SCRUM Domain Optimization Model

TruelyMostWanted

August 12, 2025

Contents

1 Sets (Entities)

- **P** (Projects): The product or initiative to be developed.
- **T** (Teams): Self-organized, cross-functional development teams.
- **W** (Workers): Individual team members working on the project.
- **F** (Features): Mid-sized functionality.
- **S** (Skills): Professional or social competences of workers.
- **R** (Roles): Defined responsibilities within the Scrum team.
- **PO** (Product Owners): Responsible for product vision and Product Backlog.
- **SM** (Scrum Masters): Support teams in applying Scrum.
- **PB** (Product Backlogs): Ordered list of all requirements.
- **SP** (Sprints): Fixed time periods for creating an increment.
- **SPP** (Sprint Plannings): Kick-off meetings for Sprint preparation.
- **DS** (Daily Scrums): Daily 15-minute team meetings.
- **SR** (Sprint Reviews): Presentation and acceptance of results.
- **SRE** (Sprint Retrospectives): Retrospectives for process improvement.
- **SBL** (Sprint Backlogs): Selected backlog items + implementation plan.
- **SG** (Sprint Goals): Objectives to be achieved within the sprint.
- **E** (Epics): Large requirements that can be split into stories.
- **US** (User Stories): Requirements from the perspective of a user.
- **TSK** (Tasks): Smallest work units within a sprint.
- **DEV** (Development Snapshots): Product at the end of a sprint.
- **BL** (Blockers): Obstacles hindering progress.
- **SH** (Stakeholders): Interested parties in the product.
- **VEL** (Velocities): Average amount of work per sprint (per team).
- **REP** (Release Plans): Plans for releasing specific features.
- **RM** (Roadmaps): Long-term planning across releases.
- **SCB** (Scrum Boards): Visual representations of sprint tasks.
- **FED** (Feature Documentations): Documentation for a specific feature.

2 Indices

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

Attribute Parameters (from Entities.csv). For any entity set $\mathbf{X}$ and attribute name a listed in `Entities.csv`, we use parameter notation a_x (or $a_{x,y}$ if relational) to refer to the data value on element(s) $x \in \mathbf{X}$. Examples (non-exhaustive): $team\_size_t$, $availability_w$, $story\_points_{us}$, $effort_{tsk}$, $avgSP_{vel}$ for `avg._story_points`, etc. Dates/status/labels act as parameters or logical flags as appropriate.

3 Goals

We aggregate multiple goals via a weighted sum; "min" goals appear with a negative sign. Let ω_g be the weight of goal g and $\phi_g(\cdot)$ its contribution. The scalarized objective is:

$$\max \sum_{g \in \mathcal{G}_{\max}} \omega_g \, \phi_g \; - \sum_{g \in \mathcal{G}_{\min}} \omega_g \, \phi_g.$$

Each goal below defines ϕ_g (sums apply over the referenced entity set when `IsSum=True`).

- **G0 maximize_team_velocity** (Velocity / `avg._story_points`; weight 1.0)
  Logical: Increase average story points delivered across velocity records.
  Mathematical: $\phi_{G0} = \sum_{vel \in \mathbf{VEL}} avgSP_{vel}$.

- **G1 maximize_peak_velocity** (Velocity / `max_velocity`; weight 0.8)
  Logical: Reward higher observed peak velocity.
  Mathematical: $\phi_{G1} = \max_{vel \in \mathbf{VEL}} max\_velocity_{vel}$.

- **G2 minimize_velocity_variation** (Velocity / `min_velocity`; weight 0.6)
  Logical: Reduce variability by shrinking the gap between peak and trough (proxy via raising minima).
  Mathematical (penalty): $\phi_{G2} = \sum_{vel \in \mathbf{VEL}} \left(\overline{M} - min\_velocity_{vel}\right)$, where $\overline{M}$ is a constant upper bound.

- **G3 minimize_open_blocker_severity** (Blocker / `severity`; weight 1.0)
  Logical: Reduce cumulative severity of open blockers. Let $open_{bl} \in \{0, 1\}$ flag open status.
  Mathematical: $\phi_{G3} = \sum_{bl \in \mathbf{BL}} open_{bl} \cdot severity_{bl}$.

- **G4 minimize_blocker_age** (Blocker / `detected_on`; weight 0.4)
  Logical: Minimize age of blockers. Let $age_{bl} := now - detected\_on_{bl}$.
  Mathematical: $\phi_{G4} = \sum_{bl \in \mathbf{BL}} age_{bl}$.

