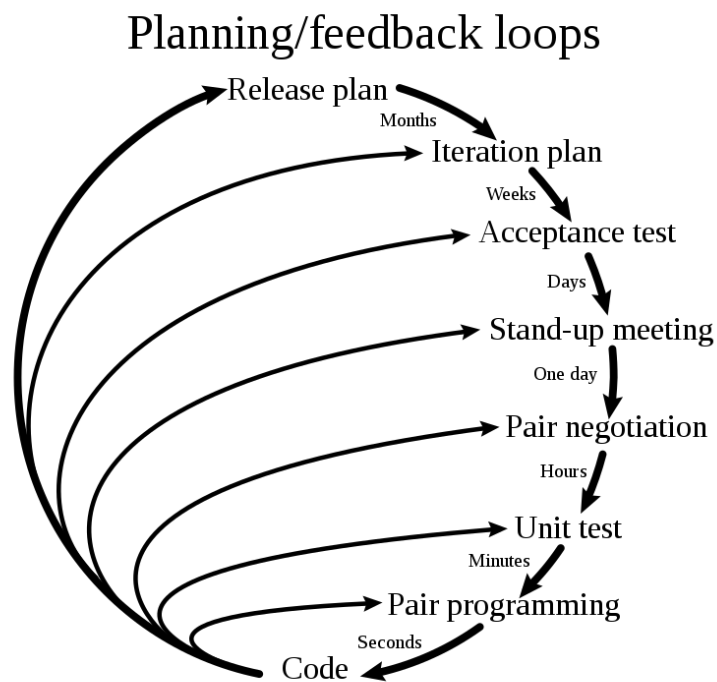
- **Společné vlastnictví** – Zamezuje vytváření tříd se svými vlastníky, ale umožňuje obecné znalosti celého týmu.
- **Nepřetržitá integrace** – Testování a sestavování celého systému vícekrát v průběhu dne. [52][53][54]

Výhody

- Složitě fungování metodiky
- Každý programátor má obecnou znalost o systému
- Zamezuje chybám

Nevýhody

- Pro velké společnosti nemožné
- Těžké na implementaci



Obrázek - 15 - Extrémní programování [60]

3.5.6 Shrnutí

Množství postupů vývojových procesů, ze kterých si můžeme vybrat je mnoho. Jako nejlepší se jeví vodopádový model, který je ale ovšem nejtěžší na implementaci a to z důvodu absurdní analýzy. Při vytváření digitálních dvojčat je tento model těžko použitelný, jelikož ve většině případů už reálný model existuje a stačí tedy vytvořit pouze digitální repliku. To samé platí o prototypovém modelu a RAD, který se již nevrací po fázi testování. Silnými modely pro použití jsou poslední dva a to spirální a extrémní programování. Obě tyto metody s testovací fází mohou přejít zpátky na fázi navrhovací. Příkladem lze uvést, kdy máme turbínovou vrtuli a zjistíme, že materiál, z kterého jí chceme vyrobit, by silné větrné podmínky nevydržel a musíme tedy měnit design a projekt. ^{[52][53][54]}

3.6 Vývojové nástroje

Jsou nedílnou součástí digitálních dvojčat, která slouží nejen k naprogramování, ale i k vytvoření celého systému. Pracují s podrobnými principy kreslení a plánování jako jsou CAD/CAM případně CAE (Computer aided engineering) o kterém jsme se již bavili. Zajímavostí je, že žádné zmíněné prostředí NASA nepoužívá, jelikož mají své interní.^[2]

Z důsledku velkého množství vývojových nástrojů na tvorbu digitálních dvojčat se vybralo 6 nástrojů, které díky své veliké popularitě vyhrávají jsou jimi:

- Siemens NX
- Unity
- Unreal Engine
- Microsoft Azure
- ArcGis
- IBM Watson

Veškerá prostředí, která jsou v rámci práce použita, jsou neplacené verze a zároveň je lze volně dostupně stáhnout na webových stránkách tvůrců. Výběr vývojového prostředí je zásadní. Důležité je nahlédnout na oblast, ve které budeme pracovat a vytvářet digitální dvojče. Ve většině případů dochází k výběru kompromisní varianty.

CAD (Computer aided design)

Slouží čistě pro návrh, kde využívá matematické, geometrické a grafické nástroje pro vykreslení plošných výkresů v lepších verzích se následně využívá i na analýzu a řízení systému.^[33]

CAM (Computer aided manufacturing)

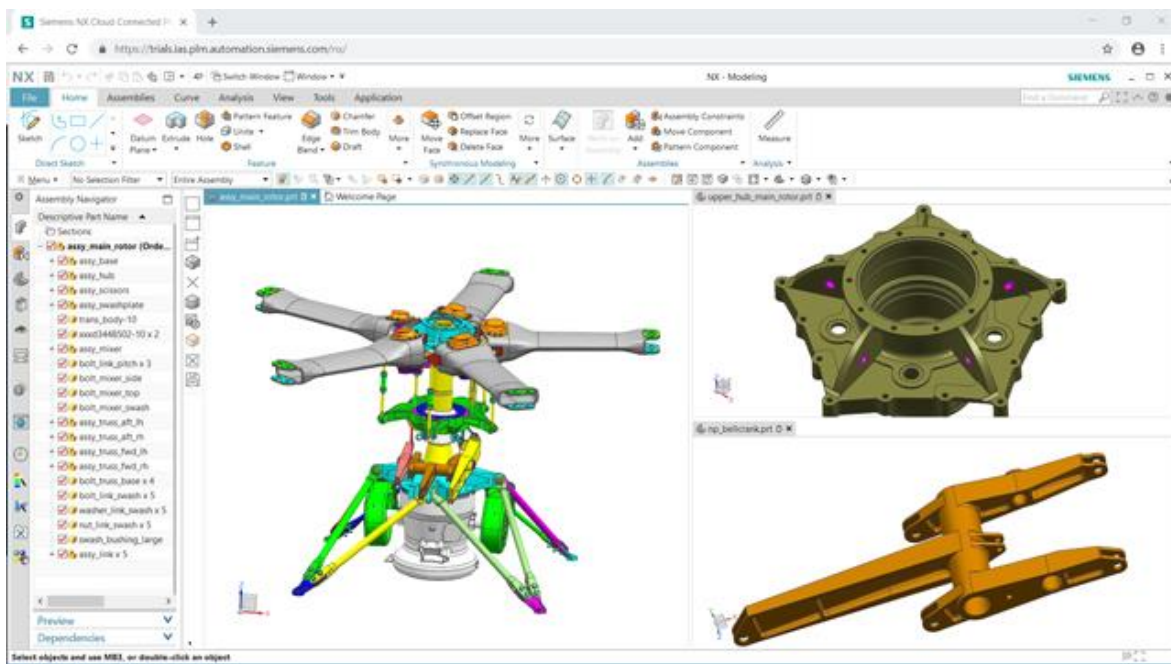
Je technika obrábění, která se u digitálních dvojčat využívá a umožňuje navrhnout proces vyrábění tak, aby byla co nejvíce efektivní a nejlevnější.^[33]

3.6.1 Siemens NX

Dříve bývalí Unigraphics je komerční program, který využívá CAD/CAM a CAE. Běží nad objektově orientovanou databází a je to jeden z nejstarších programů na světě. V roce 1969 byl vyvinut první CAM produkt. Využívá veškeré nejnovější funkce jako Real-time rendering, re-use podpora, Teamcenter atd.^[25]

Program je cenově drahý. Pro studenty je ovšem zdarma s určitým omezením, ale v případě verze Designer je jeho cena kolem 3500 dolarů ročně což je něco kolem 85 000 korun českých. V případě profesionální verze stojí dvakrát tolik. Skrz ostatní verze krom verze Designer je cenové rozpětí větší a to kolem 6000 až 20 000 dolarů.^[25]

Siemen NX má jednoduché přívětivé rozhraní, které můžeme vidět na obrázku 16 níže a veškeré modely se dají vytvořit jako v programech od Autodesku CAD. Obsahuje podobný model rozhraní jako program Microsoft Word, který má horní panel zaměřený na možnosti úpravy a následně levý panel k pohybu navigace.^[30]



Obrázek - 16 – Siemens tvorba digitálních dvojčat^[30]

Programování v Siemens NX je velice jednoduchý s jejich 22letou praxí a obrovskou výhodou, která umožňuje vytvoření API v jiném programovacím jazyce, než je program napsán. Knihovny podporují programovací jazyky jako: C++, VB.NET, C# and Java.^[32]

Na základě informací ze stránek tvůrce bylo možné vytvořit tabulku 2 níže, která nám udává základní informace o nástroji. Jako jeden z mála nástrojů neumožňuje vyzkoušet si program zdarma pokud již nejste studentem. Zároveň ani popularitou není tak oblíbený a to se dá odůvodnit tím, že obsah, kterým disponuje je především určen pro tvorbu digitálních dvojčat.^{[65][66]}

Nástroj	Verze pro studenty?	Verze pro vyzkoušení?	Obsah	Popularita	Cena základní licence
Siemens NX	Ano	Ne	Standardizované knihovny, sdílené programování	740 000 uživatelů	80 800 Kč ročně

Tabulka 2 - Siemens NX^{[65][66]}

3.6.2 Unity

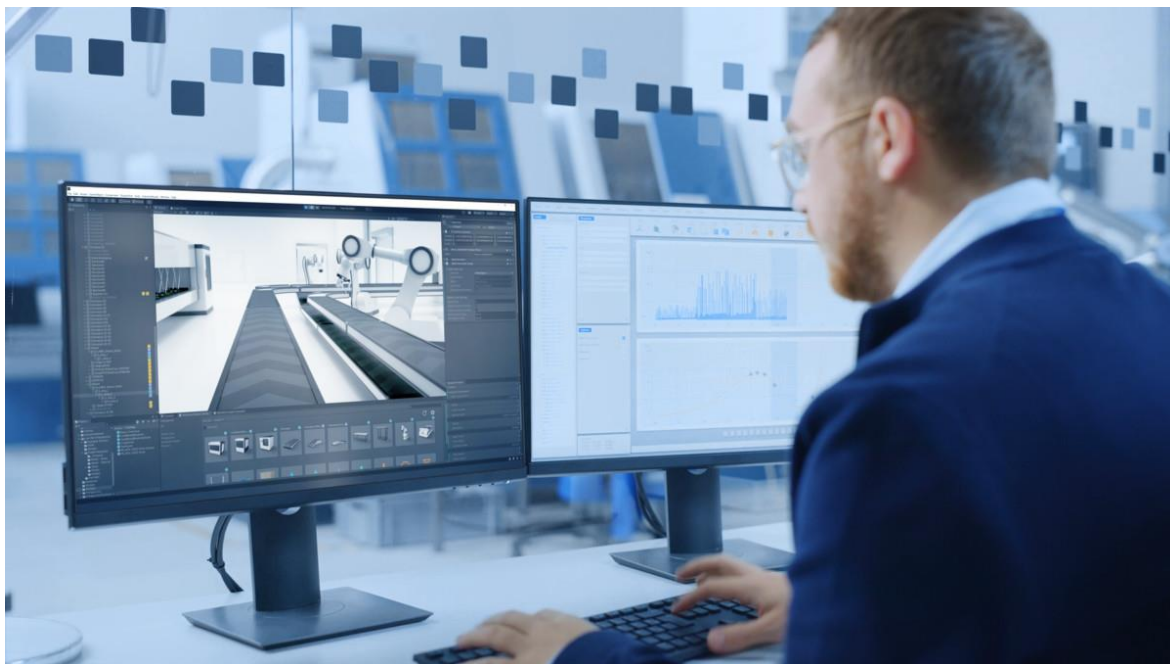
Je multiplatformní herní engine, vyvinutý společností Unity a byl představen roku 2005 na konferenci firmy Apple. Velikou výhodou je, že jeho využití není jenom u her, ale je globální a lze tak vytvářet aplikace, digitální dvojčata, modely atd. [34]

Značnou výhodou Unity je nástroj, který umožňuje podporu Visual Studia. Jelikož obsahuje Debugger, refactoring, možnosti rychlé opravy a další nabídky možností atd. V normálním případě by se kód musel skriptovat a psaní obvykle v tom to případě trvá déle.^[34]

Programovací jazyky Unity jsou C# a C++ a zároveň Unity umožňuje podrobné modelování, které bohatě postačí i nejnáročnějším vývojářům.^[34]

Licence je o dost levnější než v ostatních případech. Standardní verzi lze využívat se všemi funkcemi až výdělku 100 000 dolarů, což je něco kolem 2 500 00 českých korun. Veškeré další přídatky, kterým říkáme knihovny jsou už pak z většiny případů placené. Dále můžeme krom přídatků přidávat Assets což je věc, která se hojně používá a v našem projektu to může být například krásně vymodelovaný model od jiného tvůrce. Tento model se nachází v obchodu Assetstore, ze kterého ho stáhneme a můžeme jej využít v našem projektu nebo jsou zde i nové funkce, která mohou přidat nové pohledy, nastavení, animace apod.^[34]

Unity perspective digital twin je Asset, který umožňuje vytvoření virtuální testovací reality pro daný předmět a ten můžeme následně upravovat a konfigurovat k našemu očekávání. Tento přídatek je hojně využíván pro tvorbu Digitálních dvojčat a je volně stažitelný na stránkách Assetstore. Na oficiálních stránkách pak lze zhlédnout video o tvorbě mechanických rukou v dílnách viz. Obrázek 17 na další stránce.^[35]



Obrázek - 17 – Unity tvorba digitálních dvojčat ^[35]

Unity též obsahuje externí knihovny, které se liší od ostatních vývojových nástrojů. Tento vývojový nástroj je jeden z nejpopulárnějších a zároveň velice dostupný pro širokou veřejnost. Veškeré podrobnosti jsou vidět v tabulce 3 níže. ^[67]

Nástroj	Verze pro studenty?	Verze pro vyzkoušení?	Obsah	Popularita	Cena základní licence
Unity	Ano	Ano	Externí knihovny, horší importování vlastních	2,7 billionů uživatelů	48 000 Kč ročně

Tabulka 3 - Unity

3.6.3 Unreal Engine

Je herní engine, který sloužil k vytváření počítačových her. První verze byla zaměřena ze začátku pouze pro střílečky z pohledu první osoby. Byl vytvořen roku 1996 společností Epic Games a první hra, kterou software vytvořil byla Unreal. Je jedním z nejúspěšnějších vývojářských nástrojů k tvorbě počítačových her. Dále je využíván i k tvorbě softwaru a aplikací. ^[36]

Veškeré hry jsou napsané v programovacím jazyce C++ jako samostatné jádro software. Podpora byla ze začátku pouze pro hry a tím pádem použití digitálních dvojčat

nebylo reálné v rámci možností softwaru. K možnosti využití tohoto vývojářského nástroje došlo až s příchodem nové verze Unreal Engine 4, která představila zcela nové rozšíření softwaru, a to Unreal Engine Architecture. ^[36]

