

Software Development Done Right.





Successful Software Development, Explained.

No two software projects are alike, but all the successful ones share certain principles.

Here are a few things to keep in mind at every step -- from the preparation phase, to the choice of a development partner, to post-delivery maintenance -- to ensure that your development project gets successfully completed on time and on budget, and fully meets the expectations of the software's ordering customer and users.

1

Defining business goals

Do you seek to boost productivity? cut costs? improve customer satisfaction? increase sales? retain clients? Clearly defined business goals will guide the development team in formulating a delivery strategy, prioritizing requests, setting the criteria for success and creating a roadmap for the long-term. Defining your business goals will also make it possible to evaluate the worth of a project's various components and assess the project's return on investment (ROI).



2

Drawing up your budget

The wider the gap between what you want and what you can spend, the lower the chances of success – and we all know that resources are always scarce. That is why it is important to know a project's ROI before setting the budget. When funds are scarce but the net ROI of your software project is positive, one option is to seek bank financing. ROI calculations should be automatic for every project: without it, it is impossible to decide whether an investment is worth it. Remember to be as realistic as possible when calculating the ROI.

Ideally, you should have a preliminary budget in hand when you contact your technology partner. Your partner will be able to tell you, very quickly, whether your expectations are realistic or not, based on how much you can invest. Your technology partner may also suggest alternative approaches. If the budget is set in stone, there will be choices to make. Is software that is within budget but only meets 80% of needs acceptable? Only you can decide that.



3

Spelling out your expectations

Stating your expectations is one of the most critical steps in a software project. There can be no glossing over or jumping to conclusions. Each point should be discussed and analyzed in detail with everyone involved. The goal here is to develop specific and detailed specifications for the application. In “Waterfall” mode, setting things out precisely is of the utmost importance. In “Agile” mode, it is the vision of the project that is detailed, which saves time, as goals are not meticulously defined in advance only to fall by the wayside later.

This is also the step at which you separate the “must have” from the “would be nice”. Expectations are also prioritized according to business goals, to determine what the minimum viable product (MVP) would be.

Expectations can be set with your technology partner. In fact, it’s a good idea to involve your technology partner at this step, to ensure you share a common understanding as early on in the process as possible, agree on a strategy, and lay the groundwork for the kind of high quality cooperation needed to ensure that the project goes smoothly.

Minimum viable product (MVP)

A product that has the functionalities required to meet the needs of first users, test certain hypotheses and generate comments that can be used for future development or for market launch.

4

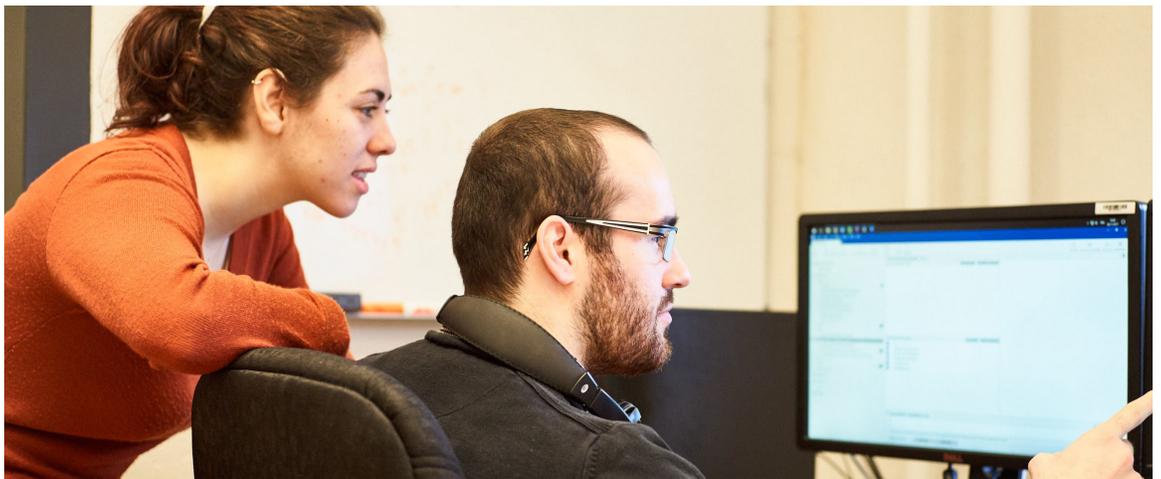
Choosing your development partner

Price matters, but choosing the lowest bidder is rarely a good idea. Price is only one of the important factors involved in choosing the right technology partner.

