

A Forrester Total Economic
Impact™ Study
Commissioned By
Microsoft

Project Director:
Adam Schlegel
November 2016

The Total Economic Impact™ Of Xamarin For Visual Studio

Cost Savings And Business Benefits
Enabled By Xamarin For Visual Studio
Cross-Platform Mobile Application
Development

Table Of Contents

Executive Summary	3
Disclosures	5
TEI Framework And Methodology	6
Analysis	7
Financial Summary	23
Xamarin For Visual Studio: Overview.....	24
Appendix A: Total Economic Impact™ Overview.....	25
Appendix B: Glossary.....	26
Appendix C: Endnotes.....	27

ABOUT FORRESTER CONSULTING

Forrester Consulting provides independent and objective research-based consulting to help leaders succeed in their organizations. Ranging in scope from a short strategy session to custom projects, Forrester's Consulting services connect you directly with research analysts who apply expert insight to your specific business challenges. For more information, visit forrester.com/consulting.

© 2016, Forrester Research, Inc. All rights reserved. Unauthorized reproduction is strictly prohibited. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change. Forrester®, Technographics®, Forrester Wave, RoleView, TechRadar, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. For additional information, go to www.forrester.com.

Executive Summary

Microsoft commissioned Forrester Consulting to conduct a Total Economic Impact™ (TEI) study to examine the potential return on investment (ROI) enterprises may realize by building fully native mobile applications using Xamarin for Visual Studio. The purpose of this study is to provide readers with a framework to evaluate the potential financial impact of Xamarin for Visual Studio on their organizations, and the impact that rapid native, cross-platform mobile application development could have on their organization's ability to deliver a differentiated end user mobile experience that helps win, serve, and retain customers.

To better understand the benefits, costs, and risks associated with a Xamarin for Visual Studio implementation, Forrester interviewed several consumer- and business-facing enterprise and consulting customers with multiple years of experience building mobile applications on Xamarin for Visual Studio.

Xamarin for Visual Studio enables mobile developers to build mobile applications with full native fidelity for multiple mobile operating systems leveraging a shared C# code base and their existing developer resources and skillsets. Using Xamarin, mobile developers can rapidly bring mobile applications in their pipeline to market across multiple endpoint devices, reducing mobile application development time and cost; optimizing application life-cycle management efficiencies; and expediting ongoing bug fixes and updates to mobile application functionality, security, and data interfaces. For more details on Xamarin for Visual Studio, see the section titled: Xamarin for Visual Studio: Overview.

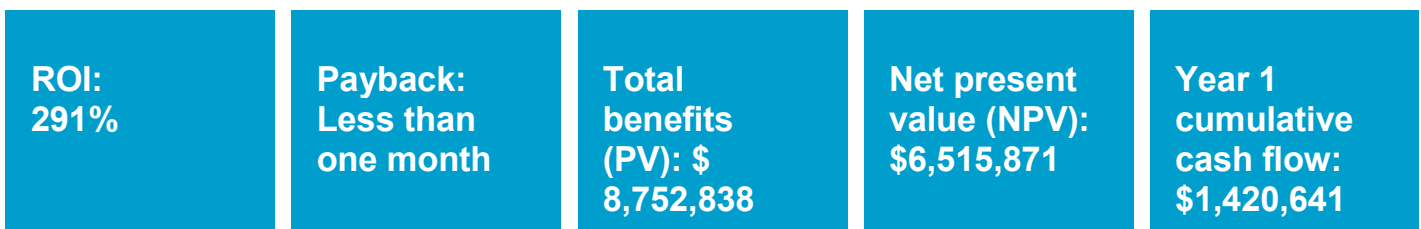
Prior to implementing Xamarin for Visual Studio to build the native applications in their mobile pipelines, interviewed customers were building native applications for each mobile operating system using indigenous, platform-specific languages, tools, and developer teams, resulting in high costs, application development delays, life-cycle management time and cost inefficiencies, and mobile development team siloes. Using Xamarin for Visual Studio, organizations were able to write less code; expedite the delivery of mobile applications in their pipelines; streamline application life-cycle management activities; and lower the organization's reliance on siloed, platform-specific development teams and outsourced programming talent. More importantly, Xamarin for Visual Studio enabled organizations to deliver a fully native user interface (UI) and native application performance levels to their customers and employees.

XAMARIN FOR VISUAL STUDIO REDUCES MOBILE APPLICATION DEVELOPMENT AND LIFE-CYCLE MANAGEMENT COSTS, WHILE OPTIMIZING THE UTILIZATION OF EXISTING DEVELOPER RESOURCES

Our interviews with four existing customers and subsequent financial analysis found that a composite organization based on these interviewed organizations experienced the risk-adjusted ROI and benefits shown in Figure 1.

FIGURE 1

Financial Summary Showing Three-Year Risk-Adjusted Results



Source: Forrester Research, Inc.

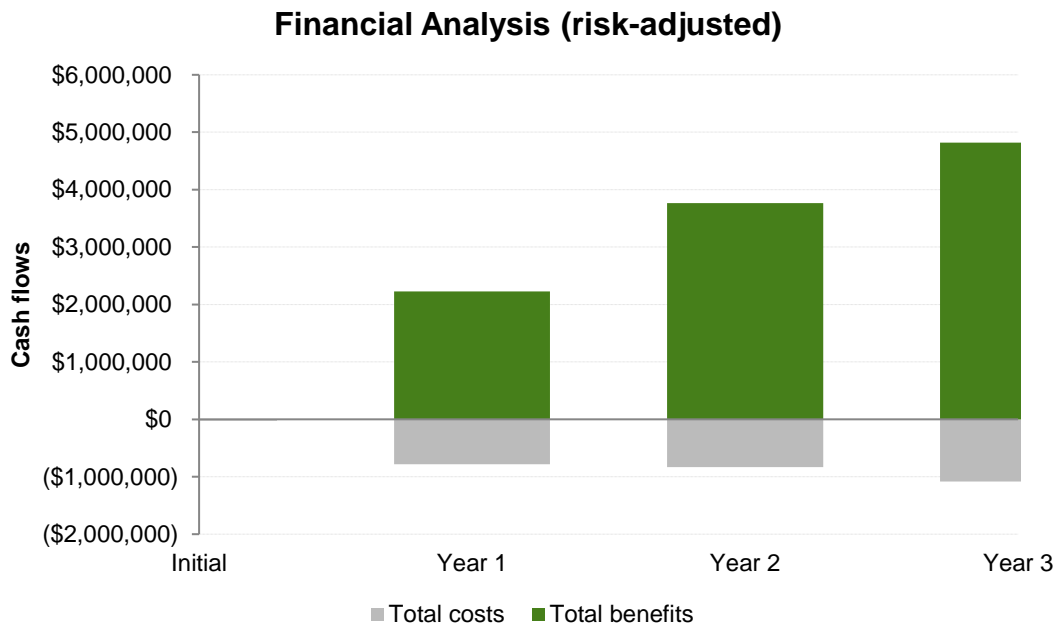
Xamarin for Visual Studio helps mobile developers rapidly and cost effectively build and port fully native mobile applications across iOS, Android, and Windows devices using a single programming language and a shared C# code base.