Unreal Engine 1

Byl první vydanou verzí roku 1996. Prvním produktem, který se ukázal světu byla hra Unreal vydaná května 1998. Velkým pokrokem bylo řešení umělé inteligence nepřátel, kteří dokázali při objasnění uskakovat projektilům, které na ně byly vystřeleny. Tato verze mimo jiné již uměla detekovat kolize s objekty, vykreslovat obsah, a i spravovat souborový systém. V dané době to byl nejlepší nástroj na tvorbu her, jelikož těsně před vydáním se implementovala technologie DirectX, která byla také na vzestupu. Ovšem multiplatformní rozhraní OpenGL se již nedotáhlo do konce, jelikož tam docházelo k velkým rozvojem. Z toho důvodu bylo nedostatek ovladačů na správu Unreal Engine 1. ^[36]

Unreal Engine 1.5

Pozdější verze, která oficiálně přišla v roce 2001 byla pouze takovým vylepšením nebo opravou verze předchozí. Došlo menšímu vylepšení umělé inteligence na spolupráci nepřátel, zlepšení rozlišení a nové podpory platform jako PlayStation 2 anebo Dreamcast. ^[36]

Unreal Engine 2

Zde došlo konečně k rozšíření vývojového prostředí do více žánru, jelikož se přepsal celý kód engine a tím pádem bylo možné vytvářet hry jako MMORPG, RPG anebo adventury. Unreal Engine 2 vyšel roku 2002 a již plně podporoval DirectX a OpenGL. Opět došlo ke zlepšení rozlišení a zároveň bylo nově vytvořené osvětlení objektů, tak aby lampy vrhaly správné světlo a stíny tuto změnu též reflektovaly. ^[36]

Unreal Engine 3

Před touto verzí přišly verze 2, které ovšem neobsahovaly žádné nové funkce, ale pouze opravy anebo optimalizace. ^[36]

Třetí verze byla představena roku 2006 a už umožňovala vytvářet hry všech kategorií, a to jak simulátory, tak i strategie. Tato verze byla nejpopulárnější engine v té době. V této verzi vznikaly jedny z nejlepších her té doby ovšem k digitálním dvojčatům ještě nebyla přizpůsobena. ^[36]

Unreal Engine 4

Byl první verzí, která představila implementaci umění, vědy, architektury, simulací a všeho dalšího za pomoci kódu C++. Představena byla v roce 2012. Byla to velká novinka, protože engine umožňoval vytvářet veškerý obsah a zároveň byl zpřístupněn pro studenty a učitele zdarma na školách či univerzitách. Ačkoliv to přineslo mnoho výhod tak největší, kterou tato verze obsahovala, bylo globální osvětlení. Tento nový přírůstek umožňoval, kde za pomoci algoritmů fyzikálně simulovat šíření světla. Zde se poprvé začalo pracovat s paprsky, jejich odrazy a lomy. Zároveň přišel obchod na Assety, které byly představeny v kapitole o Unity. [36]

Unreal Engine 5

Poslední a aktuální verzi, která vyšla 13. května 2020. Pro digitální dvojčata byla asi největším rozvojem, protože vylepšila možnost Assetů a to, že se ulehčilo importování, a to i CAD modelů. Dále představila kompletně přepracovaný systém osvětlení zvaný Lumen, který umožňuje úpravu světél v reálném čase bez nutnosti vykreslení scény. Druhou velkou novinkou bylo představení systému Nanite, které přepracoval práci s polygony a kompletně změnil pohled na detaily ve 3D prostředí [36]



Obrázek - 18 – Digitální dvojčec v Unreal Engine [44]

V Unreal engineu je modelování jednou z nejjednodušších funkcí, proto se také hodí pro vytváření městských infrastruktur. Je to velice populární nástroj ve videoherním průmyslu, proto je tak hojně využíván. Velikou výhodou jsou ceny licencí, kde se neplatí

žádný poplatek za používání, ale pouze pokud je zisk větší než 1 milión poté již bere 5% z daného výdělku. Více informací je na Tabulce 4 níže. [68]

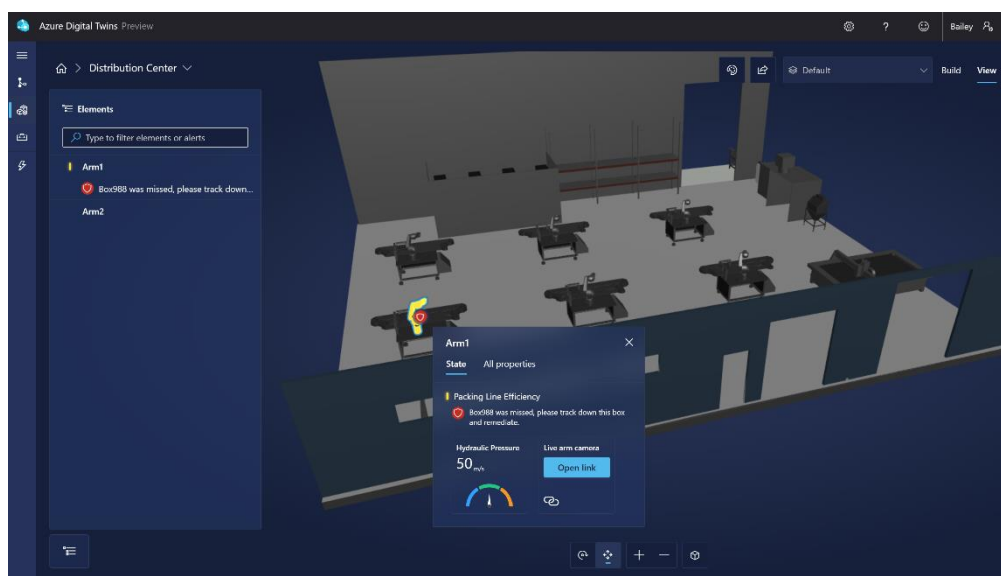
Nástroj	Verze pro studenty?	Verze pro vyzkoušení?	Obsah	Popularita	Cena základní licence
Unreal Engine	Ano	Ano	Modelovací infrastruktura, simulace a efekty	7,5 miliónu uživatelů	Zdarma do vydělání 1 miliónu poté 5% z výdělku

Tabulka 4 - Unreal Engine

3.6.4 Azure

Je cloudová platforma od společnosti Microsoft, která umožňuje přístup k rozšířeným webovým aplikacím a na základě toho lze vytvářet digitální dvojčata. Výhodou je, že uložení dat se nebude ukládat na interní uložení, ale na Cloud. Tím pádem i následná implementace je jednodušší, jelikož se vše nachází na bezpečném místě a není důvod testovat bezpečnost daného softwaru. [42]

První verze byla vytvořena v říjnu 2008 a je napsaná v programovacím jazyku Java. Na obrázku 19 níže vidíme vytvořené modely a hydraulický tlak, kterým simulujeme dané objekty. [43]



Obrázek - 19 - Microsoft Azure [43]

Azure má zde přepočítanou hodnotu z ceny licence, jelikož se do ceny bere více faktorů. Prvním je cena za milión zpráv, která stojí jeden dolar. Druhým je milión provedených operací to vychází na 2.5 dolarů. Posledním je milión dotazů, který stojí 0.5 dolarů. V průměru by takové projekty mohli stát něco kolem 23 500 Kč ročně. Opět záleží na velikosti a množství používání. Veškeré potřebné informace jsou později v Tabulce 5 níže [69]

Nástroj	Verze pro studenty?	Verze pro vyzkoušení?	Obsah	Popularita	Cena základní licence
Azure	Ano	Ano	IaaS (placení za reálné využití), bezpečnost, výkon	715 miliónů uživatelů	23 500 Kč ročně v průměru.

Tabulka 5 - Azure

3.6.5 IBM

Je americká společnost, která je především známá pro svoje vynálezy jako: osobní počítače, programovací jazyk SQL, čárový kód atd. Jako Microsoft Azure, IBM umožňuje tzv. „balíčky“, které se jednotlivě kupují za určitým cílem. Výhodou je, že cena se odvíjí podle toho, kolik toho spotřebujeme, případně kolik toho odesíláme.

Digital Twin Exchange

Je jeden ze zmiňovaných „balíčků“, obsahující nástroj, který je specifický v tom, že umožňuje správu veškerých digitálních dvojčat. Žádná digitální dvojčata se v něm nevytváří, ale koupí se spolu s 3D CAD soubory, materiály a manuály. Digital Twin Exchange funguje jako taková knihovna, která umožňuje firmám nakupovat, prohlížet a spravovat jednotlivá digitální dvojčata. [72]

Maximo® Application Suite

Umožňuje monitorování za pomoci umělé inteligence, jednotlivých IoT zařízení. Slouží také pro kartografických map a je zde opět možnost zaplacení za pouhý obsah, který se využije. Dále ho lze využít jako multi-cloudovou platformu. [72]

Z popisu IBM bychom mohli říci, že je to taková kombinace nástroje Azure a ArcGIS, kde cenově se platí za jednotlivé služby, které potřebujeme a zároveň umožňuje tvorbu kartografických map. Svou popularitu si získal díky historickému pozadí. Více v tabulce 7 níže ^[72]

Nástroj	Verze pro studenty?	Verze pro vyzkoušení?	Obsah	Popularita	Cena základní licence
IBM Exchange	Ano	Ano	Náklady, bezpečnost, kartografie	1 000 000 uživatelů	2 300 Kč ročně

Tabulka 6 - IBM Exchange

3.6.6 ArcGIS

Je vytvořen firmou Esri, která má mnoho aplikací pro jednotlivé použití ze všeho nejvíce nás zajímá varianta ArcGis Desktop, která již v sobě nese většinu aplikací a taky zároveň umožňuje správu a tvorbu digitálních dvojčat.

Rozdělení se větví dle produktu, a to na část zobrazovací (ArcView) a část editovací (ArcEditor). Zobrazovací část je lehce zavádějící, jelikož je zde možnost vytvářet a spravovat prostorová data s kterými následně pracujeme. Editovací je především pro ukládání dat v relační databázi.^[50]

Jednou ze zajímavostí je projekt, který byl představen na Hackathonu roku 2021 jménem Datel. Jeho účelem bylo vytvoření aplikace „Building Infomanagement“ sloužící k „poskytování standardizovaných informací o budovách v kampusu, jejich vybavení a umístění“. Byl vytvořen za pomoci 3 technologií GIS, BIM a CAD.^[51]

Tabulka 6 níže nám ukazuje veškeré shrnutí, které je potřeba si zapamatovat jako tomu je u IBM. ArcGIS umožňuje vytvářet kartografické nástroje také obsahuje technologii GIS, která podporuje geografické informace. ^{[70][71]}

Nástroj	Verze pro studenty?	Verze pro vyzkoušení?	Obsah	Popularita	Cena základní licence
ArcGis	Ano	Ne	GIS software, kartografie	350 000 uživatelů	2 350 Kč ročně

Tabulka 7 - ArcGis

3.6.7 Shrnutí

Ze všech nástrojů nelze vybrat jediný, který by byl nejlepší, protože takový není. Obecně tedy můžeme říct, že každý nástroj je v některých věcech lepší a v jiných zase o něco horší. Právě důvod použití je to, co ovlivňuje preferenci jednoho nástroje nad druhým. Například při výběru nástroje na tvorbu digitálních dvojčat si ještě musíme určit více kritérií. Kdybychom nebyli velká firma, ale pouze jednotlivec, tak je důležité hledět na cenu. Naopak jako firma hledíme na kvalitní obsah.

Z přehledu výše uvedených vývojových nástrojů můžeme říci, že Unity je nejpobulárnější. Azure a IBM digital obsahují IaaS, kde se plátí za reálné využití. Unreal Engine využívá nejlepší modelování. Siemens NX je nejdražší varianta, která má, ale kvalitní obsah a na závěr ArcGis jako výborný GIS software.

Nástroj	Verze pro studenty?	Verze pro vyzkoušení?	Obsah	Popularita	Cena základní licence
Unity	Ano	Ano	Externí knihovny, horší importování vlastních	2,7 billionů uživatelů	48 000 Kč ročně
Azure	Ano	Ano	IaaS (placení za reálné využití), bezpečnost, výkon	715 milionů uživatelů	23 500 Kč ročně v průměru.
Unreal Engine	Ano	Ano	Modelovací infrastruktura, simulace a efekty	7,5 milionu uživatelů	Zdarma do vydělání 1 milionu poté 5% z výtěžku
Siemens NX	Ano	Ne	Standardizované knihovny, sdílené programování	>4,926 firem s 50-200 rozměry lidí	80 800 Kč ročně
ArcGis	Ano	Ne	GIS software, kartografie	350 000 uživatelů	2 350 Kč ročně
IBM Exchange	Ano	Ano	IaaS (placení za reálné využití), Náklady, bezpečnost	>6929 firem s 50-200 rozměry lidí	2 300 Kč ročně

Tabulka 8 - Shrnutí nástrojů

3.7 Modely vícekritériální analýzy variant

Nevýhodou teorie a modelu vícekritériální analýzy variant je výběr jedné či více variant z množiny přípustných a doporučit je k realizaci. V této metodě bývá dosti důležité, aby rozhodovatel zůstal, pokud možno co nejvíce objektivním hodnotitelem k čemuž používá různé aparáty dané analýzy. Častokrát se tedy může zohlednit jiná varianta, než ta objektivně „nejlepší“, která by byla vybraná v praxi.

Kritérium je hledisko hodnocení variant a může nabývat kvalitativních (objektivně měřitelné údaje) nebo kvantitativních (nelze objektivně změřit) podob. Více kritérií ovšem

může značně ovlivnit výslednou dominující variantu, kde se výsledek stává více nepřehledný a taky jednotlivá kritéria již nemusí splňovat podmínku nezávislosti a zároveň se překrývají. Máme dvě povahy kritérií:

- **Kritéria maximalizační:** Nejlepší kritérium má nejvyšší ohodnocení
- **Kritéria minimalizační:** Nejhorší kritérium má nejvyšší ohodnocení.

Dále jsou tu preference, které nám udávají prioritu jednotlivých kritérií na základě, kterých je dále hodnotíme. Úlohy dále také dělíme podle typu informací:

Žádná informace – neexistuje informace o preferencích, nemůžeme tedy určit váhu jednotlivých kritérií a tím pádem nelze danou úlohu řešit.

Nominální informace – rozděluje jednotlivá kritéria na akceptovatelné či neakceptovatelné.

Ordinální informace – každá varianta se ohodnotí podle důležitosti kritérií.

Kardinální informace – je často používaným typem informací, kde se rozlišuje mezi kvalitativním nebo kvantitativním charakterem. Rozhoduje se o kolik je jedno hodnocení lepší než to druhé a v případě preference kritérií to rozlišují váhy v případě konkrétního nejčastěji vyčíslitelného vyjádření to určuje hodnota.

