FERI

University of Maribor

Faculty of Electrical Engineering and Computer Science

Scrum Task Allocation Based on Particle Swarm Optimization

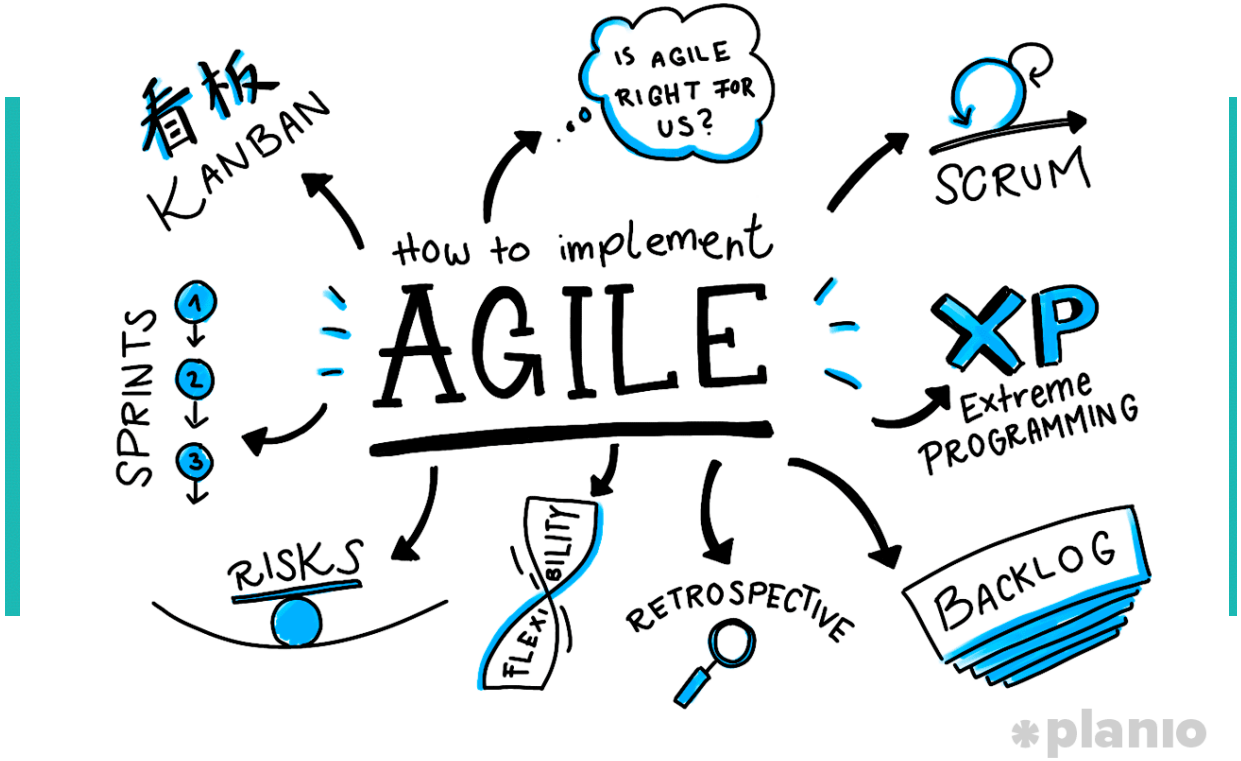
Authors: Lucija Brezočnik, Iztok Fister Jr., and Vili Podgorelec
Faculty of Electrical Engineering and Computer Science
University of Maribor, Slovenia

Conference: 8th International Conference on Bioinspired Optimization Methods and their Applications

Introduction

University of Maribor
Faculty of Electrical Engineering and Computer Science

How can a organization survive on market?

