

CREATING
CUSTOM
SOFTWARE
SOLUTIONS -

THAT THRIVE BEYOND LAUNCH





CONTENTS

Introduction	3
Choose your partner	8
Plan your project	13
Development and implementation	19
Post launch	30
Summary	32
Case studies	33

INTRODUCTION

A SOLUTION LIKE NO OTHER.

Custom software development is the process of designing, creating, deploying and maintaining your own software solution from the ground up. Unlike commercial off-the-shelf software, these solutions have their own specific set of requirements based on a range of variables, like the challenge you're trying to solve, the service you're trying to provide, the industry you operate in and even the profile of your potential users. In fewer words, a custom solution is exactly that – custom. Designed to your exact specifications with the aim of achieving your unique goal.

Perhaps the most obvious advantage of this is the full control it gives you over every element of the solution itself; like the features it provides, how it functions, what it looks like and everything in between. In comparison, a lot of (if not all) these aspects may be limited when using an off-the-shelf solution.

So why would anyone go with off-the-shelf? It comes down to what exactly you need your solution to do and how important tailoring is. Before building the next industry-shattering mobile app, you need to make sure that an existing solution isn't delivering more than 80% of the required functions, along with the ability to white-label it. In this case, buying off the shelf might make more sense.

Once you've identified that your unique requirements definitely warrant a custom build, you can start getting excited about creating your new software solution. To get started, you need to decide how you're going to make it happen, and you have two options; build the new solution in-house with the teams you have, or outsource the work to a specialist custom software development partner.

INTRODUCTION

DO I REALLY NEED TO OUTSOURCE?

Looking internally to build a new software solution might seem logical if you have a development team on-hand, but it isn't always the best way to go. Consider whether you have the technical expertise, the right infrastructure and the available scope to complete the project in-house. If not, this is where outsourcing not only makes sure your project gets off the ground, but also brings unique and profitable benefits.

The demands of correcting the shortcomings previously mentioned versus the increased level of production an external partner brings means outsourcing will not only save costs, but also give your team more capacity to focus on their core competencies.

Outsourcing can provide a competitive advantage by minimizing costs, enhancing customer service and improving product quality.

Teaming up with an external software development partner helps you create a higher quality solution for a number of reasons:

Technical skills – External developers bring the latest technical expertise and are fluent in the latest methodologies to get the best performance from the solution. For you, that means no time or budget investment in training your current team.

Diverse experience – Working with different clients and sectors provides them with a unique pool of experience, which can be continually drawn from to overcome challenges and identify best courses of action.



INTRODUCTION

Better tech – They are not tied to a particular platform or outdated system, meaning they have more choice when it comes to building the architecture of your solution. This means the foundations of your solution can be optimised for both performance and future scalability, regardless of your current systems.

“Outsource partners can leverage their collective experience to overcome difficult challenges. Working with many companies on a variety of projects, they can interject critical skills and expertise to any product or software development.”¹

Enhanced security – Handling the data and intellectual property of a roster of clients, they can't afford to be lapse when it comes to security. Your solution and the data it uses will be as safe as possible using the latest code, processes and technology.

¹ <https://www.salesforce.com/blog/2016/09/how-to-successfully-outsource-product-management.html>

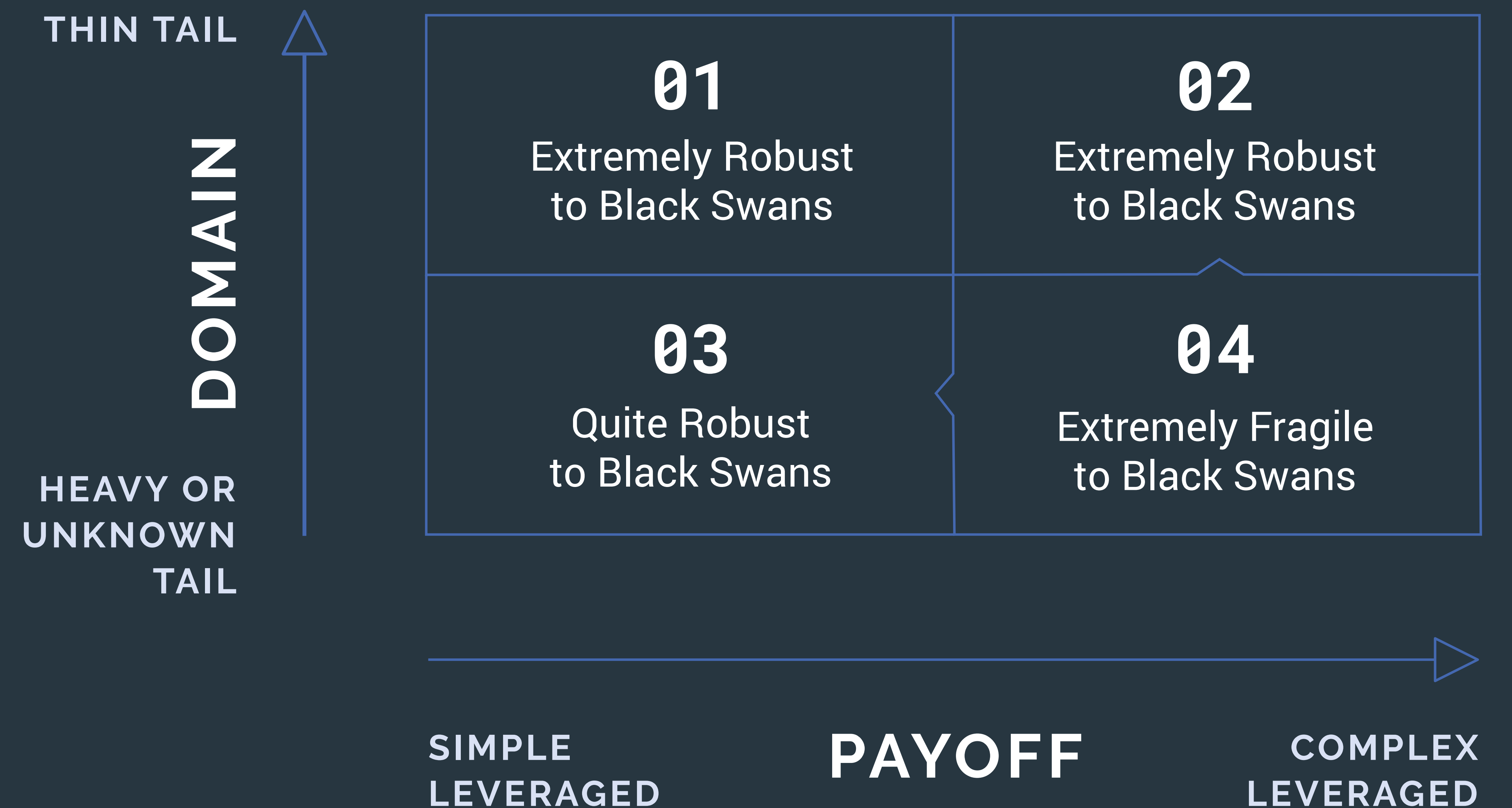
INTRODUCTION

AVOIDING THE 'BLACK SWAN'

Considering the benefits previously mentioned, outsourcing could well be the best way to create your next custom software solution. But whichever way you go, creating a new solution from scratch takes a lot of talent, expertise and time. You need to do everything you can to avoid the project becoming a “Black Swan”.

Popularized by Nassim Taleb, “Black Swan Theory” is a metaphor used to describe an unpredictable event that brings significant negative impacts, particularly ones that seem avoidable in hindsight. In the software development community, the term is often conversationally applied to failed projects – especially those that had absurdly short timeframes and overly complex or unclear objectives.

