Time Monitoring Tool
Software Requirements Specifications

Version <1.0>

Revision History

Date	Version	Description	Author
<01/01/2001>	<1.0>	First version	Martin Robillard

Table of Contents

Software Requirements Specifications

1. Introduction

1.1 Purpose

The purpose of this document is to describe the requirement specifications for a time monitoring tool for software development teams.

The intended audience of this document includes the prospective developers of the tool and the technical assessment personnel of the client organization.

1.2 Scope

The software system to be produced is a Time Monitoring Tool, which will be referred to as "TMT" thorough this document.

The Time Monitoring Tool will allow developers working within a defined software development process to record the time spent on the various software development activities, such as designing, coding, testing, or debugging. The TMT will also allow a manager to derive analyses and produce reports based on the data entered in the system.

The TMT could be used in any software development application to record the resources spent on the various software development activities. The objective of the TMT is to record the various activities being done. The goals are manifolds depending on the users of the data. The developers use TMT to record the activities that are being performed. The managers use data recorded from TMT to validate its planning, budgets and schedules. The software process manager use data from TMT to better understand the various software processes prescribed practices and to guide software process improvement concerns.

1.3 Definitions, acronyms, and abbreviations

Activity: nature of what is being done. The software process manager defines activities. Examples of activities are: analyzing, coding, designing, testing, etc.

Artifact: physical entity that results from an activity. Required artifacts are defined by the software engineering process. Examples of artifacts are SRS, architecture diagram, UML diagrams, source code, test scripts, user manuals.

DCM :   Developer Client Module

JSP :   Java Server Pages ,

MCM :   Manager Client Module

MCW: Manager Client Window

SM: Server Module

SRS: Software Requirement Specification

Task: unit of work defined by the manager and for which the developer is accountable. task is a component of the schedule with a start and end date.

Timestamp: **A qualified unit of time spent on a software development-related activity. The unit is qualified in terms of the following pieces of information: week, duration, project, task, activity, and sub-activity.**

TMT: **Time Monitoring Tool, which is the name of the tool to be developed**

TMW: **Time Monitoring Window**

1.4   References

IEEE Std. 830-1993: IEEE Recommended Practice for Software Requirements Specifications
JSP: http://www.esperanto.org.nz/jsp/jspfaq.jsp#q1

1.5   Overview

The rest of this document contains an overall description of the Time Monitoring Tool software system (section 2), and the specific requirements for the system (section 3).

2. Overall description

2.1 Product perspective

In some software development organizations, developers are required to record the time spent on the various activities that are related to the development of software. This information is commonly used for various administration tasks, such as budget planning. Without tool support, precisely recording the time spent on each activity can be a tedious task. A time monitoring tool can ease this task.

2.1.1 System Interfaces

The TMT system to be developed is a stand-alone tool that is integrated within the organization's Intranet. It consists in four major components: a Developer Client Module, a Server Module, a Database, and a Manager Client Module (see diagram 1).
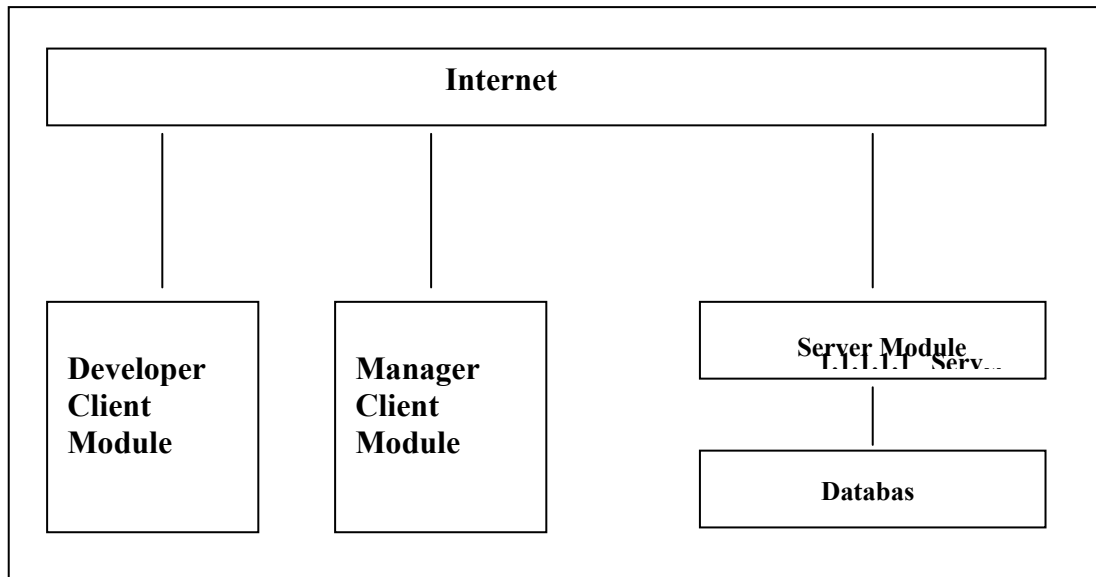
The DCMallows developers to log onto the TMT system and to record timestamps corresponding to their activities in a convenient way. The server module is a daemon that accepts connections from Developer and Manager Client Modules and serves as an interface between these modules and a database. The Manager Client Module allows a manager to retrieve timestamp information from the database to produce analyses and reports. The Database component can be any type of database. It does not have to be developed as a part of the TMT system, as long as the Server Module can interface with an available database system.
All components must execute on WindowsNT.

