vamp

WHITEPAPER

FREE FROM THE DEPARTMENT OF 'NO': RELEASE FEATURES, TEST AND OPTIMIZE REVENUE WITHOUT IT

TOO LONG TO READ?

Product owners want to release features, test and optimize revenue at will, but how do you get there?

This paper shows four key best practices that allow product owners real-time control over feature release without long IT processes.

These are:

- Testing in production for better time-to-market and time-to-value
- Precise user segmentation for real-time insight into customer impact
- Data-driven release strategies for hands-off revenue optimization
- Automating a safety net so customer experience is always protected

CONTENT

BE FREE OF LONG SPRINT CYCLES, ENDLESS REQUESTS IN JIRA, RELEASE DATE TABOOS AND RELEASE AT WILL WITH THIS FOUR KEYS MINI-GUIDE FOR PRODUCT OWNERS IN ECOMMERCE

You're 12 hours into a production outage. Or to be more precise: 12 hours, 1 spoiled holiday, 3 disappointed family members, 10 phone calls, an 8-point drop in customer satisfaction, a 50% drop in sales conversions...and counting. As you try to talk your ops engineers out of the crisis, you think ahead to a single point in the future - the moment you're knocking on your boss's door to answer for the damages.

As a product owner, it's your responsibility to push the product forward on customer experience and revenue, especially in this era of slim margins and customer-centric e-commerce. But the need to push forward on delivering value comes hand-in-hand with facing IT bottlenecks. Technology stacks are outdated, feature releases are manual and error-prone, and their future impact on customers and the business comes down to guesswork. At the same time, customers are always online, and, in reality, business never sleeps. So, why should product innovation be kept painstakingly slow?

FREE FEATURE RELEASE FROM THE DEPARTMENT OF ‘NO’

Releasing new features to production is usually thought of as the moment someone in a technical role deploys a new version of code to production. While preparing a new feature takes a lot of work from business analysts, product owners and developers, it only brings value when it’s actually released.

In truth, because you’re exposing your software to customers, releasing is the only part of the software delivery process that directly impacts business value. That means how you manage your release process can have a direct impact on customer satisfaction, platform performance and revenue.

PRODUCT OWNERS IN THE DRIVER’S SEAT

As new and innovative technologies emerge to replace outdated tech stacks and processes, it’s not only IT teams that can get excited by the prospect of software innovation.

For product owners, these technologies make going to production a non-event, provide real-time insight into the customer impact of releases in production, ensure that fixing bugs is just an automated rollback away, and allow them to set up small experiments in little time, instead of over several sprints’ time. For the first time, product owners are being put in the driver’s seat of feature releases without depending on IT.

RELEASE, TEST AND OPTIMIZE AT WILL:
FOUR KEY BEST PRACTICES

With control over feature releases, product owners now have the tools to improve customer satisfaction, overall product revenue, as well as the revenue impact of each release in real-time, without relying on long and opaque IT processes. But how do you get there? In this paper, you'll learn how to rethink your release process to:

- Improve time-to-market and time-to-value by testing in production
- Have real-time control over which customers see what feature and when
- Rely on data-driven metrics for real-time release decisions and experimentation
- Reduce the impact of failing software on your customers

Let's look at how these key best practices can free you from relying on IT for releasing features, testing and optimizing customer experience and revenue.

1. IMPROVE TIME-TO-MARKET AND TIME-TO-VALUE BY TESTING IN PRODUCTION:

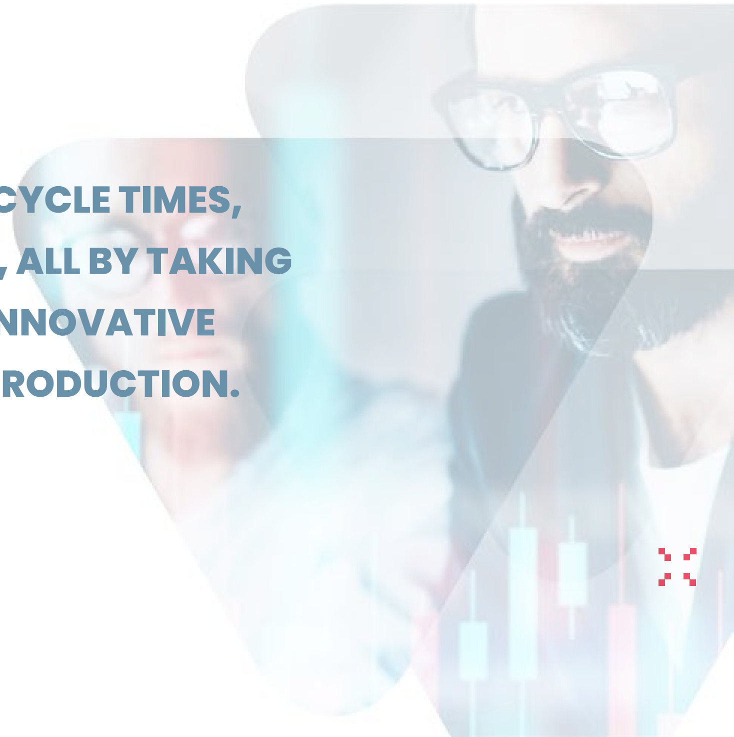
Doesn't everything in IT just... take so long? Especially the testing lifecycle can be a vicious circle. When the release moment is the first time you expose code changes to real users in production, releasing is risky, and everybody in the process gets understandably nervous.

