ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2000/00(00)-0000-00
doi: 10.3778/j.issn.1673-9418.

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-89056056

看板软件开发过程的系统动力学仿真*

龚浩杰,刘博涵,张贺+,邵栋,荣国平

南京大学 软件学院, 江苏省 南京市 210093

+ Corresponding author: Phn: +86-18502531658, E-mail: hezhang@nju.edu.cn

The SD Simulation Model of Kanban Software Development Process*

Haojie Gong, Bohan Liu , He Zhang +, Dong Shao, Guoping Rong

Software Institute, Nanjing University, Nanjing 210093, China
+ Corresponding author: Phn: +86-18502531658, E-mail: hezhang@nju.edu.cn

Author. Title. Journal of Frontiers of Computer Science and Technology, 2000, 0(0): 1-000.

Abstract: With the introduction of Kanban, Kanban software development process became a popular development method between many agile development teams in recent decade. The more Kanban software development is widely used the more essential the optimization of Kanban development process is, so we need to provide a Kanban development process simulation and analyse the influence factor such as WIP (Work In Process) and staffs configuration. By using SD (System Dynamics) method, a research about the parameters of Kanban development process and the tendency of software products is provided. WIP has little impact on the Kanban development process from a macro view and developer to tester Ratio plays a vital part of Kanban development process.

Key words: Kanban software development; system dynamics; simulation model; software process

*The **** Foundation of China under Grant No.****, **** (基金中文完整名称); the **** Foundation of China under Grant No.****, **** (基金中文完整名称). [需中英文齐全].

Received 2000-00, Accepted 2000-00.

摘 要: 当看板被引入到软件开发后，近十年来看板软件开发过程成为许多敏捷开发团队中广受追捧的开发方法。随着看板软件开发的广泛使用，对于开发过程的优化和改进变得尤为重要，因此对于看板开发过程的建模和对于影响因素的分析是有必要的，如在开发产品(Work In Process, WIP)，人员配置等都会影响开发结果。借助于系统动力学仿真的方法，对看板软件开发过程中可能存在的变化因素和产品完成数的趋势进行了研究，分别定性和定量的分析了这些因素对开发过程的影响，最终发现 WIP 从宏观角度看对看板软件开发过程的影响不大，开发过程中开发人员与测试人员的合理配置对于开发结果有着巨大的影响。

关键词: 看板软件开发；系统动力学；仿真模型；软件过程

文献标志码：A 中图分类号：****

1 研究介绍

看板最早来源于丰田的生产方式(Toyota Production System, TPS)，主要是通过卡片的方式来实现非集中“拉动式”生产控制[1]。

看板旨在通过拆分和控制生产周期来降低资源的占用和浪费。例如：超市进货，当发现货架上的货物少于定值，就需要供货商及时补货，同时又不能大量堆积货物[2]。

看板被引入到软件开发中的目的是确保软件开发过程中只有下游阶段需要上游阶段产物时才提供开发完的产物，例如：当开发过程中测试空闲时，开发人员需要立即提供开发产物交付测试。在敏捷软件开发中使用看板过程管理能够保证质量和效率的有效结合，进而最大化的减少开发过程中在开发产品(Work In Process, WIP)。

看板方法由 Hiranabe 引入到实体的敏捷软件开发[3]，相似于丰田的看板生产方式，看板软件开发的特性例如：实体化，限制 WIP，连续流通性，“拉动式”，可视化，附着性等也随之被引入到开发过程中。

通过搜索相关研究 ,我们发现现阶段基于看板的软件开发更多的是依靠于软件开发者们的经验信息。而随着看板开发过程被广泛使用，此时开发人员不能仅停留在经验阶段，因此对于看板过程的优化和改进的研究就变得尤为突出。于是对于看板开发过程的建模和对于影响因素的分析是有必要的，这可以帮助开发人员分析并改进开发过程。

系统动力学(System Dynamics, SD)是软件过程仿真社区在过去的数十年里运用的最多的仿真建模方法，由 Abdel-Hamid首次应用于软件工程领域[4]。由于系统动力学仿真有反馈回路，结构性，定量性等特点，属于高抽象层次建模，更能够从宏观层面展示软件开发过程[5]，这正符合看板软件开发过程的特性。因此我们选择使用系统动力学方法仿真看板开发流程。