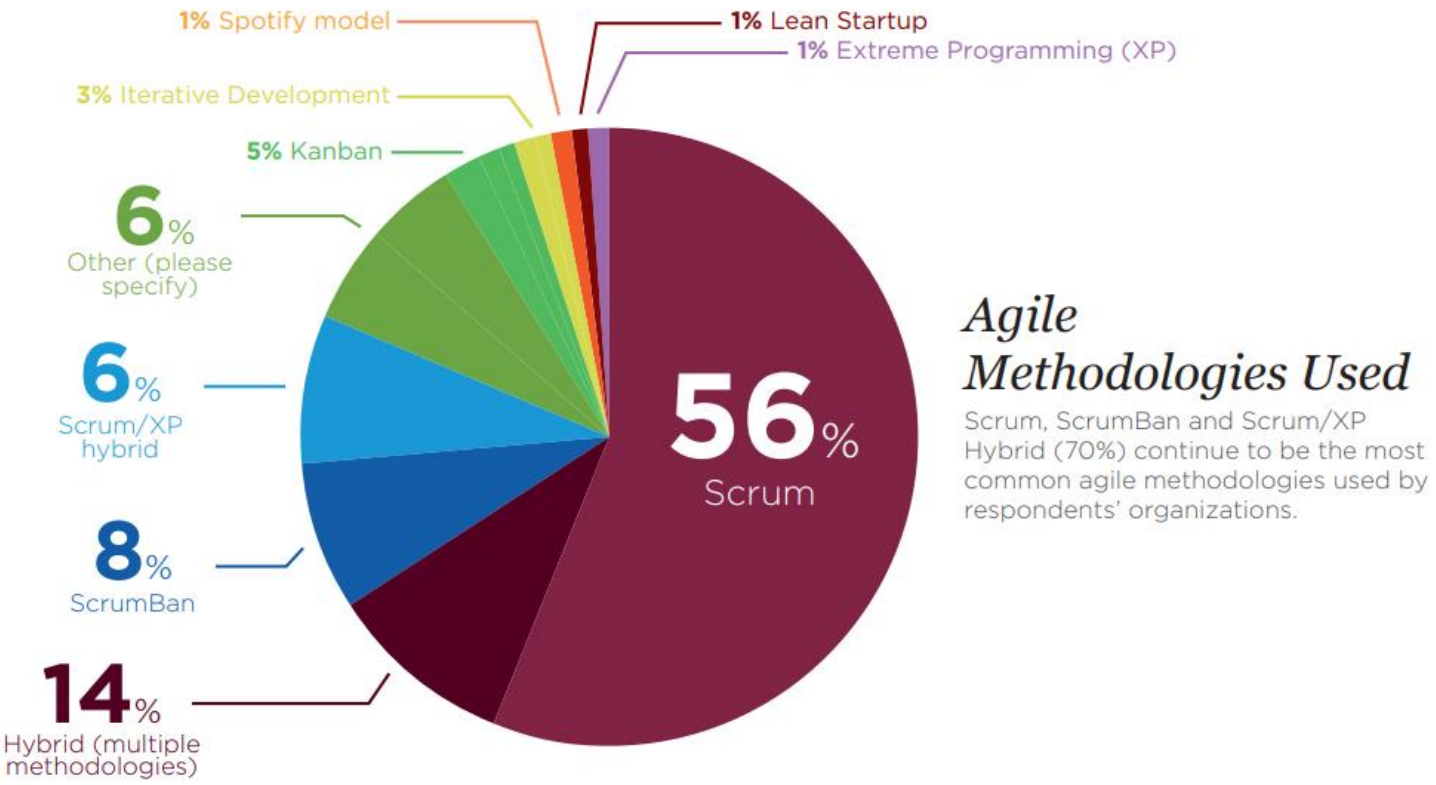


Source: https://plan.io/blog/ultimate-guide-to-implementing-agile-project-management-and-scrum/

