



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií



Vytvoření výukové aplikace řešící blokové diagramy bezporuchovosti (RBD)

Ročníkový projekt

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie
Autor práce: **Jan Špecián**
Vedoucí práce: Ing. Josef Chudoba, Ph.D.



Tento list nahradte
originálem zadání.

Prohlášení

Byl jsem seznámen s tím, že na můj ročníkový projekt se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasaahuje do mých autorských práv užitím mého ročníkového projektu pro vnitřní potřebu TUL.

Užiji-li ročníkový projekt nebo poskytnu-li licenci k jeho využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Ročníkový projekt jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mého ročníkového projektu a konzultantem.

Současně čestně prohlašuji, že texty tištěné verze práce a elektronické verze práce vložené do IS STAG se shodují.

7. 5. 2019

Jan Špecián

Vytvoření výukové aplikace řešící blokové diagramy bezporuchovosti (RBD)

Abstrakt

Práce je zaměřena na tvorbu desktopové aplikace v .NET Frameworku pro tvorbu RBD diagramů a s tím spojených výpočtů a vizualizací. Výsledná aplikace by měla být schopna dle uživatelem zadaného schématu funkčních bloků s definovanými parametry spočítat pravděpodobnost bezporuchového provozu, že výrobek se do času t nedostane do poruchového stavu.

Klíčová slova: Reliability block diagram, Pravděpodobnost poruchy, Intenzita poruch

Abstract

The work is focused on creating a desktop application in .NET Framework for creating RBD diagrams and related calculations and visualizations. Software application as result of the work should be able to follow a user-defined function block diagram with predefined parameters and should be able to calculate the probability of non failure operationability of the product or system.

Keywords: Reliability block diagram, Probability of failure, Reliability engineering

Poděkování

Tímto bych rád poděkoval Ing. Josef Chudobovi, Ph.D. za věnovaný čas v konzultacích a odborné vedení plné trpělivosti a s tím spojené nabyté zkušenosti.

Obsah

Seznam zkratek	8
1 Teoretický úvod	10
1.1 Distribuční funkce spojité náhodné veličiny	10
1.2 Exponenciální rozdělení	11
1.3 Spolehlivost a střední doba mezi poruchami	11
1.4 Analýza blokového diagramu bezporuchovosti (RBD)	12
1.5 Základní zapojení bloků	12
2 Přehled existujících softwarových nástrojů	14
3 Návrh desktopové aplikace .NET	15
3.1 Objektová struktura	15
3.2 Pomocné třídy	15
4 Průběh vývoje	17
4.1 Rozdělení projektu na subprojekty	17
4.2 Zjednodušení diagramu	17
4.3 Testování	17
5 Návod k použití	19
6 Závěr	21

Seznam obrázků

1.1	Příklad průběhu distribuční funkce exponenciálního rozdělení	10
1.2	Příklad grafu hustoty pravděpodobnosti exponenciálního rozdělení . .	11
1.3	Příklad sériového zapojení komponent	13
1.4	Příklad paralelního zapojení komponent	13
3.1	Zjednodušený class diagram nejčastěji používaných objektů	16
4.1	Možnosti tvorby diagramu	18
5.1	Ovládací panel grafického uživatelského rozhraní	20

Seznam zkratek

RBD	Reliability Block Diagram
WPF	Windows Presentation Foundation
GUI	Graphical User Interface
CLI	Command Line Interface

Úvod

U každého systému je velmi důležitá jeho funkční spolehlivost během doby jeho životnosti. Každý systém, pokud má existovat a fungovat co nejdéle a přitom bez závad, nebo alespoň s jejich co nejmenším počtem, musí splňovat jednu zásadní vlastnost, a tou je spolehlivost. Požadavek na dostatečně velkou a často až maximální spolehlivost námi užívaných systémů má tudíž zcela zásadní význam z hlediska bezpečnostního, ekonomického i ekologického.

