

在实际工作中我们会遇到很多的需要流式处理的程序，这种程序我们一般都会采用如下的两种设计：

设计一：

```
public interface IProcessor
{
    Object step1(...);

    Object step2(...);

    Object step3(...);
}
```

由于我们的任务处理基本可以分为三个流程，所以我们在IProcessor接口中定义了三个步骤，接下来我们会实现很多个IProcessor，但是假如有一天我们需要在第一步和第二步之间增加一步，这时所有的接口都得加一个步骤，而且这样做也不利于把有些处理变成服务。

设计二：

```
public class TaskContext
{
    . . . .
}
```

```
public interface ITaskProcessor
{
    Void ProcessTask(TaskContext taskContext);
}
```

这里我们定义了一个单一的处理步骤，针对每一种处理步骤我们可以实现一个具体的流程处理器，在最外层使用流程控制器来组合各种流程处理器来实现一个大的业务处理，这里你也可以将其看做是一个管道式的处理（顺序工作流），而且我们还可以更进一步将一些通用的处理变成服务，然后用工作流来串接组装服务形成一系列的工作流，这样的话流程的多版本运行也都比较好办了。