SCRUM Domain Optimization Model

TruelyMostWanted

August 11, 2025

Contents

1 1. Sets (Entities)

- $\mathcal{P}$ (Projects, index p): product or initiative; attributes {id, name, project_start, project_end, description, buc
- $\mathcal{T}$ (Teams, index t): cross-functional team; attributes {id, name, team_size, team_start, team_status, location
- $\mathcal{W}$ (Workers, index w): individual members; attributes {id, name, first_name, email, start_date, status, availa
- $\mathcal{F}$ (Features, index f): mid-sized functionality; attributes {id, title, description, status, priority, estimated_ef
- $\mathcal{S}$ (Skills, index s): competences; attributes {id, label, description, level, certified, category}.
- $\mathcal{R}$ (Roles, index r): responsibilities; attributes {id, role_name, description, area_of_responsibility}.
- $\mathcal{PO}$ (Product Owners, index po): attributes {id, name, email, availability}.
- $\mathcal{SM}$ (Scrum Masters, index sm): attributes {id, name, email, experience}.
- $\mathcal{PB}$ (Product Backlogs, index pb): attributes {id, created_on, last_updated, number_of_entries, status}.
- $\mathcal{SP}$ (Sprints, index sp): attributes {id, sprint_number, start_date, end_date, status, achievement_of_goal}.
- $\mathcal{SPP}$ (Sprint Plannings, index spp): attributes {id, date, duration_(min), moderation, outcome_documentat
- $\mathcal{DS}$ (Daily Scrums, index ds): attributes {id, date, time, duration, moderation}.
- $\mathcal{SR}$ (Sprint Reviews, index sr): attributes {id, date, duration, feedback_documentation, attendees_count}.
- $\mathcal{SRE}$ (Sprint Retrospectives, index sre): attributes {id, date, duration, improvement_actions, team_satisfact
- $\mathcal{SBL}$ (Sprint Backlogs, index sbl): attributes {id, number_of_tasks, last_updated, status, total_effort}.
- $\mathcal{SG}$ (Sprint Goals, index sg): attributes {id, objective_description, achievement_status, benefit}.
- $\mathcal{E}$ (Epics, index e): attributes {id, title, description, priority, status, estimated_effort}.
- $\mathcal{US}$ (User Stories, index us): attributes {id, title, description, acceptance_criteria, priority, story_points, statu
- $\mathcal{TSK}$ (Tasks, index tsk): attributes {id, title, description, status, effort, type}.
- $\mathcal{DEV}$ (Development Snapshots, index dev): attributes {id, version_number, creation_date, test_status, deploy
- $\mathcal{BL}$ (Blockers, index bl): attributes {id, title, description, severity, status, detected_on, resolved_on}.
- $\mathcal{SH}$ (Stakeholders, index sh): attributes {id, name, organization, role, email, area_of_interest, influence_level,
- $\mathcal{VEL}$ (Velocities, index vel): attributes {id, number_of_sprints_used, avg._story_points, max_velocity, min_vel
- $\mathcal{REP}$ (Release Plans, index rep): attributes {id, version, planned_date, included_features, status}.
- $\mathcal{RM}$ (Roadmaps, index rm): attributes {id, start_date, end_date, milestones, objectives, versions}.
- $\mathcal{SCB}$ (Scrum Boards, index scb): attributes {id, board_type, columns, number_of_cards, last_updated}.
- $\mathcal{FED}$ (Feature Documentations, index fed): attributes {id, title, description, creation_date, change_log, linke

Relationship-derived index sets (from Relationships.csv). We use the following incidence sets:

- TeamProj(t) $\subseteq \mathcal{P}$ from R1 (team t assigned to project(s)).
- TeamOf(w) $\subseteq \mathcal{T}$ from R2 (worker w belongs to team(s)).
- Skills(w) $\subseteq \mathcal{S}$ from R3; Roles(w) $\subseteq \mathcal{R}$ from R4.
- POManages(po) $\in \mathcal{PB}$ from R5; SMOf(t) $\in \mathcal{SM}$ from R6.
- Features(pb) $\subseteq \mathcal{F}$ (R7); Epics(pb) $\subseteq \mathcal{E}$ (R8).
- Stories(e) $\subseteq \mathcal{US}$ (R9); Tasks(us) $\subseteq \mathcal{TSK}$ (R10).
- SBStories(sbl) $\subseteq \mathcal{US}$ (R11); SprintOf(sbl) $\in \mathcal{SP}$ (R12).
- GoalOf(sp) $\in \mathcal{SG}$ (R13); BoardTasks(scb) $\subseteq \mathcal{TSK}$ (R14).
- DocOf(fed) $\in \mathcal{F}$ (R15); Blocks(tsk) $\subseteq \mathcal{BL}$ (R16).
- StakeholdersIn(sr) $\subseteq \mathcal{SH}$ (R17); Moderates(sm) $\subseteq \mathcal{SRE}$ (R18).
- VelOf(vel) $\in \mathcal{T}$ (R19); FeaturesIn(rep) $\subseteq \mathcal{F}$ (R20).
- RoadmapOf(rep) $\in \mathcal{RM}$ (R21); SnapshotOf(sp) $\in \mathcal{DEV}$ (R22).