Over a three-year period, an organization with multiple mobile applications can expect to:

- **Reduce mobile application development costs by \$1,365,003.**
- **Accrue mobile application maintenance and upgrade efficiencies of \$829,475.**
- **Avoid \$6,558,360 in platform-specific mobile application developer expenses.**

FIGURE 2

Financial Summary Showing Three-Year Risk-Adjusted Results



Source: Forrester Research, Inc.

By offsetting total three-year risk- and present value-adjusted costs of \$2,236,967 with labor cost savings and efficiency gains of \$8,752,838, the composite organization's investment in the Xamarin for Visual Studio solution delivered a net present value of \$6,515,871.

› **Benefits.** The composite organization experienced the following risk- and present value-adjusted benefits, totaling \$8,752,838 over the three-year forecast period:

- **Reduction in mobile application development costs of \$1,365,003 due to code sharing efficiencies.** By sharing 70% of its C# code base across mobile platforms after building its initial iOS application with Xamarin for Visual Studio, the composite organization was able to write less code and significantly reduce its research and development (R&D) and application development expenses associated with porting its mobile applications to Android and the Universal Windows Platform (UWP).
- **Mobile application maintenance and upgrade efficiency gains of \$829,475.** Through the use of a shared code base in the business logic layer of its iOS, Android, and Windows Phone mobile applications, the composite organization no longer needed to fix, manage, and update several separate code bases using multiple developer skillsets and platform-specific tools. As such, it was able to accrue application maintenance life-cycle efficiencies related to providing ongoing updates and augmentations to the functionality, security, and data interfaces of the mobile applications in its portfolio.
- **Cost avoidance of \$6,558,360 by mitigating the need for platform-specific mobile application developer talent.** By unifying its mobile development team on Xamarin for Visual Studio and using a single programming language, the composite organization was able to optimize use of its existing C# developer talent, mitigating the need for multiple, siloed mobile development teams organized by indigenous, pure native programming language.

› **Costs.** The composite organization experienced the following risk- and present value-adjusted costs, totaling \$2,236,967 over the three-year forecast:

- **Visual Studio Enterprise annual software license and Xamarin University subscription fees of \$67,320.** These are ongoing, annual software subscription fees for an enterprise-level subscription to Visual Studio Enterprise, which includes Xamarin for Visual Studio. In addition, this cost category includes one-time Xamarin University training seats for each new Xamarin mobile developer.
- **Xamarin for Visual Studio training, implementation, staffing, and skill acquisition of \$2,099,193.** This cost includes the allocated salary expenses of the organization's existing C# web developers, used in lieu of platform-specific mobile developers, who were repurposed to mobile application development projects. In addition, this cost category includes one-time IT labor costs required to install and deploy Xamarin for Visual Studio, as well as costs to train and certify each new developer in building iOS, Android, and UWP applications using Xamarin for Visual Studio. Lastly, this cost category includes the expense of bringing outsourced, full-time Xamarin for Visual Studio developer talent into the organization during years 1 and 2 to provide staff augmentation services.
- **Microsoft partner professional service expense of \$70,455.** These are one-time costs incurred to bring in experienced senior engineers with a high degree of expertise in Xamarin for Visual Studio to expedite the development of the composite organization's first few mobile applications and provide incremental mobile developer skill and capability development support.

Disclosures

The reader should be aware of the following:

- › The study is commissioned by Microsoft and delivered by Forrester Consulting. It is not meant to be used as a competitive analysis.
- › Forrester makes no assumptions as to the potential ROI that other organizations will receive. Forrester strongly advises that readers use their own estimates within the framework provided in the report to determine the appropriateness of an investment in Xamarin for Visual Studio.
- › Microsoft reviewed and provided feedback to Forrester, but Forrester maintains editorial control over the study and its findings and does not accept changes to the study that contradict Forrester's findings or obscure the meaning of the study.
- › Microsoft provided the customer names for the interviews but did not participate in the interviews.

TEI Framework And Methodology

INTRODUCTION

From the information provided in the interviews, Forrester has constructed a Total Economic Impact (TEI) framework for those organizations considering implementing Xamarin for Visual Studio. The objective of the framework is to identify the cost, benefit, flexibility, and risk factors that affect the investment decision, to help organizations understand how to take advantage of specific benefits, reduce costs, and improve their ability to win, serve, and retain customers.

APPROACH AND METHODOLOGY

Forrester employed four fundamental elements of TEI in modeling Xamarin for Visual Studio: benefits, costs, flexibility, and risks.

Forrester took a multistep approach to evaluate the impact that Xamarin for Visual Studio can have on an organization (see Figure 2). Specifically, we:

- › Interviewed Microsoft marketing, sales, product development, and developer relations personnel, along with Forrester analysts, to gather data relative to the platform and the marketplace for mobile application development solutions.
- › Interviewed four organizations currently using Xamarin for Visual Studio to build and port mobile applications in order to obtain data with respect to costs, benefits, and risks.
- › Designed a composite organization based on characteristics of the interviewed organizations.
- › Constructed a financial model representative of the interviews using the TEI methodology. The financial model is populated with the cost and benefit data obtained from the interviews as applied to the composite organization.
- › Risk-adjusted the financial model based on issues and concerns the interviewed organizations highlighted in interviews. Risk adjustment is a key part of the TEI methodology. While interviewed organizations provided cost and benefit estimates, some categories included a broad range of responses or had a number of outside forces that might have affected the results. For that reason, some cost and benefit totals have been risk-adjusted and are detailed in each relevant section.

Given the increasing sophistication that enterprises have regarding ROI analyses related to IT investments, Forrester's TEI methodology serves to provide a complete picture of the total economic impact of purchase decisions. Please see Appendix A for additional information on the TEI methodology.

FIGURE 3

TEI Approach



Source: Forrester Research, Inc.

Analysis

INTERVIEWED ORGANIZATIONS

Forrester interviewed the following four customers for this case study, each of which had built and ported fully native mobile applications using Xamarin for Visual Studio:

- › **A global entertainment company with over 500 properties located across 16 countries, with annual revenue in excess of \$2.5 billion.** In order to meet the evolving and increasingly sophisticated needs of its consumers, the company needed to deliver rich mobile experiences for its patrons across mobile endpoint devices. After outsourcing the development of a native, platform-specific iOS mobile application, the company struggled with the tradeoff between the cost of building additional native mobile applications using an indigenous, pure native approach and delivering inferior mobile experiences using hybrid-built business-to-consumer (B2C) mobile applications.
- › **A US-based strategy and technology consulting company with 500 consultants providing mobile strategy and application development for enterprise customers across a myriad of industry verticals.** The company has experience with all mobile development approaches including mobile web, hybrid, cross-platform tools, and pure native, to guide their clients in selecting the appropriate approach considering their current use case, future direction, and in-house skillsets. Back in 2010 and 2011, the company's engagements were more focused on mobile web and hybrid, as enterprise applications were not seen to have the same user experience (UX) and UI requirements as consumer-facing mobile applications. Over the years, bring-your-own-device (BYOD) policies and increased mobile user sophistication have grown demand for the company to build and deploy enterprise mobile applications that have the same high-end, native performance levels and UX as consumer-facing mobile applications.
- › **A global mobile application development firm with 35 mobile developers leveraging a combination of technologies, including pure native, hybrid, and cross-platform mobile development tools, to solve enterprise business challenges.** Historically, the company primarily built native iOS and Android mobile applications using a platform-specific, pure native approach, while occasionally leveraging hybrid mobile development tools when faced with organizational budgetary constraints. While hybrid mobile application development platforms offered some time and cost savings, the firm found these tools weren't conducive to effective application delivery. While 65% to 70% of the company's mobile development requests for proposals (RFPs) were for platform-specific, native-built mobile applications in 2015, it needed to meet growing market demand to build native mobile applications for multiple endpoint devices in a cost-effective way. As such, in 2016, 50% of its business to date has been for cross-platform-built mobile applications.
- › **A global portfolio of casual dining restaurant chains with over 1,000 restaurants, 100,000 employees, and in excess of \$4 billion in annual revenue.** Following poor app store reviews for its third-party-built mobile application, the company analyzed its customers' digital journey and mobile customer experience. Through this process, the company identified the need to improve its customers' experience by providing them with a mobile toolset that reduced the amount of time required to be seated in-restaurant and pay their bill after finishing their meal. In order to meet the increasingly

