

A Forrester Total Economic Impact™
Study Commissioned By JetBrains
October 2018

The Total Economic Impact™ Of JetBrains IntelliJ IDEA

Cost Savings And Business Benefits
Enabled By IntelliJ IDEA



An abbreviated summary of this Total Economic Impact case study is available here:

https://resources.jetbrains.com/storage/products/intellij-idea/docs/forrester_tei_summary_en.pdf

Project Director:
Benjamin Brown

Table Of Contents

Executive Summary	1
Key Findings	1
TEI Framework And Methodology	3
The IntelliJ IDEA Customer Journey	4
Interviewed Organizations	4
Key Challenges Before IntelliJ IDEA	4
Why JetBrains IntelliJ IDEA?	5
Key Results With IntelliJ IDEA	5
Composite Organization	6
Analysis Of Benefits	7
Benefit 1: Development	7
Benefit 2: Testing And Debugging	10
Benefit 3: Maintenance	11
Benefit 4: Onboarding	12
Unquantified Benefits	13
Flexibility	14
Analysis Of Costs	15
Cost 1: Licensing	15
Cost 2: Systems Administration	15
Cost 3: Training And Customization	16
Financial Summary	17
JetBrains IntelliJ IDEA: Overview	18
Appendix A: Total Economic Impact	19
Appendix B: Endnotes	20

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.

© 2018, 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 forrester.com.

Executive Summary

JetBrains commissioned Forrester Consulting to conduct a Total Economic Impact™ (TEI) study and examine the potential return on investment (ROI) enterprises may realize by utilizing IntelliJ IDEA. The purpose of this study is to provide readers with a framework to evaluate the potential financial impact for their organizations.

IntelliJ IDEA is an integrated development environment (IDE) that helps developers be more efficient by writing better code faster and with fewer bugs. To better understand the benefits, costs, and risks associated with this investment, Forrester interviewed four customers with years of experience using IntelliJ IDEA.

Prior to using IntelliJ IDEA, interviewed organizations used alternative solutions such as text editors or open-source IDEs. However, compared to IntelliJ IDEA, these prior solutions created friction and inefficiencies that reduced developer productivity, increased the number of bugs, slowed the release of new functionality, and led to an accumulation of technical debt over time.

Forrester developed a composite organization based on data gathered from the customer interviews to reflect the total economic impact that IntelliJ IDEA could have on an organization. The composite organization is representative of the organizations that Forrester interviewed and is used to present the aggregate financial analysis in this study. All values are reported in risk-adjusted three-year present value (PV) unless otherwise indicated. Note that these numbers will vary based on industry, size of company, adoption rate, and other aspects particular to a specific organization.

Key Findings

Quantified benefits. The following benefits reflect the financial analysis associated with the composite organization:

- › **Improved developer productivity resulting in a benefit of \$7.6 million.** Customers noted that developers spent less time fixing bugs and code maintenance. Additionally, developers improved their productivity while writing code due to: code auto-completion, refactoring, avoiding fixing basic errors, enhanced focus, and reduced technical debt. In total, the composite organization's 1,000 developers recaptured 162,250 hours of productivity because of IntelliJ IDEA.
- › **Improved testing and debugging productivity and savings resulting in a benefit of \$7.7 million.** Interviewed organizations noted benefits related to testing and debugging, as tester productivity improved due to the IntelliJ IDEA debugger, improved testing plans, and reduced defect density from developers. In total, the 1,000 developers and 500 testers recaptured 161,720 hours of productivity.
- › **Improved code maintenance effort resulting in a benefit of \$3.3 million.** Interviewed organizations noted benefits related to code maintenance due to reduced defect density. As a result, the 1,000 developers recaptured 69,680 hours of productivity.

Benefits And Costs



Time saved by developers for development, debugging, and maintenance:

**270 hours
per user, per year**



Net annual benefit per IntelliJ IDEA user:

**\$3,868 per user,
per year**



Return on investment:

**\$8.50 in benefits
per \$1.00 in costs**



ROI
850%



Benefits PV
\$19.5 million



Costs PV
\$2.0 million



NPV
\$17.4 million



IntelliJ IDEA enhanced productivity and code quality.

- › **Improved new hire onboarding resulting in a benefit of \$845,195.** Interviewed organizations noted a decrease in overall onboarding time for new employees with IntelliJ IDEA due to the reduced technical debt, better user interface, shared configuration files, and implementation of styles guides and templates. In total, 18,000 hours are saved in the onboarding process because of IntelliJ IDEA.

Unquantified benefits. The interviewed organizations experienced additional qualitative benefits. These are not quantified in the financial analysis, but were mentioned as significant benefits by customers:

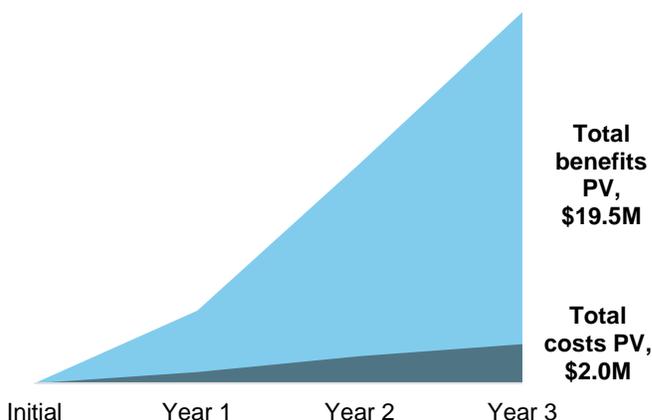
- › Elevated peer reviews.
- › Enhanced collaboration.
- › Accelerated time-to-market.
- › Reduced customer impact of bugs.
- › Recruiting and retaining top talent and reducing employee churn.

Costs. The following costs reflect the financial analysis associated with the composite organization.

