

APSP¹

Abraham James², Christopher Rebstock³, Kareemullah Quadir⁴,
Roshan Rammohan⁵, Vinod Doshi⁶, Yaroslav Halchenko⁷
<http://apsp.onerussian.com>

March 8, 2002

¹Automated Personal Software Process

²ajames@cs.unm.edu

³chrisr@cs.unm.edu

⁴quadir@cs.unm.edu

⁵roshan@cs.unm.edu

⁶vdoshi@cs.unm.edu

⁷yoh@cs.unm.edu

Contents

- 1 Introduction** **3**

- 2 Requirements** **4**
 - 2.1 Use Cases 5
 - 2.1.1 Use Case “**UserConfig**”: Configuration of the System for the User 5
 - 2.1.2 Use Case “**AdminConfig**”: Global System Installation and Configuration 6
 - 2.1.3 Use Case “**SystemStartUp**”: Startup and Initialization of System 7
 - 2.1.4 Use Case “**RegisterUser**”: Register a new user 8
 - 2.1.5 Use Case “**UserAuth**”: User authentication (Login) 9
 - 2.1.6 Use Case “**LogOut**”: User Logs Out 10
 - 2.1.7 Use Case “**DeleteAccount**”: Delete user account 11
 - 2.1.8 Use Case “**ChangePassword**”: Change of User’s Password 12
 - 2.1.9 Use Case “**ForgotPassword**”: User forgot his password 14
 - 2.1.10 Use Case “**SwitchPhase**”: User switches between project phases. 15
 - 2.1.11 Use Case “**AddEditPhaseData**”: Add/Edit project phase data. 16
 - 2.1.12 Use Case “**SubmitDefects**”: User submits Defect recording Log (DRL) 17
 - 2.1.13 Use Case “**FillPPS**” : User adds data to the Project Plan Summary (PPS) form. 19
 - 2.1.14 Use Case “**FillPIP**” : User submits the Process Improvement Proposal (PIP) form. 21
 - 2.1.15 Use Case “**ViewStat**”: User views Graphs. 22

- 3 User Interface** **24**
 - 3.0.16 Login 25
 - 3.0.17 Register User 26
 - 3.0.18 Delete Account 27
 - 3.0.19 Change Password 28
 - 3.0.20 Forgot Password 29
 - 3.0.21 Main Window 30
 - 3.0.22 DRL form 32
 - 3.0.23 PPS form 35

3.0.24 PIP form	37
3.0.25 Graphs	39

Chapter 1

Introduction

In this paper we presented user interface for our project and also included use cases from requirements part in case when user interface cites them.

Some images appear at the end of document instead of appearing after text where it was cited. We just couldn't resolve this problem in latex, sorry for inconvenience.

Chapter 2

Requirements

2.1 Use Cases

2.1.1 Use Case “UserConfig”: Configuration of the System for the User

Goal in Context

System is properly configured for the user

Preconditions

1. System is installed on the computer by Administrator

Success End Conditions

System is properly configured

Failed End Conditions

System is not properly configured

Primary Actor

User

Trigger

User decides to configure installed system

User runs system without proper configuration file present, so system switches to this scenario.

Description (Main Success Scenario)

1. System prompts to fill necessary fields
2. User sets at least one path to a project’s data
3. User sets the place (work, home etc)
4. User (optionally) specifies the refresh time

Variations

- Configuration for the user can be specified by modifying configuration file, so system wouldn’t be involved in this process.

2.1.2 Use Case “AdminConfig”: Global System Installation and Configuration

Goal in Context

System is properly configured

Success End Conditions

System is installed and properly configured

Failed End Conditions

System is not properly configured

Primary Actor

Administrator

Trigger

Administrator decides to install and configure system

Description (Main Success Scenario)

1. Administrator installs system
2. Administrator initiates configuration process
3. System prompts Administrator to fill-in necessary data
4. Administrator specifies paths for system

2.1.3 Use Case “SystemStartUp”: Startup and Initialization of System

Goal in Context

System is initialized, the path(s) to projects and refresh time (time interval between checking for changes in project files) are set.

Preconditions

1. User has system properly installed.
2. Configuration file was properly created.

Success End Conditions

System starts successfully and begins authentication.

Failed End Conditions

System did not start.

Primary Actor

User

Trigger

User executes system. System is executed from automated login script.

Description (Main Success Scenario)

1. User executes system.
2. System verifies correctness of configuration file
3. System begins authentication. (use case “UseCase”)

Extensions

- 1a The system is not installed properly and gives error upon execution.
- 1b The configuration file is not found. The system informs the user that the configuration file was not found and switches to use case “UserConfig”
- 1c A path is not set. The system informs the user that a path must be set.
- 1d The refresh time is not set. The system informs the user that the default value will be used

2.1.4 Use Case “RegisterUser”: Register a new user

Goal in Context

Register a new user to give him access to the system.

Preconditions

Success End Conditions

User is registered, access granted

Failed End Conditions

User wasn't registered.

Primary Actor

Administrator

Trigger

Administrator receives request to register a user

Description (Main Success Scenario)

1. Administrator issues a command to register a user
2. System asks for personal user information: login name and email address.
3. Administrator enters the required information.
4. System registers user and generates random password.
5. System informs user through email that he was registered and sends his login name and password.

Extensions

- 4a A valid email wasn't entered. The user is prompted to re-enter a valid email. Email validation also possible.
- 4b User name already exists in the system. The user is prompted to enter another one.
- 5a System will require change of user's randomly generated password after first login.

2.1.5 Use Case “UserAuth”: User authentication (Login)

Note

This use case is separate from “SystemStartUp” use case because system can be already running but with no user currently logged in (after use case “LogOut” for instance).

Goal in Context

Authenticate user so as to protect the user’s data.

Preconditions

User has a valid account with system.

Success End Conditions

User is authenticated and system begin logging in data.

Failed End Conditions

User authentication fails.

Primary Actor

User

Trigger

The user initiates request for authentication.