“We came to the realization that we needed some kind of cross-platform tool, since cost is always a factor in any kind of mobile pursuit. So if we can provide savings through technology, that’s a win for everybody and it makes us much more competitive.”

~ Practice lead, mobile development, global mobile development firm

sophisticated mobile needs of its clientele, the company needed to find a way to build fully native mobile applications that met its customers' needs, on any device and in their time of need, while leveraging its existing .NET and C# capabilities.

COMPOSITE ORGANIZATION

Based on these interviews with current Xamarin for Visual Studio customers, Forrester constructed a TEI framework, a composite company, and an associated ROI analysis that illustrates the areas financially affected. The composite organization that Forrester synthesized from these results represents an organization with the following characteristics:

- › The US-based consumer products retailer has 15,000 employees working throughout 800 stores. These stores are located across its international footprint spanning North American, Latin America, Europe, and Asia Pacific.
- › It has annual revenue in excess of \$3.5 billion.
- › The company owns and operates three distinct brands in the US and five additional unique brands internationally. Each brand has its own eCommerce and mobile program.
- › The organization's marketing and sales functions are distributed to each of the organization's domestic and international brands.
- › The organization has five horizontal shared services divisions, including human resources, finance, information technology, manufacturing, and corporate services. Furthermore, the organization's IT group has strong existing .NET web application development and C# programming capabilities, developed through the company's mature eCommerce and web programs.
- › As part of the organization's strategic mobile road map, each of the company's three domestic and five international brands has its own B2C mobile application built for iOS and Android. Each of the company's B2C mobile applications delivers the following mobile customer experiences and functionality:
 - A GPS store locator and a store-specific way of finding tools to assist customers in locating departments and products of interest.
 - A product discovery and in-store product availability tool.
 - A barcode-enabled product detail discovery.
 - Coupons and promotions for daily and weekly deals, including opt-in push notifications.
 - A loyalty program that tracks and illustrates transaction history and reward point balance.
- › In addition to the company's B2C mobile applications, each of the company's three domestic brands has developed additional business-to-employee (B2E) mobile applications to support employees with sales and inventory management activities.
- › In order to promote better employee productivity and reduce IT spend, the organization has a BYOD policy. As such, the organization needs to support its B2E applications on endpoint devices running iOS, Android, and UWP.

“A lot of the knowledge that we have embedded within our developers and within our team is all around Microsoft technologies, and that’s why adopting Xamarin made absolute sense to us.”

~ VP of application development, global entertainment company

Note that organizations can input their own estimates and assumptions using the TEI framework and the study's economic analysis to estimate the potential financial impact of a Xamarin for Visual Studio deployment in their own environment.

INTERVIEW HIGHLIGHTS

Technology-empowered customers are driving organizations to deliver differentiated mobile experiences that provide customers, partners, and employees with what they want, on any device, in their time of need.¹ In the past, enterprise mobile applications were perceived to require less UI and UX componentry, leading many organizations to build mobile web experiences that resulted in poorly rated and underutilized mobile tools. The proliferation of BYOD policies and the increased sophistication and UX associated with consumer-facing mobile applications have raised the bar in terms of mobile application performance, responsiveness, and user experience requirements across all consumer-, business-, and enterprise-facing mobile applications. By building both their customer- and employee-facing mobile applications on Xamarin for Visual Studio's cross-platform mobile application development tool, businesses are able to leverage a single programming language and a shared C# code base to rapidly and cost effectively deliver native mobile experiences across iOS, Android, and Windows devices.

“In the past, enterprise apps weren’t thought to need the same UX and UI component as consumer apps. Today, with all of the BYOD policies and with people getting used to that great native experience they’re getting from consumer apps, we’re starting to see a trend where even enterprise apps require that high-end native performance, native responsiveness, and native user experience.”

~ Principal, North American strategy and consulting firm

One of the interviewed organizations indicated that it was able to share and reuse 75% of the C# code base from its first iOS mobile application when porting that mobile application to the Android platform, significantly reducing mobile application development costs and improving time-to-value. In addition to reducing the cost to port its mobile applications to additional mobile platforms in a native way, interviewed organizations indicated that they invested in Xamarin for Visual Studio to:

- › Leverage and repurpose their existing .NET application development and C# programming skills in order to build and scale their mobile solutions portfolios.
- › Eliminate and consolidate their siloed Objective-C, Java, and UWP developer teams into a single, collaborative mobile development team using a single programming language for mobile application development.
- › Improve their customer experience and satisfaction levels by delivering a consistent and robust mobile experience with full native fidelity across mobile endpoint devices.

The interview revealed that:

- › **Cost is always a factor in any mobile pursuit, and organizations need a cross-platform mobile development tool that provides cost savings through technology.** Interviewed organizations struggled with the tradeoff decision between two options: 1) incurring the high cost of porting mobile applications in their portfolio to additional mobile platforms using a pure native, platform-specific approach using siloed developer teams with Objective-C and Java programming skillsets and 2) delivering inferior mobile experiences through hybrid-built B2C mobile applications. By adopting Xamarin for Visual

Studio, interviewed organizations indicated that they were able to circumvent this challenge, leveraging their existing .NET development skillsets to deliver fully native mobile applications for iOS, Android, and Windows devices.

- › **Xamarin for Visual Studio enables organizations to rapidly identify and resolve issues and bugs with their mobile applications and quickly incorporate new features across the mobile application life cycle.** Through the use of a shared C# code base in the business logic layer of its mobile applications, interviewed organizations eliminated the need to fix, manage, and update several separate code bases. As a result, interviewees were able to significantly reduce the time and resource requirements needed to provide ongoing mobile application life-cycle updates and fixes, while improving the cadence of their development process.
- › **Xamarin empowers mobile developers to transform their customers' mobile experience, improving app store performance and app engagement scores.** Several interviewees indicated that Xamarin for Visual Studio served as a catalyst for a paradigm shift in their mobile offering. Before adopting Xamarin for Visual Studio to develop and port their mobile applications, interviewees leveraged a number of platform-specific and hybrid mobile development toolsets that resulted in poor application performance and low app store ratings. After building and porting their native mobile applications using Xamarin for Visual Studio, interviewed organizations indicated that they saw significant improvements in app store performance, download count, and mobile application engagement metrics. One interviewed organization saw its average app store review grow from 2.5 stars to 4.5 stars after rebuilding its mobile app with Xamarin for Visual Studio, making it one of the highest rated apps in its sector.