本文采用了系统动力学仿真建模对看板软件开发做深入剖析，目的在于直观的展示看板软件开发产品的增长趋势 ;探究 WIP 对于看板软件开发过程的影响；调整开发人员和测试人员数量对最终开发结果的影响。

本文的组织结构如下：第二章节讲述看板仿真的背景及相关研究；第三章节提出研究问题和研究方法；仿真模型的结果和分析在第四章节给出；我们在第五章节总结研究；并在最后一章提出展望。

Fig.1 Kanban development process

图 1 看板开发流程

2 看板仿真的背景及相关工作

2.1 看板开发流程

从开发者们的经验和实际的看板敏捷软件开发中可以了解到“待开发”，“开发中”，“待测试”，“测试中”和“已完成”是看板开发过程中主要的 5 个阶段[3]。如图 1 是看板软件开发的流程，敏捷开发团队通过会议方式确立需要开发的各个功能并制作成卡片，出现在“待开发”阶段，每张卡片都会通过开发和测试人员的拉动最终出现在“已完成”阶段。

2.1.1 待开发

待开发阶段是看板软件开发过程的开始阶段，所有已被确认的功能都会出现在这个阶段，并由开发人员通过拉动的方式移动到下一个阶段。虽然待开发阶段存在 WIP，但开发人员通常更关注开发中阶段的 WIP ,所以往往不限制待开发阶段的 WIP[3]。

2.1.2 开发中

在开发中阶段 ,开发人员根据各自的开发能力和其他原因从上一阶段选取各自将要进行开发的卡片并贴于此阶段。开发人员可以同时选择多张卡片进行开发，例如：开发 A 卡片时遇到障碍，先开发 B 卡片，当障碍得以解决，再返回 A 卡片进行开发。但是由于此阶段的 WIP 值的限制，开发人员不能无限制的同时选取多张卡片。

在 Cao 等人的研究中提到敏捷软件开发过程中开发人员的开发形式具有多样性[6]。不同的开发方式导致的开发速率会不同[7]。当开发人员选取需要进行开发的任务时，无论何种开发方式，由于每个任务不同的复杂度，开发人员的开发速率必然会不同[8]。

2.1.3 待测试

当开发人员完成某张卡片的开发后 ,将会把卡片从“开发中”阶段拉动到“待测试”阶段等待测试人员的测试。

2.1.4 测试中

当测试人员发现正在测试的产品越来越少时，测试人员会从上一个阶段选取等待测试的卡片贴于此阶段。这个阶段就如同上文提到的“超市补货”的例子。但如果开发阶段的开发速率较快而测试速率较慢时，在“待测试”和“测试中”就会出现堆积现象。由于敏捷软件开发具有灵活的人员分配特性，因此当测试阶段发生堆积，开发人员会参与到测试过程中[9]，这种策略类似于精益开发的弹性产能策略[10]。

2.1.5 已完成

此阶段是看板软件开发过程的最后一个阶段，当测试人员完成测试后并通过，把卡片最终贴在已完成阶段。在此阶段的卡片表示其功能已经完成开发，正在等待部署。

2.2 看板开发中资源的分配

从看板软件开发过程的测试阶段，我们可以看到开发人员与测试人员之间的分配会对产品的生产进度有一定的影响。在经验信息和现有的研究中都纷纷指出敏捷开发中的开发与测试比(也称测试开发比，Developer to Tester Ratios)是近十年来一个较为

热门的话题[11,12,13]。

Chris 等人在文章中[11]提到，根据经验开发的初始阶段由于测试相对较为空闲，因此没有一个固定的开发测试比。

霍泰稳[12]在一次近千人的互联网测试交流大会的报告中指出，许多互联网公司的测试人员在新的项目经理到来之时都会被问及他们是如何做测试的，开发测试比是多少等等，而这往往是每个项目经理参与开发时的第一件事。同时他还提到，开发测试比其实反映的是公司内部开发和测试结构的关系。总而言之，大到公司，小到开发团队，人们都会特别关注开发测试比，因此对于开发测试比的研究就被突显出来。

2.3 系统动力学仿真及其开发工具

系统动力学方法最初是由麻省理工大学的 Jay Forrester 于 1961 年提出的连续性系统建模方法[14]，并且由 Abdel-Hamid 首次应用于软件工程领域[4]。因果关系图(causal loop diagram)和流图 (flows diagram)是系统动力学构建动态模型的两种机制。定义变量之间的因果关系能够表示出系统的反馈机制，流图则采用栈/库存(stock)，流率(flow rate)，水平(level)，常量(constant)，辅助变量(auxiliary) 和延迟(delay)等来反映系统的逻辑结构[5]。由于系统动力学有着反馈回路，结构性，高抽象性等特点，因此其优势在于能够准确地捕捉反馈的效果并清晰地表示动态变量之间的关系，而这正好符合看板软件开发过程仿真的需求。

