Goal-management v1.0b
User Manual

Version 1.0

Introduction **3**
Why another Project Management System? 3
Goal Management Building Blocks 3
Goal properties: 4
Tasks properties: 5
Team members (Resources) assignment to tasks: 6
Team Members Definition: 6
Task Scheduling 7
Completion Dates 8

Introduction

Why another Project Management System?

The majority of software projects do not meet their deadlines, about two-thirds overrun their estimates by 25 to 50 percent. Reasons for such delays are:

- Milestones and deadlines are often set to optimistically or under pressure.
- "best case" estimate are assumed as deadlines.
- Projects are deliberately underestimated by management, sales or developers.
- Projects begin with realistic schedules, but new features are piled on to the project, and before long the project is running under an overly optimistic schedule.
- Project are simply estimated poorly.

A more reliable product would result if the deadlines were determined by the workload and the available resources. Furthermore, software projects do not end after installation. Maintenance, bug fixing and further development of the product are just as important as developing the first version. These tasks also need to be planned and monitored.

Goal management tries to address some of the issues responsible of delayed production. The focus of goal-management is on:

- **Estimation:** product size, man month. Presenting estimations, changing estimations
- **Scheduling:** After estimation is complete, the project has to be scheduled. Goals and tasks are set / prioritized and assigned to the team members.
- **Feature Set control / Change management:** The feature set of the product can change in all stages of the project. Cutting feature to save time. Implementing cheaper versions of features. Adding features as system is better understood. Adding features to stay competitive...
- **Maintenance:** bug reporting / fixing and further development of the product. Re-implementing cut features.

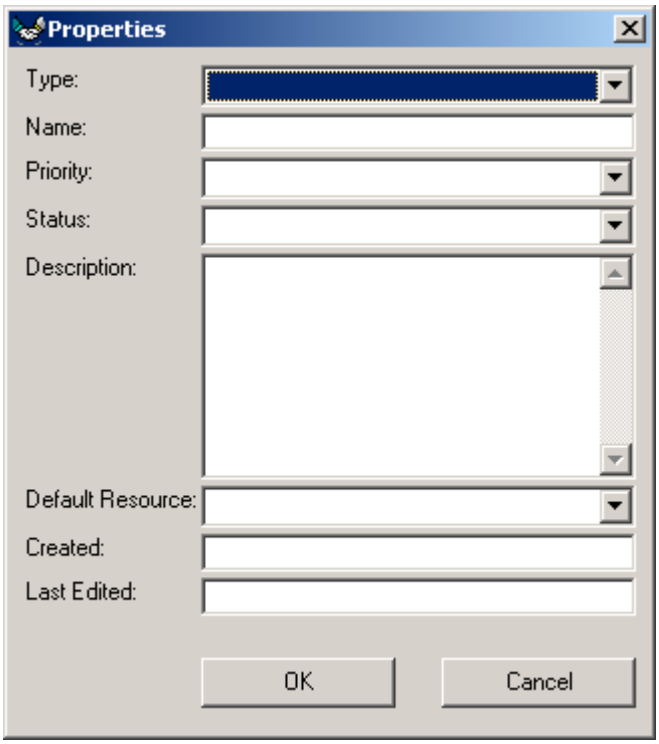
Goal Management Building Blocks

Goal-management contains two main building blocks; Goals/Sub-goals and Tasks. Using these building blocks the user can build a hierarchical, flexible project structure.

Each Project has one main goal. To allow a better overview, the main goal can be split into several sub-goals and these sub-goals can again be split into sub-goals.

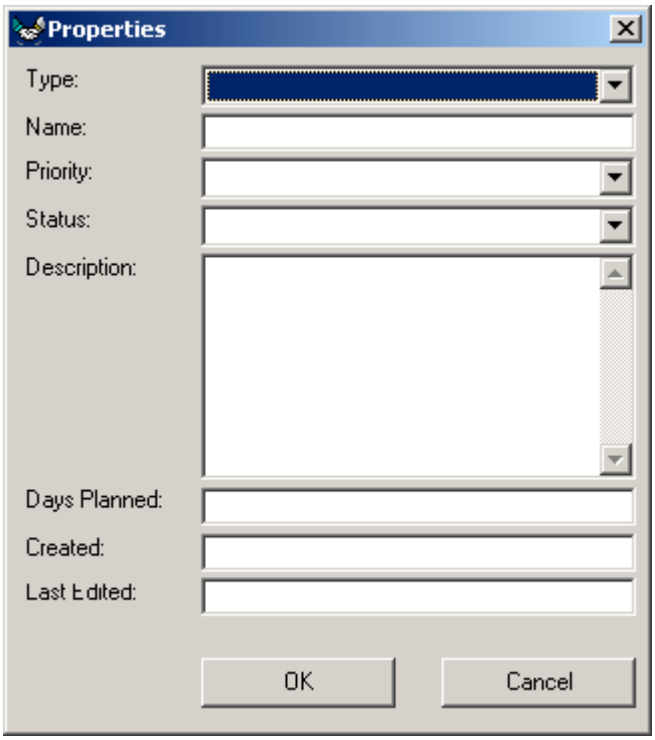
Tasks define the steps and workload needed to be completed to achieve a sub-goal. Tasks can be defined at all levels of the project hierarchy. The overall project workload is calculated by aggregating the individual task workloads.