2 2. Indices

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

Parameters (derived from attributes; all are numeric or suitably encoded).

- B_p (budget of project p); A_w (availability of worker w); SP_{us} (story points of user story us); Prio_f (priority of feature f).
- Effort_{tsk}; VelAvg_{vel}; Rel_{sh} (stakeholder relevance); $\mathrm{AchGoal}_{sp} \in \{0, 1\}$.
- $\mathrm{PlanDur}_{spp}$ (planning minutes); Att_{sr} (review attendees); WIP_{scb} (cards on board).
- $\mathrm{FreshPB}_{pb}$ (freshness score from `last_updated`); Sev_{bl} (blocker severity).
- Status/flags as binary parameters: e.g., $\mathrm{ActiveProj}_p$, $\mathrm{Blocked}_{tsk}$, $\mathrm{Certified}_s$, $\mathrm{POAvail}_{po}$, etc.
- Thresholds and bounds (given/decided by policy): $\underline{A}$ (min availability), $\underline{\mathrm{Prio}}$ (min priority), $\overline{SBL}$ (sprint capacity), $\overline{\mathrm{Sev}}$ (severity limit), allowed team size $[\underline{N}, \overline{N}]$.
- Incidence masks from relationships, e.g., $\mathbf{1}\{us \in \mathrm{SBStories}(sbl)\}$, $\mathbf{1}\{bl \in \mathrm{Blocks}(tsk)\}$, etc.

3 3. Goals

We scalarize multiple goals via a single composite objective

$$\min Z \;=\; \sum_{g \in \mathcal{G}} \omega_g \, \phi_g,$$

where each goal g contributes ϕ_g with weight ω_g. For a *maximize_·* goal we negate the measure (so it becomes minimizing the negative utility). When `IsSum=True`, we sum over the corresponding entity set; when `IsSum=False`, we use the average over that set (or the single instance if unique).

- **[G0] minimize_total_budget** (*min*, weight 1.0): logical: reduce overall spend.
  $\phi_{G0} = \sum_{p \in \mathcal{P}} B_p.$
- **[G1] maximize_worker_availability** (*max*, weight 1.0): logical: increase available capacity.
  $\phi_{G1} = -\sum_{w \in \mathcal{W}} A_w.$
- **[G2] maximize_story_throughput** (*max*, weight 1.0): logical: maximize delivered story points.
  $\phi_{G2} = -\sum_{us \in \mathcal{US}} \mathrm{SP}_{us}.$
- **[G3] minimize_open_blocker_severity** (*min*, weight 1.0): logical: lower active blocker burden.
  $\phi_{G3} = \sum_{bl \in \mathcal{BL}} \mathrm{Sev}_{bl} \cdot \mathbf{1}\{\mathrm{Resolved}_{bl} = 0\}.$
- **[G4] maximize_feature_priority** (*max*, weight 1.2): logical: favor high-priority features.
  $\phi_{G4} = -\sum_{f \in \mathcal{F}} \mathrm{Prio}_f.$
- **[G5] minimize_total_task_effort** (*min*, weight 1.0): logical: reduce total effort.
  $\phi_{G5} = \sum_{tsk \in \mathcal{TSK}} \mathrm{Effort}_{tsk}.$
- **[G6] maximize_team_velocity** (*max*, weight 1.3): logical: increase average velocity.
  $\phi_{G6} = -\sum_{vel \in \mathcal{VEL}} \mathrm{VelAvg}_{vel}.$
- **[G7] maximize_stakeholder_relevance** (*max*, weight 0.8): logical: increase value for most relevant stakeholders.
  $\phi_{G7} = -\sum_{sh \in \mathcal{SH}} \mathrm{Rel}_{sh}.$
- **[G8] maximize_sprint_goal_achievement** (*max*, weight 1.5): logical: maximize achieved sprint goals.
  $\phi_{G8} = -\sum_{sp \in \mathcal{SP}} \mathrm{AchGoal}_{sp}.$
- **[G9] minimize_meeting_time** (*min*, weight 0.6): logical: reduce sprint planning duration.
  $\phi_{G9} = \sum_{spp \in \mathcal{SPP}} \mathrm{PlanDur}_{spp}.$

