Multi-Objective Optimization Model for a Scrum-Based Software Development Company

Generated from Entities, Relationships, Goals, Conditions, Decision Variables CSVs

August 12, 2025

Contents

Modeling Overview

We formulate a mixed-integer, multi-objective model that uses: (i) the **Entities** as sets and attributes, (ii) the **Relationships** as incidence sets between entities, (iii) the provided **Decision Variables**, and (iv) **Goals** and **Conditions** as (soft/hard) objectives and constraints. We scalarize the goals via a weighted sum with signs derived from `GoalType`.

Notation conventions. For each entity attribute used in goals/conditions we introduce parameters named intuitively:

a_w : Worker availability (from `Worker.availability`)
b_p : Project budget ceiling (from `Project.budget`)
sp_{us} : UserStory story points (from `UserStory.story_points`)
ef_{tsk} : Task effort (from `Task.effort`)
pr_f : Feature priority (from `Feature.priority`)
nt_{sbl} : SprintBacklog number of tasks (from `SprintBacklog.number_of_tasks`)
ag_{sp} : Sprint goal achievement (from `Sprint.achievement_of_goal`)
ts_{sre} : Team satisfaction (from `SprintRetrospective.team_satisfaction`)
sev_{bl} : Blocker severity (from `Blocker.severity`)
$\overline{\mathrm{sp}}_{vel}$: Velocity average story points (from `Velocity.avg._story_points`)
tr_{vel} : Velocity trend (from `Velocity.trend`)
att_{sr} : SprintReview attendees count (from `SprintReview.attendees_count`)

For relationships, we use incidence sets, e.g. $(us, tsk) \in \mathcal{E}^{US\text{-}TSK}$ iff `UserStory` *us consists of* `Task` *tsk* (R10).

1 1. Sets (Entities)

- $\mathcal{P}$: Projects (`Project`)
- $\mathcal{T}$: Teams (`Team`)
- $\mathcal{W}$: Workers (`Worker`)
- $\mathcal{F}$: Features (`Feature`)
- $\mathcal{S}$: Skills (`Skill`)
- $\mathcal{R}$: Roles (`Role`)
- $\mathcal{PO}$: Product Owners (`ProductOwner`)
- $\mathcal{SM}$: Scrum Masters (`ScrumMaster`)
- $\mathcal{PB}$: Product Backlogs (`ProductBacklog`)
- $\mathcal{SP}$: Sprints (`Sprint`)
- $\mathcal{SPP}$: Sprint Plannings (`SprintPlanning`)
- $\mathcal{DS}$: Daily Scrums (`DailyScrum`)
- $\mathcal{SR}$: Sprint Reviews (`SprintReview`)

- $\mathcal{SRE}$: Sprint Retrospectives (`SprintRetrospective`)
- $\mathcal{SBL}$: Sprint Backlogs (`SprintBacklog`)
- $\mathcal{SG}$: Sprint Goals (`SprintGoal`)
- $\mathcal{E}$: Epics (`Epic`)
- $\mathcal{US}$: User Stories (`UserStory`)
- $\mathcal{TSK}$: Tasks (`Task`)
- $\mathcal{DEV}$: Development Snapshots (`DevelopmentSnapshot`)
- $\mathcal{BL}$: Blockers (`Blocker`)
- $\mathcal{SH}$: Stakeholders (`Stakeholder`)
- $\mathcal{VEL}$: Velocities (`Velocity`)
- $\mathcal{REP}$: Release Plans (`ReleasePlan`)
- $\mathcal{RM}$: Roadmaps (`Roadmap`)
- $\mathcal{SCB}$: Scrum Boards (`ScrumBoard`)
- $\mathcal{FED}$: Feature Documentations (`FeatureDocumentation`)

Relationship incidence sets (from `Relationships.csv`). We introduce:

