
Projekt CSUPP

Naročnik: Šaj d.o.o.
Vodja projekta: Anton Zhezhov

Začetek: 09.10.2019

Konec: 31.01.2020

Ime in Priimek	Vloga	e - Naslov	Opomba
Anton Zhezhov	Preverjanje	anton.zhezhov@student.um.si	
Žiga Zorc	Razvoj	ziga.zorc@student.um.si	

0 Naročnikove zahteve

0.1 Splošne informacije

Dokument	Verzija 1.0
Naročnik	Šaj d.o.o.
Lokacija dokumenta	Github url
Odgovorna oseba	Direktor podjetja Šaj d.o.o.

0.2 Zahteve

V podjetju Šaj d.o.o. se ukvarjamo z razvojem inovativnih rešitev na področju avtomatizacije in digitalizacije upravljanja poslovnih prostorov. Pri načrtovanju naših rešitev dajemo velik poudarek na okoljsko trajnost, energetsko učinkovitost ter ergonomičnost produktov, saj se zavedamo, da omenjene lastnosti pozitivno vplivajo tako na izboljšano uporabniško izkušnjo kot na optimizacijo poslovanja skozi nižanje stroškov.

V podjetju smo prepoznali pomanjkanje rešitev, ki bi celovito naslovile problem zastarelosti poslovnih prostorov. V ta namen načrtujemo razvoj centralnega sistema za upravljanje poslovnega prostora, s čimer se nadejamo preboja na trg in s tem izboljšanja poslovnega uspeha. Projekt že ima izoblikovano idejno zasnovo, in sicer tako glede strojne opreme kot izgleda in funkcionalnosti. Sedaj smo v fazi iskanja resnega partnerja, ki bi prevzel razvoj programske opreme. Ker želimo preveriti osnovni koncept in delovanje centralnega sistema za upravljanje prostora, naj bo program napisan v obliki simulatorja. Najprej potrebujemo preprost simulator brez grafičnega vmesnika, ki bo izdelan kot konzolna aplikacija v jeziku C++ v integriranem razvojnem okolju Visual Studio. Od simulatorja pričakujemo brezhibno in robustno delovanje v operacijskem sistemu Windows. Poleg tega mora biti simulator hiter in preprost za uporabo. Simulator naj omogoča krmiljenje temperature, vlage in osvetljenosti prostora. Predpogoj je, da uporabnik v tekstovno datoteko vpiše želene ambientalne lastnosti v obliki:

TEMPERATURA: vrednost

VLAZNOST: vrednost v obliki relativne vlažnosti [%]

OSVETLJENOST: vrednost v luksih [lx]

V datoteki naj bo še:

INTERVAL TEMPERATURE: [10,40]

STOPNJA VLAZNOSTI: [30,60]

INTERVAL OSVETLJENOSTI: [10,10000]

Simulator naj pred pričetkom prebere vrednosti iz datoteke, nato pa naj omogoča izbiro med tremi načini delovanja:

1. Testni način: Uporabnik v program vnese dejansko temperaturo v prostoru. Računalnik vneseno temperaturo pretvori v ostale relevantne merske enote. Nato naj izračuna razliko do želene temperature (v vseh izbranih merskih enotah) in izvede ukaz za regulacijo temperature. Analogno naj simulator omogoča vpis, izračun in izvedbo ukazov še za vlažnost in osvetljenost. Simulacija se izvaja, dokler je ne prekine uporabnik.

2. Avtomatski način: Računalnik naj si izmisli dejansko temperaturo na intervalu podanem v datoteki, pri čemer jo pretvori v najpomembnejše preostale merske enote. Izmisli naj si še relativno stopnjo vlažnosti, in sicer med 30 in 60 %, ter osvetljenost na intervalu z datoteke. Nato naj za vsako posamezno meritev izračuna odstopanje od zelenih vrednosti ter izvede ukaze za popravek. Simulator naj izvede 100 meritev, pri čemer izvede posamezno meritev vsake 3 sekunde. Na koncu simulacije naj izračuna povprečno vrednost meritev ter povprečno odstopanje od zelenih vrednosti za posamezne parametre.

3. Avtomatski način 2: Simulator naredi isto kot v točki 2, pri čemer naj uporabniku omogoča izbiro pri številu meritev in časovnem razmiku med njimi. Izvajalec mora natančno slediti vsem internim standardom in poskrbeti za dokumentacijo.