- **[G10] maximize_review_attendance** (*max*, weight 0.7): logical: increase review attendees.
  $\phi_{G10} = -\sum_{sr \in \mathcal{SR}} \text{Att}_{sr}$.
- **[G11] minimize_wip_on_board** (*min*, weight 1.1): logical: limit cards on boards.
  $\phi_{G11} = \sum_{scb \in \mathcal{SCB}} WIP_{scb}$.
- **[G12] maximize_backlog_freshness** (*max*, weight 0.5, `IsSum=False`): logical: keep PB updated.
  $\phi_{G12} = -\frac{1}{|\mathcal{PB}|} \sum_{pb \in \mathcal{PB}} \text{FreshPB}_{pb}$.

4 4. Conditions

Each condition encodes a logical rule and a constraint. We reference decision variables from Section 5 and relationship sets from Section 1.

- **[C0] must_match_active_projects** (`Must-Match`): only active projects considered.
  Logical: $p \in \mathcal{P} \Rightarrow \text{ActiveProj}_p = 1$. Math: $\sum_{us \in \mathcal{US}} \sum_{sp \in \mathcal{SP}} x^{\text{US-SP}}_{us,sp,p} \leq M \cdot \text{ActiveProj}_p$, $\forall p$ (big-M gating by project activity).
- **[C1] cannot_match_blocked_tasks** (`Cannot-Match`): exclude blocked tasks unless resolved.
  $x^{\text{TSK-W}}_{tsk,w} \leq \min_{bl \in \text{Blocks}(tsk)} z^{\text{BL-Resolved}}_{bl}$, $\forall tsk, w$, where $z^{\text{BL-Resolved}}_{bl} \in \{0,1\}$ indicates resolution; if any blocker unresolved, assignment is 0.
- **[C2] must_match_available_workers** (`Must-Match`): assign only to available workers.
  $\sum_{tsk \in \mathcal{TSK}} \text{Effort}_{tsk}\, x^{\text{TSK-W}}_{tsk,w} \leq A_w$, $\forall w$.
- **[C3] must_match_certified_skills** (`Must-Match`): when a task requires certified skill s, assign to worker with $\text{Certified}_s = 1$.
  $x^{\text{TSK-W}}_{tsk,w} \leq \sum_{s \in \text{Skills}(w)} \text{Certified}_s \cdot \mathbf{1}\{s \text{ required by } tsk\}$, $\forall tsk, w$.
- **[C4] must_match_product_owner_availability** (`Must-Match`): Sprint pulling from pb needs available po with $\text{POManages}(po) = pb$.
  $\sum_{us \in \text{SBStories}(sbl)} x^{\text{US-SP}}_{us,sp} \leq M \cdot \text{POAvail}_{po}$, $\forall sbl : \text{SprintOf}(sbl) = sp$, $po : \text{POManages}(po) = pb$.
- **[C5] must_match_sprint_has_goal** (`Must-Match`): each sprint must have a goal.
  $\text{AchGoal}_{sp} \leq \mathbf{1}\{\text{GoalOf}(sp) \text{ exists}\}$, $\forall sp$.
- **[C6] cannot_match_low_priority_features** (`Cannot-Match`): features with $\text{Prio}_f < \underline{\text{Prio}}$ cannot be planned.
  $x^{\text{F-REP}}_{f,rep} \leq \mathbf{1}\{\text{Prio}_f \geq \underline{\text{Prio}}\}$, $\forall f, rep$.
- **[C7] must_match_active_backlog** (`Must-Match`): pull only from active product backlogs.
  $\sum_{f \in \text{Features}(pb)} x^{\text{F-REP}}_{f,rep} \leq M \cdot \mathbf{1}\{\text{Status}_{pb} = \text{active}\}$, $\forall pb, rep$.