Description (Main Success Scenario)

1. The system prompts the user for username and password.
2. The system validates the entered information.
3. The user is given access to the system and the data.

Extensions

- 1a The username and/or password is incorrect, the user is prompted to re-enter the information.

2.1.6 Use Case “LogOut”: User Logs Out

Goal in Context

Stop all logging for the user.

Preconditions

User is currently logged in

Success End Conditions

Logging is stopped and user’s status is set to logged out.

Failed End Conditions

- System crashes and begins logging erroneous data.
- System crashes and is no longer able to collect data.

Primary Actor

User

Trigger

User issues command to log out.

Description (Main Success Scenario)

1. System logs user out and stops all logging.

2.1.7 Use Case “DeleteAccount”: Delete user account

Goal in Context

Allow administrator to be delete user account.

Preconditions

User account exists.

Success End Conditions

Terminated the account. System deletes the account and denies access.

Failed End Conditions

Account wasn't deleted.

Primary Actor

Administrator.

Trigger

Administrator issues command to delete user account.

Description (Main Success Scenario)

1. Administrator logs in to the system.
2. Administrator deletes specified user account.

Extensions

- 1a No valid user account for specified login name.

2.1.8 Use Case “ChangePassword”: Change of User’s Password

Goal in Context

Allow user to changes his password.

Preconditions

1. User has valid account.
2. User knows his current password.
3. User is currently logged in.

Success End Conditions

User changed his password to a new one, which he specified.

Failed End Conditions

Password wasn’t changed.

Primary Actor

User

Trigger

User issues command change password.

Description (Main Success Scenario)

1. User issues command change password.
2. System asks old password.
3. User enters old password.
4. System asks new password.
5. User enters new password.
6. System asks user to re-enter new password.
7. User re-enters new password.
8. System changes user’s password to the new one.

Extensions

- 3a Typed in old password is wrong, system informs the user and goes back to the step 2.
- 3b Typed in new password differs from the typed in password in verification field. System informs the user and goes back to the step 2.

Variations

- User may cancel transaction at any time before step 4, so passwords wouldn't change

2.1.9 Use Case “ForgotPassword”: User forgot his password

Goal in Context

Give possibility to provide the user with a new password for his account, trying to be secure enough.

Preconditions

1. User has a valid account with system.
2. User doesn't know his current password and hence cannot login to the system.
3. User provided a valid email address when account was created.

Success End Conditions

User gets new temporary valid password to his account.

Failed End Conditions

User did not obtain a new password.

Primary Actor

User

Trigger

User initiates request for a new password.

Description (Main Success Scenario)

1. User provides the system with login name.
2. System changes password for that account to a new, randomly generated one.
3. System sends the new password to the specified email address

Extensions

- 2a If entered email address doesn't correspond to his login name - system informs the user and returns to the step 1.
- 4a E-mail address expired and no longer valid, so user's password can't be changed this way - failed end condition.

Variations

- 5a System asks user to change his new password to another one as soon as logs into the system. (Previous Use Case: Change of the Password.)

2.1.10 Use Case “SwitchPhase”: User switches between project phases.

Goal in Context

User wishes to switch from one project phase to another.

Preconditions

User is currently logged in.

Success End Conditions

1. The change of phase is indicated to the user.
2. The system starts recording data for the new phase.

Failed End Conditions

The system does not switch to the new phase.

Primary Actor

User

Trigger

User gives a command to change to another phase.

Description (Main Success Scenario)

1. User gives a command to change to another phase.
2. System changes the phase.
3. System indicates the change of phase to the user and starts recording data for the new phase.

Extensions

None.

2.1.11 Use Case “AddEditPhaseData”: Add/Edit project phase data.

Goal in Context

User wishes to add/edit existing information of a particular phase.

Preconditions

User is currently logged in.

Success End Conditions

The user added/edited project phase data.

Failed End Conditions

The user could not add/edit the data.

Primary Actor

User

Trigger

User gives a command to add/edit data of a project phase.

Description (Main Success Scenario)

1. User gives a command to add/edit data of a project phase.
2. System opens the project phase.
3. User adds/edits data for that phase and submits it.
4. System updates the data for that phase successfully.

2.1.12 Use Case “SubmitDefects”: User submits Defect recording Log (DRL)

Comment

System will provide basic statistics of number of defects detected/solved during development. It will not be provide full capability of defect tracking system.

Goal in Context

To hold data on each defect as the user finds and corrects it.

Preconditions

User is currently logged in.

Success End Conditions

User submits the defects and system is updated successfully.

Failed End Conditions

Primary Actor

User

Trigger

User issues the command open DRL.

Description (Main Success Scenario)

1. User issues the command open DRL.
2. System opens a new DRL form.
3. User enters the following data for each defect:
 - Defect Type.
 - Phase in which he injected the defect.
 - Phase in which he removed the defect.
 - Time in minutes he took to fix the defect.
 - Brief description of the defect and the fix.
4. User enters as many defects as he wishes.
5. User submits the log.
6. System is updated successfully.

Variations

- 1a User tries to submit an incomplete log and system prompts an error message asking the user to complete the log.

2.1.13 Use Case “FillPPS” : User adds data to the Project Plan Summary (PPS) form.

Goal in Context

To hold the estimated and actual project data in a convenient and readily retrievable form.

Preconditions

1. User is logged in.

Success End Conditions

User successfully submits the PPS form and the System is updated.

Failed End Conditions

Primary Actor

User

Trigger

User gives the command 'open PPS form'.

Description (Main Success Scenario)

1. User issues command open PPS form.
2. System opens a new empty PPS form.
3. User enters the following data:
 - Estimated LOC/Hour for the current project.
 - Estimated Total LOC for the current project.
 - To Date Total LOC.
 - Estimated time taken for Planning phase for the current project.
 - Estimated time taken for Design phase for the current project.
 - Estimated time taken for Code phase for the current project.
 - Estimated time taken for Compile phase for the current project.
 - Estimated time taken for Test phase for the current project.
 - Estimated Defects Injected in Planning phase for the current project.
 - Estimated Defects Injected in Design phase for the current project.
 - Estimated Defects Injected in Code phase for the current project.
 - Estimated Defects Injected in Compile phase for the current project.
 - Estimated Defects Injected in Test phase for the current project.
 - Estimated Defects Removed in Planning phase for the current project.