2.1.2 User interfaces

The Developer and Manager Client Modules must provide a user interface that is available through the Netscape WWW browser. The Server Module does not have a user interface other than a command to launch it. The Database module does not have a user interface.

2.1.3 Hardware interfaces

All components must be able to execute on a personal computer.

2.1.4 Software Interfaces

The Developer and Manager Client Modules must be Java Applets running within Netscape. The Server Module mu st integrate with a DBMS through the Java Database Connectivity (JDBC) interface. The Server must run within a Web Server available for WindowsNT, using dynamic content technology (e.g., Java Server Pages (JSP), CGI/Perl, and Cold Fusion).

2.1.5 Communication interfaces

The Developer and Manager Client Modules must communicate with the server over a TCP/IP connection. The Server and the Database components should be located on the same host.

2.1.6 Memory Constraints

The Developer and Manager Clients must be able to operate within 64MB (including memory for Netscape), and the server and database within 128MB.

2.1.7 Operations

The operation of the Developer and Manager client Modules must be easy and intuitive for professional software developers and mangers. No specific formation must be required to use the tool. The Server module must be installed and maintained with no interaction with existing software and should not required any specific technical skill from a network administrator.

Back up operations must be defined.

Recovery operations must be specified in case of network failure, user machine failure and database failure.

2.1.8 Site adaptation requirements

No specific site adaptation should be required.

2.2 Product functions

The two main functions of the Time Monitor software system are to allow developers to use a www browser to store timestamp records in a database, and to allow a manager to analyze these timestamp records.

A timestamp record consists of the time duration of a specific activity with its unique identification. The unique identification is made of three components: the project, the user and the date when the activity is taken place. The description of an activity is divided into three components: a task name, an activity, and an artifact.

For managerial purposes it is often useful to define the date in terms of the current week. The current week is defined as the week starting on the Monday immediately preceding the current day of the week, and ending on the Sunday immediately following the current day of the week, inclusively.

A task is a unit of work defined by the manager and for which the developer is accountable. A task is a component of the schedule with a start and end date. Examples of tasks are Implement module A, Design library XYZ. Developers usually work on assigned tasks. One developer may work on many tasks and a given task may involve many developers.

An activity is the nature of what is being done. Software developers are mostly involved in cognitive activities. The various cognitive activities composed the practices that are defined from the software engineering processes. The software process manager defines the activities and the developer must select the activities that are most representative of what is being done during the recorded time. Typical examples of activities are analyzing, coding, designing, testing, etc.

An artifact is a physical entity that results from an activity. The software engineering process defines required artifacts . Typical artifacts are SRS, architecture diagram, UML diagrams, source code, test scripts, user manuals, etc....Developers must select artifacts among predefined lists.

Names of Tasks, Activities and Artifacts are predefined for a project and are stored in the database of the TMT system.

The database stores two different types of items: user configurations and time stamp records. User configurations consist in a username, a password, and a list of projects, tasks, activities and artifacts currently available for this user.

Time stamp records stored in the database consist of a time stamp as described above, plus a field identifying the week, and a field identifying the username.

2.3 User characteristics

Users are software developers familiar with the web technology. Users are knowledgeable of the software engineering process and have good understanding of the tasks, activities and artifacts they are being involved with.

2.4 Constraints

The system should enforce user authentication security and guaranty timestamps reliability.

2.5 Assumptions and dependencies

No specific assumption or dependencies.

