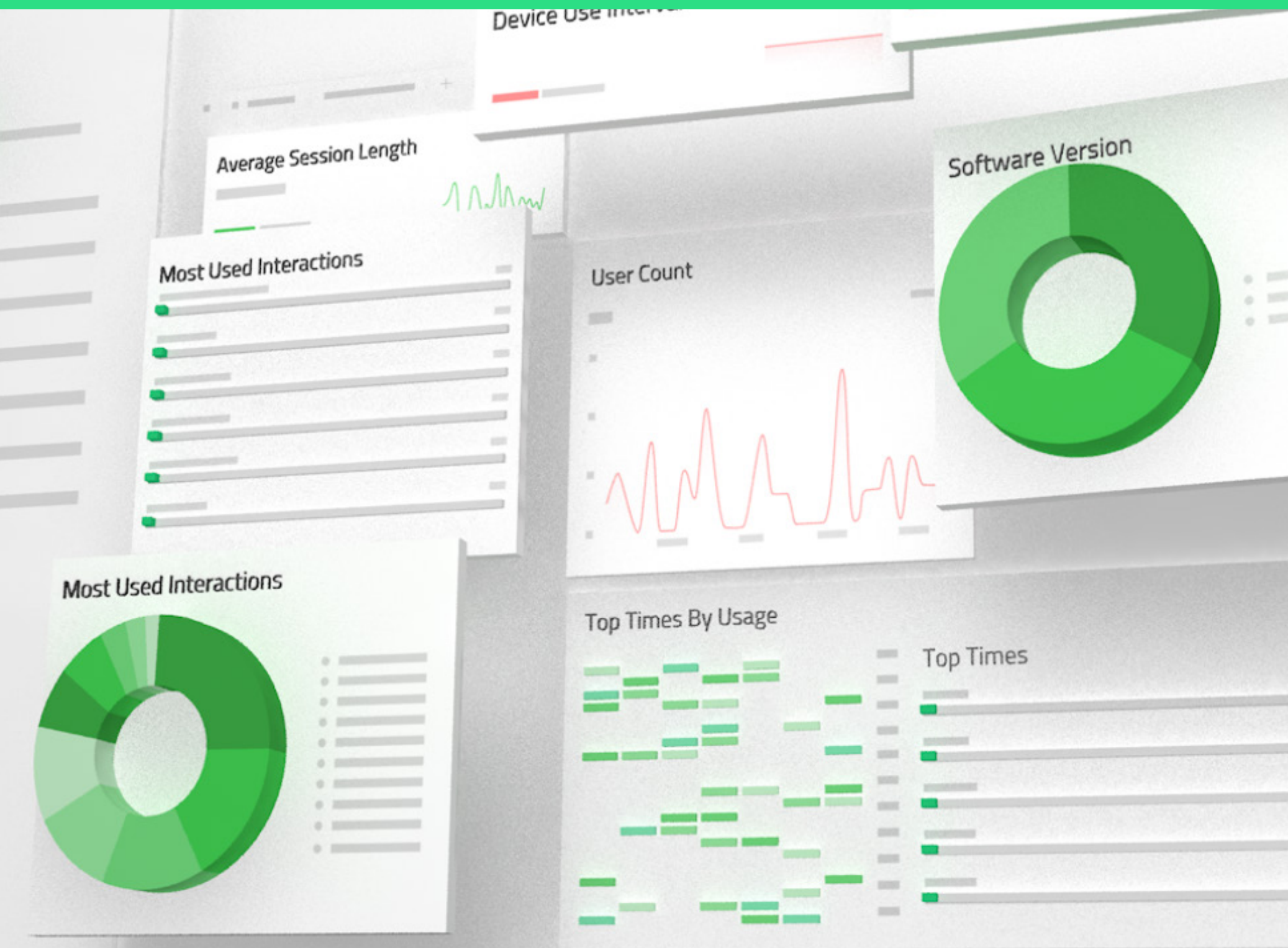




The Product Manager's Guide to Analytics



The Product Manager's Guide to Analytics

Content

Introduction	2
Product Analytics in the Product Design Phase	2
Field Testing Optimization	2
Onboarding Optimization	3
Product Analytics during the Product Lifecycle	4
Feature Adoption Analysis	5
UI Path Analytics	7
User Profiling	8
Time To Value Enhancement	9
Product Analytics towards the End of Life	10
Dead Code Removal	11
Key Performance Indicators for Product Analytics	12
KPIs for Commercial Product Managers	12
KPIs for Technical Product Managers	13
Product Analytics and Product Ops	13
Working with Qt Insight	14
System Architecture of	
Embedded Device Product Analytics Solutions	14
Getting Started in Product Analytics with Qt Insight	15
Product Analytics and AI – The Future	17



Introduction

Customer Experience, Product-led Growth, Customer Intimacy. All these concepts force a change in how product management is judged. Product Managers face increasing challenges in justifying their roadmap and prioritization questions. Product Analytics is becoming a powerful tool for learning the end user's everyday behavior.

This guide for Product Managers describes the capabilities of typical product analytics solutions chronologically along the product lifecycle. This document contains complementary information on standard analytics KPIs and the Product Ops team concept. Furthermore, this guide explains how Qt Insight, the product analytics solution of the Qt Company, is structured and how it can be taken into use.

Product Analytics in the Product Design Phase

Field Testing Optimization

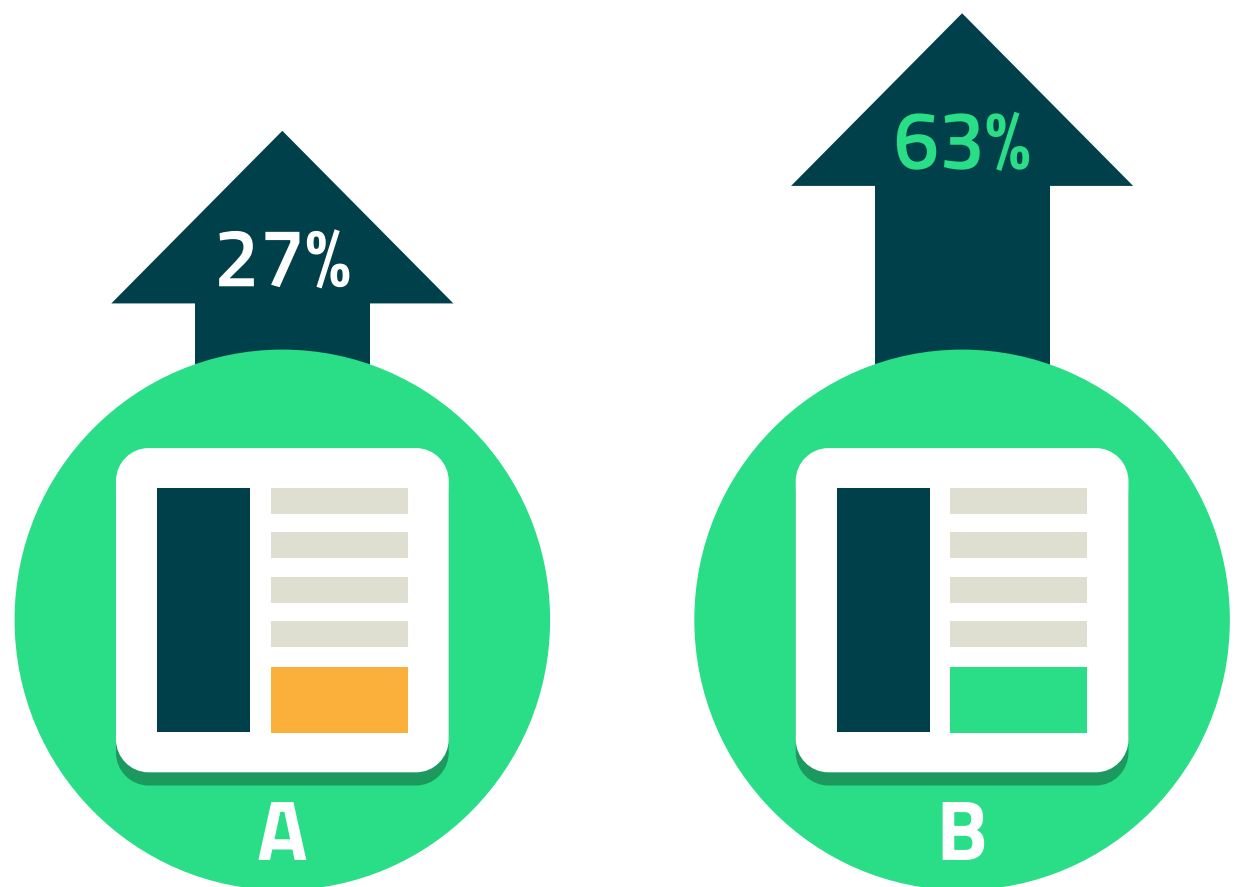
During the weeks of production ramp-up, with only prototypes and pre-production models available, Product Managers need more means to collect insight into the everyday use of the product. However, field testing with friendly pilot users, employees, strategic resellers, and key customers is perfect for finetuning the product through A/B testing.