DECISION THEORY OF ANTI-FRAGILITY



INTRODUCTION

Though Black Swans – by their very definition – may be unexpected, they are far from uncommon when it comes to software:

75%

of IT executives admit that their software projects are “doomed from the start”.¹

² <http://www.geneca.com/75-business-executives-anticipate-software-projects-fail/>

WHY?

Simply put, there are so many moving parts to get right, resulting in a practically unlimited number of variables that can cause failure. Some of the most common reasons include:

- **Not enough time** – Unrealistic or random deadline with no data or programmer knowledge to support the decision.
- **Inadequate planning** – Underestimating the true time and cost demands, or not initially gaining an executive sponsor to see the project through to completion.
- **Unclear requirements** – Not outlining the project specs, considering future needs or focusing on required business outcomes.
- **Too many people** – Having large project teams that compromise communication or mixing code from too many developers.
- **Lack of testing** – Sacrificing testing because of time constraints, leading to functionality and security issues.
- **No end user involvement** – Failure to consider and test from a user perspective as well as a functional one.

Though the reasons for failure are plenty, they are also well known and completely avoidable. There are three main areas you can focus on to significantly mitigate the chances of failure; the partner you choose, how well you plan, and the methods and technologies used in development and implementation. Assuming you’ve decided to outsource development, this guide will help you ensure your next solution not only makes it to launch, but also flawlessly performs the functionalities you need it to.

CHOOSE YOUR PARTNER

SIGNING UP A SOFTWARE DEVELOPMENT PARTNER

The best custom software development partners don't just take a brief, disappear and come back later with a finished solution. They work collaboratively with you throughout the entire process, making incremental changes that allows the solution to become the best it can be.

Once you've made the decision to outsource the development of your custom software solution, it's extremely important to invest some time and effort in deciding who to go with.

After all, this is the team you'll spend considerable time collaborating with throughout the life of the project. You'll work together to make key decisions, find ways to overcome challenges and keep the overall progress tight and on track. For these reasons, it's best to think of the outsourced developer as a technical partner rather than a vendor or supplier that you're simply paying for a service.

INTELYGENZ - CUSTOM SOFTWARE SOLUTIONS THAT THRIVE



SHORTLISTING THE BEST

1. THEY HAVE STRONG BUSINESS SKILLS

A good software development partner will excel in all areas of programming. A great one will also have strong commercial awareness that allows them to understand your goal from a business perspective and create the required functionalities to achieve it.

2. THEY USE THE AGILE

The Agile methodology is a collaborative approach that brings you closer to the incremental development process and allows high quality solutions to be delivered faster and more flexibly (more on this later).

3. THEY'RE NOT CHEAP

Creating custom software takes time, management and expertise. Any partner trying to win on cost can only do so by making sacrifices somewhere. Often, this happens in testing, compromising quality, functionality and security.

4. THEY CONTINUALLY EVOLVE

Software development is a rapidly changing discipline that's always being shaped by new technologies and methodologies. The best partners avoid becoming stagnant, and instead are continually learning and adopting cutting-edge tools and state-of-the-art techniques.

5. THEY HAVE VARIED EXPERIENCE

The more diverse a partner's client roster and portfolio, the more likely they'll have worked on similar projects and solved challenges you would never think of internally.

6. THEY'RE RELIABLE AND CREDIBLE

Partners who can prove their reliability and credibility have normally been in business for some time, will pass any extensive background checks you make and would be happy to put you in touch with previous clients. Always do due diligence in terms of evaluating the history of the company and any legal implications that a partnership would bring.

SHORTLISTING THE BEST CONT'D

7. THEY'RE OPERATIONALLY COMPATIBLE

With today's technology it's not impossible to work with a software development partner on the other side of the planet. Since communication is so important to the process, minimize language and time zone barriers as much as possible.

8. THEY'RE PREPARED FOR POST IMPLEMENTATION

You'll likely work with your partner long after launch. So make sure you choose someone who has a proven model that covers successful validation, maintenance, training and optimization.

9. THEY'RE A GOOD CULTURE FIT

How well you gel with your partner will depend a lot on your work cultures. Try to find a company that shares similar performance standards, values and working practices for maximum efficiency.

10. THEY RETAIN THEIR CLIENTS

Long standing client relationships are a key sign that a partner not only delivers high quality solutions, but also are easy to work with and provide invaluable ongoing technical support. This also shows they value collaboration, which is vital for the success of any development project.

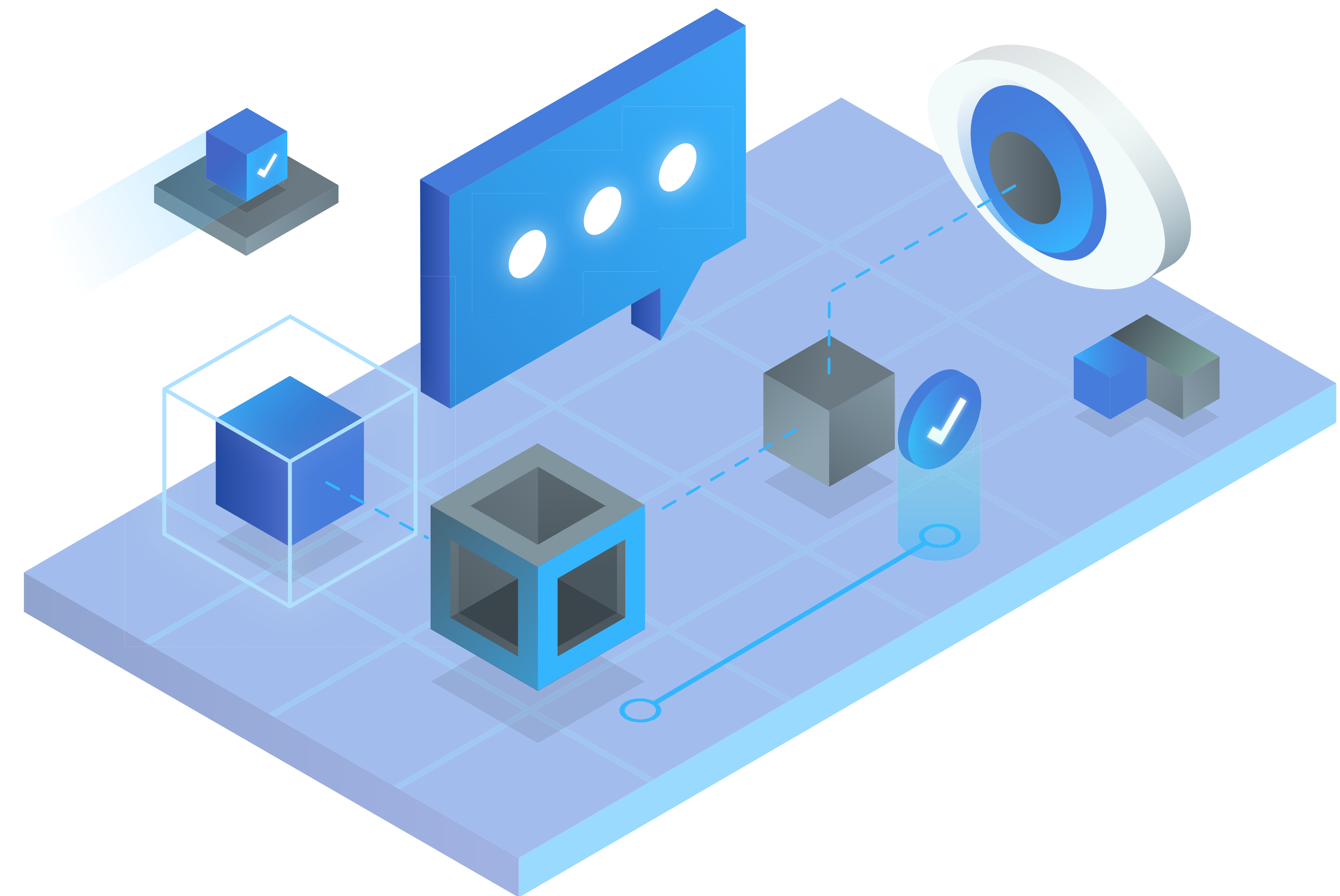
CHOOSE YOUR PARTNER

BEFORE YOU APPROACH

We've delivered some of our finest work to customers who were already able to clarify a few things when they first approached us. To determine your perfect partner as efficiently as possible, follow these three tips:

DEFINE YOUR TOP-LEVEL GOAL

If you've decided to create your own software solution, you should already have a good idea of the main objective you're trying to achieve. Whether it's addressing a customer pain point or a whole new experience that will change how customers interact with your company, make sure you can define this as clearly as possible. This helps potential partners to elaborate on how they will help you achieve it.