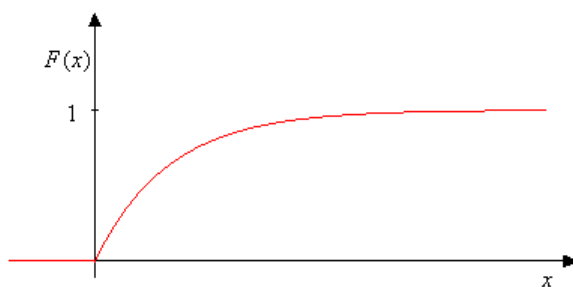
Cílem ročníkového projektu je navrhnout a implementovat desktopovou aplikaci pro tvorbu a jednoduchou vizualizaci RBD diagramů a výpočet parametrů spolehlivosti. Aplikace by měla být schopna spočítat dobu do poruchy, či pravděpodobnost bezporuchovosti za zadané období pro každou komponentu zvlášť a i pro celkový, uživatelem nadefinovaný diagram. Dále navrhnout další možnosti rozvoje aplikace.

1 Teoretický úvod

1.1 Distribuční funkce spojité náhodné veličiny

Jedním z prostředků pro popis náhodné veličiny je distribuční funkce, která každému reálnému číslu přiřazuje pravděpodobnost, že náhodná veličina nabude hodnoty menší nebo rovné než toto číslo.[5] U spojité náhodné veličiny se užívá k jejímu popisu distribuční funkce $F(x)$ definované vztahem:

$$F(x_i) = P(X < x_i)$$



Obrázek 1.1: Příklad průběhu distribuční funkce exponenciálního rozdělení

Vlastnosti distribuční funkce

- Hodnoty distribuční funkce leží v intervalu od nuly do jedné.

$$0 \leq F(x) \leq 1$$

- Distribuční funkce je neklesající.

$$P(x_1 \leq X < x_2) = F(x_2) - F(x_1) \text{ pro } x_1 < x_2$$

- V záporném nekonečnu se blíží k nule, v kladném nekonečnu se blíží k jedné.

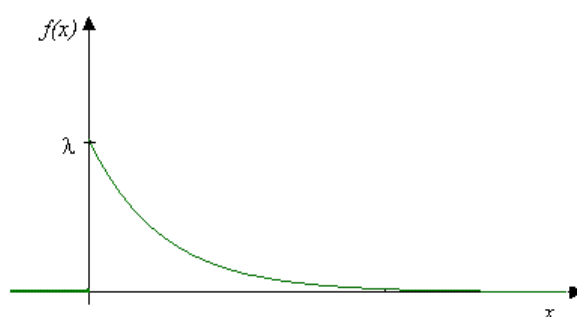
$$F(-\infty) = 0, F(\infty) = 1$$

1.2 Exponenciální rozdělení

Toto rozdělení má spojitá náhodná veličina X , která představuje dobu čekání do nastoupení (poissonovského) náhodného jevu, nebo délku intervalu (časového nebo délkového) mezi takovými dvěma jevy (např. doba čekání na obsluhu, vzdálenost mezi dvěma poškozenými místy na silnici, doba do poruchy). Závisí na parametru λ , což je převrácená hodnota střední hodnoty doby čekání do nastoupení sledovaného jevu. [5]

Náhodná veličina X má exponenciální rozdělení $\text{Exp}(\lambda)$ právě tehdy, když je hustota pravděpodobnosti dána vztahem:

$$f(x) = \begin{cases} 0 & \text{pro } x < 0 \\ \lambda \cdot e^{-\lambda \cdot x} & \text{pro } x \geq 0 \end{cases}$$



Obrázek 1.2: Příklad grafu hustoty pravděpodobnosti exponenciálního rozdělení

Další vlastnosti

- $F(x) = \begin{cases} 0 & \text{pro } x < 0 \\ 1 - e^{-\lambda \cdot x} & \text{pro } x \geq 0 \end{cases}$
- $E(x) = \frac{1}{\lambda}$
- $D(x) = \frac{1}{\lambda^2}$

1.3 Spolehlivost a střední doba mezi poruchami

Střední doba mezi poruchami

Základní veličinou pro měření spolehlivosti systému je střední doba mezi poruchami (MTBF, Mean Time Between Failure). Obvykle je udávána v hodinách. Čím vyšší je hodnota MTBF, tím vyšší je spolehlivost produktu. [3] Je statistická veličina používaná ke kvantifikaci spolehlivosti součásti, či celého výrobku. Určuje se pro výrobek nebo zařízení, které se opravuje. [3]