A/B testing during the product design phase can be split into two categories:

- 1) A/B testing of different implementation options, where A and B refer to other software versions of the product.
- 2) A/B testing of the same software implementation under various conditions: A and B might refer to things such as different countries, different devices, or screen resolutions.

Product managers are unlikely to have the resources to build multiple features, test which one is more attractive, and only launch one of them in the final product. Such a luxury rarely exists. But testing which implementation option, for example, the placement of text or an interaction, can be beneficial to optimize the experience. This can be done during field testing a new product or major feature. By designating a different application version number to the various implementation options, Product Managers can study the behavior of users and decide on the ideal implementation option. Because these early devices or beta application releases are planned to be replaced anyway with the final product, no documentation of the implementation options is mandatory.





The field testing of prototypes can be further augmented by geographical or device information for deep dives into the behaviors of users.

Using A/B Testing, Product Managers can:

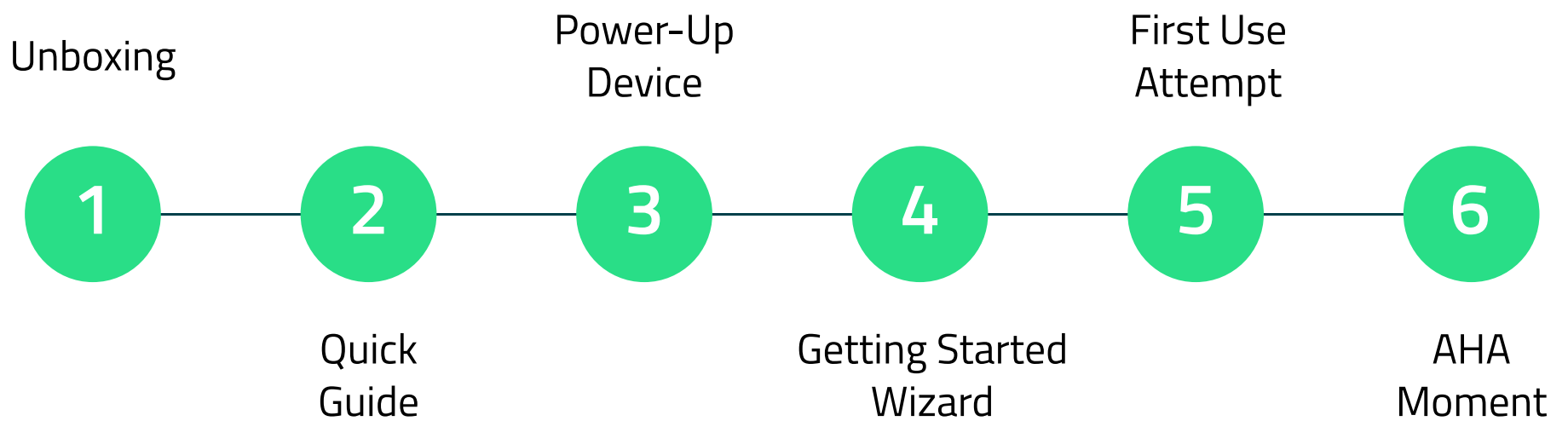
- Resolve uncertainties in the feature implementation that couldn't be resolved by referring to best-in-class solutions or user interviews
- spot market-specific differences in the use of the product due to cultural or other reasons
- Identify device-specific behavior differences caused by problems in the UI scaling or performance of the application on the target device

Onboarding Optimization

While the optimization of the UI navigation might focus on the everyday, repeated use of the product, onboarding experience optimization focuses on the first-use experience. The perfect everyday experience might not be the best onboarding experience. New end users may not find high-value features immediately or get stuck in the UI.

The primary task of onboarding optimization is to ensure that the first use of the product (or a new feature) is successful without manuals and deep exploration of the UI.

A smooth onboarding experience is also essential for enterprises executing a product-led growth strategy because it ensures that end users quickly get to the “aha moment” of the product. The “aha moment” is the point in time when the product meets the customer’s expectations.



Even a perfectly designed product with high usability can benefit from onboarding experience optimization because factors other than usability might drive the ideal product design optimizing business performance.

A superior onboarding experience helps to:

- Increase conversion from free trial offers to for-pay subscriptions because “sticky” features are used frequently during the early days of use
- Drive word-of-mouth marketing through influencers and others due to differentiating or “fun” features being easily discovered

Product Analytics helps to improve the onboarding experience by displaying how users navigate through the UI with a behavior flow chart.

In addition, Product Analytics solutions can show which features are most used by new users, indicating which features are often discovered quickly.

Onboarding Analytics allows Product Managers:

- To accelerate the learning curve of end users, together with UI Designers
- To find a good balance between everyday and first use
- To improve the conversion of free trials to commercial subscriptions
- To implement auxiliary tools improving first-use experience without hurting the everyday experience, such as wizards, tooltips, in-app messages, or interactive self-help guides

The faster the customer gets to the value of the product, the higher the customer satisfaction and likeliness to continue to use the product.