Na tabulce 9 níže je možnost vidět jednotlivé metody a výstupy dle informací. V našem případě pro tvorbu digitálních dvojčat bude vybírána kardinální informace, jelikož máme možnost ohodnotit jednotlivá preferenční kritéria. Následně si jednotlivě popíšeme dvě metody kardinálních informací Bodovací a Saatyho.

Informace o preferencích mezi kritérii		
Informace	Metoda	Výstup
Žádná	Entropická Metoda	Vektor vah kritérií
Nominální	Metoda aspiračních úrovní	Aspirační úrovně kritérií
Ordinální	Metoda pořadí	Vektor vah kritérií
	Fullerova metoda	
Kardinální	Bodovací metoda	
	Saatyho metoda	

Tabulka 9 - Informace o preferencích mezi kritérii ^[62]

3.7.1 Bodovací metoda

Obvykle tuto metodu hodnotí více expertů. Každý expert hodnotí jednotlivé kritérium, kterému přiřazuje určitý počet bodů, a to značí jeho důležitost. Je zde bodovací škála, která se určuje na začátku použití metody. Čím je kritérium důležitější tím více se bodů dává neboli při použití stupnice od 0 do 10 může mít kritérium 10 bodů od experta, podle kterého je nejdůležitější, a 0 bodů od experta, který jí považuje za bezvýznamné. ^[62]

Je velice podobná metodě pořadí, která také určuje prioritu kritérií. Stupnice se dá také zobrazit v grafické podobě, kde se vyjádří za pomoci úsečky. ^[62]

Výsledek se následně může ještě jednou porovnat, kde se původní porovnání vezme pouze jako odhad bodového hodnocení, a následně se znovu posoudí a případně se odstraní nesrovnalosti. Finální součet poté vyhodnocujeme, kde kritérium s nejvyšším číslem je pro nás kompromisní variantou. ^[62]

3.7.2 Saatyho metoda

Pojmenovaná podle Thomase L. Saatyho, který metodu vymyslel. Hodnotí pouze jeden expert, kde se zároveň stanovuje kvantitativní párové porovnání kritérií. V případě

hodnocení více experty už je vhodné využít postup podle metody AHP (Analytický hierarchický proces).^[62]

Stupnice je zde již předefinovaná pro ohodnocení párových kritérií. Pracuje s devítibodovou stupnicí, kde je možnost i hodnotit v mezistupních (hodnoty 2, 4, 6, 8).^[62]

- 1 – kritéria i a j jsou rovnocenná
- 3 – kritérium i je slabě preferované před kritériem j
- 5 – kritérium i je silně preferované před kritériem j
- 7 – kritérium i je velmi silně preferované před kritériem j
- 9 – kritérium i je absolutně preferované před kritériem j

Metoda je nekonzistentní a je zde veliký prostor pro chybu. Čím rozsáhlejší úloha tím je daná metoda méně přesná. Nejčastější chyba je u odhadování poměrů vah, proto zde slouží kontrolní míra konzistence, která měří přesnost podílu.^[62]

3.7.3 Shrnutí

Z důvodu komplexnějšího rozhodování při výběru jednotlivých nástrojů se musí vybrat kardinální informace o preferencích kritérií, je to z důvodu toho, že porovnává jak kvalitativní, tak i kvantitativní typ informací. Kardinální informaci využívá metoda bodovací a funguje podobně jako metoda pořadí, kdy uspořádá prvky dle škály a následně vybírá nejlepší. Saatyho metoda funguje na stejném principu při porovnávání jednotlivých párových kritérií jako bodovací metoda. Z rozboru obou metod se pro komparaci pro jednoho experta jasně jeví lepší využití Saatyho metody, kde dochází i ke kontrole za pomoci indexu míry konzistence.

4 Praktická část

Praktická část se skládá ze 3 kroků

1. Konzistence trendu vycházející z odborné studie
2. Výběr jednotlivých nástrojů a následná komparace
3. Implementace modelu kompromisní variantou

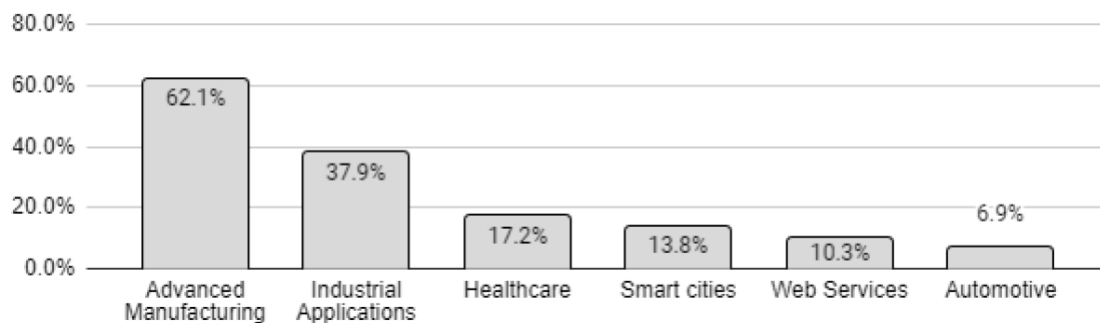
Dle odborné studie bude probrána problematika digitálních dvojčat, a následně se bude na základě těchto poznatků řídit implementace a komparace. Tento postup se provádí, jelikož samotné téma je dost veliké na to, aby se mohlo samostatně celé vypracovat. Je tedy potřeba se řídit určitými podněty vyjmuté z odborné studie. S veškerými materiály a experty by implementace práce digitálních dvojčat zabrala přes měsíc s tím, že průměrná doba tvoření digitálních dvojčat je půl roku. Proto dle studie najdeme trend, který je nejčastější a pokusíme se ho v implementační části nastínit. ^[74]

V dalších krocích bude potřeba si definovat jednotlivé nástroje a za pomoci jednoduchých srovnání, zmenšit počet vývojových nástrojů na minimum, tak aby Saatyho metoda párového porovnání byla prováděna s co nejpřesnějším modelem a nebyla nepřesná. Po výběru se určí kritéria, podle kterých vývojové nástroje budeme komparovat, a určí se jejich váhy, kde zůstanou nejvíc diskutabilní kritéria s co nejmenší vahou. V Závěrečném kroku je potřeba najít kompromisní variantu, s kterou by se následně tvořila implementace.

Implementační fáze, bude vytvořena dle daného vývojového nástroje na základě výsledku vícekritériální analýzy. Následně bude zpracována pomocí vývojových postupů, které se v práci rozebírali.

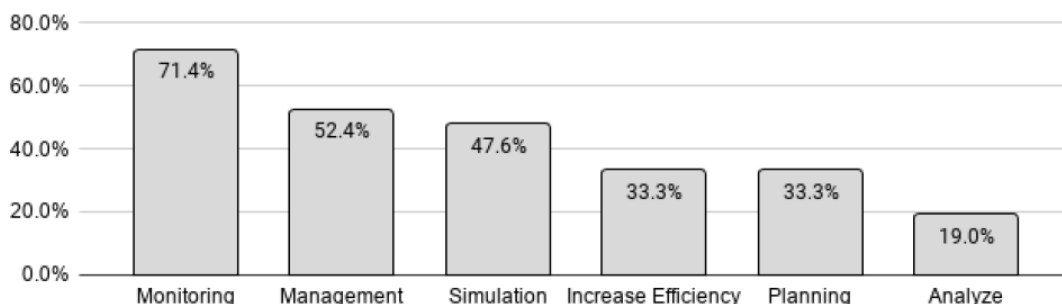
4.1 Konzistence trendu v odborné studii

V odborné studii, která se dá najít ve zdroji 74, jsou podrobné informace o trendech a výhodách digitálních dvojčat, z které se dá následně vycházet. Digitální dvojčata jsou velkým tématem popisující několik technologií zároveň. Pro zjednodušení se tedy z odborné studie vezmou jen některé výsledky, podle kterých budeme dělat implementační a komparační část. ^[74]



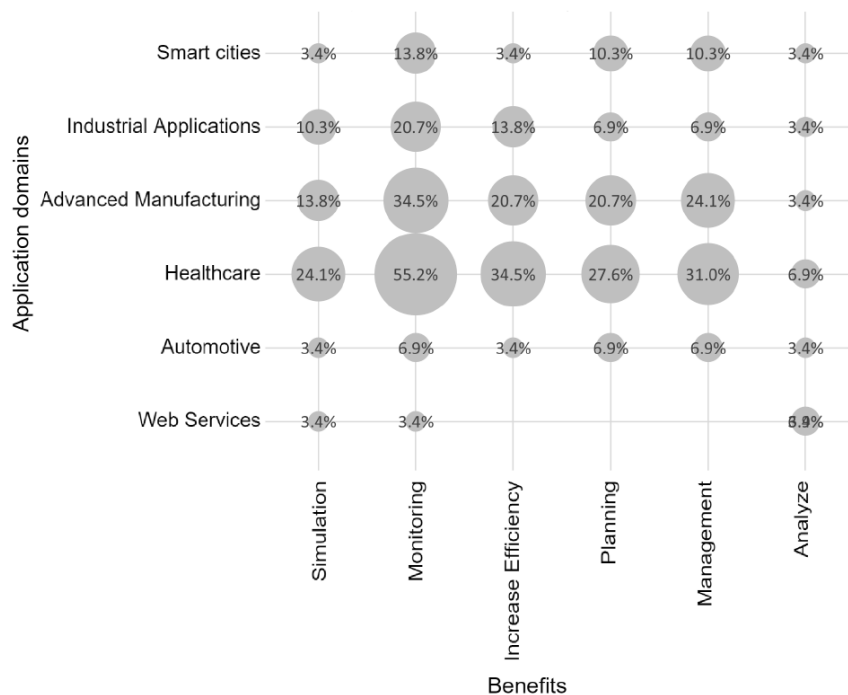
Graf 1 – Oblasti použití digitálních dvojčat

První máme graf 1 výše, který rozebírá nejčastější použití digitálních dvojčat. Jako nejčastější výskyt je v pokročilé výrobě, průmyslových aplikacích, zdravotnictví a v chytrých městech. Z hlediska trendu je hlavním krokem vždy vytvoření modelu, který následně v sobě obsahuje IoT zařízení, například továrny obsahují stroje, které jsou na dálku řízené nebo další příklad je osvětlení, s kterým se dá manipulovat. Ovšem je to pouze jedna část digitálních dvojčat. U továren se mohou například simulovat a předpovídat výsledky produkce, tržby atd. [74]



Graf 2 – Využití digitálních dvojčat

Více pak je zajímavější graf 2 výše, který poukazuje na využití digitálních dvojčat, konkrétně nám říká, co uživatelé při vytváření nejvíce používají. V praxi tedy vidíme, že monitorování je nejvyužívanější funkcí. Je o cca 20% používanější než zbylé funkce jako management nebo simulace. [74]



Graf 3 - Využití digitálních dvojčat v různých oblastech použití

Úplně nejzajímavějším grafem je graf 3 výše, který shrnuje oba předchozí grafy dohromady. Říká nám, v jaké oblasti se nejčastěji využívá jednotlivá funkce. Dle výsledků vidíme, že v každém odvětví se nejvíce používá monitorování. Z druhého pohledu můžeme říct, že nejčastěji se funkce využívají ve zdravotnictví a poté v průmyslové výrobě. Jako trendem tedy je, že ať už je použití jakýkoliv, skoro pokaždé bude aplikována monitorovací část digitálních dvojčat. [74]

4.1.1 Shrnutí

Na základě poznatků z předešlé kapitoly lze konstatovat, že digitální dvojčata se využívají především k monitorování. Ze statistik vyplývá, že až 70 % uživatelů požaduje tuto funkci. V prolnutém grafu 3 výše poté vidíme, že ať už je to zdravotnictví nebo pokročilá výroba, vždy je monitoring žádanou funkcí. Na základě toho je možné tedy usoudit, že trendem digitálních dvojčat je monitorovací část, která se využívá úplně v každém odvětví.

4.2 Selektce vývojových nástrojů

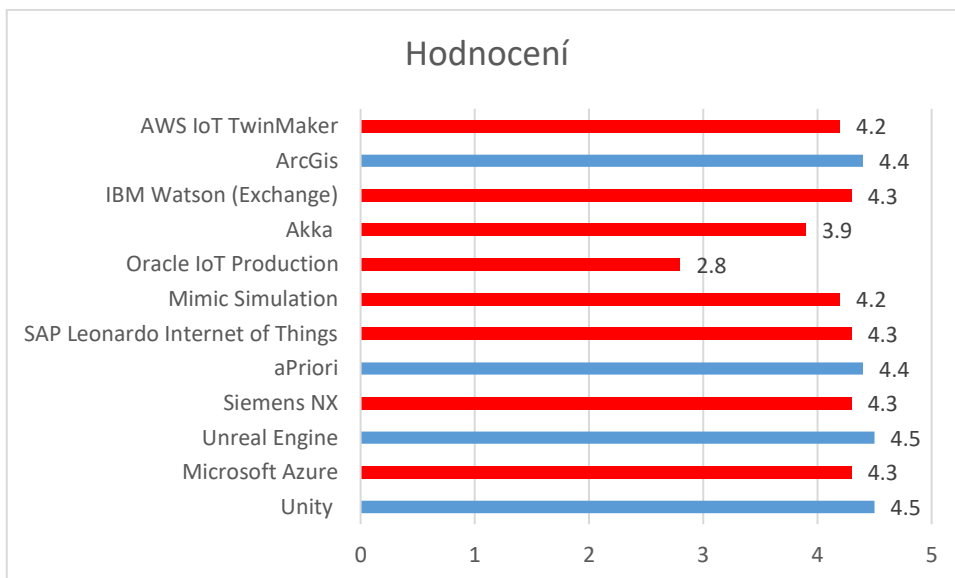
Na základě teoretické části, kde byly zpracovány informace o jednotlivých vývojových nástrojích, se k podrobnějšímu popisu dostalo pouze pár, a to z důvodu obsáhlosti a velkého

počtu vývojových prostředí. Ze statistik o používání se tedy budou vybírat následující nástroje:

- ArcGis
- IBM Watson (Exchange)
- Akka
- Oracle IoT Production
- Mimic Simulation
- SAP Leonardo Internet of Things
- aPriori
- Siemens NX
- Unreal Engine
- Microsoft Azure
- Unity

