



RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte

Sprintu plānošana

Studiju darbs mācību priekšmetā
“Ievads operāciju pētīšanā”

Izstrādāja:

Nikita Psenicnihs

Studenta apliecības nr.

231RDB237

Pārbaudīja:

Lektors Ralfs Matisons

• SATURS

• SATURS.....	2
• DARBA UZDEVUMS.....	2
• DARBA VARIANTA APRAKSTS.....	3
• VARIANTAM ATBILSTOŠAIS MATEMĀTISKAIS MODELIS	4
• MODEĻA RISINĀJUMS	4
• REZULTĀTU APSTRĀDE UN ANALĪZE	5
• Bāzes modelis	5
• Pilnais modelis	6
• Salīdzinoša analīze	6
• SECINĀJUMI.....	6
• ATSAUCES	7

• DARBA UZDEVUMS

Studiju darba mērķis ir realizēt optimizācijas modeli, kas ļauj efektīvi plānot uzdevumu , ievērojot spējās (Agile) programmatūras izstrādes principus. Darba ietvaros jāizmanto lineārās programmēšanas metodes, lai saņemtu maksimālu lietderības koeficientu, vienlaikus ievērojot laika ierobežojumus, riska novērtējumus, lietotāja stāstu radniecības un atkarību ierobežojumus.

Darba galvenie uzdevumi:

- Analizēt individuālā varianta lietotāja stāstu datus un to parametrus;
- Izveidot matemātisko optimizācijas modeli
- Realizēt modeli, izmantojot Python ar PuLP
- Analizēt iegūto risinājumu.

● DARBA VARIANTA APRAKSTS

Darba variants sastāv no 19 lietotāja stāstiem (User Stories), kuriem dots stāsta lietderības raksturojums (U_j), stāsta punktu skaits (P_j), riska (r_j) un korelācijas (a_j) novērtējums.

Dati tika aizpildīti saskaņā ar Lietotāja stāsta informāciju.

Sprinta Nr	1	2	3	4	5			
Kapacitāte P_i^{\max}	20	20	35	30	20			

User Story	Rentabilitāte U_j (10-100)	Story point P_j (1-10)	Ietekme r_j (1;1.3;1.7;2)	Korelācijas ietekme a_j (0-0.5)	Korelējošās US Y_j	Pirmsnorises D_j
R.01	80	9	1.3	0.0		
R.02	85	8	1.7	0.1	R.03	
R.05	85	9	1.3	0.3	R.02	
R.06	60	5	1.0	0.3		R.01, R.02
R.07	50	6	1.0	0.0		R.09, R.10
R.08	40	4	1.0	0.0		R.01
R.09	50	6	1.0	0.0		
R.10	40	3	1.0	0.0		R.09
R.13	70	9	1.3	0.3	R.14	
R.14	75	7	1.3	0.3	R.13	
R.15	60	7	1.0	0.0		
R.16	55	7	1.0	0.0		
R.17	20	4	1.7	0.2	R.18, R.19	R.13, R.14
R.18	25	4	1.0	0.0		R.17
R.19	25	5	1.0	0.0		R.13, R.14
R.20	15	6	1.0	0.0		
R.53	80	7	1.3	0.0		
R.54	55	7	1.0	0.0		
R.55	50	4	1.0	0.0		

• VARIANTAM ATBILSTOŠAIS MATEMĀTISKAIS MODELIS

Lai optimāli sadalītu 19 lietotāja stāstus pa pieciem sprintiem, tiek izveidots lineārās programmēšanas (LP) modelis ar binārajām mainīgajām. Modelis pamatojas uz vairāku mugursomu problēmu (sakarā ar uzdevumu) paplašinājumu, ņemot vērā prioritātes, riskus, korelācijas un atkarības starp uzdevumiem.

1) Modeļa mainīgie

$x_{ij} \in \{0, 1\}$ — bināra mainīga (galvena): vai uzdevums i ir iekļauts sprintā j

$y_{ij} \in \{0, 1\}$ — palīgmainīga, kas tiek izmantota, ieverojot korelāciju.