BENEFITS

The composite organization experienced a number of quantified benefits in this case study:

- › Reduction in mobile application development costs.
- › Mobile application maintenance and upgrade efficiency gains.
- › Cost avoidance of platform-specific mobile application developer talent.

Another important benefit mentioned by several interviewed customers was an increase in mobile application performance and customer satisfaction scores. Past mobile development solutions presented the difficult tradeoff decision between native mobile application performance and the ability to serve a broader customer base across multiple mobile platforms. Prior to adopting Xamarin for Visual Studio, interviewed organizations attempted to deliver mobile offerings to the widest set of customers by building primitive, poorly rated mobile application experiences, resulting in low app store ratings, low downloads counts, and poor time in application and daily active user performance levels. By leveraging Xamarin for Visual Studio to build and port their mobile applications, these organizations saw significant improvements in their app store download count and time in app engagement performance.

Since interviewed customers did not have sufficient data to explicitly quantify this benefit, it has not been included in the ROI calculation for this study. Nonetheless, this is a benefit that potential adopters of Xamarin for Visual Studio may very well experience.



Reduction In Mobile Application Development Costs

The composite organization identified the ability to significantly reduce the time and cost associated with porting customer- and employee-facing iOS mobile applications across additional mobile operating systems and device types with full native fidelity as a key benefit of Xamarin for Visual Studio. Prior to using Xamarin for Visual Studio to build and port its mobile applications, the composite organization built and maintained two or more separate code bases, platform-specific toolsets, and pure native development teams with Objective-C and Java skillsets, making it extremely time consuming, resource intensive, and expensive to build the mobile applications in its pipeline and meet the increasingly sophisticated mobile needs of its clientele. Due to the high cost associated with maintaining multiple platform-specific code bases and developer teams, the organization experimented with hybrid mobile development tools, but this ultimately resulted in poor mobile experiences and low app store performance.

The organization's mobile road map tasked the organization with porting each iOS B2C application to the Android platform and each B2E iOS application to Android and UWP. In addition, management expected the organization's mobile development team to deliver strong, consistent, and fully native mobile experiences across iOS, Android, and, in the case of B2E applications, Windows-based devices. Given the organization's constrained resource base, it consistently struggled with the

“We had a lot of problems with native development, since we are a Microsoft shop and we just didn’t have Objective-C and Java skillsets internally. We decided that we really needed to get away from pure native platforms and find something that will allow us to be both on iOS and Android.”

~ Mobile solutions architect, global mobile application development firm

tradeoff between time, performance, and cost. In order to meet the organization's ambitious mobile goals within its budgetary constraints, it needed a way to deliver savings and efficiencies through technology.

Following the adoption of Xamarin for Visual Studio, the composite organization was able to share 70% of its C# code base across mobile platforms after deploying its initial Xamarin-built iOS mobile application, enabling it to write less code and significantly reduce its R&D and application development expenses. Using Xamarin for Visual Studio, the composite organization was able to reduce the time to port each of its mobile applications to an additional mobile platform to six weeks, compared with 20 weeks using a pure native, platform-specific approach. Since the composite organization ported a total of eight B2C mobile applications to the Android platform over the three-year forecast period, and ported a total of three B2E mobile applications to the Android and UWP in years 2 and 3 of the analysis, it was able to reduce the total time to port its mobile applications to additional mobile platforms by a total of 196 weeks over the three-year forecast period. At an average cost to build a fully native mobile application of \$192,000, the composite organization was able to save a total of \$1,881,600 over the three-year forecast period before adjusting for risk.

Interviewed organizations provided a broad range of mobile code sharing levels across multiple mobile platforms, depending on the unique nature of their mobile needs, usage of the Xamarin.Forms library to allow code sharing at the UI level, and their mobile maturity levels. To compensate for these factors, this benefit was risk-adjusted (reduced) by 10% in Table 1.

TABLE 1

Reduction In Mobile Application Development Costs

Ref.	Metric	Calculation	Year 1	Year 2	Year 3
A1	Number of B2C iOS mobile applications developed		2	4	2
A2	Number of B2E iOS mobile applications developed			1	2
A3	Number of additional mobile platforms targeted with B2C apps	Android only	1	1	1
A4	Number of additional mobile platforms targeted with B2E apps	Android and Windows Phone	2	2	2
A5	Weeks to port each mobile application to an additional mobile platform using a pure native approach		20	20	20
A6	Shared C# code base with Xamarin for Visual Studio		70%	70%	70%
A7	Weeks to port mobile application to an additional mobile platform on Xamarin for Visual Studio	$A5*(1-.7)$	6	6	6
A8	Cost to build fully native mobile application on platform-specific tool	4 developers * \$80 per hour * 40 hours per week * 12 weeks	\$192,000	\$192,000	\$192,000
At	Reduction in mobile application development costs through the use of shared C# code base across mobile platforms	$((A1*A3)+(A2*A4))*(A8*A6)$	\$268,800	\$806,400	\$806,400
	Risk adjustment	↓10%			
Atr	Reduction in mobile application development costs through the use of shared C# code base across mobile platforms (risk-adjusted)		\$241,920	\$725,760	\$725,760

Source: Forrester Research, Inc.



Mobile Application Maintenance And Upgrade Efficiency Gains

The advantages of a shared C# code base extend beyond the upfront development and porting of mobile applications from iOS to additional mobile platforms into the mobile application life cycle. Prior to adopting Xamarin for Visual Studio to build and port the mobile applications in its pipeline, the composite organization had to perform ongoing updates to mobile application features, functionality, and business logic at the individual mobile platform level, consuming significant developer resources. In addition, the labor intensity associated with fixing and updating multiple code bases created large backlogs for user-reported bug fixes and mobile application updates and releases on its mobile road map.

Following the implementation of Xamarin for Visual Studio, the composite organization was able to reduce the time spent fixing user-reported bugs and providing ongoing updates to the functionality, security, and data interfaces of each ported application by 50% after making the initial iOS fixes and updates. At a fully loaded average hourly cost of \$80 and a four-person developer team assigned to the ongoing maintenance of each B2C and B2E mobile application in its portfolio, the composite organization was able to reduce its annual per-application maintenance and upgrade labor costs by a total of \$92,160 in Year 1; \$368,640 in Year 2; and \$645,120 in Year 3, before adjusting for risk and present value. Notably, these time savings also enabled the composite organization to redirect its developer resources toward improving the mobile experience across its portfolio, improving the customer experience and customer satisfaction with its brands.

The average fully loaded salary of mobile developer talent will vary by region, skillset, and tenure with the company, and the nature of each mobile bug fix and feature enhancement project will vary significantly across mobile offerings. To compensate, this benefit was risk-adjusted and reduced by 5% in Table 2. See the section on Risks for more detail.

“Prior to Xamarin for Visual Studio, we got bugs that unfortunately would last for over a month. Xamarin has provided us with mobile application life-cycle efficiencies that have allowed us to release a lot more rapidly than we have in the past.”