- $\mathcal{E}^{PB\text{-}F} \subseteq \mathcal{PB} \times \mathcal{F}$ for R7 (*contains_feature*)
- $\mathcal{E}^{PB\text{-}E} \subseteq \mathcal{PB} \times \mathcal{E}$ for R8 (*contains_epic*)
- $\mathcal{E}^{E\text{-}US} \subseteq \mathcal{E} \times \mathcal{US}$ for R9 (*contains_user_story*)
- $\mathcal{E}^{US\text{-}TSK} \subseteq \mathcal{US} \times \mathcal{TSK}$ for R10 (*consists_of_tasks*)
- $\mathcal{E}^{SBL\text{-}SP} \subseteq \mathcal{SBL} \times \mathcal{SP}$ for R12 (*belongs_to_sprint*)
- $\mathcal{E}^{SP\text{-}SG} \subseteq \mathcal{SP} \times \mathcal{SG}$ for R13 (*pursues_goal*)
- $\mathcal{E}^{SCB\text{-}TSK} \subseteq \mathcal{SCB} \times \mathcal{TSK}$ for R14 (*contains_tasks*)
- $\mathcal{E}^{FED\text{-}F} \subseteq \mathcal{FED} \times \mathcal{F}$ for R15 (*documents_feature*)
- $\mathcal{E}^{TSK\text{-}BL} \subseteq \mathcal{TSK} \times \mathcal{BL}$ for R16 (*is_blocked_by*)
- $\mathcal{E}^{VEL\text{-}T} \subseteq \mathcal{VEL} \times \mathcal{T}$ for R19 (*refers_to_team*)
- $\mathcal{E}^{REP\text{-}F} \subseteq \mathcal{REP} \times \mathcal{F}$ for R20 (*plans_release*)
- $\mathcal{E}^{REP\text{-}RM} \subseteq \mathcal{REP} \times \mathcal{RM}$ for R21 (*is_part_of_roadmap*)
- $\mathcal{E}^{SP\text{-}DEV} \subseteq \mathcal{SP} \times \mathcal{DEV}$ for R22 (*generates_snapshot*)

(Other relationships may be handled analogously as metadata or fixed associations.)

2 2. Indices

- $p \in \mathcal{P}$, $t \in \mathcal{T}$, $w \in \mathcal{W}$, $f \in \mathcal{F}$, $s \in \mathcal{S}$, $r \in \mathcal{R}$,
- $po \in \mathcal{PO}$, $sm \in \mathcal{SM}$, $pb \in \mathcal{PB}$, $sp \in \mathcal{SP}$, $spp \in \mathcal{SPP}$, $ds \in \mathcal{DS}$,
- $sr \in \mathcal{SR}$, $sre \in \mathcal{SRE}$, $sbl \in \mathcal{SBL}$, $sg \in \mathcal{SG}$, $e \in \mathcal{E}$,
- $us \in \mathcal{US}$, $tsk \in \mathcal{TSK}$, $dev \in \mathcal{DEV}$, $bl \in \mathcal{BL}$, $sh \in \mathcal{SH}$,
- $vel \in \mathcal{VEL}$, $rep \in \mathcal{REP}$, $rm \in \mathcal{RM}$, $scb \in \mathcal{SCB}$, $fed \in \mathcal{FED}$.

3 3. Goals

We scalarize goals with weights $w_i^{(G)}$ and signs from `GoalType`:

$$\max\ Z \;=\; \sum_{\text{max goals } i} w_i^{(G)}\, T_i \;-\; \sum_{\text{min goals } j} w_j^{(G)}\, T_j \quad \text{(+ soft-condition preferences; see Section 4).}$$

Each item lists its ID/Name and the term $T_\cdot$ it contributes.

