Implementing DevOps in Financial Markets

在金融市场实施DevOps

The drivers for adopting better operations practices in financial

enterprises are clear. The success stories are compelling. There are

challenges, as we’ve seen—but these challenges can be overcome.

在金融企业中采用更好的运营实践的驱动因素是明确的。成功的故事令人信服。正如我们所看到的，过程中有挑战，但这些挑战都是可以克服的。

Where to Start?

从哪里开始？

DevOps, in the end, is about changing the way that IT is done. This

can lead to fundamental changes in the structure and culture of an

entire organization. Look at what ING and Capital One did, and are

still doing.

DevOps，归根结底是要改变IT的工作方式。这个可以导致整个组织的结构和文化上的根本改变。看看荷兰国际集团和Capital One过去过的和正在做的事情就知道了。

Books like Gary Gruver and Tommy Mouser’s *Leading the Transformation*

(IT Revolution) and Jez Humble, Joanne Molesky, and Barry O’Reilly’s *Lean Enterprise* (O’Reilly) can help you understand how to implement Agile and DevOps in large-scale programs, how to manage cultural change within the organization, how to secure executive sponsorship, and how to shift toward Lean thinking across development and IT operations and across the business as a whole.

像Gary Gruver和Tommy Mouser的书《引领变革》（IT革命出版社），Jez Humble、Joanne Molesky和Barry O’Reilly的《精益企业》（O’Reilly出版社）可以帮助您了解如何在大型项目中实施敏捷和DevOps，如何管理组织内的文化变革，如何确保高管人员的支持，以及如何在开发和IT运营中和在整个企业中推动精益思维转型。

Organizational change on this scale is expensive and risky. DevOps

can also be implemented incrementally—in small batches, from the ground up—by building first on Agile development. Start by creating self-service tools and putting them into the hands of developers,and making testing more streamlined and efficient.

这种规模的组织变革既昂贵又危险。DevOps也可以通过首先实施敏捷开发，以增量方式小批量、从零开始实施。从创建自助服务工具并将其交给开发人员、使测试更加流畅和高效开始。

There’s a lot to be gained by going after obvious pain points first,

like manual configuration and deployment. As one example, by

automating the release and deployment steps, Fidelity Worldwide

Investment was able to significantly speed up development and testing

on key trading applications, reducing time to market while also

reducing operational risk, and saving millions of dollars per year.21

先找一些明显的痛点比如手动配置和部署，会有很多收获。例如，通过自动化发布和部署步骤，Fidelity Worldwide Investment公司在关键交易应用程序上开发和测试速度得以显著加快，减少了上市时间的同时降低运营风险，每年可节省数百万美元。21

Other initiatives like this are already underway in many financial

organizations. Several of them are creating cross-functional DevOps

teams like Capital One did to start: small, hands-on teams focused

on automating builds and release engineering, automating testing

and system provisioning, and designing and implementing Continuous

Integration and Continuous Delivery toolchains and pipelines.

These teams are laying the technical foundation that will enable the

rest of the organization to move faster and more effectively.

其他类似的举措已经在许多金融机构中展开。其中一些组织正在像Capital One一样创建跨功能的DevOp团队：亲力亲为的小团队，专注于自动化构建和版本发布工程、自动化测试和系统部署，以及设计和实施持续集成和持续交付工具链和管道。这些团队正在建设技术基础设施，这将使组织中的其他成员可以更快、更有效地运作。

While some practitioners see dedicated, embedded DevOps teams like this as an anti-pattern,22 these teams can help bridge silos between development, operations, compliance, and InfoSec; they can begin to open up communications, quickly identify and deal with process bottlenecks and other inefficiencies, and bootstrap the adoption of new practices and different ways of thinking.

当一些实践者将专门的、嵌入式的DevOps团视为一种反模式22，这些团队可以帮助桥接开发、运营、合规和信息安全之间的仓筒；他们可以引入沟通、快速识别和处理流程瓶颈和其他低效率问题，以及引导和采用新的实践和不同的思维方式。

A DevOps Journey

DevOps之旅

Where I work, we didn’t know about DevOps when we started down

this path—but DevOps happened anyway.

在我工作的地方，我们刚开始朝这个方向起步的时候，我们并不知道DevOps，但DevOps确实发生了。

When we launched our financial trading platform 10 years ago, the CEO made it clear that all of us (sales, development, operations,compliance, and management) shared the same priorities. In order for customers to trust us with their business and their customers’business, we had to ensure a high level of integrity, reliability, and regulatory compliance. While delivering new capabilities and new integration channels quickly to get more customers on board was critical to our survival, it was even more important to protect our

existing customers’ interests.