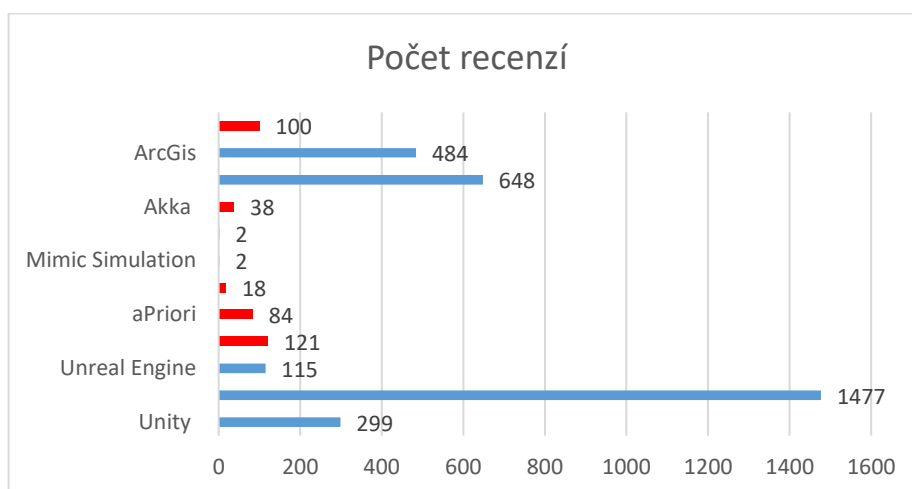
První srovnání, které se provedlo na základě stránky G2^[73], která se specializuje na srovnávání veškerého softwaru v rámci uživatelských zkušeností je hodnocení. Zde se vybrala veškerá data z počtu recenzí a udělal se aritmetický průměr, který nám dal výslednou hodnotu. Na grafu 4 níže vidíme 5 nejlépe hodnocených vývojových prostředí, kterými jsou:

- ArcGis
- aPriori
- Unreal Engine
- Microsoft Azure
- Unity



Graf 4 - Hodnocení vývojových nástrojů

Dle dalšího srovnání se vzal počet recenzí, který nám značí, zda dané hodnocení lze brát v potaz. Pokud by se hodnocení bralo dle minimálního počtu recenzí, tak nelze takové hodnocení brát jako relevantní. K tomuto srovnání byla opět použita stránka G2^[73]



Graf 5 - Počet recenzí vývojových nástrojů

Na grafu 5 výše vidíme, že doopravdy číslo hodnocení není absolutně relevantní na rozdíl od počtu recenzí na daný vývojový prostředí. Pouze prolnutí obou grafu nám dá výsledek, který hledáme. Například při vysokém počtu recenzí s nejnižším hodnocením můžeme usoudit, že takový nástroj určitě brát nebudeme. Vyšel nám tedy výsledek, kde jako dominantní varianta je Microsoft Azure. Jeden z nejhlavnějších důvodů je ten, že Microsoft Azure se nepoužívá pouze na vytváření digitálních dvojčat, ale je to primárně cloudová

platforma, z které obsahuje velikou škálu aplikací. Její součástí je takzvaný „balík“, ve kterém se nachází více služeb. Jedna z nich je naše nejdůležitější, a to je ta pro tvorbu digitálních dvojčat. Na základě tohoto zjištění nelze vybírat vývojová prostředí dle jejich recenzí. U cloudových služeb se dají ovšem vzít pouze jako podklad pro úspěšný software, jelikož daná recenze nemusí být spjata s vývojem digitálních dvojčat, ale z větší části již uživatel, který recenzi psal, tak se setkal s daným softwarem. Tím pádem lze odvodit, že alespoň součástí recenze je vždy hodnocení daného uživatelské rozhraní, ceny licencí, výkon softwaru, problémovost softwaru atd.

Na základě provedené analýzy se odvodila vývojová prostředí, na kterých se použije Saatyho metoda a následně se vybere, kompromisní řešení. Odvozená prostředí jsou:

- ArcGis
- IBM Watson (Exchange)
- Unreal Engine
- Microsoft Azure
- Unity

4.3 Výběr a ohodnocení kritérií

U výběru se bude potřeba držet faktu, že se vybírá pro jednotlivce nikoli pro firmu. Proto tedy zde bude prioritní například cena licence, nebo zda daný nástroj je open source atd. Hlavní prioritou jsou zde náklady. Každé kritérium bude podrobně zdůvodněno a případně i podloženo na základě čeho se vybíralo, ovšem je nutné zde znovu zmínit, že se poté bude opět provádět Saatyho metoda s ohodnocenými kritérii, kde se na základě podmínek určí váha kritérií.

Na základě srovnání se vybralo 5 důležitých kritérií, která jsou dohledatelná na stránkách tvůrce. Jejich výběr bude jednotlivě odůvodněn a následně bude vždy shrnutí ohodnocení. Vybraná kritéria:

- Použitelnost
- Hardwarové požadavky
- Doba učení
- Náročnost UI
- Cena licence

4.3.1 Použitelnost

Je nejdůležitějším kritériem, které bude nést nejvyšší váhu ze všech ostatních. Hlavním důvodem je, že většinou použitelnost je ten parametr, který uživatel hledá. Například pokud se jedná o někoho, kdo nemá žádné volné místo na disku a již vytvořené modely má, tak využije služby IBM Watson nebo Microsoft Azure, které disponují externím uložištěm.

Tabulka 10 níže nám ukazuje ohodnocení nástrojů, dle použitelnosti. V konkrétním případě se zde musí zmínit pár bodů na základě, kterých se bude hodnotit. Prvním a tím nejdůležitějším je monitoring, který tedy obsahuje všechny varianty, což je důležitou funkcí, jelikož na základě ní budeme vytvářet implementaci. Dále potřebujeme modelovací nástroj, abychom si mohli implementaci vymodelovat. Proto tedy v použitelnosti bude dbát na možnost modelace a městskou infrastrukturu. Další funkce je simulace, která ovšem nebude potřebná pro implementaci. Poslední a nejméně důležitou funkcí je externí uložiště, protože se data budou ukládat na disk.

Nástroj	Monitoring	Možnost modelace	Externí uložiště	Městská infrastruktura (GIS)	Simulace
ArcGis	Ano	Ano	Ne	Ano	Ne
IBM Watson	Ano	Ne	Ano	Ne	Ne
Unreal Engine	Ano	Ano	Ne	Ano	Ano
Microsoft Azure	Ano	Ne	Ano	Ne	Ano
Unity	Ano	Ano	Ne	Ano	Ano

Tabulka 10 - Použitelnost

Na základě toho nám vychází hodnocení, které vidíme na tabulce 11 níže. Jedná se o maximalizační hodnocení a Unreal Engine dostává nejlepší hodnotu z důvodu nejlepšího modelování a možnosti například simulace procesu obrábění, které nelze provést v nástroji ArcGis.

Nástroj	Použitelnost
ArcGis	3
IMB Watson	1
Unreal Engine	5
Microsoft Azure	2
Unity	4

Tabulka 11 - Ohodnocení použitelnosti

4.3.2 Hardwarové požadavky

V této části se nejdříve popíší jednotlivé požadavky na nástroje a poté bude přiřazeno ohodnocení do vícekritériální analýzy variant. Hardwarové požadavky jsou potřebným kritériem, které mohou ovlivnit větší míru uživatelů, a to z důvodu například menšího uložení, nedostačující RAM paměti nebo využívání starší verze operačního systému.

Nástroj	CPU	RAM (GB)	Uložiště (GB)	Operační systém
ArcGis	2,6 GHZ 4 Core	8	6	Windows 8.1 Basic
IBM Watson	-	-	-	-
Unreal Engine	2,5 GHZ 8 Core	8	10	Windows 7 x64
Microsoft Azure	-	-	-	-
Unity	x64 s podporou SSE2 (stará technologie z roku 2000)	4	80	Windows 7 x64

Tabulka 12 - Hardwarové požadavky

Jak vidíme z tabulky 12 výše, IBM Watson a Microsoft Azure neobsahují žádná specifika, jelikož se jedná o službu, na kterou postačí jakýkoliv základní počítač s přístupem na internet.

Výsledné ohodnocení na Tabulce 13 níže, vidíme že IBM Watson spolu s Microsoft Azure dostala nejlepší výsledek, jelikož nevyžadují žádné hardwarové požadavky. Unity další v pořadí, protože nevyžaduje náročný procesor. Unreal Engine je obdobný ArcGis až na zásadní detail, a to v podpoře operačního systému, kde již nepodporuje Windows 7 a na základě toho je ohodnocení ArcGis nejhorší.

Nástroj	Hardwarové požadavky
ArcGis	1
IBM Watson	5
Unreal Engine	3
Microsoft Azure	5
Unity	4

Tabulka 13 - Ohodnocení hardwarových požadavků

4.3.3 Doba učení

Je na základě zdrojů ideálním kritériem pro porovnávání daného nástroje, protože je na základě statistik jasně daná většinou vydavatelem. Určuje, jak dlouho trvá uživateli naučit se jednotlivý nástroj a následně toho s ním vytvořit výsledek (pro nás digitální dvojče).

Nástroj	Doba učení
ArcGis	6-12 měsíců
IBM Watson	3-6 měsíců
Unreal Engine	3-6 jednoduchý model, 1-2 roky pro těžší model
Microsoft Azure	2 měsíce
Unity	1-3 týdny

Tabulka 14 - Ohodnocení doby učení

Dle tabulky 14 výše se jedná o minimalizační ohodnocení, které se bude převádět na maximalizační, proto aby veškerá ohodnocení byla na stejném měřítku. U doby učení se bude poté převádět z kvalitativního na kvantitativní hodnotu. Do měsíce = 1, do 2 měsíců = 2, do roka = 3, více než rok = 4, více než 2 roky = 5. Vždy se bere největší hraniční hodnota.

4.3.4 Náročnost UI

Definuje většinou uživatel, a to, jak je uživatelské rozhraní přívětivé. Neboli zda si uživatel může rozhraní jednotlivě upravovat a vytvořit si ho dle libosti nebo zda je každý prvek pevně zamknutý na svém místě. Toto kritérium by bylo nepřesné, kdyby se bral v potaz názor uživatelů. My ovšem vycházíme z běžných parametrů, které program nabízí. Jedním z nich je například možnost editace rozhraní, základní rozložení panelů, kolik prvků se zobrazuje uživateli hned při spuštění a v poslední řadě kolik jich nakonec potřebuje. I přes veškeré parametry dostane náročnost UI nejmenší váhu z toho důvodu, že pro vytvoření monitoringu nemá kritérium skoro žádný vliv.

Nástroj	Náročnost UI
ArcGis	5
IMB Watson	1
Unreal Engine	3
Microsoft Azure	2
Unity	4

Tabulka 15 - Ohodnocení náročnosti UI

Na tabulce 15 výše vidíme, že se jedná o maximalizační ohodnocení. V tabulce vidíme, že ArcGis má nejvíce přívětivé rozhraní pro nového uživatele z toho důvodu, že je obdobné Microsoftu Office. Zkrátka si uživatel nemusí na nic nového zvykat a je jako ryba ve vodě. Unity je ohodnoceno čtyřkou z důvodu přehlednosti, kdy pro každou věc je zde většinou nové okno. Toto už může být pro nového uživatele matoucí a častokrát se v tom ztratí. Zbytek je již jednoduchý na naučení. Unreal Engine je ohodnocen trojkou, jelikož má velkou škálu funkcí a je potřeba s nimi strávit opravdu hodně času, než se dají naučit nějaké základní dovednosti. Microsoft Azure, dostává horší ohodnocení, jelikož software, který využívají je velice nepřehledný s tím, že pro uživatele je velice náročné poznat, jaké balíčky potřebuje pro vytváření modelů a jaké pro monitoring atd. IBM Watson dostává nejhorší, a

to z hlediska, že pouze obsahuje prostředí, ve kterém se dají digitální dvojčata spravovat a následně monitorovat v tvorbě a možné implementaci.

4.3.5 Cena licence

Zde si budeme muset určit z jakého hlediska se díváme na licenci. Pokud jako student, tak veškeré nástroje jsou zdarma, ovšem větší procento lidí jsou obyčejnými občany a tak z důvodu všeobecnosti se v této práci bude zabývat klasickou licencí pro všechny. Proto z výsledků, které se odvodily v práci, můžeme vzít tabulku 16 níže, která nám zobrazuje jak na tom, vybraný nástroj je a následně ohodnotit toto kritérium do Saatyho metody.

Nástroj	Cena licence
ArcGis	Placené
IMB Watson	Středně placené
Unreal Engine	Zdarma
Microsoft Azure	Středně placené
Unity	Zdarma

Tabulka 16 - Ohodnocení ceny licence

Na základě tabulky 16 výše vidíme, že se jedná o minimalizační ohodnocení, které se poté bude převádět na maximalizační, proto aby veškerá ohodnocení byla na stejném měřítku. Konkrétně v ceně se jedná o převedení do kvantitativních hodnot. Placené = 5, Středně placené = 3 a Zdarma je jako nejlepší za 1.

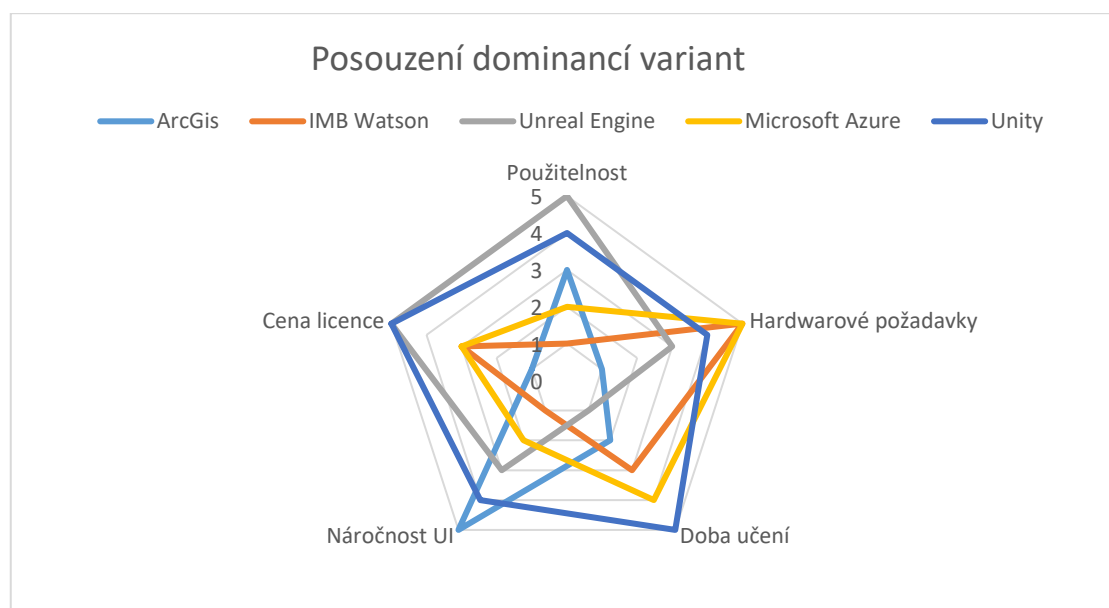
4.4 Vícekriteriální analýza variant – Stanovení kompromisní varianty

Za pomoci ohodnocení kritérií se veškeré hodnoty dostaly do tabulky, které poté byly převedeny na maximalizační tvar viz tabulka 17 níže. Už z tohoto pohledu můžeme usoudit, že v bodovací metodě bude vývojové prostředí Unity nejlepší variantou, jelikož ve všech kritériích dosahuje na nejvyšší příčky.

