HOW TO PACKAGE & PRICE EMBEDDED ANALYTICS

Practical Frameworks to Monetize Embedded Analytics

Logi ANALYTICS

SoftwarePricing PARTNERS

Table of Contents

01 Embedding Analytics to Lift Value and Revenue

Today's organizations are embedding analytics in commercial software applications at an exponential rate. According to the 2017 State of Embedded Analytics Report, 93 percent of applications offer some form of embedded analytics capabilities. What's more, these features contribute 54 percent of the overall value of software products, as estimated by independent software vendors and SaaS companies.

However, the added value of embedded analytics doesn't always translate to increased sales revenue. Many companies struggle to integrate analytics into their products, business models, and pricing strategies in a way that meets a range of user needs and drives top-line growth.

One challenge is that different software users perceive the value of analytics features differently. For some, a customizable dashboard of real-time metrics inside their primary application is all they need. Sophisticated capabilities such as write-back and workflow actions would rarely be used, if at all—so customers don't want to pay more for them. On the other hand, some customers absolutely need advanced capabilities like embedded self-service and the means to pull new data sources into the application—and they are happy to pay more for these sophisticated features.

How can companies handle this range of customer needs and their willingness to pay more for analytics? One of the best methods is to **create tiered offerings**. These are packages of analytics enhancements comprising feature sets of increasing value, priced accordingly.

Are you packaging and pricing for the masses or the few?

The two most common mistakes when pricing and packaging embedded analytics are:

Charging Too Much. If you significantly increase the price of your enhanced product for all your users, you risk upsetting customers who don't care about and may never use the new features. Customers may abandon your application for a competitor's if they're paying for features they don't need.

Charging Too Little. On the other hand, if you decide not to raise the price or only increase it slightly, you could leave enormous amounts of potential revenue on the table.

02 Customize to Monetize: The Strategic Way to Create Tiered Offerings

To create tiered offerings that help monetize your embedded analytics, you'll need to do some customization. This doesn't mean building a different offering for each type of user you want to target. Instead, build analytics enhancements just once, and use them strategically. With this approach, you turn features off and on through visibility controls, conditional logic, and your application's security model. It allows you to offer some features to everyone and limit access to more advanced capabilities only to certain (paying) customers.

Customizing embedded analytics in this way means thinking about packaging and pricing early in product development cycles. Too often, packaging and pricing decisions that are fundamental to software monetization are made as an afterthought. This is especially true when it comes to enhancements like embedded analytics. Teams of agile developers race along the product roadmap to deliver new features, but often wait until the last minute to decide how to price these features.

This ebook provides simple frameworks to help software vendors effectively package and price embedded analytics from the start.

03 Packaging Decision Framework

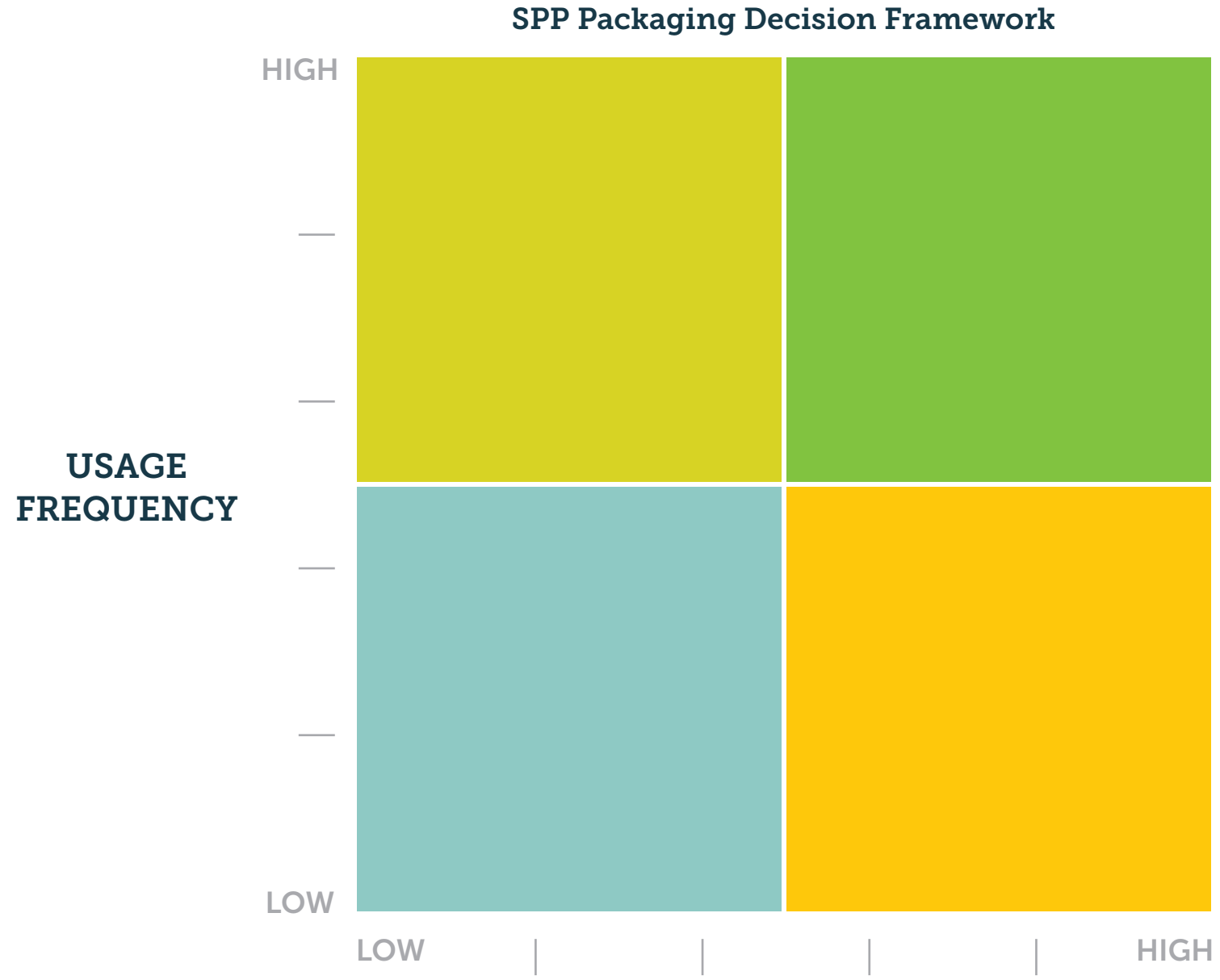
To add embedded analytics into multiple offerings, you first need to separate your user base according to usage and perception of value, including application features and embedded analytics features.