Introduction

University of Maribor
Faculty of Electrical Engineering and Computer Science

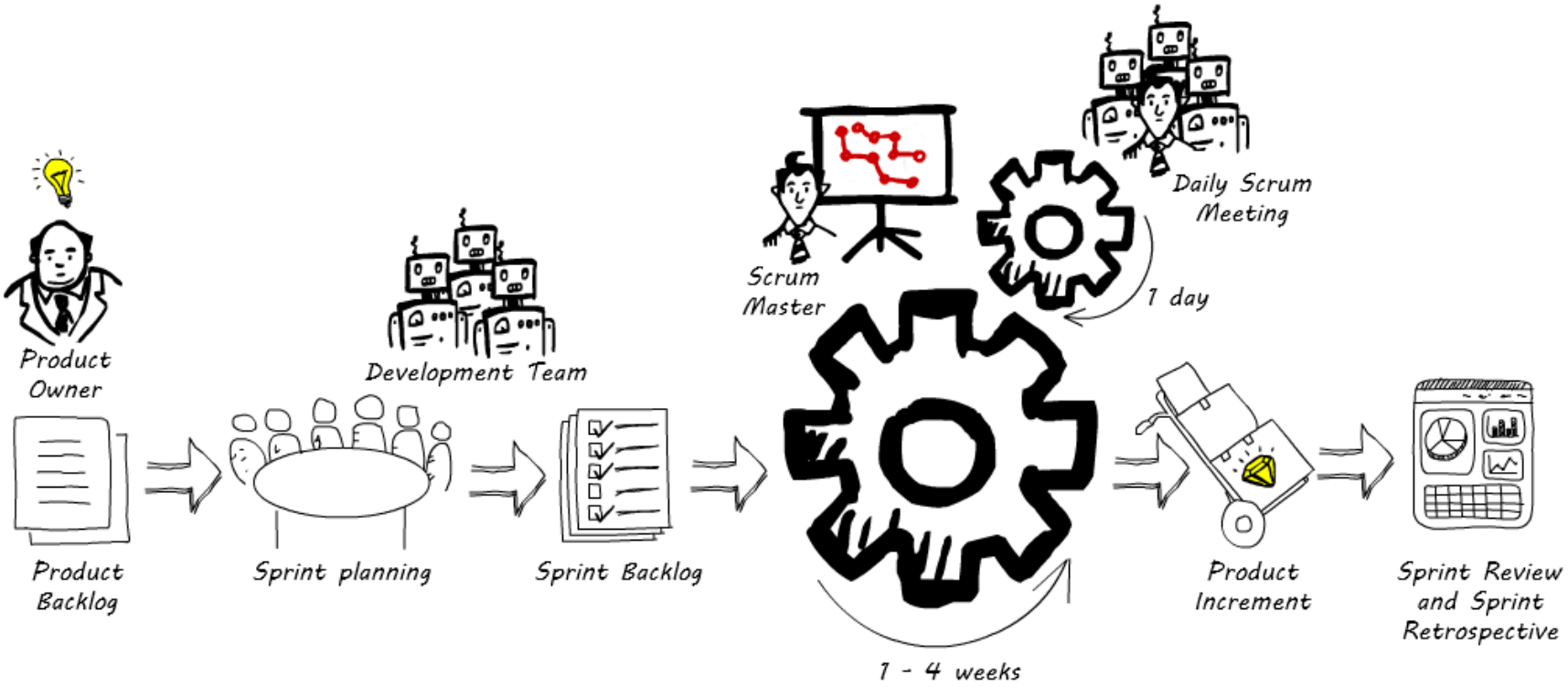
Which agile methodology should we choose?