- **G5 maximize_sprint_goal_achievement** (Sprint / `achievement_of_goal`; weight 1.0)
  Logical: Increase achieved sprint goals (binary or rate).
  Mathematical: $\phi_{G5} = \sum_{sp \in \mathbf{SP}} achievement\_of\_goal_{sp}$.

- **G6 minimize_sprint_overrun** (Sprint / `end_date`; weight 0.3)
  Logical: Penalize late sprint end vs. planned end $planEnd_{sp}$.
  Mathematical: $\phi_{G6} = \sum_{sp \in \mathbf{SP}} \max\{0, \, end\_date_{sp} - planEnd_{sp}\}$.

- **G7 maximize_feature_throughput** (Feature / `status`; weight 0.7)
  Logical: Reward completed features; let $done_f \in \{0,1\}$.
  Mathematical: $\phi_{G7} = \sum_{f \in \mathbf{F}} done_f$.

- **G8 minimize_story_points_in_progress** (UserStory / `story_points`; weight 0.9)
  Logical: Reduce WIP story points; let $inProg_{us} \in \{0,1\}$.
  Mathematical: $\phi_{G8} = \sum_{us \in \mathbf{US}} inProg_{us} \cdot story\_points_{us}$.

- **G9 maximize_team_utilization** (Worker / `availability`; weight 0.9)
  Logical: Increase usable worker capacity.
  Mathematical: $\phi_{G9} = \sum_{w \in \mathbf{W}} availability_w$.

- **G10 minimize_task_effort** (Task / `effort`; weight 1.0)
  Logical: Reduce total implementation effort selected for sprints (uses DV $b_{tsk,sp}$).
  Mathematical: $\phi_{G10} = \sum_{sp \in \mathbf{SP}} \sum_{tsk \in \mathbf{TSK}} b_{tsk,sp} \cdot effort_{tsk}$.

- **G11 maximize_release_readiness** (ReleasePlan / `status`; weight 0.5)
  Logical: Reward release plans marked ready; let $ready_{rep} \in \{0,1\}$.
  Mathematical: $\phi_{G11} = \sum_{rep \in \mathbf{REP}} ready_{rep}$.

4 Conditions

Decision Variables (symbols). We use the following symbols (defined fully in Section 5):

$x_{w,t} \in \{0,1\}$	(worker w assigned to team t)
$y_{us,sp} \in \{0,1\}$	(user story us scheduled in sprint sp)
$z_f \in \{0,1\}$	(feature f selected for next release)
$r_f \in \mathbb{Z}_+$	(backlog rank of feature f)
$a_t \in \mathbb{Z}_+$	(story points capacity allocated to team t)
$m_{sm,t} \in \{0,1\}$	(Scrum Master sm supports team t)
$po_{po,pb} \in \{0,1\}$	(Product Owner po manages backlog pb)
$b_{tsk,sp} \in \{0,1\}$	(task tsk selected for sprint sp)
$g_{sp} \in \{0,1\}$	(sprint sp commits to its goal)
$u_{bl} \in \{0,1\}$	(blocker bl marked resolved)
$B_f \in \mathbb{R}_+$	(budget allocated to feature f)
$d_{sp} \in \mathbb{Z}_+$	(duration in days of sprint sp)

Operating Constraints (from `Conditions.csv`).

- **C0 min_team_size** (*must-match*). Ensure teams meet minimum size $\underline{n}^{\text{team}}$.
  $\sum_{w \in \mathbf{W}} x_{w,t} \geq \underline{n}^{\text{team}}, \quad \forall t \in \mathbf{T}$.

- **C1 max_team_size** (*must-match*). Keep teams below upper size bound $\overline{n}^{\text{team}}$.
  $\sum_{w \in \mathbf{W}} x_{w,t} \leq \overline{n}^{\text{team}}, \quad \forall t \in \mathbf{T}$.

- **C2 worker_must_be_available** (*must-match*). Assign only available workers ($availability_w \in [0,1]$).
  $\sum_{t \in \mathbf{T}} x_{w,t} \;\le\; \mathbf{1}\{availability_w > 0\}, \quad \forall w \in \mathbf{W}.$

- **C3 story_points_cap_per_sprint** (*must-match*). Limit total story points per sprint to cap $\overline{SP}_{sp}$.
  $\sum_{us \in \mathbf{US}} story\_points_{us}\, y_{us,sp} \;\le\; \overline{SP}_{sp}, \quad \forall sp \in \mathbf{SP}.$

