Optimization Model for SCRUM-based Software Development

Generated by ChatGPT

August 6, 2025

Problem Overview

This optimization model formalizes the decision-making structure for a SCRUM-based software development company. The goal is to maximize productivity, skill utilization, and stakeholder satisfaction, while minimizing blockers and project duration, all under resource and procedural constraints.

Sets

- P – Set of Projects
- T – Set of Teams
- E – Set of Employees
- F – Set of Features
- S – Set of Skills
- R – Set of Roles
- PO – Set of Product Owners
- SM – Set of Scrum Masters
- PB – Set of Product Backlogs
- SP – Set of Sprints
- EP – Set of Epics
- US – Set of User Stories
- TK – Set of Tasks
- STK – Set of Stakeholders

Decision Variables

- $x_{e,t} \in \{0,1\}$: 1 if employee $e \in E$ is assigned to team $t \in T$
- $y_{t,p} \in \{0,1\}$: 1 if team $t \in T$ is assigned to project $p \in P$
- $z_{e,s} \in \{0,1\}$: 1 if employee $e \in E$ has skill $s \in S$
- $a_{us,tk} \in \{0,1\}$: 1 if task $tk \in TK$ belongs to user story $us \in US$
- $g_{sp} \in \{0,1\}$: 1 if sprint $sp \in SP$ achieves its goal
- v_t: Velocity of team $t \in T$

Objective Function

Maximize:

$$\max \left(\sum_{t \in T} v_t + \sum_{e \in E} \sum_{s \in S} cert_{e,s} \cdot z_{e,s} + \sum_{sp \in SP} g_{sp} - \sum_{tk \in TK} blocker_{tk} \right)$$

Constraints

Each employee assigned to at most one team:	$\sum_{t \in T} x_{e,t} \leq 1 \quad \forall e \in E$
Each team assigned to one project:	$\sum_{p \in P} y_{t,p} = 1 \quad \forall t \in T$
Certified skill match:	$z_{e,s} \leq skillMatch_{e,s} \quad \forall e \in E, s \in S$
Each task must belong to one user story:	$\sum_{us \in US} a_{us,tk} = 1 \quad \forall tk \in TK$
Sprint must pursue a goal:	$g_{sp} \geq minGoal_{sp} \quad \forall sp \in SP$
Blocker limit:	$blocker_{tk} = 0$ for critical tasks
Availability check:	$availability_e \cdot x_{e,t} \geq minAvailability \quad \forall e \in E, t \in T$

Notes

- Goal types are encoded in the objective function, e.g., maximization of availability, skill match, velocity; minimization of blockers and project duration.
- Logical constraints such as "must-match" or "cannot-match" are embedded in constraints or objective penalties.