Mathematical Optimization Model for Scrum-based Software Development

Gemini AI

September 4, 2025

Contents

1 Sets (Entities) **2**

2 Indices **3**

3 Goals **4**

4 Conditions **5**

5 Decision Variables **7**

This document provides a formal mathematical and logical specification for an optimization model based on a Scrum-based software development domain. The model is defined by its sets (entities), indices, decision variables, conditions (constraints), and goals (objective functions). The definitions are derived from the provided CSV files: `Entities.csv`, `Relationships.csv`, `Goals.csv`, `Conditions.csv`, and `DecisionVariables.csv`.

1 Sets (Entities)

The fundamental sets of the model correspond to the entities defined in the domain. Each entity is represented by a calligraphic letter.

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

2 Indices

Indices are used to refer to individual elements within the sets defined above.

- $p \in \mathcal{P}$: Index for Projects.
- $t \in \mathcal{T}$: Index for Teams.
- $w \in \mathcal{W}$: Index for Workers.
- $f \in \mathcal{F}$: Index for Features.
- $s \in \mathcal{S}$: Index for Skills.
- $r \in \mathcal{R}$: Index for Roles.
- $po \in \mathcal{PO}$: Index for Product Owners.
- $sm \in \mathcal{SM}$: Index for Scrum Masters.
- $pb \in \mathcal{PB}$: Index for Product Backlogs.
- $sp \in \mathcal{SP}$: Index for Sprints.
- $spp \in \mathcal{SPP}$: Index for Sprint Planning meetings.
- $ds \in \mathcal{DS}$: Index for Daily Scrums.
- $sr \in \mathcal{SR}$: Index for Sprint Reviews.
- $sre \in \mathcal{SRE}$: Index for Sprint Retrospectives.
- $sbl \in \mathcal{SBL}$: Index for Sprint Backlogs.
- $sg \in \mathcal{SG}$: Index for Sprint Goals.
- $e \in \mathcal{E}$: Index for Epics.
- $us \in \mathcal{US}$: Index for User Stories.
- $tsk \in \mathcal{TSK}$: Index for Tasks.
- $dev \in \mathcal{DEV}$: Index for Development Snapshots.
- $bl \in \mathcal{BL}$: Index for Blockers.
- $sh \in \mathcal{SH}$: Index for Stakeholders.
- $vel \in \mathcal{VEL}$: Index for Velocity measurements.
- $rep \in \mathcal{REP}$: Index for Release Plans.
- $rm \in \mathcal{RM}$: Index for Roadmaps.
- $scb \in \mathcal{SCB}$: Index for Scrum Boards.
- $fed \in \mathcal{FED}$: Index for Feature Documentations.

3 Goals

The goals represent the objective functions of the optimization problem. The overall objective would be a weighted sum of these individual goals. Parameters are denoted by lowercase text (e.g., $priority_f$). Decision variables are denoted by letters like x, y, z.

- **G0: maximize_feature_value**
  Maximize the business value, represented by the sum of priorities of all features selected for development.
  $$\text{maximize} \sum_{f \in \mathcal{F}} priority_f \cdot z_f$$
  (where $z_f = 1$ if feature f is selected)

- **G1: maximize_story_points_per_sprint**
  Maximize the total story points of user stories completed across all sprints.
  $$\text{maximize} \sum_{sp \in \mathcal{SP}} \sum_{us \in \mathcal{US}} storypoints_{us} \cdot y_{us,sp}$$
  (where $y_{us,sp} = 1$ if user story us is completed in sprint sp)

- **G2: minimize_total_effort**
  Minimize the sum of estimated effort for all features selected for a release.
  $$\text{minimize} \sum_{rep \in \mathcal{REP}} \sum_{f \in \mathcal{F}} effort_f \cdot z_{f,rep}$$
  (where $z_{f,rep} = 1$ if feature f is in release plan rep)

- **G3: maximize_team_velocity**
  Maximize the average story points achieved by the teams. This is an outcome, modeled as maximizing the velocity parameter V_t for each team.
  $$\text{maximize} \sum_{t \in \mathcal{T}} V_t$$