- **C4 task_effort_cap_per_sprint** (*must-match*). Limit total task effort per sprint to cap $\overline{E}_{sp}$.
  $\sum_{tsk \in \mathbf{TSK}} effort_{tsk}\, b_{tsk,sp} \;\le\; \overline{E}_{sp}, \quad \forall sp \in \mathbf{SP}.$

- **C5 feature_priority_floor** (*may-match*). Enforce minimum priority level $\underline{\pi}$ for selected features.
  $r_f \;\le\; \overline{r} \cdot \mathbf{1}\{priority_f \ge \underline{\pi}\}, \quad \forall f \in \mathbf{F}.$

- **C6 epic_effort_ceiling** (*may-match*). Control epic estimated effort by ceiling $\overline{EE}$.
  $estimated\_effort_e \;\le\; \overline{EE}, \quad \forall e \in \mathbf{E}.$

- **C7 sprint_within_dates** (*must-match*). Sprints must fit the calendar window $[planStart_{sp}, planEnd_{sp}]$.
  $start\_date_{sp} \ge planStart_{sp}, \;\; end\_date_{sp} \le planEnd_{sp}, \quad \forall sp \in \mathbf{SP}.$

- **C8 sprint_end_before_release** (*must-match*). Sprint ends before planned release date $planned\_date_{rep}$.
  $end\_date_{sp} \le planned\_date_{rep}, \quad \forall sp \in \mathbf{SP}, \forall rep \in \mathbf{REP}$ linked to sp.

- **C9 backlog_status_active** (*must-match*). Only active backlogs considered ($status_{pb} =$ active).
  $\sum_{po \in \mathbf{PO}} po_{po,pb} \;\le\; \mathbf{1}\{status_{pb} = \text{active}\}, \quad \forall pb \in \mathbf{PB}.$

- **C10 blocker_status_must_be_open_for_penalty** (*must-match*). Only open blockers incur penalties; resolution switches them off.
  $open_{bl} \;\le\; 1 - u_{bl}, \quad \forall bl \in \mathbf{BL}.$

- **C11 skill_level_requirement** (*must-match*). Required skill level $\underline{\ell}_{f,s}$ when assigning workers to features needing skill s. Let $hasSkill_{w,s} \in \{0,1\}$.
  $hasSkill_{w,s} \ge \mathbf{1}\{level_{w,s} \ge \underline{\ell}_{f,s}\}$ whenever worker w is allocated to feature f.

Structural Constraints (from `Relationships.csv`).

- **R1 is_assigned_to_project** (Team → Project; 1:N). Each team belongs to exactly one project: $\sum_{p \in \mathbf{P}} \tau_{t,p} = 1, \; \forall t \in \mathbf{T}$; a project can have many teams ($\tau_{t,p} \in \{0,1\}$).

- **R2 belongs_to_team** (Worker → Team; N:1). Workers belong to exactly one team: $\sum_t x_{w,t} = 1, \; \forall w$.

- **R3 has_skill** (Worker ↔ Skill; N:M). Skill coverage: $\sigma_{w,s} \in \{0,1\}$ free; many-to-many allowed.

- **R4 takes_on_role** (Worker ↔ Role; N:M). Role assignment $\rho_{w,r} \in \{0,1\}$, many-to-many.

