
XLQSI Documentation

Release

Grégoire Lurton

Jan 28, 2018

CONTENTS

1	General Presentation	1
1.1	From LQSI to XLQSI	1
1.2	XLQSI overview	2
2	Setting-up a Data entry file	3
2.1	Downloading and unpacking	3
2.2	Set language	3
2.3	Define the completion checks	3
3	Collecting data	5
4	Data vizualisation	7
5	Technical Description	9
6	Helping improving XLQSI	11

GENERAL PRESENTATION

The need for an Excel tool to collect data related to the continuous quality strengthening of laboratories was first raised by the Laboratory Systems Strengthening team in ITech. In order to use the Laboratory Quality Stepwise Implementation (LQSI) tool, developed by the World Health Organization in collaboration with the Dutch Royal Tropical Institute (KIT), field teams needed a tool that would be easy to use, adaptable to specific conditions or local regulations in the field, and that would allow a better follow-up of laboratories progresses and results.

The wide availability of Excel on computers around the world, and the increasing familiarity of most health workers with this software made it an easy choice for a technology in which to build a new tool. A first round of transcription of LQSI in Excel in Cambodia proved successful in offering an easy to use, widely understood implementation. In the summer of 2017, the decision was made to adapt this first version to make it an adaptable tool that could be adapted in different settings, and that would be well suited for advanced data analysis.

1.1 From LQSI to XLQSI

The LQSI is comprehensive framework developed by WHO and KIT to help laboratory managers to implement quality improvement programs. The framework is presented [on WHO's website](#) which allows for an easy navigation and comprehensive explanations.

LQSI is a four phases process towards laboratory quality improvement. In each phase, a set of essential activities are defined, that should be carried out. These activities are classified in 12 categories. Additionally, for each activity, a checklist is defined to ensure that it has been well implemented. The WHO's website conveniently provides different ways to generate checklists by phase or by category of activity, and roadmaps to follow the relationships and successions of each activity.

Meanwhile, when implementing this framework, two main problems can be encountered: 1. The high dimensionality of the framework and the multiplicity of items can make hard to navigate and get an overview of. 2. The checklists defined are generic, and may be hard to use in any specific setting.

To palliate this problem, XLQSI has been designed to allow users to easily **navigate and visualize** activities categories, and to have **synthetic visualizations** of the activities carried and their results. It also contains different **completion checks** and **progression bars** to allow users to manage their work time efficiently. Finally, XLQSI offers a framework to **adapt completion checks** for the different activities to local contexts, thus helping improving the usability of the tool. Refer to [the appropriate paragraph](#) for an explanation of how to define local checklists.

Important: An important terminology note: we differentiate between the *checklist items*, which are the generic checks offered in LQSI, and the *completions checks*, which are the context specific checks defined by the users when they set-up data collection.

1.2 XLQSI overview

XLQSI is made of two main modules. All the modules can be [downloaded from the github repository](#). The current version of XLQSI is developed in Excel enriched in Python using the [xlwings library](#) and [add-in](#). The functions packaged in the downloaded zip file do not necessitate any further installation, but only on Windows. Please refer to [technical section](#) or on the github repository to learn how to access and modify non packaged versions.

Warning: While setting-up the data entry module and importing data for visualization, XLQSI needs to access the packaged Python libraries. For this reason, it is important that the original set-up and the data importation be made in an arborescence that respects the file organization in the downloaded zip file.

1.2.1 Data Entry Module

The data entry module is the main tool for field workers to collect data on their laboratory quality activities. It can be adapted to a specific setting before being used. The different steps for the initial setup of the data entry is described in the [data entry set-up](#) page.

Once a data entry template is defined, it can be shared and used in different labs. The data is entered in two different steps, which are described in the [data entry](#) page. At this stage, the user can already have an overview of the current results and performance of the facility.

1.2.2 Data Visualization Module

The data visualization module is a separate Excel workbook, in which the user can compile all the different data collection iterations for a given laboratory, and visualize the evolutions of the performance, and the strong and weak points of this facility. The use of this module is described in the [Data Visualization](#) page.

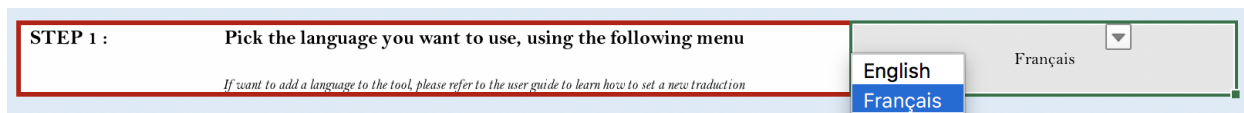
SETTING-UP A DATA ENTRY FILE

2.1 Downloading and unpacking

2.2 Set language

It is currently possible to use XLQSI in five different languages. The English, French, Russian and Spanish versions have been directly taken from the WHO's website. The Khmer translation was made by ITech's team in Cambodia.

Language selection happens on the first tab of the data entry workbook, just by choosing the desired language in the dropdown menu as shown in [Fig. 2.2](#).



Adding a new translation language to XLQSI is a simple even if very systematic task. The procedure to adapt translations or add new ones is described in *technical documentation*.

2.3 Define the completion checks

The second step in setting-up a data entry file is to create the completion checks for each activity. To understand the link between the two items, please refer to the *introductory content*.

COLLECTING DATA

DATA VIZUALISATION

TECHNICAL DESCRIPTION

Todo: Include explanation of development and compiling of XLQSI

HELPING IMPROVING XLQSI