Goal properties:



- Type:
  - Goal: The object is a goal
  - Application: The object is a software product
- Name: Full name of Goal /Application
- Priority: Between 0 and 99. 0..Highest priority, 99… lowest priority
- Status:
  - Completed: Goal / Application has been completed
  - Open: Goal / Application is currently being worked on
  - On Hold: Goal / Application is currently on hold. The estimated workload will not be considered in the calculations.
  - Closed: Goal / Application has been closed and will not be completed
- Description: Long description of Goal / Application
- Default Team Member (Resource): The default team member is automatically assigned to the tasks
- Created: Creation timestamp
- Last Edited: Last edited timestamp

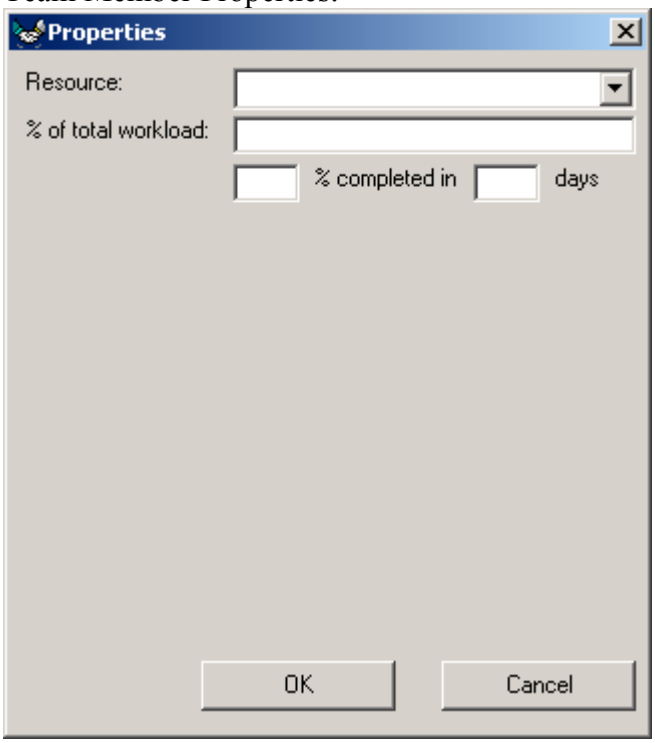
Tasks properties:



- Type:
  - Task: The object is a task
  - Software Object: The object is a software component. E.g. Table, Input Form, report, library…
- Name: Short name of object
- Priority: Between 0 and 99. 0..Highest priority, 99… lowest priority
- Status:
  - Completed: Task / Software component has been completed
  - Open: Task / Software component is currently being worked on
  - On Hold: Task / Software component is currently on hold. The estimated workload will not be considered in the calculations.
  - Closed: Task / Software component has been closed and will not be completed
- Description: Long description of Task / Software component
- Days Planned: Days planned to complete Task / Software component. (1 Day ⇔ 8 working hours)
- Created: Creation timestamp
- Last Edited: Last edited timestamp

Team members (Resources) assignment to tasks:

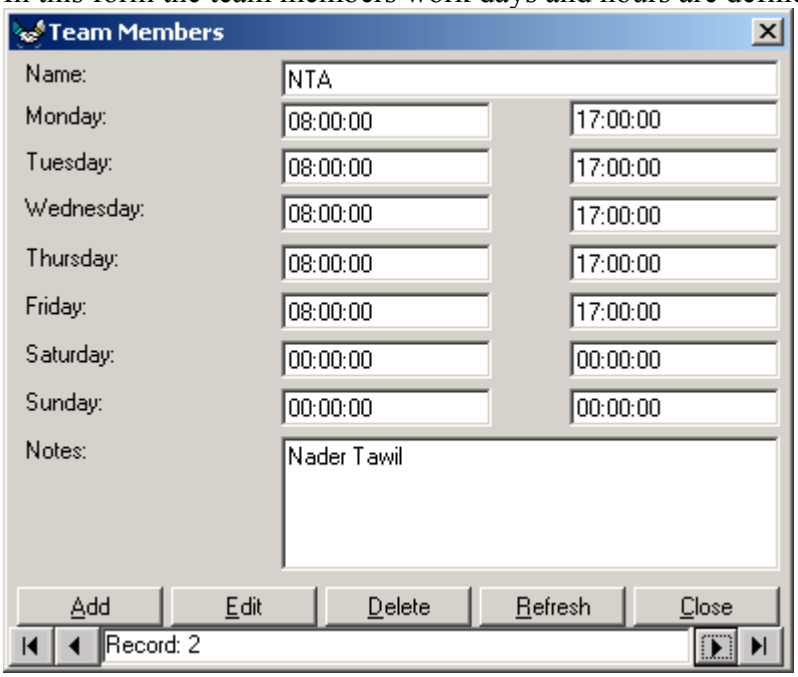
perform the actual tasks. Each task can be completed by one or more team members.
Team Member Properties:



- % of total workload: % of total workload assigned to team mamber
- % completed: % of assigned workload completed
- days needed: number of days needed by team member to complete the completed percentage

Team Members Definition:

In this form the team members work days and hours are defined.



