

Content

Introduction.....	1
The main concept.....	1
How to start.....	2
System sections.....	2
Security and access rights.....	2
Contacts.....	3
Projects.....	4
Tasks.....	4
Issues.....	4
Notes.....	4
Time Board and Time tracks.....	4

Introduction

PushOk SEMA is primarily developed for small companies or working groups with centralized hierarchical management. This is not group collaboration software, this is group management software. SEMA allows to manage employers, tasks assigned to them, issues that can arise during work. Also it allows to track time for all activities and maintain virtual time board. All above allows to effectively manage small working group, control and maintain group productivity.

Primarily SEMA designed for IT business, however data logic model allows to extent it usage to any other business that requires management of personal efforts. This can be travel agency, service company and so on.

The main concept.

Our experience says that any group management system, even it is claimed very effective and simple, can be found not usable for particular group because of time which will be required to input data. This is especially important for small group, where there usually just not possible to spend lot of time for supporting tasks (e.g. write large specs and so on). So we design the system as much as possible minimalistic during normal work. Most tasks that should make normal employer are in one click range from the main (home page) of system. Home page shows all important for specific users objects: active tasks, issues, gant graph for this week, active projects and so on. Normally there no necessity to make anything outside this home page. They user (developer) can view his tasks there, add issues, book his time and so on. If system not shows some objects that for some reason are in user interest range he can add such object into favorite list and they appears on home page.

Of course the system does not allows unauthorized access to system. Each user should be register in the system and require to provide his login and password to login. SEMA maintain access right list. The implementation is the mix from UNIX and Windows access control schemes. Each object can have list of users who have access to it. Each user can have individual rights to object. All sub objects of specific object inherit security descriptor of parent object. Usually just enough to adjust access right for project and do not touch any sub objects (tasks, issues and so on). For simplicity on project level this possible just sorting out each roles (manager, developer, tester and so on) have user on project. During implementation of access control system we take into account the fact that small groups usually make something for customers (external for this group). Sometime customers take place in project development, providing some comments and so on. Of course they not need to see all intrinsic activity (small internal tasks, internal issues and so on). In the same time low level developers may not need to know some details of communications between customer and manager. Because of that all object have additional flags, which shows for internal (I) or for external (E) users they available. In the same time each user by default either internal (developer) either external for group (customer).

Another thing, that was in mind during SEMA design is the notification system. It is very hard to use group management system if it cannot notify you when something in your interest range is changed (new task for you, or changes in tasks etc). SEMA will notify you about all, you can be interested in by e-mail.

How to start

The startup task is complex enough, since it requires to setup some system properties like user list, access rights and so on. This can be hard to do for new system when there are no ideas why this is needed and how this will be used later. So, in few steps we will try to explain how to start system after installation.

After successful installation you will have the empty system with only one user (root/root). So the first step is to create users who will work in your group. This can be done on «Contacts» section. Clicking on link «New contact» you can add new employee. The main things you should provide there is the user name, login, password and e-mail. Another important field is user flags. This field allows to mark user as (I)nternal or (E)xternal for your group or as (R)oot user. User can be either internal or external. This flag just means which flag will be assigned to objects created by this user. All group employees should have the 'I' flag while customers have the 'E'. After creation of users you should assign the default rights in system for newly created users. Default right is the right of creation of top level objects (projects) or right for objects created for users. These «Global rights» can be configured clicking on user and then clicking on «Global rights» link. Global rights assigned to each type of objects known by system. In details meanings of right flags will be explained later. Just now it will be enough to know that managers should have the 'RW12345IE' rights, developers 'RW123I' right, customers 'RW12E' rights. Without assigning the global rights users (managers, developers etc) will not be able to use system.

Then, when users are created and global right assigned you can logout as user root and login as any user who has manager rights. As in any system user root is an administrator rather than normal user and it is a bad idea to assign such right to any normal user. As manager you should create your first project. SEMA is a project-centric system and all management is around project. You, as the creator of project will have right to access it, however all others cannot see new project. So after creating project you should assign right to it. This can be done through assigning roles to project clicking on «Roles» link. All who will have role on project will have at least «read» right to project and «action rights» depending of his role. It is a good idea to include yourself as project manager into roles. This does not affect your access right but marks you as «project manager». Person who has «project manager» role is used to assign newly created issues and so on.

Well, that's almost all. Now you can add new task, view gantt task and so on. Developers can book time for task, mark his time in time board, create issues and so on. I.e. system is in live stage.