SET REALISTIC EXPECTATIONS

Unrealistic expectations are pretty much a guaranteed path to failure. And expectations start right at the beginning, before you even have a partner in place. You need to think of how you'll measure the performance of the solution, what KPIs you expect it to hit, the time frame you're expecting to build within and of course, budget. Try to get input from internal stakeholders on these things. If this is the first time you're outsourcing custom software development, defining this could be tricky. But doing some research and setting some loose expectations is better than having none.



DID YOU KNOW?

14% of project managers and 16% of developers say unrealistic expectations was the reason their software project failed.³

CONSIDER TECHNICAL REQUIREMENTS

If you have a good level of technical knowledge, it's worth exploring the requirements that creating your custom solution might entail. Think about your existing systems, what integrations you might need or what skills will be required during development. Though you can rely more on the partners to have this knowledge, considering it before approaching can help you rule out those that simply can't deliver from a technical perspective.

³ <https://dzone.com/articles/interesting-facts-about-software-development-stati>

PLAN YOUR PROJECT

PLAN YOUR PROJECT

With your new software development partner on board, it's time to start putting together a flawless project plan. This is absolutely critical to ensuring the project stays as close to your timeframe and budget as possible – all while producing the highest quality final output. Because of the complexity and unique challenges of custom software development, this plan needs to be both detailed and flexible.

There will be changes and decision-points that you simply can't predict, which is why choosing a technical partner that uses agile is so important.

Their knowledge and expertise will help you expect the unexpected and their incremental development approach will keep you regularly updated, allowing you to continually feedback as the solution takes shape.

If you don't invest the time to come up with a proper project plan, or if you create a plan that's too rigid, you're destined to go massively over time and budget, the quality of the solution will be sacrificed or it won't even make it to launch altogether.

INTELYGENZ - CUSTOM SOFTWARE SOLUTIONS THAT THRIVE



PLAN YOUR PROJECT

PLAN YOUR PROJECT

1 in 6 IT projects go

200%

over cost and



70%

over schedule⁴

⁴ <https://hbr.org/2011/09/why-your-it-project-may-be-riskier-than-you-think>

DEVELOPING A BRIEF THAT ENSURES SUCCESS

Before you start putting your plan together, you need to create an initial brief to guide the development of your solution towards its final form. This gives everyone involved a clear target to aim for, solidifies expectations and acts as a continuous point of reference as the project evolves.

If you haven't already, start by defining the market your solution will enter, as well as the users/customers that drive it.

ASK YOURSELF:

- **Is there a demand for the solution I want to create?**
- **What specific need/pain point(s) will it address?**
- **What will make it different to others on the market (why will people choose it)?**
- **How will it make/save my business money?**
- **How will you make it sustainable and scalable in the future?**

Answering these questions will help you set crystal clear goals and objectives – the most important part of the brief. Think about exactly what you need the solution to achieve from a business perspective. Whatever it is, make sure it's well defined and measurable. If you have multiple goals, prioritise them in order of importance and set KPIs for each.

ONLY 55%

of IT executives feel that the business objectives of their projects are clear.⁵

⁵ <https://www.geneca.com/why-up-to-75-of-software-projects-will-fail/>

You might already have some ideas for features that you'd like to see in the final solution. Add these to the brief and try to determine which are must-haves and which are nice-to-haves. This helps the team maintain realistic expectations as the project progresses. Remember that coming up with new requirements later on will cause delays, so try to nail this down as early as possible. If you can at this point, try to define the Minimal Viable Product (MVP).

An MVP is an Agile Development term used to outline a product with just enough features to satisfy early adopting customers and generate feedback for future improvements and additions.

With these elements in place, you can start to plan the delivery stages; defining what needs to be delivered and when. This should include room for plenty of incremental feedback and stages of testing that will allow the development team to iron out any technical issues. Your software development partner can help you make sure the delivery plan is realistic and achievable. They'll have their own workflows and systems in place, which will dictate what's possible from a scheduling point of view.

PLAN YOUR PROJECT

LAWS AND REGULATIONS

One last thing to consider is if there are any industry laws and regulations that the developing partner needs to adhere to. For example, are there limitations on how you acquire and handle data? The earlier you can make your partner aware of this, the less likely you are to run into any breaches down the line.

INTELYGENZ - CUSTOM SOFTWARE SOLUTIONS THAT THRIVE



ASSEMBLE THE PERFECT TEAM

The last part of putting together your master plan is to assemble the team that will be executing it. Though you are already best positioned to do this, stick to these top tips to maximize efficiency:

ONE SPONSOR

Assign a single senior project sponsor to make key decisions, keep the project progressing in a steady direction, appoint the team, oversee duties and prevent bottlenecks.



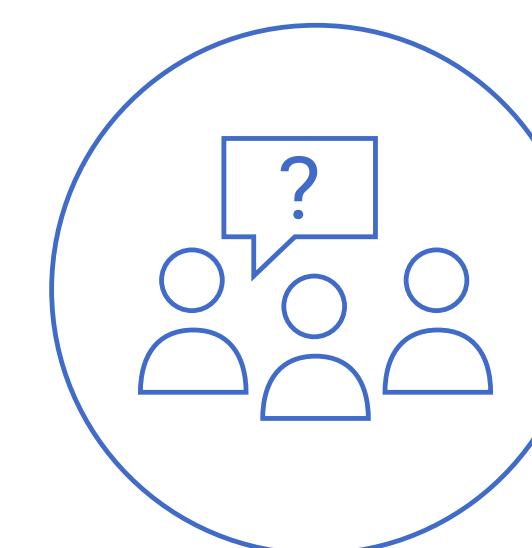
CORE TEAM

Too many people on board drives costs up, adds confusion and slows progress. Be ruthless and objective when picking your team and stick to the minimum number of people it will take to get the project done.



CLEAR DUTIES

Map the process across key members of the team and make the day-to-day tasks and responsibilities of each person as clear as you can.



ONE UNITED TEAM

Consider your software development partner a member of your internal team. This helps you maintain the closest relationship and ensures clear communication between all parties.



PICKING THE RIGHT METHODOLOGY

So you have your technical partner signed up, an air-tight brief to guide the project, a flexible yet detailed plan laid out and the dream team to make it happen... it's time to finally start executing.

When it comes to developing custom software, there are many different philosophies, approaches, processes and technologies that a team can use to create the final solution. Though you may not be doing any development in-house, it's useful to have some base knowledge on the process from the point of view of your software development partner.

This is because firstly, the best methods in software development are always being evolved, refined and combined with others, so knowing what to look out for comes in handy for any future projects. Secondly, most of today's best-in-class approaches are based heavily on collaboration, meaning you'll be kept close to the development process throughout the project and will need to know what's going on.