在系统动力学开发工具 Vensim 中，关系网通常使用单箭头的直线或弧线来连接，流图则是使用单箭头双线来串联整个流程。流图中的栈/库存使用的是矩形框，两头分别是库存的输入流率和输出流率，常量和辅助变量通过关系网来影响整个流程，并由延迟等来横向调整过程(时间轴)，最终加入时间单位(天、小时等)以实现系统动力学仿真。

2.4 看板仿真相关工作

Richard 等人在他们的研究中[15]按照通用看板方法，仿真了一个特定看板过程用来在快速响应的环境中支持一次软件开发，并通过与传统开发过程作对比，以确定看板软件开发过程是否有价值。

他们的研究结果表明了看板软件开发过程比传统开发过程更有价值。当我们了解了看板开发的价值，则我们需要考虑哪些因素会影响看板开发过程。

3 研究方法

3.1 研究问题

当前阶段对于看板开发过程的仿真并不多，且缺少对于“影响看板开发过程的因素”的相关研究。为了能够定性的分析看板软件开发过程的，我们首先需要确认看板软件开发过程产品的增长趋势，通过这种趋势分析有哪些因素影响看板软件开发过程。因此我们提出了以下研究问题：

研究问题 1：通过看板软件开发过程开发的产品，开发结果呈现什么趋势？

通过仿真模型直观的展示使用看板软件开发过程开发的结果。是否线性，有哪些参数影响了开发的结果。

研究问题 2 :WIP 是否影响了看板软件开发过程？

“开发中”阶段的 WIP 是否会影响正在开发的数量。这种影响呈现出什么样的趋势。

研究问题 3：在看板软件开发过程中开发测试比的调整对最终结果有什么影响？

看板软件开发过程中是否需要调整开发测试比。

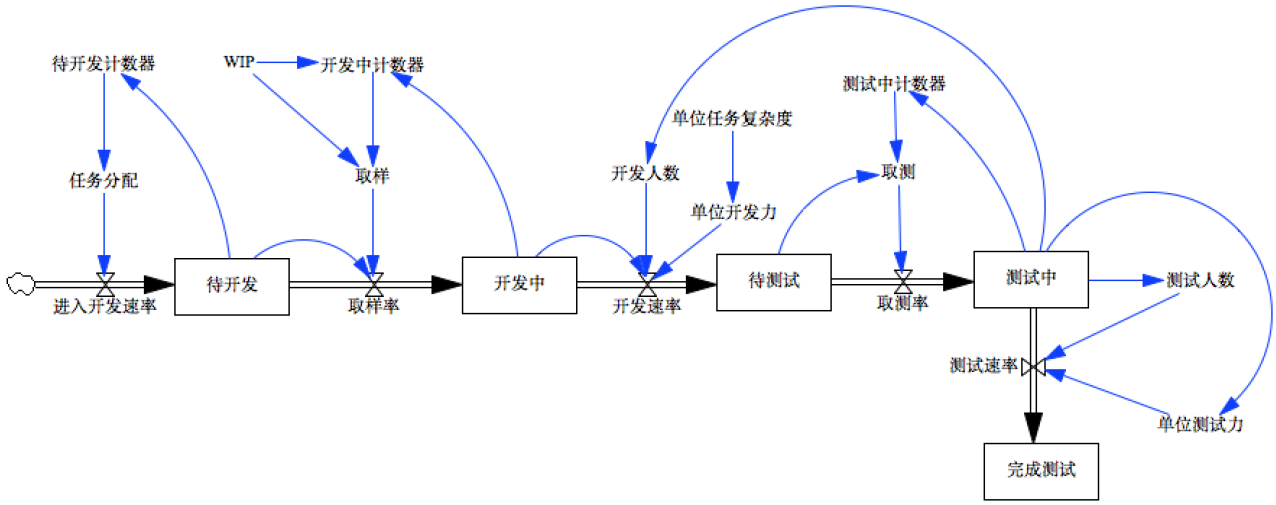

Fig.2 Kanban SD simulation

图 2 看板的系统动力学仿真图

人员是否被合理分配对最终的开发结果有什么影响和差异。这种差异是否明显。

3.2 建模过程

本文使用系统动力学建模工具 Vensim 来仿真看板软件开发过程。

首先我们需要确定我们的仿真对于数据的研究角度，对于相同的项目，测试的“任务”(任务的单位为“个”)应该是相同的，可以研究在相同任务的情况下不同的策略所需要的时间的差异。但是考虑到如果定义的任务量较少会导致仿真的时间区域较短，普适性可能降低，因此我们选择了相同的时间情况下讨论完成的任务数。另一个原因是，对时间的确定比对任务数的确定更可控。

在模型的开发过程中，我们遇到了 3 个问题，通过小组讨论等方式规划解决方案并成功解决这些问题。