Source: https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report

Scrum

University of Maribor
Faculty of Electrical Engineering and Computer Science

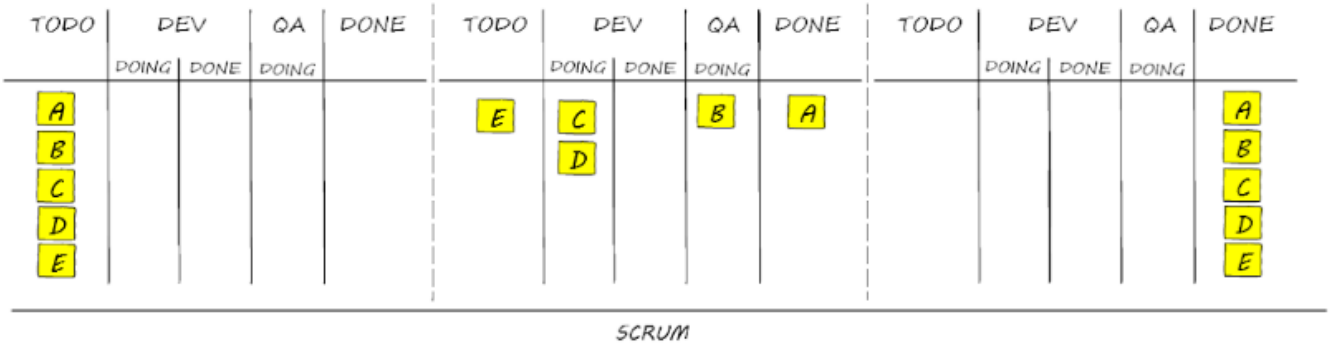


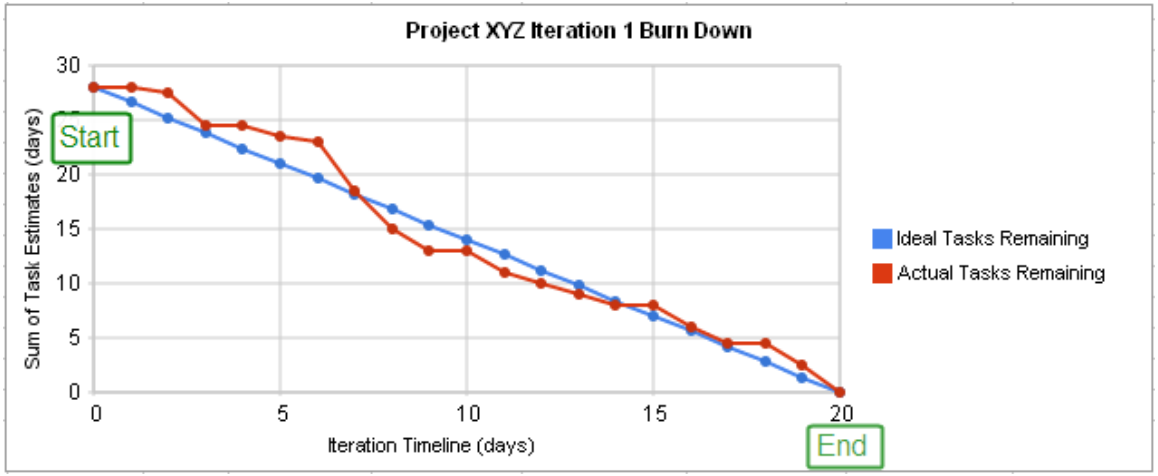
A framework for developing, delivering, and sustaining complex products.

Scrum

University of Maribor
Faculty of Electrical Engineering and Computer Science

- Sprint/iteration planning is applied in 88 % of all agile projects





Source: https://en.wikipedia.org/wiki/Burn_down_chart

Motivation

University of Maribor
Faculty of Electrical Engineering and Computer Science

- Represent Scrum task allocation as an optimization problem.
  - Problem is dealt daily in organizations all around the world.
- Propose the Particle Swarm Optimization algorithm for solving Scrum task allocation, or simply STAPSO.
- Test the proposed algorithm on a real dataset.