- Estimated Defects Removed in Design phase for the current project.
 - Estimated Defects Removed in Code phase for the current project.
 - Estimated Defects Removed in Compile phase for the current project.
 - Estimated Defects Removed in Test phase for the current project.
4. User submits the PPS form.
 5. System is updated successfully.

Extensions

- 2a User tries to submit form with invalid data. System prompts an error message asking the user to reenter information.
- 2b User tries to submit an incomplete form. System prompts an error message asking the user to complete the form.

2.1.14 Use Case “FillPIP” : User submits the Process Improvement Proposal (PIP) form.

Goal in Context

To record process problems and improvement ideas by filling out the PIP form.

Preconditions

1. User is logged in.
2. User has completed the PPS form.

Success End Conditions

User submits the PIP form and system is updated successfully.

Failed End Conditions

Primary Actor

User

Trigger

User issues command to open PIP form.

Description (Main Success Scenario)

1. User gives command open PIP form.
2. System opens a new empty PIP form.
3. User enters the following data for each problem:
 - Problem description.
 - Proposal description.
4. User enters as many problems as he wishes.
5. User may enter notes and comments.
6. User submits the form.
7. System is updated successfully.

Extensions

- 2a User tries to submit an incomplete form. System prompts an error message asking the user to complete the form.

2.1.15 Use Case “ViewStat”: User views Graphs.

Goal in Context

User wants to choose and view the graphs on his progress.

Preconditions

1. User is currently logged in.
2. The system has collected statistical data in projects on his previous projects.

Success End Conditions

User is able to choose and view the desired graph.

Primary Actor

User.

Trigger

User issues a command to view graphs.

Description (Main Success Scenario)

1. User issues a command to view graphs.
2. System displays list of graphs namely:
 - Actual Size(LOC) vs Time.
 - Lines added vs Time.
 - Lines removed vs Time.
 - Lines modified vs Time.
 - Total number of defects up to this point vs Time.
 - Number of commits to CVS vs Time.
 - Number of updates from CVS vs Time.
 - Actual Size(LOC) vs Project.
 - Planning Time vs Project.
 - Test Time vs Project.
 - Size Estimating Error vs Project.
 - Time Estimating Error vs Project.
 - Actual Development Time vs Project.
 - Compile Time vs Project.
 - Productivity vs Project.

- Compile Defects vs Test Defects.
 - Defects removed in compile vs Project.
 - Defect Injection by Phase vs Project.
 - Defects Removed in Test vs Project.
 - Defects Injected in Design vs Project.
 - Defects Removed by Phase vs Project.
 - Defects Removed by Phase vs Project.
 - Total Defects by vs Project.
 - Defects Injected in Code vs Project.
 - Number of commits to CVS.
 - Number of updates from CVS.
3. User chooses the graph he wishes to view.
 4. System displays the previous projects and prompts the user to enter the project range he wishes to view the graph for.
 5. User enters the project range.
 6. System displays the graph.
 7. User goes back to step 1 or exits.

Extensions

None.

Variations

- 5a User need not specify project range - so graph for all previous projects will be shown.

Chapter 3

User Interface

3.0.16 Login

The Login Window has the following components in its interface: 3.1.

- System Message Line
The system shows the user error messages, if any.
- UserName text box
The User types in his Login name here.
- Password field
The user types his Password here.
- Login button
User clicks this to login.
- Cancel button
User clicks this to escape out of this window without logging in.

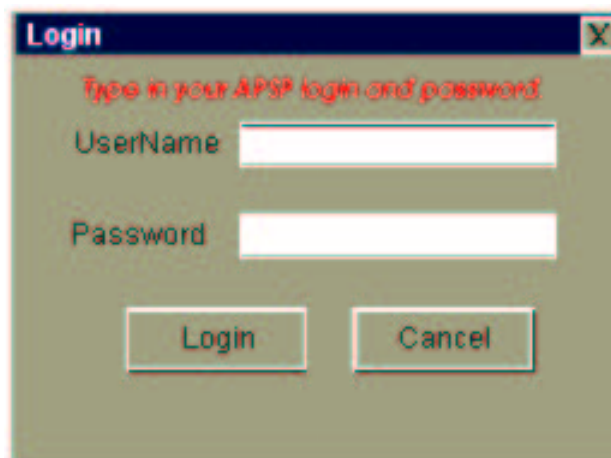


Figure 3.1: "Login window"

3.0.17 Register User

The Register User Window has the following components in its interface: 3.2.

- System Message Line
The system shows the user error messages, if any.
- UserName text box
The Administrator types in a User name here.
- Email address text box
The Administrator types in the new user's email address here.
- Register button
Administrator clicks this to login.
- Cancel button
Administrator clicks this to escape out of this window without registering user.

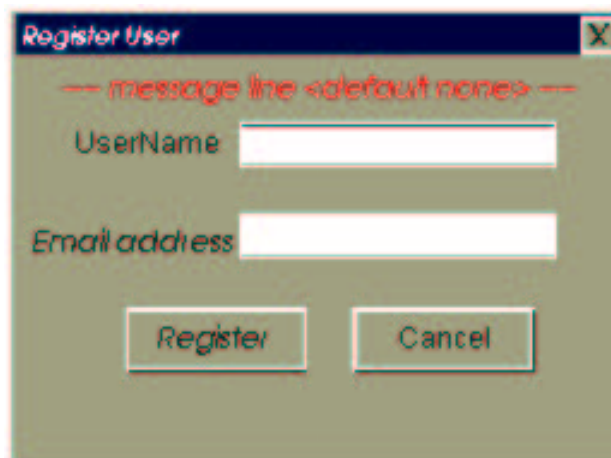


Figure 3.2: "Register User"

3.0.18 Delete Account

The Delete Account window has the following components in its interface, See Fig.3.3.