Sestavni del projekta sta tudi razvijalska dokumentacija in uporabniški priročnik. Od izvajalca pričakujemo, da do 24. 10. 2019 do 23.55 odda plan projekta, ki vključuje ceno. Program in dokumentacija morata biti oddana najkasneje 23. 1. 2020 do 23.55. Projekt bo plačan po posameznih zaključenih fazah. Za vsak teden zamude bo odbitih 10 % plačila.

Maribor 01.10.2019

Direktor podjetja Šaj d.o.o.

1 Plan projekta

1.1 Tabela

1.2 Kratek opis problema

Podjetje Šaj d.o.o. (v nadaljevanju naročnik) je dne 1. 10. 2019 naročilo razvoj centralnega sistema za upravljanje poslovnega prostora.

Naročnik želi optimizirati svoje poslovne prostore z avtomatiziranim sistemom, ki meri in upravlja s parametri. Sistem je preprost "simulator", ki je sposoben prilagajanja parametrov tako avtomatsko kot na specifične uporabnikove zahteve.

1.2.1 Globalni cilji(globalne zahteve), ki jih želimo s produktom doseči

- Izdelati simulator, ki primerno regulira parametre v prostoru
- Simulator mora biti hiter in preprost za uporabo

1.2.2 Omejitve

- Programski jezik: C++
- Operacijski sistem: Windows
- Izdelan mora biti kot simulator
- Konzolna aplikacija oz. brez grafičnega vmesnika

1.2.3 Rok za zaključitev projekta, skupni stroški

- Do 22.10.2019 do 23:55 oddan plan projekta
- Do 23.01.2020 do 23:55 oddan projekt

1.2.4 Funkcije

- Pretvarjanje temperature v merske enote (Fahrenheit [$^{\circ}F$], Kelvin [K], Rankine [$^{\circ}R$], Delisle [$^{\circ}De$], Newton [$^{\circ}N$], Réaumur [$^{\circ}R\acute{e}$], Rømer [$^{\circ}R\emptyset$])
- Razlika do želene temperature
- Regulacija temperature
- Računalnik simulira (ustvari svoje vrednosti) in na to izračuna odstop od ustvarjene vrednosti

1.2.5 Pomembne karakteristike

- Preprost za uporabo oz. intuitiven in hiter
- Delovanje v OS Windows

1.2.6 Neizvedljive zahteve

- Brezhibnost
- Robustnost

1.2.7 Označevanje verzij

- Verzija: vx.y_DDMMLLLL
- x - velike spremembe, y - manjše spremembe
- Primer: v3.1_17112019

1.3 Zagotavljanje kakovnosti (Načrt preverjanja)

1.3.1 Objekti preverjanja

- D1 Naročnikove zahteve
- D2 Plan projekta
- D3 Sistemske specifikacije
- D4 Testni primeri
- D5 Poročilo o preverjanju
- D6 Načrtovalsko dokumentacijo
- D7 Uporabniški priročnik

Glede na izbran model razvoja obstajajo delni in končni produkti, ki jih je potrebno na koncu vsake faze preveriti (glej tabelo Pregled po produktih in aktivnostih). Kompletan terminski plan je podan v nadaljevanju tega dokumenta. Končni produkt predstavljajo dokumenti D1-D7.

a) Preverjanje programa v1.0

Program v1.0 bomo preverili s pregledom izvirne kode (stil kodiranja, skladnost s standardom) in testiranjem. Pripravljeni bodo določeni testni vzorci in postopki, ki jih bo natančneje definiral dokument Testni primeri. Preverjanje izvaja preverjevalec. Po preverjanju se izpolnijo pisna poročila o najdenih neustreznostih. Na podlagi teh poročil se izvede odpravljanje neustreznosti. Najprej se bodo preverili tipični testni vzorci, če pri njih ne najdemo resne hibe, se izvedejo tudi ostali testi. Ne izvaja se nobenih regresijskih testov.

b) Preverjanje programa v2.0

Program v2.0 bomo preverili s pregledom izvirne kode (stil kodiranja, skladnost s standardom) in testiranjem. Pripravljeni bodo določeni testni vzorci in postopki, ki jih bo natančneje definiral dokument Testni primeri. Preverjanje izvaja preverjevalec. Po preverjanju se izpolnijo pisna poročila o najdenih neustreznostih. Izvedejo se vsi testi (regresijsko testiranje).