2) Mērķa funkcija

Bāze:

$$\text{Max } Q = \sum \sum U_j * r_j * x_{ij}$$

Ar korelāciju:

$$\text{Max } Q = \sum \sum U_j * r_j * x_{ij} + \sum \sum U_j * a_j * (Y_{ij} / Y_j)$$

3) Ierobežojumi

Sprintu kapacitātes ierobežojums:

$$\sum P_j * x_{ij} \leq P_i \text{ max}$$

Uzdevuma piešķiršana tieši vienam sprintam:

$$\sum x_{ij} \leq 1$$

Korelāciju nosacījumi:

$$y_{ij} \leq \sum x_{ik}, y_{ij} \leq |Y_j| * x_{ij}$$

• MODEĻA RISINĀJUMS

Lai atrisinātu izstrādāto optimizācijas modeli, tika izmantota Python programmēšanas valoda ar bibliotēku PuLP. Modelis tika izstrādāts atbilstoši aprakstītajiem ierobežojumiem, datiem, mērķa funkcijai, mainīgiem, un tt. Un tas tika izmantots, lai optimāli sadalītu 19 lietotāja stāstus (User Stories) piecos sprintos.

Iegutie dati:

Status: Optimal Score: 1283.5 Sprint 1 (max. 20 SP): --- R.01 (Uj=80, SP=9) --- R.09 (Uj=50, SP=6) Used: 15/20	Sprint 2 (max. 20 SP): --- R.10 (Uj=40, SP=3) --- R.13 (Uj=70, SP=9) --- R.14 (Uj=75, SP=7) Used: 19/20	Sprint 3 (max. 35 SP): --- R.02 (Uj=85, SP=8) --- R.05 (Uj=85, SP=9) --- R.07 (Uj=50, SP=6) --- R.15 (Uj=60, SP=7) --- R.55 (Uj=50, SP=4) Used: 34/35	Sprint 4 (max. 30 SP): --- R.06 (Uj=60, SP=5) --- R.08 (Uj=40, SP=4) --- R.16 (Uj=55, SP=7) --- R.17 (Uj=20, SP=4) --- R.18 (Uj=25, SP=4) --- R.19 (Uj=25, SP=5) Used: 29/30	Sprint 5 (max. 20 SP): --- R.20 (Uj=15, SP=6) --- R.53 (Uj=80, SP=7) --- R.54 (Uj=55, SP=7) Used: 20/20
---	---	---	---	---

Galvenas programmas daļas:

```

1)                                     Modeļa                               inicializācija:
model                                +=                                     (
    pulp.lpSum(Uj[j] * rj[j] * X[i, j] for i in Sprints for j in UserStories) +
    pulp.lpSum(Uj[j] * aj[j] * Y[i, j] / len(Yj[j]) for i in Sprints for j in Yj if len(Yj[j]) > 0)
)

```

```

2) Mainīgie:
X = pulp.LpVariable.dicts("X", ((i, j) for i in Sprints for j in UserStories), cat="Binary")
Y = pulp.LpVariable.dicts("Y", ((i, j) for i in Sprints for j in Yj), lowBound=0, cat="Integer")

```

```

3) ierobežojumi:
for i in Sprints:
    model += pulp.lpSum(Pj[j] * X[i, j] for j in UserStories) <= Capacity[i]

```

```

4) Pārbaude:
for j in UserStories:
    model += pulp.lpSum(X[i, j] for i in Sprints) <= 1

```

```

5) Korelācija:
for i in Sprints:
    for j in Yj:
        model += Y[i, j] <= pulp.lpSum(X[i, k] for k in Yj[j])
        model += Y[i, j] <= len(Yj[j]) * X[i, j]

```

```

6) Pirmsnorises:
for j, preds in Dj.items():
    for p in preds:
        for t in Sprints:
            model += pulp.lpSum(X[k, p] for k in range(0, t + 1)) >= X[t, j]

```

• REZULTĀTU APSTRĀDE UN ANALĪZE

• Bāzes modelis

Status: Optimal Score: 1210.5 Sprint 1 (max. 20 SP): --- R.01 (Uj=80, SP=9) --- R.13 (Uj=70, SP=9) Used: 18/20	Sprint 2 (max. 20 SP): --- R.02 (Uj=85, SP=8) --- R.06 (Uj=60, SP=5) --- R.14 (Uj=75, SP=7) Used: 20/20	Sprint 3 (max. 35 SP): --- R.09 (Uj=50, SP=6) --- R.15 (Uj=60, SP=7) --- R.16 (Uj=55, SP=7) --- R.19 (Uj=25, SP=5) --- R.20 (Uj=15, SP=6) --- R.55 (Uj=50, SP=4) Used: 35/35	Sprint 4 (max. 30 SP): --- R.05 (Uj=85, SP=9) --- R.08 (Uj=40, SP=4) --- R.10 (Uj=40, SP=3) --- R.53 (Uj=80, SP=7) --- R.54 (Uj=55, SP=7) Used: 30/30	Sprint 5 (max. 20 SP): --- R.07 (Uj=50, SP=6) --- R.17 (Uj=20, SP=4) --- R.18 (Uj=25, SP=4) Used: 14/20
---	---	---	---	---

