Optimization Model for SCRUM-Based Software Development

AI Assistant

September 4, 2025

Abstract

This document presents a formal mathematical optimization model for resource allocation and project planning within a software development company utilizing the SCRUM framework. The model is derived from a defined domain of entities (e.g., Teams, Workers, Tasks) and their relationships. It aims to maximize efficiency and output while minimizing delays and blockers through a set of goals, constraints, and decision variables.

Contents

1 Sets (Entities)

- P = $\{p_1, p_2, ..., p_n\}$: Set of all Projects.
- T = $\{t_1, t_2, ..., t_n\}$: Set of all Teams.
- W = $\{w_1, w_2, ..., w_n\}$: Set of all Workers.
- F = $\{f_1, f_2, ..., f_n\}$: Set of all Features.
- S = $\{s_1, s_2, ..., s_n\}$: Set of all Skills.
- US = $\{us_1, us_2, ..., us_n\}$: Set of all User Stories.
- TSK = $\{tsk_1, tsk_2, ..., tsk_n\}$: Set of all Tasks.
- SP = $\{sp_1, sp_2, ..., sp_n\}$: Set of all Sprints.
- SBL = $\{sbl_1, sbl_2, ..., sbl_n\}$: Set of all Sprint Backlogs.
- BL = $\{bl_1, bl_2, ..., bl_n\}$: Set of all Blockers.
- PO = $\{po_1, po_2, ..., po_n\}$: Set of all Product Owners.
- SM = $\{sm_1, sm_2, ..., sm_n\}$: Set of all Scrum Masters.

2 Indices

- $p, p' \in$ P: Index over Projects.
- $t \in$ T: Index over Teams.
- $w \in$ W: Index over Workers.
- $f \in$ F: Index over Features.
- $s \in$ S: Index over Skills.
- $us \in$ US: Index over User Stories.
- $tsk \in$ TSK: Index over Tasks.
- $sp \in$ SP: Index over Sprints.
- $sbl \in$ SBL: Index over Sprint Backlogs.
- $bl \in$ BL: Index over Blockers.

3 Goals

G0 **maximize_team_velocity**: Maximize the average velocity of all teams.
Mathematical: $\max \sum_{t \in T} \text{velocity}_t$

G1 **minimize_project_duration**: Minimize the total duration of all projects.
Mathematical: $\min \sum_{p \in P} (\text{project\_end}_p - \text{project\_start}_p)$

G2 **minimize_blocker_severity**: Minimize the total severity of all active blockers.
Mathematical: $\min \sum_{bl \in BL} \text{severity}_{bl} \cdot \mathbb{I}(\text{status}_{bl} = \text{active})$

G3 maximize_feature_completion: Maximize the number of features with status 'done'.
Mathematical: $\max \sum_{f \in F} \mathbb{I}(\text{status}_f = \text{done})$

G4 minimize_task_effort_variance: Minimize the variance in estimated effort across all tasks in a sprint.
Mathematical: $\min \quad \text{Var}(\{\text{effort}_{tsk} \mid tsk \in \text{TSK}, \text{sprint}_{tsk} = sp\})$

G5 maximize_worker_utilization: Maximize the average utilization of all workers.
Mathematical: $\max \frac{1}{|\text{W}|} \sum_{w \in W} \text{utilization\_rate}_w$

G6 minimize_sprint_goal_failure: Minimize the number of sprints where the goal was not achieved.
Mathematical: $\min \sum_{sp \in SP} (1 - \text{achievement\_status}_{sp})$

4 Conditions

C0 team_has_scrum_master: Every team must have exactly one Scrum Master assigned.
Logical: $\forall t \in T, \exists! sm \in SM : \text{is\_supported\_by}(t, sm)$

C1 worker_availability_not_exceeded: The sum of effort from tasks assigned to a worker must not exceed their availability.
Mathematical: $\forall w \in W, \sum_{\substack{tsk \in TSK \\ \text{assigned}(tsk,w)}} \text{effort}_{tsk} \leq \text{availability}_w$

C2 sprint_backlog_effort_within_velocity: The total effort of tasks in a sprint backlog must not exceed the team's historical velocity.
Mathematical: $\forall sbl \in SBL, \sum_{tsk \in sbl} \text{effort}_{tsk} \leq \text{velocity}_{\text{team}(sbl)}$

C3 feature_requires_documentation: Every feature with status 'done' must have linked documentation.
Logical: $\forall f \in F, \text{status}_f = \text{done} \implies \exists \text{doc} \in \text{FED} : \text{documents\_feature}(\text{doc}, f)$

C4 task_assigned_to_skilled_worker: A task can only be assigned to a worker who has a required skill.
Logical: $\forall tsk \in TSK, \forall w \in W, \text{assigned}(tsk, w) \implies \text{has\_skill}(w, \text{required\_skill}(tsk))$

C5 project_budget_not_exceeded: The total cost of all resources on a project must not exceed the project budget.
Mathematical: $\forall p \in P, \sum_{w \in \text{team}(p)} (\text{cost}_w \cdot \text{time}_w) \leq \text{budget}_p$

C6 user_story_has_acceptance_criteria: No user story can be added to a sprint backlog without defined acceptance criteria.
Logical: $\forall us \in US, \text{in\_sprint\_backlog}(us) \implies \text{acceptance\_criteria}_{us} \neq \emptyset$

C7 minimize_context_switching: Prefer assigning a worker to fewer concurrent tasks.
Mathematical: $\min \sum_{w \in W} \text{number\_of\_concurrent\_tasks}_w$ (Soft Constraint)

C8 prefer_high_priority_features: Assign resources to higher priority features first.
Logical: $\forall f, f' \in F, \text{priority}_f > \text{priority}_{f'} \implies \text{resources\_assigned}(f) \geq \text{resources\_assigned}(f')$ (Soft Constraint)

5 Decision Variables

- $\text{assign}_{tsk,w} \in \{0, 1\}$: (DV0) Binary assignment of Task tsk to Worker w.
- $\text{start}_{sp} \in \mathbb{Z}^+$: (DV1) Start date of Sprint sp.

– $\text{prio}_f \in \{1, 2, 3, 4, 5\}$: (DV2) Numerical priority of Feature f.

– $\text{util}_w \in [0, 1]$: (DV3) Utilization rate of Worker w.

– $\text{points}_{us} \in \{1, 2, 3, 5, 8, 13\}$: (DV4) Story points for User Story us.

– $\text{target\_vel}_{t,sp} \in [5, 50]$: (DV5) Target velocity for Team t in Sprint sp.

– $\text{stat}_p \in \{0, 1, 2\}$: (DV6) Status of Project p.

– $\text{goal}_{sp} \in \{0, 1\}$: (DV7) Achievement of Sprint Goal for Sprint sp.

– $\text{resolve}_{bl} \in [0, 30]$: (DV8) Resolution time for Blocker bl.

– $\text{req\_level}_{tsk,s} \in \{1, 2, 3, 4, 5\}$: (DV9) Required skill level for Task tsk.