- System Message Line
The system shows the user error messages, if any.
- UserName text box
The Administrator types in the username of the account he would like to delete.
- Delete button
Administrator clicks this to delete the account.
- Cancel button
Administrator clicks this to escape out of this window without deleting any account.

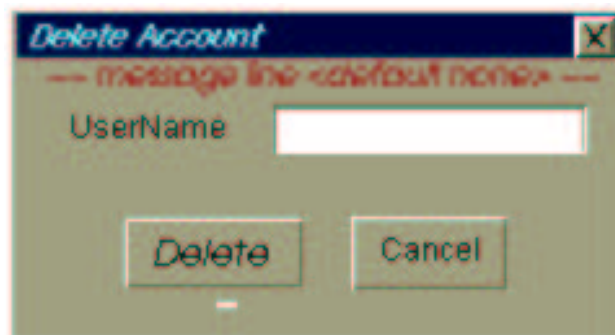


Figure 3.3: "Delete Account"

3.0.19 Change Password

The Change Password window has the following components in its interface: 3.4.

- System Message Line
The system shows the user error messages, if any.
- Old Password (password field)
The User types in old password for validation here.
- New Password (password field)
The user types his new password here.
- Re-enter Password (password field)
The user re-types his new password here.
- Change button
User clicks this to attempt to change his password.
- Cancel button
User clicks this to escape out of this window without changing password.

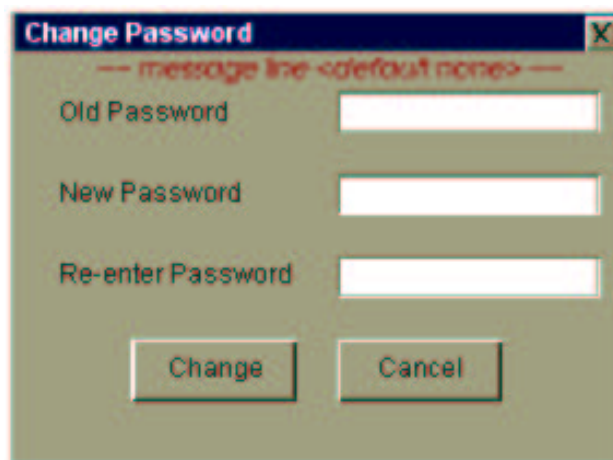


Figure 3.4: “Change Password”

3.0.20 Forgot Password

The Forgot Password window has the following components in its interface:

- System Message Line
The system shows the user error messages, if any.
- UserName text box
The User types his login name over here.
- OK button
User clicks this to ask the system to mail a password to his email id.
- Cancel button
User clicks this to escape out of this window without requesting password.

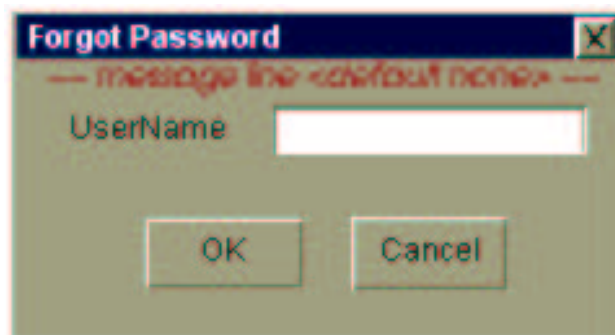


Figure 3.5: "Forgot Password"

3.0.21 Main Window

The window Fig.3.6 has a split screen view as a default when a user logs in. The bottom text window could be closed by clicking on the close button. The bottom window acts as a status(logging) window.

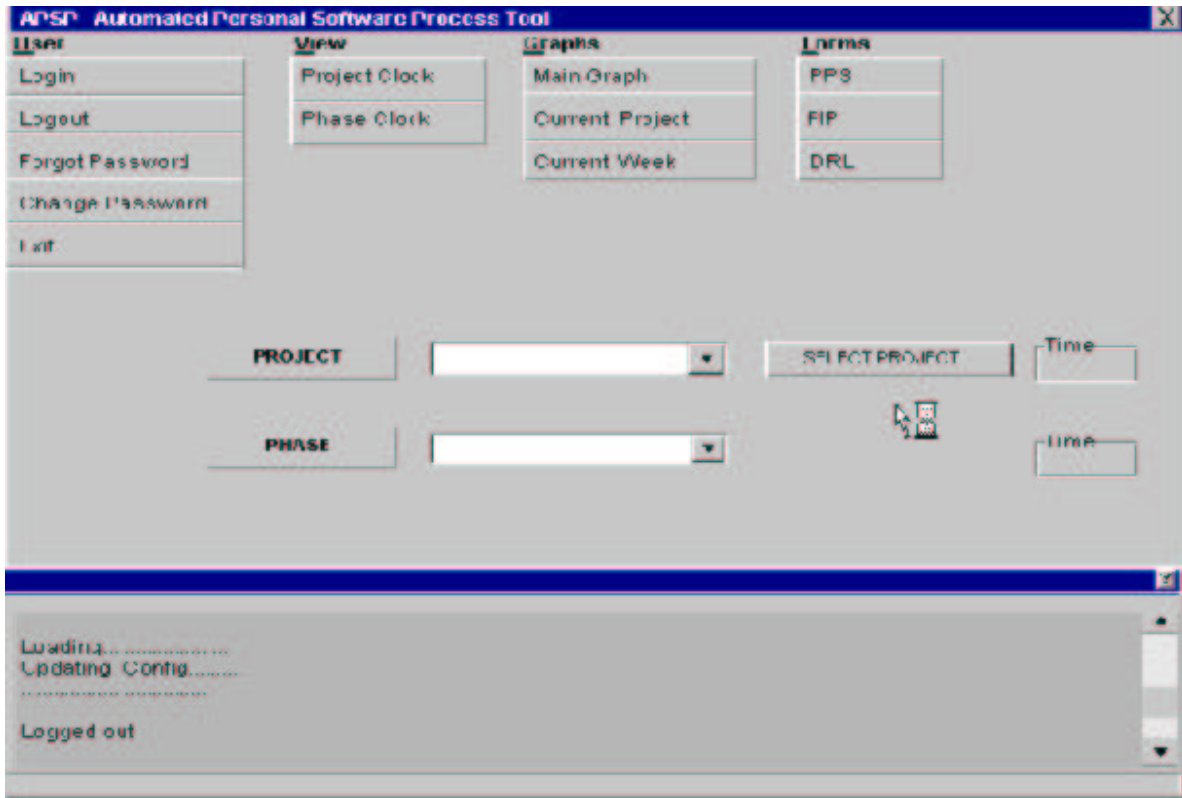


Figure 3.6: “Main window”

It basically shows the current status of the system.