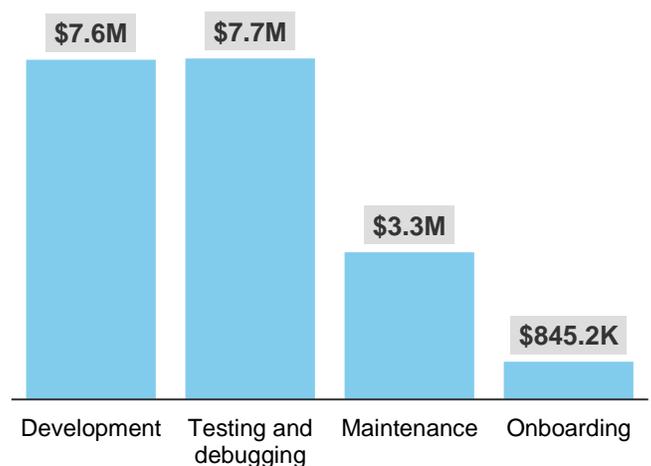
- › **Licensing costs of \$1.2 million.** Licensing costs are provided by JetBrains and total \$499/user in Year 1, \$399/user in Year 2, and \$299/user in Year 3. The composite organization has 750 users with IntelliJ IDEA licenses in Year 1, and 1,500 users in years 2 and 3.
- › **System administration costs of \$86,034.** Customers noted systems administration effort associated with working with JetBrains, setting up user accounts, and providing support to users.
- › **Training and customization costs of \$732,239.** Customers noted that training and customization efforts associated with IntelliJ IDEA are minimal, as the product is widely known and has an easy-to-use interface, and configuration files that have been created previously for employees can be shared across machines.

Forrester's interviews with four existing customers and subsequent financial analysis found that an organization based on these interviewed organizations experienced benefits of \$19.5 million over three years versus costs of \$2.0 million, adding up to a net present value (NPV) of \$17.4 million and an ROI of 850%.

Financial Summary



Benefits (Three-Year)



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 Framework And Methodology

From the information provided in the interviews, Forrester has constructed a Total Economic Impact™ (TEI) framework for those organizations considering implementing JetBrains IntelliJ IDEA.

The objective of the framework is to identify the cost, benefit, flexibility, and risk factors that affect the investment decision. Forrester took a multistep approach to evaluate the impact that JetBrains IntelliJ IDEA can have on an organization:



DUE DILIGENCE

Interviewed JetBrains stakeholders and Forrester analysts to gather data relative to IntelliJ IDEA.



CUSTOMER INTERVIEWS

Interviewed four organizations using IntelliJ IDEA to obtain data with respect to costs, benefits, and risks.



COMPOSITE ORGANIZATION

Designed a composite organization based on characteristics of the interviewed organizations.



FINANCIAL MODEL FRAMEWORK

Constructed a financial model representative of the interviews using the TEI methodology and risk-adjusted the financial model based on issues and concerns of the interviewed organizations.



CASE STUDY

Employed four fundamental elements of TEI in modeling JetBrains IntelliJ IDEA's impact: benefits, costs, flexibility, and risks. 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.

DISCLOSURES

Readers should be aware of the following:

This study is commissioned by JetBrains 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 JetBrains IntelliJ IDEA.

JetBrains 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.

JetBrains provided the customer names for the interviews but did not participate in the interviews.

The IntelliJ IDEA Customer Journey

BEFORE AND AFTER THE INTELLIJ IDEA INVESTMENT

Interviewed Organizations

For this study, Forrester conducted four interviews with JetBrains IntelliJ IDEA customers. Interviewed customers include the following:

INDUSTRY	REGION	INTERVIEWEE	NUMBER OF DEVELOPERS/ EMPLOYEES
Software automation	North America	CEO	Less than 50 developers Less than 100 employees
Media	Global	Lead engineer Software developer	500 to 1,000 developers 1,000 to 5,000 employees
Online gaming	Europe	Software developer	100 to 500 developers 1,000 to 5,000 employees
Development platform	Global	Technical lead	50 to 100 developers 100 to 500 employees

Key Challenges Before IntelliJ IDEA

Before the investment in IntelliJ IDEA, interviewees described the following challenges with their previous solution:

- › **Inefficiencies resulting in reduced developer productivity.** Customers noted that text editors and open-source IDEs resulted in significant inefficiencies that impacted developer productivity. A technical lead for a development platform stated simply, “I would forbid developers from using text editors at our organization if I could.”
- › **Inability to reduce the number of bugs — with negative customer outcomes.** The CEO of a software automation company told Forrester: “We had an issue where it took us two weeks to get most of the bugs ironed out after a release. This led to a significant increase in customer churn; it was painful. Months later our customers were still mentioning the slowdowns — it feels like it takes them years to forget that kind of stuff. And so bugs have that serious, serious impact to the customer that they really don’t seem to forget.”
- › **Reduced capability to deploy new features quickly.** The CEO of a software automation company stated: “A company has to keep pumping out features in order to stay competitive. It really comes down to: can you develop new features in a reasonable amount of time, have it work within our performance envelope, and do it all with a reasonable amount of bugs.”
- › **Accumulation of technical debt over time.** A technical lead for a development platform said: “With our previous solution, you had this accumulation of technical debt. For example, bad names that just stay around. It’s hard to put a number on it, but it would be prohibitively expensive to go back and clean it up.”

“Our [legacy IDE] was horrible. Learning the way your [editor or IDE] works can be one of the biggest pain points, but [IntelliJ IDEA] has good aesthetics right out of the box and the ergonomics are very thoughtful.”

CEO, software automation company



“The refactoring and coding insights are accurate, and that gives you the confidence to change or update your code.”

Technical lead, development platform



Why JetBrains IntelliJ IDEA?

