**Modeling and analysis论文解决的问题：**

软件的开发压缩在尽可能短的时间内同时避免多次发布而造成的问题

**方法原理：**

1. SRGM模型【1】即利用在测试过程中搜集的错误数据来在软件测试过程（FDP）中出现的随机行为，有理由假设每一次错误都很重要，每一个之间的间隔都是互相独立、互不影响的。缺点：大多集中一个于只有一个版本的软件开发进程
2. NHPP模型【2】：即非齐次泊松过程模型，一个广泛应用到大量软件测试开发项目和预测可靠性，以及做重大决定之中的模型，（论文中并没有详细说明）
3. 适用于本篇论文的模型：基于上述模型，优化上述缺点的模型

**论文步骤：**

1.）现有的SRGMs大多集中于软件开发进程只有一个版本。因此有必要进行调查可靠性的变化来自于正在进行的版本，也就是一个相当复杂的问题。在这方面进行了几项研究在文献中。例如，Smidts等【3】(1998)应用软件执行可靠性评估的前一个版本的故障数据对当前的一个版本，并开发了一个早期的预测模型使用主观和/或客观的贝叶斯框架来自旧项目的数据。胡等(2011)在哪个软件开发团队开发、测试和发布软件按版本进行了修改，并提出了一些实用的方法假设。Smidts, C.【4】(2011)后来提出了一个聚焦的模型在OSS上，将试工作随时间的变化视为一个驼峰曲线。最近，Pachauri等人(2015)提出了一种建模方法框架考虑了s型弯道的故障减少因素并将该模型扩展到多版本软件中。许多其他因素，如故障严重程度和测试资源，也被纳入多版本软件的建模中。描述错误纠正难度的不同严重程度是Garmabaki【5】等人(2011)在升级过程中考虑，谁认为以前休眠断层的严重性版本可能在后续版本中更改。Kapur【6】等(2012)在以前发布的版本中发现了一些休眠的错误是否可以在后续版本的测试中删除，并提出考虑测试资源的SGRMs链采用柯布道格拉斯生产函数进行优化升级建模和发布时间预测。

2）缺点：移动总是需要时间的，而且是意料之中的在任何给定的时间内，被移除的故障数小于预计检测到的故障数量(Gokhale等，2004)【7】

3）校正方法：一些研究人员也考虑到了错误校正处理(FCP)并使用修正的故障数据来表示修正时间延迟。FDP和FCP的建模都需要更多的信息从软件测试记录，但改善估计和预测结果。Schneidewind提出了一种FCP方法使用恒定延迟FDP的建模(Schneidewind, 2001)【8】。他假设错误校正率与……成正比故障检出率。还有许多已经检测到但仍然存在的错误在某些应用中，模型通常需要等待修正低估代码中剩余的错误。Lo和黄2006【9】)提出了一种综合的检测分析方法以及使用微分方程进行校正的过程。吴等人al.(2007)【10】将Schneidewind的模型扩展到一个连续版本用时延函数代替常数时延。基于上述基于nhpp的FDP和FCP建模框架，LSE和MLE(极大似然估计)已经提出了一些办法。此外，胡【11】等(2007)开发了一种神经网络配置方法表征预测重复离散度的因子用于FDP和FCP的同步建模。黄【12】(2010)后来应用排队模型来描述这两个过程具有多个更改点。合并测试工作功能且调试不完善，Peng et al.(2014)【13】近期提出a用于分析这两个过程的框架。最近，Gaver和Jacobs(2014)【14】提出了一种基于不同失效模式假设的队列模型。

4）本论文针对上述所有问题提出一个适用于多版本发布软件的发布模型

该模型步骤如下：第一步是对故障流入进行量化，即期望的检测和校正故障数，即m d (t)、m c (t)。第二是不同类型时滞模型的比较。本文采用的比较标准是均方误差(MSE)和r2，即衡量模型拟合优度的多重相关系数的平方。选择MSE较小或r2较高的模型。调整后的r2也被用来测量额外参数的影响。最后，在上述成本模型的基础上，通过最小化Eq.(17)中的成本模型得到最优升级时间