The main window has a drop down menu bar at the top¹.

Root components of the menu bar are User, View, Graphs and Forms.

The drop down on the User has the following components:

- Login
This is in reference to the user authorization use case
- Logout
This action points to the logout use case
- *Forgot password*
This action points to the forgot password use case
- *Change Password*
This points to the change password use case

¹Menu will be also implemented as a pop-up window when clicked anywhere in Main Window with a right mouse click

- *Exit*

These are the actions that a user performs.

The drop down on the View has the following components:

- Project Clock
This displays the amount of time a user has spend on the current project
- Phase Clock
This displays the amount of time a uses has spend in the current phase

They are implemented as check boxes.

The drop down on the Graphs has the following components:

- Main Graph
This displays all the graphs associated with all the projects. So you can adjust them as you wish as is described in user interface “Graphs”
- Current Project Graph
This displays the graphs for currently selected project
- Graph for current week
This displays the graph for all the projects worked that particular week

This points to the “**ViewStat**” use case.

The drop down on the Forms has the following components:

- PPS
User completes this form to extract the estimated and actual project data in a retrievable form
- PIP
User completes this form to record process problems and improvement ideas
- DRL
User submits defect recording log and the system will in turn provide basic statistics on the number of defects detected and solved during the development phase.

The main part of the window has two list boxes for the user to choose a Project/Subdir/File and the phase that he or she wants to work on.

To the right of the project list box there is a select project button which when clicked opens up a window Fig.3.7 with list of projects with the directories subdirectories and the respective files in them

To the right of these list boxes are the respective timers for both the project and the phase the user is working on.

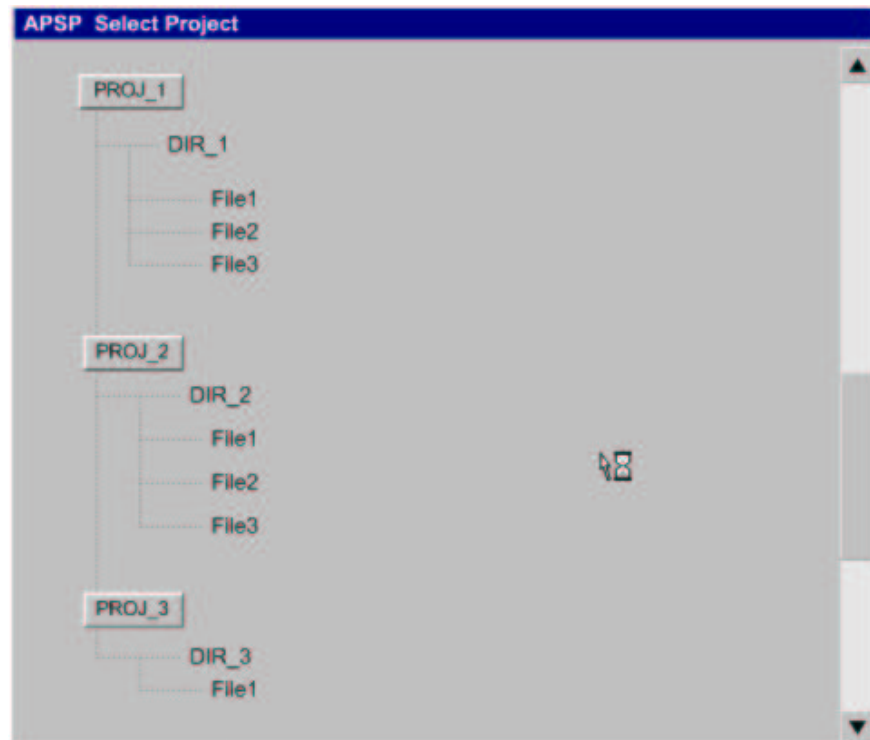


Figure 3.7: “Select Project”

3.0.22 DRL form

The DRL form Fig.3.8 has the following components in its interface:

- Title
- Date
- Defect type select box
This allows the user to select the type of defect he wants to record.
- Inject Phase select box
This allows the user to select the phase in which the defect was injected.
- Remove Phase select box
This allows the user to select the phase in which the defect was removed.
- Fix time text area
This allows the user to key in the number of minutes he took to fix the defect.
- Description text area
This allows the user to key a short description of the defect.
- Add Defect button
When the user clicks this button the information about the defect is saved.

- Clear button
When the user clicks this button all the input fields are cleared.
- Defect Log View Area
This is a scrollable pane where a list of all defects are displayed in chronological order.

<<Project Name>> - Defect Recording Log

Date <Current Date>

Defect Type Inject Phase

Remove Phase Fix Time Min

Description

No.	Date	Defect Type	Inject Phase	Remove Phase	Fix Time(mins)	Description

Figure 3.8: "Defect Recording Log"

3.0.23 PPS form

The PPS form Fig.3.9 has the following components in its interface:

- Title
- Columns : Parameter, Plan, Actual, ToDate and ToDate
- Parameters have the following values
 - LOC/hr **
 - LOC TOTAL **
 - Time in Planning Phase
 - Time in Desing Phase
 - Time in Code Phase
 - Time in Compile Phase
 - Time in Test Phase
 - Total time in all Phases *
 - Defects injected in Planning Phase
 - Defects injected in Desing Phase
 - Defects injected in Code Phase
 - Defects injected in Compile Phase
 - Defects injected in Test Phase
 - Total defects injected in all Phases *
 - Defects removed in Planning Phase
 - Defects removed in Desing Phase
 - Defects removed in Code Phase
 - Defects removed in Compile Phase
 - Defects removed in Test Phase
 - Total defects removed in all Phases *

The Plan Column for all these parameters will have an editable text field (except for paremeters marked with *) which will be pre-set to a value determined by the regression analysis algorithms of APSP. However, the user can edit this data. The user cannot modify the Plan column for parameters marked with *.