 当我们10年前推出金融交易平台时，首席执行官明确表示，我们所有人（销售、开发、运营、合规和管理）具有相同的优先权。为了让客户在他们的业务上和他们客户的业务上信任我们，我们必须确保高水平的完整性、可靠性和合规性。虽然快速提供新功能和新整合渠道以吸引更多客户对我们的生存至关重要，保护我们现有客户的利益更为至关重要。

After we went live, we had to make the switch from a project delivery

mindset to an operational one. This meant putting operational

readiness and risk management ahead of features and schedules;

spending more time on change control, building in backward compatibility,

testing failover and rollback, ensuring traceability; and building in operational transparency and safety checks. This meant that developers and operations and compliance had to work together, and understand each other better.

在我们上线后，我们必须从项目交付的心态转换到运营模式。这意味着将运营就绪和风险管理的优先级置于特性和日程表之前；在变更控制上花费更多时间，构建向后兼容性、测试故障转移和回滚、确保可追溯性；以及加强操作透明度和安全检查。这意味着开发、运营和合规人员必须一起工作，更好地彼此了解。

We started making smaller changes in smaller and smaller batches,

because smaller, incremental changes were easier to test and safer to

deploy, and because working this way helped us to keep up with rapidly

changing requirements as more customers came on board. And because we were making changes more often, we were forced to automate more of the steps in delivery: testing and compliance checks, system provisioning and configuration, build and deployment. The more that we automated this work, and the better the

tools became, the safer and easier it was for us to make changes. The

more often that we made changes, the better we got at it: more efficient,

more repeatable, more dependable.

我们开始在越来越小的批次中进行较小的变更，因为较小的增量变更更容易测试、更安全

部署，并且因为以这种方式工作，有助于我们快速跟上随着获得更多客户而带来的需求不断变更。因为我们经常变更，我们被迫自动化交付中的更多步骤：测试和合规检查、系统供应和配置、构建和部署。我们把这项工作自动化的越多，这些工具就越好用，我们的变更实施也更安全和容易。我们做变更的频率越高，我们就越能做到：效率越高，更可重复以及更可靠。

In my organization, operations and development are separate organizational

silos reporting up to different executives, in different cities.We also have independent QA. Although we created a strong engineering culture, with disciplined code reviews and automated testing and developers being held accountable for their work, and although we’ve had automated Continuous Integration and build pipelines in place for a long time, we still rely on the QA team’s manual testing and reviews to catch edge conditions and to hunt for

operational and usability bugs and to look for holes in our automated test coverage. Their job is not to try to find all of the bugs in the system. We rely on them to identify risks, and to provide information that we can use to learn and to improve our controls and engineering processes.

在我的组织中，运营和开发团队是独立的筒仓组织，在不同的城市，向不同的管理者汇报。我们也有独立的质量保证团队。虽然我们创造了一个强大的工程师文化，有严格的代码审查和自动化测试，开发人员也对他们的工作认真负责，以及尽管我们也早已经实现了自动化持续集成和构建流水线，在很长一段时间内，我们仍然依赖于质量保证团队的手动测试和审查，以发现边界条件和找出运营和易用性缺陷，并寻找出我们自动化测试的覆盖率漏洞。他们的工作不是找出系统中的所有缺陷。我们依靠他们来识别风险，并提供信息给我们，用来学习和改进我们的控制和软件工程的流程。

We have separate organizational silos because they help us to maintain

control over change, to minimize security and operational risks,and to satisfy compliance and governance requirements. But because we all share the same goals and priorities, this structure doesn’t get in the way of people working together. They are boundaries,not walls that cannot be crossed. Developers and QA and Ops collaborate regularly and closely on design and problem solving,

provisioning and configuring systems, implementing security and compliance controls, coordinating changes, responding to incidents.They share ideas, problems, practices, and tools.

我们有独立的筒仓组织，因为它们帮助我们维持变更控制，以最大限度地降低安全和运营风险，并满足合规和治理要求。但是因为我们都有相同的目标和优先级，这种结构不会妨碍人们在一起工作。它们是边界，但不是不能跨越的墙。开发和质量保证、运营人员经常在一起紧密合作，设计系统和解决问题、提供和配置系统、实现安全性和合规控制、协调变更、响应事件。他们分享想法、问题、实践和工具。

Market operations and QA and compliance decide if and when changes go into production—not developers. Deployment is done by operations, after the reviews and checks are complete, using automated tooling, with developers monitoring closely and standing by. We don’t do Continuous Deployment to production, or anything close to it. We still have some manual testing and manual approval

gates, and probably always will. But we can make changes quickly and with confidence, taking advantage of automation and agility.