3. Specific functional requirements

3.1 Developer Client Module (DCM)

3.1.1 The user shall be able to load the DCM within Netscape

3.1.2 The DCM shall support the logging of users.

3.1.2.1 The initial window of the DCM shall contain a field for a user name, a field for a password, and a button labeled login. The password field shall be a "secret" field, which does not display what the user types.

3.1.2.2 When a user presses the login button, the DCM shall send a request to the SM to login the user.

3.1.3 If the logging of a user is successful (see 3.1.2.2), the DCM shall display the TMW.

3.1.3.1 The TMW shall always display the identifier for the current week.

3.1.3.2 The TMW shall always display the username of the user currently logged in.

3.1.3.3 The TMW shall display all the validated and non-validated time stamp records previously entered for the current week by the user currently logged in.

3.1.3.3.1 The validated and non-validated time stamp records shall be displayed in a tabular fashion, with every record being on a separate row.

3.1.3.3.2 The fields should be identified with labels. Intuitive and non-confusing abreviations can be used if necessary.

3.1.3.3.3 The fields should be non modifiable to prevent modification of the time stamp records.

3.1.3.4 The TMW shall display all the refused time stamp records previously entered by the user currently logged in.

> **3.1.3.4.1   The refused time stamp records shall be displayed in a tabular fashion, with every record being on a separate row.**

> **3.1.3.4.2   The fields should be identified with labels. Intuitive and non-confusing abreviations can be used if necessary.**

> **3.1.3.4.3   The fields should me modifiable to allow correction of the time stamp records.**

3.1.4   The existing TMW timestamps shall be updateable

3.1.4.1 It shall only be possible to modify the Project, Task, Activity, and Artifacts fields with values that are allowed for the user currently logged in.

3.1.4.2 It shall only be possible to modify the time fields for every day of the week using a positive numerical value of maximum 24.

3.1.4.3 Modifications to the records shall not be synchronized automatically with the server, that is, it should be possible to modify several values and then to click a bottom or perform another mechanism to update the server.

3.1.5   The TMW shall support the entry of new timestamps

3.1.5.1 It shall be possible to add new timestamps according to predefined fields

3.1.5.2 All new timestamps should come with the predefined selections for a given user.

3.1.5.3 It shall be possible to create up to 100 new timestamps for a given user and a given current week.

3.1.5.4 All time entry shall be in hours and minutes H:mn

3.1.6   The TMW shall support entry of timestamps

3.1.6.1 All fields of a timestamp shall have predefined values for the logged in user.

3.1.6.2 Project, Task, Activity, Artifact and at least one time field must be filled out before the record is sent to the database.

3.1.6.3 Time entry could be in duration or taxi mode.

3.1.6.4 G clock for the taxi mode.

3.1.7   The TMW shall provide cumulative totals

3.1.7.1 Each column day of the current week shall have the total number of hours.

3.1.7.2 Each project and task line of the current week shall have the total number of ours recorded.

3.1.7.3 The total number of hours recorded in the current week shall be displayed

3.1.7.4 All time records shall be displayed in hours and minutes: H:mn.

3.2   Manager Client Module (MCM)

3.2.1   The manager shall be able to load the Manager Client Module within Netscape

3.2.2   The Manager Client Module shall support the logging of managers.

3.2.2.1 The initial window of the MCM shall contain a field for a user name, a field for a password, and a button labeled login. The password field shall be a "secret" field, which does not display what the user types.

3.2.2.2 When a user presses the login button, the MCM shall send a request to the SM to login the user.

3.2.3   If the logging of a user is successful (see 3.2.2.2), the MCM shall display the MCW.

3.2.3.1 The MCW shall always display the username of the manager currently logged in.

3.2.3.2 The MCW shall display two groups of icons, which are for the user management and the project management.

3.2.4   MCW shall support user's management

3.2.4.1 The user management icons shall included: add a user, display/modify/disable users, and validate timestamps.

3.2.4.2 Clicking add user icons shall display a fill in form for adding a user to the project

3.2.4.2.1 Add user form shall enable the recording of the user id, the projects, the user's supervisor identification, and the selection of predefined fields for this user.

3.2.4.2.2 Add user form shall be validated for completeness before being sent to the Server Module.

3.2.4.3 Clicking Display all users icon should display the list of all users in alphabetical order with the identification of the projects that they are involved in and their supervisor.