All other columns (Actual, ToDate and ToDate%) will display values automatically generated by APSP (for parameters marked with **). For parameters marked with ** the ToDate% column will be blank.

Buttons : The form will also have two buttons, SUBMIT and CANCEL at the bottom of the form. Clicking on SUBMIT saves all information in the form and exits. Clicking on CANCEL just closes the form without saving it.

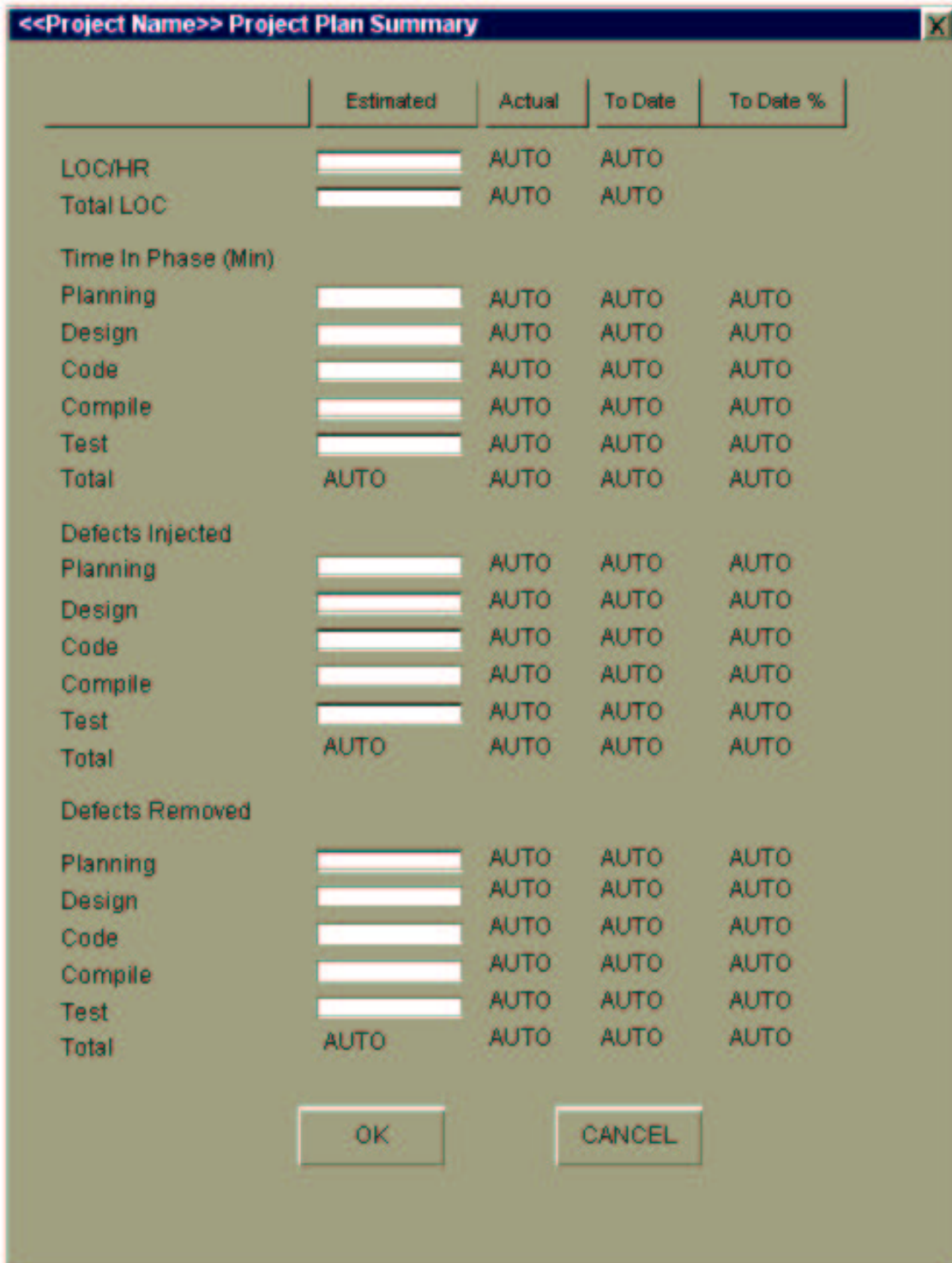


Figure 3.9: "Project Plan Summary"

3.0.24 PIP form

The PIP form Fig.3.10 has the following components in its interface:

- Title : Contains the name of the project.
- Text Fields : Problem, Proposal
The user can use these text areas to describe his problem and a proposal for that problem.
- Buttons : Add, Clear and Close.
Clicking on Add will save the text in the Problem and Proposal areas along with an automatically generated PIP number and added to PIP list.
Clicking on Clear will clear the text in the Problem and Proposal areas.
Clicking on Close will close this form.
- Display area
This area will contain a scrollable list of all the PIPs submitted by the user for this project.
- Notes and Comments text area
This area allows the user to type in notes and comments after he has added all the Process improvement proposals. Of course the user will be able to type in notes anytime, not necessarily at the end of the project, and save it. In this case this text area will display the previously stored text, but the user may edit this if he so wishes.