Product Analytics during the Product Lifecycle

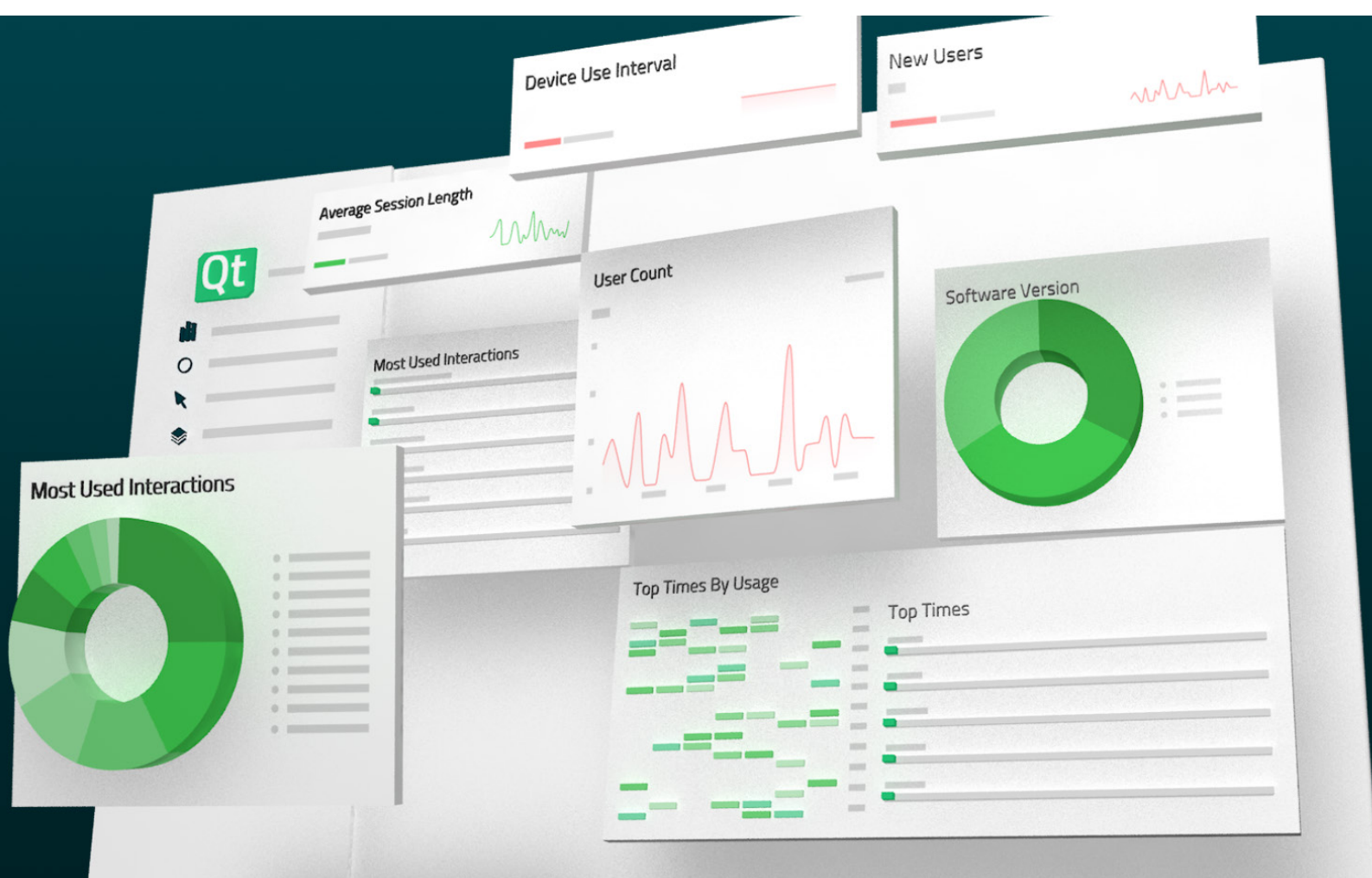
Feature Adoption Analysis

Getting the ranking of features in the backlog right is impossible. You collect input for the roadmap from multiple sources, including your customers, competitors, market analysts, and internal stakeholders such as developers, support engineers, and sales and marketing folks. And yet, the prioritization of new feature development is magic that sometimes is too complex to comprehend due to a large number of input factors.

You should match the input against your product strategy (or the next quarter's sales quota if you are more in a survival mode) and draft your feature roadmap. But, all this input information is biased. It doesn't matter how much time you spend with crucial customers, power users, or novice users of your product. The data will always depend on a subjective opinion as long it represents the opinion but not the actual usage patterns of selected users. It is more likely that end users will highlight what is missing or doesn't work as expected than bring up the things that work already and they use every day.

Feature adoption analysis with 24/7 tracking of the actual usage of your digital product opens an entirely new perspective into what your customers use, when, and how often. It gives Product Managers an idea of which capabilities end users embrace and which they ignore.

Product Managers are already now increasingly relying on product adoption. In recent years, tracking product adoption from interactions such as software installations or logins has become much more relevant for SaaS solutions and connected embedded devices. However, feature adoption still needs to be understood better in many industries to impact the weekly backlog decisions.



Product analytics solutions record and visualize feature adoption across geographies, user groups, and devices over time. Feature adoption analysis records an event for every user interaction on product elements you marked as relevant to be studied. These events are anonymously uploaded to a cloud solution and then aggregated to meaningful charts. Suddenly, you will understand whether a feature is used within the first 2 minutes of the customer journey, is used 20 times a day, is used a second time (ever), or is used only once a year.

In general, we can distinguish between 4 types of feature adoption:

1) Depth of Adoption:

How often do end users interact with the feature? This measure directly indicates how valuable a feature is for different customer segments. The more a feature is used over time, the more likely it is to contribute to the product's stickiness and, therefore, its retention.

2) Breadth of Adoption:

How many of your end users are adopting a particular feature? Are all customers using the feature regularly, or is it only a specific group of people using it, but then intensely? Breadth of adoption analysis might give you insight into customer segmentation and indicate potential features for premium pricing.

3) Time to Adoption:

How long does it take end users to make use of a feature after the first use of the product or after the software upgrade of the product? The time to adoption metric delivers information on how well the feature is accessible in the UI navigation. Furthermore, it gives product managers data on how much needs of the end user the feature addresses: A feature the end user is spending more time to find is likely to be such that it is more relevant than some feature which is discovered only by "accident." Measuring time to adoption also helps to plan product marketing activities. Features that are known to drive value but are not discovered easily enough can benefit from complementary marketing activities.

4) Duration of Adoption:

How long after the initial use are end users interacting with a feature? Is the feature only used due to its initial novelty a few times in the first days, or does it provide real everyday value?



Using a feature only once a year (or even less) is okay if that's what you want. Let's imagine you have a feature that allows your product to be reset to its factory settings if the end user changes. You really want this feature to work to give your product a long lifetime value and support a cyclical economy, but you wouldn't expect it to be used frequently. Whether or not the frequency of product usage meets your design expectation is what matters.

Features that you thought deliver a lot of value but which find little adoption in everyday life are such that deserve your attention. If users need help finding or understanding functionality, you must ensure that you place or communicate it better. Alternatively, the feature might not be as valuable as intended, and you need to reconsider whether you defocus the enhancement of this feature and reposition the feature in the UI navigation not to take up unnecessary real estate. You do not want to build a bridge to nowhere by extending a feature that nobody uses.