Spolehlivost

Spolehlivost je schopnost systému nebo součásti vykonávat požadované funkce za daných podmínek po určené časové období [1]

$$Spolehlivost = e^{-\left(\frac{Doba}{MTBF}\right)}$$

Pravděpodobnost bezporuchovosti

Značíme $R(t)$. Udává pravděpodobnost, že výrobek do zadaného času t nebude mít poruchu. Jedná se o doplněk k hodnotě distribuční funkce $F(t)$, což udává pravděpodobnost, že výrobek bude mít do času t poruchu.[4]

Pravděpodobnost bezporuchového provozu $R(t)$ je dána vztahem:

$$R(t) = 1 - F(t)$$

1.4 Analýza blokového diagramu bezporuchovosti (RBD)

Analýza blokového diagramu bezporuchovosti (RBD - Reliability Block Diagram) je metoda analýzy systému. Diagram RBD je grafická reprezentace logické struktury systému v podobě podsystémů, nebo jednotlivých součástí. To umožňuje, aby byly cesty úspěchu (funkceschopného stavu) reprezentovány tak, jak jsou bloky (podsystemy/součásti) logicky propojeny.[1]

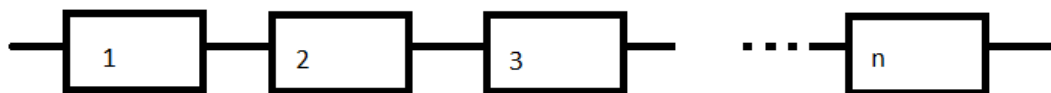
Blokové diagramy jsou mezi prvními úkoly dokončenými během etapy vymezení produktu. Mají být vypracovány jako součást vývoje počáteční koncepce. Práce na nich mají být zahájeny, jakmile existuje vymezení programu, a mají být dokončeny jako součást analýzy požadavků a mají se neustále rozšiřovat do větších úrovní podrobnosti, jakmile budou k dispozici data, aby bylo možné činit rozhodnutí a provádět optimalizace nákladů a přínosů.[2]

1.5 Základní zapojení bloků

Sériové zapojení

Při poruše jedné komponenty dojde k poruše celého systému. Systém je v bezporuchovém stavu, pokud všechny jeho komponenty nemají poruchu.[4] Pravděpodobnost, že všechny komponenty (n) v sériovém systému budou mít poruchu je dána vztahem:

$$R = R_1 R_2 \dots R_n = e^{-\lambda_1 t} e^{-\lambda_2 t} \dots e^{-\lambda_n t} = e^{-(\lambda_1 + \lambda_2 + \dots + \lambda_n) t}$$

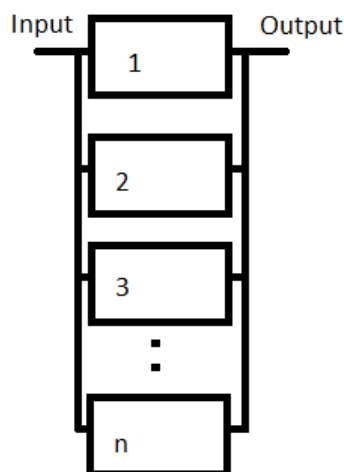


Obrázek 1.3: Příklad sériového zapojení komponent

Paralelní zapojení

K poruše celého systému dochází pokud jsou v poruše všechny jeho komponenty. Bezporuchový stav trvá, dokud je alespoň jedna komponenta v bezporuchovém stavu. Z hlediska odhadu pravděpodobnosti představuje paralelní systém nejlepší variantu pro odhad pravděpodobnosti bezporuchového stavu. [4] Pravděpodobnost, že všechny komponenty (n) v paralelním systému budou mít poruchu je dána vztahem:

$$F = F_1 F_2 \dots F_n = (1 - e^{-\lambda_1 t})(1 - e^{-\lambda_2 t}) \dots (1 - e^{-\lambda_n t})$$



Obrázek 1.4: Příklad paralelního zapojení komponent

2 Přehled existujících softwarových nástrojů

Zde uvedu seznam existujících softwarových řešení pro výpočty v oblasti pravděpodobnosti a teorie spolehlivosti, jejichž prospekty a ukázky jsem prošel.