Interviewed organizations stated the following reasons on why they chose JetBrains IntelliJ IDEA to address their challenges:

- › **Ease of use.** A software developer for a media company told Forrester, “It’s much easier for developers to become familiar with IntelliJ IDEA than the other alternatives out there.”
- › **Number of effective features.** Customers listed IntelliJ IDEA’s features as a key differentiator, which include: multi-language support, contextual data, code completion, refactoring, debugger, and the ability to enforce a style guide. A software developer for an online gaming organization told Forrester, “It’s the best IDE on the market right now.”
- › **Opportunity to improve developer productivity with minimal risk.** The CEO of a software automation company noted, “With what we’re paying our developers to deliver quality code, even a 1% improvement in productivity is worth it for just the few hundred bucks it costs per year for the subscription to IntelliJ IDEA.”

Key Results With IntelliJ IDEA

The interviews revealed that the investment in IntelliJ IDEA addressed the challenges interviewees were facing, ultimately increasing productivity and improving code quality. Key results include:

IMPROVED PRODUCTIVITY, REDUCED BUGS, INCREASED SPEED

- › **Improved ability to focus on writing code.** The CEO of a software automation company said: “IntelliJ IDEA lets the brain stay in that state where you’re developing that mental model of what your code looks like, and a little less on being mechanical, like going around and finding places to change the name of something; staying in that flow state that we all seek to stay in as long as we can.”
- › **More accurate code writing.** A technical lead for a development platform stated: “It definitely helps me write code right the first time around. IntelliJ IDEA helps me along the way, and then when I go and actually use the build tool I don’t get an error.”
- › **Fewer feedback cycles for QA.** A lead engineer for a media company told Forrester: “I would say the biggest impact is the duration speed. The feedback cycle is shorter. It takes fewer iterations for developers to arrive at a fresh new code.”

REDUCED TECHNICAL DEBT

- › **Refactoring capability.** The CEO of a software automation company said: “Refactoring is a big thing. Let’s say that you have to add or remove a parameter; in just in one step, with refactoring, it gets changed everywhere . . . it is a big timesaver.”
- › **Developers more quickly understand the code base.** A technical lead for a development platform stated: “I switch between different people’s code a lot, and being able to jump into another person’s code base and quickly being able to understand it is important.”

“JetBrains made that big leap in the early 2000s with refactoring and code productivity. They wrote great code. They continue to hold onto that lead very well.”

CEO, software automation company



“We hired an engineer who showcased [IntelliJ IDEA]. It was more effective than [our other IDEs], and he evangelized it to spread to his local development team and then onwards in the organization.”

Lead engineer, media



“There’s a limit to how many lines of code you can write a day, and it goes up when using IntelliJ IDEA.”

Technical lead, development platform



“The refactoring and coding insights are accurate, and that gives you the confidence to change or update your code.”

Technical lead, development platform



ADDITIONAL BENEFITS

- › **Improved employee experience and satisfaction.** A lead engineer for a media company noted: “Having IntelliJ IDEA as an organization definitely makes it a better workplace for our developers. They’re happier now that we’re paying for a license and not making them use some free tool that’s not nearly as effective.”
- › **Ability to attract and retain top talent.** A lead engineer for a media company told Forrester: “The time savings and productivity from IntelliJ IDEA is a great benefit. But it goes even beyond that. Companies really want to hire good talent, and a lot of [potential employees] would be put off by a company saying, ‘We’re not going to pay for your IDE so you have to use [an open-source text editor or IDE] that’s free.’ It’s more about attracting talent and retaining talent, and IntelliJ IDEA helps us do that.”

Composite Organization

Based on the interviews, Forrester constructed a TEI framework, a composite organization, and an associated ROI analysis that illustrates the areas financially affected. The composite organization is representative of the four companies that Forrester interviewed and is used to present the aggregate financial analysis in the next section. The composite organization that Forrester synthesized from the customer interviews has the following characteristics:

Description of composite organization. The composite organization is a global enterprise based in the US. The organization has a 2:1 ratio of developers to testers, with 1,000 developers and 500 testers. By the beginning of the first year, IntelliJ IDEA licenses are rolled out to half of the developers and testers at the composite organization. By the beginning of Year 2, a license for IntelliJ IDEA is provided for all developers and testers as the composite organization chooses to standardize usage of IntelliJ IDEA across the organization.

“You hear developers say they don’t like the open office and they don’t like interruptions. It’s because of that ‘flow’ state; they want to preserve it. IntelliJ IDEA’s features help with preserving that ‘flow.’”

CEO, software automation company



“Companies really want to hire good talent, and a lot of [potential employees] would be put off by a company saying, ‘We’re not going to pay for your IDE so you have to use [an open-source text editor or IDE] that’s free.’ It’s more about attracting talent and retaining talent, and IntelliJ IDEA helps us do that.”

Lead engineer, media



Key assumptions:

- › 1,000 developers
- › 500 testers
- › 1,500 IntelliJ IDEA licenses

Analysis Of Benefits

QUANTIFIED BENEFIT DATA AS APPLIED TO THE COMPOSITE ORGANIZATION

Total Benefits

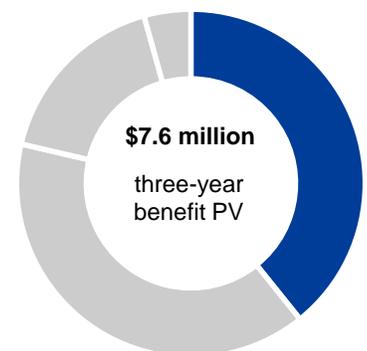
REF.	BENEFIT	YEAR 1	YEAR 2	YEAR 3	TOTAL	PRESENT VALUE
Atr	Development	\$1,447,875	\$3,627,000	\$4,416,750	\$9,491,625	\$7,632,140
Btr	Testing and debugging	\$1,764,360	\$3,711,240	\$3,985,020	\$9,460,620	\$7,665,109
Ctr	Maintenance	\$790,920	\$1,642,680	\$1,642,680	\$4,076,280	\$3,310,775
Dtr	Onboarding	\$175,500	\$351,000	\$526,500	\$1,053,000	\$845,195
	Total benefits (risk-adjusted)	\$4,178,655	\$9,331,920	\$10,570,950	\$24,081,525	\$19,453,219