~ Mobile solutions architect, global mobile application development firm

“Since Xamarin for Visual Studio is basically .NET, we can more easily share code not only within an app, but we can also share it with other apps. So if we’re developing a suite of apps for a company, then we get that additional lift of being able to share it across the apps.”

~ Principal, North American strategy and consulting firm

TABLE 2

Mobile Application Maintenance And Upgrade Efficiency Gains

Ref.	Metric	Calculation	Year 1	Year 2	Year 3
B1	Reduction in B2C and B2E mobile application code bases maintained using Xamarin for Visual Studio	# of total mobile apps* total # of additional mobile platforms	2	8	14
B2	Number of developers per mobile application update		4	4	4
B3	Annual average number of hours spent troubleshooting, fixing bugs, and updating mobile applications, per app	Includes an average of 4 bug fixes and other improvements to application functionality	288	288	288
B4	Reduction in time spent updating functionality on each additional platform using Xamarin for Visual Studio shared code base		50%	50%	50%
B5	Number of mobile application maintenance and upgrade hours saved using Xamarin for Visual Studio, per platform	$B3*B4$	144	144	144
B6	Average hourly cost of a mobile application developer		\$80	\$80	\$80
Bt	Mobile application maintenance and upgrade efficiency gains using Xamarin for Visual Studio shared code base	$B1*B2*B5*B6$	\$92,160	\$368,640	\$645,120
	Risk adjustment	↓5%			
Btr	Mobile application maintenance and upgrade efficiency gains using Xamarin for Visual Studio shared code base (risk-adjusted)		\$87,552	\$350,208	\$612,864

Source: Forrester Research, Inc.



Cost Avoidance Of Platform-Specific Mobile Application Development Talent

The composite organization indicated that a key benefit of Xamarin for Visual Studio was the ability to better utilize its existing developer resources and avoid the need to build siloed Objective-C, Java, and UWP developer teams, instead using a single, collaborative Xamarin for Visual Studio developer team for mobile application development. As such, the company wanted to leverage its existing .NET web application development and C# programming capabilities in building out its mobile practice.

Over the three-year forecast period, the organization needed to port eight of its Xamarin for Visual Studio-built iOS B2C mobile applications to the Android platform and three of its Xamarin for Visual Studio-built B2E mobile applications to both Android and UWP. By leveraging Xamarin for Visual Studio to build its mobile applications, the company avoided the cost of additional Objective-C, Java, and UWP developer talent that would be required to build the fully native mobile applications in the company's pipeline using a pure native approach (see Figure 4).

If the organization had chosen a platform-specific, pure native mobile application development approach, it would have required six additional Objective-C and Java developers to build and port the B2C mobile applications in its Year 1 pipeline. As its mobile environment scaled to include incremental B2C and B2E mobile applications

across its global brands, the organization was able to avoid hiring 17 platform-specific mobile developers in Year 2 and 22 platform-specific mobile developers in Year 3.

Mobile projects vary significantly in the degree of UI complexity and the use of complementary time- and cost-saving Xamarin tools, including Xamarin.Forms. Additionally, the number of mobile developers avoided by an organization will depend on its existing developer skillsets. In order to account for these factors, along with variance in mobile application developer salaries, this benefit was risk-adjusted and reduced by 5% in Table 3.

TABLE 3

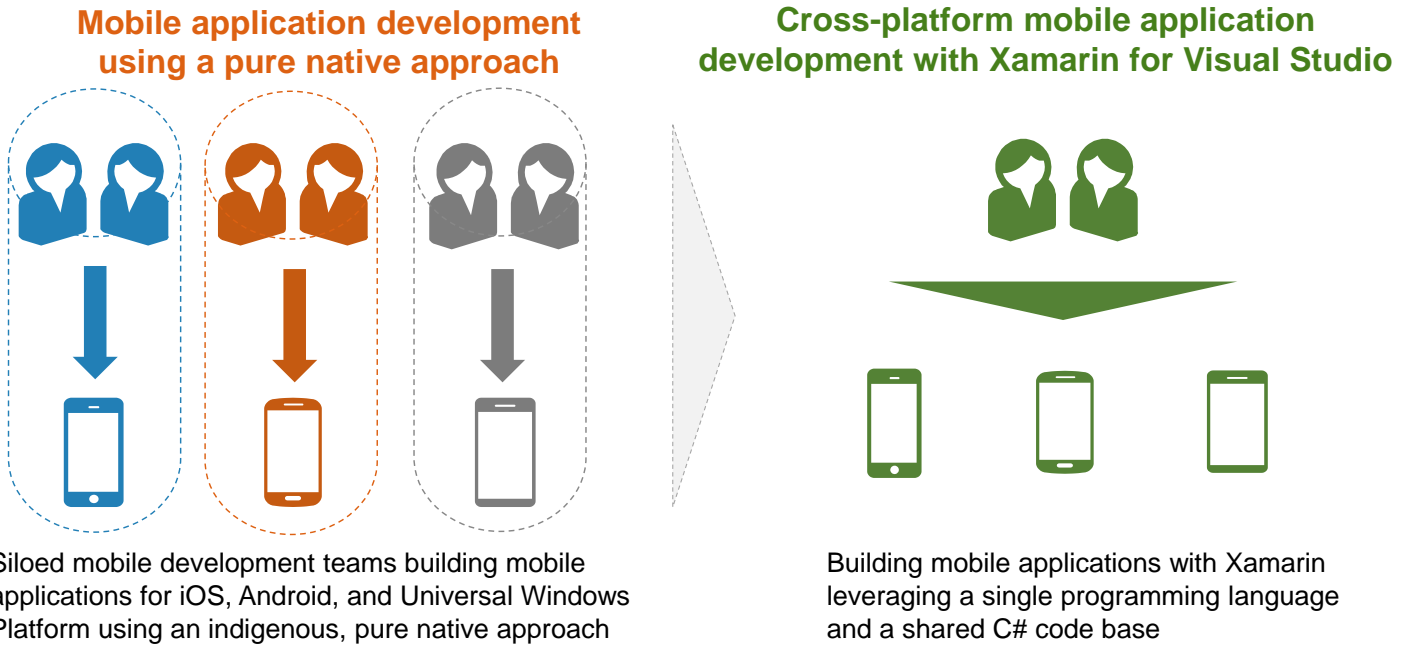
Cost Avoidance of Platform-Specific Mobile Application Development Talent

Ref.	Metric	Calculation	Year 1	Year 2	Year 3
C1	Number of Java mobile developers avoided (full-time equivalents [FTEs])		6	8	10
C2	Number of Objective-C mobile developers avoided (FTEs)		6	8	10
C3	Number of Universal Windows Platform developers avoided (FTEs)			1	2
C4	Total number of platform-specific developers avoided, per year		12	17	22
C5	Average fully loaded annual cost per developer		166,400	166,400	166,400
Ct	Cost avoidance of additional platform-specific mobile application developer talent	C4*C5	\$1,996,800	\$2,828,800	\$3,660,800
	Risk adjustment	↓5%			
Ctr	Cost avoidance of additional platform-specific mobile application developer talent (risk-adjusted)		\$1,896,960	\$2,687,360	\$3,477,760

Source: Forrester Research, Inc.

FIGURE 4

Mitigate The Need For Multiple, Siloed, Pure Native Development Teams



Source: Forrester Research, Inc.