如图 2 所示 ,基于看板软件开发过程的 5 个阶段我们设置了 5 个方程变量(积分量)分别是“待开发”，“开发中”，“待测试”，“测试中”和“完成测试”。考虑到看板中每个阶段都不能无限制的粘贴卡片，我们设置了计数器，当两个阶段间（A 阶段和 B 阶段）出现堆积现象，即 B 阶段的计数器达到了预先设置的阈值，则会触发停止 A 阶段继续向 B 阶段输入其产物的操作。

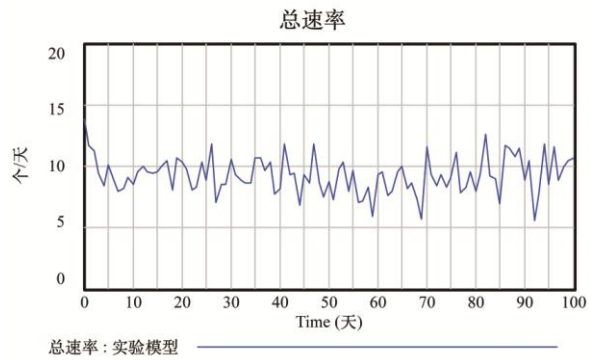

Fig.3 Development rate synthesis

图 3 开发速率合成

在建模过程中的第二个问题是 ,每个开发人员从“待开发”中取得的任务复杂度各不相同，且每个人的开发方式和开发能力的不同，导致每个人之间开发速率各不相同。借鉴于 GENSIM2.0 中提到的速率合成技术[16]，我们将任务差异和个人差异叠加在一起，即全部转换为了开发速率的差异。这种

差异在仿真中是通过输入一个满足正态分布的随机量来体现的。为了得到开发总速率，我们将不同的开发速率进行合成用于检查满足正态分布的不同速率的合成是否也满足正态分布，并最终得到图 3 所示的开发总速率。

建模过程中遇到的第三个问题是开发测试比自动分配问题。我们可以从 Chris 的文章中[9]了解到敏捷开发团队会关注于开发人员预测人员之间的人员分配。而当前现状是，人员分配更多的依赖于经验，并没有一种统计数据来支撑。因此我们需要解决的是通过定性和定量分析开发测试比的调整对开发结果的影响。如表 1 所示 ,为了解决这个问题，我们设立 3 种开发人员测试比，总人数为 6 人。其中不调整与调整的比较目的在于确认开发测试比对开发结果是否有影响；比较调整程度的目的在于量化这种影响有多大。

Table1 The rate of developers and testers

表 1 开发测试比

状态	初始比	调整比
不调整	5:1	5:1
调整为 4:2	5:1	4:2
调整为 3:3	5:1	3:3

基于上述设计中的思考和系统动力学建模方法的应用[4]，我们最终形成了看板开发过程的 SD 模型，如图 2 所示，上述问题中涉及的参数方程及积分量方程在表 2 展示，表 2 对应于图 2 中不同阶段的状态变量、计数器、单位任务复杂度和人员配置。需要注意的是，表 2 中“任务”的单位为“个”，开发人数和测试人数方程中的 N 为自定义数值表示允许正在测试中的任务的最大值，调整 N 值是触发开发测试比调整的关键，且我们默认开发测试比为 5:1(注：比为开发人员:测试人员，下同)当触发调整时则开发测试比为 3:3。另外由于表 2 容量的限制，我们将简写参数名称，对照如下：进入开发速率(IDR)；取样率(SR)；开发速率(DR)；取测率(PR)；测试速率(TR)；开发中(DIG)；测试中(TIG)。