A good place to start is the Packaging Decision Framework created by Software Pricing Partners (SPP). This deceptively simple framework is actually a dynamic tool for organizing thinking and action. It's a method to help you capture how customers perceive value, how your customer mix evolves over time, and how competitive forces affect your packaging efforts.

In the pages ahead, you'll learn how to use this framework to steer internal stakeholder teams toward consensus and your product roadmaps toward monetization. We'll also show you how to cut through the fog of the many packaging and pricing possibilities to identify your best options.

Discover which analytics features will carry your application into the future >

03 Packaging Decision Framework



Plotting Features

To start working with the framework, make a list of all the features you can or want to offer, including embedded analytics capabilities. Place a dot for each feature in one of the quadrants based on how often people use it (x-axis) and how valuable they perceive it to be (y-axis).

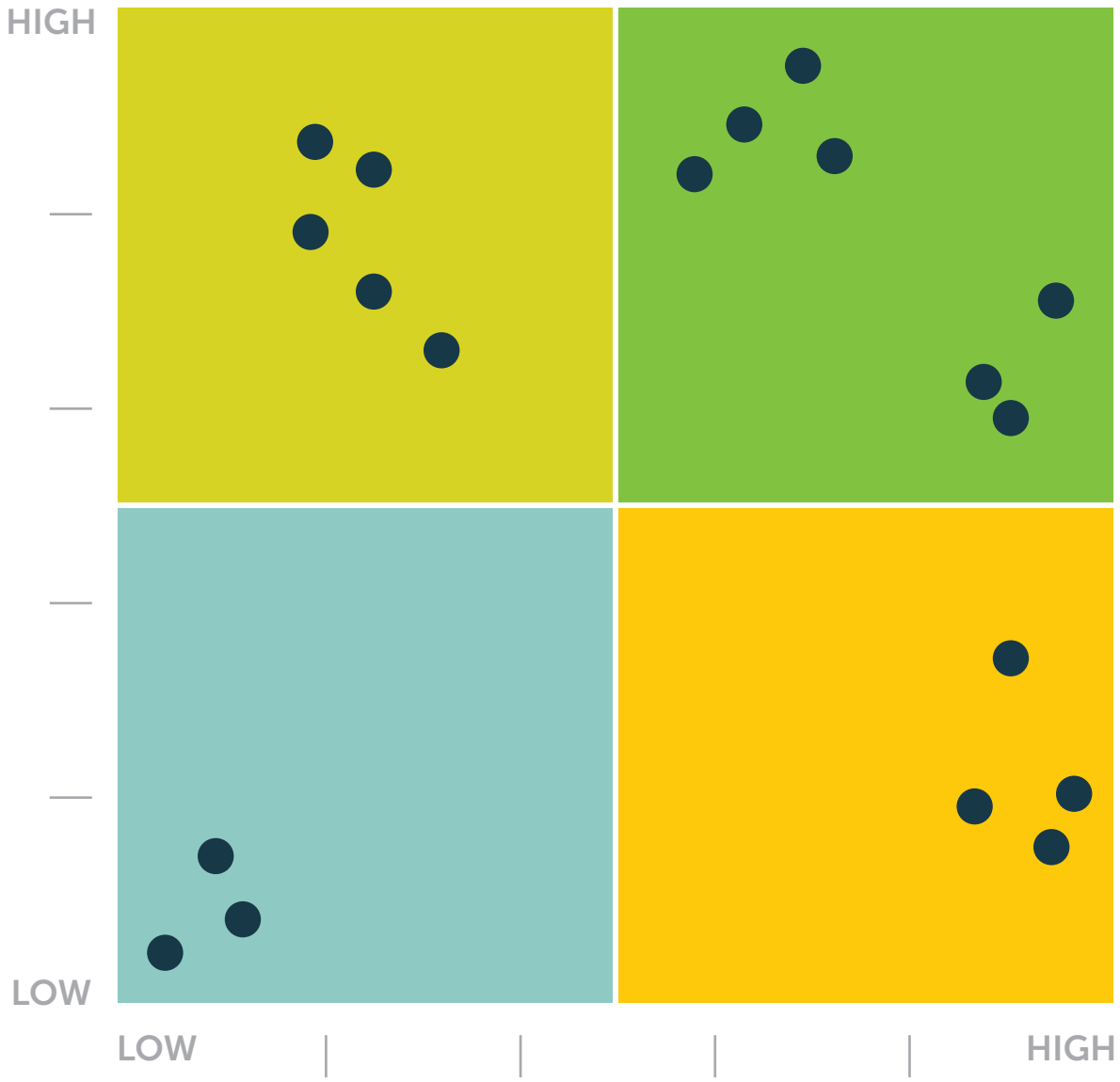
To get the position right, you'll need to do some research into the main user classes that make up your customer mix. These are comprised of users who are similar in terms of how often they use (or will use) each feature in your application and how valuable they perceive these features to be.