市场运作、质量保证和合规人员决定是否以及何时将变更投产，而不是开发人员。在审查和检查完成后，由运营使用自动化工具完成部署，开发人员则密切监控并在一旁待命。我们不会在生产环境持续部署。我们仍旧有一些手工测试和人工批准，或许永远都会由。但是利用自动化和敏捷性，我们可以快速而自信地做出改变。

This isn’t how teams at Etsy or Netflix work—but it is DevOps.

这不是Etsy或Netflix团队的工作方式，而是DevOps。

In the financial industry, regulators, security and compliance officers,

risk managers, auditors, and even customers are all concerned

that business lines and Agile development teams may put speed of

delivery ahead of data safety, security, and operational reliability. For

us, and for other financial firms, adopting DevOps practices like Continuous Delivery and Infrastructure as Code, and improving collaboration and communications between the business lines and engineering and operations and governance teams, is about reducing operational and technical risks, improving efficiency, and increasing accountability and transparency. Optimizing time to

market comes as a happy side effect.

 在金融行业，监管机构、安全和合规官，风险经理、审计师甚至客户都很担心业务线和敏捷开发团队可能会将交付速率优先于数据安全、安全性和运行可靠性。对于我们和其他金融公司来说，采用如持续交付和基础设施即代码，并改进业务线、软件工程、运营和治理团队之间的协作和沟通这些DevOps实践，是关于减少运营和技术风险、提高效率以及增进可信度和透明度。优化上市时间只是一个令人愉快的副作用。

Done this way, the ROI case for DevOps seems clear. An approach to managing IT changes that cuts time to delivery and operational costs, minimizes technical and operational risks, and makes compliance happy? That’s a win, win, win.

这样做之后，DevOps的投资回报率情况似乎很清楚。这是一种管理IT变更，缩短交付时间和运营成本，最大限度地降低技术和运营风险，并使合规人员欣喜地方法吗？那是一场胜利，胜利，胜利。

Wealthfront: A Financial Services Unicorn

Wealthfront：金融服务独角兽

There are already DevOps unicorns in the financial industry, as we’ve seen looking at LMAX, ING, and Capital One. Wealthfront is another DevOps unicorn that shows how far DevOps ideas and practices can be taken in financial services.

金融业已经有了DevOps独角兽，我们看到过LMAX, 荷兰国际集团和Capital One。Wealthfront是另一只DevOps独角兽，其展示的一些思想和实践可以为金融服务业采纳。

Wealthfront, a retail automated investment platform (“robo advisor”) that was launched in 2011, is not a conventional financial services company. It started as an online portfolio management game on Facebook called “KaChing,” and then, following Eric Ries’s Lean Startup approach, continued to pivot to its current business model. Today, Wealthfront manages $2.5 billion in assets for thousands

of customers.

Wealthfront是在2011年推出的一个零售自动化投资平台（机器人投顾），不是传统的金融

服务公司。其最初是一个在Facebook上叫做“KaChing”的在线投资组合管理游戏。然后，跟随Eric Ries的精益创业方法，以其当前业务模式为中心继续发展。如今，Wealthfront为数千个客户管理着价值25亿美元的资产。

Wealthfront was built using DevOps ideas from the start. It follows Continuous Deployment, where changes are pushed out by developers directly, 10 or 20 or 50 or more times per day, like at Etsy.And, like at Etsy, Wealthfront has an engineering-driven culture where developers are encouraged to push code changes to production

on their first day of work. But this is all done in a highly regulated environment that handles investment money and private customer records.

Wealthfront从一开始就是用DevOps的思想建立起来的。它遵循持续部署，类似于Etsy，由开发人员直接推送变更，数量在每天10次、20次、50次甚至更多。而且，和Etsy一样，Wealthfront也有一种工程师驱动的文化。开发人员在他们上班的第一天就被鼓励将变更代码推送到生产环境中。他们处理投资资金和私人客户记录，因而这一切都是在一个高度管制的环境中完成的。

How do they do it? By following many of the practices and ideas described in this book—to the extreme.

他们是怎么做到的？通过遵循许多本书中提到的实践和思想，并把它们做到极致。

Developers at Wealthfront are obsessed with writing good, testable code. They enforce consistent coding standards, run static analysis (dependency checks, identifying forbidden function calls, source code analysis with tools like FindBugs and PMD to find bad code and common coding mistakes), and review all code changes.

