# 高适应性

## 适应性及其必要性

敏捷软件开发方法过程是敏捷开发组织的复杂适应性的过程，所以敏捷开发组织对于系统特性具有高适应性。这里提及的适应性是指软件使不同的系统约束条件和用户需求得到满足的容易程度。它要求软件尽可能适应各种硬软件运行环境，以便软件的推广和移植[1]。敏捷软件开发方法接受用户需求的不断变化并且能迅速做出响应，以用户需求为导向，以核心价值观为指导，以原则为约束，采用灵活可靠的技术，迭代开发，从而达到最终目的。迭代开发就是将用户需求分割成细小的用户story，规定迭代周期并阶段性循环实现[2]。下文将通过敏捷软件开发方法的特性等来从复杂适应系统（Complex Adaptive System，CAS）理论着手分析其高适应性[3]。

## 敏捷软件开发方法特性

敏捷软件开发方法区别于传统方法的特性之一是拥抱快速变更、具有适应性而非预设性。传统方法如计划驱动开发方法等试图对一个软件开发项目在很长的时间跨度内做出详细的计划，然后依计划、合同、规格说明等文档进行开发，依赖于显式的文档化知识。

其实追溯传统方法的来源，其基本思路通常是从其他工程领域借鉴而来的，比如土木工程等。在这类工程实践中，通常非常强调施工前的设计规划。只要图纸设计得合理并考虑充分，施工队伍可以完全遵照图纸顺利建造，并且支持施工队之间的并行施工[4]。但是，软件开发与传统的工业工程有着巨大的差异。软件设计是一个复杂度极高的工程，而且土木工程师在设计时所使用的模型是基于多年的工程实践的，一些设计上的关键部分都是建立于坚实的数学分析之上，而在软件设计中，没有类似的基础，实践经验也积累不足。面对这样巨大的差异，仍旧生搬硬套传统工程借鉴来的开发方法自然存在许多固有缺陷，从而导致软件开发的效率低下、拒绝变更等后果。加之，软件的设计之所以难以实现，问题在于软件需求的不稳定，从而导致软件过程的不可预测、变更频繁。但是传统的控制项目的模式都是针对可预测的环境的，在不可预测的环境下，就无法使用这些方法。可是我们必须对这样的过程进行监控，以使得整个过程能向我们期望的目标前进[5]。于是需要引入敏捷软件开发这种具有适应性的方法，该方法使用反馈机制对不可预测过程进行控制。而敏捷软件开发方法本来就是是为了更好地适应变化而发展出来的，甚至能允许改变自身来适应变化。

## 复杂适应系统理论

复杂适应系统理论基本内容是把系统中的成员成为具有适应性的主体，即成员能够与环境以及其他主体进行交互作用，且交互过程中不断自我学习、积累经验并根据学到的经验改变自身的结构和行为方式[3]。主体是具有主管能动性的实体，能够通过自我学习进化、采取适应性行为，从而具有适应性。该理论核心思想可以概括为适应性造就复杂性，可见市场对于适应性方面的需求之高，不可不重视。

敏捷开发组织是由可以思维的人组成，能够根据具体情况具体分析并作出调整来适应环境。而且敏捷竞争环境与敏捷开发组织相互作用、交互式发展是环境演变的主要动力。行业竞争极为激烈，想要立于不败之地只能在充满持续性不可预见的商业市场中迅速变更以适应变幻莫测的大环境。敏捷软件开发方法以用较小的代价高效开发合格的软件为目标、以适应性和以人为导向的指导思想，没有固定的模式和方法，而是强调根据实际情况，采取甚至创造最适合具体环境的开发模式或管理办法来适应需求不断变化的软件开发环境。

# 敏捷开发方法与CMMI分析对比

## CMMI模型

CMMI（Capability Maturity Model Integration），即软件能力成熟度集成模型，其目的是帮助软件企业对软件工程过程进行管理和改进，增强开发与改进能力，从而能按时地、不超预算地开发出高质量的软件。CMMI是一个成功的、广泛使用的软件过程改进模型。CMMI强调通过对项目进行阶段性监控，提高项目过程可视程度，使软件研发过程愈发稳定，公司对各类项目的控制力度得以增强。过程的可见性及可控性要求是CMMI的重中之重[6]。