Feature Adoption Analytics enables Product Managers:

- To justify better the backlog ranking to other stakeholders based on everyday end-user data
- To understand more precise which bugs in which features to fix first because of the improved understanding of the impact on everyday usage
- To identify premium features used frequently by a particular customer segment

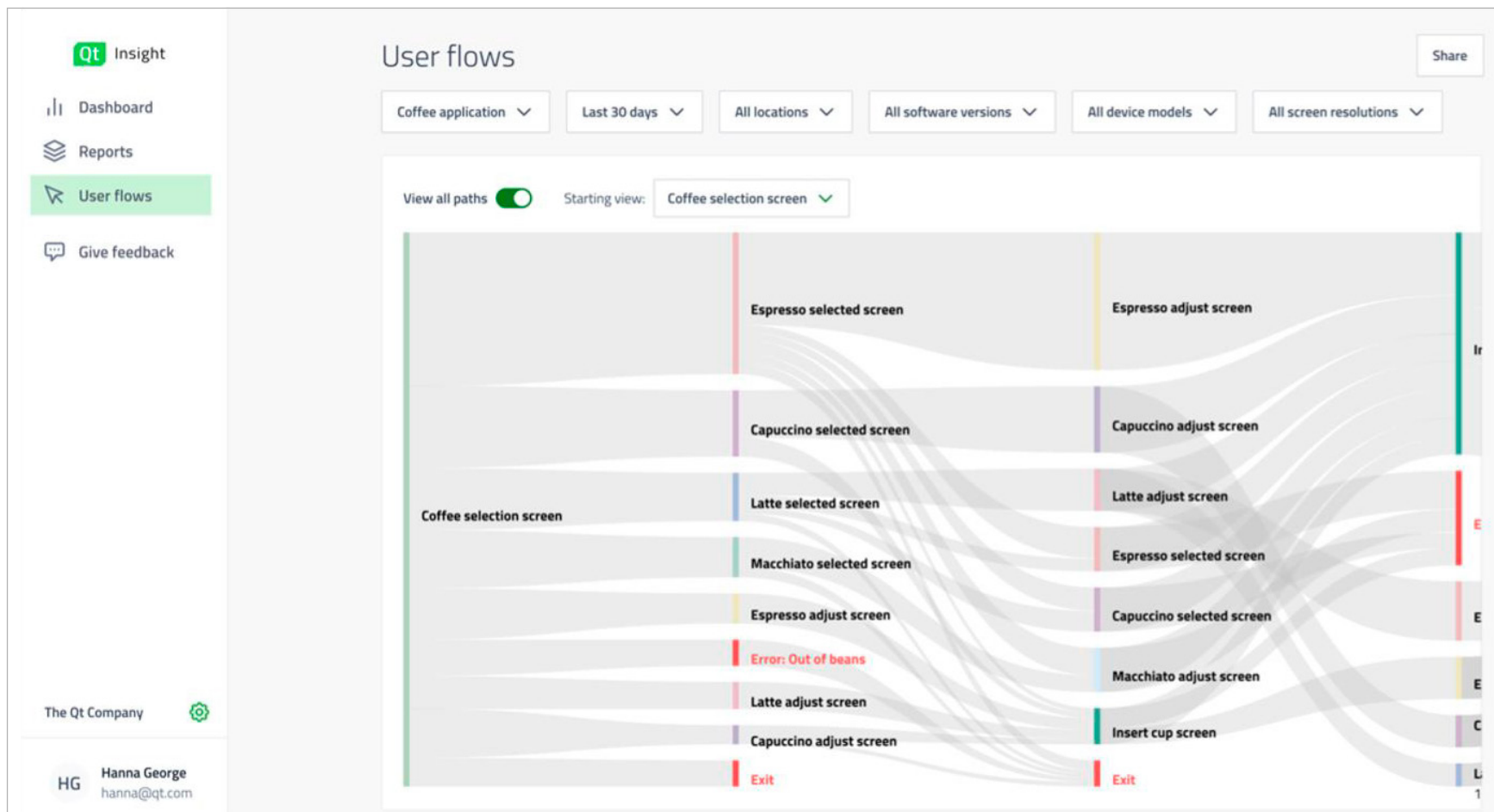
Product Managers can use feature adoption analysis to fight against becoming a feature factory. Feature adoption analysis serves as a complementary input for making better product decisions.

UI Path Analytics

Understanding which features are hot and which are not is complex, even with Feature Adoption Analytics. Feature Adoption Analytics "only" delivers the total number of interactions but doesn't give the Product Manager information on how much energy the end user had to invest in making these interactions. Was the feature available from the home screen or buried three levels down in the UI navigation? Comparing the total number of interactions without context might be dangerous.



UI Path Analytics can supplement feature adoption analysis and give valuable information on which features are buried too deep in the UI. Product Analytics solutions can provide insight into the behaviors of end users. The solutions display how customers navigate from the home screen through the UI during a particular session. The solutions display which path users take and how many users choose which path, from opening the application to exiting it.



UI Path Analytics is also essential for UX Designers. Seeing how cohorts of users navigate the user interface day in and day out can be truly enlightening. In a perfect world, Product Managers do not need to get involved in UI navigation optimization. But in many enterprises and software development teams there is no (dedicated) User Experience specialist. In other software creation teams, the UI designer focuses only on creating meaningful and beautiful interfaces without looking at the bigger picture. When this happens, and this might be the result of the UX designer being only available during the initial product design, then the Product Manager itself can benefit from UI Path Analytics to simulate the role of a UX Designer.



With UI Path Analytics, Product Managers learn:

- What it takes for end users to generate the measured amount of feature interactions giving context to Feature Adoption Analytics
- Where are “dead ends” in the UI navigation when end users jump off and exit the application prematurely
- Where are opportunities to create UI shortcuts or reorganize the UI navigation for frequently used interactions which require the end user to perform multiple interactions to even get to the desired feature

UI Path Analytics is a continuous effort. It is precious in the early stage of the product lifecycle. With every significant change in the UI, it should be studied whether the product ideally supports the behavior of users. Even at the end of the product’s lifecycle, UI Path Analytics can find features that end users have grown accustomed to over time and still deliver value.

User Profiling

Your customers might be different from the ones you thought. As a Product Manager, you should have defined your buyer personas for your Go-To-Market strategy. But based on what information have you drafted these personas? How many of your future customers have really met in person? Depending on your product, your buyer might be someone other than your end user. Interviewing people understanding their behaviors and to which group of customers they belong is hard to scale on a global scale and, therefore, still needs to be done.