To firm up certainty that you're releasing stable code to production, the intuitive response in many organizations is to mitigate risk by adding more testing before releasing. As testing complexity increases, release cycles drag, more changes are piled into each release, making each new version even bigger and riskier. But the result isn't better. Your customers still suffer from bugs in production. And the product and business suffer because testing and acceptance cycles take up the majority of your total time-to-market, eat up an ever-increasing share of costs and offer no prediction of the revenue impact of a new version.

THE KEY? VALIDATE EARLY AND OFTEN USING REAL USER TRAFFIC

Instead of trying to shadow customer behavior in pre-production and still falling short, it's now possible to safely expose your new version to a controlled subset of real customers in a live production environment. The subset can be made up of internal testers, beta customers or an external low-value customer group.

That limits the impact of a failed release and lowers the risk of releasing full stop, which makes releasing to production a high-occurrence, low-impact, no-sweat event. By cutting out the resource drain of extensive testing and acceptance cycles, releasing becomes iterative, cycle times speed up, and your organization can move to market faster. And crucially, the results of this first stage of testing are the best predictors for what will happen with your premium customers if you expose them to a new version.



CONSIDER THAT YOU COULD SIGNIFICANTLY SPEED UP CYCLE TIMES, REDUCE TESTING COSTS AND IMPROVE TIME-TO-MARKET, ALL BY TAKING ADVANTAGE OF THE OPPORTUNITIES THAT TODAY'S INNOVATIVE TECHNOLOGIES PROVIDE IN ORDER TO SAFELY TEST IN PRODUCTION.

2. HAVE REAL-TIME CONTROL OVER WHICH CUSTOMERS SEE WHAT FEATURE AND WHEN:

Not having to expose all your customers to a new version in one “big bang” means that now, product owners can start asking themselves: which segment of my customer base do I want to expose to a new feature, when and under which specific conditions? Meaning, they can start taking control over releasing to benefit the business, while not being limited by technical complexities.

That new-found control is also supported by the fact that today’s release technologies allow for the separation of code deployments from feature releases. In the past, deploying and releasing were a single complex act carried out by engineering. Now, rather than having to log change tickets into Jira or call their ops engineer out of bed to make a change, product owners can release at will with the click of a button. That ability coupled with the ability to apply customer segmentation to releases is for the first-time putting product owners and managers in the driver’s seat of feature release.

THE KEY? CUSTOMER SEGMENTATION FOR REAL-TIME CONTROL

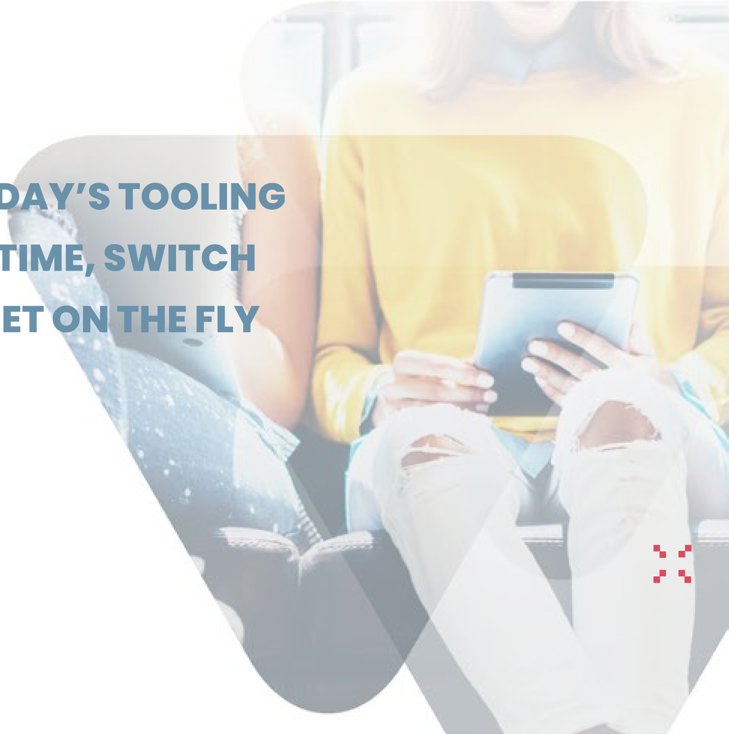
Organizing your release process around user segmentation puts you as a product owner in direct control of your customer experience. But randomly sending a small amount of traffic to test a release is not the smart way of doing things. You need more control over traffic segmentation to select the right subset of user traffic for each release.