CMMI为改进一个组织的各种过程提供了一个单一的集成化框架，新的集成模型框架消除了各个模型的不一致性，减少了模型间的重复，增加透明度和理解，建立了一个自动的、可扩展的框架。因而能够从总体上改进组织的质量和效率。CMMI主要关注点就是成本效益、明确重点、过程集中和灵活性四个方面。其所依据的想法是：只要集中精力持续努力去建立有效的软件工程过程的基础结构，不断进行管理的实践和过程的改进，就可以克服软件开发中的困难。

## 敏捷开发方法

敏捷式开发的原则就是交付一些有价值的东西，强调面向任务目标快速非结构性的开发，所以一般来说是以非常小的周期快速的交付一些功能，从而使客户能够尽快的看到你的交付物。

敏捷开发技术有以下几个特点：1）测试驱动开发：测试驱动开发是敏捷开发的重要组成部分，我们知道系统的功能是由业务需求来的，程序员首先把需求分成很多部分，编写一个个测试，让每个测试都通过，测试的目的是什么，达到的预期的结果是什么，这样程序员首先对业务就非常理解了，知道要做什么，然后再去编写开发的功能代码。2）持续集成：版本不断丰富，持续改进。 3）重构：每一次的改动，程序员都可以对以前的代码进行重构，如果发现设计不合理都可以随时重构，但要注意范围要小，要能通过单元测试。4）结对编程：一个人写代码，一个人检查，互相讨论程序设计的问题，这样可以把沟通发挥到极致。 5）站立会议：每天早晨，项目组的成员都会抽出半个小时的时间大家在一起进行沟通，谈谈昨天工作当中出现的问题，以及今天的目标是什么。6）小版本发布：摆脱从前从需求确定下来，就开始闭门造车，也不论这样做好不好， 现在可以让需求一次一次的修改，每个小版本发布出去后让一小部分人去适用，适用当中反馈存在的问题，再进行修改。7）较少的文档：文档不需要详细，只需要精，大家一看就能理解项目的结构就行。8）以合作为中心：项目中的每一个人都可以知道系统里每一部分的代码，甚至看写的不合理，不管是谁写的，都可以进行重构。敏捷开发中，客户是与开发团队一起工作的，团队到客户现场进行开发或者邀请客户到团队公司里来开发。如果开发过程中有什么问题或者产品经过一个迭代后，能够以最快速度得到客户的反馈。9）自动化测试：为了减小人力或者重复劳动，所有的测试包括单元测试、功能测试或集成测试等都是自动化的，这对QA人员提出了更高的要求。他们要熟悉开发语言、自动化测试工具，能够编写自动化测试脚本或者用工具录制。10）可调整计划：敏捷开发中计划是可调整的，并不是像以往的开发过程中，需求分析->概要设计->详细设计->开发->测试->交付，每一个阶段都是有计划的进行，一个阶段结束便开始下一个阶段。而敏捷开发中只有一次一次的迭代，小版本的发布，根据客户反馈随时作出相应的调整和变化[7]。

## 分析对比

CMMI不是具体过程方法，而是一个开发团队过程改进的参考模型了框架。它是对现行最佳开发管理实践的抽象，它描绘了良好开发过程所具有的行为特征。因此在CMMI模型中，我们看到的是“应该做什么”，而缺少“如何做到”，即没有关于具体实施的方法论。既然CMMI并不限定采用哪种具体的过程方法，所以从根本上讲，实施CMMI采用何种方法论都应该是可以考虑的，敏捷软件开发方法也应在候选之列。而敏捷软件开发方法是一种或者若干种轻量级的软件开发方法学的集合。敏捷方法有多种不同的形式:如极限编程（XP）、Scrum、Crystal Methods、FDD、TDD等，每种方法各有侧重，但基本点是相同的，比如：它的基本特征是迭代的、增量式的开发；强调沟通和协作；重视人的作用；强调自主性和积极主动；强调效率和价值。敏捷宣言是这些方法的共同的价值取向。敏捷方法在敏捷原则的基础上提供了具体的工程实践和管理实践，用来帮助团队高效进行软件开发过程[6]。

