vii.那么，我们如何伪造理想的过程呢？

前面的内容描述了我们希望遵循的理想过程以及在这一过程中将产生的文件。如果我们以理想的方式完成了工作，那么我们就可以通过生成文档来伪造流程。我们试图按照我们所描述的顺序提出文件如果我们不能得到一条信息，我们注意到在文档中信息应该放在哪里的部分，并继续设计，就好像97条信息预期会发生变化一样。如果我们发现错误，就会更改它们，并在后续文件的相应改动。我们将文档作为设计的媒介，在将文档纳入文档之前，不会考虑作出任何设计决策。在各个层面都证明了，不管我们经常在中途遇到什么问题，最终的文档都会更容易理解和准确。我们不展示事情发生的方式，我们展示我们希望它们发生的方式，以及事情发生的方式。甚至数学——我们中许多人认为最理性的学科，也遵循这个过程。数学家们孜孜不倦地完善他们的证明，通常提出一个与他们发现的第一个非常不同的证明。第一次证明往往是一个痛苦的发现过程的结果。当数学家研究证明，理解生长和简化被发现。最后，一些数学家找到了一个更简单的证明，使定理的真实性更加明显。简单的证明之所以出版，是因为读者对定理的真实性感兴趣，而不是发现它的过程。我们认为类比推理更适用于软件。 在那些阅读软件文档的人想要理解这个程序，而不是重温他们的发现。通过提供合理的文档，我们提供了他们需要的东西。我们的文档在一个重要的方面不同于理想化的文档。我们制造了一个记录我们所有采纳和排除的所有备选方案，包括记录在文档早期版本中的决定。对于每一个问题，我们都解释了为什么它被采纳，为什么它最终被排除。几个月，几个星期，甚至几个20年后，此软件的维护者将有许多相同的问题，他将在我们的文件中找到答案。

软件初阳171班

201736900105

方梓安