User Classes vs. Marketing Buying Personas

While companies may buy your software product, it's the end users who ultimately use the features. In fact, some buyers may only see a demo and never actually use your software. This is why relying on marketing buying personas to drive pricing is a huge mistake. It's more powerful to think in terms of monetizing use of your software via user classes (which often include a number of marketing buying personas). Attempting to use marketing buying personas in monetization decisions introduces unnecessary risk and results in incorrect (and sometimes dangerous) decisions.

Develop your user classes by interviewing a representative sample of users within your customer mix. This sample doesn't have to be large. About six to a dozen conversations per market segment usually do the trick. A few well-executed interviews can beat out hundreds of surveys. It's crucial to engage your users live, especially the first time through this framework. Having a conversation where the user does most of the talking is the best way to discover what he or she really thinks.

03 Packaging Decision Framework



Try to understand the context of the business problem users are trying to solve with your solution. This is key to accurate research. Listen carefully to what is said and unsaid.

Don't forget to reach out to internal experts on your executive team and in tech support, customer service, product development, and other customer-facing roles. Their understanding of customers will include deep insights that are important to capture.

Remember to use good judgment.

At the end of the day, you still have a business to run and a profitable product to make. Monetizing is more about carefully balancing customer input with your goals and objectives. The most successful software companies learn how to effectively balance these competing factors while keeping customers happy with their products and services.

Determining Tiers

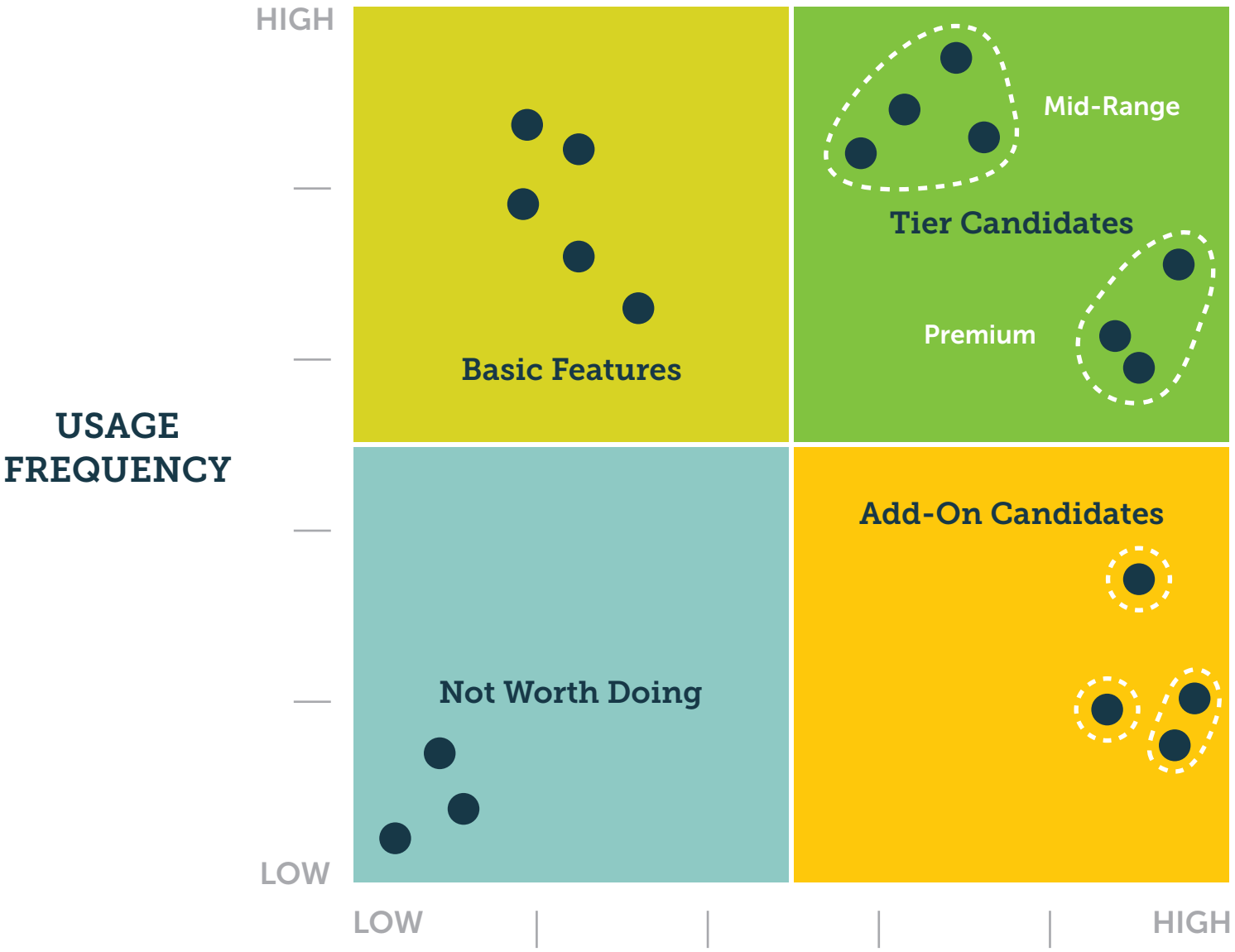
Next, group each of the features from your product roadmap (represented by dots and dotted cirlces) on the framework, suggesting possibilities for structuring your analytic enhancements.

- **High Usage/Low Value (upper left quadrant):** These features are your basic must-haves. They're likely to be the foundation for all your offerings, but are unlikely to warrant a higher price tag.
- **High Usage/High Value (upper right quadrant):** These features are candidates for premium tiers.
- **Low Usage/High Value (lower right quadrant):** These features are valuable to some of your user base, but most customers wouldn't want to pay for them. They're good candidates for extra-cost options or add-on products. Add-ons can help customize the fit of your packaging tiers to individual user needs.
- **Low Usage/Low Value (lower left quadrant):** These features won't add any value and won't be used. Don't waste time or development dollars here: Nix them.