此外，CMMI不仅仅包含开发过程，它是适用于整个开发组织的、全范围的参考模型。CMMI模型的作用范围，除了覆盖到开发组织的软件开发过程以外，还作用于开发组织的其它相关过程,比如采购、培训和过程改进等。因此，要想真正在全组织范围实施CMMI模型（成熟度3级以上），除了考虑要采用适当的软件开发过程方法外，还必须覆盖到组织的其他业务单元。对于以开发为主的组织而言，软件开发过程是最主要的过程，但不是全部。而敏捷软件开发方法主要关注软件项目级别的管理和实施。目前常见的敏捷方法主要关注于软件项目级的开发和管理，还没有注意到组织管理和运营层面的实践。因此，有关组织级的管理实践，特别是关于组织战略层面的,比如软件复用、知识管理，在敏捷开发方法中还没有有效的实践支持。A公司在实施敏捷方法的同时,也在探索和积累产品级管理、产品运营、知识管理等方面的实践,来一起建立组织级研发管理能力。

在团队文化方面，敏捷方法表现出更大的关注力度更关注于团队文化的建设。敏捷方法的本质是建立良好的自组织团队文化，并以此为支撑点,建立高效积极的组织文化。CMMI定义了做什么，而不是如何做，可以使用敏捷方法的实践去实现如何做。从当今的发展趋势来看，CMMI模型与敏捷方法是可以融合的。

# 结对编程

结对编程技术是一个非常简单和直观的概念，能达到事半功倍的工作效果，是极限编程的一个重要原则[8]。

## 结对编程技术

在传统开发过程中，每个开发人员负责系统的一部分开发任务，各自分工、互不干扰，而在结对编程的情景下，是两个开发人员结为一对，来共同完成同一个开发任务。但是，人与人之间的合作不是一件简单的事情——尤其当人们都早已习惯了独自工作的时候。

实施结对编程技术将给软件项目的开发工作带来好处，只是这些好处必须经过缜密的思考和计划才能真正体现出来。而另一方面，两个有经验的人可能会发现配对编程里没有什么技能的转移，但是让他们在不同的抽象层次解决同一个问题会让他们更快地找到解决方案，而且错误更少。因为两个程序员具有相同的缺点和盲点的可能性很小，所以当我们采用结对编程的时候会获得一个强大的解决方案，而这个解决方案恰恰是其它软件工程方法学中所没有的。

在实际实施过程中，两人在同一台计算机面前进行编程活动，一个同伴使用鼠标和键盘来编码，另一个同伴观察代码并考虑设计问题。一方发现问题时，暂停编码工作，双方讨论解决。结对者的角色根据需要来不断调整和交换，包括与别的结对组交换成员。这种结对方式可以让每个成员对项目有一个整体的认识，并且有利于团队建立起良好的合作和学习氛围。极限编程的其他几项重要原则包括：频繁地小规模发布软件，简单设计，集体拥有代码和持续集成。

结对编程开发的成果保证了一部分代码至少有两个人熟悉，编码的同时也在被复查，既改善了复查的效果也在开发团队内部传播了知识，比传统的Desktop Review更有意义。原始的那种让程序员互相检查程序的方式是一种相当形式化的手段，其实效果不好，因为检查者无法像开发时那样完全融入程序的逻辑之中，很难从中发现业务逻辑错误，多数情况下根本都不会仔细去看。事实上在两个人同时编写程序的过程中，每个人的注意力都会比自己一个人编程时更加集中，考虑问题更加全面，互相提醒需要注意的事项。