They’ve followed test-driven development from the beginning to build an extensive automated test suite. If code coverage is too low in key areas of the code, the build fails. Every couple of months they run Fix-It Days to clean up tests and improve test coverage in key areas. The same practices are followed for infrastructure changes, using Chef.

Wealthfront的开发人员热衷于编写好的、可测试的代码。它们执行一致的编码规范，运行静态分析（依赖检查，识别禁止的函数调用，使用Findbugs和PMD等工具进行代码分析以查找错误代码以及常见的编码错误），并对所有代码修改进行评审。他们从一开始就遵循测试驱动开发模式并构建了一个全面的自动化测试套件。如果关键区域模块的覆盖率过低会导致构建失败。他们每隔几个月运行修复日(Fix It Days)，以清理测试并提高关键模块的的测试覆盖率。针对基础设施的变更也遵循同样的实践，并使用Chef。

Wealthfront engineers’ priorities are to optimize for safety as well as

speed. The company continually invests in its platforms and tools to

make it easy for engineers to do things the right way by default.

They routinely dark-launch new features; they use canary deployments

to roll changes out incrementally; and they’ve built a runtime

“immune system,” as described in the Lean Startup methodology, to

monitor logs and key application and system metrics after changes

are deployed and automatically roll back the most recent change if

it looks like something is going wrong.

Wealthfront工程师们的首要任务是既安全又快速。公司不断投资其平台和工具，以使工程师很容易在默认情况下以正确的方式工作。他们经常以暗发布的方式发布新功能；他们使用金丝雀部署以增量方式推出变更；如精益创业方法论所述，他们构建了一个运行时“免疫系统”，该系统监控日志和关键应用程序和系统指标，如果在变更部署后，如果好像出了什么问题，该系统将自动回滚最近的变更。

Wealthfront has no operations staff or QA staff: the system is designed, developed, tested, and run by engineers. All of this sounds more like an engineering-driven Internet startup than a financial services provider, and Wealthfront is the exception, rather than the rule—at least, for now.20

WealthFront没有运营人员或质量保证人员：系统是由工程师设计、开发、测试和运行的。所有这些听起来更像是一家工程师驱动的互联网初创企业，而不是一个金融服务提供商，而WealthFront则是一个例外而不是惯例，至少现在是。20

Barclays: Building on Islands of Agility

巴克莱：建立在敏捷岛上

Barclays, one of the world’s largest global banks, is currently undergoing

an organization-wide Agile and DevOps transformation: not just within IT, but across business lines, including legal, compliance, finance, HR, and even real estate functions.

巴克莱是全球最大的银行之一，目前正在经历整个组织范围的敏捷和DevOps转型：不只是

IT内部，而实跨业务线，包括法律、合规、财务、人力资源甚至不动产职能。

Like most financial enterprises, Barclays was following a highly

structured Waterfall project delivery model, with multiple reviews

and approval gates. Each change to an IT system required filling

out 28 mandatory artifacts, with the change management process

taking an average of 56 days to complete.

与大多数金融企业一样，巴克莱银行也采用高度结构化的瀑布项目交付模型，具有多个评审以及审批门。对IT系统的每次更改都需要28个强制性的工件，而变更管理流程平均需要56天才能完成。

Barclays started two years ago by building small “islands of agility”

that they are now linking up and extending; they’re breaking large

programs and departments down into smaller problems that can be

solved by independent, fast-moving, multidisciplinary teams following

Lean and iterative practices, and strangling large, monolithic

systems and breaking them into microservices.

两年前，巴克莱银行开始建造小型“敏捷岛”，而他们现在正在连接和扩展这些岛屿；他们把大型项目和部门分解成更小的问题，由独立、快速移动、多领域团队来解决。这些团队采用精益和迭代实践，将大型的单体系统分解为微服务。

Barclays now has more than 10,000 people working in Agile and DevOps teams. Their lead time to delivery has improved by more than 300%, and change control approvals now only take 1 day instead of 56. At the same time, code quality has improved by more than 50% and occurrences of production incidents have significantly decreased.2

巴克莱现在的敏捷和DevOps团队超过10000人。他们的交付前置时间降低了超过300%，变更控制批准现在只需1天而不是56天。与此同时，代码质量提高50%以上，生产事故发生率显著减少。2