Task allocation problem

University of Maribor
Faculty of Electrical Engineering and Computer Science

- *n_days* – duration of the Sprint
- *t_effort* – total estimated effort
- *Oline* – optimal line
- *Cline* – current line
- *Edone* – summarized effort of the tasks per given day

$$opt\_day = \frac{t\_effort}{n\_days}$$

$$Oline = \frac{t\_effort}{n\_days} * x + t\_effort$$

$$Cline = Oline(x) - (Oline(x-1) - Edone)$$

PSO

University of Maribor
Faculty of Electrical Engineering and Computer Science

Algorithm 1. Pseudocode of the basic PSO algorithm

Input: PSO population of particles $\mathbf{x_i} = (x_{i1}, \ldots, x_{iD})^T$ for $i = 1 \ldots Np$, MAX_FEs.
Output: The best solution $\mathbf{x}_{best}$ and its corresponding value $f_{min} = \min(f(\mathbf{x}))$.

```
1: init_particles;
2: eval = 0;
3: while termination_condition_not_meet do
4:    for i = 1 to Np do
5:       f_i = evaluate_the_new_solution(x_i);
6:       eval = eval + 1;
7:       if f_i ≤ pBest_i then
8:          p_i = x_i; pBest_i = f_i;  // save the local best solution
9:       end if
10:      if f_i ≤ f_min then
11:         x_best = x_i; f_min = f_i;  // save the global best solution
12:      end if
13:      x_i = generate_new_solution(x_i);
14:   end for
15: end while
```

PSO

University of Maribor
Faculty of Electrical Engineering and Computer Science

- Modifications:
  - representation of individuals,
  - design of fitness function, and
  - constraint handling.

Representation of individuals

University of Maribor
Faculty of Electrical Engineering and Computer Science

- PSO: candidate solutions – real-valued vectors **x**
- STAPSO: problem space – integer vector **y** (symbolizing the effort of a particular task)

mapping

Task_ID	Effort
0	3
1	2
2	4
3	3
4	5

	Dimension j				
Elements i	0	1	2	3	4
Candidate solution $\mathbf{x}_i$	0.70	0.42	0.21	0.94	0.52
Permutation π_i	3	1	0	4	2
Task allocation $\mathbf{y}_i$	3	2	3	5	4

Fitness function

University of Maribor
Faculty of Electrical Engineering and Computer Science

$$f(x) = |\sum_{j=0}^{n\_days} (calculated\_effort\_per\_day_j)|$$

- calculated effort per day:

$$\forall d \in \{1, 2, \ldots, n\_days\}, \forall t \in \{1, 2, \ldots, n\_tasks(d)\},$$
$$\sum_{i=1}^{t} effort(i) \leq opt\_d$$

Constraint handling

University of Maribor
Faculty of Electrical Engineering and Computer Science

- **Problem**: particular order (dependency) of some tasks
- unfeasible solutions are penalized

Algorithm 2 Constraint handling in STAPSO

```
violations = 0;
fitness = f(x);{calculated by Eq. 6}
for i = 1 to Num_Rules do
    if is_violated() then
        violations = violations + 1;
    end if
end for
if violations > 0 then
    fitness = violations * 1000;
end if
```

Experimental setup

University of Maribor
Faculty of Electrical Engineering and Computer Science

- Parameter settings:
  - population size *Np*: 75,
  - dimension of the problem *D*: 60,
  - number of function evaluations per run *MAXFEs*= 30000,
  - total runs: 25,
  - cognitive component *C1* = 2.0,
  - social component *C2* = 2.0,
  - velocity: [-4, 4],
  - number of days: 10,
  - number of Sprint: 1.

Experiment – test data and constraints

University of Maribor
Faculty of Electrical Engineering and Computer Science

- Test data:
  - internal project
  - hard to get test data – why?

- Constraints:

$$\Psi = \{(T7, T3), (T6, T22), (T4, T58), (T33, T31)\}$$

Ψ denotes the implementation order of the tasks.

