Schriftliche Ausarbeitung zum Thema:

Optimierung des Task-Managements in einem Software-Entwicklungsteam

Vorgelegt von:

Florian Particke
Lupinenweg 28
91058 Erlangen

Matrikelnummer: 9231757

Modul: Planen mit mathematischen Modellen
Semester: Wintersemester 2025/26

Inhaltsverzeichnis

1. Einführung

In Softwareentwicklungsteams treffen Aufgaben (im Folgenden *Tasks*) oft zufällig ein und müssen mit begrenzten Ressourcen bearbeitet werden. Klassische Warteschlangenmodelle wie das M/M/1-System bieten hierfür ein gut untersuchtes Grundmodell. Sie stoßen jedoch in der Praxis an Grenzen, da je nach Laufzeit beliebig große Warteschlangen entstehen können. Aus diesem Grund werden Tasks häufig in geplanten Intervallen gebündelt, priorisiert oder teilweise verworfen – etwa im Rahmen agiler Sprintplanungen. Ziel dieser Ausarbeitung ist es, eine ereignisbasierte Simulation eines Einkanal-Bedienungssystems zu untersuchen und zwei Bearbeitungsstrategien miteinander zu vergleichen:

- **Strategie 1:** Kontinuierliche Abarbeitung nach FIFO (M/M/1), also „Abarbeiten auf Zuruf“
- **Strategie 2:** Periodische Sprintplanung mit begrenzter Kapazität

Beide Ansätze werden in Python implementiert. Der zugehörige Code befindet sich im Anhang. Funktionsreferenzen finden sich in *main.py* in den jeweiligen Kommentaren.

2. Modelle

2.1 Strategie 1 „Abarbeiten auf Zuruf“

Strategie 1 bildet ein klassisches Einkanal-Bedienungssystem ab. Die Annahmen lauten:

- K1: Poisson-Ankunftsprozess mit Rate α → exponentialverteilte Zwischenankunftszeiten.
- K2: Poisson-Bedienprozess mit Rate β → exponentialverteilte Bediendauern.
- K3: Eine Ressource (vereinfachend: ein Team)
- K4: Unendlicher Warteraum
- K5: FIFO-Warteschlange

Die Simulation erfolgt ereignisbasiert. Bei jedem Ankunftsereignis wird entschieden, ob ein Task sofort bearbeitet oder in die Warteschlange eingeordnet wird. Die Bearbeitung erfolgt strikt in der Reihenfolge des Eintreffens.

2.2 Strategie 2 „Sprintplanung“

Strategie 2 erweitert das Basismodell um eine periodische Planungslogik, wie man sie aus agilen Entwicklungsprozessen kennt. Die Annahmen K1–K3 sind identisch zu Strategie 1. Zusätzlich gelten:

- K4: Unendlicher Warteraum, aber Kapazitätslimit pro Sprint
- K5: Auswahl der Tasks zu Sprintbeginn per Zufall und Priorisierung nach Sprintkapazität

Der Ablauf ist wie folgt:

- Es existiert ein festes Planungsintervall T (Sprintdauer).

- Alle während eines Sprints eingehenden Tasks werden in einem Backlog gesammelt.
- Zu Beginn jedes Sprints:
    - Zufällige Auswahl der Tasks
    - Abarbeitung bis zur maximalen Sprintkapazität
    - Nicht ausgewählte Tasks werden verworfen.

3. Ergebnisse

3.1 Ergebnisse Strategie 1: „Abarbeiten auf Zuruf“

In diesem Abschnitt wird zuerst ein Beispiel mit einem kleinen Simulationszeitraum dargestellt, bevor die relevanten Output-Metriken mit Konfidenzintervallen für eine größere Wiederholungszahl ermittelt werden.

3.1.1 Beispielhafte Auswertung

Für ein kurzes Simulationsbeispiel gelten folgende Parameter:

- Ankunftsrate α = 1.5 Tasks / Tag
- Bedienrate β = 1 Tasks / Tag
- Simulationsdauer $T_s = 10$ Tage.

Die detaillierten Ergebnisse finden sich in Anhang I.