<<Project Name>> - Process Improvement Proposal

Problem

Proposal

ADD CLEAR CLOSE

PIP No.	Problem	Proposal
---------	---------	----------

Notes & Comments

Figure 3.10: "Process improvement proposal"

3.0.25 Graphs

Current UI form is an implementation for use case “ViewStat” (see 2.1.15). It provides capability to display information about user’s progress in two dimensions. In time and between projects. All further cites of names of the elements of UI form’s elements are connected to Fig. 3.12 and Fig. 3.11 (current data presented in graphs is just for visualization - it is based on random data, that is why total doesn’t sum up and data is in every cell).

This form will be implemented in HTML so to be able viewed with compatible web browser. That is why navigation and interaction of user with control elements mostly determined by a browser, so user can switch to the next element with mouse or using “TAB” key (“SHIFT-TAB” will switch to a previous one) in most of them but toggling and pressing control elements can differ. Also pressing “ENTER” key while being interacting with a form is the same as pressing button “UPDATE” on it (will not work in old version of Netscape for instance).

Following control elements are involved in interaction with a user and specifies parameters for display of graphs.

To update presented graphs in accordance with changed parameters, user has to press(click on) button “UPDATE”

- “In Time graphs” and “In Projects graphs” checkboxes toggle appearance of graphics. So on Fig. 3.12 we have “In Time” checkbox turned-on – graphical (colored bars) representation of data in time is presented. User also can see specific value if he moves mouse over the colored bar he is interested in - small hint (usually black text on yellow background) window will appear with name of the graph and value at current point.

On Fig. 3.11 “In Time graphs” is off – user is presented with current numbers. The same logic works for “In Project” representation of the data.

- Mutli-selectable boxes under “... graphs” checkboxes serve to select types of graphs you want to have graphs presented on the screen. Use case “ViewStat” list graphs which would be presented. Vertical bars correspond to the data vs. Time and horizontal bars corresponds to the graphs vs.Project².

Background of “DATEX” cell in a “In. Time” graph will provide user with information about a phase of the project he was working during that period of time. So there is a list of colors for different phases presented in upper-left corner of the figure. If during that period of time many phases were involved, then phase which was used most of the time during that period will be chosen to display³

- “From” and “To” date selection fields presented with usual drop-down select boxes separate for day, month and year⁴; so user can select a time range for which he wants to see statistics on his progress (days specified in “From” and “To” are included in that range).

Depending on selected time range each column (“DATE1”, “DATE2”...“DATEX”...) will zoom to represent cumulative results for next periods.

- Selected range > 1 year – each DATEX corresponds to a single year.
- Selected range > 1 month – each DATEX corresponds to a single month.
- Selected range > 10 days – each DATEX corresponds to 3 days.
- Selected range > 1 day – each DATEX corresponds to a single days.
- Selected range is a single day – each DATEX corresponds to 2 hours.
- If specified “From” field’s date is lies after “To” - no data is presented on the screen and no warning is presented.

- “Full Time Range” button: after pressing it user will be presented with report for whole time he was using this system, so from 1st till last registered entry in the system for his login name.

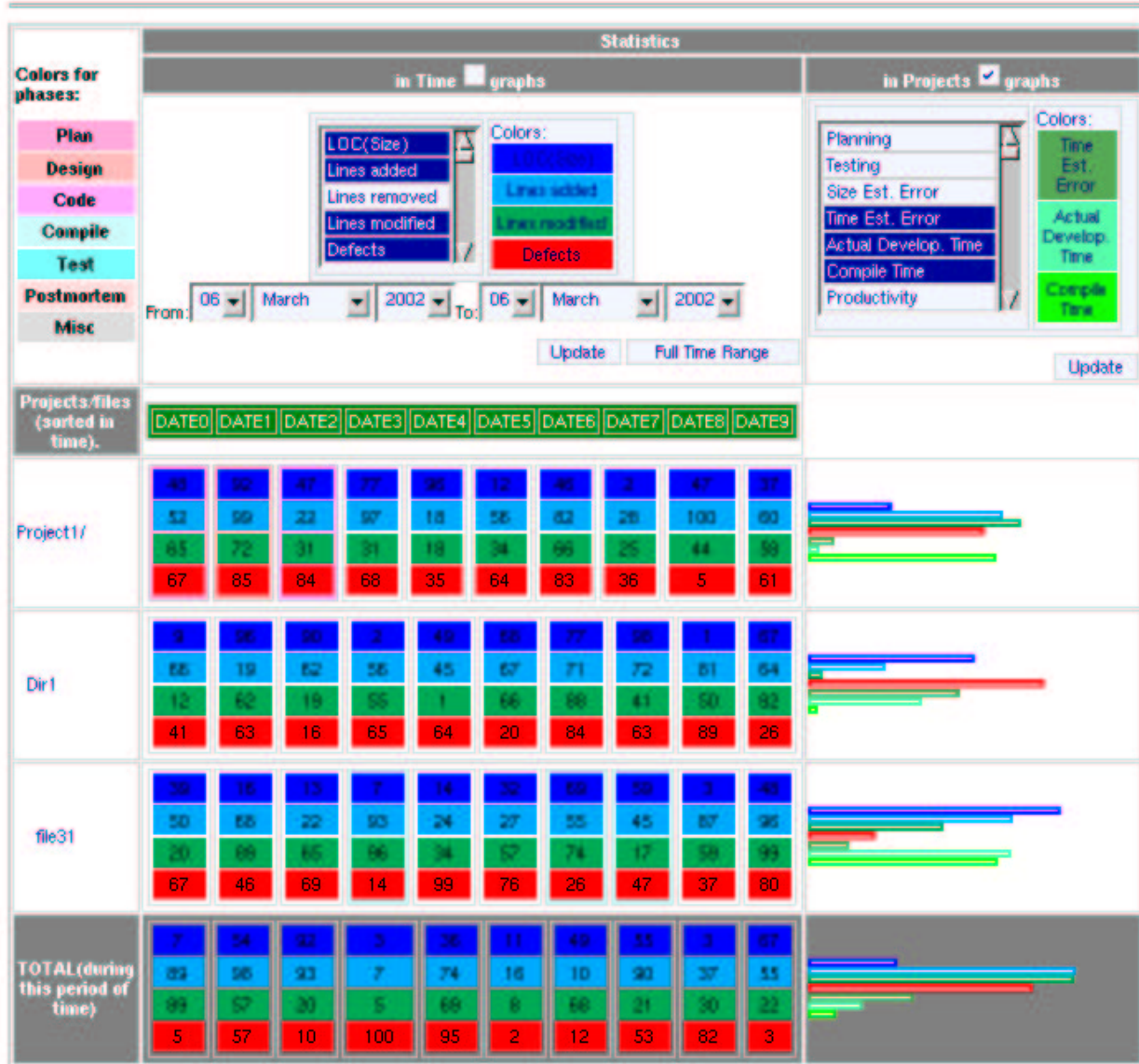
²Graph “Compile Defects vs Test Defects” would be presented just as a ration between compile and test defects, not real 2d graph