Results

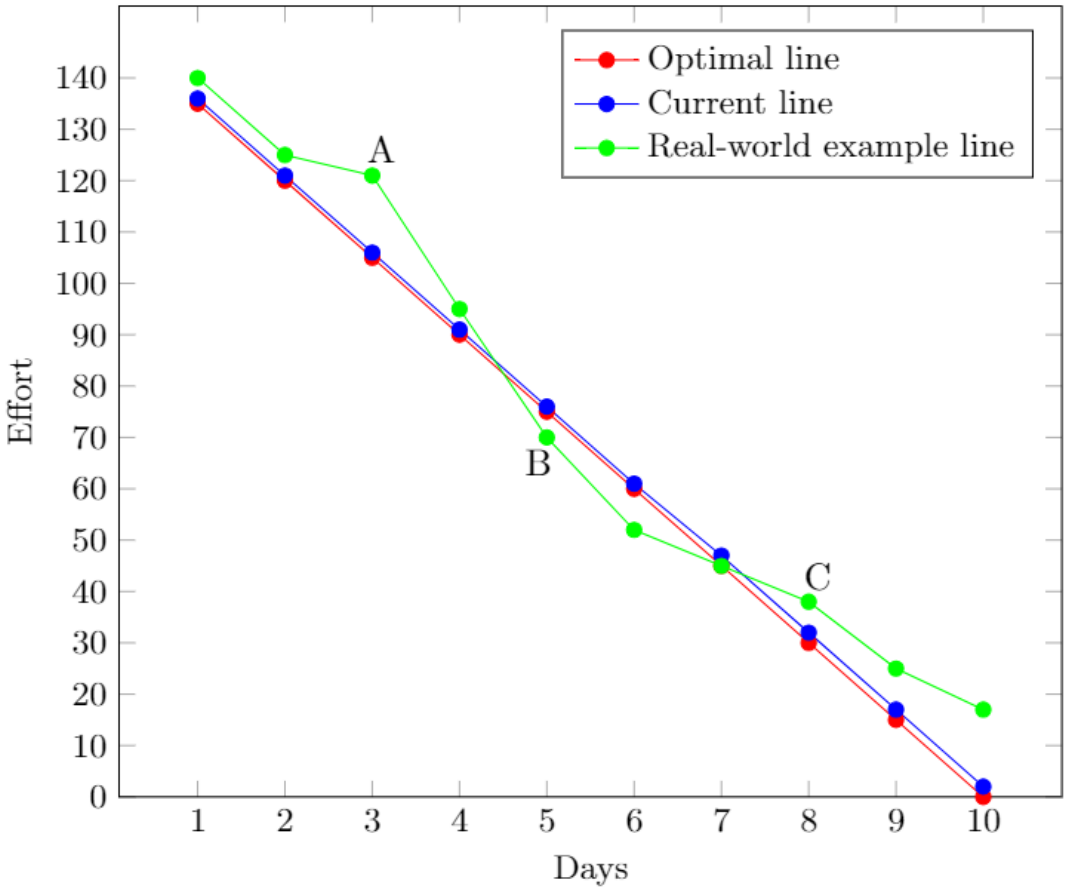
University of Maribor
Faculty of Electrical Engineering and Computer Science

Day	Tasks allocated	Tasks' effort	Number of tasks	Effort remaining
1	4, 12, 15, 17, 21, 32, 42	5, 3, 1, 1, 1, 2, 2	7	0
2	43, 27, 49, 48, 58, 33	3, 3, 3, 1, 1, 4	6	0
3	51, 7, 50, 5	1, 5, 4, 5	4	0
4	24, 26, 45, 35, 57, 54, 25	2, 1, 2, 2, 4, 2, 2	7	0
5	18, 10, 29, 16	2, 5, 3, 5	4	0
6	6, 22, 8, 53, 31	5, 4, 3, 1, 2	5	0
7	28, 44, 19, 0, 30, 3	4, 2, 1, 2, 4, 2	6	0
8	1, 14, 20, 37, 40, 52, 23, 38	2, 2, 2, 1, 1, 3, 3, 1	8	0
9	56, 34, 41, 11, 2, 9, 13	2, 3, 1, 2, 1, 3, 3	7	0
10	36, 39, 46, 47, 55, 59	3, 2, 4, 2, 2, 2	6	0
Σ	**60**	**150**	**60**	**0**

Example of an optimal solution.