当两个人共同开发时，充分的交流所带来的效率是传统纸张文档所不可比拟的，也更容易深入业务逻辑，发掘问题并拓展解决方案的思路，增进团队间的友谊。而且通过大量的交流和沟通，两个人在技术水平和业务能力上的提升都远比一个人闷着头干活儿快得多。不要吝音于和自己的partner交流，因为就算是经验老道的程序员也能从新手身上学到有价值的东西，开阔思路，对公司整体员工质素的提升也是很好的促进[4]。

结对编程保证了任何一行程序都有两个以上的人了解，也就是说对于项目每一个模块都有buckup人选，这对公司抵抗人员流动风险的能力有极大的帮助。由于参与结对的成员是经常在变化的(通常半天可以重新组队一次)，所以对于同一块程序，所有相关人员全部离职的可能性相当低。对于一个需要修改的程序，只要有程序员是熟悉它的，无论过了多久，也比一个“新手”做起来快，一个对此程序完全陌生的程序员相较于backup人选来说必然需要更长的时间去挖清楚其中的逻辑和实现风格，一定程度上增加了公司维护程序、升级程序的成本。

## 问题与演化

当然，凡事有利有弊，结对编程也不例外。有人垢病结对开发的效率问题，认为传统的两个人各自码代码的速度肯定比同一时刻只有一个人打字的速度要快得多，自然结对编程的效率就相对比较低下了。但从整体开发角度，开发过程中也许表面上会慢，其实由两个人同时开发所能避免的错误和返工将是非常客观的，对于经验并不老道的程序员来说尤其如此，再算上维护的时间和成本，应当说结对开发的效率是只高不低的[10]。

不过，结对编程并不是什么灵丹妙药，其中有一个相当关键的要素就是程序员们都是抱着期望提升代码质量和自身素质的目的参与到项目中的，如果所有的程序员都是以混日子、磨洋工的风格工作，以完成任务了事为最高使命的话，那无论是么开发方式都不会有任何的帮助，整个开发团队只会步步走向崩溃[11]。

由于现在开发环境的变更发展，出现了演化变形的结对编程形式——远程结对编程和乒乓结对编程。远程结对编程，又称虚拟结对编程或分布式结对编程，是指两个程序员不在同一地点，通过协同编辑器，共享桌面，或远程结对编程的IDE插件进行的结对编程。远程结对编程破除了传统结对编程对地域上的限制，更加适应大环境的发展，但同时还引入了一些在面对面的结对编程中不存在的困难，例如协作的额外时延，更多的依赖“重量级”的任务跟踪工具，而不是“轻量级”的索引卡片，以及没有口头交流导致的在类似谁“控制键盘”问题上的混乱和冲突[12]。乒乓结对编程是指观察者编写失败的测试用例，驾驶者修改代码以通过该用例，观察者编写新的单元测试用例，等等。这个循环持续到观察者不能写出失败的测试用例，但是实际实施过程花费时间会比估计的计划要超出很多。

# 参考文献

1. 王海鹏(译者),基于组件开发,人民邮电出版社,2003.9
2. 崔康，Agile Developer创始人谈敏捷适应性，2013
3. 杨小东,徐琳. 敏捷组织的复杂适应性及其行为模式研究[J]. 微计算机信息. 2012(02)
4. 周莹莹. 敏捷软件开发技术研究[D]. 长春理工大学 2006
5. Alistair Cockburn，Agile Software Develpment，Addison Wesley/Pearsno，2003.11
6. Scrum敏捷项目管理实战;书籍作者:(美)肯·施瓦伯著，李国彪、孙媛译;清华大学出版社
7. 敏捷和CMMI的区别和特色，<https://www.douban.com/group/topic/18564288/>
8. Beck K. Extreme Programming Explained: Embrace Change [M] .2nd ed. Boston: Addison-Wesley, 2004.
9. 钟扬,刘业政,马向辉.小团队结对编程实践研究和重构[J]. 计算机技术与发展. 2007(11)
10. 杨涛.杨晓云(译者)，结对编程技术，机械工业出版社，2004.1
11. 王巨宏. 敏捷方法在一家互联网公司的应用和实践[D]. 北京邮电大学 2010
12. 臧正军. XP结对编程研究、改进和实践[D]. 东北师范大学 2008