在金融市场实施DevOps

在财务方面采用更好运营实践的驱动因素

企业是明确的。成功的故事令人信服。有

挑战，正如我们所看到的，但这些挑战是可以克服的。

从哪里开始？

DevOps最终是要改变它的工作方式。这个

可以导致结构和文化的根本改变

整个组织。看看荷兰国际集团和Capital One做了什么，

还在做。

像加里·格鲁夫和汤米·穆瑟的书引领变革

（IT革命）、Jez Humble、Joanne Molesky和Barry

O'Reilly（O'Reilly）可以帮助您了解如何精益企业

在大型项目中实现敏捷和开发，如何管理

组织内的文化变革，如何确保行政人员的安全

赞助，以及如何在整个开发过程中转向精益思维

以及整个企业的IT运营。

这种规模的组织变革既昂贵又危险。德沃斯

也可以从

以敏捷开发为基础。从创建开始

自助服务工具并将其交给开发人员，

使测试更加简化和高效。

先找一些明显的痛点，会有很多收获，

比如手动配置和部署。例如，通过

自动化发布和部署步骤，Fidelity Worldwide

投资能够显著加速开发和测试

在关键交易应用程序上，减少上市时间，同时

降低运营风险，每年节省数百万美元。21

其他类似的举措已经在许多金融机构中展开。

组织。其中一些人正在创建跨功能的DevOp

像Capital One这样的团队从一开始就做到了：小型的、注重实践的团队

自动化构建和发布工程，自动化测试

和系统供应，以及设计和实现连续的

集成和持续交付工具链和管道。

这些球队正在铺设技术基础，这将使

组织中的其他成员可以更快、更有效地行动。

当一些实践者看到专用的、嵌入式的DevOps团队时

像这样作为一种反模式，22个小组可以帮助桥接筒仓

在开发、运营、合规和信息安全之间；他们

可以开始开放沟通，快速识别和处理

由于流程瓶颈和其他效率低下，以及引导

采用新的实践和不同的思维方式。

DevOps之旅

在我工作的地方，我们刚开始的时候不知道DevOps

这条路，但无论如何都发生了DevOps。

当我们10年前推出金融交易平台时，

首席执行官明确表示，我们所有人（销售、开发、运营、

合规和管理）具有相同的优先权。整齐

让客户相信我们的业务和客户

业务方面，我们必须确保高水平的完整性、可靠性和

合规性。同时提供新的功能和新的

快速整合渠道以吸引更多客户

对我们的生存至关重要，保护我们的

现有客户的利益。

在我们上线后，我们必须从项目交付中进行转换。

心态要有操作性。这意味着投入运营

在特性和时间表之前做好准备并进行风险管理；

在变更控制上花费更多时间，构建向后兼容性，

测试故障转移和回滚，确保可追溯性；以及

加强操作透明度和安全检查。这意味着

开发人员、运营和法规遵从性必须发挥作用

一起，更好地了解彼此。

我们开始在越来越小的批次中进行较小的更改，

因为较小的增量变化更容易测试，更安全

部署，因为以这种方式工作有助于我们快速跟上

随着更多客户的加入，需求也在不断变化。和

因为我们经常做改变，我们被迫

自动化交付中的更多步骤：测试和遵从性

检查、系统供应和配置、构建和部署。

我们把这项工作自动化的越多，越好

工具变得越来越安全，我们更容易进行更改。这个

我们做出改变的频率越高，我们就越能做到：效率越高，

更可重复，更可靠。

在我的组织中，运营和开发是独立的组织

在不同的城市向不同的管理者汇报的信息孤岛。

我们也有独立的质量保证。虽然我们创造了一个强大的工程

文化，有严格的代码审查和自动化测试

开发人员对他们的工作负责，以及

尽管我们已经实现了自动化的持续集成和构建

在很长一段时间内，我们仍然依赖于质量保证团队的

手动测试和审查，以捕捉边缘条件和寻找

操作和可用性错误，并在我们的自动化系统中寻找漏洞

测试覆盖率。他们的工作不是在

系统。我们依靠他们来识别风险，并提供信息

我们可以用来学习和改进我们的控制和工程

过程。

我们有独立的组织筒仓，因为它们帮助我们维持

控制变更，以最大限度地降低安全和运营风险，

并满足法规遵从性和治理要求。但是

因为我们都有相同的目标和优先顺序，这种结构

不会妨碍人们一起工作。它们是边界，

不是不能跨越的墙。开发人员和质量保证和运营

在设计和解决问题方面定期密切合作，

提供和配置系统，实现安全性和

合规控制、协调变更、响应事件。

他们分享想法、问题、实践和工具。

市场运作和质量保证和合规决定是否和何时