³This solution is temporary for current release of software. More sofisticated presentation of phase will be chosen for future releases

⁴Values for years in that drop-down box figured out from period of time system was used till current year

APSP Report

User: yoh Date: 03-06-2002 19:35



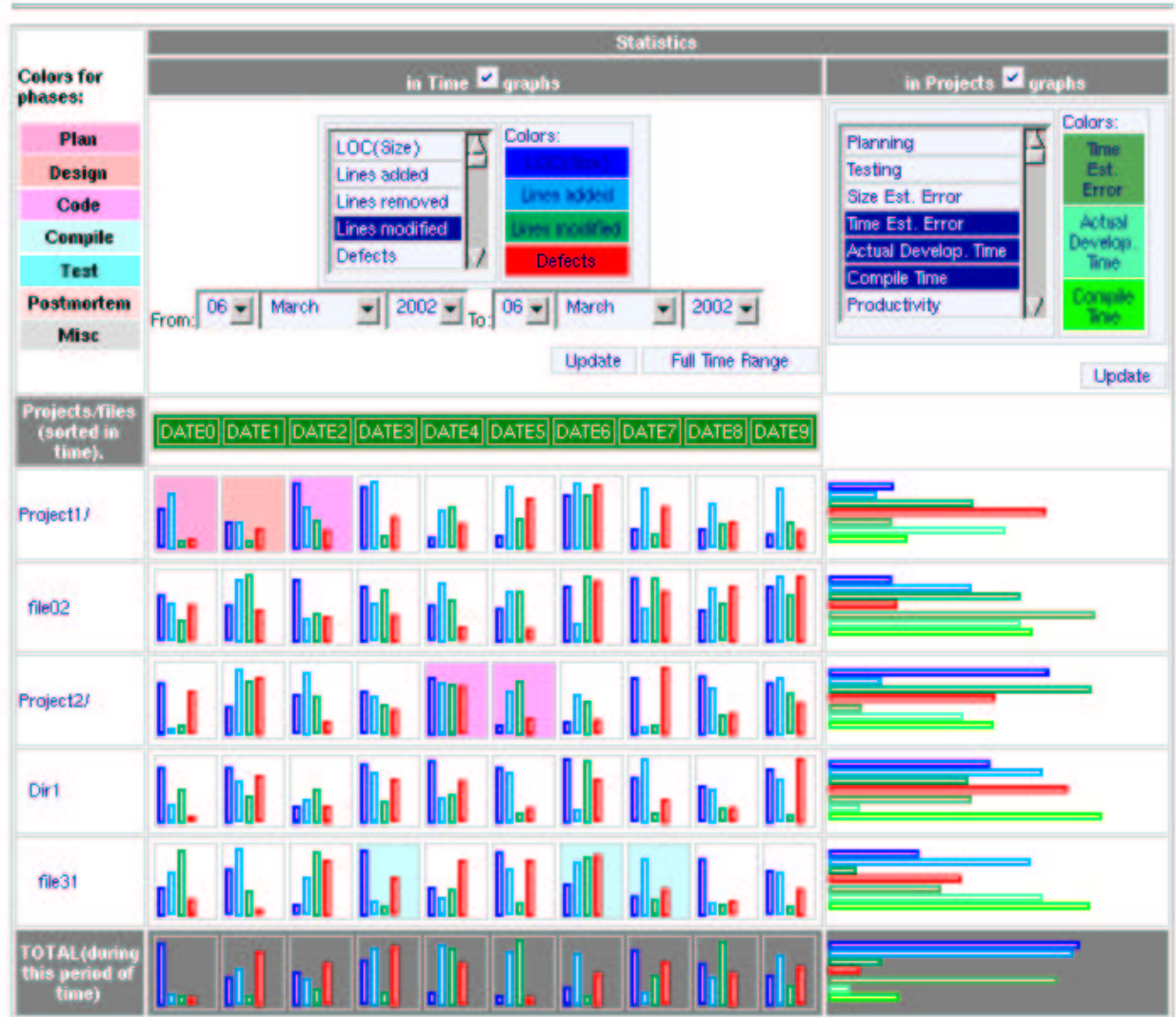
Web-design by Yaroslav Halchenko CS UNM. 2002



Figure 3.11: "Graphs"UI. View 2 (graphs + numbers)

APSP Report

User: yoh Date: 03-06-2002 19:30



Web-design by Yaroslav Halchenko C&S UNM. 2002



Figure 3.12: "Graphs"UI. View 1 (all graphs)

- Activating link on specific “DATEX” in the row of dates will zoom in time to that range of time if “DATEX” corresponds to a period of time more than a day.
- Left column of the table has list of projects, directories and files organized hierarchically using indentation. If user clicks with mouse (or activates by moving between links in html document with help of keyboard) on name of the project, then if project/folder was already opened, it closes so information in graphs will be presented only cumulative about project/directory, not separate for its members. If folder was closed (folded) it would be unfold after clicking on it, presenting information about its direct (next level in hierarchy) members - other directories and files.
Mouse click on the file name doesn't do anything.
- Last row in the table presents total (cumulative) statistics for specific “DATEX”, so user can assess general progress on his work in time if he currently involved in many projects.

References