> **3.2.4.3.1 This option could provide a list ordered alphabetically by last name of the user or by project or by supervisor.**

> **3.2.4.3.2 By selecting a user, his profil can be changed or disabled.**

3.2.4.4 Clicking validate timestamps should display the list of all timestamps records submit by users.

> **3.2.4.4.1 The timestamp records shall be displayed in a tabular fashion, every record being on a separate row.**

> **3.2.4.4.2 The fields should be identified with labels. Intuitive and non-confusing abreviations can be used if necessary.**

> **3.2.4.4.3 The manager can validate a timestamp record, refuse a record or leave timestamp record non-validated.**

> **3.2.4.4.4 The validated timestamps records should be inserted in the database.**

> **3.2.4.4.5 The refused timestamps records should be returned to the user to allow correction.**

> **3.2.4.4.6 The non-validated timestamps record should be stayed in the table to allow manager to validate or refuse a timestamp record another time.**

3.2.5   *MCW shall support project's management*

3.2.5.1 The project's management icon shall include the add project, add task, MS EXCEL and MSProject icons.

3.2.5.2 Clicking add project icon shall display a form to identify the new project to add.

3.2.5.3 Clicking the add task icon shall display a form to add a new task in a project selected.

3.2.5.4 Clicking the EXCEL icon shall display a form to identify the project; users, week's data and the EXCEL file where the corresponding data from the database should be transferred.

3.2.5.5 Clicking the MSProject icon shall display a form to identify the project; users, week's data and the MSProject file where the corresponding data from the database should be transferred.

3.3   Server Module (SM)

3.3.1   The SM shall be the only intermediate between the two client modules and the database.

3.3.2   The SM shall receive all the requests and format the pages.

3.3.3   The SM shall accept all connections from developers and manager client modules.

3.3.4   Upon log in request from the DCM the SM shall produce the TMW as specified in 3.1.2

3.3.5   Upon request for updating time stamp records from the DCM the SM shall update the database to reflect the new state of the DCM.

3.3.6   The SM shall validate and execute all requests coming from the DCM.

3.3.7   Upon log in request from the MCM the SM shall produce the MCW as specified in 3.2.2.

3.3.8   The SM shall validate and execute all requests coming from the MCM.

3.3.9   The SM shall display the TMT status

3.3.9.1 Any error of execution, communication, validation or else shall be identified and appropriate comment display.

3.3.9.2 The SM shall try to recovery from most common errors.

4.   Classification of functional requirements

Functionality	Type
3.1.1 The user shall be able to load the DCMwithin Netscape	**Essential**
3.1.2 The DCMshall support the logging of users.	**Essential**
3.1.2.1 The initial window of the DCM shall contain a field for a user name, a field for a password, and a button labeled login. The password field shall be a "secret" field, which does not display what the user types.	**Essential**
3.1.2.2 When a user presses the login button, the DCM shall send a request to the SM to login the user.	**Essential**
3.1.3 If the logging of a user is successful (see 3.1.2.2), the DCM shall display the TMW.	**Essential**
3.1.3.1 The TMW shall always display the identifier for the current week.	**Desirable**
3.1.3.2 The TMW shall always display the username of the user currently logged in.	**Desirable**
3.1.3.3 The TMW shall display all the time stamp records previously entered for the current week by the user currently logged in.	**Essential**
3.1.3.3.1 The time stamp records shall be displayed in a tabular fashion, with every record being on a separate row.	**Desirable**
3.1.3.3.2 The fields should be identified with labels. Intuitive and non-confusing abbreviations can be used if necessary.	**Desirable**
3.1.4 The existing TMW timestamps shall be updateable.	**Essential**
3.1.4.1 It shall only be possible to modify the Project, Task, Activity, and Artifacts fields with values that are allowed for the user currently logged in.	**Essential**
3.1.4.2 It shall only be possible to modify the time fields for every day of the week using a positive numerical value of maximum 24.	**Essential**
3.1.4.3 Modifications to the records shall not be synchronized automatically with the server, that is, it should be possible to modify several values and then to click a bottom or perform another mechanism to update the server.	**Essential**
3.1.5 The TMW shall support the entry of new timestamps.	**Essential**
3.1.5.1 It shall be possible to add new timestamps according to predefined fields.	**Desirable**

