**Panzura automation framework用例编写规范：**

1. **创建项目(New Project**)

项目名应该为有意义的单词或则简写， Type选择directory， 利于用例管理.

1. **创建测试集(New Suite)**
2. 测试集的名字必须易读有意义，应该小于20个字符，根据名字可以知道包含了哪些类型的用例，文件类型应该为txt
3. 测试集的名字是自动根据文件、目录的名字创建的，后缀名会被截去，下划线会转换为空格，如果名字全部为小写，每个单词的首字母会大写。

例如：cloud\_connector\_test\_suite.txt-> Cloud Connector Test Suite

1. **创建测试用例(New Case)**
2. 测试用例的名字必须易读有意义，应该小于40个字符，采用大驼峰命名规则（每个单词首字母大写，其它字母小写）， 如有case id需加在最前面。

例如：QT-22609 Amazon-Cloud-Upload-Download。

1. 应该根据测试用例的步骤，注释及条件更新用例文档。
2. 为每一个case给定合适的tag，方便分类
3. 一般情况下测试之间应该是独立的，如果存在依赖，应该给予详细的注释，并通过${PREV TEST STATUS}变量验证上一个测试的状态。
4. 如果存在重复的步骤应该封装成关键字，即减少代码量也方便维护。
5. 避免使用sleep， 应该使用Wait的内置关键字。
6. **创建测试资源(New Resource)**
7. 资源文件统一放入Resources文件夹，文件名最好不要超过20个字符，文件类型为robot.
8. 根据资源的类别进行分组，同一类型的数据应该放入同一资源文件，页面对象应该放入\GUI\_Xpath文件夹，页面的基础方法和功能逻辑应该放入\sub文件夹。
9. 资源文件应该逐级引用，避免重复混淆
10. **用户关键字、方法(New User Keyword)**
11. 方法的名字必须易读有意义，应该小于30个字符，采用大驼峰命名规则（每个单词首字母大写，其它字母小写）
12. 为了增加方法的可读性，应该增加一个前缀。

例如: Get, Set, Check, Is, Wait…

1. 应该根据方法的用途，变量，返回值更新关键字文档。
2. 添加方法前应当查看有无相似方法，不要添加重复的方法。
3. **变量(Variable)**
   1. 变量的名字必须易读有意义，应该小于20个字符。
   2. 常量、全局变量应该全部大写。例如：APP\_URL,DB\_SERVER, 局部变量采用小驼峰命名规则（第一个单词小写，其它单词首字母大写）
   3. 常量、全局变量应该放在脚本的最前面
   4. 变量可以使用下划线，不推荐使用空格

**Xpath 编写规范：**

1. **定位**

Web元素应优先使用id定位，如果元素没有id，使用xpath保证元素是唯一识别的元素，id定位 > name定位 > 属性定位。避免使用序号定位，例如：//div[2]

1. **命名**

为了增加xpath变量的可读性，应该增加一个后缀， 常见的控件后缀：

* + - Button -> Btn
    - TextBox -> Txt
    - CheckBox -> Chk
    - ComboBox -> Cmb
    - Label -> Lbl
    - LinkLabel -> Llb
    - ListBox -> Lst
    - PictureBox -> Pic
    - ProgressBar -> Prg
    - RadioButton -> Rdo
    - Div -> Div

**抽象层次:**

我们的PZA-AF项目大致分为测试用例层，功能模块层和元素操作层，我们需要将元素操作层分解为最小，甚至一个点击操作就是一个关键字，这样做的好处在于可以使用元素关键字封装成不同的功能方法，而不需要重新再写一遍，提高了方法的重用性，而且便于维护。

例如：

LoginUI方法可分解为InputUser，InputPassword, ClickLoginButton关键字