敏捷开发是一种以人为核心、迭代、循序渐进的开发方法，相对于传统软件开发方法的非敏捷，更强调程序员团队与业务专家之间的紧密协作、面对面的沟通（认为比书面的文档更有效）、频繁交付新的软件版本、紧凑而自我组织型的团队、能够很好地适应需求变化的代码编写和团队组织方法，也更注重软件开发中人的作用。

传统软工方法一般包含迭代开发，瀑布开发，螺旋开发。下面我们比较他们之间的差异。

敏捷开发相比迭代式开发，相同的是两者都强调在较短的开发周期提交软件。基于Scrum的迭代开发一般会在一个比较长的一个迭代周期频率下不断交付，同时迭代中不允许有变化的需求，这样就有一些场景让这种迭代很困难，例如紧急的技术支持、临时增加的非常高的优先级的需求等等，另外项目的估算非常难，导致不容易承诺。相比较，敏捷方法的周期可能更短，并且更加强调队伍中的高度协作。敏捷开发的原则之一是拥抱变化需求时刻在变，人们对于需求的理解也时刻在变，项目环境也在不停的变化，因此你的开发方法必须要能够反映这种现实，敏捷开发方法就是属于适应性的开发方法，而非预设性。另外，敏捷开发更适用于小型团队，在一个办公室工作，属于那种通信基本靠吼的状态，当然团队成员之间的交互会更方便。另外敏捷开发强调用户（或用户代表）要与开发团队在一起工作，便于及时沟通交流。重视交互也应该可以算是最明显的区别之一。这样还有一个好处，就是有利于团队中知识的迅速传播。即使有人离开团队，另外的人也能完成相应的工作。因此，“人与交互”被列为敏捷开发价值观之一，并排在第一位。

敏捷开发区别于瀑布式的特征很明显，敏捷开发是以一种迭代的方式推进的，而瀑布模型

式是最典型的预见性的方法，严格遵循预先计划的需求分析、设计、编码、集成、测试、维护的步骤顺序进行，步骤成果作为衡量进度的方法，例如需求规格，设计文档，测试计划和代码审阅等等。敏捷开发过程中，软件一直处于可使用状态，它将项目分成若干相互联系、可以独立运行的子项目，因此，每个阶段软件都是可见的，客户可以直观的体验并提出意见。如果按照瀑布式流程，客户可能在签订开发合同3个月后，看到的还只是各种文档（需求文档、设计文档、详细设计文档等等），客户心理或许会不踏实。瀑布式的主要的问题是它的严格分级导致的自由度降低，项目早期即作出承诺导致对后期需求的变化难以调整，代价高昂。瀑布式方法在需求不明并且在项目进行过程中可能变化的情况下基本是不可行的。在瀑布式开发中，需求修改的时间越靠后，成本越大，所以需求分析人员的压力很大。由于敏捷开发是迭代式的，并且迭代周期较短，因此很容易相应新需求或是对旧需求的修改。瀑布式开发有很多文档，但敏捷开发不是没有文档，而是轻文档。在敏捷开发过程中，适量的文档还是很有帮助，有助于整理思路，加快沟通和讨论，比如概念设计文档、架构图、SWOT分析文档等等，这些文档在每个产品版本开始之前会有产生，在每个迭代的过程中根据业务人员和市场的反馈也会有一些变更。通过实践证明，这对产品的思路、沟通讨论都非常有帮助。而且这些文档，大多是几页PPT，书写和维护工作都很小。

和螺旋开发方法相比，敏捷方法强调适应性而非预见性。螺旋开发将瀑布模型和快速原型模型结合起来，强调了其他模型所忽视的风险分析，特别适合于大型复杂的系统。“螺旋模型”的核心就在于您不需要在刚开始的时候就把所有事情都定义的清清楚楚，您轻松上阵，定义最重要的功能，实现它，然后听取客户的意见，之后再进入到下一个阶段。如此不断轮回重复，直到得到您满意的最终产品。螺旋开发很大程度上是一种风险驱动的方法体系，因为在每个阶段之前及经常发生的循环之前，都必须首先进行风险评估。而敏捷开发，针对软件开过程中诸多的不可预见性，强调的是适应性，适应性的方法集中在快速适应现实的变化。当项目的需求起了变化，团队应该迅速适应。这个团队可能很难确切描述未来将会如何变化。

我们的项目是一个unity横版动作游戏，由于项目规模较小，开发目标可能会有较多变化，项目开发人员便于沟通，且比较灵活，所以我们采用敏捷方法来开发项目。我们会发挥人的主观能动性，给开发人员充分的发挥空间，把重点放在项目本身上，快速制作出初步满足需求的产品，并在之后进行不断变动，完成产品。如果采用传统软件工程方法，对于我们的小型项目而言，在项目前期，我们可能会把大量的时间花费在开发前的各种标准文件上，受限于团队规模，我们可能无法在之后的时间内完成令需求方满意的产品。对于游戏而言，玩法创新十分重要，我们采用敏捷开发可以随时依据开发团队的奇思妙想进行项目的调整，从而达到更好的效果，而传统开发方法无法满足我们在开发过程对项目的变动需求。