To help you understand the basics, here is an overview of today's leading methodologies, as well as the reasons why they are super effective.

LEADING METHODOLOGIES

BASIC SDLC

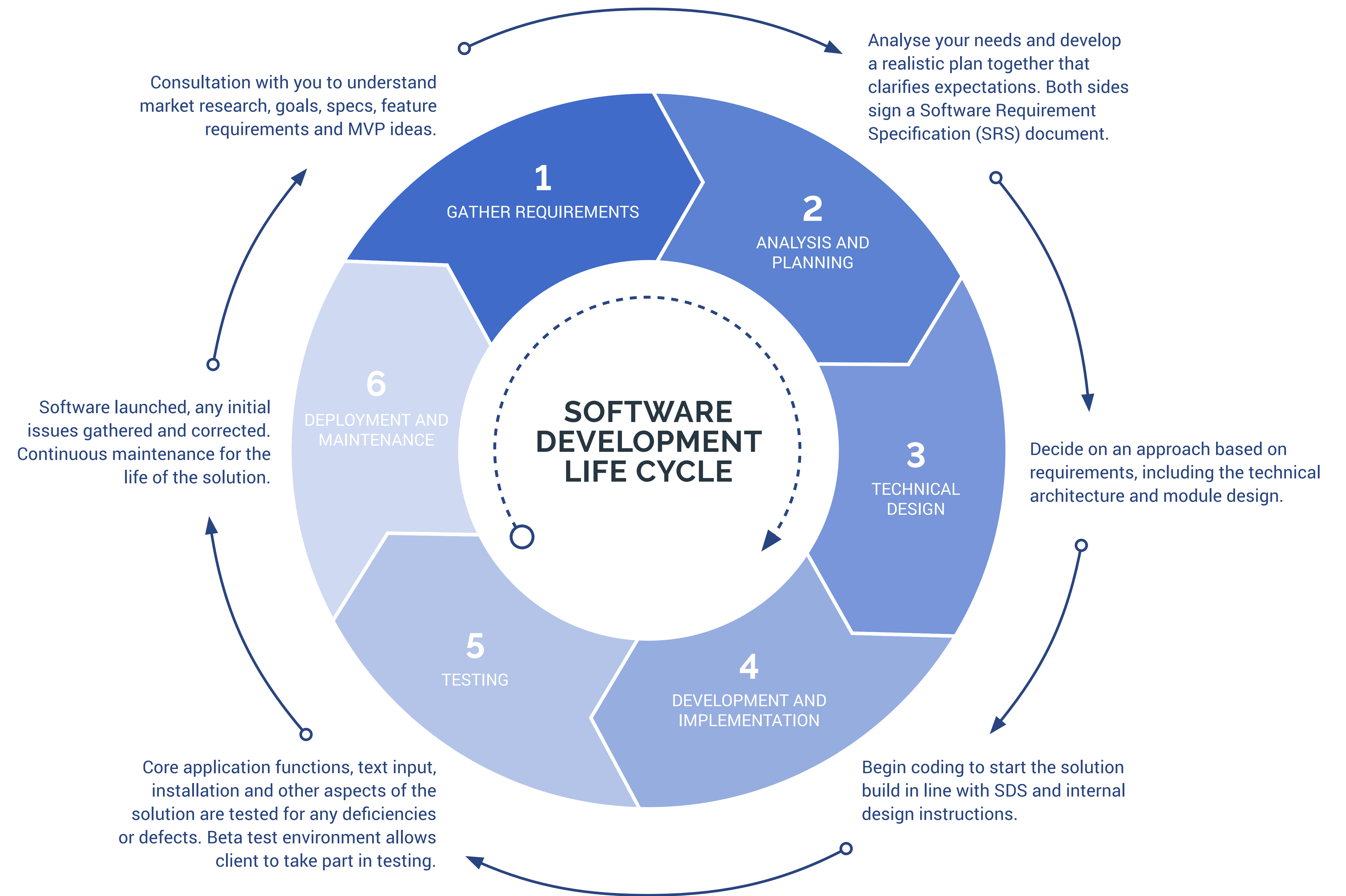
UNSTOPPABLE AGILE

TESTING WITH ATDD

CULTURE & DEVOPS

BASIC SDLC

Before diving into specific approaches, let's take a look at a basic and traditional Software Development Lifecycle (SDLC). Many of today's various approaches are evolutions and refinements of this top-line 6 stage sequence.



UNSTOPPABLE AGILE

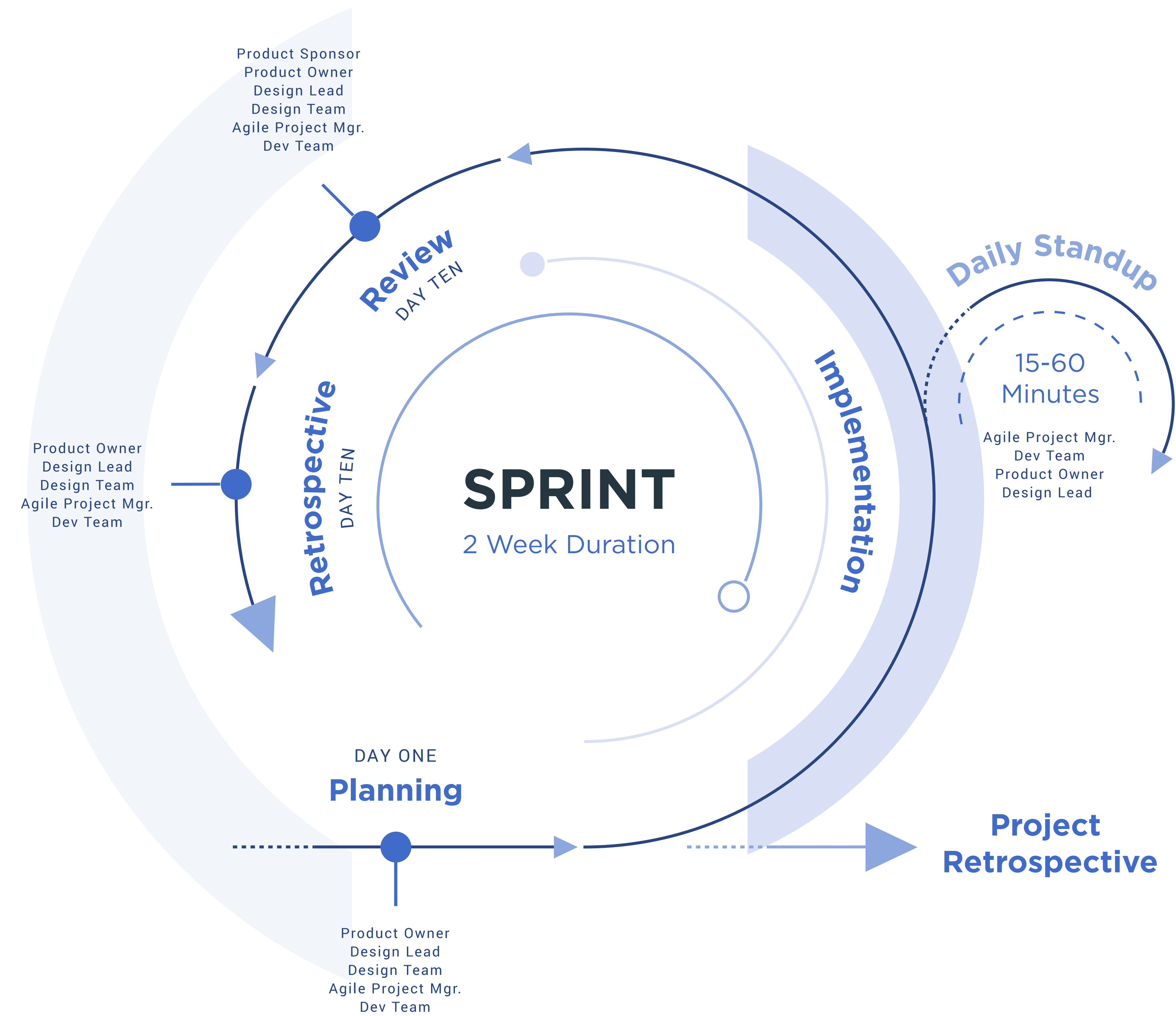
As we've touched on already, the Agile methodology is a flexible development approach built on collaboration. The process of an Agile project uses incremental stages, allowing for frequent feedback, as well as fine tuning and improvements as the solution evolves. In our experience working with different clients, we've found this benefits both the solution owner and the software development partner by reducing the chance of errors and delays, and rapidly speeds up the development process.