Segmenting traffic is a great technical capability, but to be able to make releasing software an easily repeatable, no-stress endeavor, we need to apply business logic to apply the right kind of traffic segmentation to each release, automatically.

By using repeatable policies that define the specific conditions for traffic segmentation, applying different segmentation rules to different types of releases becomes a simple, business-oriented part of releasing a new software version. Release policies are rules to govern a release and are easy for those in non-technical roles to create at the moment they write the business requirements for a new feature. Release policies allow product owners to determine how a release should go to production, without dependence on developers or operational staff.

Traffic is segmented in any number of ways, based on traffic and user characteristics like source IP-address and geography, device type, customer type, log-in status, and much more. By codifying these rules into policies, segmentation becomes re-usable across individual versions, across multiple releases and parallel releases to different customer segments.

The ability to carry out fine-grained segmentation makes releasing new features more KPI-driven, going above and beyond looking at just technical metrics and health. If a change doesn't meet the business criteria, today's tooling allows you as a product owner to react in real-time, switch back to a previous version, or change a feature set on the fly, without putting in change requests. You can now build policies and allow releases to run, even on the weekend.



IF A CHANGE DOESN'T MEET YOUR BUSINESS CRITERIA, TODAY'S TOOLING ALLOWS YOU AS A PRODUCT OWNER TO REACT IN REAL-TIME, SWITCH BACK TO A PREVIOUS VERSION, OR CHANGE A FEATURE SET ON THE FLY

3. USE DATA-DRIVEN BUSINESS METRICS FOR REAL-TIME RELEASE DECISIONS AND EXPERIMENTATION

If you as a product owner want to make a true positive impact on customer satisfaction or your NPS score, you want to be obsessed with real-time data-driven insights. Once upon a time, you had to wait on a weekly report that was two weeks behind the curve by the time you received it (if you were lucky), and would never contain technical data, such as response times.

Not having real-time access to both tech and business metrics leaves product owners hobbled in terms of what decisions will improve customer experience and revenue. That's quickly becoming a thing of the past. With new innovative tech, your data isn't just parked somewhere waiting for an analyst to dig it up. New release technologies make it discoverable, actionable, and deliver it directly to you so you can make decisions using real-time KPIs.

THE KEY? SMART RELEASE STRATEGIES INTEGRATED WITH TECH AND BUSINESS ANALYTICS

By applying business logic to traffic segmentation through release policies and allowing for testing in production, you can now easily tie IT metrics to business outcomes. The result is that you can analyze releases in production beyond the initial launch. When that process is fed with metrics from monitoring systems, as well as information from distributed tracing, you have real-time access to both the technical and business metrics that report how releases are performing across the board.

That’s great for experimentation, which now becomes smart and hands-off. With policies set by you instructing new technologies, you as a product owner can run experiments and have them automatically implemented. You no longer have to wait for the outcome of an experiment and make a change request to see it implemented, but a release solution integrated with the APIs in tools such as Optimizely and Adobe Test and Target can do that for you.

What’s more, you can take the results of a set of experiments or AB tests and create a higher-level release strategy out of a number of release policies to automatically and continuously optimize revenue. In this way, you can automate the roll out of winning experiments to the entire customer base. With smart release strategies in place, production is no longer a black box. You always have the real-time data-driven access you need to drive the product forward.

YOU CAN TAKE THE RESULTS OF A SET OF AB TESTS AND CREATE A HIGHER-LEVEL RELEASE STRATEGY OUT OF A NUMBER OF RELEASE POLICIES TO AUTOMATICALLY AND CONTINUOUSLY OPTIMIZE REVENUE

4. REDUCE THE IMPACT OF FAILING SOFTWARE ON YOUR CUSTOMERS:

You can be well on your way to improving customer experience and optimizing revenue, but if there's one thing that everyone in IT has in the back of their minds, it's that there will always be broken code. You know that, sooner or later, you're going to have bugs in production but you and your support team never know when the shoe is going to drop.