Use this framework to focus packaging discussions among internal stakeholders. In meetings, adjust the position of the dots to reflect any new inputs.

How exact do you have to be in placing the dots? Aim for the most accurate representation you can, but don't get tangled in analysis paralysis. Use your collective judgment to make the best packaging decisions you can. Remember that the framework represents one moment in time. This exercise is not a one-shot deal. Your packaging should evolve as your customer mix, competitive landscape, and solution evolve.

03 Packaging Decision Framework



Tracking Results

Even after you've decided which features are basic, which to charge more for, and which to nix, you're not done. First, make sure you have developed ways to monitor feature usage within your application. Once that's done, keep the framework on hand as you monitor post-launch results.

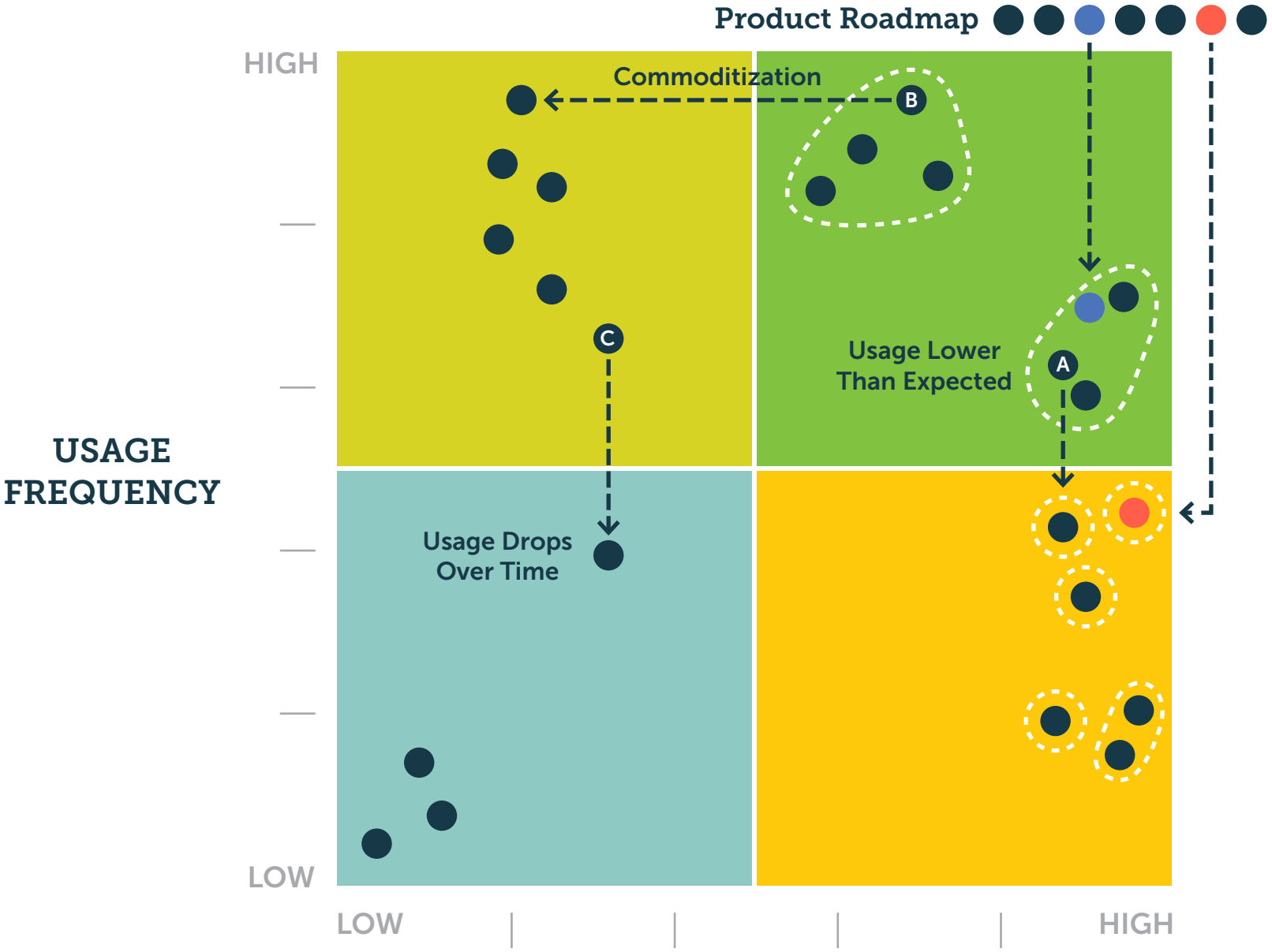
Let's say you placed Feature A in the upper right quadrant and packaged it into a premium tier, but it turns out usage is lower than you expected. The feature really belongs in the lower right quadrant, suggesting you may have missed an opportunity for an add-on product or capability and risk leaving potential revenue on the table.

Listen closely to salespeople: Are prospects asking for discounts because they won't use all the functionality in the premium tier? If so, moving the feature to an optional add-on may also reduce friction in the sales process.

Don't try to recapture missed revenue from existing customers when making these changes. Going forward, focus on capturing more revenue from new customers.

In the same way, you can use the framework to systematically map the effects of changes as they occur in your company's market, among your competitors, or in your customer mix. For example, if you placed Feature B in the upper right quadrant and it's now being offered by so many competitors that it has become a commodity, it no longer belongs there. If usage is dropping on a low-value feature (Feature C), you may want to stop investing in it.

Organizations that learn to use the framework in a systematic, disciplined way will be able to maximize revenue by speeding up the sales process—not just in current deals, but (as we'll explain later) in future revenue streams as well.



04 Packaging Mechanics

Customization Essentials

By now, you have a good idea of what features you need to offer and how to best package them for your user classes. But how do you efficiently develop and deliver customized tiers that include analytics?