KEY BENEFITS:

- ✓ Faster project delivery
- ✓ Less bugs
- ✓ Higher quality final solution
- ✓ Shared accountability
- ✓ Less risk of going over time/ budget
- ✓ Improved efficiency
- ✓ See solution benefits sooner
- ✓ Continuous monitoring
- ✓ Transparency of progress

SDLC + AGILE

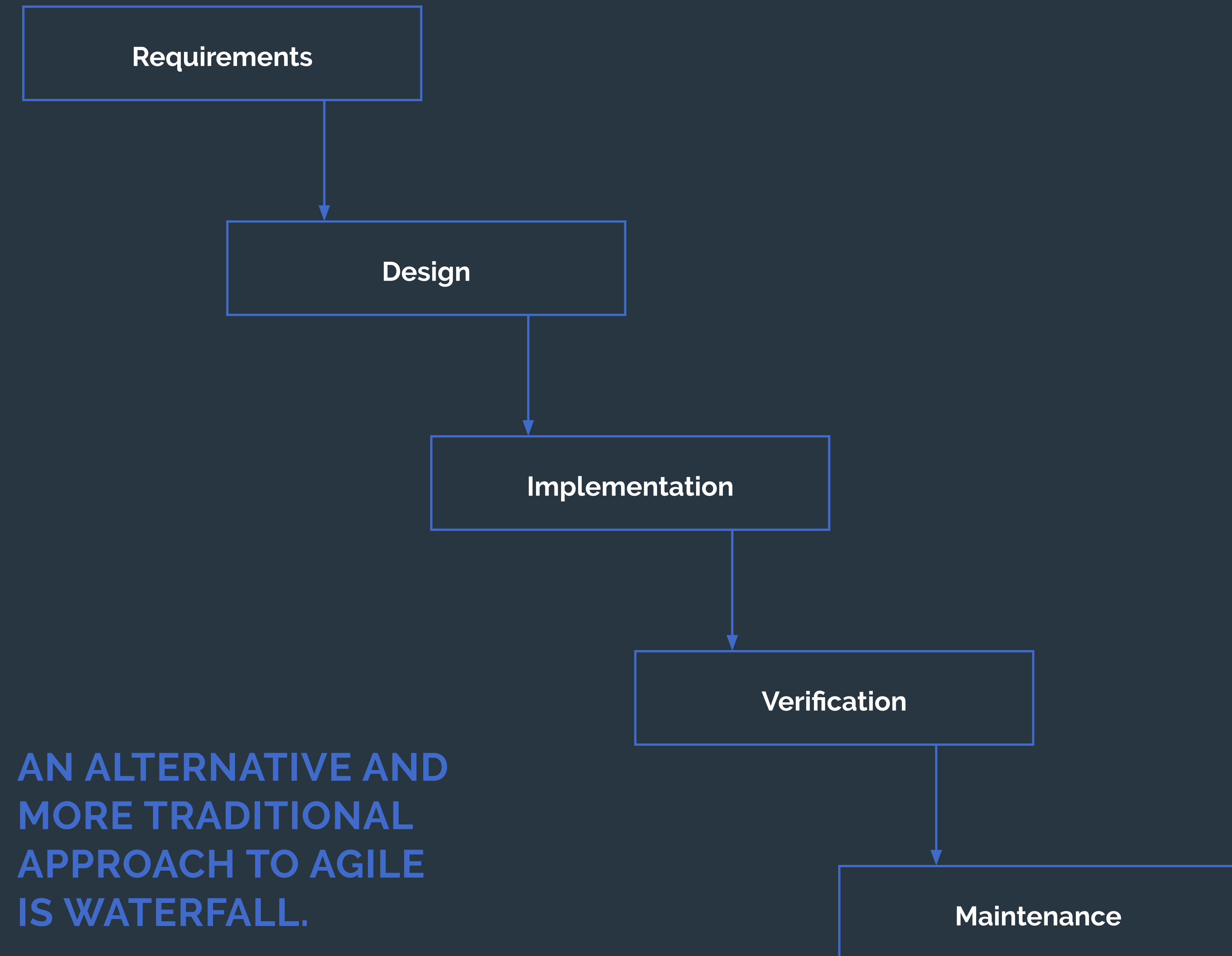
In relation to the basic SDLC above, Agile turns the last three stages into a cycle that's continuously repeated. Though there are many variants of specific workflows within the Agile methodology, the diagram to the right illustrates what one of our own 2 week sprints might look like:



WATERFALL

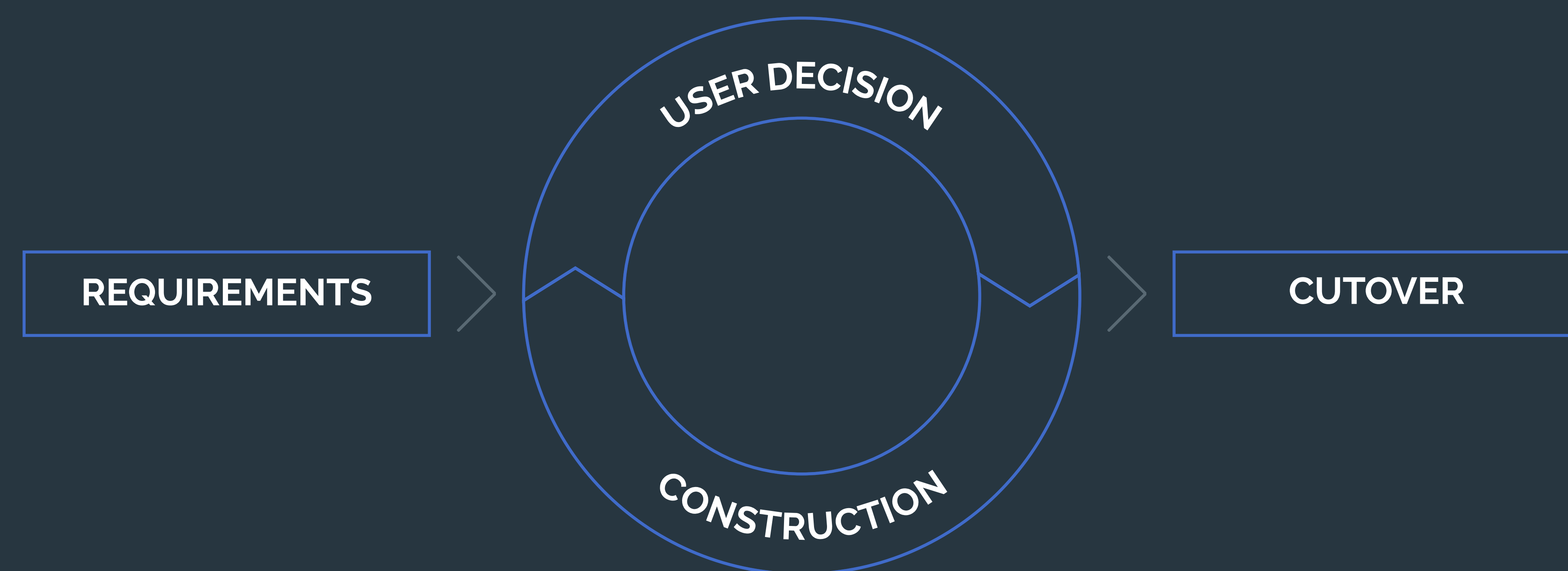
This is a much more linear process consisting of long phases, each of which must be 100% complete before the next can begin. Though this may be easier to understand than the Agile method, it is slower, more costly and fails to provide the incremental adjustments that give Agile projects their robustness. Waterfall is normally used for less complex, short-term projects that have a single end state delivery.