Table2 Main parameters equation

表 2 主要参数方程

参数名	参数方程
待开发	$\int_0^t (IDR - SR)dt$
开发中	$\int_0^t (SR - DR)dt$
待测试	$\int_0^t (DR - PR)dt$
测试中	$\int_0^t (PR - TR)dt$
完成测试	$\int_0^t TRdt$
开发中计数器	IF THEN ELSE(DIG<WIP , DIG , WIP)
单位任务复杂度	RANDOM NOMAL(1 , 10 , 5 , 20 , 4)
开发人数	IF THEN ELSE(TIG<N , 5 , 3)
测试人数	IF THEN ELSE(TIG<N , 1 , 3)

4 仿真结果及分析

4.1 使用看板开发软件产品的完成度趋势

从图 4 中我们可以看出有开发测试比调整的看板软件开发呈现出一种稳步上升的趋势。但是同样我们也会注意到这种稳步上升的趋势中伴有抖动存在，因为看板开发过程中可能会存在某个阶段中发生堆积，而这种堆积的根本原因就在于开发和测试阶段人员调整不均使得测试总速率发生变化，对比 Richard 等人的研究[15]，这种不均在极端情况下表现出一段时间的完成数的缓慢增长。

但是值得注意的是 ,真实情况可能会存在人员转换时的开发与测试能力的转换损耗(长期处于开发状态的开发人员一时无法完全投入测试状态，反

之亦然) ,这种转换损耗可能需要大量的数据拟合最终得出转换方程，由于我们这次研究并不能完整的包含所有影响看板开发结果的因素，因此我们没有重点关注此问题。

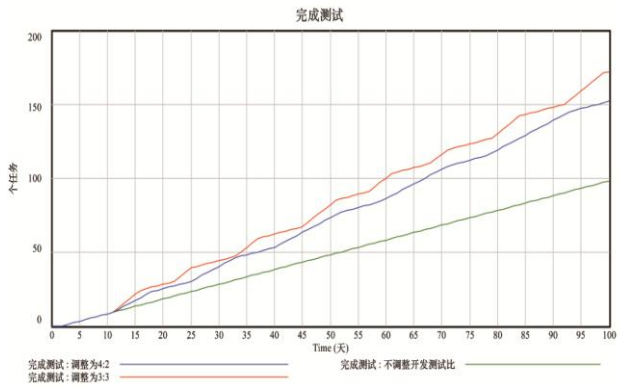

Fig.4 The trend of product completion and the impact of developers and testers rate

图 4 产品完成趋势及开发测试比的影响对比图

与此同时，我们还可以观察到这种堆积-疏通关系导致的抖动并不是完全的不利于开发，相反，当发现开发完成度的改变很平缓的时候，才是最值得开发人员的重视，因为此时可能存在某个阶段的工作进度缓慢。

4.2 WIP 对看板开发过程的影响

我们可以直观的在图 5 中观察到 4 种不同的 WIP 展现出了 3 个不同的趋势，说明其中必有 2 种 WIP 导致完成度趋势发生了重合，因此我们需要数据辅助展示 WIP 对开发过程的影响。图 6 展示了开发第 100 天时 ,不同的 WIP 对于任务完成数的影响，值得注意的是，开发时间的 100 天是为了更直观的对比不同 WIP 值在同一段开发时间内的任务完成数而设定的 ,我们可以看到 WIP=4 与 WIP=6 第 100 天任务完成数不同，而 WIP=10 与 WIP>10(包括 WIP=15)是完全相同的。结合图 5 中的第 100 天的任务完成数，我们可以看出 WIP<6 时 WIP 对于看板软件开发过程是有影响的，而当 WIP>10 时 WIP 对于看板开发过程是没有影响的，而在 6<WIP<10 这个区间通过二分法必然可以找到一个临界值使得当 WIP>临界值时 ,WIP 对于看板开发没有影响，这种现象值得我们做进一步思考。