Uporabljene bodo naslednje strategije (podroben opis v prilogi)

- prisotnost zahtev (Z)
- prepovedane vrednosti - za preverjanje robustnosti (R)
- mejne vrednosti (M)
- ugibanje napak oziroma nepravilnosti (U)

1.4 Naloge in rezultirajoči dokumenti (izbran razvojni model)

1.4.1 Pogled po produktih in aktivnostih

	Produkt	Planirana kompleksnost	Dejanska kompleksnost	Odgovorna oseba za produkt	V&V metoda	Odgovorna oseba za V&V	Način sporočanja o V&V	Opomba
D1	Naročnikove zahteve	1.5 strani		naročnik	splošni pregled		ustno	
D2	Plan projekta	6 strani		Anton Zhezhov	splošni pregled	Anton Zhezhov	ustno	
D3	Sistemske specifikacije	10 strani		Anton Zhezhov	splošni pregled	Anton Zhezhov	ustno	
	Program v1.0	700 LOC		Žiga Zorc	sploš. pregled + test	Anton Zhezhov	interni zapisnik	
D4	Testni primeri	50 testnih primerov		Anton Zhezhov	sploš. pregled + test	Anton Zhezhov	ustno	
D5	Testno poročilo	6 strani		Anton Zhezhov	splošni pregled	Anton Zhezhov	ustno	
D6	Načrtovalska dokumentacija	5 strani		Anton Zhezhov	splošni pregled	Anton Zhezhov	ustno	
D7	Uporabniški priročnik	8 strani		Žiga Zorc	splošni pregled	Žiga Zorc	ustno	
	Program v2.0	1300 LOC		Žiga Zorc	sploš. pregled + test	Anton Zhezhov	interni zapisnik	
	Kompleten produkt	1500 LOC		vsi	sploš. pregled + test	Naročnik		

1.4.2 Rok in stroški

	Aktivnost	Planiran rok	Dejanski rok	Planirani napor	Planirani stroški	Dejanski napor	Dejanski stroški	Izvajalec	Odgovorna oseba
A1	Planiranje projekta in analiza zahtev	22.10.2019		4	400				
A2	Načrtovanje	03.11.2019		4	400				
A3	Implementacija programa v1.0	03.01.2020		4	400				
A4	Implementacija programa v2.0	16.01.2020		6	600				
A5	Načrtovanje testnih primerov	03.01.2020		2	200				
A6	Preverjanje programa v1.0	09.01.2020		3	300				
A7	Preverjanje programa v2.0	TBD		2	200				
A8	Izdelava kompletne dokumentacije	16.01.2020		7	700				
A9	Prevzem	23.01.2020		1	100				
A1	Skupaj napor - stroški				3300				

Enota napora: človek-dan
 Stroški enote napora: 100 EUR

1.5 Resursi

1.5.1 Osebj

	Oseba	Aktivnost	Vloga
P1	Direktor podjetja	<ul style="list-style-type: none">• nadzor• prevzem	naročnik
P2	Anton Zhezhov	<ul style="list-style-type: none">• načrtovanje testnih primerov• testiranje• planiranje projekta• izdelava načrtovalske dokumentacije• prevzem	preverjevalec
P3	Žiga Zorc	<ul style="list-style-type: none">• analiza zahtev• načrtovanje• implementacija programa v1.0• implementacija programa v2.0	razvojniki

1.5.2 Potrebna programska orodja, knjižnice

Orodje	Namen, funkcija
Microsoft Visual C++	Kodiranje, odpravljanje neustreznosti
L ^A T _E X	Vodenje dokumentacije
TBD	Merilnik kompleksnosti

1.5.3 Potrebna strojna oprema

Orodje	Namen, funkcija
PC	Kodiranje, odpravljanje neustreznosti, vodenje dokumentacije, testiranje
Printer	Izpis dokumentacije