For the majority of projects, the benefits of Agile strongly outweigh those of the clunky, non-cyclical and dated Waterfall methodology. When it comes to success, this translates to a huge difference in results.



DEVELOPMENT AND IMPLEMENTATION

To provide one more comparison, Rapid Application Development (RAD) is another potential approach. This method is designed to deliver a high quality system at a lower investment cost. Using the four broad phases below, it allows developers to continuously adjust the solution based on shifting requirements.



The downside to RAD is that it demands an incredibly deep level of application knowledge from the solution owner as well as the development team. Without this, a condensed timeline and lower budget simply can't be maintained as changes are needed, because the understanding required for rapid decision making isn't there. RAD projects are only really suitable for solutions with very simple computational requirements, and many spiral out of scope due to knowledge gaps.

With several benefits over Waterfall and RAD, it's no surprise that today's cutting-edge development companies tend to use the Agile methodology. That being said, the best companies will pick an approach based on its suitability for that specific project. It just so happens that Agile is extremely versatile and is the most effective for the majority of custom software projects.

DEVELOPMENT AND IMPLEMENTATION

AGILE PROJECTS

2x

more likely to succeed

1/3

less likely to fail than
Waterfall projects.⁶

85%

of software developers use the
Agile methodology.⁷

⁶ <https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>

⁷ <https://insights.stackoverflow.com/survey/2018#development-practices>

TESTING WITH ATDD

One of today's most efficient and user-centric approaches to testing software is Acceptance Test Driven Development (ATDD). In this method, the development team creates and performs acceptance tests before writing any code. The diagram below shows a typical ATDD process in five steps:





DEVELOPMENT AND IMPLEMENTATION

TESTING WITH ATDD

Unlike traditional Test-Driven Development (TDD), where tests are created from a developer's perspective, ATDD takes on the end user's perspective to validate functionalities. This brings benefits to the solution owner, development team and the end user:

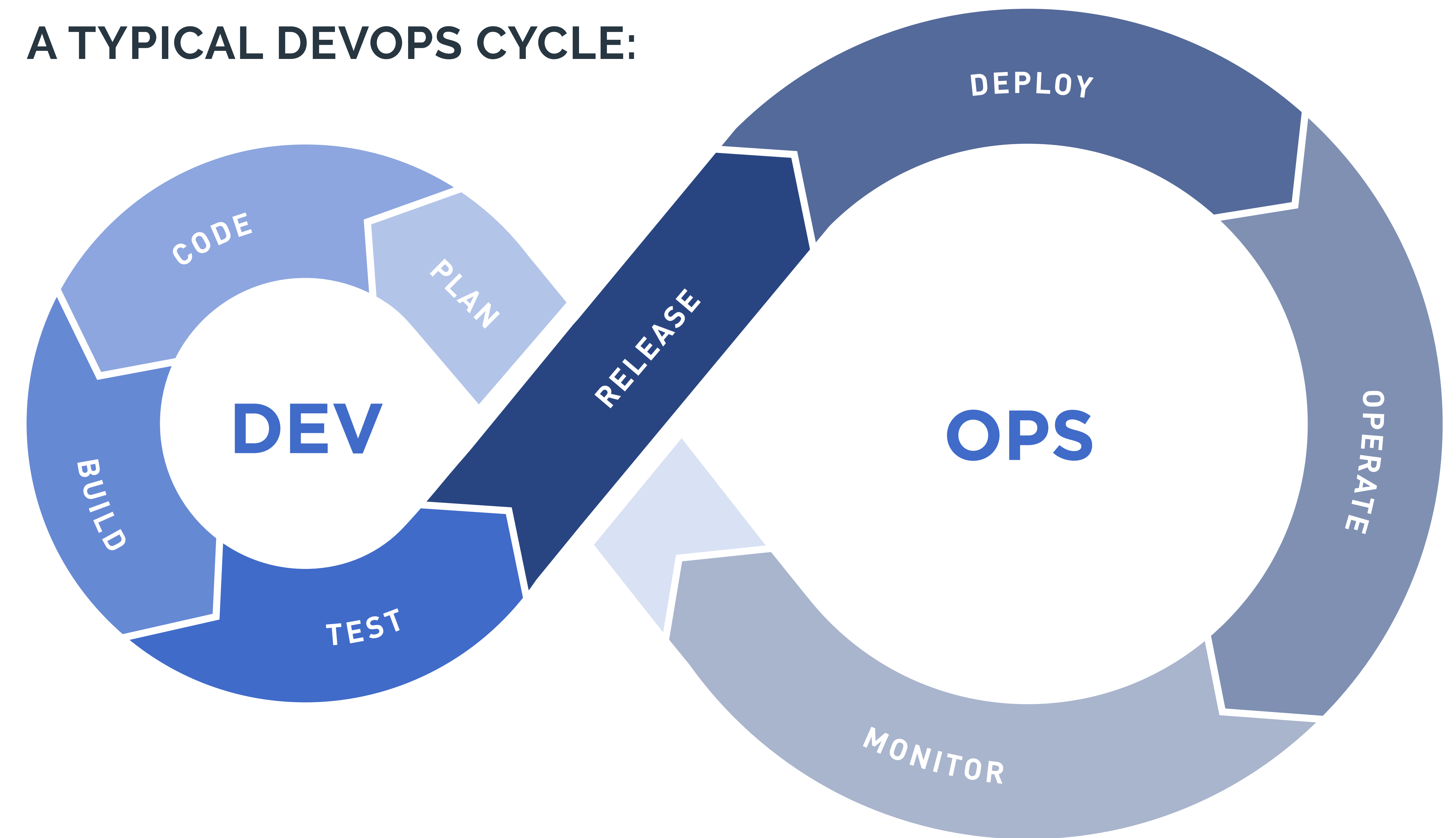
- ✓ Ensures a better user experience.
- ✓ Robust code quality.
- ✓ Low failure rates.
- ✓ Confidence in every release.
- ✓ Transparent progress in real time.

T E S T
C O D E
R E F A C T O R
R E P E A T _

CULTURE AND DEVOPS

In your search for a suitable software development partner, you'll likely come across the term "DevOps". This is a set of principles and a work culture that software companies use to unite internal development, IT operations and quality assurance teams. Using a range of methodologies, tools and practices, it is a collaborative approach that allows more robust, high-quality software to be delivered in less time.

A TYPICAL DEVOPS CYCLE:



CULTURE AND DEVOPS

It works by reducing the gap between development and operations, enabling continuous delivery and feedback, and optimizing quality throughout the development lifecycle. For you, this translates to time and cost savings, a more efficient project process and – most importantly – a better end solution.

38% of companies using DevOps report a higher quality of code production.⁸

The combination of Agile, ATDD and DevOps promises maximum speed, flexibility and quality throughout the development and implementation of a custom software solution. Best-in-class development companies use these approaches to not only optimize their processes internally, but to pass on the best benefits to both you and the end user.