	Použitelnost	Hardwarové požadavky	Doba učení	Náročnost UI	Cena licence
ArcGis	3	1	2	5	1
IMB Watson	1	5	3	1	3
Unreal Engine	5	3	1	3	5
Microsoft Azure	2	5	4	2	3
Unity	4	4	5	4	5
Charakter kritéria	Max	Max	Max	Max	Max

Tabulka 17 - Maximalizační tabulka ohodnocení

Posouzení dominancí variant nám dává větší přehled o tom, jaké varianty jsou dominantní a jaké jsou dominovány ostatními. Na tomto základě se vytváří graf, podle kterého se usuzuje o variantách. V grafu 6 níže vidíme, že IBM Watson je dominována variantou Microsoft Azure, jelikož ve všech oblastech zaostává nad danou variantou a mohl by se z analýzy odstranit.



Graf 6 - Posouzení dominancí variant

Příloha 1 obsahuje stanovení vah kritérií neboli Saatyho metodu párových porovnání. Metoda nám určuje váhu jednotlivých kritérií a následně i její přesnost za pomoci indexu konzistence, který musí být $IS < 0,1$. Při porovnání hodnot se následně určilo, že použitelnost má největší preferenci neboli váhu $s = 0,53$. Poté hardwarové požadavky $s = 0,28$ a zbytek je již po menších hodnotách, jelikož jejich důležitost nenabývá hodnot potřebných pro výslednou implementaci.

Závěrečným krokem tedy bylo provést výběr kompromisní varianty za pomoci bodovací metody s vahami viz tabulka 18 níže. Bodovací metoda vyšla podle předpokladů, kdy nejvíce bodů dostalo vývojové prostředí Unity. Ovšem kde už je to těsnější, je při bodovací metodě s vahami, kdy je sice kompromisní variantou Unity, ale Unreal Engine je mu těsně v patách s rozdílem 0,13 bodů a při výběru bychom i o této variantě mohli

Výběr kompromisní varianty pomocí:

Bodovací metoda s vahami

Matice bodování variant dle kritérií

	Použitelnost	Hardwarové požadavky	Doba učení	Náročnost UI	Cena licence	Bodovací metoda	Bodovací metoda s vahami	Pořadí
ArcGis	3	1	2	5	1	12	2,26	5
IMB Watson	1	5	3	1	3	13	2,45	4
Unreal Engine	5	3	1	3	5	17	4,03	2
Microsoft Azure	2	5	4	2	3	16	3,09	3
Unity	4	4	5	4	5	22	4,16	1
Charakter kritéria	max	max	max	max	max			
	0,53	0,28	0,08	0,03	0,08			

přemýšlet. Zbylé nástroje jsou vždy v něčem nevýhodné. Specificky ArcGis, kde licencemi a hardwarovými požadavky je na posledním místě.

Tabulka 18 - Bodovací metoda s vahami

4.4.1 Shrnutí

Při výběru vývojových nástrojů, bylo ze začátku porovnáváno větší množství a na základě toho se dle počtu recenzí a hodnocení vybralo pouze pět z nich. Následně se vybrala kritéria, která sloužila k porovnání vývojových nástrojů ve vícekritériální analýze variant. Závěrem tedy bylo možné provést Saatyho metodu, kde se ohodnotila kritéria mezi sebou, určil se index konzistence. Závěrem byl výběr kompromisní varianty za pomoci bodovací

metody s vahami, kde jako nejlepší možností vyšlo vývojové prostředí Unity a hned těsně za ním byl Unreal Engine s rozdílem 0,13 bodů.

4.5 Implementace

V implementační části se bude postupovat na základě vodopádového modelu, jelikož zde je zákazník zároveň vývojářem, tudíž cíl je předem a jasně definovaný a můžeme tedy model dodržet. Dalším krokem bude využít monitorovací část u digitálních dvojčat vycházející z trendu rozebraného již v kapitole 4.1. Při vytvoření monitoringu se ještě bude požadovat vytvoření dashboardu, který dává širší přehled o datech a vytváří tak perfektní část pro monitoring digitálních dvojčat.

4.5.1 Požadavky

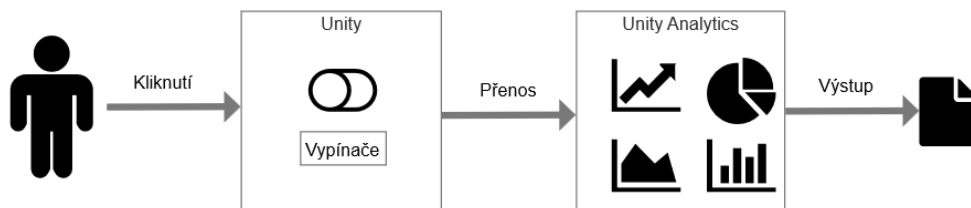
Na základě požadavků stavíme celý projekt a určujeme, co je a není reálné. Pro nás tedy bude hlavní vytvořit příklad monitorovacího zařízení IoT, kterým by disponovalo digitální dvojče. V praxi bychom mohli použít například jednoduchý model, kde máme budovu a skrz všechny místnosti budeme snímat počet zapnutých světel (v našem případě se bude jednat o jedno konkrétní světlo). Na základě toho můžeme nastavit požadavky, které se musí skrz projekt dodržet:

- Monitoring světla neboli spotřeba energie.
- Využit k tomu dashboard („uživatel“ nebude chtít koukat do Unity pro své data)
- Dashboard bude víceúčelový, tedy nebude sloužit jenom pro monitoring světel.

Ke konci projektu bude shrnuto, zda požadavky byly splněny.

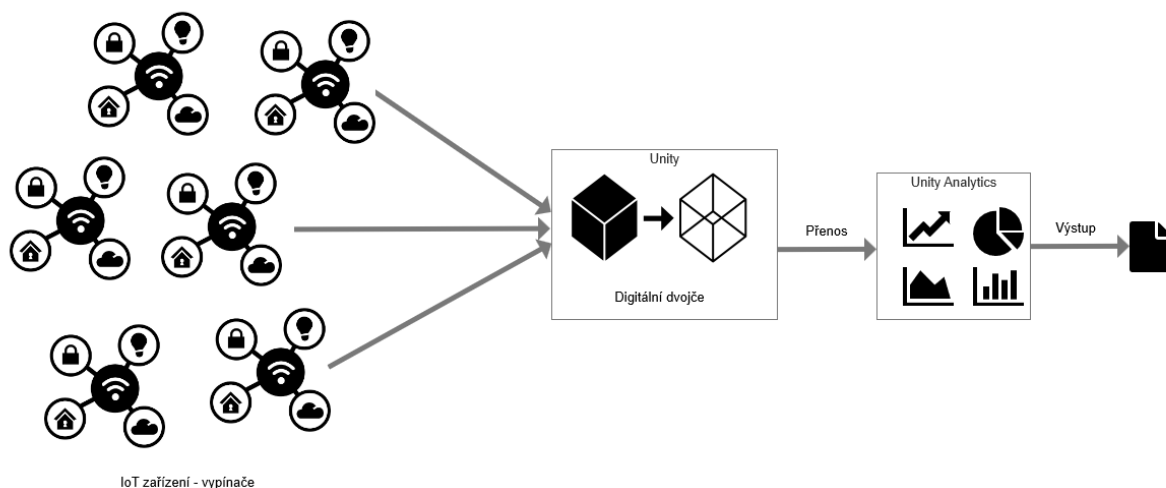
4.5.2 Návrh

Součástí modelu se vytvoří jednoduchý návrh sloužící o lepší představě, kam vývoj bude směřovat. Za pomoci Axure RP, který se využívá pro vytváření prototypů, specifikací, diagramů atd. vytvoříme návrh modelu, který nám dá lepší představu o tom, jak budeme postupovat.



Obrázek - 20 - Návrh modelu

Na obrázku 20 výše vidíme uživatele, který za pomoci kliknutí posílá informace do Unity o tom, zda je nebo není vypínač zapnutý a na základě toho přenášíme data do Unity Analytics. Odtud dostáváme veškeré informace o tom, kolikrát byl vypínač stisknut nebo jak dlouho uživatel nechal spínač zapnutý atd.



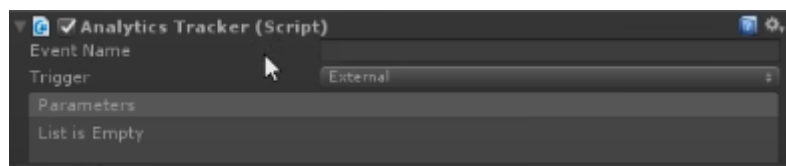
Obrázek - 21 - Reálný návrh modelu

Pokud bychom nedělali teoretický model, ale brali v úvahu, že již máme předem vytvořené digitální dvojče viz obrázek 21 výše, které představuje například hotel, tak veškeré vypínače na hotelu budou obsahovat senzor, které po spuštění předají informaci o tom, že jsou spuštěny a zaznamenávají čas. Proto je zde velký rozdíl, kde v našem modelu disponuje hlavní rolí člověk, který musí vypínač zapínat a vypínat, ale v reálném modelu již role člověka vůbec není potřeba.

4.5.3 Vývoj

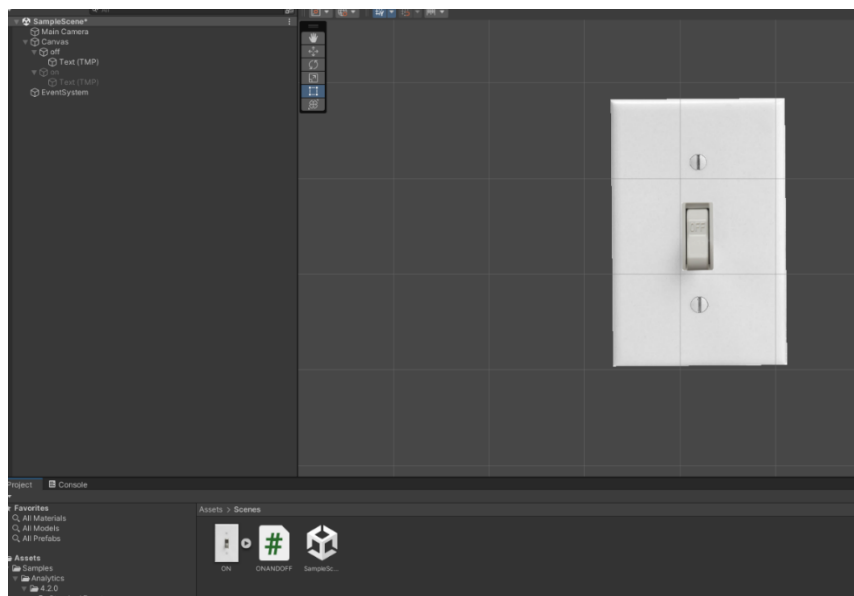
Než se dáme do vývoje v Unity, tak je nejdříve potřeba nastavit Unity Analytics. K vytvoření je potřeba si zažádat přes webovou stránku a provést platbu, která není zdarma, nýbrž IaaS (Integrace jako služba), kde za určitý výdělek jde procento povozníkům služby. Veškeré nastavení lze najít v příloze 2, která obsahuje jak nastavení Unity, tak nastavení dashboardu, aby mohl fungovat.

Dále nastavení musí vypadat jako na obrázku 22 níže. Kde nastavíme externí spouštění, aby se tato akce splnila, pokaždé když se z vnější zavolá. Pro nás to tedy bude na stisknutí tlačítka.



Obrázek - 22 - Přesun dat

Na základě toho vytvoříme 2 tlačítka, obě budou úplně stejná až na jejich funkci. Tlačítko „off“ bude fungovat stylem, že zašle údaje o tom, že jsme zmáčkli vypínač tím pádem, zhasínáme světlo a v dashboardu následně vidíme počet stisknutých tlačítek. A tlačítko „on“. Základní design tedy je vidět na obrázku 23 níže, kde tlačítko „on“ je ze začátku vypnuté, které se později zapíná až ve skriptu.

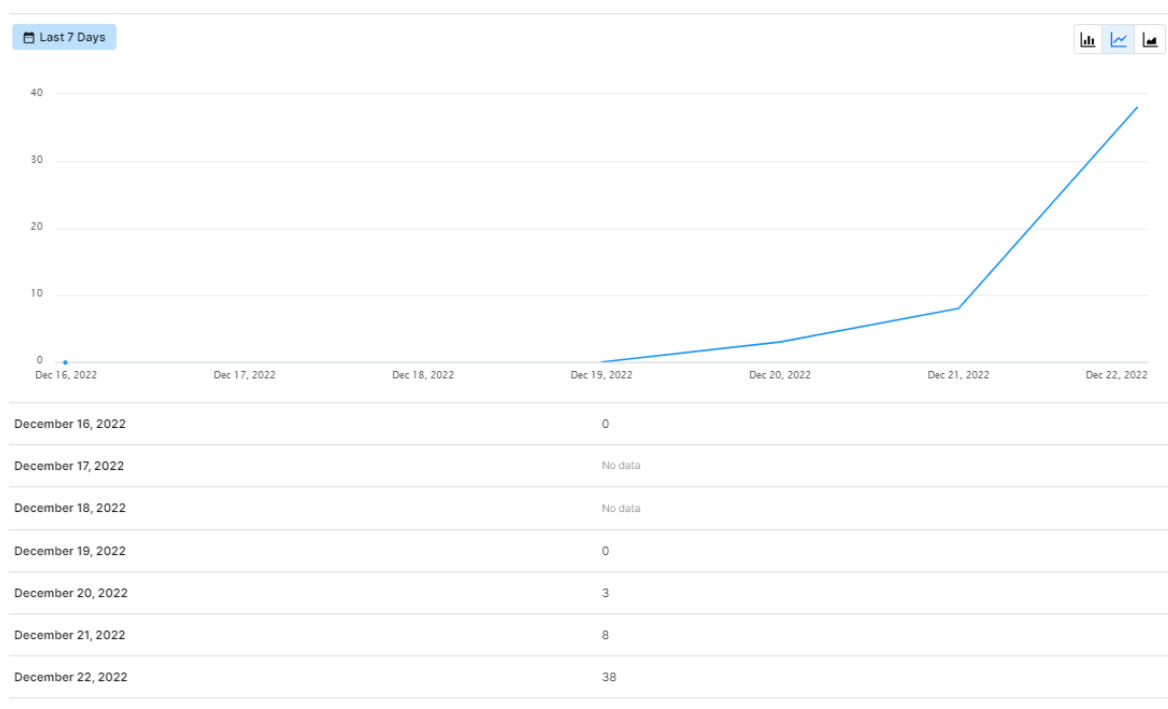


Obrázek - 23 - Design modelu

Na zobrazování a skrývání vytvoříme jednoduchý skript, který nám bude sloužit pro obě tlačítka. Kde se na základě kliknutí aktivuje tlačítko druhé a tím pádem můžeme proklikávat vypínače, tak jako by tomu bylo v reálném světě. Jednoduchý kód je v Příloze 3.