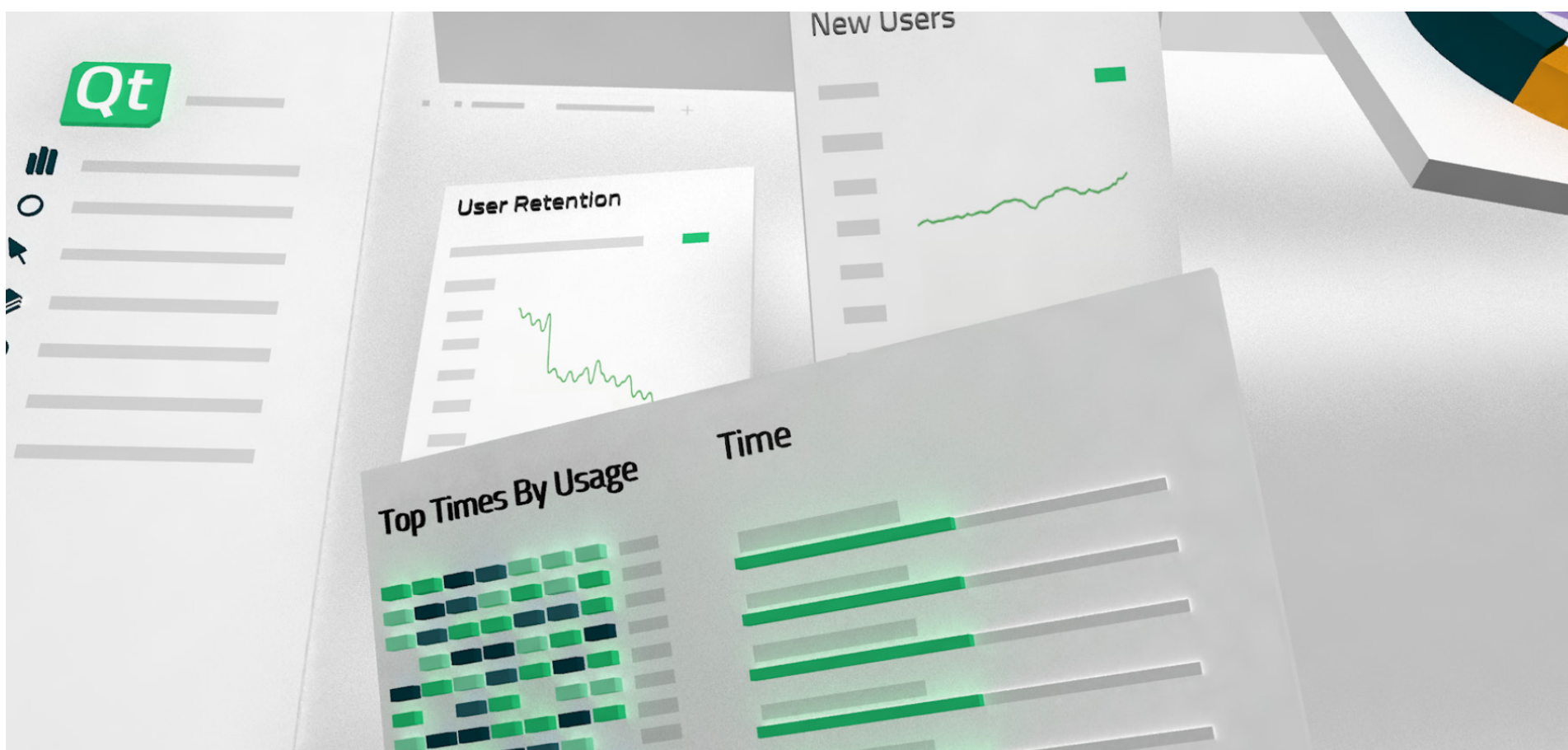
Your opinion on who your end users should be can be biased also by the vision major resellers have for your product. It had happened before in product creation history that a product was designed for a specific customer segment solving a particular end user need. Still, it ends up being used ultimately by a very different group of users. For all these reasons, your user profiling, which you use for marketing, pricing, and UI design purposes, might need to be revised compared to the real world.

Product Analytics can help you with user profiling. Product Analytics solutions typically work with anonymous data for privacy reasons and give you no direct indication of things like age, gender, spending power, etc. However, these solutions can provide Product Managers end user data that can be correlated to meaningful groups.



The following Product Analytics data can help to make conclusions on the customer profile:

- 1) Geography/country information: Is the user in a country with a high or low Gross Domestic Product? This can give you information on the potential spending power of such users.
- 2) Device type/screen resolution: The device type can indicate age, gender, and spending power. A MacBook user is likelier to have decent spending power for your desktop application. An end user with an embedded device running a larger screen is more likely to spend additional money for a subscription-enabled premium feature.
- 3) Utilization rates can also serve as a base for user profiling: Based on the total number of sessions, session length, and the number of interactions, Product Managers can extrapolate on who are power and who are



Product Analytics can help in user profiling with data correlated with particular personas based on consumption patterns and geographic and device information. User profiles derived from Product Analytics serve as input for pricing, positioning, and roadmap decisions.

Time To Value Enhancement

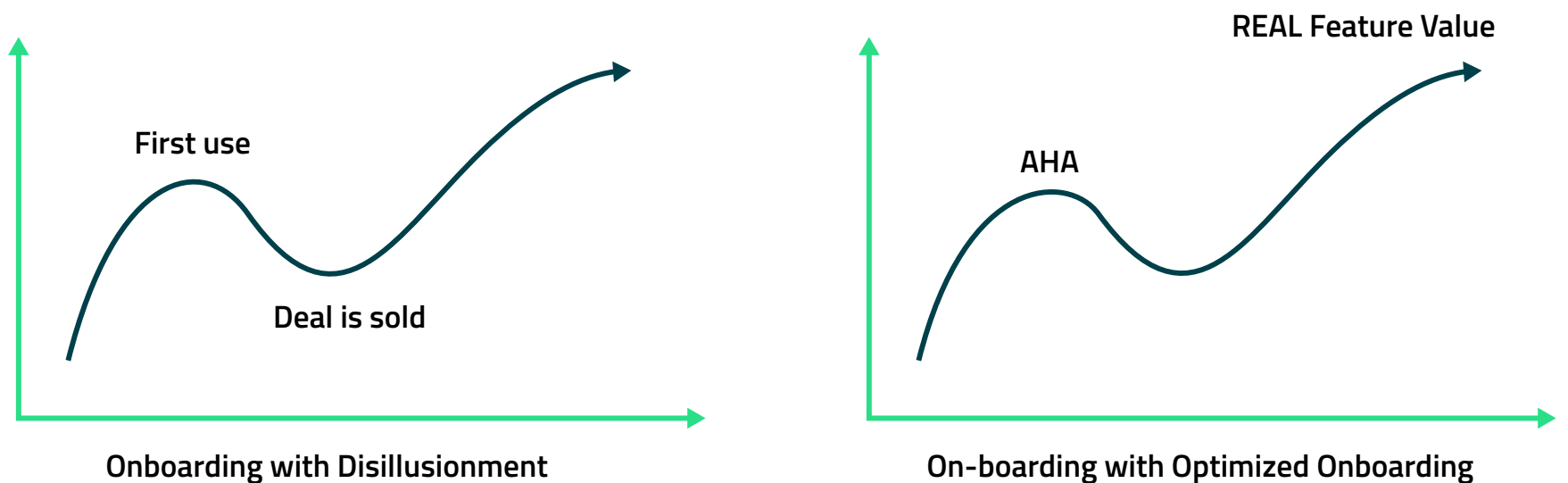
Your product might have a laser-focused onboarding experience, optimized usability for all user groups, and many features. Yet, customers do walk away from your product after the trial period or after days of using your freemium version. What has gone wrong?

While customers may have reached the “aha moment” quickly through an optimized onboarding experience, they still have not gained the product’s actual value. End users might get an initial perception of the value that your product messaging promised, but the product has yet to deliver any real value. Therefore, your product or premium feature might not



be sticky enough yet, and customers are walking through those revolving doors of customer retention.