Aim for balance.

You want to make the right combination of choices for each targeted class of users. Your aim is to deliver analytics that fits user needs and a user experience that fits how they work.

Don't skip these essentials:

As shown at the top level of the Customization Essentials graphic, you ideally want to be able to **customize any and all analytics elements**. This includes everything from dashboards, reports, and pages to the individual charts, table headings, rows, columns, and text they comprise.

The middle level shows the importance of being able to control these elements through both **browser-based display** (show/no show) and **server-based conditional logic** (if this, then load that).

The bottom level of the graphic is the foundation. You need to be able to **use your existing security model** to directly control access to analytics features and the data feeding them—as well as trigger both server- and browser-based customizations—based on existing user identities, roles, and permissions.

Learn how modern analytics platforms are leveraging adaptive security models >

CUSTOMIZATION ESSENTIALS

CONTENT CUSTOMIZATION

Anything can be tailored to user classes

- Dashboards • Widgets • Pages • Tables
- Rows • Columns • Menus • Lists • Dialog Boxes
- Tooltips • Toolboxes • Libraries • Etc.

APPLICATION CONTROLS

Multi-level controls

- Browser-level show/hide controls what's visible
- Server-level conditional logic controls what loads

SECURITY MODEL

Adaptive security

- Embedded analytics uses your product's existing security model
- Single sign-on
- Existing user identities, roles and permissions determine what users can see and do

Embedded Choices

As we've discussed, an effective software monetization strategy depends on creating tiered offerings that meet the analytic needs of different classes of users. That's the *what*.

The *how* is also important. How you choose to embed these analytic features affects user experience and therefore perception of value.

The three most important choices to make are:

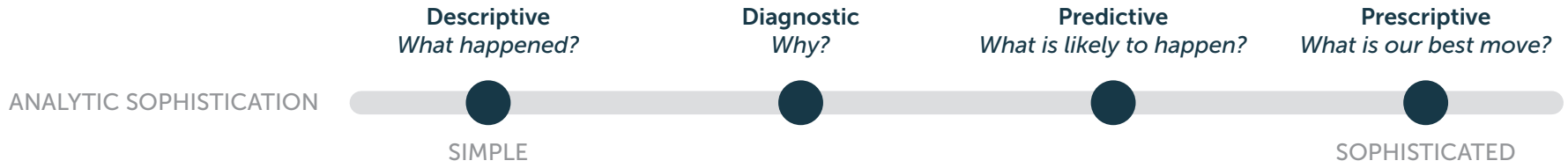
- **Range of User Flexibility.** How much can users do? For example, are they seeing a static set of charts or can they configure their own dashboards?
- **Depth of Integration.** How tightly do the analytics cooperate with the rest of your software? For example, are they in a separate tab or do they interact with other application features?
- **Degree of Customization.** How well does the UI reflect the user? For example, do elementary users have tooltips to help them understand new concepts? Do sophisticated users have toolboxes to quickly select the features they want?

The aim isn't always to be at the far-right side of each choice. More is not always better, nor is it always worth more to customers.

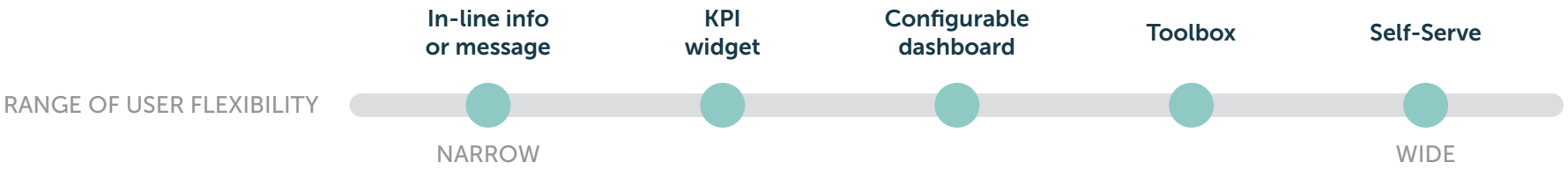
See some examples of this framework in action on the next page.

YOUR CHOICES AFFECT VALUE PERCEPTION

WHAT analytics features will you offer?



HOW will you embed them in your products?



05 Pricing for Value

Organizing Your Tiers

When you're ready to price your tiered product offering, Software Pricing Partners recommends using a pricing framework based on relative value.

Value is not based on feature count.

