Mathematical Optimization Model for a Scrum-based Software Development Process

Gemini AI Assistant

September 4, 2025

Contents

1 Sets (Entities)

- Project (P): The product or initiative to be developed.
- Team (T): Self-organized, cross-functional development team.
- Worker (W): Individual team member working on the project.
- Feature (F): Mid-sized functionality.
- Skill (S): Professional or social competence of a worker.
- Role (R): Defined responsibilities within the Scrum team.
- ProductOwner (PO): Responsible for product vision and Product Backlog.
- ScrumMaster (SM): Supports the team in applying Scrum.
- ProductBacklog (PB): Ordered list of all requirements.
- Sprint (SP): Fixed time period for creating an increment.
- SprintPlanning (SPP): Kick-off meeting for Sprint preparation.
- DailyScrum (DS): Daily 15-minute team meeting.
- SprintReview (SR): Presentation and acceptance of results.
- SprintRetrospective (SRE): Retrospective for process improvement.
- SprintBacklog (SBL): Selected backlog items + implementation plan.
- SprintGoal (SG): Objective to be achieved within the sprint.
- Epic (E): Large requirement that can be split into stories.
- UserStory (US): Requirement from the perspective of a user.
- Task (TSK): Smallest unit of work within a sprint.
- DevelopmentSnapshot (DEV): Product at the end of a sprint.
- Blocker (BL): Obstacle hindering progress.
- Stakeholder (SH): Interested party in the product (internal/external).
- Velocity (VEL): Average amount of work per sprint for a team.
- ReleasePlan (REP): Plan for releasing specific features.
- Roadmap (RM): Long-term planning across releases.
- ScrumBoard (SCB): Visual representation of tasks during the sprint.
- FeatureDocumentation (FED): Documentation for a specific feature.

2 Indices

- $p \in P$: Index for the set of Projects.
- $t \in T$: Index for the set of Teams.
- $w \in W$: Index for the set of Workers.
- $f \in F$: Index for the set of Features.
- $s \in S$: Index for the set of Skills.
- $r \in R$: Index for the set of Roles.
- $po \in PO$: Index for the set of Product Owners.
- $sm \in SM$: Index for the set of Scrum Masters.
- $pb \in PB$: Index for the set of Product Backlogs.
- $sp \in SP$: Index for the set of Sprints.
- $spp \in SPP$: Index for the set of Sprint Plannings.
- $ds \in DS$: Index for the set of Daily Scrums.
- $sr \in SR$: Index for the set of Sprint Reviews.
- $sre \in SRE$: Index for the set of Sprint Retrospectives.
- $sbl \in SBL$: Index for the set of Sprint Backlogs.
- $sg \in SG$: Index for the set of Sprint Goals.
- $e \in E$: Index for the set of Epics.
- $us \in US$: Index for the set of User Stories.
- $tsk \in TSK$: Index for the set of Tasks.
- $dev \in DEV$: Index for the set of Development Snapshots.
- $bl \in BL$: Index for the set of Blockers.
- $sh \in SH$: Index for the set of Stakeholders.
- $vel \in VEL$: Index for the set of Velocities.
- $rep \in REP$: Index for the set of Release Plans.
- $rm \in RM$: Index for the set of Roadmaps.
- $scb \in SCB$: Index for the set of Scrum Boards.
- $fed \in FED$: Index for the set of Feature Documentations.

3 Decision Variables

DV0 **assign_worker_to_task** ($assign_{w,tsk}$): A binary variable indicating if worker w is assigned to task tsk.

$$assign_{w,tsk} \in \{0,1\} \quad \forall w \in W, \forall tsk \in TSK$$

DV1 **select_user_story_for_sprint** ($select_{us,sp}$): A binary variable indicating if user story us is selected for sprint sp.

$$select_{us,sp} \in \{0,1\} \quad \forall us \in US, \forall sp \in SP$$

DV2 **select_feature_for_release** ($select_{f,rep}$): A binary variable indicating if feature f is included in release plan rep.

$$select_{f,rep} \in \{0,1\} \quad \forall f \in F, \forall rep \in REP$$

DV3 **assign_team_to_sprint** ($assign_{t,sp}$): A binary variable indicating if team t is assigned to work on sprint sp.

$$assign_{t,sp} \in \{0,1\} \quad \forall t \in T, \forall sp \in SP$$

DV4 **sprint_start_date** ($start\_date_{sp}$): An integer variable representing the start day of sprint sp.

$$start\_date_{sp} \in \mathbb{Z}^+ \quad \forall sp \in SP$$

DV5 **feature_priority_value** ($set\_priority_f$): An integer variable to set the priority level of a feature f.

$$set\_priority_f \in \{1, ..., 5\} \quad \forall f \in F$$

DV6 **worker_allocation_percentage** ($alloc_{w,sp}$): A continuous variable for the percentage of worker w's capacity allocated to sprint sp.

$$0 \leq alloc_{w,sp} \leq 1 \quad \forall w \in W, \forall sp \in SP$$

DV7 **budget_allocated_to_epic** ($budget_e$): A continuous variable representing the amount of budget assigned to epic e.

$$budget_e \geq 0 \quad \forall e \in E$$

4 Goals

Let parameters such as $priority_f$, $story\_points_{us}$, $budget_p$, etc., be the attribute values of the respective entities.