Benefit 1: Development

IntelliJ IDEA enabled developers to work more efficiently. Through better focus, live analysis, and real-time recommendations, developers were able to complete their work faster and with less disruption. Further, IntelliJ IDEA enabled developers to write better code — with consistent style, fewer bugs or exceptions, better referencing of existing code base, better structure, and beyond. Better quality code led to further productivity improvements for both developers and testers as technical debt decreased overtime. Interviewed organizations described the following benefits from using IntelliJ IDEA:

- › Increased speed by accepting intelligent code completion recommendations rather than writing them out, or potentially trying to come up with alternative ways to solve the problem. Developers were posed with almost 200 code recommendations per day, accepting over half of them instantly and only rejecting a small percentage. To the software developer at the media company, “The code completion features in IntelliJ [IDEA] are, at least in my mind, one of the killer features.”
- › Avoided writing and needing to fix basic errors such as spelling and syntax through live analysis and code completion. The lead engineer for the media company explained: “[IntelliJ IDEA] has a live analysis — if you type something that’s prone to a null pointer exception, for instance, it can highlight that. And if you mouseover that code, it explains the issue and helps you fix it. This is another killer feature of IntelliJ IDEA, it’s so prevalent in day-to-day work, you use it constantly.”
- › Avoided complex and potentially risky rewriting of code by using automated refactoring. Instead of trying to find each place a function or variable was referenced, developers could simply change it once and let IntelliJ IDEA automatically fix it throughout the code base. Further, poorly written code could be automatically restructured to better style.
 - The media company’s software engineer explained: “I use the refactoring tool daily. It improves the quality of the code. If I want to extract some piece of code to a method, for example, I have a lot more confidence that IntelliJ IDEA would do it properly, instead if I were to do it myself manually because I

The table above shows the total of all benefits across the areas listed below, as well as present values (PVs) discounted at 10%. Over three years, the composite organization expects risk-adjusted total benefits to have a PV of nearly \$19.5 million.



Development:
39% of total benefits

“Before, using a text editor, you wouldn’t change things because you worried that you would miss renaming something and your code wouldn’t compile anymore.”

Technical lead, development platform



might make mistakes. It's also much faster. I use refactoring regularly because it's better and faster than I am at doing it."

- › Improved focus by reducing the need to switch windows, search other documents, look things up, and go through wizards. Not only does this save developers time, it also reduces developer distraction.
 - The technical lead for the development platform explained: "[IntelliJ IDEA] gives you a lot of contextual information about where you are in your code. It goes into that source file with thousands of lines of code, lots of definitions, and it gives you a breadcrumbs view — you're in this class, this method, and so on. Not having to have this in your head all the time definitely helps, especially when you're switching between different projects."
- › Reduced time spent debugging during testing and maintaining production code, enabling more time to be dedicated to building functionality.
- › Reduced technical debt over time with better quality code, helping to make the code base easier to understand, reference, build upon, and revise in the future with less confusion and risk of unforeseen issues.
 - The technical lead for the development platform described: "The hurdle to cleaning up things that you happen to notice while you're working goes down [with IntelliJ IDEA] because it can help you fix it and feel confident that you got it right. It helps a lot."

Based on the customer interviews, Forrester estimates for the composite organization:

- › Fifty percent of developers adopt IntelliJ IDEA in Year 1, which increases to 100% in years 2 and 3 as the organization chooses to standardize usage of IntelliJ IDEA.
- › Developers spend 37% of their time working on new functionality; when combined with 20% dedicated to maintenance and 18% to maintenance, they work at an estimated 75% utilization rate.
- › Productivity improvements from code auto-completion, refactoring, avoiding fixing basic errors, enhanced focus, and reduced technical debt, along with reclaimed time from debugging and maintenance.
- › Productivity improvements are applied to the time developers recapture from debugging and maintenance, because not only do they gain those hours for development work, those hours are also spent working more efficiently than in the past.
- › Productivity capture rate of 50%.
- › Fully burdened salary of \$65/hr.

This benefit can vary due to uncertainty related to:

- › Structure and size of development teams.
- › Release cycle length and approach (such as Agile or Waterfall).
- › Company code base and style nuances.
- › Adoption rate of IntelliJ IDEA.
- › Fully burdened salary.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding an annual benefit ranging from \$1.4 million to \$4.4 million,

"I use refactoring regularly because it's better and faster than I am at doing it."

Software developer, media



"We have extremely high standards. By using IntelliJ, where you can interactively see problems without having to go through a huge build cycle, or worst case, integration cycle, where you need to push the code somewhere else is key. It helps us have extremely high-quality code."

Lead engineer, media



Impact risk is the risk that the business or technology needs of the organization may not be met by the investment, resulting in lower overall total benefits. The greater the uncertainty, the wider the potential range of outcomes for benefit estimates.

and a three-year risk-adjusted total PV of \$7.6 million.

Development: Calculation Table

REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
A1	Number of developers		1,000	1,000	1,000
A2	Percentage of developers using IntelliJ IDEA		50%	100%	100%
A3	Number of developers using IntelliJ IDEA	A1*A2	500	1,000	1,000
A4	Baseline percent of developer time spent on building functionality		37.0%	37.0%	37.0%
A5	Percent of developer time reclaimed from bug fixing	B11-B12	1.5%	3.0%	3.0%
A6	Percent of developer time reclaimed from maintenance	C2-C4	1.3%	2.7%	2.7%
A7	Percent of developer time spent building functionality with IntelliJ IDEA	A4+A5+A6	39.8%	42.7%	42.7%
A8	Productivity improvement: code auto-completion		3%	3%	3%
A9	Productivity improvement: refactoring code		2%	2%	2%
A10	Productivity improvement: avoiding fixing basic errors and debugging		4%	4%	4%
A11	Productivity improvement: enhanced focus		3%	3%	3%
A12	Productivity improvement from reduced technical debt	Technical debt is reduced over time	0%	2%	5%
A13	Total percentage improvement in development productivity	A8+A9+A10+A11+A12	12%	14%	17%
A14	Hours saved per developer, per year	2,080*A13*A7	99	124	151
A15	Total hours saved by developers	A3*A14	49,500	124,000	151,000
A16	Percentage of time recaptured for productivity		50%	50%	50%
A17	Hours recaptured for productivity	A15*A16	24,750	62,000	75,500
A18	Fully burdened hourly salary	\$130K annually	\$65	\$65	\$65
At	Development	A17*A18	\$1,608,750	\$4,030,000	\$4,907,500
	Risk adjustment	↓10%			
Atr	Development (risk-adjusted)		\$1,447,875	\$3,627,000	\$4,416,750

