Optimization Model for SCRUM Software Development

AI Analyst

September 5, 2025

Contents

1 Sets (Entities)

- $Project$: The product or initiative to be developed
- $Team$: Self-organized, cross-functional development team
- $Worker$: Individual team member working on the project
- $Feature$: Mid-sized functionality
- $Skill$: Professional or social competence of a worker
- $Role$: Defined responsibilities within the Scrum team
- $ProductOwner$: Responsible for product vision and Product Backlog
- $ScrumMaster$: Supports the team in applying Scrum
- $ProductBacklog$: Ordered list of all requirements
- $Sprint$: Fixed time period for creating an increment
- $SprintPlanning$: Kick-off meeting for Sprint preparation
- $DailyScrum$: Daily 15-minute team meeting

- $SprintReview$: Presentation and acceptance of results
- $SprintRetrospective$: Retrospective for process improvement
- $SprintBacklog$: Selected backlog items + implementation plan
- $SprintGoal$: Objective to be achieved within the sprint
- $Epic$: Large requirement that can be split into stories
- $UserStory$: Requirement from the perspective of a user
- $Task$: Smallest unit of work within a sprint
- $DevelopmentSnapshot$: Product at the end of a sprint
- $Blocker$: Obstacle hindering progress
- $Stakeholder$: Interested party in the product (internal/external)
- $Velocity$: Average amount of work per sprint
- $ReleasePlan$: Plan for releasing specific features
- $Roadmap$: Long-term planning across releases
- $ScrumBoard$: Visual representation of tasks during the sprint
- $FeatureDocumentation$: Documentation for a specific feature

2 Indices

- $p, p' \in Project$
- $t, t' \in Team$
- $w, w' \in Worker$
- $f, f' \in Feature$
- $s, s' \in Skill$
- $r, r' \in Role$
- $po \in ProductOwner$
- $sm \in ScrumMaster$
- $pb \in ProductBacklog$
- $sp, sp' \in Sprint$
- $spp \in SprintPlanning$

- $ds \in DailyScrum$
- $sr \in SprintReview$
- $sre \in SprintRetrospective$
- $sbl \in SprintBacklog$
- $sg \in SprintGoal$
- $e, e' \in Epic$
- $us, us' \in UserStory$
- $tsk, tsk' \in Task$
- $dev \in DevelopmentSnapshot$
- $bl, bl' \in Blocker$
- $sh, sh' \in Stakeholder$
- $vel \in Velocity$
- $rep \in ReleasePlan$
- $rm \in Roadmap$
- $scb \in ScrumBoard$
- $fed \in FeatureDocumentation$

3 Goals

- **G0: maximize_team_velocity** - Maximize the average velocity of the team

$$\text{maximize} \sum_{t \in Team} vel.avg\_story\_points(t)$$

- **G1: minimize_blocker_severity** - Minimize the total severity of active blockers

$$\text{minimize} \sum_{bl \in Blocker} bl.severity(bl)$$

- **G2: maximize_feature_priority** - Maximize the total priority of features in the release

$$\text{maximize} \sum_{f \in Feature} f.priority(f)$$

- **G3: minimize_sprint_effort_variance** - Minimize the variance between estimated and actual effort in a sprint

$$\text{minimize } |sbl.total\_effort_{estimated} - sbl.total\_effort_{actual}|$$

- **G4: maximize_worker_availability** - Maximize the total availability of workers

$$\text{maximize} \sum_{w \in Worker} w.availability(w)$$

- **G5: minimize_project_budget** - Minimize the total project budget

$$\text{minimize } p.budget(p)$$

- **G6: maximize_stakeholder_satisfaction** - Maximize the average satisfaction from sprint reviews

$$\text{maximize} \sum_{sr \in SprintReview} sr.feedback\_documentation(sr)$$

- **G7: minimize_task_blocked_time** - Minimize the time tasks spend in a blocked state

$$\text{minimize} \sum_{bl \in Blocker} (bl.resolved\_on(bl) - bl.detected\_on(bl))$$

- **G8: maximize_skill_coverage** - Maximize the coverage of required skills in the team

$$\text{maximize} \sum_{s \in Skill} s.level(s)$$

- **G9: minimize_sprint_goal_failure** - Minimize the number of sprints where the goal was not achieved

$$\text{minimize} \sum_{sg \in SprintGoal} (1 - sg.achievement\_status(sg))$$

4 Conditions

- **C0: team_has_scrum_master** - A team must have a Scrum Master assigned

$$\forall t \in Team, \exists sm \in ScrumMaster : is\_supported\_by(t, sm)$$

- **C1: worker_availability_gt_80** - Worker availability must be greater than 80%

$$\forall w \in Worker, w.availability(w) > 0.8$$

- **C2: project_budget_lt_100000** - Project budget must not exceed 100,000

$$\forall p \in Project, p.budget(p) \leq 100000$$

- **C3: sprint_duration_lte_4_weeks** - Sprint duration must be 4 weeks or less

$$\forall sp \in Sprint, (sp.end\_date(sp) - sp.start\_date(sp)) \leq 28$$

- **C4: user_story_has_acceptance_criteria** - A User Story must have defined acceptance criteria

$$\forall us \in UserStory, us.acceptance\_criteria(us) \neq \emptyset$$

- **C5: feature_in_backlog_has_priority** - A Feature in the backlog must have a priority set

$$\forall f \in Feature, f.priority(f) \neq \emptyset$$

- **C6: blocker_must_have_severity** - A Blocker must have a severity level defined

$$\forall bl \in Blocker, bl.severity(bl) \neq \emptyset$$

- **C7: sprint_goal_defined** - A Sprint must have a defined goal

$$\forall sp \in Sprint, \exists sg \in SprintGoal : pursues\_goal(sp, sg)$$

- **C8: task_effort_gt_0** - A Task must have an effort greater than 0

$$\forall tsk \in Task, tsk.effort(tsk) > 0$$

- **C9: velocity_based_on_min_1_sprint** - Velocity calculation must be based on at least one sprint

$$\forall vel \in Velocity, vel.number\_of\_sprints\_used(vel) \geq 1$$

5 DecisionVariables

- $DV0_{w,t}$: assign_worker_to_team $\in$ 0,1
- $DV1_{f,sp}$: select_feature_for_sprint $\in$ 0,1
- $DV2_{sp}$: set_sprint_duration $\in$ 7,14,21,28
- $DV3_{f}$: allocate_budget_to_feature $\in$ [0, 100000]
- $DV4_{w}$: set_worker_availability $\in$ [0, 1]
- $DV5_{us}$: assign_story_points $\in N$, $1 \leq DV5_{us} \leq 20$

- $DV6_f$: prioritize_feature $\in$ 1,2,3,4,5
- $DV7_{bl}$: set_blocker_severity $\in$ 1,2,3,4,5
- $DV8_t$: define_team_size $\in N$, $3 \leq DV8_t \leq 9$
- $DV9_{sg}$: set_sprint_goal_status $\in$ 0,1