SCRUM Planning and Delivery Optimization Model

TruelyMostWanted

September 5, 2025

Contents

1 1. Sets (Entities)

- P (Project): set of projects (Entity `Project`).
- T (Team): set of teams (Entity `Team`).
- W (Worker): set of workers (Entity `Worker`).
- F (Feature): set of features (Entity `Feature`).
- S (Skill): set of skills (Entity `Skill`).
- R (Role): set of roles (Entity `Role`).
- O (ProductOwner): set of product owners.
- M (ScrumMaster): set of scrum masters.
- B (ProductBacklog): set of product backlogs.
- SP (Sprint): set of sprints.
- SPP (SprintPlanning): set of sprint planning events.
- DS (DailyScrum): set of daily scrums.
- SR (SprintReview): set of sprint reviews.
- SRE (SprintRetrospective): set of sprint retrospectives.
- SBL (SprintBacklog): set of sprint backlogs.
- G (SprintGoal): set of sprint goals.
- E (Epic): set of epics.
- U (UserStory): set of user stories.
- K (Task): set of tasks.
- D (DevelopmentSnapshot): set of development snapshots.
- BL (Blocker): set of blockers.
- H (Stakeholder): set of stakeholders.
- V (Velocity): set of velocity measurements.
- RP (ReleasePlan): set of release plans.
- RM (Roadmap): set of roadmaps.
- CB (ScrumBoard): set of scrum boards.
- FD (FeatureDocumentation): set of feature documentation records.

2 2. Indices

- $p \in P,\ t \in T,\ w \in W,\ f \in F,\ s \in S,\ r \in R,$
- $o \in O,\ m \in M,\ b \in B,\ sp \in SP,\ spp \in SPP,\ ds \in DS,$
- $sr \in SR,\ sre \in SRE,\ sbl \in SBL,\ g \in G,\ e \in E,\ u \in U,$
- $k \in K,\ d\_v \in D,\ bl \in BL,\ h \in H,\ v \in V,\ rp \in RP,$
- $rm \in RM,\ cb \in CB,\ fd \in FD.$

Relationship incidence (from Relationships.csv). We interpret *Employee* as *Worker* and *Sprint Review* as *SprintReview*. Let the following binary parameters encode relations:

$\mathsf{R1}[t,p] \in \{0,1\}$ Team t is assigned to Project p,
$\mathsf{R2}[w,t],\ \mathsf{R3}[w,s],\ \mathsf{R4}[w,r],$
$\mathsf{R5}[o,b],\ \mathsf{R6}[t,m],\ \mathsf{R7}[b,f],\ \mathsf{R8}[b,e],$
$\mathsf{R9}[e,u],\ \mathsf{R10}[u,k],\ \mathsf{R11}[u,sbl],\ \mathsf{R12}[sbl,sp],$
$\mathsf{R13}[sp,g],\ \mathsf{R14}[cb,k],\ \mathsf{R15}[fd,f],\ \mathsf{R16}[k,bl],$
$\mathsf{R17}[h,sr],\ \mathsf{R18}[m,sre],\ \mathsf{R19}[v,t],\ \mathsf{R20}[rp,f],$
$\mathsf{R21}[rp,rm],\ \mathsf{R22}[sp,d\_v].$

Attributes as parameters (from Entities.csv). For each entity, its attributes become data parameters. Examples (non-exhaustive):

Project budget: $\text{budget}[p] \geq 0$; Feature effort: $\text{eff}^F[f] \geq 0$; Epic effort: $\text{eff}^E[e] \geq 0$;
Task effort: $\text{eff}^K[k] \geq 0$; User Story points: $\text{sp}[u] \geq 0$;
Velocity average SP: $\overline{\text{SP}}[v] \geq 0$; Velocity trend: $\text{trend}[v] \in \mathbb{R}$;
Blocker severity: $\text{sev}[bl] \geq 0$; Review attendees: $\text{att}[sr] \geq 0$.

Costs and calendars. Let $c^F, c^E, c^K \geq 0$ be cost per effort unit for features, epics, and tasks; dates are converted to numeric time (e.g., days).

3 3. Goals

`points_completed` *Logic:* maximize delivered value measured in story points.
Math:

$$\max\ Z\_0 = \sum \_sp \in SP \sum \_u \in U \text{sp}[u]\ y\_u, sp$$

where $y\_u, sp = \text{DV0}$.

`goal_achievement` *Logic:* reward sprints achieving their goals.
Math:

$$\max\ Z\_1 = \sum \_sp \in SP g\_sp$$

where $g\_sp = \text{DV10}$ corresponds to Sprint.achievement_of_goal.

`blocker_severity` *Logic:* drive down severity of unresolved blockers.
Math:

$$\min\ Z\_2 = \sum \_bl \in BL \text{sev}[bl]\ (1 - r\_bl)$$

where r_bl = DV8 (1 if resolved).

`l_feature_effort` *Logic:* prefer lower-effort features.
Math:

$$\min\ Z\_3 = \sum \_f \in F \text{eff}^F[f]\ s\_f$$

where s_f = DV4.

`otal_task_effort` *Logic:* minimize workload planned in sprints.
Math:

$$\min\ Z\_4 = \sum \_sp \in SP \sum \_k \in K \text{eff}^K[k]\ z\_k, sp$$

where z_k, sp = DV1.

`e_velocity_trend` *Logic:* encourage improving velocity.
Math:

$$\max\ Z\_5 = \sum \_v \in V \text{trend}[v]$$

`iority_delivered` *Logic:* deliver higher-priority features.
Math:

$$\max\ Z\_6 = \sum \_f \in F \text{priority}[f]\ s\_f$$

`iority_delivered` *Logic:* deliver higher-priority epics.
Math:

$$\max\ Z\_7 = \sum \_e \in E \text{priority}[e]\ s\_e$$

where s_e = DV5.