- ReliaSoft - analýza spolehlivosti
- Weibull.com - Reliability Analysis Software
- ItemSoft - Reliability Engineering Software
- AldService - Reliability Engineering
- Bqr - Leader in Reliability & Maintenance Engineering and EDA
- Goldsim - Reliability Engineering and Risk Analysis for Complex Engineered Systems
- R Project - free software environment for statistical computing and graphics

3 Návrh desktopové aplikace .NET

3.1 Objektová struktura

Item

Instance třídy Item jsou základní datové struktury vyžadující název bloku a rozdělení pravděpodobnosti.

Block

Jedná se o hlavní stavební prvek RBD diagramu. Může reprezentovat pouze jednu komponentu, či kolekci paralelních komponent typu Item.

IDistribution

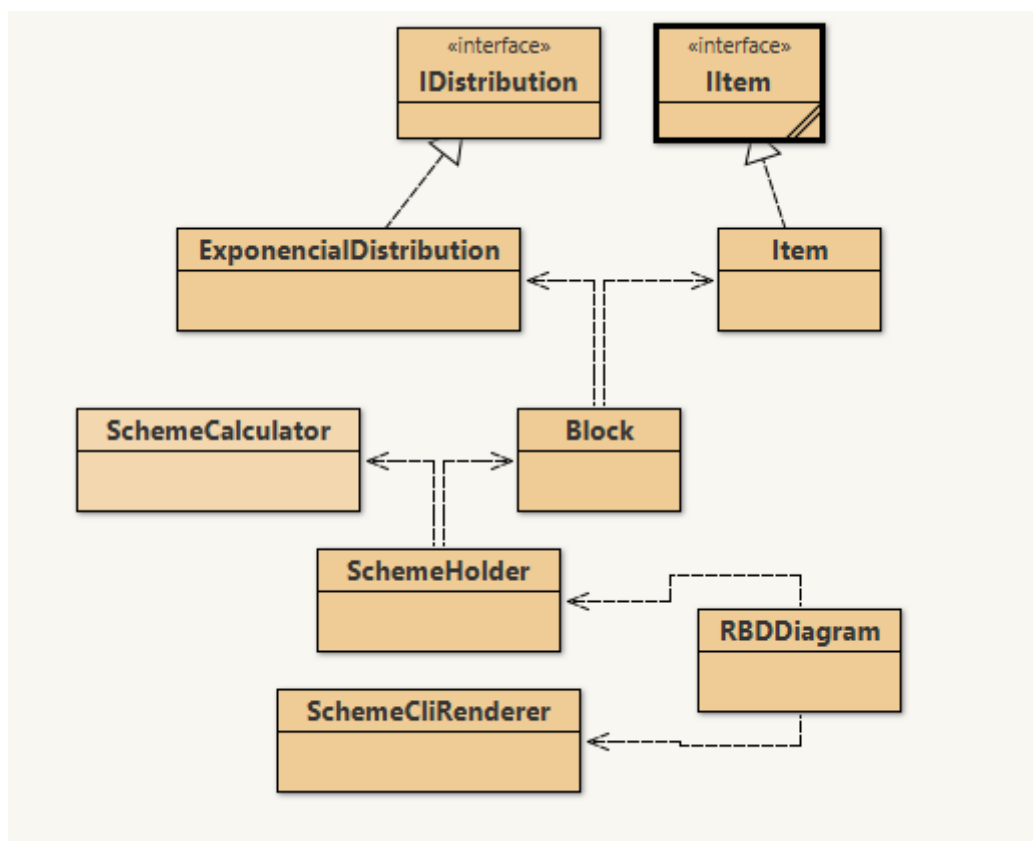
Rozhraní pro rozdělení pravděpodobnosti, které je jedním z atributů objektů typu Item, či může být spočítáno v objektu typu Block.

ExponencialDistribution

Instance této třídy reprezentuje konkrétní rozdělení pravděpodobnosti. Každá komponenta obsahuje vlastní objekt rozdělení.

3.2 Pomocné třídy

SchemeHolder je objekt držící jednotlivé bloky v seznamu, tak jak byly vloženy. SchemeCalculator je servisní třída, která má za úkol na základě předaných parametrů na uloženém schématu spočítat dílčí pravděpodobnosti bezporuchovosti jednotlivých komponent, tak i celkového diagramu.