Mit der Datentabelle ergeben sich folgende Werte für die Output Metriken unter Zuhilfenahme der Formeln aus dem Lehrbrief: 8 Tasks wurden bearbeitet, die Schlangenlänge ist am Ende der Simulation $\zeta = 6$ Tasks, die mittlere Wartezeit $\bar{\omega} = 2.73$ Tage, die mittlere Schlangenlänge $\bar{\zeta} = 2.68$ Tasks und der Auslastungsgrad $\bar{\rho} = 0.81$. Schon bei kurzen Laufzeiten zeigt sich, dass die FIFO-Abarbeitung bei hoher Auslastung schnell an Grenzen stößt, da die mittlere Wartezeit stark ansteigt.

3.1.2 Konfidenzintervalle

Zur Bewertung der statistischen Unsicherheit wird ein Konfidenzniveau von $1 - \alpha = 95\%$ verwendet. Die Konfidenzintervalle für

- mittlere Wartezeit $\bar{\omega}$,
- mittlere Schlangenlänge $\bar{\zeta}$ und
- Auslastungsgrad $\bar{\rho}$

werden anhand der Stichprobenmittelwerte und -varianzen über $N = 10000$ Durchläufe ermittelt. Die Simulationsdauer beträgt $T_s = 240$ Tage, was einer vereinfachten Annahme von Arbeitstagen pro Jahr entspricht. Die Berechnung erfolgt gemäß den Formeln aus dem Lehrbrief. Die verwendeten Daten stehen in *log/analyse_strategie_1.log*.

Für eine Variable x führt dies zu

$$\bar{\bar{x}} \pm \frac{u_{0.975} \cdot s_{10000}}{\sqrt{N}}.$$

Daraus ergibt sich beispielhaft für die Berechnung des Konfidenzintervalls für eine über $N = 10000$ Durchläufe gemittelte mittlere Wartezeit $\bar{\bar{\omega}}$: $[\bar{\bar{\omega}} - \frac{1.96 \cdot s_{10000}}{100}; \bar{\bar{\omega}} + \frac{1.96 \cdot s_{10000}}{100}]$.

Es ergeben sich folgende Konfidenzintervalle:

- Mittlere Wartezeit $\bar{\bar{\omega}}$: [61.21; 61.78]
- Mittlere Schlangenlänge $\bar{\bar{\zeta}}$: [60.61; 61.103]
- Auslastungsgrad $\bar{\bar{\rho}}$:[0.9944; 0.9946]

Von besonderem Interesse ist der Vergleich der mittleren Wartezeit und der Anzahl der verworfenen Tasks im Rahmen von Strategie 2 („Sprintplanung“), da diese Kennzahlen in der Praxis eine hohe Relevanz besitzen. Im Gegensatz dazu liegt die Anzahl der verworfenen Tasks bei Strategie 1 („Abarbeiten auf Zuruf“) stets bei $n_T = 0$. Dieser spezifische Vergleich wird im folgenden Abschnitt detailliert analysiert.

3.2 Vergleich mit Strategie 2: „Sprintplanung“

3.2.1 Beispielhafte Auswertung

Parameter:

- Ankunftsrate α = 1.5 Tasks/Tag
- Bedienrate β = 1 Task/Tag
- Sprintdauer $T = 10$ Tage
- Simulationsdauer $T_s = 30$ Tage
- Kapazität pro Sprint: 10 Tasks

Die mittlere Wartezeit liegt bei $\bar{\omega} = 10.42$ Tage und die Anzahl der verworfenen Tasks bei $n_T = 7$. Im Gegenzug zur Reduktion der mittleren Wartezeit müssen jedoch Tasks verworfen werden, wenn die Sprintkapazität überschritten wird – was insbesondere bei hohen Ankunftsraten häufig vorkommt. Die detaillierten Ergebnisse stehen im Anhang II.

3.2.2 Konfidenzintervalle

Zur Bewertung der statistischen Unsicherheit werden die Konfidenzintervalle analog zu Abschnitt 3.1.2 berechnet, wobei die Anzahl der Simulationsläufe wiederum $N = 10000$ beträgt. Die Simulationsparameter sind äquivalent zu Abschnitt 3.2.1 mit Ausnahme der Simulationsdauer $T_s = 240$ Tage. Die zur Berechnung hinzugezogenen Ergebnisse befinden sich in “*log/analyse_strategie_2.log*”.