Benefit 2: Testing And Debugging

Interviewed organizations noted benefits related to testing and debugging for both testers and developers. Tester productivity improved due to the IntelliJ IDEA debugger, improved testing plans, and reduced defect density from developers. Further, developers themselves reduced the time they spent debugging and working with code during testing steps.

- › The CEO of the software automation company described: “The debugger is a lifesaver. Just being able to either set break points or step through code, it’s just awesome. When you have text editor, to debug, you need to put in print statements to see, ‘why is it doing this?’ Being able to slowly step through stuff or have breakpoints is a huge speed increase. You figure out what’s going on so much faster.”
- › The technical lead for the development platform described how IntelliJ IDEA accelerates debugging, “The trial and error of using a text editor is unnecessary with IntelliJ IDEA. It knows a lot more about the code than you can just read in the text editor. You no longer need to leave where you currently are, go hunt around for a method, figure out what types were involved, write a print statement, and try to find it.”

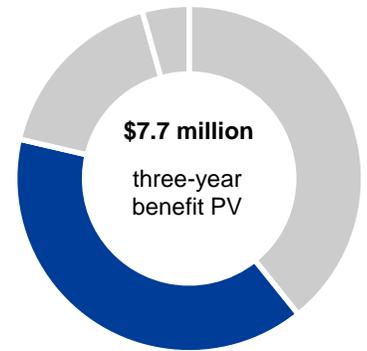
Based on the customer interviews, Forrester estimates for the composite organization:

- › Fifty percent of testers adopt IntelliJ IDEA in Year 1, increasing to 100% in years 2 and 3 as the organization standardizes usage of IntelliJ IDEA.
- › Tester productivity improves due to the IntelliJ IDEA debugger, improved testing plans, and reduced defect density from developers.
- › Developers spend 20% of their time fixing bugs, based on Forrester’s Global Business Technographics® Developer Survey, 2018.¹
- › Productivity capture rate of 50%.
- › Fully burdened salary of \$65/hr.

This benefit can vary due to uncertainty related to:

- › Achieved reduction in defect density.
- › Structure and size of development teams.
- › Release cycle length and approach (such as Agile or Waterfall).
- › Company code base and style nuances.
- › Adoption rate of IntelliJ IDEA.
- › Fully burdened salary.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding an annual benefit ranging from \$1.8 million to \$4.0 million, and a three-year risk-adjusted total PV of \$7.7 million.



Testing and debugging:
40% of total benefits

“The trial and error of using a text editor is unnecessary with IntelliJ IDEA. It knows a lot more about the code than you can just read in the text editor.”

Technical lead, development platform



Testing And Debugging: Calculation Table

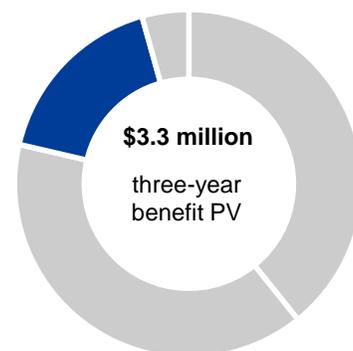
REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
B1	Number of testers	1:2 tester to developer ratio	500	500	500
B2	Percentage of testers using IntelliJ IDEA		50%	100%	100%
B3	Number of testers using IntelliJ IDEA	$B1*B2$	250	500	500
B4	Productivity improvement from using the IntelliJ IDEA debugger		2%	2%	2%
B5	Hours of tester time saved with debugger	$B3*B4*2,080$	10,400	20,800	20,800
B6	Percent of time developing tests		30.0%	30%	30%
B7	Improved productivity for testers developing tests	A13	12.0%	14.0%	17.0%
B8	Hours of tester time saved for writing tests	$2,080*B3*B6*B7$	18,720	43,680	53,040
B9	Reduced defect density with IntelliJ IDEA		7.5%	15.0%	15.0%
B10	Number of developers	A1	1,000	1,000	1,000
B11	Baseline percentage of developer time dedicated to bug fixing		20.0%	20.0%	20.0%
B12	Reduced percentage of developer time dedicated to bug fixing	$B11*(1-B9)$	18.5%	17.0%	17.0%
B13	Hours of developer time saved from reduced defect density	$B10*2,080*(B11-B12)$	31,200	62,400	62,400
B14	Total hours saved	$B5+B8+B13$	60,320	126,880	136,240
B15	Percentage of time recaptured for productivity		50%	50%	50%
B16	Hours recaptured for productivity	$B14*B15$	30,160	63,440	68,120
B17	Fully burdened hourly salary	A18	\$65	\$65	\$65
Bt	Testing and debugging	$B16*B17$	\$1,960,400	\$4,123,600	\$4,427,800
	Risk adjustment	↓10%			
Btr	Testing and debugging (risk-adjusted)		\$1,764,360	\$3,711,240	\$3,985,020

Benefit 3: Maintenance

Interviewed organizations noted benefits related to code maintenance due to reduced defect density, a byproduct of better code quality with IntelliJ IDEA. With developers writing better quality code, there were fewer bugs that risked slipping through QA and reduced technical debt that could lead to future issues — ultimately reducing the number of bugs that reached the production environment. With reduced bugs comes reduced need to maintain the code and fix issues when they occur.