Obrázek 3.1: Zjednodušený class diagram nejčastěji používaných objektů

4 Průběh vývoje

4.1 Rozdělení projektu na subprojekty

Celkové řešení je rozděleno na několik podprojektů, tak aby každý odpovídal svému účelu použití.

- SpecianPRJ - projekt obsahující veškeré objektové struktury.
- SpecianPRJ.Cli - projekt zaměřený na manuální testování navržených struktur.
- SpecianPRJ.Gui - grafické uživatelské rozhraní pro výslednou aplikaci.
- SpecianPRJ.Tests - unit testy pro výpočty pravděpodobnosti na exponenciálním rozdělení.

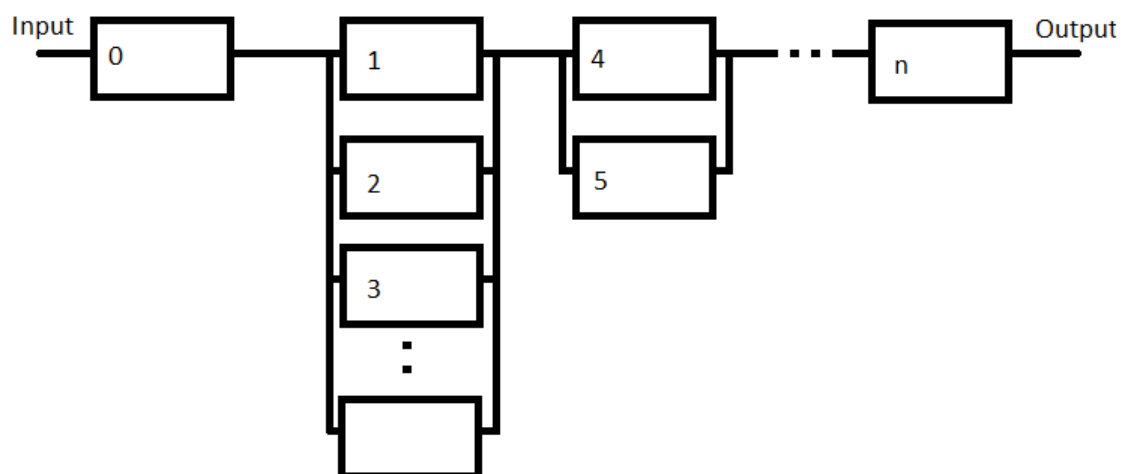
4.2 Zjednodušení diagramu

Pro zjednodušení tvorby diagramu bylo zavedeno pravidlo, že se systém skládá pouze ze série bloků, z nichž některé mohou reprezentovat paralelní zapojení. V aplikaci nelze vytvořit diagram obsahující vazby mezi bloky, které spolu bezprostředně ne-sousedí.

4.3 Testování

Pro testování funkčních bloků byla použita výchozí knihovna pro Unit testování v prostředí .NET pro desktopové aplikace MSTest. Za pomoci testování jsem došel ke správným výsledkům za pomoci připravené konfigurace a tím jsem ušetřil práci manuálním testováním. Další nespornou výhodou testování je odhalení chyb při změně tím, že testovací metody odhalí neočekávané výsledky.

Testované byly třídy pro výpočet distribuční funkce. Testování probíhá za pomoci zaokrouhlení na 6 desetinných míst.



Obrázek 4.1: Možnosti tvorby diagramu

5 Návod k použití

Pro spuštění aplikace pro vývoj je potřeba mít nainstalované Visual Studio 2017 a novější. V příloženém CD ve složce SpecianPRJ spustte soubor SpecianPRJ.sln.

Pro standartní spuštění aplikace stačí otevřít soubor SpecianPRJ.Gui.exe ve složce SpecianPRJ\SpecianPRJ.Gui\bin\Debug

Pro obě varianty spuštění je nutným předpokladem nainstalovaný plný .NET Framework 4.6.1 a novější.

Založení nového diagramu

V současné podobě aplikace je nový diagram založen při každém spuštění aplikace. Jedná se o prázdný seznam bloků v instanci SchemeHolder.