3.1.5.2 It shall be possible to create up to 100 new timestamps for a given user and a given current week.	**Essential**
3.1.5.3 All new timestamps should come with the predefined selections for a given user.	**Desirable**
3.1.5.4 All time entry shall be in hours and minutes H:mn.	**Essential**
3.1.5.5 All empty time fields shall be at 0	**Desirable**
3.1.6 The TMW shall support entry of timestamps	**Essential**
3.1.6.1 All fields of a timestamp shall have predefined values for the logged in user.	**Desirable**
3.1.6.2 Project, Task, Activity, Artifact and at least one time field must be filled out before the record is sent to the database.	**Essential**
3.1.6.3 Time entry could be in duration or taxi mode.	**Desirable**
3.1.6.4 TMW shall provide a running clock for the taxi mode.	**Desirable**
3.1.7 The TMW shall provide cumulative totals.	**Essential**
3.1.7.1 Each column day of the current week shall have the total number of hours recorded.	**Essential**
3.1.7.2 Each project and task line of the current week shall have the total number of hours recorded.	**Essential**
3.1.7.3 The total number of hours recorded in the current week shall be displayed.	**Essential**
3.1.7.4 All time records shall be displayed in hours and minutes: H:mn.	**Essential**
3.2.1 The manager shall be able to load the Manager Client Module within Netscape.	**Essential**
3.2.2 The Manager Client Module shall support the logging of managers.	**Essential**
3.2.2.1 The initial window of the MCM shall contain a field for a user name, a field for a password, and a button labeled login. The password field shall be a "secret" field, which does not display what the user types.	**Essential**
3.2.2.2 When a user presses the login button, the MCM shall send a request to the SM to login the user.	**Essential**
3.2.3 If the logging of a user is successful (see 2.2.2), the MCM shall display the MCW.	**Essential**
3.2.3.1 The MCW shall always display the username of the manager currently logged in.	**Desirable**
3.2.3.2 The MCW shall display two groups of icons, which are for the user management and the project management.	**Desirable**

3.2.4 MCW shall support user's management.	Essential
3.2.4.1 The user management icons shall included: add a user, delete a user, display all users, and modify user's file.	Essential
3.2.4.2 Clicking add user icons shall display a fill in form for adding a user to the project.	Essential
3.2.4.2.1 Add user form shall enable the recording of the user id, the projects, the user's supervisor identification, and the selection of predefined fields for this user.	Essential
3.2.4.2.2 Add user form shall be validated for completeness before being sent to the SM.	Essential
3.2.4.3 Clicking delete user icons shall display the list of all users in alphabetical order with a check box for delete user and the date of the day that could be overwritten.	Essential
3.2.4.4 Clicking Display all users icon should display the list of all users in alphabetical order with the identification of the projects that they are involved in and their supervisor.	Essential
3.2.4.4.1 Display all users could provide a list ordered alphabetically by last name of the user or by project or by supervisor.	Desirable
3.2.4.5 Clicking Modify user's file icon shall display the list of users in alphabetical order with a check box for modify user's file.	Essential
3.2.4.5.1 Clicking the modify user file shall display the complete user file an enable overwriting of any field.	Essential
3.2.5 MCW shall support project's management	Optional
3.2.5.1 The project's management icon shall include the EXCELL and MSProject icons.	Optional
3.2.5.2 Clicking the EXCELL icons shall display a form to identify the project; users, week's data and the EXCELL file where the corresponding data from the database should be transferred.	Optional
3.2.5.3 Clicking the MSProject icons shall display a form to identify the project; users, week's data and the MSProject file where the corresponding data from the database should be transferred	Optional
3.2.5.4 All forms shall be sent to SM for validation.	Essential
3.2.5.4.1 Unvalidated fields shall be clearly indicated with appropriate comments.	Desirable
3.3.1 The SM shall be the only intermediate between the two client modules and the database.	Essential

3.3.2 The SM shall receive all the requests and format the pages.	**Essential**
3.3.3 The SM shall accept all connections from developers and manager client modules.	**Essential**
3.3.4 Upon log in request from the DCM the SM shall produce the TMW as specified in 3.1.2.	**Essential**
3.3.5 Upon request for updating time stamp records from the DCM the SM shall update the database to reflect the new state of the DCM.	**Essential**
3.3.6 The SM shall validate and execute all requests coming from the DCM.	**Essential**
3.3.7 Upon log in request from the MCM the SM shall produce the MCW as specified in 3.2.2	**Essential**
3.3.8 The SM shall validate and execute all requests coming from the MCM.	**Essential**
3.3.9 The SM shall display the TMT status.	**Essential**
3.3.9.1 Any error of execution, communication, validation or else shall be identified and appropriate comment display.	**Desirable**
3.3.9.2 The SM shall try to recovery from most common errors.	**Desirable**