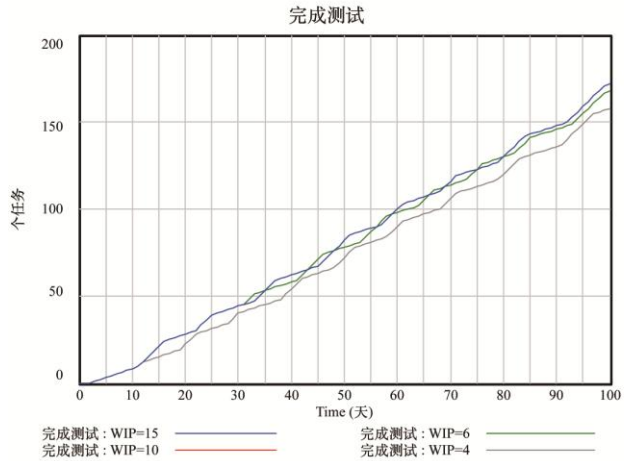

Fig.5 The effects with different WIP

图 5 不同 WIP 的影响

为了探究其原因 ,我们采访了工业界熟悉看板开发的人员及资深导师(采访包括是否了解看板中的 WIP ;看板中的 WIP 是否会对看板开发产生影响；调整 WIP 对于开发结果是否具有较大的影响等)，他们给出的结论是：根据开发经验，WIP 对于看板开发过程没有太大的影响。对于 WIP>临界值来说，这种经验的确吻合仿真结果，但是对于 WIP<临界值的部分，我们做了深入探究，来解释“当 WIP<临界值时会对任务完成度产生巨大的影响”这种现象。通过图 4 的反复观察，我们发现，当 WIP 值很小，任务完成数变化表现地很平缓，即某个阶段速率受到影响。我们发现，当 WIP 值较小，而此时开发人员的实际每天开发速率是不变的，即开发中的任务少而开发速率快，使得开发经常被停滞(如图 7 所示)，严重的影响了整体开发进度。这也同时验证

了当完成度的改变变得平缓，即完成速度不变时，则某个阶段可能存在工作进度的缓慢。

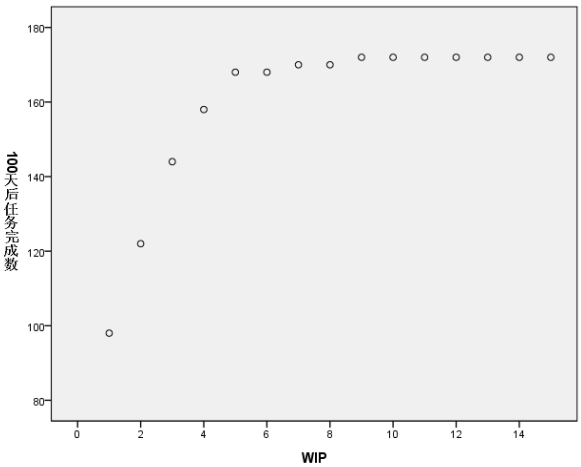

Fig.6 The number of completed tasks at the 100th day with different WIP

图 6 不同 WIP 在 100 天后任务完成数

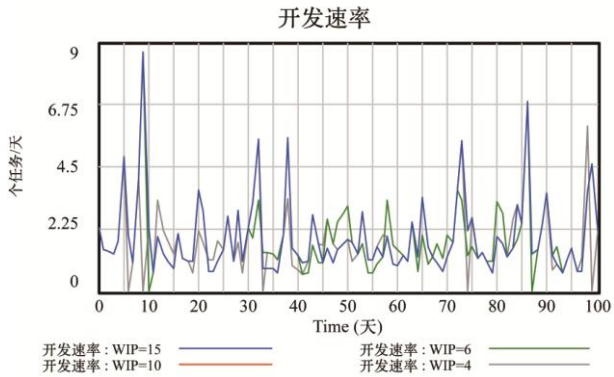

Fig.7 Development rate

图 7 开发速率

真实的看板开发过程，项目经理会根据开发速率的快慢及时的调整 WIP 值，使得开发人员处于忙碌状态，因为开发团队不会使开发人员闲置。在整个项目开始阶段，当开发人员有较高的开发速率时，其对应的 WIP 临界值较高，因此项目经理需要将 WIP 提升，若依然低于临界值，则继续提升，直至大于临界值。则此后的开发阶段由于每个开发人员真实的开发速率不会发生巨大的变化(根据个人能力提升存在较小增长)，再提升 WIP 值将会变得多余。因此在真实情况下，大部分的开发阶段 WIP 值对于看板软件开发过程不会有太大影响。

仔细观察图 5 我们还可以发现在长达 100 天的开发时间内，不同的 WIP(小 WIP 值除外，有经验的项目经理并不会设置过小的 WIP 值)对于最终开发结果的差异并不是十分巨大。而大部分的看板开发过程周期并不会很长。我们认为这也是另一个为什么 WIP 对开发影响不大的原因。