Přidání prvku

Pomocí ovládacích prvků lze přidat bloky a do nich jednotlivé paralelní komponenty. Blok může reprezentovat jedna komponenta typu Item

Výpočty

Uživatel po nadefinování schématu se může dotázat na pravděpodobnost bezporuchovosti systému v zadaném čase.

Block Adding

Name

Add Block

Parallel Item Adding

Name

Lambda

Add parallel Item to

Compute

Time

Compute

Rbd diagram

Obrázek 5.1: Ovládací panel grafického uživatelského rozhraní

6 Závěr

Dosažené výsledky

Ve vývojovém studiu Microsoft Visual Studio jsem vytvořil odpovídající projekt s rozdělenou strukturou podprojektů. Seznámil jsem se s programovacím jazykem C# a jeho základními objektovými principy. Seznámil jsem se s teorií pravděpodobnosti a teorií spolehlivosti. Navrhl jsem jednoduchou objektovou strukturu, na níž jsem si vyzkoušel spočítat pravděpodobnost bezporuchovosti, distribuční funkci pro exponenciální rozdělení a další potřebné charakteristiky.

Na základě této objektové struktury jsem navrhl přehledné grafické rozhraní za pomoci knihoven WinForms, ve kterém uživatel aplikace snadno nadefinuje strukturu komponent v sérii a může se na definovanou strukturu dotázat na pravděpodobnost bezporuchovosti v zadaném čase. Správnost výpočtů pravděpodobností a hodno distribuční funkce byla otestována za pomoci unit testů.

Tvorba a zapojení komponent v RBD diagramu bylo zjednodušeno na sériový systém, kde jednotlivé komponenty mohou reprezentovat paralelní systém.

Při vývoji aplikace jsem používal verzovací nástroj git, který mi umožnil si v různých větvích držet různé funkční a nefunkční cesty vývoje a spojovat do sebe různé větve pro dosažení funkčního celku. [6]

Možnosti rozvoje aplikace

Aplikace nenabízí možnost tvorby složitějších schémat a propojení komponent tak, aby diagram odpovídal reálnému zadání. Pro takové vylepšení by bylo nutné předit objektový model včetně výpočetního postupu. Dále by bylo vhodné k aplikaci přidat možnost uložení a znovuotevírání uložených diagramů, nyní je aplikace pouze demonstrační a diagram je nutné manuálně vytvořit po spuštění.

V práci jsem si neporadil s vykreslením schématu, ani průběhů distribuční funkce. Veškeré výstupy jsou v textové podobě.

Pro grafické uživatelské rozhraní by bylo vhodné přejít na knihovny WPF, které jsou nástupcem zastaralých Windows Forms.

Literatura

- [1] Analýza stromu událostí (ETA). Wwww.qmprofi.cz [online]. 2007 [cit. 2019-05-01]. Dostupné z: <https://www.qmprofi.cz/33/analyza-stromu-udalosti-eta-uniqueidmRRWSbk196FNf8-jVUh4EuiqB7MU5EZkI4KJhlhjbS2LV4wcZRqCHQ/>
- [2] NOVÁK, Bc. Rudolf. Analýza spolehlivosti v oblasti IT. 2009. Diplomová práce. JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH. Vedoucí práce Ing. Ludvík Friebeľ, Ph.D.
- [3] TORELL, Wendy. Střední doba mezi poruchami: vysvětlení a standardy. [Http://gabben.wbs.cz](http://gabben.wbs.cz) [online]. [cit. 2019-05-01]. Dostupné z: <http://gabben.wbs.cz/mtbf1.pdf>
- [4] PAVEL, Fuchs, CSc. a Vališ, Ph.D. DAVID. Bezporuchovost a životnost: Techniky analýzy bezporuchovosti.
- [5] OTIPKA, Petr a Vladislav ŠMAJSTRLA. PRAVDĚPODOBNOST A STATISTIKA [online]. VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA [cit. 2019-05-01]. Dostupné z: <https://homen.vsb.cz/oti73/cdpast1/>
- [6] We bring the awesome Git SCM to Windows. Git for windows [online]. [cit.2019-04-24]. Dostupné z: <https://gitforwindows.org/>