Major price gaps between different bids for a given project generally mean that at least one of the bidders does not truly grasp the technical challenges or development work involved. Ask bidders to explain price differences. Do not hesitate to ask specific questions to test the credibility of a bid and to make sure that all bidders really understand what you are asking for.

You should also make sure that your technology partner shares your values, and that your companies have compatible business cultures. Strong affinity and a common philosophy are the cornerstone of high quality cooperation between your team and your partner's team. In fact, this compatibility is essential if you want to create a hybrid team that is made up of developers from your company and your technology partner's company. In this scenario, working with a local team can be a real asset, especially when it comes to good communication.

Finally, unless you have very specific technical choices in mind, it is recommended that you lean toward an "agnostic" partner, one who will not lock you into a single technology and who will be able to suggest various different solutions.



5

Fixed price or hourly rate

Information technology development projects tend to be billed by the hour. If you want to pay a fixed price, your functionality specifications must be perfectly set out to the last detail, and you must plan on making few or no changes down the road. Since uncertainty must be kept to a minimum, make sure your project does not involve new or somewhat experimental technology. Proven, time-tested technology will give you a clearer picture of overall costs.

Building software is like building a house. If you have sufficiently detailed blueprints, you can make accurate cost estimates from the outset. If you change your mind along the way, you will inevitably incur 'extra' costs. However, when it comes to software development, rarely does a project not evolve over time. And the greater the scope of the project, the greater the chance of changes. That is why billing tends to be by the hour. This also prevents potential differences of opinion about what should and should not be covered by the fixed price, which can create tension during project execution. By-the-hour billing is particularly well suited to projects with iterative development cycles, as long as costs are monitored closely.

There is also a third possibility: a project can have some carefully defined stages that are covered by a fixed price, and other stages that are billed by the hour. This approach makes it possible to nail down part of the budget while leaving room to adapt to any needs that may arise along the way.



6 Identifying constraints and managing risk

It is easier to take precautions or choose the right approach when you know what pitfalls to avoid. It is important to have a healthy awareness of a project's limitations, regardless of their source: be it the people, equipment or technology involved, or the available time and money. The biggest constraints are usually created by dependence on external systems, none of which should be overlooked, and all of which must be identified properly.

Every project has risks. Taking stock of those risks from the outset, as the project is being defined, will help you devise a strategy to mitigate them. Since clients are the experts when it comes to their business, having them present at all key stages of the project is a sure-fire way of avoiding most problems.

7

Choosing iterative development

It's a fact: large-scale software projects evolve constantly during development. In order to carry out such projects successfully, your technology partner will have to keep adapting to changing priorities. Priorities can shift due to external causes such as a changing market or new needs. The project development method must therefore leave room for a pragmatic and flexible approach. Larger-scale projects should ideally be carried out in iterative mode, with deliverables at set intervals.

Deliveries should be an opportunity to meet with your technology partner, see how the work is progressing, and check to make sure that your expectations are being met. Obtaining deliverables early on can considerably reduce the risk of ending up with software that does not really meet your needs. Short delivery cycles also enable you to make adjustments immediately, and even change the direction of your project to deal with unforeseen developments.

Agile

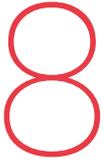
Agile is a set of management methods for IT development projects, based on some fundamental principles: a focus on communication and cooperation; rapid delivery of functionalities; constant client involvement; and openness to change. In practice, when people say "Agile development", they mean a method based on iterative, incremental and adaptive development cycles.

Waterfall

The waterfall method involves dividing the project up into different phases that follow a strict order, with no overlaps or iterative cycles. This traditional approach is suited to projects that are very predictable, with fixed requirements that are known in advance, and for which there is complete technological certainty. The main advantage of the waterfall method is that it shows exactly what will be delivered, and when. The disadvantage is that it does not allow for any reflection or review, and leaves no room for error.

Scrum

The Scrum is a set of practices consistent with the Agile philosophy, and is the most widespread approach to developing software projects. It is particularly suited to long, complex, somewhat hazily defined projects, and is based on chunking a project into time segments, generally two-to-three weeks long, called Sprints. Each Sprint begins with a planning meeting and ends with a demo of what was achieved during the Sprint, as well as a review of the way that Sprint went. The Daily Scrum, at which the entire team meets, helps track the project's progress and promotes cooperation.



Using best practices

If your technological partner uses best practices, your software project will be less likely to fail or to be found lacking. Best practices are a kind of insurance against the inherent risks of the industry. Malfunctions, service interruptions and breaches can carry a heavy price. Higher quality always means better performance.