Total Benefits

Table 4 shows the total of all benefits across the three areas listed above, as well as present values (PVs) discounted at 10%. Over three years, the composite organization expects risk-adjusted total benefits to be a PV of \$8,752,838.

TABLE 4

Total Benefits (Risk-Adjusted)

Ref.	Benefit Category	Year 1	Year 2	Year 3	Total	Present Value
Atr	Reduction in mobile application development costs through the use of shared C# code base across mobile platforms	\$241,920	\$725,760	\$725,760	\$1,693,440	\$1,365,003
Btr	Mobile application maintenance and upgrade efficiency gains using Xamarin for Visual Studio's shared code base	\$87,552	\$350,208	\$612,864	\$1,050,624	\$829,475
Ctr	Cost avoidance of additional platform-specific mobile application developer talent	\$1,896,960	\$2,687,360	\$3,477,760	\$8,062,080	\$6,558,360
	Total benefits (risk-adjusted)	\$2,226,432	\$3,763,328	\$4,816,384	\$10,806,144	\$8,752,838

Source: Forrester Research, Inc.

COSTS

In order to adopt Xamarin for Visual Studio, the composite organization made the following investments:

- › Visual Studio Enterprise annual software license and Xamarin University subscription fees.
- › Xamarin for Visual Studio training, implementation, staffing, and skill acquisition costs.
- › Microsoft partner professional services.



Visual Studio Enterprise Annual Software License And Xamarin University Subscription Fees

The composite organization paid annual software subscription fees for an enterprise-level Visual Studio subscription, which includes Xamarin for Visual Studio, for the internal and outsourced developers in its mobile practice. In addition, the organization incurred a one-time expense for Xamarin University seats to train and certify each mobile developer in C# cross-platform mobile development using Xamarin for Visual Studio. The organization sent six mobile developers through Xamarin University and procured six enterprise-level licenses in Year 1. As the organization's mobile program scaled in years 2 and 3, it purchased an additional two Xamarin University seats and enterprise-level platform licenses in each year.

Over the three-year forecast period, the composite organization spent a total of \$81,966 on Visual Studio Enterprise software license and Xamarin University subscription fees. For existing Visual Studio customers, Xamarin for Visual Studio is already included in their subscription and will result in no additional software subscription charges.

TABLE 5

Visual Studio Enterprise Annual Software License And Xamarin University Subscription Fees

Ref.	Metric	Calculation	Initial	Year 1	Year 2	Year 3
D1	Number of mobile developers added to Xamarin for Visual Studio development team	Includes internal and outsourced staff building on the platform		6	2	2
D2	Total number of mobile developers building on Xamarin for Visual Studio			6	8	10
D3	Annual Visual Studio Enterprise software subscription cost, per developer			\$2,999	\$2,999	\$2,999
D4	Annual Xamarin University licensing costs, per developer			\$999	\$999	\$999
Dt	Visual Studio Enterprise annual software and Xamarin University subscriptions	$(D1 \cdot D4) + (D2 \cdot D3)$		\$23,988	\$25,990	\$31,988
	Risk adjustment	0%				
Dtr	Visual Studio Enterprise annual software and Xamarin University subscriptions (risk-adjusted)		\$0	\$23,988	\$25,990	\$31,988

Source: Forrester Research, Inc.



Xamarin For Visual Studio Training, Implementation, Staffing, And Skill Acquisition Costs

During the first year after deploying Xamarin for Visual Studio, the composite organization needed to train and onboard six existing .NET web application developers in cross-platform mobile development for iOS, Android, and UWP platforms using Xamarin for Visual Studio. In addition to incurring costs for Xamarin University seats, each developer spent 60 hours studying and becoming certified in Xamarin C# mobile development and an additional 10 hours learning iOS and Android APIs. At an average fully loaded mobile developer hourly cost of \$80, the composite organization spent \$33,600 on developer training in Year 1, and an additional \$11,200 in both years 2 and 3 as it added two additional developers to its Xamarin mobile development team in each year.

Given the composite organization's strategic focus on building its mobile program, it repurposed a number of its existing .NET web developers with significant existing C# programming acumen to build the mobile applications in its pipeline. In addition to putting these developers through rigorous Xamarin training and certification in cross-platform mobile development, the organization leveraged three outsourced, highly skilled Xamarin for Visual Studio developers in Year 1 and two outsourced developers in Year 2. These developers assisted the organization in building the mobile applications in its pipeline, while concurrently augmenting the skillsets of its repurposed C# developers. The composite organization spent \$499,200 and \$332,800 on staff augmentation services in years 1 and 2, respectively. Lastly, embedded in this cost category is the allocated salary expense for repurposed existing C# programming talent allocated to the organization's mobile development projects. At an average fully loaded annual salary of \$141,440 for these developers, which is 15% lower than Objective-C and Java developers, the composite organization spent a total of \$140,025, \$424,320, and \$990,080 in years 1, 2, and 3, respectively.

Forrester risk-adjusted the training, implementation, staffing, and skill acquisition costs upward by 5% to reflect variance in the number of study hours required to become certified in Xamarin for Visual Studio and variance in the average fully loaded salary of C# mobile developer talent.

TABLE 6

Xamarin For Visual Studio Training, Implementation, Staffing, And Skill Acquisition Costs

Ref.	Metric	Calculation	Initial	Year 1	Year 2	Year 3
E1	Xamarin certification and study hours per developer			60	60	60
E2	iOS and Android API training hours			10	10	10
E3	Total number of mobile developers added to Xamarin development team			6	2	2
E4	Initial internal Xamarin resource requirements (man-hours)		4			
E5	Average hourly cost of a mobile application developer		\$80	\$80	\$80	\$80
E6	Number of C# developers repurposed and allocated to the Xamarin for Visual Studio mobile development team			3	6	10
E7	Percentage of repurposed C# developer time allocated to mobile development projects			33%	50%	70%
E8	Total number of FTEs allocated to the Xamarin for Visual Studio mobile development team	$E6 * E7$		1	3	7
E9	Average fully loaded annual cost per C# developer			\$141,440	\$141,440	\$141,440
E10	Allocated salary expenses for repurposed Xamarin for Visual Studio developers	$E6 * E7 * E9$		\$140,026	\$424,320	\$990,080
E11	Outsourced additional seasoned C# developer resources (staff augmentation)	Outsourced		\$499,200	\$332,800	\$0
Et	Xamarin for Visual Studio training, implementation, staffing, and skill acquisition costs	$((E1+E2)*E3*E5)+(E4*E5)+E10+E11$	\$320	\$672,826	\$768,320	\$1,001,280
	Risk adjustment	↑5%				
Etr	Xamarin for Visual Studio training, implementation, staffing, and skill acquisition costs (risk-adjusted)		\$336	\$706,467	\$806,736	\$1,051,344

Source: Forrester Research, Inc.



Microsoft Partner Professional Services

Following the deployment of Xamarin for Visual Studio, the composite organization engaged Microsoft professional services to bring in experienced senior engineers with a high degree of Xamarin expertise to expedite the development of the composite organization's first few mobile applications and provide incremental mobile developer skill and capability development support at a cost of \$25,000. As the organization scaled its mobile development practice in Year 1, the composite purchased an addition \$50,000 engagement to help jumpstart its mobile application development initiatives and further build the cross-platform mobile development capabilities of the organization's repurposed C# programming talent.