⁸<https://www.upguard.com/blog/devops-success-stats>

POST LAUNCH

LAUNCH IS JUST THE BEGINNING

Creating custom software is not a simple case of completing a project and releasing it to the world. In fact, launch is just the beginning. You'll need to add continuous iterations that optimize its improvement, ensure its compatibility as IT systems evolve, fix errors as they occur and do whatever it takes to provide the best possible user experience.

Establishing a clear channel of communication and **inviting continuous feedback** from both internal and external users will flag everything from bugs to ideas for improvements. You might come across the need for a new functionality you never considered, discover unexpected compatibility issues or find new ways to extend the life of the solution.

The best way to maintain and improve your product is through user feedback.

MEASURING SUCCESS

With your new software solution live on the screens of your customers, you'll need to take a step back and determine its success. This is best approached by referring back to the KPIs set out in your original brief. Some of these may be time-sensitive, but even if it's a little early, you can still identify whether your solution is on track to achieve them.

As well as measuring how well your solution is performing against your expectations, it's also a good idea to reflect on the success of the project process. This will help you determine if the development company you worked with were a good fit and performed as you'd hoped. Remember to also be self-critical and look at the performance of your internal teams, as this gives you the opportunity to learn and improve for the future.

For every \$1 billion invested in the United States, \$122 million was wasted due to poor project performance.

ASK A NEUTRAL PROJECT MANAGER TO EVALUATE:

- **Project processes**
- **Technology used**
- **People in team**
- **Functionality of the solution**

Understanding how well the entire project went from start to finish is important, especially if your technical partner is helping you maintain and support your solution throughout its lifespan.

SUMMARY

RECAP

Avoiding a Black Swan and ensuring your custom software solution thrives beyond launch is a challenging yet immensely rewarding feat. Once you've invested your time, effort and attention in refining the best partner, plan and technical approach, you'll not only defy the high failure rate associated with IT projects, but also begin reaping the rewards of a high performing solution.

[Here is a quick recap of the advice covered in this brochure:](#)

1 OUTSOURCE FOR A COMPETITIVE ADVANTAGE

Outsourcing software development provides access to a higher level of expertise, a diverse pool of experience, the latest technology and methodologies, and enhanced security performance.

2 PARTNER WITH THE BEST

Take the time to find a collaborative, innovative software development partner with a strong track record, that fits well with your culture and operations. Refer to the top 10 signs of an awesome software development partner to help your shortlist.

3 CRAFT A BULLETPROOF PLAN

Start with a detailed brief that approaches the project from a business perspective; outlining the demand, the need/pain point addressed, how the solution will generate revenue and scale. Define your MVP and work closely with your software development partner to scope out the project. Finally, assemble a minimal core team with a project sponsor that can keep things moving forward.

4 TAKE AN APPROACH BUILT ON SUCCESS

Brush up your knowledge on the latest development approaches so you can work better with your software development partner. A powerful combination of approaches is the Agile Methodology, ATDD testing process and DevOps. This promises maximum speed, flexibility and quality throughout development and implementation.

5 MAINTAIN EFFORTS BEYOND LAUNCH

Launch is just the beginning. You'll need to continually optimize improvement, ensure compatibility as IT systems evolve and fix errors. Use internal and external user feedback to enhance the solution and refer back to your initial KPIs to measure project success. Consider the performance of both your team and software development partner.

C A S E S T U D I E S _

Here are two examples of businesses we've collaborated with to successfully launch custom software solutions. As with all of our projects, these were delivered in line with our three cornerstone approaches: Agile, ATDD and DevOps.

CASE STUDY

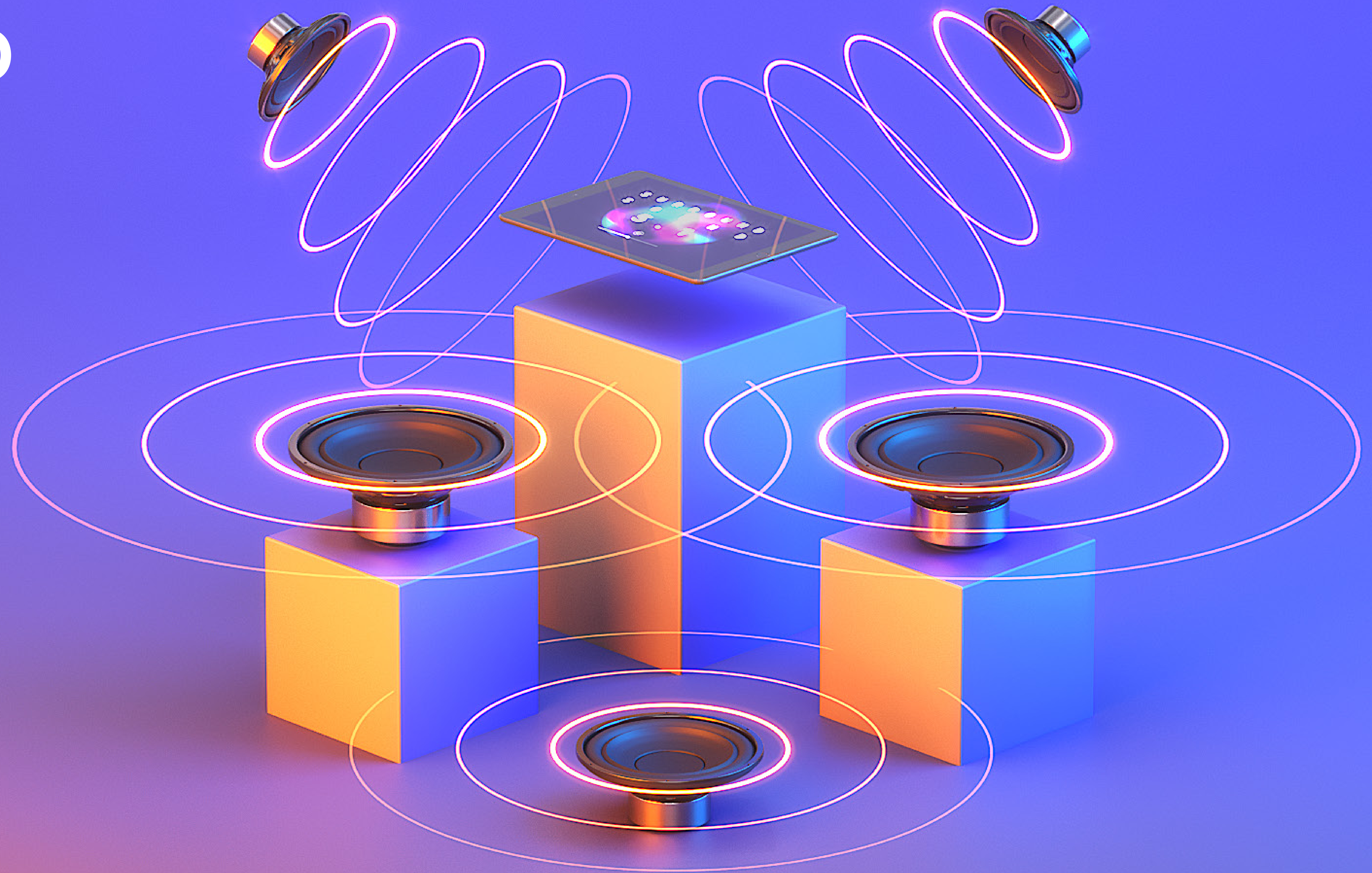
ENDLESS SOUND

! THE ASK

An international audio technology company that produces a vast array of sound solutions – from telephone and home cinemas to professional studio suites – wanted a practical way to showcase the sound quality of their products.