- **[G0] maximize_team_availability** (IsSum=True, max, `Worker.availability`, weight = 1.0)
  $T_{G0} = \sum_{w \in \mathcal{W}} a_w.$
- **[G1] minimize_total_project_budget** (IsSum=True, min, `Project.budget`, weight = 1.0)
  Decision uses allocated budget B_p; limit by b_p (see constraints).
  $T_{G1} = \sum_{p \in \mathcal{P}} B_p.$
- **[G2] maximize_story_points_delivered** (IsSum=True, max, `UserStory.story_points`, weight = 1.2)
  $T_{G2} = \sum_{us \in \mathcal{US}} \mathrm{sp}_{us}\, x_{us}.$
- **[G3] minimize_task_effort** (IsSum=True, min, `Task.effort`, weight = 1.0)
  $T_{G3} = \sum_{tsk \in \mathcal{TSK}} \mathrm{ef}_{tsk}\, x_{tsk}.$
- **[G4] maximize_feature_priority** (IsSum=True, max, `Feature.priority`, weight = 0.8)
  $T_{G4} = \sum_{f \in \mathcal{F}} \mathrm{pr}_f\, x_f.$
- **[G5] minimize_number_of_tasks_in_sprint_backlog** (IsSum=True, min, `SprintBacklog.number_of_tas`
  weight = 0.7)
  $T_{G5} = \sum_{sbl \in \mathcal{SBL}} \mathrm{nt}_{sbl}.$
- **[G6] maximize_sprint_goal_achievement** (IsSum=True, max, `Sprint.achievement_of_goal`, weight = 1.5)
  Uses user-set scalar weight ω on goal attainment if desired.
  $T_{G6} = \sum_{sp \in \mathcal{SP}} \omega\, \mathrm{ag}_{sp}.$

- **[G7] maximize_team_satisfaction** (IsSum=True, max, `SprintRetrospective.team_satisfaction`, weight = 1.3)
  $T_{G7} = \sum_{sre \in \mathcal{SRE}} \mathrm{ts}_{sre}.$

- **[G8] minimize_blocker_severity** (IsSum=True, min, `Blocker.severity`, weight = 1.1)
  $T_{G8} = \sum_{bl \in \mathcal{BL}} \mathrm{sev}_{bl}.$

- **[G9] maximize_average_velocity** (IsSum=True, max, `Velocity.avg._story_points`, weight = 1.4)
  $T_{G9} = \sum_{vel \in \mathcal{VEL}} \overline{\mathrm{sp}}_{vel}.$

- **[G10] maximize_velocity_trend** (IsSum=True, max, `Velocity.trend`, weight = 0.9)
  $T_{G10} = \sum_{vel \in \mathcal{VEL}} \mathrm{tr}_{vel}.$

- **[G11] maximize_review_attendance** (IsSum=True, max, `SprintReview.attendees_count`, weight = 0.5)
  $T_{G11} = \sum_{sr \in \mathcal{SR}} \mathrm{att}_{sr}.$

4 4. Conditions

Hard *Must-Match* (CriteriaType = 2) constraints are enforced strictly. *Cannot-Match* (CriteriaType = 0) exclude items. *May-Match* (CriteriaType = 1) are modeled as soft preferences via bonus terms P_k added to Z with their weights.

- **[C0] must_have_certified_skills** (Must-Match, `Skill.certified`, weight = 1.0)
  Logical: only certified skills are considered in any skill-dependent assignment.
  Math: define $\mathcal{S}^{cert} = \{s \in \mathcal{S} \mid \texttt{certified}(s) = 1\}$ and restrict any skill-related sums to $\mathcal{S}^{cert}$.