- **G4: minimize_blocker_resolution_time**
  Minimize the time between when a blocker is detected and when it is resolved.
  $$\text{minimize} \sum_{bl \in \mathcal{BL}} (resolved\_on_{bl} - detected\_on_{bl})$$

- **G5: maximize_stakeholder_satisfaction**
  Prioritize features relevant to stakeholders with high influence.
  $$\text{maximize} \sum_{f \in \mathcal{F}} \sum_{sh \in \mathcal{SH}} influence_{sh} \cdot relevance_{sh,f} \cdot z_f$$
  (where $relevance_{sh,f}$ links stakeholder sh to feature f)

- **G6: maximize_sprint_goal_achievement**
  Maximize the successful achievement of sprint goals.
  $$\text{maximize} \sum_{sg \in \mathcal{SG}} achievement\_status_{sg}$$

- **G7: minimize_project_duration**
  Minimize the total time span of projects.

$$\text{minimize} \sum_{p \in \mathcal{P}} (project\_end_p - project\_start_p)$$

- **G8: maximize_team_satisfaction**
  Maximize the self-reported satisfaction scores of all teams from retrospectives.

$$\text{maximize} \sum_{sre \in \mathcal{SRE}} team\_satisfaction_{sre}$$

- **G9: minimize_budget_consumption**
  Minimize the total cost, modeled as a function of effort for completed tasks. Let C be a cost-per-effort-unit factor.

$$\text{minimize} \sum_{tsk \in \mathcal{TSK}} effort_{tsk} \cdot C \cdot x_{tsk}$$

  (where $x_{tsk} = 1$ if task tsk is completed)

- **G10: maximize_task_completion_rate**
  Maximize the number of tasks that reach the 'done' status.

$$\text{maximize} \sum_{tsk \in \mathcal{TSK}} x_{tsk}$$

- **G11: maximize_epic_priority**
  Maximize the sum of priorities for all epics being worked on.

$$\text{maximize} \sum_{e \in \mathcal{E}} priority_e \cdot w_e$$

  (where $w_e = 1$ if epic e is being worked on)

4 Conditions

The conditions represent the constraints of the optimization problem, ensuring that solutions are feasible and respect the rules of the domain.

- **C0: enforce_team_size_limit**
  The number of workers in a team must be within a defined range $[S_{min}, S_{max}]$.

$$\forall t \in \mathcal{T}: \quad S_{min} \leq team\_size_t \leq S_{max}$$

- **C1: respect_worker_availability**
  The sum of effort for all tasks assigned to a worker must not exceed their availability.

$$\forall w \in \mathcal{W}: \quad \sum_{tsk \in \mathcal{TSK}} effort_{tsk} \cdot a_{w,tsk} \leq availability_w$$

  (where $a_{w,tsk} = 1$ if worker w is assigned to task tsk)

- **C2: stay_within_sprint_capacity**
  The total story points of user stories assigned to a sprint must not exceed the velocity of the team performing the sprint. Let $t(sp)$ be the team for sprint sp.
  $$\forall sp \in \mathcal{SP}: \quad \sum_{us \in \mathcal{US}} storypoints_{us} \cdot y_{us,sp} \leq V_{t(sp)}$$

- **C3: project_must_be_within_budget**
  The total cost of a project must not exceed its allocated budget.
  $$\forall p \in \mathcal{P}: \quad \sum_{tsk \in \mathcal{TSK}(p)} effort_{tsk} \cdot C \leq budget_p$$
  (where $\mathcal{TSK}(p)$ are all tasks belonging to project p)

- **C4: user_story_requires_criteria**
  A user story can only be assigned to a sprint if its acceptance criteria are defined. Let $AC_{us} = 1$ if criteria exist for user story us.
  $$\forall us \in \mathcal{US}, \forall sp \in \mathcal{SP}: \quad y_{us,sp} \leq AC_{us}$$