Everyone goes off on their weekend or holiday retreat with a vague nagging feeling that that new change to production is slightly unstable and might fail at any time. But if it does, they'll have to wait until customer operations reports it. When this happens it's up to support teams as the first line of defense to try a patch or start a potentially time-consuming, manual rollback procedure. If that doesn't work, it's up to you to take the escalation call, knowing that your customers are suffering from a malfunctioning platform.

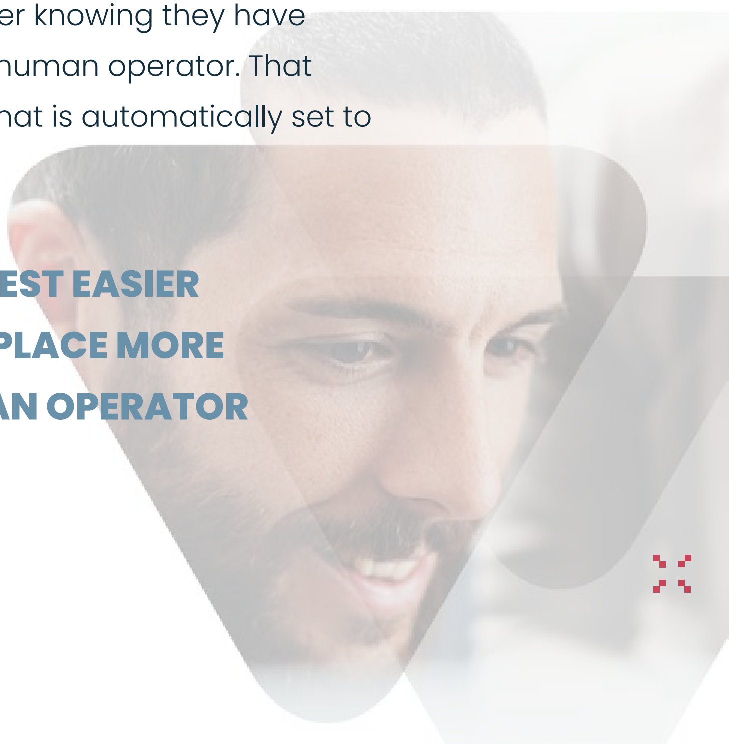
THE KEY? HAVE AUTOMATED ROLLBACK AND RISK MITIGATION PROCEDURES IN PLACE

With old technologies and processes, rollback procedures of new releases are manual and time-consuming. Hence all the built-in contingencies of pre-production testing. In addition, what if something breaks in production? There's no way of detecting it until customer experience and revenue take a hit.

By combining release policies with intelligent machine-learning-driven automation, today's release technologies ensure that systems are always online. Next-level solutions take the on-failure procedures baked into release policies and implement mitigation in real-time, before a human even has time to look.

What's more, with machine-learning capabilities constantly scanning the system landscape across all online platforms and devices, product owners and their service teams can rest easier knowing they have an automated safety net in place more vigilant and real-time responsive than any human operator. That transforms the entire process of software releasing from a risky endeavor into one that is automatically set to safely drive innovation forward.

PRODUCT OWNERS AND THEIR SERVICE TEAMS CAN REST EASIER KNOWING THEY HAVE AN AUTOMATED SAFETY NET IN PLACE MORE VIGILANT AND REAL-TIME RESPONSIVE THAN ANY HUMAN OPERATOR



FREE FEATURE RELEASE FROM THE DEPARTMENT OF ‘NO’

So where do these four key best practices leave you as product owner? Let’s cast back to the beginning of our story. Remember that production outage and stepping into your boss’s office post-event to answer for the damages? Since testing in production, customer segmentation, real-time data integration and automated mitigation remove many IT bottlenecks and the fear of software innovation, those kinds of scenarios can be a thing of the past.

These practices, and today’s release technologies that implement and automate them, create a separation of concerns between IT and business. That frees you as a product owner to fully carry out your role in pushing forward on customer experience and revenue. That frees IT too from having to play the role of the department of ‘no’. By adding a step to your software delivery process that extends your CI/CD pipeline with continuous release automation, everyone in the software delivery process can get excited about releasing at will and driving innovation forward.

ABOUT **VAMP** CLOUD-NATIVE RELEASE ORCHESTRATION:

RELEASE WITH CONFIDENCE - EVEN ON A FRIDAY AFTERNOON

Vamp helps businesses release features faster with 100% reliability for great customer experience. We are changing the face of software delivery. Vamp is smart release orchestration that takes over release decisions for you. Vamp is specifically designed to give product owners direct control over feature release, without having to rely on IT.

If you would like to learn more about how Vamp helps you release with confidence,
even on a Friday afternoon, go to
www.**vamp**.io