TABLE 7

Microsoft Partner Professional Services

Ref.	Metric	Calculation	Initial	Year 1	Year 2	Year 3
F1	Microsoft partner professional services		\$25,000	\$50,000		
Ft	Microsoft partner professional services		\$25,000	\$50,000	\$0	\$0

Source: Forrester Research, Inc.

Total Costs

Table 8 shows the total of all costs as well as associated present values (PVs), discounted at 10%. Over three years, the composite organization expects total costs to be a PV of a little less than \$2.24 million.

TABLE 8

Total Costs (Risk-Adjusted)

Ref.	Cost Category	Initial	Year 1	Year 2	Year 3	Total	Present Value
Dtr	Visual Studio Enterprise annual software and Xamarin University subscriptions	\$0	\$23,988	\$25,990	\$31,988	\$81,966	\$67,320
Etr	Xamarin training, implementation, staffing, and skill acquisition costs	\$336	\$706,467	\$806,736	\$1,051,344	\$2,564,883	\$2,099,193
Ftr	Microsoft partner professional services	\$25,000	\$50,000	\$0	\$0	\$75,000	\$70,455
Total costs (risk-adjusted)		\$25,336	\$780,455	\$832,726	\$1,083,332	\$2,721,849	\$2,236,967

Source: Forrester Research, Inc.

FLEXIBILITY

Flexibility, as defined by TEI, represents an investment in additional capacity or capability that could be turned into business benefit for some future additional investment. This provides an organization with the “right” or the ability to engage in future initiatives but not the obligation to do so. There are multiple scenarios in which a customer might choose to implement Xamarin for Visual Studio and later realize additional uses and business opportunities. Flexibility would also be quantified when evaluated as part of a specific project (described in more detail in Appendix A).

In order to meet and exceed the increasingly sophisticated mobile needs of its customer base, the composite organization is investing significant resources into building its consumer-facing mobile applications and porting these applications across mobile operating systems in order to provide its customers with the fully native experience they demand, on any device, in their time of need. In order to win, serve, and retain its diverse customer base, the organization will continue to port its consumer-facing mobile applications to additional mobile platforms, ensuring that it is delivering a consistent, native mobile experience across its customer base. Furthermore, since the organization is dedicated to incorporating new features and

technologies into its mobile applications as they become available, it expects to be able to deliver these new features quicker by using Xamarin for Visual Studio's cross-platform tool than it would using a pure native development approach.

Furthermore, the organization is heavily focused on delivering productivity-enhancing employee-facing applications, and expects to mobilize portions of the point-of-sale and inventory management functions in the coming years. In order to deliver this powerful functionality to all its employees regardless of the mobile device they own, the organization will use Xamarin for Visual Studio to mobilize these business technologies, reducing cost and improving time-to-value.

Lastly, the organization's strategic mobile road map encompasses monitoring, and, as necessary, scaling its mobile solutions to new and emerging form factors, including Android Wear, Apple Watch, and Apple TV. The company will continue to explore opportunities with emerging device form factors and will leverage Xamarin for Visual Studio to scale to these devices when it becomes strategically and economically viable.

“With devices like Apple Watch, Xamarin for Visual Studio is out there and they’re in front of it before we were ever even able to think about supporting it in our organization. It’s nice to know that by the time that we are ready to build a solution for these emerging devices, Xamarin for Visual Studio is already ahead of us, and they’re looking at things well before we even get our hands on it.”

~ VP of application development, global entertainment company

RISKS

Forrester defines two types of risk associated with this analysis: “implementation risk” and “impact risk.” Implementation risk is the risk that a proposed investment in Xamarin for Visual Studio may deviate from the original or expected requirements, resulting in higher costs than anticipated. Impact risk refers to the risk that the business or technology needs of the organization may not be met by the investment in Xamarin for Visual Studio, resulting in lower overall total benefits. The greater the uncertainty, the wider the potential range of outcomes for cost and benefit estimates.

TABLE 9

Benefit And Cost Risk Adjustments

Benefits	Adjustment
Reduction in mobile application development costs	↓ 10%
Mobile application maintenance and upgrade efficiency gains	↓ 5%
Cost avoidance of platform-specific mobile application developer talent	↓ 5%
Costs	Adjustment
Visual Studio Enterprise annual software license and Xamarin University subscription fees	↑ 0%
Xamarin for Visual Studio training, implementation, staffing, and skill acquisition costs	↑ 5%
Microsoft partner professional services	↑ 0%

Source: Forrester Research, Inc.

Quantitatively capturing implementation risk and impact risk by directly adjusting the financial results provides more meaningful and accurate estimates and a more accurate projection of the ROI. In general, risks affect costs by raising the original estimates, and they affect benefits by reducing the original estimates. The risk-adjusted numbers should be taken as “realistic” expectations since they represent the expected values considering risk.

The following impact risks that affect benefits are identified as part of the analysis:

- › The amount of C# code sharing across iOS, Android, and Windows platforms will vary depending on the unique nature of each organization’s mobile needs, usage of complementary tools such as Xamarin.Forms, and the level of customization and complexity required in the UI layer of the mobile application.
- › Labor costs avoided for mobile application development and mobile application life-cycle management will fluctuate based on the average fully loaded salary of mobile developer talent, which varies by region, skillset, and tenure with the company.
- › The number of mobile developers avoided by an organization will depend on its existing developer skillsets and the level of UI customization and complexity of each mobile project.

The following implementation risk that affects costs is identified as part of this analysis:

- › Training, implementation, staffing, and skill acquisition costs will vary based on the number of study hours required to become certified in Xamarin for Visual Studio and variance in the average fully loaded salary of C# mobile developer talent.

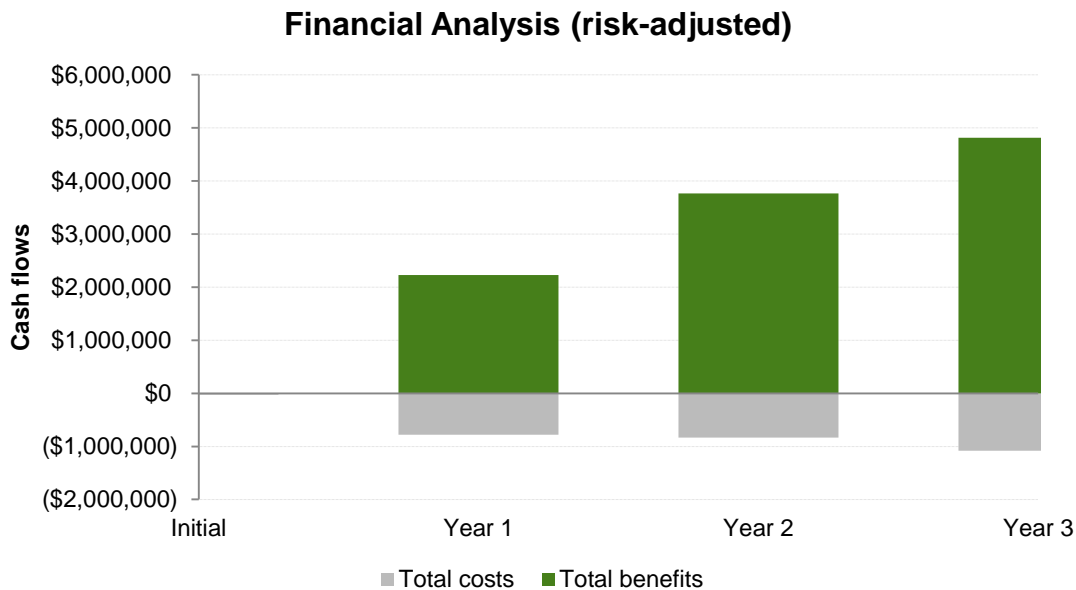
Table 9 shows the values used to adjust for risk and uncertainty in the cost and benefit estimates for the composite organization. Readers are urged to apply their own risk ranges based on their own degree of confidence in the cost and benefit estimates.