Based on the customer interviews, Forrester estimates for the composite organization:

- › Developers using IntelliJ IDEA reduced defect density by 15%. In Year 1, this results in an overall defect density reduction of 7.5%, as only half of the developers at the composite organization are using IntelliJ IDEA at that time.



Maintenance:
17% of total benefits

- › Developers spend 18% of their time on maintenance, based on Forrester's Global Business Technographics Developer Survey, 2018.²
- › Productivity capture rate of 50%.
- › Fully burdened salary of \$65/hr.

This benefit can vary due to uncertainty related to:

- › Productivity improvements and reduction in defect density.
- › Adoption rate.
- › Fully burdened salary.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding an annual benefit ranging from \$791,000 to \$1.6 million, and a three-year risk-adjusted total PV of \$3.6 million.

Maintenance: Calculation Table

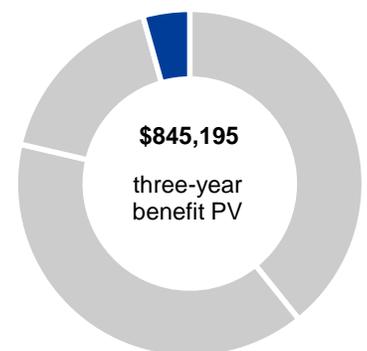
REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
C1	Number of developers	A1	1,000	1,000	1,000
C2	Baseline percentage of developer time dedicated to maintenance		18%	18%	18%
C3	Reduced defect density with developers using IntelliJ IDEA	B9	7.5%	15.0%	15.0%
C4	Reduced percentage of developer time dedicated to maintenance	$C2*(1-C3)$	16.7%	15.3%	15.3%
C5	Hours saved for maintenance	$C1*2,080*$ (C2-C4)	27,040	56,160	56,160
C6	Percentage of time recaptured for productivity		50%	50%	50%
C7	Hours recaptured for productivity	$C5*C6$	13,520	28,080	28,080
C8	Fully burdened hourly salary	A18	\$65	\$65	\$65
Ct	Maintenance	$C7*C8$	\$878,800	\$1,825,200	\$1,825,200
	Risk adjustment	↓10%			
Ctr	Maintenance (risk-adjusted)		\$790,920	\$1,642,680	\$1,642,680

Benefit 4: Onboarding

Interviewed organizations noted a decrease in overall onboarding time for new employees with IntelliJ IDEA due to the reduced technical debt, better user interface, shared configuration files, and implementation of styles guides and templates.

Based on the customer interviews, Forrester estimates for the composite organization:

- › Onboarding time for new employees reduces from 20 days to 18 days in Year 1, which improves to 16 days in Year 2 and 14 days in Year 3 as technical debt, configuration files, and templates improve each year and approach a steady state.
- › Employee churn rate of 25%.
- › Productivity capture rate of 50%.



Onboarding:
4% of total benefits

- › Fully burdened salary of \$65/hr.

This benefit can vary due to uncertainty related to:

- › Improvements in new employee onboarding.
- › Employee churn rate.
- › Fully burdened salary.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding an annual benefit ranging from \$175,500 to \$526,500, and a three-year risk-adjusted total PV of \$845,195.

Onboarding: Calculation Table					
REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
D1	Number of developers and testers	A1+B1	1,500	1,500	1,500
D2	Employee churn rate		25%	25%	25%
D3	Number of new employees per year	D1*D2	375	375	375
D4	Standard number of days for initial onboarding		20	20	20
D5	Number of days for initial onboarding with JetBrains		18	16	14
D6	Total hours saved	(D4-D5)*8*D3	6,000	12,000	18,000
D7	Percentage of time recaptured for productivity		50%	50%	50%
D8	Hours recaptured for productivity	D6*D7	3,000	6,000	9,000
D9	Fully burdened hourly salary	A18	\$65	\$65	\$65
Dt	Onboarding	D8*D9	\$195,000	\$390,000	\$585,000
	Risk adjustment	↓10%			
Dtr	Onboarding (risk-adjusted)		\$175,500	\$351,000	\$526,500

Unquantified Benefits

While there were strong and quantifiable benefits the interviewed organizations observed by using IntelliJ IDEA, there were significant qualitative benefits experienced as well.

- › **Elevated peer reviews.** Peer reviews can focus on bigger picture items like problem solving method, style, and streamlining code. With the previous solution, peer reviews would get bogged down by basic errors and syntax.
- › **Enhanced collaboration.** Developers can more easily share and understand each others' code thanks to the improved code quality and consistency resulting from live analysis, code completion, and refactoring.

"IntelliJ IDEA has a big impact on helping our developers spend more time designing and building products rather than iterating on the code, identifying bugs, and resolving them."

Technical lead, development platform



- › **Accelerated time-to-market.** IntelliJ IDEA improves business velocity, allowing organizations to release functionality faster while ensuring good architecture. Developers are emboldened and have the confidence to go faster and try new ideas with support from JetBrains IntelliJ IDEA live analysis, code completion, and refactoring features.
- › **Reduced customer impact of bugs.** Downtime can have significant impact on customer experience, churn, and overall sales — companies recognize that it must be avoided as much as possible. IntelliJ IDEA reduces the frequency and impact of bugs reaching production, and improves the overall quality of the code base — reducing instability and the risk of customer-facing issues. As the CEO of the software automation company described: “You can’t just move fast and have customers do QA. You’ll be penalized so fast by your customers if you’re shipping out bugs.”
- › **Recruiting and retaining top talent and reducing employee churn.** Top talent has an outsized impact on an organization. Additionally, reduced employee churn would further reduce onboarding costs. The media company’s software developer explained: “For me, it’s more about attracting and retaining talent — because IntelliJ IDEA is a really nice tool to work in and it saves time. It’s not the time I save that matters to me, as a developer, it’s my experience using it.”

“IntelliJ IDEA really increases my productivity and makes my work more enjoyable.”