变更进入生产，而不是开发人员。部署完成

按操作，在审查和检查完成后，使用

自动化工具，开发人员密切监控和站立

通过。我们不会持续部署到生产或其他方面

靠近它。我们还有一些手动测试和手动批准

盖茨，也许永远都会。但是我们可以很快做出改变

自信地利用自动化和敏捷性。

这不是Etsy或Netflix团队的工作方式，而是DevOps。

在金融行业，监管机构、安全和合规官，

风险经理、审计师甚至客户都很担心

业务线和敏捷开发团队可能会加快

提前交付数据安全性、安全性和操作可靠性。为了

美国和其他金融公司采用DevOps实践，如

持续交付和基础设施作为代码，并改进

业务线、工程、运营和治理团队之间的协作和通信是关于减少

操作和技术风险，提高效率，以及

提高问责制和透明度。优化时间

市场是一个愉快的副作用。

这样做，DevOps的投资回报率情况似乎很清楚。一种方法

管理IT变更，缩短交付和运营时间

成本，最大限度地降低技术和运营风险，并使合规性

高兴吗？那是一场胜利，一场胜利，一场胜利。

66℃

财富前线：金融服务独角兽

金融业已经有了德沃普独角兽，

我们看到过lmax、ing和capital one。财富前沿是

另一只DevOps独角兽展示了思想和

金融服务业可以采取一些做法。

Wealthfront是一个零售自动化投资平台（“Robo Advisor”）。

这是在2011年推出的，不是传统的金融

服务公司。它最初是一个在线投资组合管理

Facebook上的游戏叫做“Kaching”，然后，跟随Eric Ries的

精益创业方法，继续以其当前业务为中心

模型。如今，财富前线管理着价值25亿美元的资产，价值数千美元。

客户数量。

财富前线从一开始就是用DevOps的想法建立起来的。它遵循

持续部署，由开发人员推出更改

直接的，每天10次或20次或50次以上，比如在Etsy。

而且，和Etsy一样，Wealthfront也有一种工程驱动的文化。

鼓励开发人员将代码更改推送到生产环境中

在他们上班的第一天。但这一切都是在一个高度管制下完成的

处理投资资金和私人客户的环境

记录。

他们是怎么做到的？通过遵循许多实践和想法

在这本书中描述到了极端。

财富前沿的开发人员热衷于编写好的、可测试的

代码。它们执行一致的编码标准，运行静态分析

（依赖项检查，识别禁止的函数调用，源

使用findbugs和pmd等工具进行代码分析以查找错误代码

以及常见的编码错误），并检查所有代码更改。

他们从一开始就遵循测试驱动的开发

构建一个广泛的自动化测试套件。如果代码覆盖率过低

在代码的关键区域，构建失败。每隔几个月

运行fix it days清理测试并提高密钥中的测试覆盖率

地区。基础设施变更也遵循同样的做法，

使用厨师。

财富前沿工程师的首要任务是优化安全以及

速度。公司不断投资其平台和工具，以

使工程师在默认情况下以正确的方式进行工作变得容易。

他们经常暗中发布新功能；他们使用金丝雀部署

以增量方式展开更改；它们已经构建了一个运行时

“免疫系统”，如精益启动方法所述，以

更改后监控日志和关键应用程序和系统指标

如果

好像出了什么问题。

WealthFront没有操作人员或质量保证人员：系统是

由工程师设计、开发、测试和运行。所有这些

听起来更像是一个工程驱动的互联网初创企业，而不是

金融服务提供商，而财富前线则是例外

至少现在比规定的要多。20

巴克莱：建立在敏捷岛上

巴克莱是全球最大的银行之一，目前正在经历

整个组织范围的敏捷和DevOps转换：不是

就在IT内部，但跨业务线，包括法律、法规遵从性，

财务、人力资源甚至房地产职能。

与大多数金融企业一样，巴克莱银行也非常关注

结构化瀑布项目交付模型，具有多个评审

以及审批门。对IT系统的每次更改都需要填充

28个强制性的工件，包括变更管理过程

平均需要56天完成。

两年前，巴克莱银行开始建造“敏捷之岛”。

他们现在正在连接和扩展；他们正在突破

把项目和部门分解成更小的问题

由独立、快速移动、多学科团队跟随解决

精益和迭代实践，扼杀大型、整体

系统并将其分解为微服务。

巴克莱现在有超过10000人在敏捷和

DevOps团队。他们的交货期提高了更多

超过300%，变更控制批准现在只需1天

而不是56。与此同时，代码质量提高了更多

50%以上，生产事故发生率显著

减少。2