For each member the start and end time for each day are defined.

Task Scheduling

One of the main features of goal-management is the scheduling of the individual tasks. The schedules are dependant on the priorities set and the project hierarchy.

The schedules are created top down through the project tree structure. Tasks with higher priority are scheduled before tasks with lower priority. Tasks with the same priority are scheduled together. Scheduling of goal–goal or goal–task combinations having the same priorities are resolved further down the tree at task–task level.

The scheduling algorithm is best explained using examples.

Example 1: Simple Schedule

<table>
<tr><td>GOAL 1 (Priority 20)</td><td colspan="2"></td></tr>
<tr><td></td><td>Task 1.1 (Priority 10)
Task 1.2 (Priority 60)</td><td></td></tr>
<tr><td>Task 0 (Priority 30)</td><td colspan="2"></td></tr>
<tr><td>Goal 2 (Priority 50)</td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2">Task 2.1 (Priority 30)</td></tr>
<tr><td></td><td>Goal 2.1 (Priority 50)</td><td></td></tr>
<tr><td></td><td></td><td>Task 2.1.1 (Priority 20)
Task 2.1.2 (Priority 20)</td></tr>
<tr><td></td><td colspan="2">Task 2.2 (Priority 60)</td></tr>
</table>

Step 1: Schedule level 0

1. **Goal 1 (20)**
2. Task 0 (30)
3. **Goal 2 (50)**

Step 2: Schedule Level 1 (expand Goal 1 and Goal 2)

1. Task 1.1 (20, 10)
2. Task 1.2 (20, 60)
3. Task 0 (30)
4. Task 2.1 (50, 30)
5. **Goal 2.1 (50, 50)**
6. Task 2.2 (50, 60)

Step 3: Schedule level 3 (expand Goal 2.1)

1. Task 1.1 (20, 10)
2. Task 1.2 (20, 60)
3. Task 0 (30)
4. Task 2.1 (50, 30)
5. Task 2.1.1 (50, 50, 20)
   Task 2.1.2 (50, 50, 20)
6. Task 2.2 (50, 60)

The algorithm completes when all tasks have been scheduled. Tasks 2.1.1 and 2.1.1 have the same priority (20) and are therefore scheduled together.

Example 2: Goal-Task and Goal-Goal combinations with same priority

GOAL 1 (Priority 20)		
	Task 1.1 (Priority 10) *Task 1.2 (Priority 60)*	
Task 0 (Priority 20)		
Goal 2 (Priority 20)		
	Task 2.1 (Priority 30)	
	Goal 2.1 (Priority 20)	
		Task 2.1.1 (Priority 20) *Task 2.1.2 (Priority 20)*
	Task 2.2 (Priority 60)	

Step 1: Schedule level 0

All three objects at level 0 have the same priority, so the schedule is resolved at a lower level in the tree. Task 0 is compared to the tasks and sub-goals of Goal 1 and Goal 2.

1. Task 1.1 (20, 10)
2. Task 0 (20, 20)
   Goal 2.1 (20, 20)
3. Task 2.1 (20, 30)
4. Task 1.2 (20, 60)
   Task 2.2 (20, 60)

Step 2: Resolve Task 0 and Goal 1.2

1. Task 1.1 (20, 10)
2. Task 0 (20, 20, 20)
   Task 2.1.1 (20, 20, 20)
   Task 2.1.2 (20, 20, 20)
3. Task 2.1 (20, 30)
4. Task 1.2 (20, 60)
   Task 2.2 (20, 60)

Comparing the two schedules we can see how powerful such a hierarchical project structure can be. All we did was change the priorities of three objects (Task 0, Goal 2 and Goal 2.1) and the result was a very different schedule.

Completion Dates

Another main features of goal-management is the computation of completion dates. Unlike most other systems, the completion dates cannot be set. They are calculated and updated automatically and are dependant on the work load estimation, the task priorities and the team members working hours.

Once the schedules have been calculated, the system can then compute the probable completion dates. Knowing how much percentage of each task is still not completed and the number of days planned for each task the system calculates new estimates for the tasks. Then using the defined work hours (Monday to Sunday), the completion date of each task and goal is computed. It is very important to keep in mind that a workday is defined as 8 hours in the system. So if a team member works four hours per day, then a task estimated with one workdays will take two weekdays to complete.