Time to value is vital in getting something valuable back from your trial or freemium users (money, promotion, content). Time to value defines the time for real value to be derived from your product. It is the time it takes for a real customer problem to be solved or the time a tangible delight has been delivered. It's not the perception of the value but the value itself that matters. How quickly this can happen depends on your product.



Product Analytics solutions can help to cut down the time to value. Product Analytics support time to value enhancement with event-based and UI path tracking.

By setting tracking events indicating when the value has been achieved, Product Managers can measure whether the value has been gained at all for new users.

What to track depends on the product. A successful first-value experience of a taxi application might be the taxi dropping you off at the place you want to be. The first use of maximum acceleration of your new car can be equally such an event if that is a premium feature that can be trialed.

The UI paths to value can be optimized by studying end user behavior charts.

Can you spot in the UI paths that some end users are distracted by competing features that do not deliver value quickly? The UI path to features delivering your product's expected (and messaged) value should be easily discovered and quickly navigated.

And suppose none of these methods bring that all-important feeling of success to the end user. In that case, the Product Manager can and should consider whether additional, related features are necessary to deliver value faster. Some products are simply such that it takes a reasonable amount of time to derive value. A financial accounting application might only deliver value once a quarter. If the application is trialed at the beginning of the quarter, the time to value can be rather long. However, a simulation of expected results based on historical data and growth expectations might deliver substantial value faster.

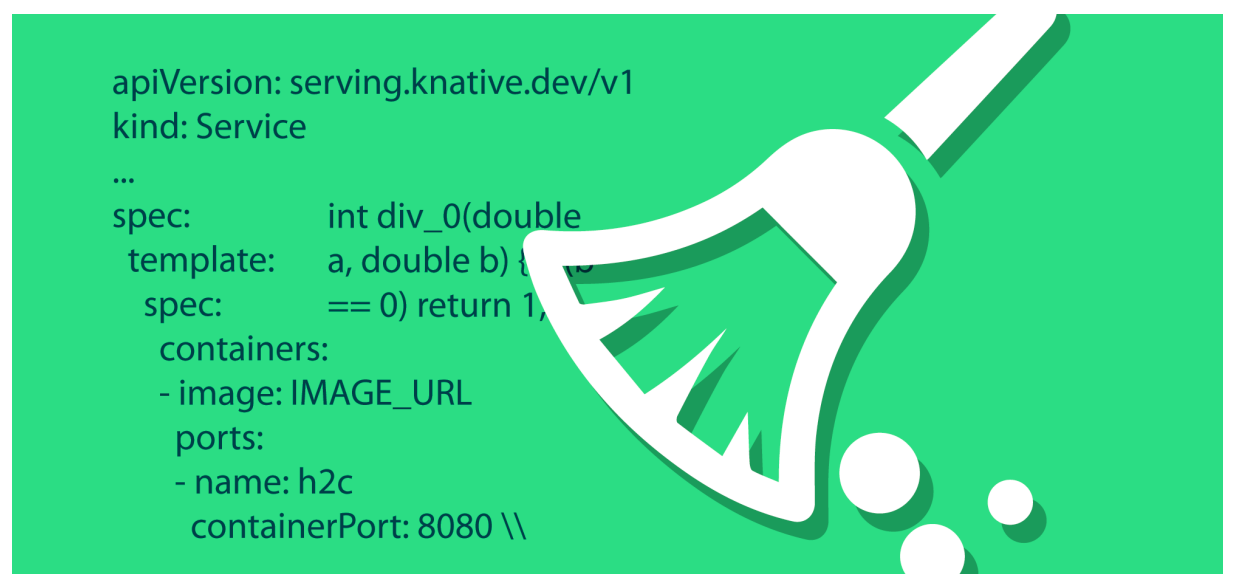


Product Analytics towards the End of Life

Dead Code Removal

According to studies at the Technical University in Munich show, 25% of code in enterprise software* is never used? Furthermore, according to other research, 80% of code in cloud solutions** is rarely or never used. This is especially true for digital products towards the end of their lifecycle.

If 25% of the software code is not used, then Product Managers carry a significant amount of dead weight. Unless the dead code is identified and removed, Product Managers (and the software developers) keep maintaining it. The maintenance includes upgrading third-party libraries, scanning the unnecessary in security and penetration testing, keeping its unit and graphical user interface test cases updated, and modifying it for newer programming language versions or compilers. So, even if nobody reports any bug for dead code – which isn't a surprise because nobody uses it – there is still a significant amount of work related to it. Not to mention that the dead code bloats the software, which causes unnecessary storage or memory consumption.



There are tools other than product analytics to identify dead code. Static code analytics can also find dead code. This is done by reachability analysis for dead software functions. These functions will never be called and, therefore, will never be executed because there are no connections from the program entry points of the system to these functions. This is a great complementary instrument for identifying dead code if the digital product is mature. This is particularly true because product analytics relies on events attached to features, which might be challenging if the code is not even reachable. However, static code analysis cannot flag features and their related code, which are hypothetically reachable, but users cannot find them or simply don't need them.



Using feature utilization analytics, Product Managers can:

- improve software's comprehensibility, testability, and maintainability by removing code of underutilized features
- identify features that can be potentially sunset
- complement static code analytics to fight software erosion over time

**Source: Research of Technical University Munich, "How Much Does Unused Code Matter for Maintenance?", 2012*

***Source: Study of Pendo.io, "The 2019 Feature Adoption Report", 2019*

Key Performance Indicators for Product Analytics

"Measure only what you can act on!" is a good rule for setting the measurement points of Product Analytics. There is a plethora of things one can monitor with Product Analytics. In general, Product Managers should only measure events and UI paths of interactions that are likely to lead to new insights. While it is quick to add a tracker event to an interaction, there is an incremental cost to doing so, as small as it might be. Therefore, Product Managers should initially consider which information is likely to lead to action and which is of a "nice-to-have" nature. The following two chapters give some suggestions on key performance indicators which Product Analytics commonly measures.

KPIs for Commercial Product Managers

Some organizations share the responsibilities of Product Managers across Commercial and Technical Product Managers. This chapter focuses on more business drivers relevant commonly to Commercial Product Managers.

Growth: Number of new users/week. Which company isn't interested in growth? This may be the most fundamental factor indicating overall demand for the product, even when some of the user growth is derived from freemium or trial users.

Churn: Number of inactive users/week. Another commonly measured metric, not only for SaaS businesses, is retention and churn. How many customers have stopped using the product altogether during an interval indicating inactivity? The interval length can range from a week to a year, depending on the usual consumption patterns.

Customer Segments: Number of users/device, country, or version. Customer intimacy starts with understanding your customer. Product analytics can help by showing characteristics such as device type, display resolution, country, or software variant indicating an association with a particular buyer group.



Killer Features: Number of user interactions with a particular feature / 12 months. Which are the features driving the most interaction? Does the 80/20 rule apply that 20% of features generate 80% of interactions? Know what to promote and what features to keep competitive.

Premium Features: Number of user interactions with spending power / 12 months. Product analytics can help carve out features that might be bundled in a premium licensing package.

These might be features that are only used by some but a selected group of users. Features used by everybody can be considered of essential value that cannot be sold in a premium package. Features that are occasionally yet frequently used by a subset of users could be such that they can be bundled in a premium offer.