Bāzes modelī katrs lietotāja stāsts tika vērtēts neatkarīgi - tika ņemti vērā tikai šādi parametri:

- Rentabilitāte
- Story Points
- Sprinta kapacitāte

Modelis optimizēja uzdevumu sadali, balstoties uz prioritāšu summu, bet neņēma vērā to, cik kritiski vai savstarpēji saistīti bija uzdevumi. Līdz ar to šis modelis radīja efektīvu, bet no projekta loģikas viedokļa ne vienmēr optimālu sadalījumu.

Iegūtais lietderības rezultāts bija nedaudz zemāks nekā pilnajā modelī.

• Pilnais modelis

Status: Optimal Score: 1283.5 Sprint 1 (max. 20 SP): --- R.01 (Uj=80, SP=9) --- R.09 (Uj=50, SP=6) Used: 15/20	Sprint 2 (max. 20 SP): --- R.10 (Uj=40, SP=3) --- R.13 (Uj=70, SP=9) --- R.14 (Uj=75, SP=7) Used: 19/20	Sprint 3 (max. 35 SP): --- R.02 (Uj=85, SP=8) --- R.05 (Uj=85, SP=9) --- R.07 (Uj=50, SP=6) --- R.15 (Uj=60, SP=7) --- R.55 (Uj=50, SP=4) Used: 34/35	Sprint 4 (max. 30 SP): --- R.06 (Uj=60, SP=5) --- R.08 (Uj=40, SP=4) --- R.16 (Uj=55, SP=7) --- R.17 (Uj=20, SP=4) --- R.18 (Uj=25, SP=4) --- R.19 (Uj=25, SP=5) Used: 29/30	Sprint 5 (max. 20 SP): --- R.20 (Uj=15, SP=6) --- R.53 (Uj=80, SP=7) --- R.54 (Uj=55, SP=7) Used: 20/20
---	---	---	---	---

Pilnais modelis

Savukārt pilnais modelis tika papildināts ar:

- Risku koeficientiem
- Korelācijām
- Priekšnosacījumiem

Šajā gadījumā optimizācijas algoritms spēja ne tikai izvēlēties prioritārākos uzdevumus, bet arī:

- Izvēlēties riskantākos uzdevumus
- Grupēt korelējošos uzdevumus
- Garantēt loģisku secību starp uzdevumiem

Kopējais lietderības rādītājs pilnajā modelī bija augstāks, kas norāda uz uzlabotu uzdevumu sadales kvalitāti un procesa efektivitāti.

• Salīdzinoša analīze

Tas ir vienkārši – pilnajā modelī tiek ņemts vērā lielāks parametru skaits, kā rezultātā tika panākts neliels efektivitātes pieaugums. Bet no tehniskā viedokļa šis modelis ir daudz sarežģītāks nekā parastais.

Atšķirība bija aptuveni 70 punkti no 1200-1300, kas var būt nenožīmīga. Tas ir atkarīgs no variantu un datu apjoma.

• SECINĀJUMI

Darba gaitā tika izstrādāts un analizēts optimizācijas modelis, kas paredzēts lietotāja stāstu (User Story) efektīvai plānošanai Agile programmatūras izstrādes procesā. Tika apskatīti un salīdzināti divi modeļa varianti – bāzes modelis un pilnais modelis, kurā tika integrēti papildus parametri: riska novērtējums, korelācijas starp uzdevumiem un atkarības (priekšnosacījumi).

Pilnais modelis sniedza augstāku kopējo lietderības vērtību. Tādējādi tas labāk atspoguļo reālās dzīves situācijas.

Darba rezultātā secināms:

Lineārās programmēšanas izmantošana ir efektīva pieeja uzdevumu sadales automatizācijai.

Korelāciju un atkarību iekļaušana uzlabo plānošanas kvalitāti.

Pilnais modelis ir piemērotāks vidēja un liela apjoma projektiem, kur nepieciešama precīza resursu un laika vadība.

● ATSAUCES

https://github.com/nikitapsenicnihs/iop_stud-darbs - tur ir mani faili