G0 **maximize_total_feature_priority**: Maximize the sum of priorities for all features selected for implementation.

$$\max \sum_{f \in F} \sum_{rep \in REP} priority_f \cdot select_{f,rep}$$

G1 **maximize_total_story_points_in_sprint**: Maximize the total story points of user stories completed within a sprint.

$$\max \sum_{us \in US} \sum_{sp \in SP} story\_points_{us} \cdot select_{us,sp}$$

G2 **minimize_total_project_budget**: Minimize the overall budget spent on the project. This can be modeled as the sum of budgets allocated to epics.

$$\min \sum_{e \in E} budget_e$$

G3 **minimize_project_end_date**: Minimize the end date of the last sprint.

$$\min \left(\max_{sp \in SP} (start\_date_{sp} + duration_{sp}) \right)$$

G4 **maximize_stakeholder_influence_satisfaction**: Maximize the value delivered to stakeholders, weighted by their influence. Assumes a parameter $value_{f,sh}$ for the value of a feature to a stakeholder.

$$\max \sum_{f \in F} \sum_{rep \in REP} \sum_{sh \in SH} (influence_{sh} \cdot value_{f,sh} \cdot select_{f,rep})$$

G5 **minimize_blocker_severity**: Minimize the severity of unresolved blockers. Let is_active_{bl} be a binary parameter.

$$\min \sum_{bl \in BL} severity_{bl} \cdot is\_active_{bl}$$

G7 **minimize_total_task_effort**: Minimize the sum of effort for all tasks associated with selected user stories. Let $part\_of_{tsk,us}$ be a binary parameter.

$$\min \sum_{tsk \in TSK} \sum_{us \in US} \sum_{sp \in SP} effort_{tsk} \cdot part\_of_{tsk,us} \cdot select_{us,sp}$$

5 Conditions

Let parameters be derived from entity attributes and relationships. M is a large constant (Big-M method).

C0 **worker_must_be_available**: A worker can only be assigned to tasks if their status is available. Let $is\_available_w$ be a binary parameter.

$$assign_{w,tsk} \leq is\_available_w \quad \forall w \in W, \forall tsk \in TSK$$

C2 **team_capacity_not_exceeded**: The sum of story points for a team in a sprint must not exceed the team's velocity. Let $velocity_t$ be the team's average velocity.

$$\sum_{us \in US} story\_points_{us} \cdot select_{us,sp} \leq velocity_t \cdot assign_{t,sp} \quad \forall t \in T, \forall sp \in SP$$

C3 **worker_must_have_skill**: A worker can only be assigned to a task if they possess the required skill. Let $requires_{tsk,s}$ and $has_{w,s}$ be binary parameters.

$$requires_{tsk,s} \cdot assign_{w,tsk} \leq has_{w,s} \quad \forall w \in W, \forall tsk \in TSK, \forall s \in S$$

C4 **prefer_high_priority_stories**: This is a soft constraint, better modeled as a weighted goal. Included in G0/G1. For a hard constraint: only stories above a certain priority P_{min} can be selected.

$$(priority_{us} - P_{min}) \cdot select_{us,sp} \geq 0 \quad \forall us \in US, \forall sp \in SP$$

C6 **sprint_must_have_goal**: Every sprint must have an associated goal. Modeled by ensuring a sprint is active only if a goal is defined.

$$\sum_{sg \in SG} defined_{sp,sg} = 1 \quad \forall sp \in SP$$

C8 **team_must_be_active**: Only active teams can be assigned to sprints. Let is_active_t be a binary parameter.

$$assign_{t,sp} \leq is\_active_t \quad \forall t \in T, \forall sp \in SP$$

C9 **epic_must_be_approved**: A user story can only be selected if its parent epic is approved. Let $belongs_{us,e}$ and $is\_approved_e$ be binary parameters.

$$select_{us,sp} \leq is\_approved_e \quad \forall us \in US, \forall sp \in SP, \forall e \in E \text{ where } belongs_{us,e} = 1$$

C10 **project_must_be_in_progress**: Sprints can only be active if the project is in progress. Let is_active_p be a binary parameter.

$$assign_{t,sp} \leq is\_active_p \quad \forall t \in T, \forall sp \in SP, \forall p \in P$$