`eview_attendance` *Logic:* increase stakeholder engagement.
Math:

$$\max\ Z\_8 = \sum \_sr \in SR \text{att}[sr]$$

`e_budget_overrun` *Logic:* keep planned cost within budgets. Let Spend $= \sum \_f c^F \text{eff}^F[f] s\_f + \sum \_e c^E \text{eff}^E[e] s\_e + \sum \_sp, k c^K \text{eff}^K[k] z\_k, sp$.
Math:

$$\min\ Z\_9 = \max\left\{0,\ \text{Spend} - \sum \_p \in P \text{budget}[p]\right\}$$

(piecewise-linear; can be linearized with an overrun slack variable).

`cklog_throughput` *Logic:* favor sprints processing more tasks.
Math:

$$\max\ Z\_10 = \sum \_sp \in SP w\_sp$$

where w_sp = DV11 approximates processed story points per sprint.

`_resolution_time` *Logic:* resolve blockers quickly. Given numeric dates det$[bl]$ and res$[bl]$.
Math:

$$\min\ Z\_11 = \sum \_bl \in BL (\text{res}[bl] - \text{det}[bl])\ r\_bl$$

`acklog_freshness` *Logic:* encourage frequent backlog updates. With numeric recency score fresh$[b]$.
Math:

$$\max\ Z\_12 = \sum \_b \in B \text{fresh}[b]$$

4 4. Conditions

`s_project_budget` *Logic:* spend within budget.
Math:

$$\sum \_fc^F \text{eff}^F[f] s\_f + \sum \_ec^E \text{eff}^E[e] s\_e + \sum \_sp, kc^K \text{eff}^K[k] z\_k, sp \ \leq \ \sum \_p \text{budget}[p].$$

`espects_velocity` *Logic:* per-sprint load capped by velocity. For each sp belonging to team t via R19 mapping.
Math:

$$\sum \_u \in U \text{sp}[u] \ y\_u, sp \ \leq \ \text{Cap}[sp], \quad \text{Cap}[sp] := \overline{\text{SP}}[v] \text{ for any } v \text{ with } \mathsf{R19}[v,t] = 1.$$

`_most_one_sprint` *Logic:* no double scheduling of a User Story.
Math: $\sum \_sp \in SP y\_u, sp \ \leq \ 1, \ \forall u \in U.$

`s_selected_story` *Logic:* tasks only if their story is selected. If $\mathsf{R10}[u,k] = 1$.
Math: $\sum \_sp z\_k, sp \ \leq \ s\_u, \ \forall (u,k) : \mathsf{R10}[u,k] = 1.$

`_implies_stories` *Logic:* selecting an epic enables its stories.
Math: $s\_u \leq s\_e, \ \forall (e,u) : \mathsf{R9}[e,u] = 1.$

`ust_be_in_backlog` *Logic:* selected features must exist in some backlog.
Math: $s\_f \leq \sum \_b \in B \mathsf{R7}[b,f], \ \forall f \in F.$

`elongs_to_sprint` *Logic:* each sprint backlog belongs to exactly one sprint (data integrity).
Math: $\sum \_sp \in SP \mathsf{R12}[sbl, sp] = 1, \ \forall sbl \in SBL.$

`before_task_done` *Logic:* blocked tasks can't be closed until blocker resolved. If $\mathsf{R16}[k,bl] = 1$.
Math: $d\_k \leq r\_bl, \ \forall (k,bl) : \mathsf{R16}[k,bl] = 1.$

`_moderated_by_sm` *Logic:* every retrospective has a Scrum Master moderator.
Math: $\sum \_m \in M \mathsf{R18}[m, sre] \geq 1, \ \forall sre \in SRE.$

`ages_one_backlog` *Logic:* each Product Owner manages exactly one backlog.
Math: $\sum \_b \in B \mathsf{R5}[o,b] = 1, \ \forall o \in O.$

`most_one_project` *Logic:* team focuses on one project.
Math: $\sum \_p \in Pa\_t, p \leq 1, \ \forall t \in T.$

`exactly_one_team` *Logic:* a worker belongs to exactly one team.
Math: $\sum \_t \in Ta\_w, t = 1, \ \forall w \in W.$

`e_sprint_if_done` *Logic:* completed tasks are scheduled in exactly one sprint.
Math: $\sum \_sp \in SP z\_k, sp = d\_k, \ \forall k \in K.$

5 5. DecisionVariables

- DV0 $y\_u, sp \in \{0,1\}$: assign User Story u to Sprint sp.
- DV1 $z\_k, sp \in \{0,1\}$: schedule Task k in Sprint sp.
- DV2 $a\_w, t \in \{0,1\}$: assign Worker w to Team t.

- DV3 $a\_t, p \in \{0, 1\}$: assign Team t to Project p.
- DV4 $s\_f \in \{0, 1\}$: select Feature f.
- DV5 $s\_e \in \{0, 1\}$: select Epic e.
- DV6 $s\_u \in \{0, 1\}$: select User Story u.
- DV7 $d\_k \in \{0, 1\}$: Task k done (within its sprint).
- DV8 $r\_bl \in \{0, 1\}$: Blocker bl resolved.
- DV9 $r\_rp, f \in \{0, 1\}$: include Feature f in Release Plan rp.
- DV10 $g\_sp \in \{0, 1\}$: Sprint sp achieves its Sprint Goal.
- DV11 $w\_sp \in \mathbb{R}\_\geq 0$: planned sprint workload (story points).
- DV12 $x\_sp \in \{0, 1\}$: Sprint sp generates a Development Snapshot.