- **[C1] prefer_high_feature_status** (May-Match, `Feature.status`, weight = 0.5)
  Logical: prefer features with advanced/ready status.
  Math (soft): $P_{C1} = \sum_{f \in \mathcal{F}} \mathbf{1}\{\texttt{status}(f) \in \text{Ready/Approved}\}\, x_f.$

- **[C2] exclude_blocked_tasks** (Cannot-Match, `Task.status`, weight = 1.0)
  Logical: tasks that are currently blocked are excluded.
  Math: $x_{tsk} = 0 \quad \forall\, tsk \in \mathcal{TSK}$ with $\texttt{status}(tsk) = \text{Blocked}.$

- **[C3] must_have_team_status_active** (Must-Match, `Team.team_status`, weight = 1.0)
  Logical: only active teams can receive workers.
  Math: $\sum_{w \in \mathcal{W}} y_{w,t} = 0 \quad \forall\, t \in \mathcal{T}$ with $\texttt{team\_status}(t) \neq \text{Active}.$

- **[C4] prefer_product_backlog_status_ready** (May-Match, `ProductBacklog.status`, weight = 0.6)
  Logical: prefer PBs in "Ready" state.
  Math (soft): $P_{C4} = \sum_{pb \in \mathcal{PB}} \mathbf{1}\{\texttt{status}(pb) = \text{Ready}\}.$

- **[C5] must_have_worker_status_active** (Must-Match, `Worker.status`, weight = 1.0)
  Logical: only active workers can be assigned.
  Math: $\sum_{t \in \mathcal{T}} y_{w,t} = 0 \quad \forall\, w \in \mathcal{W}$ with $\texttt{status}(w) \neq \text{Active}.$

- **[C6] prefer_team_type_cross_functional** (May-Match, `Team.team_type`, weight = 0.4)
  Logical: prefer cross-functional teams.
  Math (soft): $P_{C6} = \sum_{t \in \mathcal{T}} \mathbf{1}\{\texttt{team\_type}(t) = \text{CrossFunctional}\} \cdot \min\{1, \sum_w y_{w,t}\}$.

- **[C7] must_have_sprint_status_active** (Must-Match, `Sprint.status`, weight = 1.0)
  Logical: only active sprints are counted in sprint-related calculations.
  Math: restrict sums involving $\mathcal{SP}$ to $\mathcal{SP}^{act} = \{sp \in \mathcal{SP} \mid \texttt{status}(sp) = \text{Active}\}$.

- **[C8] exclude_low_priority_user_stories** (Cannot-Match, `UserStory.priority`, weight = 1.0)
  Logical: very low priority stories are excluded.
  Math: $x_{us} = 0 \quad \forall\, us \in \mathcal{US}$ with $\texttt{priority}(us) < \theta$.

- **[C9] prefer_high_benefit_sprint_goal** (May-Match, `SprintGoal.benefit`, weight = 0.7)
  Logical: prefer sprints whose goals promise higher benefit.
  Math (soft): $P_{C9} = \sum_{(sp,sg) \in \mathcal{E}^{SP\text{-}SG}} \texttt{benefit}(sg)$.

- **[C10] must_have_skill_category_required** (Must-Match, `Skill.category`, weight = 1.0)
  Logical: only skills from required categories are eligible.
  Math: restrict skill sums to $\mathcal{S}^{req} = \{s \in \mathcal{S} \mid \texttt{category}(s) \in \text{RequiredSet}\}$.

- **[C11] prefer_location_remote** (May-Match, `Team.location`, weight = 0.3)
  Logical: prefer teams that operate remotely, if allowed.
  Math (soft & gated): $P_{C11} = \rho \cdot \sum_{t \in \mathcal{T}} \mathbf{1}\{\texttt{location}(t) = \text{Remote}\} \cdot \min\{1, \sum_w y_{w,t}\}$.

Soft-condition aggregation. Let $w_k^{(C)}$ be the weight of a May-Match condition. We augment the objective:

$$Z' = Z + \sum_{\text{May-Match } k} w_k^{(C)} P_k.$$

5 5. DecisionVariables

- **Binary selections**
  - $x_f \in \{0,1\}$: (DV0) `select_feature` — feature f selected for implementation.
  - $x_{us} \in \{0,1\}$: (DV1) `select_user_story` — user story us included in the sprint.
  - $x_{tsk} \in \{0,1\}$: (DV2) `select_task` — task tsk included in the sprint plan.
  - $y_{w,t} \in \{0,1\}$: (DV3) `assign_worker_to_team` — worker w assigned to team t.
  - $\rho \in \{0,1\}$: (DV10) `allow_remote_team`.