Lead engineer, media



Flexibility

The value of flexibility is clearly unique to each customer, and the measure of its value varies from organization to organization. There are multiple scenarios in which a customer might choose to implement IntelliJ IDEA and later realize additional uses and business opportunities, including:

- › Using other JetBrains software and technology for other development and testing languages. This is made easier since JetBrains products integrate well and have the same user interface.
- › Improved pricing when bundling with other JetBrains products.
- › Availability of plug-ins from the marketplace and language support beyond Java (e.g., Scala).
- › Ability to develop in-house plug-ins.
- › Reduced support cost for bugs — fewer bugs also reduce the likelihood of support calls from customers.

Flexibility would also be quantified when evaluated as part of a specific project (described in more detail in Appendix A).

Flexibility, as defined by TEI, represents an investment in additional capacity or capability that could be turned into business benefit for a 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.

Analysis Of Costs

QUANTIFIED COST DATA AS APPLIED TO THE COMPOSITE ORGANIZATION

Total Costs							
REF.	COST	INITIAL	YEAR 1	YEAR 2	YEAR 3	TOTAL	PRESENT VALUE
Etr	Licensing	\$0	\$392,963	\$628,425	\$470,925	\$1,492,313	\$1,230,411
Ftr	Systems administration	\$0	\$50,669	\$25,334	\$25,334	\$101,338	\$86,034
Gtr	Training and customization	\$0	\$204,750	\$358,313	\$332,719	\$895,781	\$732,239
	Total costs (risk-adjusted)	\$0	\$648,381	\$1,012,072	\$828,978	\$2,489,431	\$2,048,684

Cost 1: Licensing

Licensing costs are provided by JetBrains and total \$499/user in Year 1, \$399/user in Year 2, and \$299/user in Year 3.

Licensing costs can vary due to the exact number of users; therefore Forrester adjusted this cost upward by 5%, yielding a three-year risk-adjusted total PV of \$1.2 million.

The table above shows the total of all costs across the areas listed below, as well as present values (PVs) discounted at 10%. Over three years, the composite organization expects risk-adjusted total costs to have a PV of \$2.0 million.

Licensing: Calculation Table

REF.	METRIC	CALC.	INITIAL	YEAR 1	YEAR 2	YEAR 3
E1	Number of developers using IntelliJ IDEA	A3		500	1,000	1,000
E2	Number of testers using IntelliJ IDEA	B3		250	500	500
E3	Annual license cost per user			\$499	\$399	\$299
Et	Licensing	(E1+E2)* E3	\$0	\$374,250	\$598,500	\$448,500
	Risk adjustment	↑5%				
Etr	Licensing (risk-adjusted)		\$0	\$392,963	\$628,425	\$470,925

Cost 2: Systems Administration

Customers noted minimal systems administration effort associated with working with JetBrains, setting up user accounts, and providing support to users along with creating style guides and templates. Based on the customer interviews, Forrester estimates for the composite organization:

- › Two full-time equivalents (FTEs) spend 20% of their time on system administration tasks in Year 1, and 10% in years 2 and 3.
- › Fully burdened system administrator salary of \$58/hr.

This benefit can vary due to uncertainty related to required effort for system administration and employees' fully burdened salary.

To account for these risks, Forrester adjusted this cost upward by 5%, yielding a three-year risk-adjusted total PV of \$86,034.

Implementation risk is the risk that a proposed investment may deviate from the original or expected requirements, resulting in higher costs than anticipated. The greater the uncertainty, the wider the potential range of outcomes for cost estimates.

Systems Administration: Calculation Table

REF.	METRIC	CALC.	INITIAL	YEAR 1	YEAR 2	YEAR 3
F1	Number of FTEs administering JetBrains, creating style guides and templates, and supporting users			2	2	2
F2	Percentage of time dedicated to JetBrains administration			20%	10%	10%
F3	Administrator hourly salary			\$58	\$58	\$58
Ft	Systems administration	$F1 \cdot F2 \cdot 2,080 \cdot F3$	\$0	\$48,256	\$24,128	\$24,128
	Risk adjustment	↑5%				
Ftr	Systems administration (risk-adjusted)		\$0	\$50,669	\$25,334	\$25,334

Cost 3: Training And Customization

Customers noted that training and customization efforts associated with IntelliJ IDEA are minimal, as the product is widely known and has an easy-to-use interface, and configuration files, which have been previously created for employees, can be shared across machines.

Based on the customer interviews, Forrester estimates for the composite organization:

- › Four hours per new user are spent on initial training and customization.
- › Three hours annually are spent per user on continuous learning and keeping instances updated.
- › Fully burdened salary of \$65/hr.

This benefit can vary due to uncertainty related to required initial setup and recurring training and employees' fully burdened salary.

To account for these risks, Forrester adjusted this cost upward by 5%, yielding a three-year risk-adjusted total PV of \$732,239.

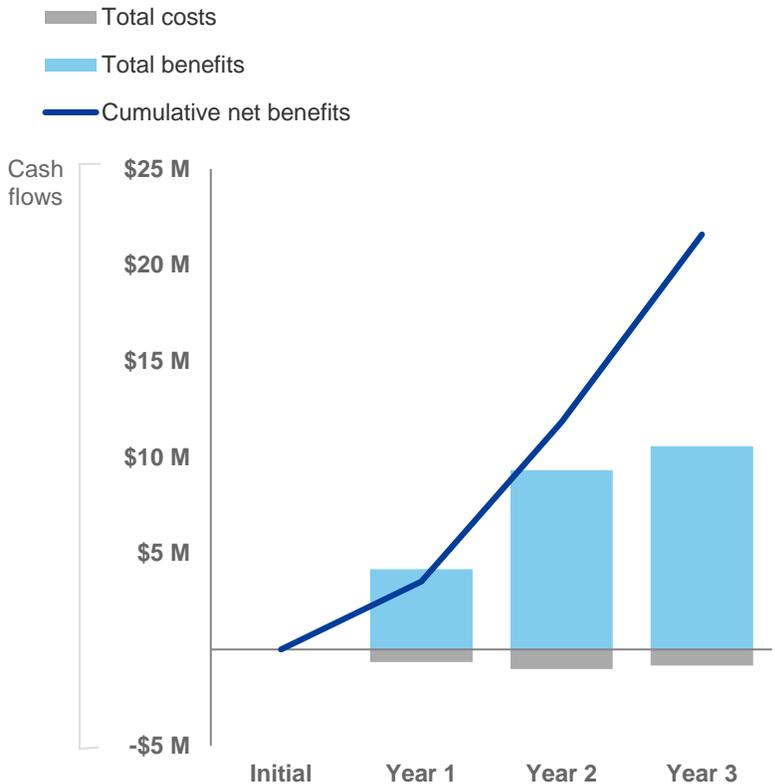
Training And Customization: Calculation Table