System sections

Security and access rights

SEMA maintains access right list. The implementation is the mix from UNIX and Windows access control schemes. Each object can have list of users who have access to it. Each user can have individual rights to object. All sub-objects of specific object inherit security descriptor of parent object. Usually just enough to adjust access right for project and do not touch any sub-objects (tasks, issues and so on). For simplicity on project level this is possible just sorting out each role (manager, developer, tester and so on) have user on project. During implementation of access control system we take into account the fact that small groups usually make something for customers (external for this group). Sometime customers take place in project development, providing some comments and so on. Of course they do not need to see all intrinsic activity (small internal tasks, internal issues and so

on). In the same time low level developers may not need to know some details of communications between customer and manager. Because of that all object have additional flags, which shows for internal (I) or for external (E) users they available. In the same time each user by default either internal (developer) either external for group (customer).

Summarizing all above.

1. Each user have the 'Ownership' flags. Flags that mark for which users it available. These flags is the combination of (I)nternal and (E)xternal. When object created it inherit flag from user flags field. I.e. internal users always created internal objects and external correspondingly.
2. Each object have the security descriptor, which can be inherited from parent object. Security descriptor contain one record for each user who have any access rights to object. Record can contain combination of flags from {RW12345IE}. The 'R' flag mean that user can read the object (view). 'W' that he can change his direct fields or create new one. The '12345' is the action flags, specific for each object. 'I' means that user can see object with (I)nternal ownership 'E' that with (E)xternal ownership. Then brief meaning of action flags are shown in table below. Note please that usually enough to setup roles on project and not change the direct permissions.

	<i>Common</i>	<i>Project</i>	<i>Task</i>	<i>Issue</i>	<i>Time board</i>
R {read}	View object and history				
W {write}	Modify or create object			Without status and responsible	
1 {observer}	Add issues, notes				
2 {worker}	Add time tracks, and time board		Can mark «Deed»		
3 {power worker}			Create subtasks Can alter	Add comments	
4 {manager}		Create tasks	Create subtasks Can alter	Change status, and responsible	Can change
5 {god}	Change rights	Change roles			

Contacts

Contact or user is the main actor of the system. Each user can have some personal attributes such as e-mails, phones and so one. They are not so important for system. What are important for system is the notification policy and preferences.

Notifications in SEMA are sent on specific events. Users not able to know when and what event can arises. However each even have two important attributes which used to route events to users. First is 'Importance'. There are 3 groups there: critical, warning, notification. This is self explanatory. Another attribute is Interest or Personalization. There 3 values also: personal, group, all. Now let's try to explain this. The explaining of the importance is simple. Task creation (or any other object creation) is critical. Changing of some attributes of object is warning. Changing of task progress in notification. The Interest group is slightly complex. When task created the personal interest group is task owner and worker. Both must to know that task created. All other members of project to which tasks belongs is in the group interest. They personally not need to know about this task. But as project members may want to know what new tasks appears on project. And all others are in the all others interest group. By default users will receive critical personal notifications. All other they can configure.

Preferences allows to select interface language, fast links and other user related things.

Projects

Project is very simple object. This is mostly the container for sub objects: tasks, issues, notes and so on. Project is also usually used as parent object for security descriptor.

Tasks

Task is the object which allows to describe piece of work and track its execution. Task should have the theme, description, worker user and responsible user (who controls execution). All other fields are optional.

Each task can be either free or fixed. Free does not have any time limits, and is not shown on a Gantt diagram. The free task is some day-to-day work, like answering support requests and so on. The fixed tasks have the start day, end date and work amount. This allows to control the progress of task execution.

Once written task description cannot be changed. It can be extended by new instructions. This new instruction can even cancel the previous part, but it still remains in the task. This is mostly to insure that the manager will not change the task and not say later that the worker made something that he did not ask for.

Task can have subtasks if needed.

The users who have the developers right to tasks cannot change it, they even cannot change the status of the task. Instead they can only move the task forward in the processing chain when they think that the work is done. This actually means that a new responsible is assigned to the task, usually this is the task owner, but it is also possible to set up another processing task. The task owner can accept the task or return it back to the developer.

Issues

This is a simple object. It allows to mark any problem specific to any other object. For example the bug on some project, adjustment for some task and so on. Issue has the responsible for it, status and worker who should solve the issue. Issue is not the task, it is not required to make something.

Notes

Simplest object, can be attached to any other object and contain anything.

Time Board and Time tracks

Time board and time tracks allow to control how much time the user spent on work, and how much time he booked during work. The time booked for task also allows to identify how effectively he works (difference between planned and real booked time).

To use the time board functionality the developer should just press «Mark my arrival» when coming to work, and «Mark TB from» when leaving work. To book time that he spends on issue, task or project it is enough to press «+time» link. User will need to put number of hours and brief explanation of what he is doing.

The results of the described activities are accessible on the time board page. Selecting the specific user page will show statistics such as: base time, booked time, result time, time track's time and so on. This information can be used to value the work of specific users.

The time tracks and tasks information are available also at the Gantt task page. Hint, to view all visible objects it is possible to use this link: sema.php?page=/web/task/task_gant.php&idparent=0.