KPIs for Technical Product Managers

Feature Adoption: Number of users/feature / total population. This is the most basic metric. How many users have adopted the feature at all, at any given time? In isolation, this metric needs to be interpreted with care, but with more context, such as the number of interactions per feature per user, this metric becomes more relevant.

Feature Use Frequency: Number of interactions/feature/user. Is the feature used as often as expected? While this is not a direct indicator of importance, it gives essential information on whether the frequency of use of features matches your plans. Low utilization might be desired, such as a recovery feature for a cloud solution. You don't want it to be used (ever) ideally, but when needed, you want key users able to activate it.

Breath of Adoption: Number of major features in active use/user. Product Managers should know which features enjoy high utilization by most users and which are only relevant for some users. Furthermore, this metric might indicate whether your product has become too complex for all users to enjoy all features. Some of the most successful solutions focus on doing a few things and doing them well.

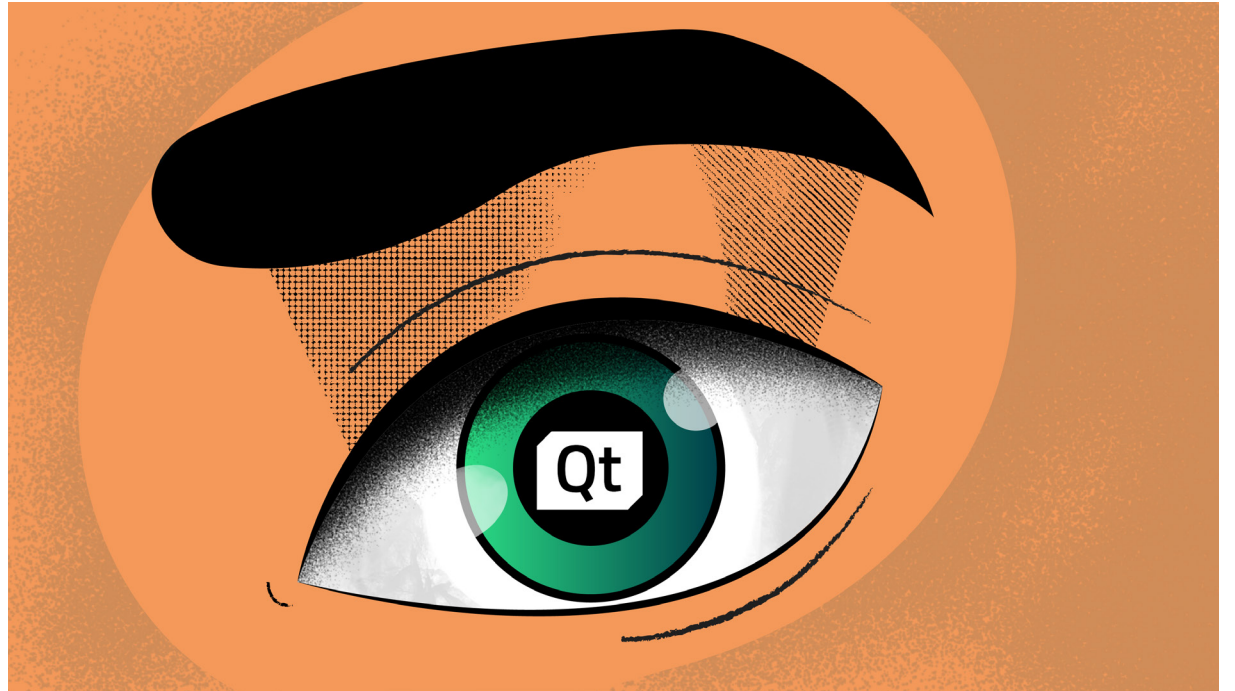
Time to Adopt: Time to use a feature for the first time. Is the time it takes for the target users in line with your expectations? Does it take longer than you expected? Product Managers should watch this metric closely to identify optimization opportunities for the onboarding experience.

Continued Adoption: The time period in which a feature is used frequently. How long after the first use do users still return and use the feature? Is the novelty wearing off quickly? Is there no daily value? If so, do reconsider whether it makes to extend and keep that feature alive.



Product Analytics and Product Ops

Introducing Product Analytics is a great way to shift your organization into a product-led powerhouse. Product-led is often considered as laser-focusing all business on the product's success. A Product Ops team consists of all product stakeholders getting together to improve the product performance based on new data on the behavior of end users.



A Product Manager might already participate in some Product Ops team without calling it such. Product Managers in agile development teams might meet weekly with their support and engineering team colleagues to review and rank the most recent incoming bugs. If these teams working on improving the product exist already, then they are a perfect place to discuss also the insights gained from product analytics. Typically, Customer Success, Marketing, Engineering, and sometimes even Sales members are great participants in a Product Ops team. With enough focus and preparation of the raw data, such meetings can be quarterly extended or refocused to review the product analytics' information within 30 minutes.



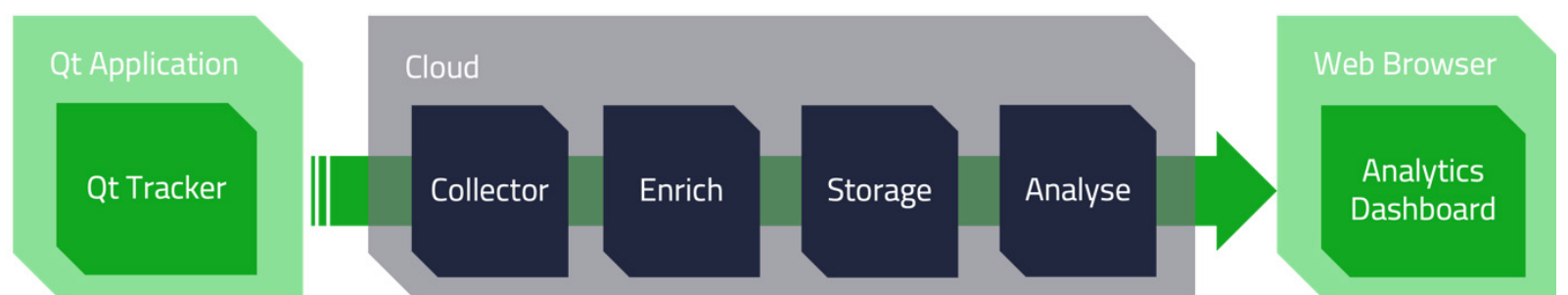
Working with Qt Insight

Qt Insight is a Product Analytics solution supporting especially connected embedded devices and desktop applications built with the Qt framework and QML, the Qt UI Modelling Language. The following chapters provide Product Managers with general knowledge of Qt Insight. However, detailed user and development guides for Product Managers and Developers are available separately, either as in-product self-help or online documentation.

System Architecture of Embedded Device Product Analytics Solutions

Qt Insight consists of three main components:

- 1) The Qt Tracker library
- 2) The Qt Insight Cloud
- 3) The Qt Insight Analytics Dashboard



The Qt Tracker is the software library that takes signals from the Qt application, caches them, and sends them to the Qt Insight Cloud. The Qt Tracker is embedded in the Qt application and runs in each device or application that shall be monitored. The Qt Tracker supports tracing UI interaction events, such as pressing a button or registering a touch display event. Furthermore, UI screens can be marked for tracking, allowing UI paths from the home screen to the application exit to be monitored.