1.6 Razdelitev stroškov

Točka 1.4.2

1.7 Terminski plan projekta

	Aktivnost	Časovna skala																																
		1	1 2	2	2 3	3	3 4	4	4 5	5	5 6	6	6 7	7	7 8	8	8 9	9	9 10	10	10 11	11	11 12	12	12 13	13	13 14	14	14 15	15				
A1	Planiranje projekta in analiza zahtev	+	+	+	+	+	+	+																										
A2	Načrtovanje								+	+	+	+	+																					
A3	Implementacija programa v1.0												+	+	+	+	+	+	+	+	+													
A4	Implementacija programa v2.0																						+	+	+	+	+							
A5	Načrtovanje testnih primerov											+	+	+	+																			
A6	Preverjanje programa v1.0												+	+	+	+	+	+	+	+	+													
A7	Preverjanje programa v2.0																						+	+	+	+	+							
A8	Izdelava kompletne dokumentacije																											+	+					
A9	Prevzem																													+				
	Dokument (skrajni rok)	1	1 2	2	2 3	3	3 4	4	4 5	5	5 6	6	6 7	7	7 8	8	8 9	9	9 10	10	10 11	11	11 12	12	12 13	13	13 14	14	14 15	15				
A1	Naročnikove zahteve	+																																
A2	Plan projekta							+																										
A3	Sistemske specifikacije									+																								
A4	Testni vzorci																					+												
A5	Testno poročilo																							+										
A6	Načrtovalska dokumentacija																													+				
A7	Uporabniški priročnik																													+				

- Legenda:

- planirani čas (+)
- dejansko porabljen čas (*)

1.8 Pojemovnik

Pojem	Razlaga
naročnik	Šaj d.o.o.
parametri	Količine, s katerimi upravlja program (temperatura prostora, relativna vlažnost prostora in osvetljenost prostora)

1.9 Priloge

Opisi uporabljenih strategij

1.9.1 Opis strategije: Prisotnost zahtev (Z)

1. Strategija je uporabna je v vseh primerih, kjer so znane specifikacije oziroma zahteve, med katerimi ni nobenih relacij. Predpostavka o napaki: določena zahteva ni implementirana. S to strategijo odkrivamo zahteve, ki niso implementirane. Razen zelo redkih izjem, ne bomo odkrili napačno implementiranih zahtev in zahtev, ki so po nepotrebnem implementirane.
2. Testirni model je seznam zahtev.
3. **Pravilo za načrtovanje testnih primerov:** Za vsako zahtevo tvori najmanj en testni primer. Vhodne podatke si poljubno izberi.
4. Z načrtovanjem testnih primerov lahko začnemo, ko so zahteve postavljene.
5. Testirna strategija je izčrpana, ko preverimo prisotnost vsake zahteve v seznamu.

1.9.2 Opis strategije za preverjanje robustnosti (R)

1. Strategija je uporabna je v vseh primerih, kjer je zahtevana robustnost in je možno tvoriti opis vhodne domene.
2. Predpostavka o nepravilnosti: program ni robusten, čeprav bi moral biti. S to strategijo ne bomo odkrili nepravilnosti, ki se pojavljajo pri procesiranju veljavnih podatkov.
3. Testirni model je opis vhodne domene.
4. **Pravilo za načrtovanje testnih primerov:** V vhodni domeni in identificiraj prepovedane razrede. Za vsak prepovedan razred tvori en testni primer.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je opisana vhodna domena.
6. Testirna strategija je izčrpana, ko smo pokrili vse neveljavne razrede v vhodni domeni. Zgornje število testnih primerov je enako številu neveljavnih razredov.

1.9.3 Opis strategije: ugibanje nepravilnosti (U)

1. Strategija je splošno uporabna.
2. Predpostavlja se, da je prisotna določena nepravilnost ali napaka.
3. Testirni model je seznam potencialnih nepravilnosti oziroma napak.

-
4. **Pravilo za načrtovanje testnih primerov:** Za vsako potencialno napako oziroma nepravilnost v seznamu tvorimo en testni primer, s katerim preverimo, ali je ta napaka/nepravilnost prisotna.
 5. Z načrtovanjem testnih primerov lahko začnemo, ko je imamo pripravljen seznam.
 6. Testirna strategija je izčrpana, ko smo pokrili celoten seznam. Zgornje število testnih primerov je enako številu napak oziroma nepravilnosti v seznamu.

1.9.4 Opis strategije: Mejne vrednosti (M)

1. Strategija je splošno uporabna.
2. Predpostavka o nepravilnosti: vhodni podatki, ki se nahajajo v okolici ali pa točno na meji med veljavnim in neveljavnim območjem, se bodo nepravilno procesirali.
3. Testirni model je vhodna in izhodna domena.
4. **Pravilo za načrtovanje testnih primerov:** določi meje med veljavnimi in neveljavnimi podatki. Izberi vrednost točno na meji, malo nad in malo pod njo.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je imamo podatkovni slovar.
6. Testirna strategija je izčrpana, ko smo uporabili vse meje.