Optimization Model for Scrum-Based Software Development

Gemini AI Assistant

September 4, 2025

Contents

1 Sets (Entities)

- The set of all **Projects**, denoted by P.
- The set of all **Teams**, denoted by T.
- The set of all **Workers**, denoted by W.
- The set of all **Features**, denoted by F.
- The set of all **Skills**, denoted by S.
- The set of all **Roles**, denoted by R.
- The set of all **Product Owners**, denoted by PO.
- The set of all **Scrum Masters**, denoted by SM.
- The set of all **Product Backlogs**, denoted by PB.
- The set of all **Sprints**, denoted by SP.
- The set of all **Sprint Plannings**, denoted by SPP.
- The set of all **Daily Scrums**, denoted by DS.
- The set of all **Sprint Reviews**, denoted by SR.
- The set of all **Sprint Retrospectives**, denoted by SRE.
- The set of all **Sprint Backlogs**, denoted by SBL.
- The set of all **Sprint Goals**, denoted by SG.
- The set of all **Epics**, denoted by E.
- The set of all **User Stories**, denoted by US.
- The set of all **Tasks**, denoted by TSK.
- The set of all **Development Snapshots**, denoted by DEV.
- The set of all **Blockers**, denoted by BL.
- The set of all **Stakeholders**, denoted by SH.
- The set of all **Velocities**, denoted by VEL.
- The set of all **Release Plans**, denoted by REP.
- The set of all **Roadmaps**, denoted by RM.
- The set of all **Scrum Boards**, denoted by SCB.
- The set of all **Feature Documentations**, denoted by FED.

2 Indices

- $p \in P$: An index for a project.
- $t \in T$: An index for a team.
- $w \in W$: An index for a worker.
- $f \in F$: An index for a feature.
- $s \in S$: An index for a skill.
- $r \in R$: An index for a role.

- $po \in PO$: An index for a product owner.
- $sm \in SM$: An index for a scrum master.
- $pb \in PB$: An index for a product backlog.
- $sp \in SP$: An index for a sprint.
- $spp \in SPP$: An index for a sprint planning event.
- $ds \in DS$: An index for a daily scrum event.
- $sr \in SR$: An index for a sprint review event.
- $sre \in SRE$: An index for a sprint retrospective event.
- $sbl \in SBL$: An index for a sprint backlog.
- $sg \in SG$: An index for a sprint goal.
- $e \in E$: An index for an epic.
- $us \in US$: An index for a user story.
- $tsk \in TSK$: An index for a task.
- $dev \in DEV$: An index for a development snapshot.
- $bl \in BL$: An index for a blocker.
- $sh \in SH$: An index for a stakeholder.
- $vel \in VEL$: An index for a velocity metric.
- $rep \in REP$: An index for a release plan.
- $rm \in RM$: An index for a roadmap.
- $scb \in SCB$: An index for a scrum board.
- $fed \in FED$: An index for feature documentation.

3 Goals

G0 **maximize_feature_priority**:

$$1.5 \times \max \sum_{f \in F} \sum_{rep \in REP} \text{priority}_f \cdot \text{AssignFeatureToRelease}_{f,rep}$$

G1 **minimize_total_task_effort**:

$$1.0 \times \min \sum_{us \in US} \sum_{sp \in SP} \sum_{tsk \in TSK_{us}} \text{effort}_{tsk} \cdot \text{AssignStoryToSprint}_{us,sp}$$

G2 **maximize_story_points_in_sprint**:

$$1.2 \times \max \sum_{us \in US} \sum_{sp \in SP} \text{story\_points}_{us} \cdot \text{AssignStoryToSprint}_{us,sp}$$

G3 **minimize_project_timeline**:

$$2.0 \times \min(\text{project\_end\_date}_p) \quad \forall p \in P$$

G4 **maximize_team_satisfaction**:

$$0.8 \times \max \frac{\sum_{sre \in SRE} \text{satisfaction}_{sre}}{|SRE|}$$

G5 **minimize_number_of_blockers**:

$$1.0 \times \min \sum_{bl \in BL} 1$$

G6 **maximize_sprint_goal_achievement**:

$$1.3 \times \max \sum_{sg \in SG} \text{achievement\_status}_{sg}$$

G7 **minimize_budget_usage**:

$$1.8 \times \min(\text{budget}_p) \quad \forall p \in P$$

4 Conditions

C0 **worker_must_have_skill**: A worker must have the required skill for an assigned task.