REF.	METRIC	CALC.	INITIAL	YEAR 1	YEAR 2	YEAR 3
G1	Number of new JetBrains users			750	750	375
G2	Number of existing JetBrains users				750	1,125
G3	Hours per developer spent for initial JetBrains training and customization			4	4	4
G4	Hours per developer spent for continuing learning and keeping instance updated			3	3	3
G5	Fully burdened hourly salary	A18		\$65	\$65	\$65
Gt	Training and customization	$((G1 \cdot G3) + (G2 \cdot G4)) \cdot G5$	\$0	\$195,000	\$341,250	\$316,875
	Risk adjustment	↑5%				
Gtr	Training and customization (risk-adjusted)		\$0	\$204,750	\$358,313	\$332,719

Financial Summary

CONSOLIDATED THREE-YEAR RISK-ADJUSTED METRICS

Cash Flow Chart (Risk-Adjusted)



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. Forrester assumes a yearly discount rate of 10% for this analysis.



These risk-adjusted ROI, NPV, and payback period values are determined by applying risk-adjustment factors to the unadjusted results in each Benefit and Cost section.

Cash Flow Table (Risk-Adjusted)

	INITIAL	YEAR 1	YEAR 2	YEAR 3	TOTAL	PRESENT VALUE
Total costs	\$0	(\$648,381)	(\$1,012,072)	(\$828,978)	(\$2,489,431)	(\$2,048,684)
Total benefits	\$0	\$4,178,655	\$9,331,920	\$10,570,950	\$24,081,525	\$19,453,219
Net benefits	\$0	\$3,530,274	\$8,319,848	\$9,741,972	\$21,592,094	\$17,404,535
ROI						850%
Payback period						<6 months

JetBrains IntelliJ IDEA: Overview

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

IntelliJ IDEA, JetBrains' flagship Java IDE, provides high-class support and productivity boosts for enterprise, mobile and web development in Java, Scala and Groovy, with all the latest technologies and frameworks supported out of the box. Every aspect of IntelliJ IDEA is specifically designed to maximize developer productivity. Together, powerful static code analysis and ergonomic design make development a productive and enjoyable experience.

After IntelliJ IDEA has indexed your source code, it offers a blazingly fast and intelligent coding experience by giving relevant suggestions in every context: instant and clever code completion, on-the-fly code analysis and reliable refactoring tools. Mission-critical tools such as integration with version controls systems and a wide variety of supported languages and frameworks are at hand — no plugin hustle included. While IntelliJ IDEA is an IDE for Java, it also understands and provides intelligent coding assistance for a large variety of other languages such as SQL, JPQL, HTML, JavaScript, and more, even when the language expression is injected into a String literal in your Java code.

Every aspect of IntelliJ IDEA is designed with ergonomics in mind. IntelliJ IDEA is built on a principle that every minute a developer spends in the flow is a good minute, and things that break developers out of that flow are bad and should be avoided. Most of the time the editor (and the code) is the only thing visible on the screen, and the developer doesn't need to leave it to do something that isn't coding-related. IntelliJ IDEA offers dedicated keyboard shortcuts for nearly everything.

Key features

- › Smart code completion
- › Data flow analysis
- › Cross-language refactorings
- › Inspections and quick-fixes
- › Shortcuts for everything
- › First-class support for top frameworks
- › Rich plugin ecosystem

Built-in tools

- › Version control (Git, etc)
- › Build tools (Maven, Gradle, etc)
- › Test runner and coverage
- › Decompiler
- › Database tools
- › Terminal
- › Docker

Enterprise frameworks

- › Spring
- › Java EE
- › GWT / Vaadin
- › JBoss
- › Play
- › Grails
- › App Servers / Clouds

Mobile development

- › Android
- › PhoneGap / Cordova / Ionic
- › React Native

Web development

- › JavaScript / TypeScript / ES6 HTML / CSS
- › AngularJS
- › React
- › Webpack / NPM
- › Node.js

Appendix A: Total Economic Impact

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.

Total Economic Impact Approach



Benefits represent the value delivered to the business by the product. The TEI methodology places 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.



Costs consider all expenses necessary to deliver the proposed value, or benefits, of the product. The cost category within TEI captures incremental costs over the existing environment for ongoing costs associated with the solution.



Flexibility represents the strategic value that can be obtained for some future additional investment building on top of the initial investment already made. Having the ability to capture that benefit has a PV that can be estimated.



Risks measure the uncertainty of benefit and cost estimates given: 1) the likelihood that estimates will meet original projections and 2) the likelihood that estimates will be tracked over time. TEI risk factors are based on "triangular distribution."

The initial investment column contains costs incurred at "time 0" or at the beginning of Year 1 that are not discounted. All other cash flows 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 in 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.



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.



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.



Return on investment (ROI)

A project's expected return in percentage terms. ROI is calculated by dividing net benefits (benefits less costs) by costs.



Discount rate

The interest rate used in cash flow analysis to take into account the time value of money. Organizations typically use discount rates between 8% and 16%.



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.

Appendix B: Endnotes

¹ Source: Forrester Analytics Global Business Technographics Developer Survey, 2018.

² Ibid.