4.5.4 Verifikace

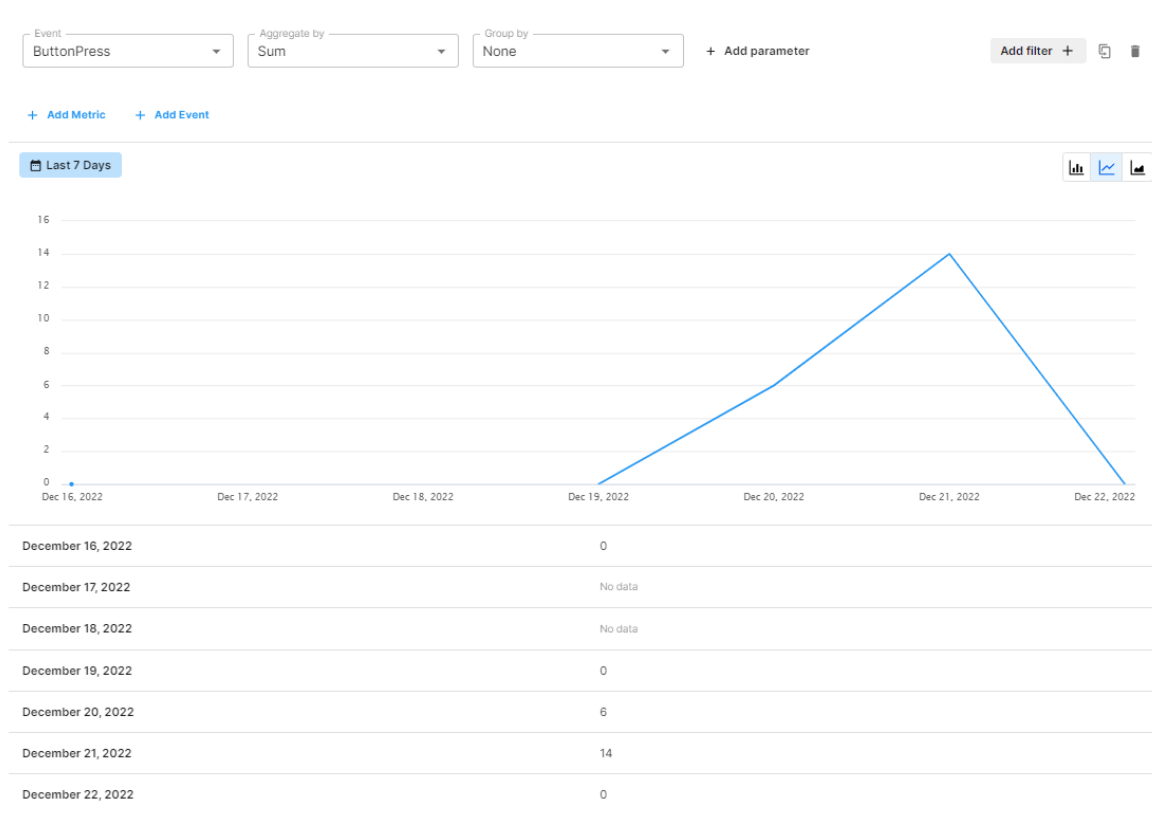
Je část, kde se již jenom potvrdí výsledky implementace a dojde ke schválení funkčnosti modelu, tak jak bylo na základě požadavku stanoveno. V principu jde o jednoduchý vývoj, který má být pouze příkladem k modelu reálnému. Při stisknutí tlačítka se tedy započítává doba strávená, kdy světlo svítí. Při vypnutí modelu končí provozní doba. Tím to by se měřila doba spotřeby energie skrz všechny IoT zařízení v našem projektu. Takovýto údaj můžeme měřit tedy jako dobu, kdy byly světla zapnuta. To můžeme považovat za relevantní čas, se kterým lze později pracovat. Dobu od zapnutí světla je označena na obrázku 25 níže.



Obrázek - 24 - Doba od zapnutí světel

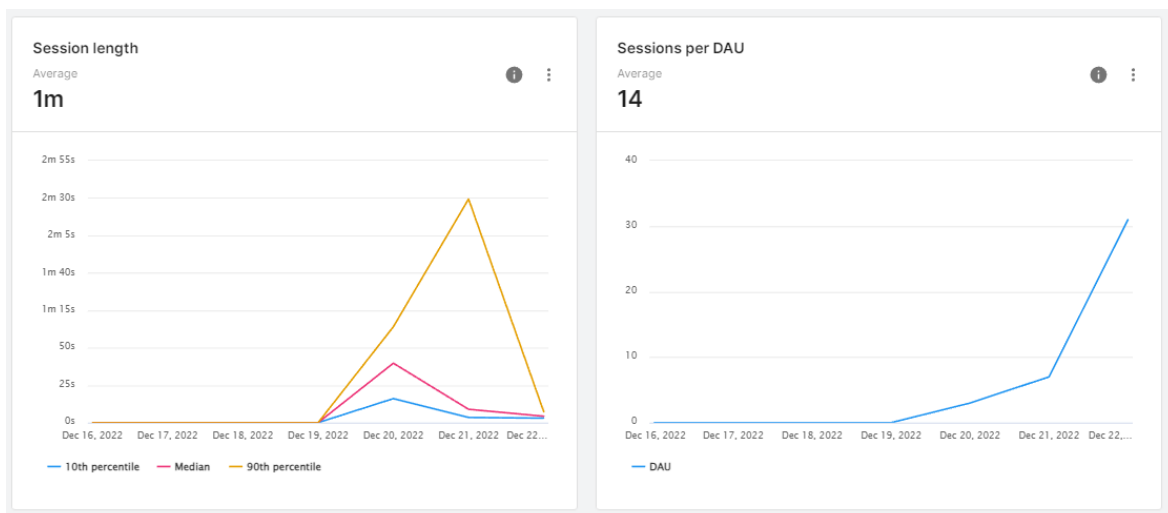
Zpět tedy k požadavkům, o které bylo žádáno. Na obrázku 26 níže vidíme počet stisknutí vypínače. Na základě události ButtonPress, který dostáváme skrz hodnotu v Unity dokážeme změřit počet kliknutí, které uživatelé namačkali. Ve větším modelu by toto znamenalo, že by bylo možné měřit, jaká světla se využívají a jaká jsou naprosto zbytečná,

jelikož například do grafu nikdy nevstoupila a tím pádem je nikdo nezmačkl. Ovšem toto je zavádějící, jelikož i když to světlo tam je, tak nikomu nevadí, že se s ním nesvití, ale například často mačkaný vypínač světla by mohl signalizovat, že je nutno koupit silnější a výkonnější světla, protože se hojně využívají.



Obrázek - 25 – Počet stisknutí vypínače

Posledním požadavkem bylo mít k dispozici dashboard spolu s ostatními daty. Unity Analytics je zde velice omezený, jelikož vidí z větší části data, která jsou zaměřená na herní průmysl, tím jsou tedy počet kliknutí na reklamy, množství přihlášených uživatelů atd. Ze zajímavějších informací tam pár grafů je, jako například délka jedné „směny“ tzv. jak dlouho bylo světlo zapnuté. V neposlední řadě je možné si zobrazit průměrný počet uživatelů, kteří světlo zmáčknuli. Grafy z Unity Analytics jsou vidět na obrázku 27.



Obrázek - 26 - Unity Analytics dashboard

Pokud by grafy v dashboardu nestačily jako je na obrázcích 25, 26 a 27, je zde ještě možná alternativa a to vzít data z Unity Analytics a přetřansformovat je do CSV formátu, se kterým by se dalo pracovat později v ideálním nástroji přímo zaměřeném na zobrazování Business Inteligence jako je Power BI. Z výstupu, který Unity převádí do CSV formátu, jak lze vidět na obrázku 28 níže, je možné následně importovat data do Power BI a zde již vytvářet funkční dashboard se všemi údaji o energetické spotřebě.

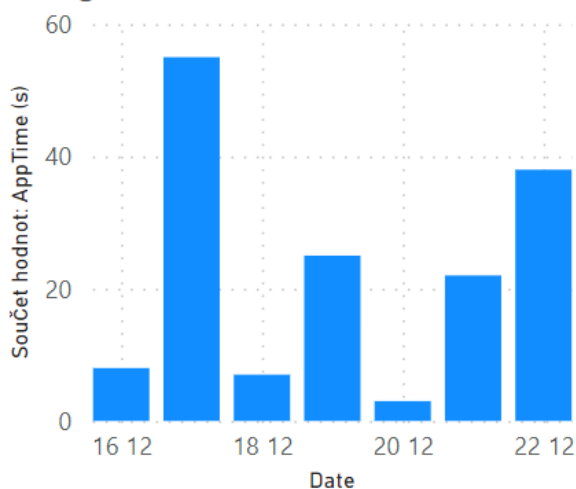
A	B	C
Date,ButtonPress Sum	Date,AppTime	
2022-12-16,-	2022-12-16,-	
2022-12-17,-	2022-12-17,-	
2022-12-18,-	2022-12-18,-	
2022-12-19,-	2022-12-19,-	
2022-12-20,6	2022-12-20,3	
2022-12-21,14	2022-12-21,8	
2022-12-22,-	2022-12-22,38	

Obrázek - 27 - Export dat z Unity Analytics

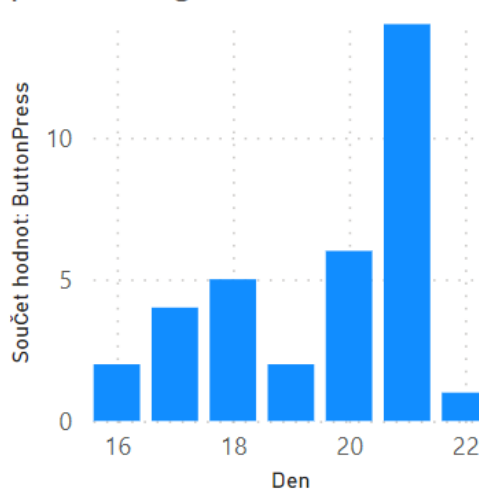
Posledním krokem verifikační části je vzít údaje z CSV formátu a přenést je do funkčního dashboardu, který by mohl vypadat jako obrázek 29 níže. Graf vlevo nahoře nám ukazuje spotřebu energie za jednotlivé dny a je možné monitorovat jaký dny jsou energeticky nejnáročnější anebo nejméně náročné. Vpravo nahoře bereme počet stisknutých vypínačů, neboli kolikrát za den se světla zapínají. Zde se dá sledovat více prvků. Prvním by mohlo být jako v předešlém grafu, a to jaký den se svítí nejvíce (ovšem zde nevíme, jak dlouho se svítilo). Druhým, jaké světlo je nejpoužívanějším a na základě toho optimalizovat

vypínač/světlo. Vrchní dva grafy jsou tedy na stejném principu a jediné co je dělí je osa Y. Poslední dva grafy již v sobě mají aplikovanou funkci průměru, která nám již dává zajímavější pohledy. Vlevo dole vidíme graf, který ukazuje časovou spotřebu na počet stisknutí a z hlediska jak to vidíme, můžeme usoudit, že počet stisknutí vypínače definitivně nemá vliv na dobu, po kterou je vypínač zapnutý. Poslední graf vpravo dole nám ukazuje převrácenou logiku, kde akorát vidíme průměr stisknutých tlačítek k časovému intervalu.

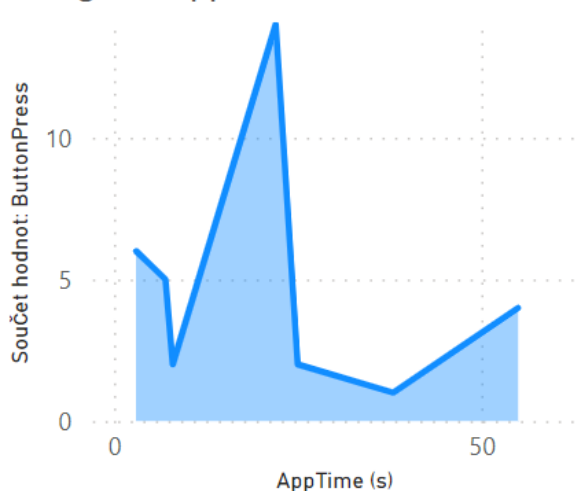
Součet hodnot: AppTime (s) podle kategorie Date



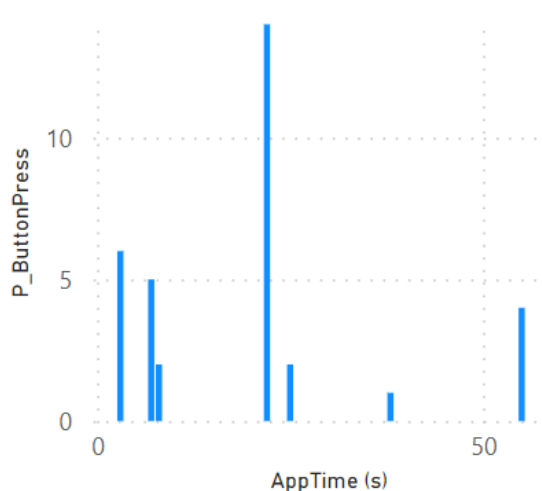
Součet hodnot: ButtonPress podle kategorie Den



Součet hodnot: ButtonPress podle kategorie AppTime (s)



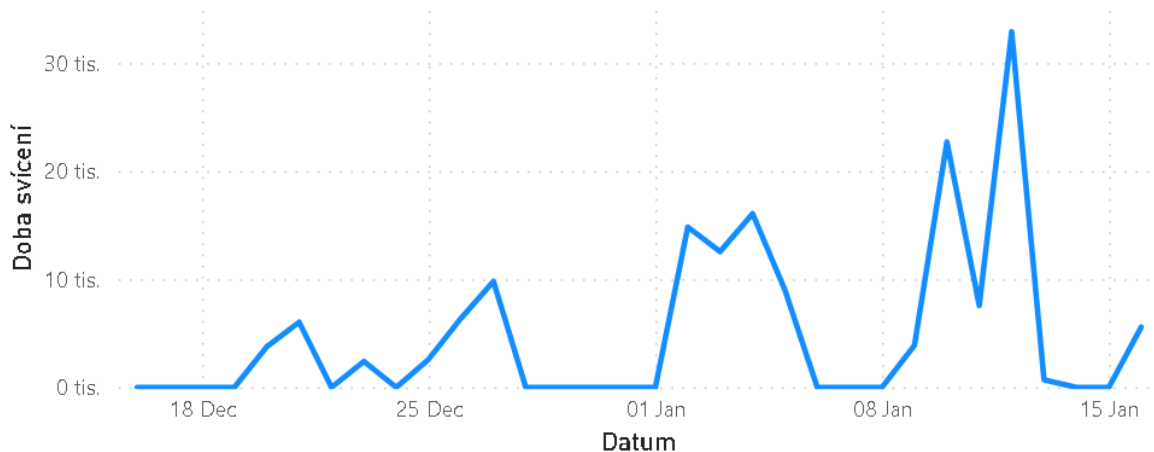
P_ButtonPress podle kategorie AppTime (s)



Obrázek - 28 - Dashboard Power BI

Při rozšíření a posílání dat po celý měsíc už dashboardy vypadají lépe, jako na obrázku 7 níže. V grafu vidíme dobu svícení za celý měsíc, kde je vidět, že o svátcích a víkendech se nesvíí.