4.3 看板开发过程中开发测试比对开发结果的影响

为了研究这个问题，我们在模型中多次改变当测试出现堆积时开发人员与测试人员的配比，如表 1 所示，并仿真出这 3 种人员分配方案下对于开发结果的不同趋势。

从图 4 中我们看到，在开发初期，这 3 种配比没有差别，原因是此时的测试任务并没有发生堆积，开发人员与测试人员的配比为原先设定的 5:1，特别是接近原点的时间段，由于此时的开发任务并没有完成，因此测试任务处于“缺货”状态，测试人员闲置。于是我们可以得出结论，在开发初期，并不需要过多的加入测试人员，相反应该更多的着重于开发阶段。

随着开发时间的推移，开发完成的任务渐渐增长，此时就存在开发人员拉动到测试阶段(待测试及测试中)的任务发生堆积。从图 4 中可以看出，当测试阶段发生堆积不调整开发测试比将会严重影响到整体开发进度。因此我们得出结论，开发测试比的调整对开发结果有着巨大的影响。

定性的了解了开发测试比的影响后，我们需要深入分析这种影响随着测试开发比值不同有着什么变化。图 4 中当测试阶段发生堆积时，测试比调

整为 3:3 与 4:2 是不同的，在第 15 天至 45 天时间段内，我们发现，这两种配比的差距并不是十分明显，仅仅只是路径不同而已，其深层原因是，开发前中期阶段发生堆积次数较少，此时解决堆积的速率不同对总的开发进度影响并不明显。但随着时间的变化，可能发生堆积的次数变多，那么解决堆积的速率将对开发结果起着重要作用。我们也可将其理解为疏通堆积的速度，疏通的原理是通过减少开发人员来及时降低开发阶段的输出，即测试阶段的输入，并增加测试阶段的输出速度，依靠测试阶段输入输出的差值来实现快速地减少堆积任务。真实的开发情景下，参与开发的人员在项目开始阶段无法保证各阶段完全不会出现堆积现象，而开发测试比的调整决定了堆积疏通的快慢。综上，找到一个最高效的开发测试调整比是非常有必要的。

另外我们需要注意的是，根据上述结论，我们可以推论，存在一个开发测试比范围，使得开发人员与测试人员数量比在这个范围内，能够保证看板开发过程中 5 个阶段不产生堆积，也可称为资源不浪费[17]。其临界值就是当开发完成，交付给测试且测试阶段恰好等于既定阈值，此时测试人员刚好完成当前测试并进行下一个测试任务。在此范围内再调整开发测试比，将对最终的开发结果没有影响。

5 总结

本文的目的在于通过系统动力学方法仿真看板软件开发过程，并修改其中的变量找出影响看板软件开发过程最终完成数量的因素，这些因素的影响有多大，以及更深层次的分析其内在的根本原因并给出自己的推论。通过这些研究成果，我们可以提供看板软件开发人员一种更直观的看板开发趋势，并帮助开发人员优化软件开发过程，预判可能存在的潜在因素。

通过多次仿真，并比较仿真结果，我们可以得出如下结论：

1. 看板软件开发过程并不是一种匀速开发过程。相反，当开发过程的速率呈现匀速时，这种现象更值得开发人员关注。
2. WIP 值在较低的情况下，会影响开发进度，开发人员需要多次调整 WIP 值，直到满足开发团队的开发现状。
3. 当 WIP 值调整到一个较为符合团队开发属性的情况下，开发人员不需要过多的考虑 WIP 的影响，也不需要经常调整 WIP 值。
4. 开发测试比对看板开发过程存在影响，开发人员需要及时的根据开发情况，找到并调整到一个合理，高效的开发测试比，以保证看板开发过程的最优状态。

在此需要指出本文的欠缺之处。由于本文缺少实际开发案例的数据，因此这是一种宏观的看板软件开发过程的趋势及分析，对于某些特殊开发情况可能存在偏差；尽管借助于对比采访获得的经验信息，以证实我们的结论和推论，但被采访的人员可能存在某些偏见，这可能会对结论的有效性产生影响；由于看板开发的整体架构涉及很多其他内容，包括需求分析，需求变更，开发人员与测试人员转换损耗，客户参与开发过程的影响[18]，返工等，因此我们的看板仿真模型并不能覆盖所有情况。