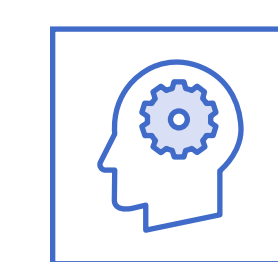
⚙️ WHAT WE DID

We created a software solution that connects their CMS to a mobile app with a built-in video player capable of playing different sound codecs. The company's employees can now pre-load an existing video into the app, and give customers a comparative example of how it sounds on different products.





CASE STUDY: ENDLESS SOUND



THE PROCESS

This was a BackOffice and mobile app integration, with the added ability to cast to Google Chromecast.

In the backend, we created a React+Bootstrap application that simplifies the creation of demo campaigns for customers, setting default player parameters for each media file and presenting all of the different capabilities they have for processing sound.

The mobile application presents content related to a particular campaign and integrates internally with their SDK, showcasing the full spectrum of their current capabilities. This allows the user to switch products in real time by applying sound filters and processors that enhance the quality of sound on the phone or tablet they are using for the demo.



SUMMARY

The application gave our client the capability to demonstrate its many sound products easily to customers. Now, users can switch combinations of sound products on and off for instant comparison by playing the video content of their choice.

With built-in market survey functionality, the application also allows the company to survey users on new sound products easily, and with rapid results.

INSTANT SOUND COMPARISONS

CASE STUDY

FREE YOUR FINANCES

Evolving international mobile banking to fill a gap in the market

! THE ASK

An entrepreneurial bank spotted an opportunity in the US banking industry: they wanted to fill a gap between consumer banking and large enterprise banking services to better serve the growing market of freelancers and small companies. Supported by the bank's mothership, a spin-off team of entrepreneurs set about building a branchless bank – that means, entirely digital – from scratch, focusing on providing valuable services for freelancers and small companies. Their model was a fee-free financial service that would help small businesses to grow.

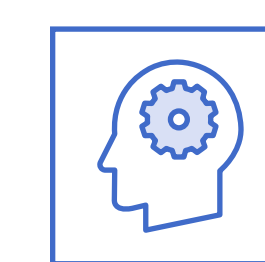
⚙️ WHAT WE DID

Working with the spin-off team, we defined a technical architecture that supported the bank's vision. Our approach allowed rapid market deployment of an MVP banking service, with real-world testing of the product market to guide its direction. The bank is now far beyond MVP – we continue to build and maintain the platform in production for thousands of customers today.





CASE STUDY: FREE YOUR FINANCES



THE PROCESS

Combining flexibility and high levels of automation with security and deployment were crucial. The architecture had to match the ability to offer a fee-free bank, something that could only be achieved using significant automation and a resilient platform for low maintenance costs.

We built and integrated an entire back-end API banking suite. The back-end attaches the front-end offerings to a core banking infrastructure that underpins all services. It features integration with multiple third-party providers who are used to automate security, smooth the onboarding process, and add a range of digital banking features.



SUMMARY

Over ten thousand freelancers and small business owners now bank on their terms with the spin-off's web and mobile app, integrating with dozens of external payment and financial services providers. They can make bank-to-bank transfers, send and receive instant payments, pay bills, mail checks, and more – all digitally, all without fees.

10,000+

Entrepreneurs banking on their own terms

AUTOMATE EVERYTHING

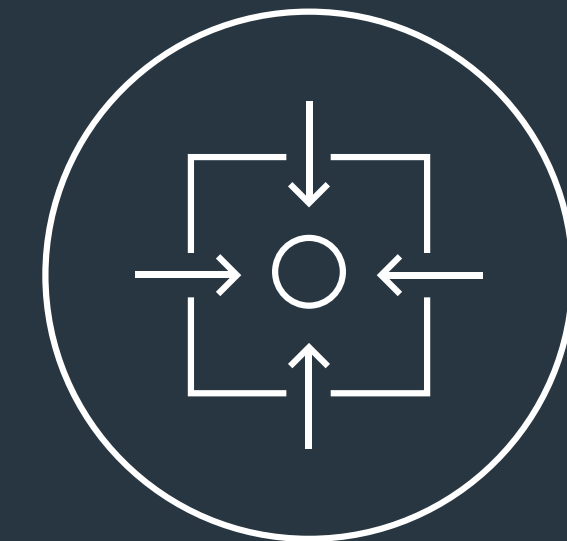
WHAT WE DO

At Intelygenz, our development approach optimizes speed, flexibility and above all quality, helping you build a custom software solution that delivers what you need it to.



APIS

Proven, complex, tested APIs. Following market standards for easy integration.



MONOLITH TO MICRO SERVICES

Modular software for high availability and resilient services.



MACHINE LEARNING

Artificial Intelligence as a part of your software.



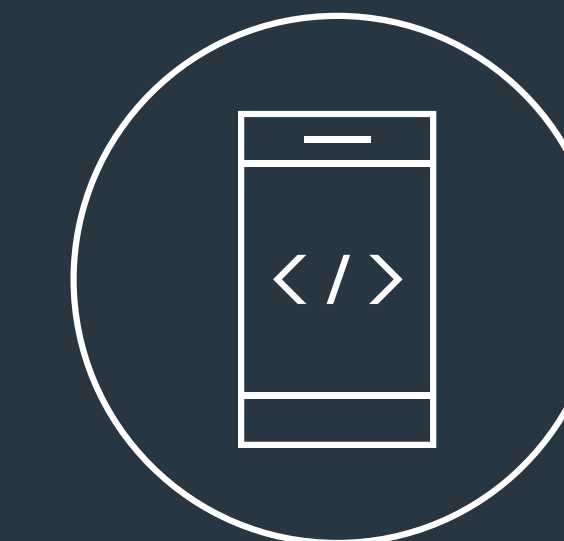
MAINTENANCE & EVOLUTION

Support services, upgrades and product evolution, all included.



INFRASTRUCTURE AUTOMATION

Environment and tools automation driven by versioned infrastructure definitions.



MOBILE APPS

Native and hybrid iOS and Android apps.



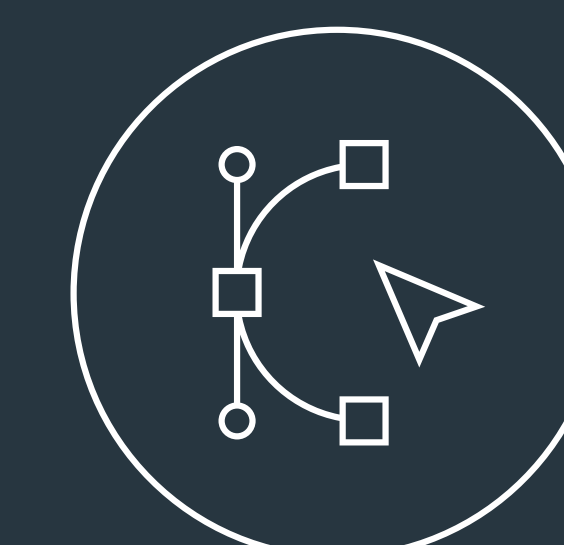
WEB APPS

Leverage browsers capacities to maximize user experience.



CLOUD SERVICES

Cloud architectures for blazing fast distributed systems.



PRODUCTS FROM SCRATCH

Product roadmaps and incremental product value focus.

DEVELOP YOUR ORIGINALITY

Get a complimentary Intelygenz consultation for your next custom software project and we'll talk you through the many ways our software services can help you hit that next big strategic goal.



FOR MORE INFORMATION PLEASE GET IN TOUCH

+1 (415) 917-1500

541 Jefferson Ave, Suite 100
Redwood City, CA, 94603
info@intelygenz.com

+34 915 35 96 12

Plaza Sta. M^a Soledad Torres
Acosta 2, 5^oC, Madrid, 28004
info@intelygenz.com