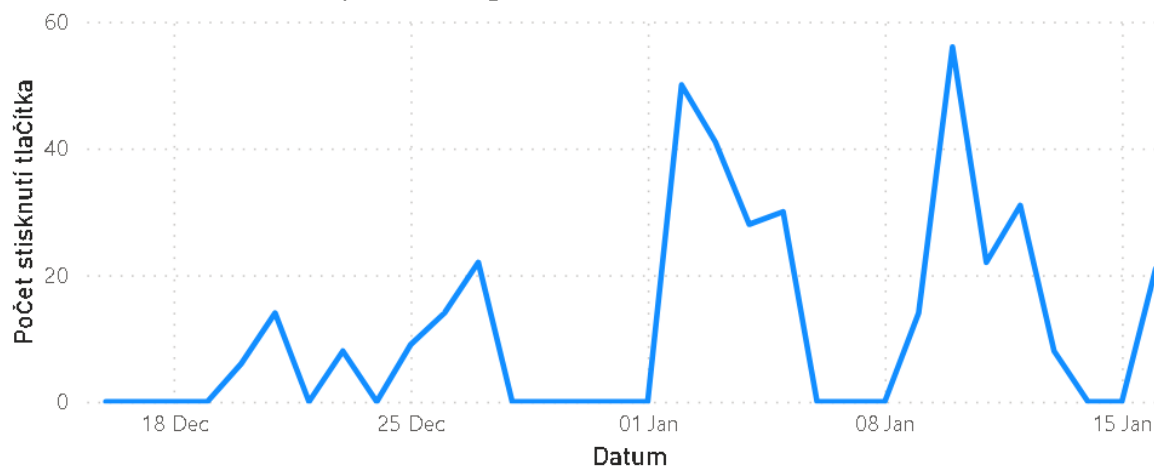
Doba svícení podle kategorie Datum



Graf 7 - Doba svícení

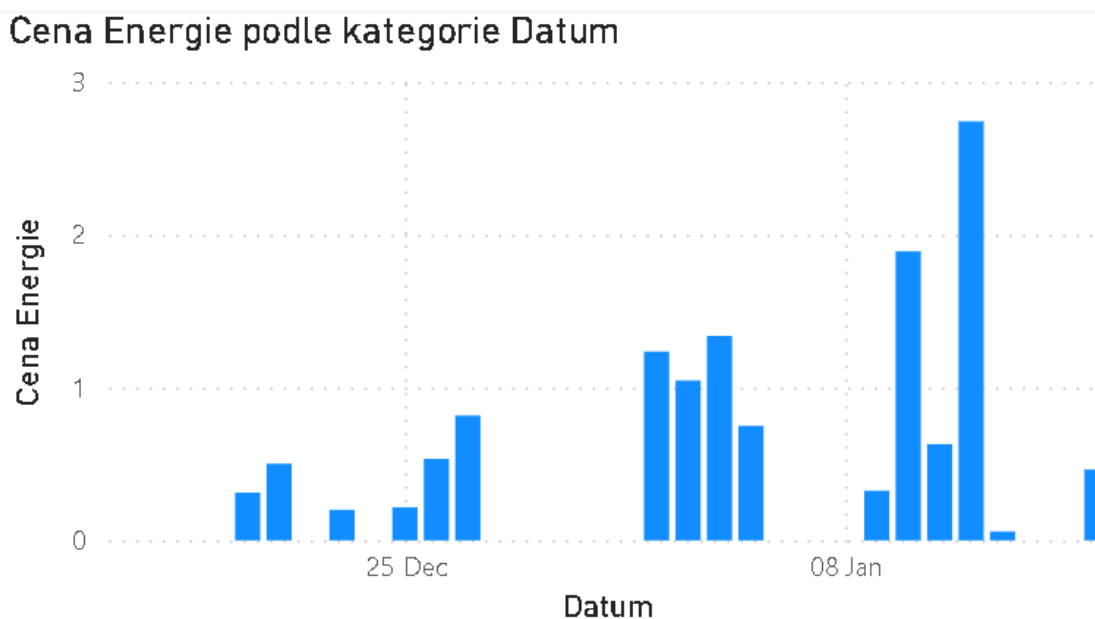
Na grafu 8 níže, vidíme počet stisknutí spínače za daný měsíc. Na první pohled je vidět, že jsou grafy 7 a 8 velice podobný v rámci doby, kdy se používali. Hlavní věcí, kterou z těchto dvou grafů můžeme zjistit je, jak často se světlo používá. Můžeme též usoudit, zda se dané světlo častokrát zapíná a vypíná nebo se jednou zapne a vypne až ke konci dne. V našem příkladu vidíme, že světlo je často zapínané a vypínané, tudíž je hojně využíváno. Doporučení v tomto případě by bylo použít žárovku s větší kapacitou, aby se předešlo častým výměnám.

Počet stisknutí tlačítka podle kategorie Datum



Graf 8 - Počet stisknutí tlačítka

Finální výsledek můžeme shrnout do grafu 9 níže, který nám říká veškeré potřebné informace, které bychom kdy mohli využít. Určuje nám Cenu energie za daný datum. Kdy se vypočetla doba používání převedená na hodiny. Následně se vypočetla kilowatthodina, kdy jedna hodina průměrné spotřeby žárovky je 0,06 kWh. Tento výsledek poté vynásobíme průměrnou spotřebou energie, která činila v roce 2022 5 korun za 1 kWh. Z výsledku tedy vidíme, že nejrizikovější den máme cenu energie za 2,7 Kč neboli 0,54 kWh.



Graf 9 - Cena energie

4.5.5 Shrnutí

Celá implementace byla řízena na základě vodopádového modelu, který mohl být dodržen, protože je zde zákazník roven vývojáři. Na základě toho byly provedeny úkony, které stanovily 3 základní požadavky modelu. Monitoring světla, využít k tomu dashboard a dashboard bude víceúčelový tedy nebude sloužit pouze pro monitoring světla. Pro samotný návrh se udělal jednoduchý model a zároveň i model, jak by vypadal ve skutečnosti. Implementační část celého modelu byla součinnost Unity a Unity Analytics, ve kterých se provedly veškeré požadavky, které byly stanoveny. Následně se ještě provedl export dat ve formátu CSV, protože lépe fungující víceúčelný dashboard lze vytvořit v Power BI, který tam byl také následně vytvořen. proveden. Závěrem můžeme tedy říct, že na základě trendu monitorování digitálních dvojčat, byl úspěšně zvládnut zkušební příklad, který by mohl sloužit jako podklad pro tvorbu monitorovacího funkčního dashboardu.

5 Závěr

Digitální dvojčata jsou technologií, která je tady s námi už mnoho let, ale stále je to málo probádána oblast s velkým potenciálem. Je to neustále se vyvíjející se proces, do kterého má smysl investovat. Cílem této práce byla charakteristika nástrojů na tvorbu digitálních dvojčat. Byly zde popsány ty nejpoužívanější nástroje a následně se zvolila kritéria na základě podložených zdrojů a vyhotovily se váhy podle Saatyho metody párových porovnání.

Zhodnocení vybraných softwarů bylo provedeno dle vícekritériální analýzy variant a jako nejvíce vyhovující variantou podle zvolených kritérií vyšlo vývojové prostředí Unity podle, kterého byla provedena implementační část.

Na základě odborné studie byl nalezen trend, který zkoumá problematiku monitorovací části digitálních dvojčat a následně na základě této části se tedy vybrala implementace modelu monitoringu. V rámci následného postupu vývoje byl zvolen vodopádový model, kvůli tomu, že zde neexistuje zákazník, ale pouze vývojář. Tím pádem veškeré požadavky a návrhy jsou stanoveny, tak jak mají být provedeny. Dle požadavků se stanovily 3 základní cíle modelu. Monitoring světla, využití dashboardu a dashboard bude víceúčelový tedy nebude sloužit pouze pro monitoring světla. Pro samotný návrh se udělal podrobný model a zároveň i model, jak by vypadal ve skutečnosti. Implementační část celého modelu byla součinnost Unity a Unity Analytics, v kterých se provedly veškeré požadavky, které byly stanoveny. Následně se ještě provedl export dat ve formátu CSV, protože lépe fungující více účelný dashboard lze vytvořit v Power Bi, který tam byl také následně vytvořen. Můžeme tedy říct, že na základě trendu monitorování digitálních dvojčat, byl úspěšně zvládnut model, který by mohl sloužit jako podklad pro tvorbu monitorovacího funkčního dashboardu.

Výsledným doporučením pro monitorování digitálních dvojčat je použít software Unity na základě, kterého dále vytvářet podrobnější dashboard například v programu Power BI. Závěrem lze tedy říct, že všechny stanovené cíle této práce byly naplněny.

6 Seznam použité literatury

- [1] Digital Twin Driven Smart Manufacturing - Fei Tao, Meng Zhang, A.Y.C. Nee - ISBN: 9780128176306 Květen 2022 [cit. 2022-05-06].
- [2] Digital Twin, Kevin Chen, ROYAL COLLINS PUB CO, 2021, ISBN: 1487804865 Květen 2022 [cit. 2022-05-06].
- [3] Digital twin technology, Taylor & Francis Ltd, 2021, ISBN: 9781000455878 Květen 2022 [cit. 2022-05-06].
- [4] E-learning – Učení (se) s digitálními technologiemi: Wolters Kluwer ČR, Praha 2021, ISBN: 978-80-7676-175-9 Květen 2022 [cit. 2022-05-06].
- [5] Twi-Global – digitalni-dvojcata rozdíl [online]. Květen 2022 [cit. 2022-07-06]. Dostupné z: <https://www.twi-global.com/technical-knowledge/faqs/simulation-vs-digital-twin>
- [6] resources – CFD [online]. Květen 2022 [cit. 2022-05-08]. Dostupné z <https://resources.sw.siemens.com/cs-CZ/video-cfd-simulation>
- [7] completecz – CFD [online]. Květen 2022 [cit. 2022-05-08]. Dostupné z <https://www.completecz.cz/datacentra/datacentrum-sluzby-pro-datovacentra/cfd-analyza-proudeni/>
- [8] simscale – Fem – metoda konečných prvků [online]. Květen 2022 [cit. 2022-05-22]. Dostupné z <https://www.simscale.com/blog/what-is-finite-element-method/>
- [9] simscale – obrázek ojnice [online]. Květen 2022 [cit. 2022-05-23]. Dostupné z https://www.simscale.com/projects/simscale/tutorial_1_connecting_rod_stress_analysis/
- [10] rascasone – IoT [online]. Květen 2022 [cit. 2022-05-25]. Dostupné z <https://www.rascasone.com/cs/blog/iot-internet-veci-definice-produkty-historie>
- [11] Networkworld – IoT [online]. Květen 2022 [cit. 2022-05-25]. Dostupné z <https://www.networkworld.com/article/3280225/what-is-digital-twin-technology-and-why-it-matters.html>
- [12] Digitální dvojče: Vůdčí technologie inteligentního průmyslu - Vše o průmyslu. Portál pro moderní výrobu - Vše o průmyslu [online]. Copyright © 2020 TRADEMEDIA INTERNATIONAL. Všechna práva vyhrazena. Červen 2022 [cit. 2022-06-01]. Dostupné z:

- <https://www.vseoprmyslu.cz/digitalizace/digitalni-prototypovani/digitalni-dvojcevudci-technologie-inteligentniho-prumyslu.html>.
- [13] Networkworld – IoT [online]. Červen 2022 [cit. 2022-06-01]. Dostupné z <https://www.challenge.org/insights/digital-twin-history/>
- [14] Vidyatec – Rozdělení a obrázek [online]. Červen 2022 [cit. 2022-06-01]. Dostupné z <https://vidyatec.com/blog/the-3-levels-of-the-digital-twin-technology-2/>
- [15] Live.orseo – OSM [online]. Červen 2022 [cit. 2022-06-01]. Dostupné z https://live.osgeo.org/ru/quickstart/osm_quickstart.html
- [16] Research gate – smart city[online]. Červen 2022 [cit. 2022-06-01]. Dostupné z https://www.researchgate.net/publication/348382801_A_digital_twin_smart_city_for_citizen_feedback
- [17] Csas – digitalni-dvojcata [online]. Červen 2022 [cit. 2022-06-14]. Dostupné z :<“<https://www.csas.cz/cs/firmy/articles/digitalni-dvojcata>“>
- [18] Digitální dvojče/Automatizace inženýrských prací Bulletin [online]. Copyright © 2019/7 Průmyslu 4.0 Červen 2022 [cit. 2022-06-14]. Dostupné z: <https://www.ncp40.cz/files/bulletin-prumyslu-40-07.pdf>
- [19] globallogic – výhody [online]. Červen 2022 [cit. 2022-06-14]. Dostupné z :<“<https://www.globallogic.com/insights/blogs/if-you-build-products-you-should-be-using-digital-twins/>“>
- [20] forbes – nevýhody [online]. Červen 2022 [cit. 2022-06-14]. Dostupné z <https://www.forbes.com/sites/bernardmarr/2022/06/20/the-best-examples-of-digital-twins-everyone-should-know-about/?sh=7699c954225f>
- [21] slingshotsimulation – bezpečnostní nevýhody [online]. Červen 2022 [cit. 2022-06-14]. Dostupné z <https://www.slingshotsimulations.com/technical/how-secure-is-your-digital-twin/>
- [22] individual.fitness – FMS [online]. Červen 2022 [cit. 2022-06-14]. Dostupné z <https://www.individual.fitness/fms/>
- [23] Deloitte – Průmysl 4.0 [online]. Červen 2022 [cit. 2022-06-14]. Dostupné z <https://www2.deloitte.com/us/en/insights/focus/industry-4-0/digital-twin-technology-smart-factory.html>

