Mathematical Optimization Model for Scrum-based Software Development

Formulated by Gemini AI

September 4, 2025

Contents

1 Sets (Entities)

This section defines the fundamental sets used in the model, based on the provided `Entities.csv`. Each set represents a collection of a specific type of entity within the software development process.

- P: The set of all projects.
- T: The set of all teams.
- W: The set of all workers.
- F: The set of all features.
- S: The set of all skills.
- R: The set of all roles.
- PO: The set of all Product Owners.
- SM: The set of all Scrum Masters.
- PB: The set of all Product Backlogs.
- SP: The set of all Sprints.
- SPP: The set of all Sprint Planning meetings.
- DS: The set of all Daily Scrums.
- SR: The set of all Sprint Reviews.
- SRE: The set of all Sprint Retrospectives.
- SBL: The set of all Sprint Backlogs.
- SG: The set of all Sprint Goals.
- E: The set of all Epics.
- US: The set of all User Stories.
- TSK: The set of all Tasks.
- DEV: The set of all Development Snapshots.
- BL: The set of all Blockers.
- SH: The set of all Stakeholders.
- VEL: The set of all Velocity records.
- REP: The set of all Release Plans.
- RM: The set of all Roadmaps.
- SCB: The set of all Scrum Boards.
- FED: The set of all Feature Documentations.

2 Indices

This section defines the indices used to refer to specific elements within the sets defined in Section 1.

- $p \in P$: An index for a specific project.
- $t \in T$: An index for a specific team.
- $w \in W$: An index for a specific worker.
- $f \in F$: An index for a specific feature.
- $s \in S$: An index for a specific skill.
- $r \in R$: An index for a specific role.
- $po \in PO$: An index for a specific Product Owner.
- $sm \in SM$: An index for a specific Scrum Master.
- $pb \in PB$: An index for a specific Product Backlog.
- $sp \in SP$: An index for a specific Sprint.
- $us \in US$: An index for a specific User Story.
- $tsk \in TSK$: An index for a specific Task.
- $rep \in REP$: An index for a specific Release Plan.

3 Goals

The primary objective of the model is to find a solution that optimizes a weighted combination of the following goals. The overall objective function is a summation of these individual terms, where each term is multiplied by its corresponding weight from `Goals.csv`. Let Z be the objective function value. The overall goal is to maximize Z.

Let prio_f, effort_f, prio_p, budget_p, etc., be the parameters representing the attributes of the entities.

- **maximize_feature_priority (G0)**: Maximize the total priority of all implemented features.
  Mathematical Form: $+w_{G0} \cdot \sum_{f \in F} \sum_{rep \in REP} \text{prio}_f \cdot \text{select\_feature\_for\_release}_{f,rep}$
- **minimize_feature_estimated_effort (G1)**: Minimize the total effort of implemented features.
  Mathematical Form: $-w_{G1} \cdot \sum_{f \in F} \sum_{rep \in REP} \text{effort}_f \cdot \text{select\_feature\_for\_release}_{f,rep}$
- **maximize_project_priority (G2)**: Maximize the priority of selected projects.
  Mathematical Form: $+w_{G2} \cdot \sum_{p \in P} \text{prio}_p \cdot \text{start\_project}_p$
- **minimize_project_budget (G3)**: Minimize the budget of selected projects.
  Mathematical Form: $-w_{G3} \cdot \sum_{p \in P} \text{budget}_p \cdot \text{start\_project}_p$
- **maximize_stakeholder_influence_satisfaction (G4)**: Maximize the influence level of stakeholders whose desired features are selected.
  Mathematical Form: $+w_{G4} \cdot \sum_{f \in F} \sum_{sh \in SH} \text{influence\_level}_{sh} \cdot \text{relevance}_{f,sh} \cdot \text{select\_feature\_for\_release}_{f,rep}$
- **maximize_sprint_goal_achievement (G10)**: Maximize the achievement status of sprint goals.
  Mathematical Form: $+w_{G10} \cdot \sum_{sp \in SP} \text{achievement\_status}_{sp}$

4 Conditions

These are the constraints that any feasible solution must satisfy. They represent the rules, limits, and logical dependencies of the Scrum process.