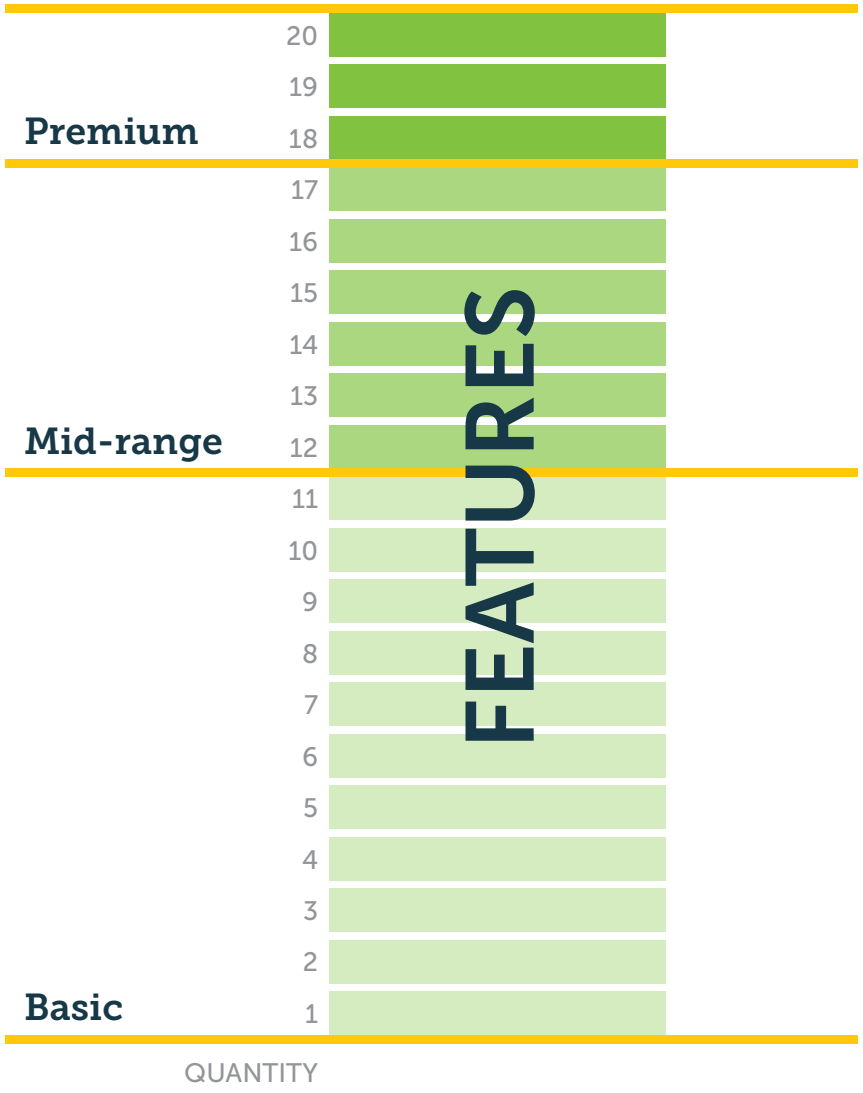
It doesn't necessarily follow that the mid-range tier represents 85 percent of your application's value (17/20 features = 85 percent). You need to take another step to estimate the percentage of total value the particular features within each tier are contributing.

There are many ways to compare relative value. One way, as shown in the Value Pricing Framework graphic, is to list the analytics features that make up your premium-tier product (the premium tier typically includes all of the features you offer).

Now, draw a horizontal line across your list to mark the cutoff points for the less feature-rich tiers. Let's say you have three tiers total, with 20 features in your premium tier, 17 in your mid-range tier, and 11 in your base tier.

The next two sections show how to determine prices.

VALUE PRICING FRAMEWORK



Determining the Relative Value of Each Tier

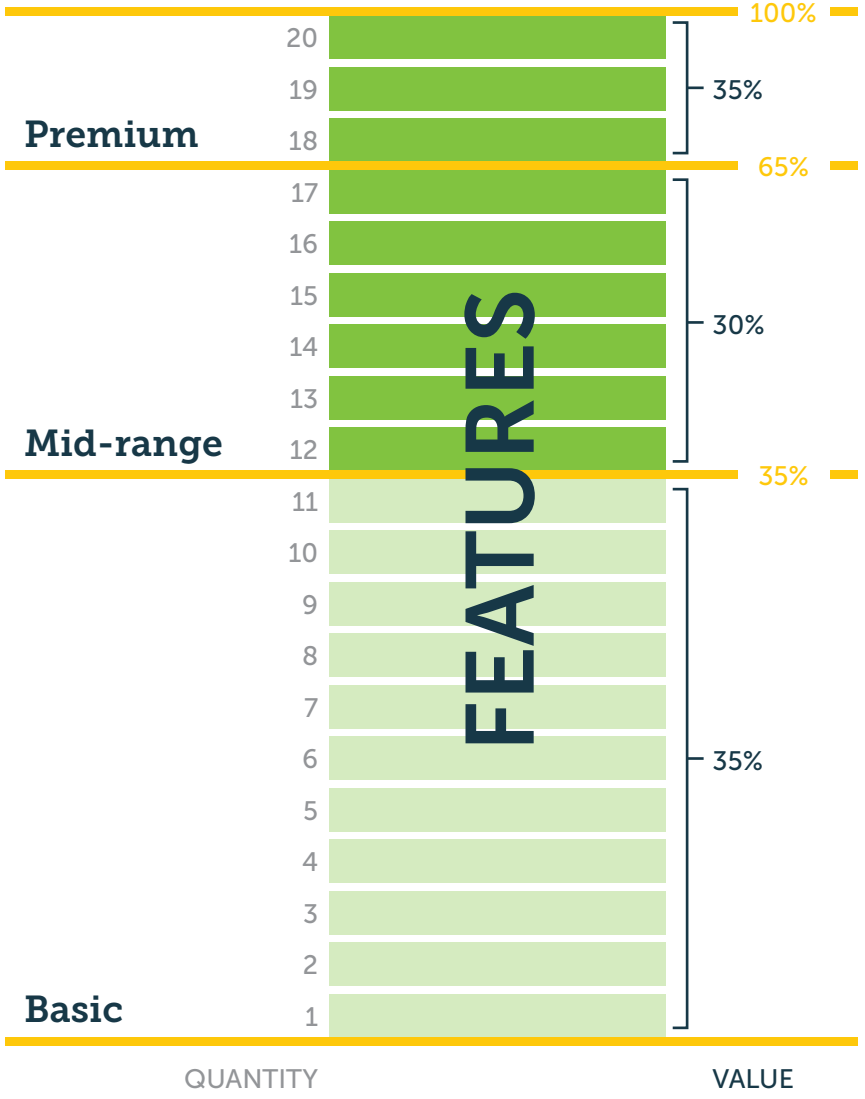
Assume that your premium level represents 100 percent of the analytics value available through your tiered offerings. Put that number at the top of the stack.

To estimate the relative value of your mid-range tier, review the feature positions in your Packaging Decision Framework, noting where on the value scale (x-axis) each feature falls compared to others. Estimate how much of the 100 percent of analytics value is being contributed by each added feature above the premium tier cutoff line.

