Optimization Model for SCRUM-Based 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_project_budget** - Minimize the total budget spent on the project

$$\text{minimize} \sum_{p \in Project} p.budget$$

- **G2: minimize_blocker_severity** - Minimize the total severity of active blockers

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

- **G3: maximize_feature_priority** - Maximize the total priority of features in the release

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

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

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

- **G5: minimize_sprint_duration** - Minimize the total duration of all sprints

$$\text{minimize} \sum_{sp \in Sprint} sp.duration\_(min)$$

- **G6: maximize_story_points_done** - Maximize the sum of completed story points

$$\text{maximize} \sum_{\substack{us \in UserStory \\ us.status =' Done'}} us.story\_points$$

- **G7: minimize_task_effort** - Minimize the total effort of all tasks

$$\text{minimize} \sum_{tsk \in Task} tsk.effort$$

- **G8: maximize_stakeholder_satisfaction** - Maximize the average team satisfaction from retrospectives

$$\text{maximize } sre.team\_satisfaction \quad \forall\ sre \in SprintRetrospective$$

- **G9: minimize_skill_gap** - Minimize the gap between required and available skill levels

$$\text{minimize} \sum_{s \in Skill} |s.level_{required} - s.level_{available}|$$

4 Conditions

- **C0: project_budget_limit** - The total project cost must not exceed the budget

$$\sum cost(p) \leq p.budget \quad \forall\ p \in Project$$

- **C1: sprint_goal_achieved** - The sprint goal must be achieved

$$sg.achievement\_status =' Achieved' \quad \forall\ sg \in SprintGoal$$

- **C2: team_cross_functional** - The team must have all required roles covered

$$\sum_{r \in Role} 1_{assigned}(r, t) \geq |RequiredRoles| \quad \forall\, t \in Team$$

- **C3: worker_availability_min** - Each worker must have at least 80% availability

$$w.availability \geq 0.8 \quad \forall\, w \in Worker$$

- **C4: feature_priority_high** - Critical features must have a priority above a threshold

$$f.priority \geq threshold_{high} \quad \forall\, f \in Feature_{critical}$$

- **C5: story_points_per_sprint** - The total story points per sprint must not exceed team velocity

$$\sum_{us \in UserStory_{sbl}} us.story\_points \leq vel.avg\_story\_points(t) \quad \forall\, sbl \in SprintBacklog$$

- **C6: no_severe_blockers** - No active blocker with severity above 'High' is allowed

$$bl.severity \notin \{4, 5\} \quad \forall\, bl \in Blocker_{active}$$

- **C7: task_status_done** - All tasks in the sprint backlog must be 'Done' by the end

$$tsk.status =' Done' \quad \forall\, tsk \in Task_{sbl}$$

- **C8: snapshot_test_passed** - The development snapshot must have a 'Passed' test status

$$dev.test\_status =' Passed' \quad \forall\, dev \in DevelopmentSnapshot$$

- **C9: review_attendance** - Key stakeholders must attend the sprint review

$$sr.attendees\_count \geq |Stakeholder_{key}| \quad \forall\, sr \in SprintReview$$

5 Decision Variables

- $DV0_{w,tsk}$: assign_worker_to$_task \in \{0, 1\}$
- $DV1_f$: select_feature_for_release $\in \{0, 1\}$
- $DV2_{us}$: select_user_story_for_sprint $\in \{0, 1\}$
- $DV3_t$: team_size $\in Z^+,\ [3, 9]$

- $DV4_{sp}$: sprint_duration $\in Z^{+}$, $[7, 21]$
- $DV5_{w}$: worker_availability_percentage $\in R^{+}$, $[0.0, 1.0]$
- $DV6_{us}$: story_point_estimate $\in Z^{+}$, $[1, 20]$
- $DV7_{tsk}$: task_effort_hours $\in R^{+}$, $[0.5, 40.0]$
- $DV8_{w,s}$: skill_level $\in \{1, 2, 3, 4, 5\}$
- $DV9_{bl}$: blocker_severity_score $\in \{1, 2, 3, 4, 5\}$