Es ergeben sich folgende Konfidenzintervalle:

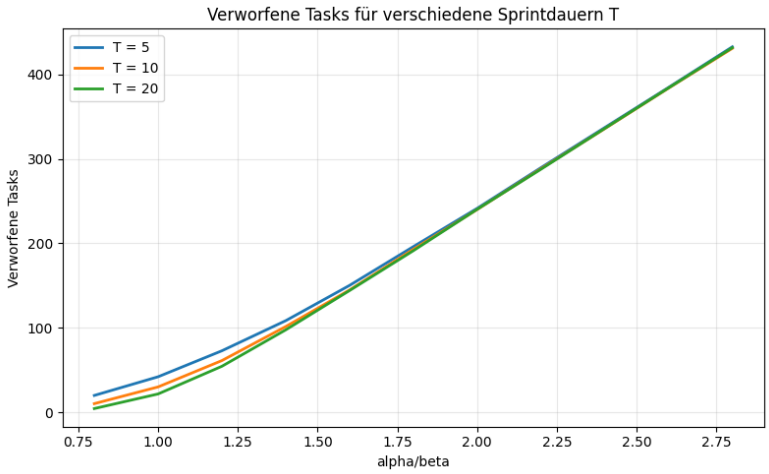
- Mittlere Wartezeit $\bar{\bar{\omega}}$: [8.98; 8.99]
- Anzahl der verworfenen Tasks n_T: [122.72; 123.42]

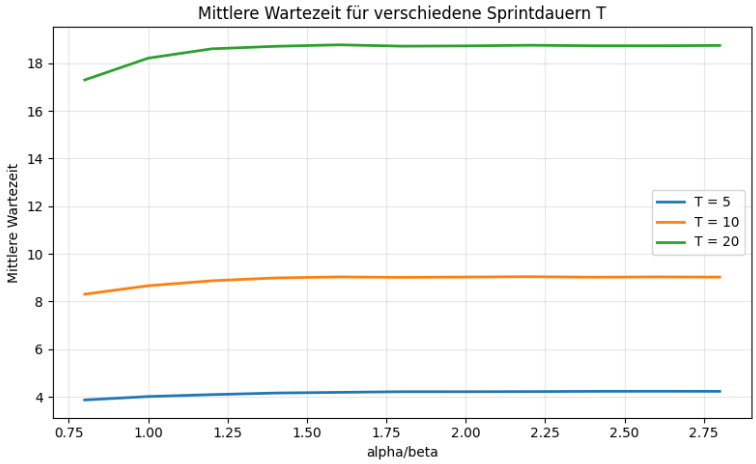
Die Wartezeit sinkt deutlich im Vergleich zu Strategie 1, allerdings auf Kosten einer hohen Anzahl verworfener Tasks.

3.2.3 Parameterstudie: Sprintintervalle und Ankunftsraten

Untersucht werden verschiedene Kombinationen aus Sprintdauer *T* und Verhältnis *α/β*. Die Parameter werden folgendermaßen gesetzt: Bedienrate *β* = 1 Task / Tag und die Simulationsdauer $T_s = 240$ Tage. Die Kapazität pro Sprint entspricht immer der Sprintdauer. Zu

untersuchende Output Metriken sind die mittlere Wartezeit $\bar{\omega}$ und die Anzahl der verworfenen Tasks n_T. Als Anzahl der Simulationsdurchläufe pro Parameterkombination wird $N = 1000$ Durchläufe gesetzt. Alle zur Darstellung verwendeten Daten finden sich in „*log/analyse_parameter_schar_strategie2.log*“.





Ergebnisse:

- Die Anzahl verworfener Tasks n_T steigt nahezu linear mit der Ankunftsrate α.
- Die mittlere Wartezeit $\bar{\omega}$ bleibt durch die Struktur der Sprintplanung weitgehend unabhängig von α und liegt stets knapp unter der Sprintdauer.

4. Fazit

Die Simulation zeigt, dass bei steigender Ankunftsrate eine Sprintplanung notwendig sein kann, um die mittlere Wartezeit zu begrenzen. Kürzere Sprintintervalle ermöglichen eine höhere Reaktivität auf eingehende Tasks. Gleichzeitig steigt aber die Zahl der verworfenen Tasks. Hier müssen Teams je nach Zielsetzung einen geeigneten Kompromiss finden.