Financial Summary

The financial results calculated in the Benefits and Costs sections can be used to determine the ROI, NPV, and payback period for the composite organization's investment in Xamarin for Visual Studio.

Table 10 below shows the risk-adjusted ROI, NPV, and payback period values. These values are determined by applying the risk-adjustment values from Table 9 in the Risks section to the unadjusted results in each relevant cost and benefit section.

FIGURE 5
Cash Flow Chart (Risk-Adjusted)



Source: Forrester Research, Inc.

TABLE 10
Cash Flow (Risk-Adjusted)

Summary	Initial	Year 1	Year 2	Year 3	Total	Present Value
Total costs	(\$25,336)	(\$780,455)	(\$832,726)	(\$1,083,332)	(\$2,721,849)	(\$2,236,967)
Total benefits	\$0	\$2,226,432	\$3,763,328	\$4,816,384	\$10,806,144	\$8,752,838
Total	(\$25,336)	\$1,445,977	\$2,930,602	\$3,733,052	\$8,084,295	\$6,515,871
ROI						291%
Payback period (months)						0.2

Source: Forrester Research, Inc.

Xamarin For Visual Studio: Overview

The following information is provided by Xamarin. Forrester has not validated any claims and does not endorse Xamarin or its offerings.

Xamarin for Visual Studio allows mobile developers to build iOS, Android, Windows, and Mac apps using a shared C# code base. Unlike other cross-platform tools, Xamarin for Visual Studio creates fully native apps, with native UI, native performance, and complete access to the native APIs of each target operating system. Anything you can do in Objective-C, Swift, or Java, you can do in C#. Xamarin for Visual Studio integrates with a suite of mobile DevOps solutions from Microsoft to build, test, deploy, monitor, and improve mobile apps.

Xamarin for Visual Studio accelerates the creation of mission-critical consumer and enterprise apps for over a million developers worldwide, including Alaska Airlines, Coca-Cola Bottling, Siemens, McKesson, and more than 100 of the Fortune 500.

To learn more about Xamarin for Visual Studio, visit <https://www.xamarin.com/>.

Appendix A: Total Economic Impact™ Overview

Total Economic Impact is a methodology developed by Forrester Research that enhances a company's technology decision-making processes and assists vendors in communicating the value proposition of their products and services to clients. The TEI methodology helps companies demonstrate, justify, and realize the tangible value of IT initiatives to both senior management and other key business stakeholders. TEI assists technology vendors in winning, serving, and retaining customers.

The TEI methodology consists of four components to evaluate investment value: benefits, costs, flexibility, and risks.

BENEFITS

Benefits represent the value delivered to the user organization — IT and/or business units — by the proposed product or project. Often, product or project justification exercises focus just on IT cost and cost reduction, leaving little room to analyze the effect of the technology on the entire organization. The TEI methodology and the resulting financial model place equal weight on the measure of benefits and the measure of costs, allowing for a full examination of the effect of the technology on the entire organization. Calculation of benefit estimates involves a clear dialogue with the user organization to understand the specific value that is created. In addition, Forrester also requires that there be a clear line of accountability established between the measurement and justification of benefit estimates after the project has been completed. This ensures that benefit estimates tie back directly to the bottom line.

COSTS

Costs represent the investment necessary to capture the value, or benefits, of the proposed project. IT or the business units may incur costs in the form of fully burdened labor, subcontractors, or materials. Costs consider all the investments and expenses necessary to deliver the proposed value. In addition, the cost category within TEI captures any incremental costs over the existing environment for ongoing costs associated with the solution. All costs must be tied to the benefits that are created.

FLEXIBILITY

Within the TEI methodology, direct benefits represent one part of the investment value. While direct benefits can typically be the primary way to justify a project, Forrester believes that organizations should be able to measure the strategic value of an investment. Flexibility represents the value that can be obtained for some future additional investment building on top of the initial investment already made. For instance, an investment in an enterprise wide upgrade of an office productivity suite can potentially increase standardization (to increase efficiency) and reduce licensing costs. However, an embedded collaboration feature may translate to greater worker productivity if activated. The collaboration can only be used with additional investment in training at some future point. However, having the ability to capture that benefit has a PV that can be estimated. The flexibility component of TEI captures that value.

RISKS

Risks measure the uncertainty of benefit and cost estimates contained within the investment. Uncertainty is measured in two ways: 1) the likelihood that the cost and benefit estimates will meet the original projections and 2) the likelihood that the estimates will be measured and tracked over time. TEI risk factors are based on a probability density function known as "triangular distribution" to the values entered. At a minimum, three values are calculated to estimate the risk factor around each cost and benefit.

Appendix B: Glossary

Discount rate: The interest rate used in cash flow analysis to take into account the time value of money. Companies set their own discount rate based on their business and investment environment. Forrester assumes a yearly discount rate of 10% for this analysis. Organizations typically use discount rates between 8% and 16% based on their current environment. Readers are urged to consult their respective organizations to determine the most appropriate discount rate to use in their own environment.

Net present value (NPV): The present or current value of (discounted) future net cash flows given an interest rate (the discount rate). A positive project NPV normally indicates that the investment should be made, unless other projects have higher NPVs.

Present value (PV): The present or current value of (discounted) cost and benefit estimates given at an interest rate (the discount rate). The PV of costs and benefits feed into the total NPV of cash flows.

Payback period: The breakeven point for an investment. This is the point in time at which net benefits (benefits minus costs) equal initial investment or cost.

Return on investment (ROI): A measure of a project's expected return in percentage terms. ROI is calculated by dividing net benefits (benefits minus costs) by costs.

A NOTE ON CASH FLOW TABLES

The following is a note on the cash flow tables used in this study (see the example table below). The initial investment column contains costs incurred at "time 0" or at the beginning of Year 1. Those costs are not discounted. All other cash flows in years 1 through 3 are discounted using the discount rate at the end of the year. PV calculations are calculated for each total cost and benefit estimate. NPV calculations are not calculated until the summary tables are the sum of the initial investment and the discounted cash flows in each year.

Sums and present value calculations of the Total Benefits, Total Costs, and Cash Flow tables may not exactly add up, as some rounding may occur.

TABLE [EXAMPLE]

Example Table

Ref.	Metric	Calculation	Year 1	Year 2	Year 3
------	--------	-------------	--------	--------	--------

Source: Forrester Research, Inc.

Appendix C: Endnotes

¹ Source: “What Does It Cost To Source A Mobile App?” Forrester Research, Inc., December 8, 2015.