

# ► SOFT TECH HUB

Create your own opportunity...

Ashutosh Kumar Choubey ► SOFT TECH HUB ► 3/25/2018

Hello My Dears,

Today I am sharing something personal with you its actually a story I got from a blog.

A group of software engineers were celebrating in a 4-star restaurant after getting their salary. At the time of billing, they offered,

100 Rs. As tips. The waiter was happy and smiling at a guy sitting in the corner (he was not the who offered the tip).

They left the restaurant and the groups started asking him “Why was he smiling at you?”

He started

My father advised me two years ago that I should never offer tips in a restaurant, especially when it comes to star rated restaurant

He told me that an average junior engineer gets 25000Rs. As their salary and a waiter in a average 4Star Restaurant gets more than

30000 Rs. As their salary.

Look at the irony, one who are getting 25000 offer tips and the one who gets 30000 is on the receiving side. And engineers feel as if

They helped someone.

So, if you want to offer tips, do that for a road side vendor or for street vendors.

One of the group asked again “that is true but why did he smile at you?”

He answered he was the same guy who advised me.

# SOFT TECH HUB

Create your own opportunity...

Writing good software requirement takes skill, practice, and patience. The key questions we face are:

**What type of language do we use?**

**What level of detail do we need?**

**In what form should we document and present the requirement?**

When faced with the prospect of writing requirement, many organizations improvise their own methods. The danger, however, is that even small oversight can lead to problems during implementation and execution.

- If a requirement is ambiguous or difficult to read, it may be misinterpreted or not read in its entirety.
- If particular details are missing, it may result in a buggy or haphazard design.
- If business rules are not completely and clearly defined, it may result in missing or improper functionality.
- If it lacks specificity, it will result in something being that is other than what was intended.

We can guard against these scenarios by following a few solid practices. In this guide

I will cover some tried-and-true techniques for creating good requirements, including the following topics:

- Writing user stories
- Defining conditions of satisfaction
- Creating workflow diagrams
- The use of wireframes and visual designs
- Defining nonfunctional requirements
- Creating test tables and test scenarios

To put these into perspective, I will run through an example of how these techniques might be used to define a simple web application. These techniques are well documented in the software community and have been time tested in software shops around the world. This is meant to be a primer for those new to writing requirements but may also be helpful to those with experience as well. In particular, if you are unfamiliar with agile methods this will help explain the user story approach and will touch on the similarities and differences with the waterfall method.