在金融市场实施DevOps

在财务方面采用更好运营实践的驱动因素

企业是明确的。成功的故事令人信服。有

挑战，正如我们所看到的，但这些挑战是可以克服的。

从哪里开始？

DevOps最终是要改变它的工作方式。这个

可以导致结构和文化的根本改变

整个组织。看看荷兰国际集团和Capital One做了什么，

还在做。

像加里·格鲁夫和汤米·穆瑟的书引领变革

（IT革命）、Jez Humble、Joanne Molesky和Barry

O'Reilly（O'Reilly）可以帮助您了解如何精益企业

在大型项目中实现敏捷和开发，如何管理

组织内的文化变革，如何确保行政人员的安全

赞助，以及如何在整个开发过程中转向精益思维

以及整个企业的IT运营。

这种规模的组织变革既昂贵又危险。德沃斯

也可以从

以敏捷开发为基础。从创建开始

自助服务工具并将其交给开发人员，

使测试更加简化和高效。

先找一些明显的痛点，会有很多收获，

比如手动配置和部署。例如，通过

自动化发布和部署步骤，Fidelity Worldwide

投资能够显著加速开发和测试

在关键交易应用程序上，减少上市时间，同时

降低运营风险，每年节省数百万美元。21

其他类似的举措已经在许多金融机构中展开。

组织。其中一些人正在创建跨功能的DevOp

像Capital One这样的团队从一开始就做到了：小型的、注重实践的团队

自动化构建和发布工程，自动化测试

和系统供应，以及设计和实现连续的

集成和持续交付工具链和管道。

这些球队正在铺设技术基础，这将使

组织中的其他成员可以更快、更有效地行动。

当一些实践者看到专用的、嵌入式的DevOps团队时

像这样作为一种反模式，22个小组可以帮助桥接筒仓

在开发、运营、合规和信息安全之间；他们

可以开始开放沟通，快速识别和处理

由于流程瓶颈和其他效率低下，以及引导

采用新的实践和不同的思维方式。

DevOps之旅

在我工作的地方，我们刚开始的时候不知道DevOps

这条路，但无论如何都发生了DevOps。

当我们10年前推出金融交易平台时，

首席执行官明确表示，我们所有人（销售、开发、运营、

合规和管理）具有相同的优先权。整齐

让客户相信我们的业务和客户

业务方面，我们必须确保高水平的完整性、可靠性和

合规性。同时提供新的功能和新的

快速整合渠道以吸引更多客户

对我们的生存至关重要，保护我们的

现有客户的利益。

在我们上线后，我们必须从项目交付中进行转换。

心态要有操作性。这意味着投入运营

在特性和时间表之前做好准备并进行风险管理；

在变更控制上花费更多时间，构建向后兼容性，

测试故障转移和回滚，确保可追溯性；以及

加强操作透明度和安全检查。这意味着

开发人员、运营和法规遵从性必须发挥作用

一起，更好地了解彼此。

我们开始在越来越小的批次中进行较小的更改，

因为较小的增量变化更容易测试，更安全

部署，因为以这种方式工作有助于我们快速跟上

随着更多客户的加入，需求也在不断变化。和

因为我们经常做改变，我们被迫

自动化交付中的更多步骤：测试和遵从性

检查、系统供应和配置、构建和部署。

我们把这项工作自动化的越多，越好

工具变得越来越安全，我们更容易进行更改。这个

我们做出改变的频率越高，我们就越能做到：效率越高，

更可重复，更可靠。

在我的组织中，运营和开发是独立的组织

在不同的城市向不同的管理者汇报的信息孤岛。

我们也有独立的质量保证。虽然我们创造了一个强大的工程

文化，有严格的代码审查和自动化测试

开发人员对他们的工作负责，以及

尽管我们已经实现了自动化的持续集成和构建

在很长一段时间内，我们仍然依赖于质量保证团队的

手动测试和审查，以捕捉边缘条件和寻找

操作和可用性错误，并在我们的自动化系统中寻找漏洞

测试覆盖率。他们的工作不是在

系统。我们依靠他们来识别风险，并提供信息

我们可以用来学习和改进我们的控制和工程

过程。

我们有独立的组织筒仓，因为它们帮助我们维持

控制变更，以最大限度地降低安全和运营风险，

并满足法规遵从性和治理要求。但是

因为我们都有相同的目标和优先顺序，这种结构

不会妨碍人们一起工作。它们是边界，

不是不能跨越的墙。开发人员和质量保证和运营

在设计和解决问题方面定期密切合作，

提供和配置系统，实现安全性和

合规控制、协调变更、响应事件。

他们分享想法、问题、实践和工具。