**实验：**

使用前面提到的三个版本的数据集表单Mozilla对所建议的模型进行测试，得到模型性能结果。利用MATLAB中的非线性最小二乘法，估计如表1所示。训练了指数时滞模型和伽玛时滞模型。使用了三个单版本模型对数据进行了估计，没有考虑到对之前版本的错误进行修正。上述结果中的MSE和r2标准都表明，所提出的模型在FDP和FCP方面都比单版本模型具有更好的拟合优度。随着每个版本的升级，多版本模型的适用性会增加。因为伽马分布的指数分布是一种特殊的形式与α= 1,该模型与更灵活的伽马时间延迟数据集提供了更好的回归,也表示较低的MSE和大R 2,比指数时间延迟模型(见表2和3)。图4描述了上述模型在测试周内的相对误差。可以看出，指数时滞模型在首次发布前对FDP的预测存在低估的趋势。在首次发布之后，指数模型和伽玛时间延迟模型在预测FDP方面趋于相似。在FCP早期，指数时滞模型的预测更可靠，而伽玛时滞在FCP后期的预测中往往更准确。

参考文献：

【1】Amin, A. , Grunske, L. , Colman, A. , 2013. An approach to software reliability predic-tion based on time series modeling. J. Syst. Softw. 86 (7), 1923–1932 .

【2】Goel, A.L. , Okumoto, K. , 1979. Time-dependent error-detection rate model for soft-ware reliability and other performance measures. IEEE Trans. Reliab. R-28 (3), 206–211 .

【3】【4】Smidts, C. , Stutzke, M. , Stoddard, R.W. , 1998. Software reliability modeling: an ap-proach to early reliability prediction. IEEE Trans. Reliab. 47 (3), 268–278 .

【5】Hu, Q.P. , Peng, R. , Xie, M. , Ng, S.H. , Levitin, G. , 2011. Software reliability modelling and optimization for multi-release software development processes. In: Pro-ceedings of *2011* IEEE International Conference on Industrial Engineering and Engineering Management. New York. IEEE, pp. 1534–1538

【6】Kapur, P.K. , Pham, H. , Aggarwal, A.G. , Kaur, G. , 2012. Two dimensional multi-release software reliability modeling and optimal release planning. IEEE Trans. Reliab. 61 (3), 758–768 .

【7】Gokhale, S.S. , Lyu, M.R. , Trivedi, K.S. , 2004. Analysis of software fault removal poli-cies using a non-homogeneous continuous time Markov chain’. Softw. Qual. J. 12 (3), 211–230 .

【8】Schneidewind, N.F. , 2001. Modelling the fault correction process. In: Proceedings of the 12th International Symposium on Software Reliability Engineering, pp. 185– 190 .

【9】Lo, J. , Huang, C. , 2006. An integration of fault detection and correction processes in software reliability analysis. J. Syst. Softw. 79 (9), 1312–1323 .

【10】Wu, Y.P. , Hu, Q.P. , Xie, M. , Ng, S.H. , 2007. Modeling and analysis of software fault detection and correction process by considering time dependency. IEEE Trans. Reliab. 56 (4), 629–642 .

【11】Huang, C.Y. , Hung, T.Y. , 2010. Software reliability analysis and assessment using queueing models with multiple change-points. Comput. Math. Appl. 60 (7), 2015–2030 .

【12】Huang, C.Y. , Lyu, M.R. , 2005. Optimal release time for software systems considering cost, testing-effort, and test efficiency. IEEE Trans. Reliab. 54 (4), 583–591 .

【13】Peng, R. , Li, Y.F. , Zhang, W.J. , Hu, Q.P. , 2014. Testing effort dependent software reli-ability model for imperfect debugging process considering both detection and correction. Reliab. Eng. Syst. Saf. 126, 37–43 .

【14】Gaver, D.P. , Jacobs, P.A. , 2014. Reliability growth by failure mode removal. Reliab. Eng. Syst. Saf. 130, 27–32