Whatever your estimate is—35 percent in this example—subtract it from 100. What remains (65 percent) is the value of the mid-range and basic features tiers combined. Use a similar approach to estimate the percentage of value being contributed by the features listed in the mid-range (30 percent in this example). What remains is the value of the basic features tier (35 percent).

You now have a general framework that will help you assign prices based on relative value.

VALUE PRICING FRAMEWORK



Customer Economics

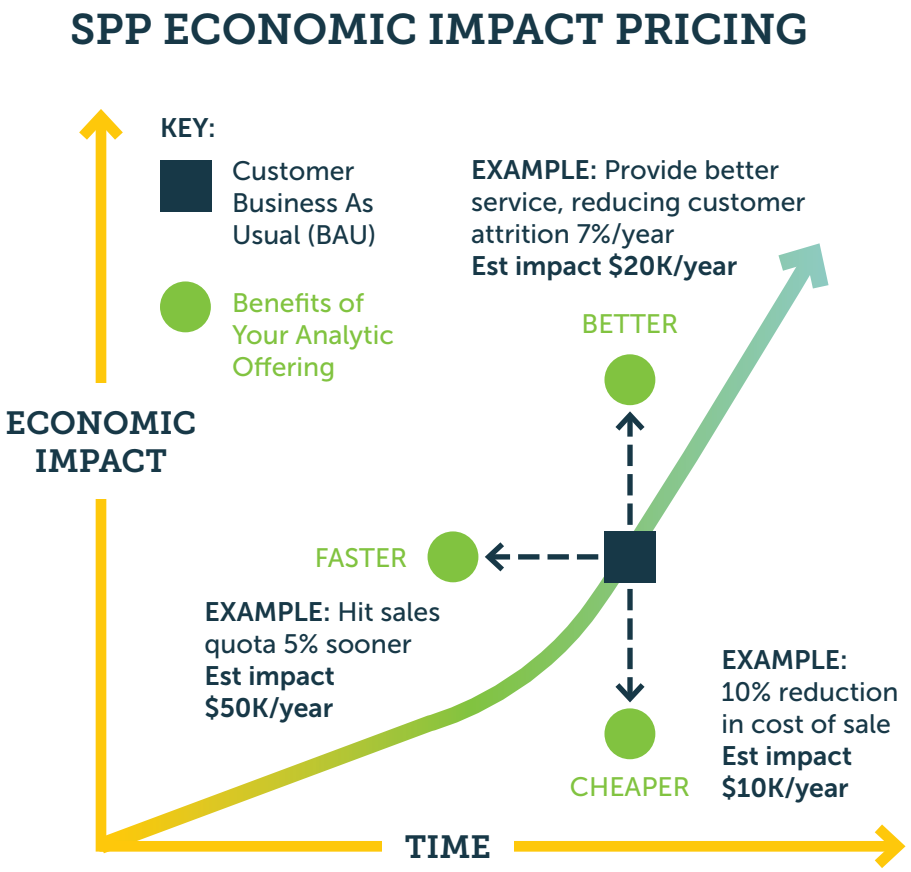
To figure out what the 100 percent price should be for your premium tier, you may need to experiment with several pricing techniques. Here, we'll take a quick look at two proven techniques used by Software Pricing Partners: economic impact analysis and competitive pricing analysis.

Economic Impact Analysis is a helpful pricing technique because it keeps the focus on value as perceived and experienced by customers.

Typically, software products enable customers to complete tasks better, faster, and/or cheaper. These capabilities boost revenue and bring cost savings.

Focusing on the revenue side (which is usually more compelling to customers), list the benefits of your premium-tier analytics offering and try to quantify the financial results. If you have them, case studies from customers are the ideal source. Use cases and pre-sales proposals can also be helpful.

Let's say you come up with $80,000 in economic value per year from your application's embedded analytics—as shown in our example here. Perceived risk, competition, effects of price on demand, and other market dynamics



will, of course, constrain you from setting a price equivalent to the full economic value you provide. Still, it's helpful to start thinking about how much economic value you might be able to capture in a proposed price, and it's a good test of reasonableness.