The Qt Insight Cloud consists of a few sub-components with a specific purpose:

The Collector is a scalable cloud component that takes events from the trackers and streams them forward.

The Enrich component is a stream processor that handles raw data from the Collector, flattens it, enriches it from sent metadata, and streams them forward.

The Storage component is a stream processor that stores the enriched event data for analysis to cloud storage.

The Analysis component computes the raw data into separate analytics tables for the different metrics.

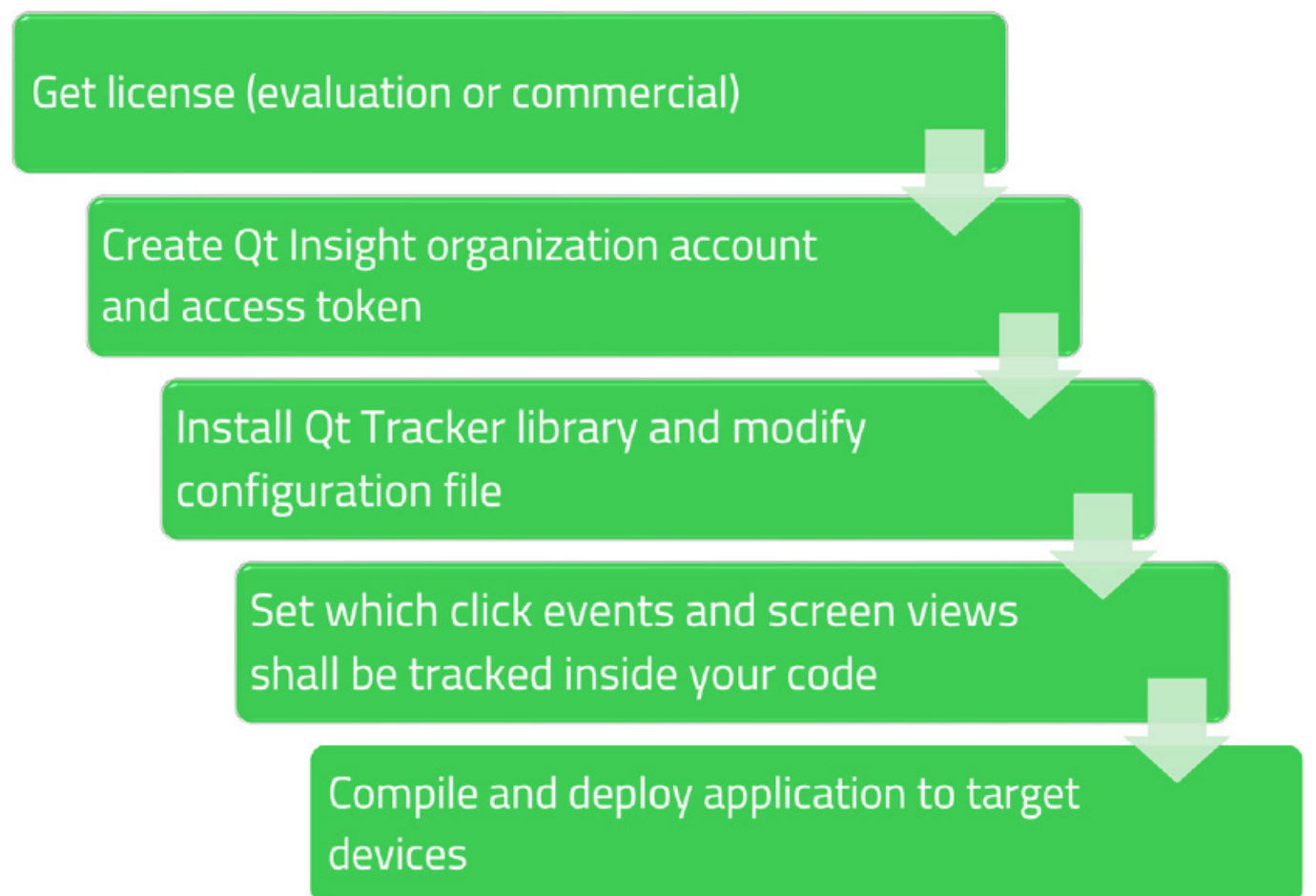
The Qt Insight Analytics Dashboard is the tool users need to access Qt Insight data. The Analytics Dashboard runs on standard web browsers. Product Managers can visualize, filter, and export raw data with this application. The web application has dedicated charts for reports on



pre-configured analytics metrics. The latest version of Qt Insight might only support some analytics mentioned above KPIs and further development is happening rapidly. The web application also has a dedicated view for visualizing UI paths.

Getting Started in Product Analytics with Qt Insight

It takes roughly five steps to benefit from your first product analytics results. This chapter introduces the required steps to get going.



Get a License: You can quickly get an evaluation license from Qt’s web pages by entering your company email address. After your evaluation license has been granted, you will have a valid Qt Account registration which will serve as a door opener for all further actions.

Create a Qt Insight Organization Account: Besides your personal Qt account, your team needs an organization account for Qt Insight. You can create this from the Qt Insight console by logging in with your Qt account information and entering a new Qt Insight organization name. This will make you the administrator of that organization’s account. This will also create the all-important access token that links the Qt Tracker data to the Qt Insight Cloud and then to the Qt Insight Analytics Dashboard.



Organization created

To start collecting analytics from your applications, you need to connect your application to Qt Insight. This is done via adding your organization token to Qt Designer or QML. You can see and create tokens from your organization settings.

Your organization token:

To add members to your team, use organization settings.

[Continue to dashboard](#)



Set Click Events and Screen View Events: The developer needs to mark the UI interactions to be tracked. This can be done on the code level by adding a copy/pasting a few lines of code, or it can be done in the Qt Design Studio on a graphical user interface.

Compile and Deploy Application: The final step is business as usual for your developer. Compile the application and deploy it on the target hardware. Distribute the app or devices to end users. When an end user is powering up your app, Qt Insight collects data and sends it to the cloud once it has a connection.

Install Qt Tracker Library and Configure Setup: You (or more likely your software developer) need to install Qt Insight Tracker from the Qt Maintenance Tool (also known as Qt Installer). It can be found under Additional Libraries. The next task is configuring the Qt Tracker library by entering the Access Token and additional information, such as the software version of your application, device type, operating system, or screen resolution.



Product Analytics and AI – The Future

Today's Product Analytics solutions focus on big data collection, intelligent algorithms consolidating the raw data, and meaningful visualizations. These solutions help Product Managers to measure and interpret known patterns. Known patterns are everything we have described in the KPI chapter of this guide.

Product Analytics solutions of the future will break through the limitations of known patterns and alert Product Managers of unknown patterns. Future solutions powered by Artificial Intelligence might use anomaly analysis to identify new emerging behavior. Unknown yet interesting end user behavior patterns might be user errors happening infrequently (remaining undetected) but consistently over a more extended period. Alternatively, unknown patterns might be that end users repeat the same steps within a short timeframe while a single interaction would be expected. This might indicate that the end user did not get the results/he expected and keeps on trying.

Anomaly analysis, or Artificial Intelligence in general, will enhance Product Analytics in the coming years. Whether Product Managers have the time and resources to interpret and act on all the knowledge gained from advanced Product Analytics remains to be seen.