- **[C8] must_match_sprint_backlog_effort_limit** (`Must-Match`): total effort in a sprint within capacity.
$\sum_{us \in \mathcal{US}} \sum_{tsk \in \text{Tasks}(us)} \text{Effort}_{tsk}\, x^{\text{US-SP}}_{us,sp} \le \overline{SBL}_{sp},\ \forall sp.$

- **[C9] cannot_match_severe_blockers** (`Cannot-Match`): items with high severity blocked until severity $\le \overline{\text{Sev}}$.
$x^{\text{US-SP}}_{us,sp} \le \prod_{tsk \in \text{Tasks}(us)} \prod_{bl \in \text{Blocks}(tsk)} \mathbf{1}\{\text{Sev}_{bl} \le \overline{\text{Sev}} \ \vee \ z^{\text{BL-Resolved}}_{bl} = 1\}.$

- **[C10] must_match_review_feedback_documented** (`Must-Match`): reviews must have feedback documented.
$\mathbf{1}\{\text{feedback\_documentation}_{sr} \text{ present}\} = 1,\ \forall sr.$

- **[C11] must_match_retro_actions_logged** (`Must-Match`): retros capture improvements.
$\mathbf{1}\{\text{improvement\_actions}_{sre} \text{ present}\} = 1,\ \forall sre.$

- **[C12] must_match_team_size_bounds** (`Must-Match`): team size within bounds.
$\underline{N} \le N^{\text{Team}}_{t} \le \overline{N},\ \forall t.$

5 5. DecisionVariables

Notation below expands the CSV-defined variables into indexed decision variables over the relevant sets (domains as in the CSV).

- **[DV0] assign_user_story_to_sprint** ($x^{\text{US-SP}}_{us,sp} \in \{0,1\}$): assign story us to sprint sp.

- **[DV1] select_feature_for_release** ($x^{\text{F-REP}}_{f,rep} \in \{0,1\}$): include feature f in release plan rep.

- **[DV2] allocate_task_to_worker** ($x^{\text{TSK-W}}_{tsk,w} \in \{0,1\}$): assign task tsk to worker w.

- **[DV3] set_sprint_number** ($n^{\text{SP}}_{sp} \in \{1,\dots,999\}$): sprint numbering.

- **[DV4] set_story_points_for_user_story** ($y^{\text{SPts}}_{us} \in \{0,\dots,100\}$): calibrated story points (can refine SP_{us}).

- **[DV5] set_task_effort_hours** ($y^{\text{Eff}}_{tsk} \in \{0,\dots,100\}$): estimated effort (can refine Effort_{tsk}).

- **[DV6] choose_board_column** ($c^{\text{Board}}_{tsk,scb} \in \{0,1,2\}$): column index for task on board scb.

- **[DV7] schedule_sprint_start_offset** ($d^{\text{Start}}_{sp} \in \{0,\dots,30\}$): day offset from project start.

- **[DV8] decide_blocker_resolution_flag** ($z^{\text{BL-Resolved}}_{bl} \in \{0,1\}$): whether blocker bl is treated as resolved for planning.

- **[DV9] set_team_size** ($N^{\text{Team}}_{t} \in \{1,\dots,20\}$): effective team size for execution horizon.

- **[DV10] set_velocity_target_story_points** ($v^{\text{Target}}_{t} \in \{0,\dots,200\}$): target team velocity.

- **[DV11] plan_review_attendees_target** ($a^{\text{Target}}_{sr} \in \{0,\dots,100\}$): target attendees for review sr.

- **[DV12] pick_roadmap_milestone_sequence** ($m^{\text{Seq}}_{rm} \in \{1,\dots,50\}$): ordering index for roadmap rm milestones.

Linking and consistency constraints (cross-cutting).

- Story-to-sprint implies its tasks are executable: $x^{\text{US-SP}}_{us,sp} \leq \frac{1}{|\text{Tasks}(us)|} \sum_{tsk \in \text{Tasks}(us)} \sum_{w \in \mathcal{W}} x^{\text{TSK-W}}_{tsk,w}, \ \forall us, sp.$
- Sprint capacity based on team availability: $\overline{SBL}_{sp} \leq \sum_{t \in \mathcal{T}} N^{\text{Team}}_t \cdot \bar{a} \cdot H$ (with nominal availability $\bar{a}$ and hours H).
- Velocity target consistency: $\sum_{us} \text{SP}_{us} \, x^{\text{US-SP}}_{us,sp} \leq \sum_t v^{\text{Target}}_t, \ \forall sp.$