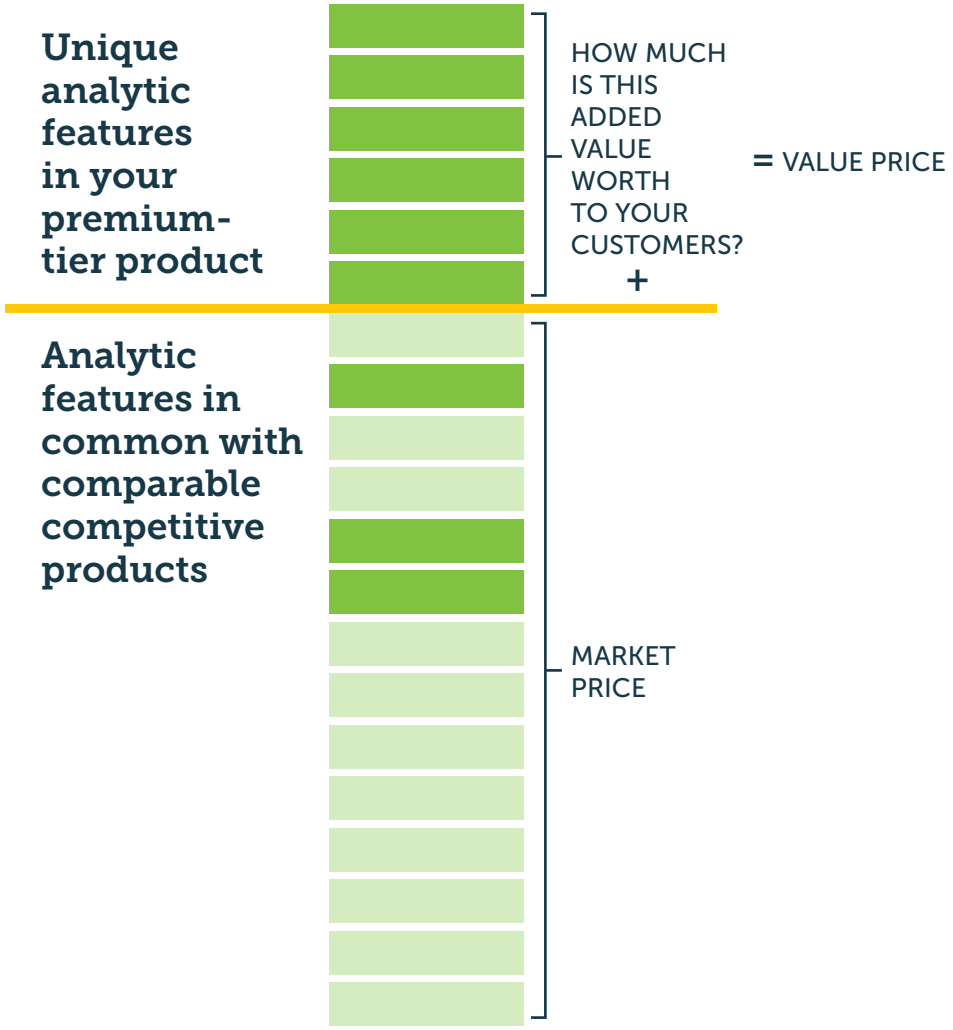
Competitive Pricing

Competitive pricing analysis all too often becomes an exercise in imitation. That's a problem because if the prices being imitated are leaving money on the table, the whole market ecosystem can end up doing the same.

To avoid copying a competitor's mistakes, Software Pricing Partners recommends taking a value-slanted approach. Here's how it works:

When you assess your analytics offerings versus the competition, you'll find comparable products have the same basic features and associated list price range in common. That's the estimated "market price," as perceived by your competitors. Remember that list prices can be deceiving since most customers pay a net price (net of various discounts).

SPP COMPETITIVE PRICING



This exercise can get you started, but it is best performed along with third-party intelligence gathering. Focus on talking with competitors' customers to determine net prices paid for software access fees, implementation, and professional services.

Presumably, your premium tier has unique value-adding features. So the question becomes: How much more than the market price can you charge for your unique value-add? How much is it worth to your customers?

To answer this question, list the analytics features in your premium tier that competitors don't offer. Look at where these features are positioned in your Packaging Decision Framework. How valuable do customers perceive them to be? Use this information to estimate your unique value-add.

Leveraging these and other pricing techniques, you should be able to create a set of overlapping price ranges for your premium tier to help you narrow down the possibilities and focus in on your best choices. Then, you can apply your relative value percentages (page 19) to assign prices to the rest of your tiers.

Your premium tier has unique value-adding features.

So the question becomes: How much more than the market price can you charge for your unique value-add? How much is it worth to your customers?

06 Get On the Embedded Analytics Revenue Escalator

Even the best embedded analytics may not warrant a higher price tag for every customer. As you decide what analytics features to embed and how to embed them, think about their effect on revenue potential at initial sale—and beyond.

For instance, embedding analytics that operate separately, but share a single sign-on with the primary application, can sometimes increase user perception of added value—justifying a premium charge. That's because the analytics piece is so clearly an addition. The drawback is that this type of integration creates a clunky user experience, which is detrimental to your top line in the long run.

Deeper integration—to the point where the analytics infuse or even seem to "disappear" into the primary application—may not make the added value as obvious to customers, but it will create a far more compelling user experience. The results may be stronger peer-recommended sales momentum, faster adoption, and more time spent in your application. You may also see higher customer satisfaction and stickiness as well as receptiveness to upgrade, up-sale, and cross-sale offers. All of this can justify a price premium.

This is where you ultimately want to go with embedded analytics: By delivering capabilities that fit your main user classes, you create a "revenue escalator." Users start out at different places on the escalator, but they're all heading upwards. They all have options for additional analytics features they can add over time to help them do their jobs even better. They all have a clear upgrade path to higher analytics tiers—and increased revenue for your company. **Learn more about monetizing analytics features >**

07 Conclusion: Lift Value & Revenue

The vast majority of software applications already include embedded analytics, and every year companies are adding more capabilities and integrating them more deeply into their products. But the value your customers perceive in these enhancements—and the incremental revenue you gain—depends on the decisions you make about packaging, pricing, and embedding.

These fundamentals of successful software monetization are too important to leave until late in your product development cycle—especially if you want to leverage the flexibility today's embedding platforms provide for creating tiered capabilities through easy customization.

Using the frameworks in this ebook, start thinking about how to package, price, and embed your analytics capabilities now. Then contact Logi Analytics or Software Pricing Partners, and we'll help you take the next step.

08 About Logi Analytics & Software Pricing Partners

Logi helps companies embed analytics into the fabric of their organizations and products. With Logi, companies can create genius analytic applications that are purpose-built to users' unique roles and skills, and delivered in the apps they already rely on—enabling anyone to analyze data when and where they need it.

More than 1,800 customers worldwide rely on Logi Analytics. The company is headquartered in McLean, Virginia. Learn more at LogiAnalytics.com.

Ready to Modernize Your Application's Embedded Analytics? Watch a Free Demo of Logi Analytics >

Software Pricing Partners (SPP) understands the pitfalls that improper pricing strategies can have on today's software companies. Their DNA dates back to the inception of the industry as a whole, with their leadership involved in the development and distribution of today's licensed software.

Over the past 35 years, SPP harnessed and expanded this insight to develop and refine a proven, proactive process. Unlike typical pricing approaches, SPP's process takes a holistic view on pricing, providing today's forward-thinking executives with the purpose-built pricing strategies they need to unlock their potential.

The company is headquartered in Charlotte, North Carolina. Learn more at SoftwarePricing.com