6 工作展望

由于研究中存在不足和需要改进的地方，因此我们需要对现有的工作提出更高的要求，并将在未来付诸实现。

本次研究通过分别更改 WIP 值与开发测试比观察其对看板开发结果的影响，基于此结果，我们更想了解同时更改两项参数对看板开发结果的影

响。下一步，我们将研究如何同时调整这两组参数，从而使看板软件开发过程更优。

在真实的看板软件开发过程中会存在更多的参数来影响看板开发的进度。真实情况下软件公司中开发和测试岗位相对固定，两者所需的技能也有较大的差别，理想化的切换角色具有一定的误差，我们需要考虑适应期的问题，以及同时掌握两类技能的转换。同样不同参与者参与开发的过程及编程策略也需要我们细化。下一步，我们将考虑开发人员与测试人员转换损耗，客户参与开发过程，结对编程[19]及返工等情况下的看板软件开发过程。

参考文献:

[1] Yasuhiro M. Toyota Production System[M]. Springer US, 1994.

[2] 汪厚俊. 基于看板管理方法的敏捷软件开发[J]. 电子技术与软件工程, 2016(17):60-60.

[3] Hiranabe K. Kanban Applied to Software Development: from Agile to Lean[J]. 2008.

[4] Abdel-Hamid T, Madnick S E. Software project dynamics: an integrated approach[M]. Prentice-Hall, Inc. 1991.

[5] Gao C, Zhang H, Jiang S. Constructing hybrid software process simulation models[C]. International Conference on Software and System Process. ACM, 2015:157-166.

[6] Cao L, Ramesh B, Abdel-Hamid T. Modeling dynamics in agile software development[J]. ACM Transactions on Management Information Systems (TMIS), 2010, 1(1):1-26.

[7] 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, 17(11):160-163.

[8] Beck K. Extreme programming explained: embrace design[J]. Addison-Wesley, Boston, MA. 2000.

[9] Laurie Williams, James Tomayko. Agile Software Development[J]. Computer Science Education, 2002, 12(3):167-168.

[10] 鲁艳青，方桃红. 浅析精益产品开发体系[J]. 工会博览•理论研究, 2009(7):125-125.

[11] Chris Sims. The Correct Ratio of Agile Testers to Developers? It Depends[EB/OL]. [2009-01-05]. https://www.infoq.com/news/2009/01/tester-to-developer-ratio.

[12] 霍泰稳. 研发团队中最合适的开发测试比是多少？[EB/OL]. [2010-11-18] http://www.infoq.com/cn/news/2010/11/discussion-dev-tester-rate/

[13] Kathy Iberle. Estimating Tester to Developer Ratios[J]. The Pacific Northwest Software Quality Conference, 2010.

[14] Jay W. F. Industrial Dynamics[M]. Cambridge, MA : MIT Press, 1961.

[15] Richard T, Raymond M, Dan I, et al. Modeling Kanban Processes in Systems Engineering[C], International Conference on Software and System Process. ACM, 2012:23-27.

[16] Khosrovian K, Pfahl D, Garousi V. GENSIM 2.0: a customizable process simulation model for software process evaluation[C]. Software Process, 2008 International Conference on Making Globally Distributed Software Development A Success Story. Springer-Verlag, 2008:294-306.

[17] Ikonen M, Kettunen P, Oza N, et al. Exploring the Sources of Waste in Kanban Software Development Projects[C]. Software Engineering and Advanced Applications. IEEE, 2010:376-381.

[18] Highsmith J, Cockburn A. Agile software development: the business of innovation[J]. Computer, 2002, 34(9):120-127.

[19] Williams, L.A, Kessler, R R. The effects of "pair-pressure" and "pair-learning" on software engineering education[C]. Software Engineering Education & Training. IEEE, 2000:59-65.

GONG Haojie was born in 1992. He is a M.S. candidate at Nanjing university. His research interests include software process, empirical software engineering, etc.

龚浩杰(1992-)，男，南京大学硕士研究生，主要研究领域为软件过程，实证软件工程等。目前发表了 1 篇国际期刊/会议/章节。

LIU Bohan was born in 1991. He is a Ph.D. candidate at Nanjing university. His research interests include software process, empirical software engineering, etc.

刘博涵(1991-)，男，南京大学博士研究生，主要研究领域为软件过程，实证软件工程等。目前发表了 2 篇国际期刊/会议/章节。