Anhang

Es bezeichnen

ID den eindeutigen Identifier des Tasks,
a_i die Zwischenankunftszeit des *i*-ten Tasks, das ist die Zeit zwischen Ankunft des *i*-ten und (*i*-1)-ten Tasks,
$t_i = \sum_{j=1}^{i} a_j$ den Ankunftszeitpunkt des *i*-ten Tasks,
b_i die Bediendauer des *i*-ten Tasks,
$e_i = \max(t_i,\ e_{i-1}) + b_i$ den Zeitpunkt des Bedienendes des *i*-ten Task,
$\omega_i = \max(0,\ e_{i-1} - t_i)$ die Wartezeit des *i*-ten Tasks vor Abarbeitung,
$\sum_j^i b_j$ die Gesamtbedienzeit,
$\sum_j^i \omega_j$ die gesamte Wartezeit.

Zusätzlich wird bei der beispielhaften Auswertung Strategie 2 „Sprintplanung“ der zugehörige Sprint mit angegeben.

I. Beispielhafte Auswertung Strategie 1 „Abarbeiten auf Zuruf“ – Datentabelle

ID	a_i	t_i	b_i	e_i	ω_i	$\sum_j^i b_j$	$\sum_j^i \omega_j$
1	2.182	2.182	0.4998	2.6818	0	0.4998	0
2	0.9887	3.1708	0.3319	3.5026	0	0.8317	0
3	0.0693	3.24	1.1934	4.696	0.2626	2.0251	0.2626
4	0.1178	3.3578	0.493	5.1891	1.3382	2.5181	1.6008
5	0.4853	3.8432	2.499	7.6881	1.3459	5.0171	2.9467
6	0.3998	4.2429	0.0396	7.7277	3.4452	5.0568	6.3918
7	0.2574	4.5004	2.0695	9.7972	3.2273	7.1263	9.6192
8	0.7271	5.2275	0.064	9.8612	4.5697	7.1902	14.1889
9	0.4735	5.701	0.2356	10.0968	4.1602	7.4259	18.3491
10	0.4945	6.1956	0.3595	10.4563	3.9013	7.7854	22.2504
11	0.6613	6.8569	1.9167	12.373	3.5995	9.702	25.8498
12	1.5629	8.4198	0.6393	13.0122	3.9532	10.3413	29.803
13	0.5939	9.0137	0.5619	13.5741	3.9986	10.9032	33.8016
14	0.1975	9.2112	0.6831	14.2572	4.3629	11.5863	38.1646

II. Beispielhafte Auswertung Strategie 2 „Sprintplanung“ – Datentabelle

Sprint	ID	a_i	t_i	b_i	e_i	ω_i	$\sum_j^i \omega_j$
1	10	0.5263	8.6568	1.6801	11.6801	1.3432	1.3432
1	7	3.0418	4.9507	1.0913	12.7715	6.7294	8.0726
1	5	2.6236	1.4324	0.2822	13.0537	11.3391	19.4117
1	4	0.0277	1.4047	0.3493	13.403	11.649	31.0607

1	12	0.6094	9.2962	2.7556	16.1586	4.1068	35.1675
1	3	0.4583	0.9463	0.096	16.2546	15.2123	50.3798
1	11	0.1131	9.1831	1.2773	17.5319	7.0715	57.4513
1	13	0.6949	9.9055	0.4945	18.0264	7.6264	65.0777
1	2	0.0323	0.9141	0.1328	18.1592	17.1123	82.19
1	1	0.002	0.9121	0.9263	19.0855	17.2471	99.4371
2	23	1.4699	16.0776	2.9115	22.9115	3.9224	103.3595
2	18	0.6575	13.6843	4.4195	27.331	9.2272	112.5866
2	20	0.0082	15.5951	0.5048	27.8358	11.7359	124.3226
2	19	1.2533	14.3418	0.212	28.0479	13.494	137.8166
2	14	0.1956	10.6004	0.9108	28.9587	17.4475	155.264
2	24	2.6733	17.5476	0.5673	29.526	11.4111	166.6751