

# Pomotux Project Presentation spech

## ***SLIDE 1 - Max***

Good afternoon, welcome to our project presentation. We are team number 8, composed by Riccardo Buttarelli, Daniel Graziotin and Massimiliano Pergher. We are going to present our project, named Pomotux. Pomotux is an activity manager based on the Pomodoro technique by Francesco Cirillo, lead member of the XPlabs crew.

## ***SLIDE 2 - Max***

Before talking about the application, I will briefly introduce some key aspects of the Pomodoro Technique. Then we will see the mission statement, the use cases of the project, some screenshots of the program.

what happens when an activity is faced by a user, the technologies behind the program, the problems faced and finally we will draw some conclusions about the experience.

## ***SLIDE 3 - Max***

Let's speak about the Pomodoro Technique:

It is a time management methodology that can be used with any type of task, or activity. The aim of the technique is to use time as a valuable ally instead of an enemy, in accomplishing what we want to do in the way we want to do. The fundamental unit of the technique is the Pomodoro, which can be seen as a non-separable unit of time (25 minutes normally).

The basic steps are: choosing an activity, start a pomodoro, work and focus on the activity until the pomodoro finishes and take a break. This not all regarding the technique, for more information please visit the website

## ***SLIDE 4 - Max***

Our mission statement was to create an activity manager implementing the basic features of the Pomodoro Technique and to offer a very simple interface to the user.

## ***SLIDE 5 - Max***

Now I will show you the general use cases diagram that we derived during the analysis phase. This graph shows a unique actor, the user, does during the use of the program. For example he can insert, delete or modify an activity, start a pomodoro, finish o break the pomodoro etc. [3-4 secondi pausa]

## ***SLIDE 6 - Max***

In this slide we can see the main window of Pomotux, which is the Activity Inventory Sheet, the general container of the activities. All the possible operations are located on the bottom side, including the button for opening the Todo Today Sheet.

## ***SLIDE 7 – Max***

The TTS is the container for the activities to be performed today. It looks like the AIS but the operations are located on the right side, to help the user to give an different identities to both windows.

## ***SLIDE 8 - Max***

On this slide we can see a screenshot of a running Pomodoro.  
Let's focus about the technical aspects of the project.

## ***SLIDE 9 – Riccardo***

Good afternoon, I'm Riccardo Buttarelli. First I will speak about the component of our system and then I will explain what happens when a user face an activity with Pomotux.

This slide shows a model for the system layers. We rely on several different technologies to achieve our goal and perform secure information hiding. Only the last layer, the GUI, is the one with which the user interacts with.

At the bottom layer we have a persistent data structure to store the activities. We decided to use a database because we have entities to be stored that are also in relationships and must be retrieved,deleted,updated and inserted easily. We chose SQLite because it is a database implementation that stores everything in a single file. (public domain source code)

Considering our tiny previous c++ knowledge and experience, we needed something to help us to communicate with the database, providing an abstraction between our logic unit and SQLite without merging code and SQL queries. This is called Object-Relational mapping. To carry out this job we found LiteSQL, which is a young open source project that tries to provide Object Relational Mapping to C++, and let developers focus about the programming aspects of the system without thinking to SQL Queries.

Between the graphical user interface and the database we put our logic unit which contains the definitions of the entities of the systems and their relations, as well as methods to access the SQLite via LiteSQL.

Finally at the top layer we have the gui which is the only part the the user sees and interacts with. We decided to use QT framework for its simplicity and the well-made documentation

## ***SLIDE 10 – Riccardo***

Let's see what happens when a user faces an activity with our program.

From the gui the user STARTs the activity. The gui sends a message to the logic unit that starts the Pomodoro (a timer) and keeps the GUI updated.

If everything continues fine, the user faces his activity and the Pomodoro finishes, launching a signal to the GUI to inform the user and perform some other internal operations, such as database update.

If the user through the gui clicks on the STOP button the timer is stopped and re-set. A different signal is returned to the GUI which informs the user and performs other (cleaning) operations.

Let's see something about the problems we faced and some conclusions about the experience.

## **SLIDE 11 – Daniel**

Good afternoon, I'm Daniel and I'm going to talk about the problems faced during the development. Then I will draw some conclusions regarding the experience and finally we will have a short demo run of Pomotux. After that, we will answer to eventual questions.

Problems faced:

- 1) apart from those related to our knowledge of C++/QT, the biggest problems were all given by LiteSQL.
- 2) Version 0.3.3 is the sixth release of the project, little community behind (3-4 people). Released for non-critical use. However we tested it and believed in it
- 3) They are not following software development methodologies: release when it's ready. Nobody likes to write documentation, that is poor. This lent us to a wrong implementation strategy at Sprint 2, we had to roll back and lost an entire week
- 4) Very serious bug discovered thanks to unit testing. We wrote to the community that enjoyed our requests and were also happy to help us in exchange of a feedback. The bug fix is set to the next LiteSQL version that came out as beta 10 days ago, just because we requested it. Meanwhile they also wrote a patch for us to test the it.

## **SLIDE 12 – Daniel**

Conclusions:

- 1) Numbers: 16/16 must have, 5/5 ought to have, 2/3 nice to have. Ready for the third
- 2) Unit testing works, really. Application and framework test
- 3) Fruitful use of frameworks: Pomotux involves lots of different queries using JOIN, GROUP BY etc. At the price of loosing some lower-level control (abstraction of frameworks) we could focus on the real aspects of the project without query related problems
- 4) Collaboration with external opensource community: was funny and also interesting, very strong reaction to our requests
- 5) This year Scrum method was appreciated however I continue to believe that Sprints of one week of length are too short and stressful. Suggestion for next year: half of sprints with double length. An introduction to QT in the labs could also help.
- 6) Knowledge expansion, at a quite high price in terms of time. Heavy course.

## **SLIDE 13 – Daniel**

Thank you very much for your attention, let's have a short demo run of the project.

This is Pomotux that starts showing the Activity Inventory Sheet. We can perform operations on the already present activities or insert a new activity. This is the Todo Today Sheet that is currently empty. The activity just inserted will now be scheduled in the Todo Today Sheet. What has just happened is a database operation that puts in relation the Activity with the Todo Today Sheet. As you see, the windows are immediately updated (this is done via signals). In the Todo Today Sheet other operations can be performed, like Activity finishing, post-pone and ordering. We set up for you a Pomodoro of 20 seconds length. The user starts the Activity, the timer shows up. The user can also handle possible interruptions by registering them during the Pomodoro. When the Pomodoro finishes, the user is alerted to make a break and a sound is played. There are other possibilities and personalizations in the Option window. This is everything for now, we would be pleased to answer to questions. Thank you.