- [24] Plm. automation – siemens [online]. Červen 2022 [cit. 2022-06-14]. Dostupné z <https://www.plm.automation.siemens.com/global/en/>
- [25] assetstore – Unity [online]. Červen 2022 [cit. 2022-08-22]. Dostupné z <https://assetstore.unity.com/packages/tools/ai/simple-traffic-system-159402>
- [26] Statista – IoT Devices [online]. září 2022 [cit. 2022-09-01]. Dostupné z <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>
- [27] GILAD MAYAAN, David. IoT zařízení za rok 2020 [online]. 13. leden 2020 [cit. 2022-09-02]. Dostupné z: <https://securitytoday.com/articles/2020/01/13/the-iotrundown-for-2020.aspx>.
- [28] 30 nejlepších příkladů z roku 2021. Jigsaw Academy [online]. [cit. 2022-09-01]. Dostupné z: <https://www.jigsawacademy.com/5-best-examples-iot-applications-real-world/>.
- [29] Geospatial – propojení digitálních dvojčat [online]. červen 17, 2021 [cit. 2022-09-01]. Dostupné z: <https://www.geospatialworld.net/blogs/digital-cities-digital-twins-for-connected-city-infrastructure/>
- [30] Význam digitálních dvojčat s IoT | IoTPort. Propojujeme firmy a lidi se zájmem v internetu věcí | IoTPort [online]. Copyright ©2020 České Radiokomunikace a.s. [cit. 30.12.2020]. Dostupné z: <https://www.iotport.cz/digital-twin-uz-jste-nekdy-potkali-digitalni-dvojce>
- [31] Siemens modelování digitálních dvojčat [online] [cit. 2022-09-01]. Dostupné z: <https://www.plm.automation.siemens.com/global/cz/products/mechanical-design/modeling-technology-platform.html>
- [32] Neoapps – siemen nx programování [online] [cit. 2022-09-01]. Dostupné z: <https://www.neoapps.de/en/services/siemens-nx-programming/>
- [33] Autodesk – CAD/CAM [online] [cit. 2022-09-01]. Dostupné z: <https://www.autodesk.com/solutions/cad-cam>
- [34] Unity – Základní informace o Unity [online] [cit. 2022-09-01]. Dostupné z: <https://unity.com/>
- [35] Assetstore – Perspective digital twin software [online] [cit. 2022-09-01]. Dostupné z: <https://assetstore.unity.com/packages/tools/utilities/perspective-digital-twin-software-161664>

- [36] Conceptartempire – Unreal Engine, Thomas Denham [online] [cit. 2022-09-01]. Dostupné z: <https://conceptartempire.com/what-is-unreal-engine/>
- [37] Johari, Aayushi. Edureka – co o je java [online] Nov 26,2020 [cit 2022-07-15] Dostupné z: <“<https://www.edureka.co/blog/what-is-java/>”>
- [38] Heller, Martin. infoworld– what is kotlin the java alternative explained. [online] Březen 23, 2020 [cit 2022-07-15] Dostupné z: <“<https://www.infoworld.com/article/3224868/what-is-kotlin-the-java-alternative-explained.html>”>
- [39] Mkhitarian Armina. medium – why is c among the most popular programming languages in the world. [online] Oct 13, 2017 [cit 2022-07-15] Dostupné z: <“<https://medium.com/sololearn/why-is-c-among-the-most-popular-programming-languages-in-the-world-ccf26824ffcb>”>
- [40] Developer.apple – Programming with ObjectiveC [online] [cit 2022-07-15] Dostupné z: <“<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>”>
- [41] Hudson, Paul. hackingwithswift – why many developers still prefer objective c to swift [online] October 4th 2017 [cit 2022-07-15] Dostupné z: <“<https://www.hackingwithswift.com/articles/27/why-many-developers-still-prefer-objective-c-to-swift>”>
- [42] Docs.swift.org – swift book [online] [cit 2022-07-15] Dostupné z: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/>
- [43] Docs.swift.org – swift book [online] [cit 2022-09-15] Dostupné z: <https://docs.microsoft.com/en-us/azure/digital-twins/overview>
- [44] AEC Magazine – Unreal engine pro tvorbu digitálních dvojčat [online] 1.3.2021 [cit 2022-10-01] Dostupné z: <https://aecmag.com/news/3d-repo-using-unreal-engine-for-digital-twin-platform/>
- [45] Design, Modeling and Implementation of Digital Twins – Mariana Segovia and Joaquin Garcia-Alfaro [online] 20.7.2022 [cit 2022-10-04] Dostupné z: <https://www.mdpi.com/1424-8220/22/14/5396/pdf?version=1658279232>

- [46] Design, Modeling and Implementation of Digital Twins – Mariana Segovia and Joaquin Garcia-Alfaro [online] 20.7.2022 [cit 2022-10-04] Dostupné z: <https://www.mdpi.com/1424-8220/22/14/5396/pdf?version=1658279232>
- [47] Internet of Things vs digital twin – Carlos Miskinis Digital twin research expert [online] Květen.2018 [cit 2022-10-06] Dostupné z: <https://www.challenge.org/insights/internet-of-things-vs-digital-twin/>
- [48] Static4 - connectivityprotocolsforsmartlighting [online] [cit 2022-10-06] Dostupné z: https://static4.arrow.com/-/media/arrow/files/pdf/c/connectivityprotocolsforsmartlighting_final.pdf
- [49] IBM - How does a digital twin work [online] [cit 2022-10-06] Dostupné z: <https://www.ibm.com/topics/what-is-a-digital-twin>
- [50] Esri - Overview [online] [cit 2022-10-22] Dostupné z: <https://www.esri.com/en-us/arcgis/about-arcgis/overview>
- [51] ČZU – Hackathon 2021 - Tereza Lisá (FAPPZ), Eliška Nováková (PEF) a David Novák (PEF) - <https://www.czu.cz/cs/r-7229-aktuality-czu/hackathon-2021-soutez-o-nejlepsi-chytra-reseni-pro-zivot-v-k.html>
- [52] Informační systém [online]. [cit. 11.10.2022]. Dostupné z: https://is.muni.cz/do/rect/el/prezentace/11_09_2013/workshopy/wsB3/priklad_y/priklad_procesni_rizeni.pdf?lang=cs
- [53] BUCHALCEVOVÁ, A. Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky. 1. vyd. Praha: Grada, 2005. 163 s. ISBN 978-80-247-1075-7.
- [54] MACIASZEK, L. A. a B. L. LIONG. Practical Software Engineering. A Case Study Approach. 1. vyd. Harlow: Addison-Wesley, 2005. 864 s. ISBN 978-03-212-0465-4.
- [55] VRANA, I. Projektování informačních systémů s UML. 1. vyd. Praha: Česká zemědělská univerzita, 2008. 147 s. ISBN 978-80-213-1817-5.
- [56] Managementmana – vodopádový model [online]. [cit. 14.10.2022]. Dostupné z <https://managementmania.com/cs/vodopadovy-model-waterfall-model>
- [57] Javatpoint – Software engineering prototype model [online]. [cit. 14.10.2022]. Dostupné z <https://www.javatpoint.com/software-engineering-prototype-model>
- [58] Managementmana – vodopádový model [online]. [cit. 14.10.2022]. Dostupné

- [59] Capterra – What is rapid application development – Toby Cox - Guest Contributor [online]. 6. Prosinec, 2019 [cit. 15.10.2022]. Dostupné z <https://www.capterra.com/resources/what-is-rapid-application-development/>
- [60] Inetwork – Extrémní programování [online]. [cit. 17.10.2022]. Dostupné z <https://www.itnetwork.cz/navrh/metodiky/extremni-programovani>
- [61] Zcu.arcao – iterativní přístup [online]. [cit. 19.10.2022]. Dostupné z https://zcu.arcao.com/statnice_ing_fav/nis/wiki_zvesela/03.htm
- [62] Šubrt, T. a kolektiv. Ekonomicko-matematické metody. 3. vyd. Čeněk. A. Plzeň, 2019. 354 s. ISBN 978-80-7380-762-7
- [63] Volná, E a Kotyrba M. Umělá inteligence. 1. vyd. Ostrava, červenec 2013. 128 s. ISBN 978-80-7464-330-9
- [64] Venturebeat – Why AI and digital twins could be the keys to a sustainable future - Cheryl Ajluni – Guest Contributor [online]. 24. října, 2022 [cit. 19.10.2022]. Dostupné z <https://venturebeat.com/ai/why-ai-digital-twins-could-be-keys-sustainable-future/>
- [65] All3dp – best cad software [online]. [cit. 19.10.2022]. Dostupné z <https://all3dp.com/1/best-cad-software/>
- [66] Plm.automation.siemens – product nx [online]. [cit. 21.10.2022]. Dostupné z <https://www.plm.automation.siemens.com/global/en/products/nx/>
- [67] Unity – informace o nástroji [online]. [cit. 21.10.2022]. Dostupné z <https://unity.com/>
- [68] Unrealengine – informace o nástroji [online]. [cit. 23.10.2022]. Dostupné z <https://www.unrealengine.com/>
- [69] Azure.microsoft – digital twins [online]. [cit. 24.10.2022]. Dostupné z <https://azure.microsoft.com/en-gb/pricing/details/digital-twins/>
- [70] Trustradius – pricing [online]. [cit. 24.10.2022]. Dostupné z <https://www.trustradius.com/products/arcgis/pricing>
- [71] Arcisgis – informace o nástroji [online]. [cit. 25.10.2022]. Dostupné z <https://www.arcgis.com/index.html>
- [72] IBM – Digital twin exchange [online]. [cit. 25.10.2022]. Dostupné z <https://www.ibm.com/products/digital-twin-exchange/pricing>
- [73] G2 – digital-twin [online]. [cit. 25.10.2022]. Dostupné z <https://www.g2.com/categories/digital-twin>

- [74] Guzina, L. (no date) Understanding digital twin: A systematic mapping study – diva portal. [Online]. [cit. 31.12.2022]. Dostupné z: <https://mdh.diva-portal.org/smash/get/diva2:1586946/FULLTEXT01.pdf>

7 Seznam příloh

7.1 Příloha 1 – Stanovení vah kritérií – Saatyho metoda párových porovnání.

	Použitelnost	Hardwarové požadavky	Doba učení	Náročnost UI	Cena licence	geom. průměr	váha (preference)
Použitelnost	1	3	7	9	7	4,2103	0,53
Hardwarové požadavky	1/3	1	5	7	5	2,2552	0,28
Doba učení	1/7	1/5	1	5	1	0,6776	0,08
Náročnost UI	1/9	1/7	1/5	1	1/3	0,2540	0,03
Cena licence	1/7	1/5	1	3	1	0,6118	0,08
						8,0089	1,00

IS

0,071305258

λ






5,28522

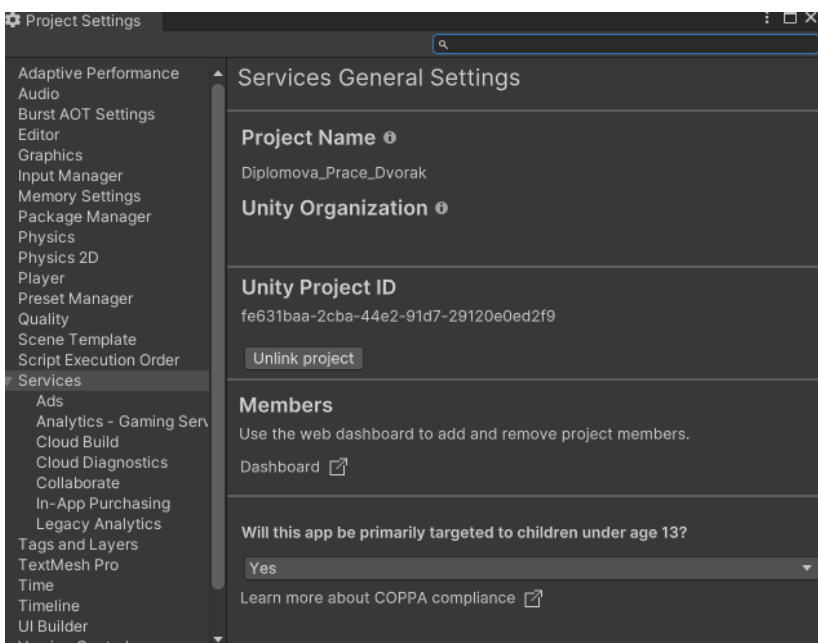
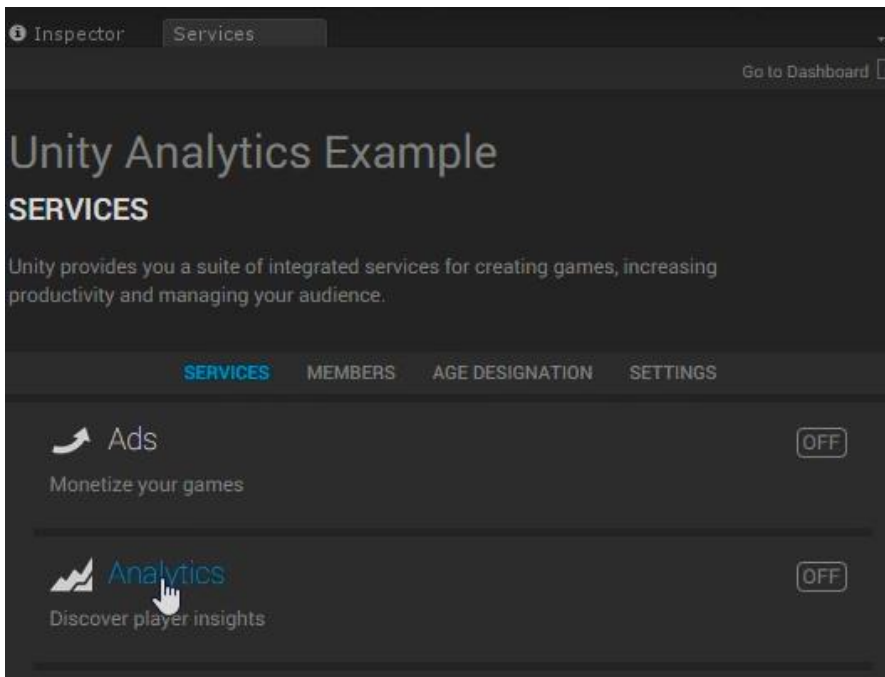
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

determinant -0,0009098

-4,2852	3	7	9	7
0,33333	-4,2852	5	7	5
0,14286	0,2	-4,2852	5	1
0,11111	0,14286	0,2	-4,2852	0,3333333
0,14286	0,2	1	3	-4,285221

7.2 Příloha 2 – Nastavení Unity Analytics

Project details These general settings apply to your project in all Unity products.	Project name	Diplomova_Prace_Dvorak	
	Project icon		
	Project ID	fe631baa-2cba-44e2-91d7-29120e0ed2f9	
In-app purchase (IAP) settings Manage your IAP settings across this project.	Google License Key	Not set	



7.3 Příloha 3 – Unity zapínání a vypínání světel

```
4  
5 public class ONANDOFF : MonoBehaviour {  
6     public GameObject ON;  
7  
8     public void ukaz()  
9     {  
10        if (ON != null)  
11        {  
12            ON.SetActive(true);  
13        }  
14        else  
15        {  
16            ON.SetActive(false);  
17        }  
18    }  
19 }  
20
```