- **C5: task_status_must_be_valid**
  A user story is considered complete only if all its constituent tasks are complete. Let $\mathcal{TSK}(us)$ be the set of tasks for user story us.
  $$\forall us \in \mathcal{US}, \forall sp \in \mathcal{SP}: \quad |\mathcal{TSK}(us)| \cdot y_{us,sp} \leq \sum_{tsk \in \mathcal{TSK}(us)} x_{tsk}$$

- **C6: sprint_has_defined_goal**
  A team can only work on a sprint if a sprint goal is defined. Let $g_{sp} = 1$ if sprint sp has a goal.
  $$\forall sp \in \mathcal{SP}, \forall us \in \mathcal{US}: \quad y_{us,sp} \leq g_{sp}$$

- **C7: feature_requires_priority**
  A feature must have a defined priority to be selected. Let $P_f = 1$ if feature f has a priority.
  $$\forall f \in \mathcal{F}: \quad z_f \leq P_f$$

- **C8: avoid_high_severity_blockers**
  A task cannot be worked on if it is blocked by an unresolved, high-severity blocker. Let $B_{tsk} = 1$ if task tsk is blocked by such a blocker.
  $$\forall w \in \mathcal{W}, \forall tsk \in \mathcal{TSK}: \quad a_{w,tsk} + B_{tsk} \leq 1$$

- **C9: sprint_duration_is_fixed**
  The duration of each sprint must be equal to a standard duration Δ_{sprint}.
  $$\forall sp \in \mathcal{SP}: \quad end\_date_{sp} - start\_date_{sp} = \Delta_{sprint}$$

- **C10: skill_level_must_be_adequate**
  A worker can only be assigned to a task if they possess the required skill at a sufficient level. Let $ReqSkill_{tsk,s}$ be the required level of skill s for task tsk, and $WorkerSkill_{w,s}$ be the level of worker w in skill s.
  $$\forall w, tsk, s: \quad a_{w,tsk} \cdot ReqSkill_{tsk,s} \leq WorkerSkill_{w,s}$$

- **C11: epic_must_be_detailed**
  A user story can only be derived from an epic if the epic has a description. Let $ED_e = 1$ if epic e has a description, and $us(e)$ be the user stories for epic e.
  $$\forall e \in \mathcal{E}, \forall us \in us(e): \quad y_{us,sp} \leq ED_e$$

5 Decision Variables

The decision variables represent the choices to be made by the model.

- **DV0: assign_story_to_sprint**
  A binary variable indicating if a user story is assigned to and completed in a sprint.

$$y_{us,sp} = \begin{cases} 1 & \text{if User Story } us \text{ is completed in Sprint } sp \\ 0 & \text{otherwise} \end{cases}$$

- **DV1: assign_worker_to_task**
  A binary variable indicating if a worker is assigned to a task.

$$a_{w,tsk} = \begin{cases} 1 & \text{if Worker } w \text{ is assigned to Task } tsk \\ 0 & \text{otherwise} \end{cases}$$

- **DV2: select_feature_for_release**
  A binary variable indicating if a feature is included in a release plan.

$$z_{f,rep} = \begin{cases} 1 & \text{if Feature } f \text{ is in Release Plan } rep \\ 0 & \text{otherwise} \end{cases}$$

- **DV3: set_user_story_priority**
  An integer variable representing the assigned priority of a user story.

$$P_{us} \in \{1, 2, 3, 4, 5\}, \quad \forall us \in \mathcal{US}$$

- **DV4: estimate_user_story_points**
  An integer variable representing the estimated effort for a user story.

$$SP_{us} \in \{1, 2, 3, 5, 8, 13, 21\}, \quad \forall us \in \mathcal{US}$$

- **DV5: adjust_team_size**
  An integer variable for the number of members in a team.

$$N_t \in \{3, 4, ..., 9\}, \quad \forall t \in \mathcal{T}$$

- **DV10: assign_team_to_project**
  A binary variable indicating if a team is assigned to a project.

$$q_{t,p} = \begin{cases} 1 & \text{if Team } t \text{ is assigned to Project } p \\ 0 & \text{otherwise} \end{cases}$$