But Financial Operations Is Not WebOps

但金融运营不是WebOps

Financial services firms are hiring DevOps engineers to automate

releases and to build Continuous Delivery pipelines, and Site Relia‐bility Engineers (patterned after Google) to work in their operations

teams. But the jobs in these firms are different in many ways,

because a global bank or a stock exchange doesn’t operate the same

way as Google or Facebook or one of the large online shopping sites.

金融服务公司正在雇用DevOps工程师来实现自动化发布并建立持续交付流水线，以及在其运营团队中招聘站点可靠性工程师（SRE,以谷歌为模式）。但这些公司的工作在很多方面都不一样，因为全球性银行或证券交易所的运作方式和谷歌、Facebook或大型在线购物网站不一样。

Here are some of the important differences:

1. Banks or investment advisers can’t run continuous, online

behavioral experiments on their users, like Facebook has done.

Something like this could violate securities laws.

以下是一些重要的区别：

1、与Fackebook不同，银行或者投资顾问公司不能对用户进行持续的、在线的行为测试，否则会违反相关证券交易法。

2. DevOps practices like “Monitoring as Testing” and giving developers

root access to production in “NoOps” environments so that they can run the systems themselves work for online social media startups, but won’t fly in highly regulated environments with strict requirements for testing and assurance, formal release approval, and segregation of duties.

2、来自在线社交媒体公司的一些DevOps实践，如“监控即测试”，以及在"NoOps"环境下给予开发人员对生产系统的root权限，让他们能自行运行系统。这些在高度监管的环境下是不会发生的，这种环境要求严格的测试和质量保证，正式的发布许可以及职责划分。

3. Web and mobile have become important channels in financial

services—especially in online banking and retail trading—and

web services are used for some B2B system-to-system transactions.

But most of what happens in financial systems is systemto-

system through industry-standard electronic messaging protocols

like FIX, FAST, and SWIFT, and low-latency proprietary

APIs with names like ITCH and OUCH. This means that tools

and ideas designed for solving web and mobile development

and operations problems can’t always be relied on.

3.Web和移动端已经成为金融服务的重要渠道，特别是在在线银行和零售交易。另外web服务也被用于一些B2B系统间交易。但是，金融系统中最多的还是通过行业标准的通信协议，如FIX, FAST和SWIFT,以及ITCH和OUCH等专属低延迟API来进行系统间交互的。这就意味着未解决Web和移动端的开发和运维问题而设计的工具和概念并不是总是可依靠的。

4. Continuous Deployment, where developers push changes out to

production immediately and automatically, works well in stateless

web applications, but it creates all kinds of challenges and

problems for interconnected B2B systems that exchange thousands

of messages per second at low latency, and where regulators

expect change schedules to be published up to two quarters

in advance. This is why this book focuses on Continuous Delivery:

building up automated pipelines so that every change is tested

and *ready* to be deployed, but leaving actual deployment of

changes to production to be coordinated and controlled by

operations and compliance teams, not developers.

4、持续部署，如开发将变更立即且自动地推送至线上地这种方式，在无状态应用中工作地非常好，但它对互联地B2B系统造成了各种问题和挑战。这些系统以极低的延迟每秒交换数千条消息，而管理者希望上线计划能提前到两个季度。这也是为什么这本书主要关注于持续交付：构建自动化的流水线让每个变更得到测试并为部署做好准备，但是实际的生产系统变更部署让运营和合规团队来协调和控制，而不是开发人员。

5. While almost all Internet businesses run 24/7, many financial

businesses, especially the financial markets, run on a shorter

trading day cycle. This means that a massive amount of activity

is compressed into a small amount of time. It also means that

there is a built-in window for after-hours maintenance and

upgrading.

5.当几乎所有的互联网也是7\*24运行，很多金融业务，尤其是金融市场，其交易时间更短。这意味着大量的活动是被压缩在一个较短时间里的。也意味着这其中有一个内嵌的用于盘后维护和升级的窗口。

6. While online companies like Etsy must meet PCI DSS regulations

for credit card data and SOX-404 auditing requirements,

this only affects the “cash register” part of the business. A financial

services organization is effectively one big cash register,where almost everything needs to be audited and almost every activity is under regulatory oversight.

6.当互联网公司如Etsy需要符合PCI DSS信用卡数据法规和SOX-404审计要求，这只影响到了“收银”业务。而一个金融服务机构是一个非常大的收银机，几乎所有的业务是需要被审计，所有的活动都受到监管。

Financial industry players were some of the earliest and biggest adopters of information technology. This long history of investing in technology also leaves them heavily weighed down by legacy systems built up over decades; systems that were not designed for rapid, iterative change. The legacy problem is made even worse by the duplication and overlap of systems inherited through mergers and acquisitions: a global investment bank can have dozens of systems performing similar functions and dozens of copies of master file data that need to be kept in sync. These systems have become more and more interconnected across the industry, which makes changes much more difficult and riskier, as problems can cascade from one system—and one organization—to another.

金融业是最早也是最大的信息技术的采用者。在技术方面漫长的投资历史也让几十年来建立起来的遗留系统成为它们背负的沉重负担：这些系统不是快速的、迭代式的变更而设计的。遗留问题因为公司合并和收购导致的重复和重叠而更为糟糕：一家全球性投资银行居然可以拥有几十个具备类似功能的系统和几十份主文件拷贝需要保持同步。行业中的这些系统已经越来越相互连接，这使得变更变得更加困难和危险，因为问题会从一个系统蔓延到另一个系统，从一个组织到另一个组织。

In addition to the forces of inertia, there are significant challenges

and costs to adopting DevOps in the financial industry. But the benefits

are too great to ignore, as are the risks of not delivering value to

customers quickly enough and losing them to competitors—especially

to disruptive online startups powered by DevOps. We’ll start

by looking at the challenges in more detail, to understand better

how financial organizations need to change in order for them to

succeed with DevOps, and how DevOps needs to be changed to

meet their requirements.

除了惯性力之外，在金融行业实施DevOps还会有许多重大挑战和成本。但是好处大到令人无法忽视，因为不能及时向客户交付价值而把客户流失给竞争对手，尤其是由DevOps驱动的颠覆性在线初创公司。我们开始将更为详细地看待这些挑战，以更好地理解金融机构需要如何改变才能在DevOps上获得成功，以及如何将DevOps进行改造以满足他们的需求。

Then we’ll look at how DevOps practices can be—and have been—

successfully adopted to develop and operate financial systems, borrowing

ideas from DevOps leaders like Etsy, Amazon, Netflix, and others.

然后借助于来自DevOps领导者的想法，如Etsy、Amazon、Netflix和其它公司，我们将审视DevOps实践是怎样已经并将继续成功应用于开发和运营金融系统的。