- **enforce_team_size_limits (C0)**: The number of workers assigned to any given team must be between 3 and 9.
  Mathematical Form: $3 \leq \sum_{w \in W} \text{assign\_worker\_to\_team}_{w,t} \leq 9 \quad \forall t \in T$
- **respect_sprint_capacity (C1)**: The sum of story points from user stories selected for a sprint cannot exceed the velocity of the team assigned to that sprint.
  Mathematical Form: $\sum_{us \in US} \text{points}_{us} \cdot \text{select\_user\_story\_for\_sprint}_{us,sp} \leq \text{velocity}_t \quad \forall sp \in SP, \forall t \in T$ where team t is assigned to sprint sp.
- **worker_must_be_active (C2)**: Tasks cannot be assigned to a worker unless their status is 'active'. Let $\text{IsActive}_w \in \{0,1\}$ be a parameter.
  Mathematical Form: $\text{assign\_task\_to\_worker}_{tsk,w} \leq \text{IsActive}_w \quad \forall tsk \in TSK, \forall w \in W$
- **assign_skilled_worker (C3)**: A worker can only be assigned to a task if they possess the required skill. Let $\text{HasSkill}_{w,s} \in \{0,1\}$ and $\text{NeedsSkill}_{tsk,s} \in \{0,1\}$ be parameters.
  Mathematical Form: $\text{NeedsSkill}_{tsk,s} \cdot \text{assign\_task\_to\_worker}_{tsk,w} \leq \text{HasSkill}_{w,s} \quad \forall tsk \in TSK, \forall w \in W, \forall s \in S$
- **respect_project_budget (C4)**: The total cost of resources (e.g., worker-hours) for a project must not exceed the project's budget.
  Mathematical Form: $\sum_{t \in T} \sum_{w \in W} (\text{cost}_w \cdot \text{assign\_worker\_to\_team}_{w,t} \cdot \text{assign\_team\_to\_project}_{t,p}) \leq \text{budget}_p \quad \forall p \in P$
- **worker_assigned_to_one_team_only (C5)**: Each worker can be assigned to at most one team.
  Mathematical Form: $\sum_{t \in T} \text{assign\_worker\_to\_team}_{w,t} \leq 1 \quad \forall w \in W$
- **sprint_must_have_a_goal (C6)**: Every active sprint must have one associated sprint goal.
  Mathematical Form: $\sum_{sg \in SG} \text{assign\_goal\_to\_sprint}_{sg,sp} = 1 \quad \forall sp \in SP$ where sprint sp is active.

5 Decision Variables

These are the variables that the optimization model will determine. They represent the decisions that need to be made.

- **assign_team_to_project (DV0)**: $\text{assign\_team\_to\_project}_{t,p} \in \{0,1\} \quad \forall t \in T, p \in P$. Binary variable: 1 if team t is assigned to project p, 0 otherwise.
- **assign_worker_to_team (DV1)**: $\text{assign\_worker\_to\_team}_{w,t} \in \{0,1\} \quad \forall w \in W, t \in T$. Binary variable: 1 if worker w is assigned to team t, 0 otherwise.
- **select_feature_for_release (DV2)**: $\text{select\_feature\_for\_release}_{f,rep} \in \{0,1\} \quad \forall f \in F, rep \in REP$. Binary variable: 1 if feature f is included in release plan rep, 0 otherwise.
- **select_user_story_for_sprint (DV3)**: $\text{select\_user\_story\_for\_sprint}_{us,sp} \in \{0,1\} \quad \forall us \in US, sp \in SP$. Binary variable: 1 if user story us is selected for sprint sp, 0 otherwise.
- **assign_task_to_worker (DV4)**: $\text{assign\_task\_to\_worker}_{tsk,w} \in \{0,1\} \quad \forall tsk \in TSK, w \in W$. Binary variable: 1 if task tsk is assigned to worker w, 0 otherwise.

- **start_project (DV5)**: $\text{start\_project}_p \in \{0, 1\} \quad \forall p \in P$. Binary variable: 1 if project p is started, 0 otherwise.
- **number_of_workers_in_team (DV6)**: $\text{num\_workers}_t \in \mathbb{Z}^+ \quad \forall t \in T$. Integer variable for the number of workers in team t. Note: This is an auxiliary variable, often defined by constraints on other variables (see C0).