$$\text{RequiresSkill}_{tsk,s} \cdot \text{AssignWorkerToTask}_{w,tsk} \leq \text{HasSkill}_{w,s} \quad \forall w, tsk, s$$

C1 **project_budget_limit**: The total cost must not exceed the project budget.

$$\sum \text{costs} \leq \text{budget}_p \quad \forall p \in P$$

C2 **team_velocity_capacity**: The sum of story points in a sprint cannot exceed the team's velocity.

$$\sum_{us \in US} \text{story\_points}_{us} \cdot \text{AssignStoryToSprint}_{us,sp} \leq \text{velocity}_t \quad \forall sp \in SP, t \in T_{sp}$$

C3 **worker_availability_limit**: A worker's assigned effort cannot exceed their availability.

$$\sum_{tsk \in TSK} \text{effort}_{tsk} \cdot \text{AssignWorkerToTask}_{w,tsk} \leq \text{availability}_w \quad \forall w \in W$$

C4 **prefer_senior_for_complex_tasks**: Soft constraint, could be modeled by adding a penalty term to the objective function if low-level workers are assigned to high-effort tasks.

C5 **user_story_dependency**: A dependent story cannot start before its prerequisite is finished.

$$\text{StartDate}_{us_A} \geq \text{EndDate}_{us_B} \quad \forall (us_A, us_B) \in \text{Dependencies}$$

C6 **worker_cannot_be_on_vacation**: A worker cannot be assigned tasks if their status is 'vacation'.

$$\text{AssignWorkerToTask}_{w,tsk} = 0 \quad \text{if status}_w = \text{'vacation'} \quad \forall w, tsk$$

C7 **sprint_must_have_goal**: Each sprint must have one and only one goal.

$$\sum_{sg \in SG} \text{AssignGoalToSprint}_{sg,sp} = 1 \quad \forall sp \in SP$$

C8 **feature_must_be_in_release**: A feature must be part of a release plan to be worked on.

$$\sum_{rep \in REP} \text{AssignFeatureToRelease}_{f,rep} \geq 1 \quad \forall f \in F_{\text{active}}$$

5 Decision Variables

DV0 **assign_worker_to_task** ($\text{AssignWorkerToTask}_{w,tsk}$): Binary variable, equal to 1 if worker $w \in W$ is assigned to task $tsk \in TSK$, and 0 otherwise.

$$\text{AssignWorkerToTask}_{w,tsk} \in \{0,1\} \quad \forall w \in W, tsk \in TSK$$

DV1 **assign_user_story_to_sprint** ($\text{AssignStoryToSprint}_{us,sp}$): Binary variable, equal to 1 if user story $us \in US$ is assigned to sprint $sp \in SP$, and 0 otherwise.

$$\text{AssignStoryToSprint}_{us,sp} \in \{0,1\} \quad \forall us \in US, sp \in SP$$

DV2 **assign_feature_to_release** ($\text{AssignFeatureToRelease}_{f,rep}$): Binary variable, equal to 1 if feature $f \in F$ is included in release plan $rep \in REP$, and 0 otherwise.

$$\text{AssignFeatureToRelease}_{f,rep} \in \{0,1\} \quad \forall f \in F, rep \in REP$$

DV3 **assign_worker_to_team** ($\text{AssignWorkerToTeam}_{w,t}$): Binary variable, equal to 1 if worker $w \in W$ is a member of team $t \in T$, and 0 otherwise.

$$\text{AssignWorkerToTeam}_{w,t} \in \{0,1\} \quad \forall w \in W, t \in T$$

DV4 **set_sprint_for_task** ($\text{SprintForTask}_{tsk}$): Integer variable defining the sprint number in which a task is scheduled.

$$\text{SprintForTask}_{tsk} \in \mathbb{Z}^+ \quad \forall tsk \in TSK$$

DV5 **determine_team_size** (TeamSize_t): Integer variable representing the number of workers in team t.

$$\text{TeamSize}_t \in \{3,4,...,9\} \quad \forall t \in T$$

DV6 **worker_allocation_percentage** ($\text{Allocation}_{w,p}$): Continuous variable for the percentage of worker w's availability allocated to project p.

$$0 \leq \text{Allocation}_{w,p} \leq 100 \quad \forall w \in W, p \in P$$

DV7 **set_user_story_priority** (Priority_{us}): Integer variable to determine the priority ranking of a user story.

$$\text{Priority}_{us} \in \{1,2,...,100\} \quad \forall us \in US$$