- **Integer/scalar choices**
  - $L \in \{1,2,3,4\}$: (DV4) `sprint_length_weeks`.
  - $B_p \in [0, 10{,}000{,}000]$: (DV5) `allocate_project_budget` for each $p \in \mathcal{P}$.
  - $N_t \in \{1,\ldots,50\}$: (DV6) `team_size` for each $t \in \mathcal{T}$.
  - $K \in \{1,\ldots,100\}$: (DV7) `max_parallel_tasks`.
  - $V \in [0,200]$: (DV8) `velocity_target`.
  - $\theta \in \{1,2,3,4,5\}$: (DV9) `priority_threshold`.
  - $\omega \in [0,10]$: (DV11) `sprint_goal_weight`.

Core structural constraints (logic from relationships/attributes).

$$\sum_{w\in\mathcal{W}} y_{w,t} \;=\; N_t \qquad \forall\, t\in\mathcal{T} \qquad \text{(team size equals a} \tag{1}$$

$$\sum_{tsk\in\mathcal{TSK}} x_{tsk} \;\le\; K \qquad \text{(cap concurrent/planned tasks; DV7)} \tag{2}$$

$$B_p \;\le\; b_p \qquad \forall\, p\in\mathcal{P} \qquad \text{(allocated budget v} \tag{3}$$

$$\sum_{us\in\mathcal{US}} \mathrm{sp}_{us}\, x_{us} \;\ge\; V \qquad \text{(meet velocity target)} \tag{4}$$

$$x_{us} \;\le\; \mathbf{1}\{\texttt{priority}(us)\ge\theta\} \qquad \forall\, us\in\mathcal{US} \qquad \text{(priority threshold;} \tag{5}$$

$$x_f \;\le\; \mathbf{1}\{\texttt{priority}(f)\ge\theta\} \qquad \forall\, f\in\mathcal{F} \qquad \text{(priority threshold} \tag{6}$$

$$x_{tsk} \;\le \sum_{us:(us,tsk)\in\mathcal{E}^{US\text{-}TSK}} x_{us} \qquad \forall\, tsk\in\mathcal{TSK} \qquad \text{(task only if its use} \tag{7}$$

$$\sum_{t\in\mathcal{T}} y_{w,t} \;\le\; 1 \qquad \forall\, w\in\mathcal{W} \qquad \text{(worker to at most} \tag{8}$$

$$y_{w,t} = 0 \qquad \forall\,(w,t) : \texttt{team\_status}(t)\ne \text{Active or } \texttt{status}(w)\ne\text{Active} \qquad \text{(C3 \& C5 hard gat} \tag{9}$$

$$x_{tsk} = 0 \qquad \forall\, tsk : \texttt{status}(tsk) = \text{Blocked} \qquad \text{(C2 gating)} \tag{10}$$

Final scalarized objective (including soft preferences). Let $\mathcal{G}^{max} = \{G0, G2, G4, G6, G7, G9, G10, G11\}$, $\mathcal{G}^{min} = \{G1, G3, G5, G8\}$ and T_{Gi} as above. Let $\mathcal{C}^{soft} = \{C1, C4, C6, C9, C11\}$ with preference terms P_{Ck}. Weights come from the CSVs. The model solves:

$$\boxed{\max\; Z'' \;=\; \sum_{i\in\mathcal{G}^{max}} w_i^{(G)}\, T_{Gi} \;-\; \sum_{j\in\mathcal{G}^{min}} w_j^{(G)}\, T_{Gj} \;+\; \sum_{k\in\mathcal{C}^{soft}} w_k^{(C)}\, P_{Ck}}$$

subject to (1)–(10), domain bounds of Section 5, and entity/relationship gating described in Section 4.