市场运作和质量保证和合规决定是否和何时

变更进入生产，而不是开发人员。部署完成

按操作，在审查和检查完成后，使用

自动化工具，开发人员密切监控和站立

通过。我们不会持续部署到生产或其他方面

靠近它。我们还有一些手动测试和手动批准

盖茨，也许永远都会。但是我们可以很快做出改变

自信地利用自动化和敏捷性。

这不是Etsy或Netflix团队的工作方式，而是DevOps。

在金融行业，监管机构、安全和合规官，

风险经理、审计师甚至客户都很担心

业务线和敏捷开发团队可能会加快

提前交付数据安全性、安全性和操作可靠性。为了

美国和其他金融公司采用DevOps实践，如

持续交付和基础设施作为代码，并改进

业务线、工程、运营和治理团队之间的协作和通信是关于减少

操作和技术风险，提高效率，以及

提高问责制和透明度。优化时间

市场是一个愉快的副作用。

这样做，DevOps的投资回报率情况似乎很清楚。一种方法

管理IT变更，缩短交付和运营时间

成本，最大限度地降低技术和运营风险，并使合规性

高兴吗？那是一场胜利，一场胜利，一场胜利。

66℃

财富前线：金融服务独角兽

金融业已经有了德沃普独角兽，

我们看到过lmax、ing和capital one。财富前沿是

另一只DevOps独角兽展示了思想和

金融服务业可以采取一些做法。

Wealthfront是一个零售自动化投资平台（“Robo Advisor”）。

这是在2011年推出的，不是传统的金融

服务公司。它最初是一个在线投资组合管理

Facebook上的游戏叫做“Kaching”，然后，跟随Eric Ries的

精益创业方法，继续以其当前业务为中心

模型。如今，财富前线管理着价值25亿美元的资产，价值数千美元。

客户数量。

财富前线从一开始就是用DevOps的想法建立起来的。它遵循

持续部署，由开发人员推出更改

直接的，每天10次或20次或50次以上，比如在Etsy。

而且，和Etsy一样，Wealthfront也有一种工程驱动的文化。

鼓励开发人员将代码更改推送到生产环境中

在他们上班的第一天。但这一切都是在一个高度管制下完成的

处理投资资金和私人客户的环境

记录。

他们是怎么做到的？通过遵循许多实践和想法

在这本书中描述到了极端。

财富前沿的开发人员热衷于编写好的、可测试的

代码。它们执行一致的编码标准，运行静态分析

（依赖项检查，识别禁止的函数调用，源

使用findbugs和pmd等工具进行代码分析以查找错误代码

以及常见的编码错误），并检查所有代码更改。

他们从一开始就遵循测试驱动的开发

构建一个广泛的自动化测试套件。如果代码覆盖率过低

在代码的关键区域，构建失败。每隔几个月

运行fix it days清理测试并提高密钥中的测试覆盖率

地区。基础设施变更也遵循同样的做法，

使用厨师。

财富前沿工程师的首要任务是优化安全以及

速度。公司不断投资其平台和工具，以

使工程师在默认情况下以正确的方式进行工作变得容易。

他们经常暗中发布新功能；他们使用金丝雀部署

以增量方式展开更改；它们已经构建了一个运行时

“免疫系统”，如精益启动方法所述，以

更改后监控日志和关键应用程序和系统指标

如果

好像出了什么问题。

WealthFront没有操作人员或质量保证人员：系统是

由工程师设计、开发、测试和运行。所有这些

听起来更像是一个工程驱动的互联网初创企业，而不是

金融服务提供商，而财富前线则是例外

至少现在比规定的要多。20

巴克莱：建立在敏捷岛上

巴克莱是全球最大的银行之一，目前正在经历

整个组织范围的敏捷和DevOps转换：不是

就在IT内部，但跨业务线，包括法律、法规遵从性，

财务、人力资源甚至房地产职能。

与大多数金融企业一样，巴克莱银行也非常关注

结构化瀑布项目交付模型，具有多个评审

以及审批门。对IT系统的每次更改都需要填充

28个强制性的工件，包括变更管理过程

平均需要56天完成。

两年前，巴克莱银行开始建造“敏捷之岛”。

他们现在正在连接和扩展；他们正在突破

把项目和部门分解成更小的问题

由独立、快速移动、多学科团队跟随解决

精益和迭代实践，扼杀大型、整体

系统并将其分解为微服务。

巴克莱现在有超过10000人在敏捷和

DevOps团队。他们的交货期提高了更多

超过300%，变更控制批准现在只需1天

而不是56。与此同时，代码质量提高了更多

50%以上，生产事故发生率显著

减少。2