- **R5 manages_backlog** (PO $\leftrightarrow$ PB; 1:1). Unique manager: $\sum_{po} po_{po,pb} = 1,\ \sum_{pb} po_{po,pb} = 1$.
- **R6 is_supported_by** (Team $\leftrightarrow$ SM; 1:1). Unique Scrum Master per team: $\sum_{sm} m_{sm,t} = 1,\ \sum_{t} m_{sm,t} \leq 1$.
- **R7 contains_feature** (PB $\rightarrow$ Feature; 1:N). Assignment $\chi_{pb,f} \in \{0,1\}$ with $\sum_{pb} \chi_{pb,f} = 1$.
- **R8 contains_epic** (PB $\rightarrow$ Epic; 1:N). Assignment $\eta_{pb,e} \in \{0,1\}$ with $\sum_{pb} \eta_{pb,e} = 1$.
- **R9 contains_user_story** (Epic $\rightarrow$ UserStory; 1:N). Link $\gamma_{e,us} \in \{0,1\}$ with $\sum_{e} \gamma_{e,us} = 1$.
- **R10 consists_of_tasks** (UserStory $\rightarrow$ Task; 1:N). Link $\kappa_{us,tsk} \in \{0,1\}$ with $\sum_{us} \kappa_{us,tsk} = 1$.
- **R11 is_in_sprint_backlog** (UserStory $\leftrightarrow$ SprintBacklog; N:M). Scheduling $y_{us,sp}$ induces *sbl* inclusion.
- **R12 belongs_to_sprint** (SprintBacklog $\leftrightarrow$ Sprint; 1:1). Each *sbl* tied to exactly one *sp*.
- **R13 pursues_goal** (Sprint $\leftrightarrow$ SprintGoal; 1:1). Commitment $g_{sp} = 1 \Rightarrow$ exactly one goal.
- **R14 contains_tasks** (ScrumBoard $\rightarrow$ Task; 1:N). Board-task incidence $\beta_{scb,tsk} \in \{0,1\}$.
- **R15 documents_feature** (FeatureDocumentation $\leftrightarrow$ Feature; 1:1). Unique documentation per feature.
- **R16 is_blocked_by** (Task $\leftrightarrow$ Blocker; N:M). If $b_{tsk,sp} = 1$ and $\iota_{tsk,bl} = 1$ with $open_{bl} = 1$, task is blocked.
- **R17 participates_in** (Stakeholder $\leftrightarrow$ SprintReview; N:M). Attendance matrix $\psi_{sh,sr} \in \{0,1\}$.
- **R18 moderates_retrospective** (SM $\rightarrow$ SRE; 1:N). Moderation $\mu_{sm,sre} \in \{0,1\}$ with $\sum_{sm} \mu_{sm,sre} = 1$.
- **R19 refers_to_team** (Velocity $\rightarrow$ Team; 1:1). Map $a_{vel,t} \in \{0,1\}$ with $\sum_{t} a_{vel,t} = 1$.
- **R20 plans_release** (REP $\rightarrow$ Feature; 1:N). Release selection z_f implies inclusion in some *rep*: $\sum_{rep} \lambda_{rep,f} = z_f$.
- **R21 is_part_of_roadmap** (REP $\rightarrow$ RM; N:1). Each plan belongs to one roadmap: $\sum_{rm} \pi_{rep,rm} = 1$.
- **R22 generates_snapshot** (Sprint $\rightarrow$ DEV; 1:1). Each sprint generates exactly one snapshot: $\sum_{dev} \delta_{sp,dev} = 1$.

5 DecisionVariables

- **DV0 assign_worker_to_team** ($x_{w,t}$): binary, domain $\{0,1\}$, min 0, max 1. Worker w is assigned to team t.
- **DV1 assign_user_story_to_sprint** ($y_{us,sp}$): binary, domain $\{0,1\}$, min 0, max 1. User story *us* scheduled in sprint *sp*.

- **DV2 select_feature_for_release** (z_f): binary, domain $\{0,1\}$, min 0, max 1. Feature f included in a release plan.
- **DV3 prioritize_feature_rank** (r_f): integer, domain $[1,100]$, min 1, max 100. Rank position of feature f.
- **DV4 allocate_story_points_to_team** (a_t): integer, domain $[0,500]$, min 0, max 500. Capacity allocated to team t.
- **DV5 assign_scrum_master_to_team** ($m_{sm,t}$): binary, domain $\{0,1\}$, min 0, max 1. SM sm supports team t.
- **DV6 assign_product_owner_to_backlog** ($po_{po,pb}$): binary, domain $\{0,1\}$, min 0, max 1. PO po manages backlog pb.
- **DV7 select_task_for_sprint_backlog** ($b_{tsk,sp}$): binary, domain $\{0,1\}$, min 0, max 1. Task tsk included in sprint sp.
- **DV8 set_sprint_goal_commitment** (g_{sp}): binary, domain $\{0,1\}$, min 0, max 1. Sprint sp commits to a specific goal.
- **DV9 mark_blocker_resolved** (u_{bl}): binary, domain $\{0,1\}$, min 0, max 1. Blocker bl resolution decision.
- **DV10 allocate_budget_to_feature** (B_f): real, domain $[0,10^6]$, min 0, max 1,000,000. Budget amount for feature f.
- **DV11 plan_sprint_duration_days** (d_{sp}): integer, domain $[7,30]$, min 7, max 30. Number of days in sprint sp.

Capacity Coupling (illustrative). To link capacity, velocity and assignment:

$$\sum_{us \in \mathbf{US}} story\_points_{us}\, y_{us,sp} \;\le\; \sum_{t \in \mathbf{T}} a_t \cdot \alpha_{t,sp}, \quad \forall sp \in \mathbf{SP},$$

where $\alpha_{t,sp} \in \{0,1\}$ indicates team t works in sprint sp, and a_t may be calibrated using **VEL** via $a_t \le \sum_{vel} a_{vel,t} \cdot avgSP_{vel}$.