Results

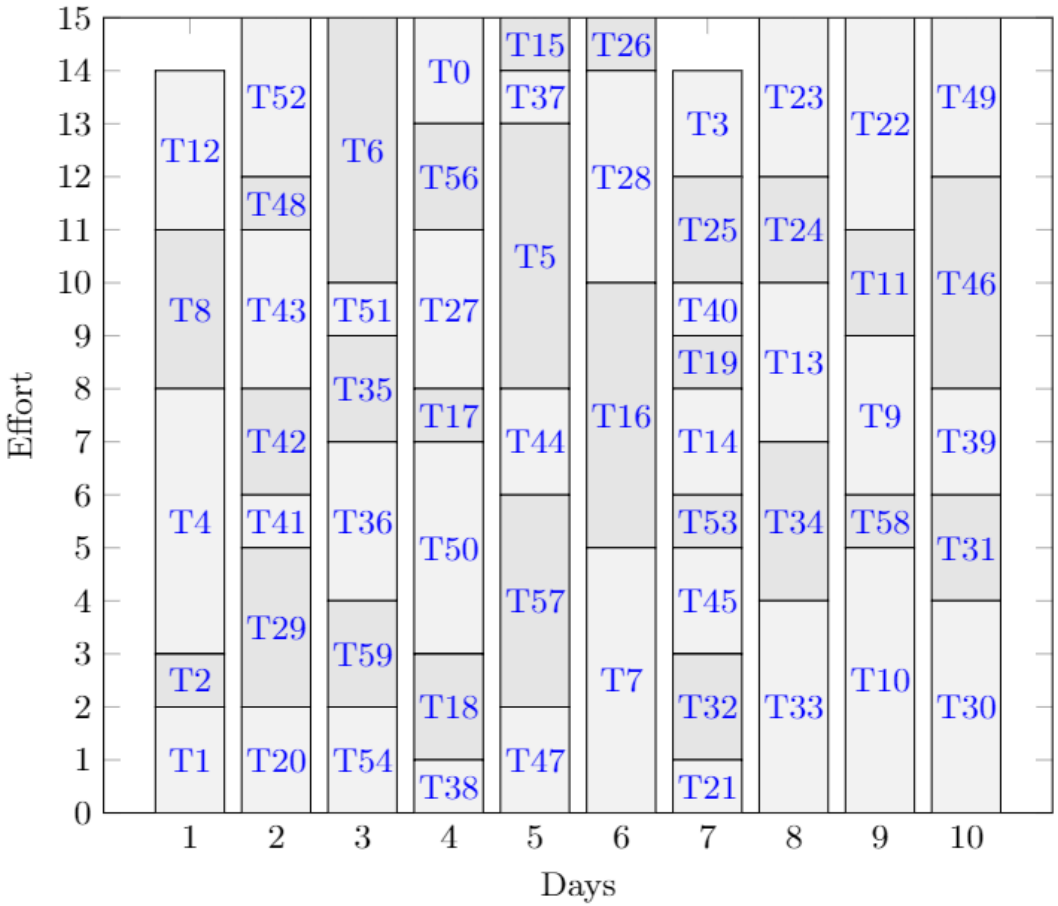
University of Maribor
Faculty of Electrical Engineering and Computer Science



Burndown chart of non-optimal solution.

Results

University of Maribor
Faculty of Electrical Engineering and Computer Science



Task allocation of non-optimal solution.

Conclusion

University of Maribor
Faculty of Electrical Engineering and Computer Science

- STAPSO algorithm:
  - offers a solution to the global problem of task allocation in the agile software development;
  - can be applied to all of the known estimation techniques, e.g. number sizing, Fibonacci sequence, and T-shirt planning;
  - it can be included in companies regardless of their size and maturity degree.

- Future plans:
  - Study the impact of various constraint handling methods.
  - Provide discrete variant of a PSO.
  - Hybridization of STAPSO with any other well-established algorithms.