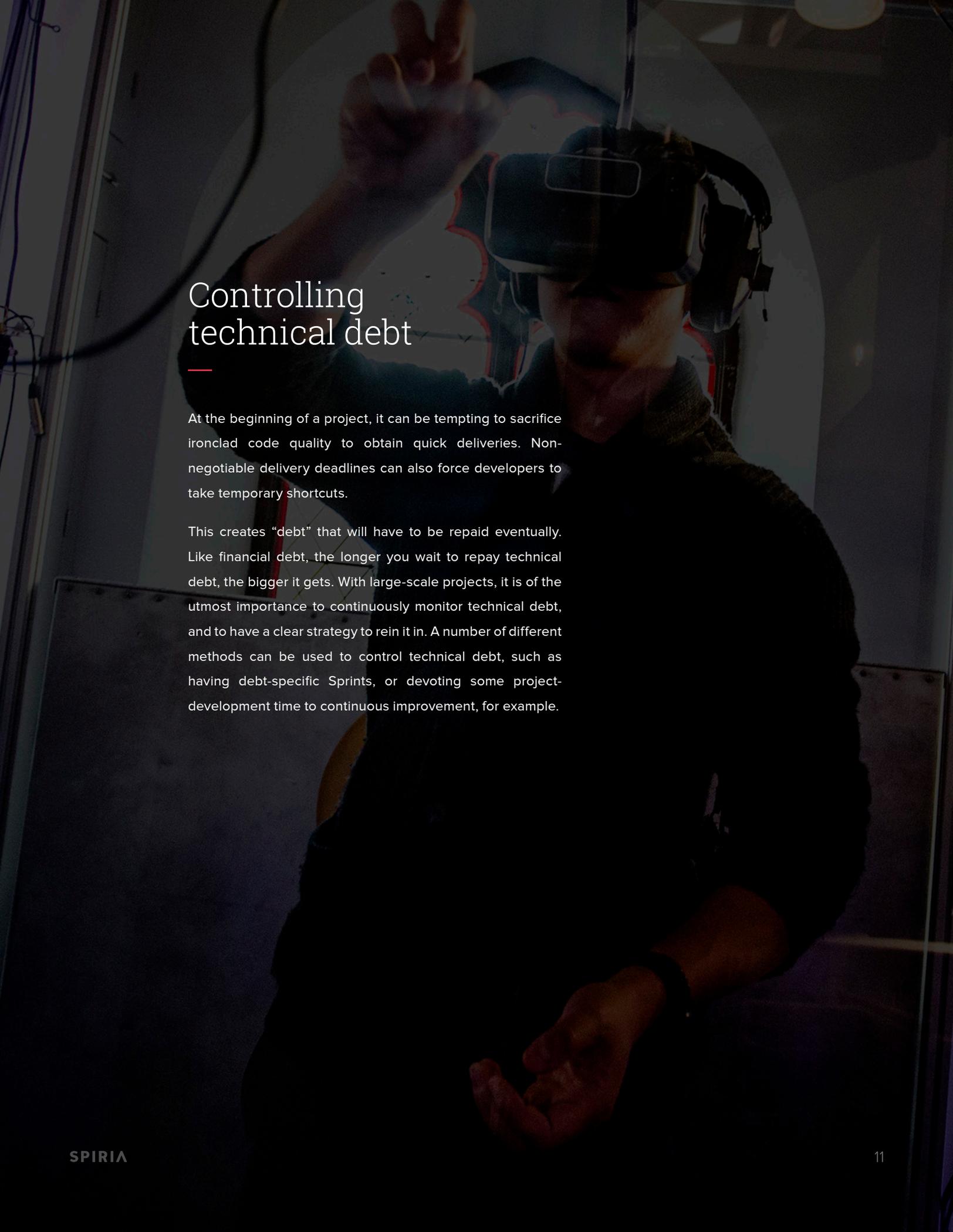
Putting users first

User experience is vital to the success of your application. Consistent design and intuitive interfaces will give users the results they want, generate greater commitment, and guarantee good uptake of the tool, which will in turn increase productivity and user satisfaction. Best practices include keeping the end user in mind throughout the development process.



UX and UI

User experience (UX) refers to the way people think and feel about using an application. UX design is the process aimed at improving user satisfaction by making an application more user friendly, accessible and just plain fun to use. User interface (UI) refers to the design of human-machine communication tools. UI is a major factor in high quality UX.

A person wearing a VR headset and hand controllers, making a peace sign gesture. The background is dark and blurry, suggesting a virtual environment.

Controlling technical debt

At the beginning of a project, it can be tempting to sacrifice ironclad code quality to obtain quick deliveries. Non-negotiable delivery deadlines can also force developers to take temporary shortcuts.

This creates “debt” that will have to be repaid eventually. Like financial debt, the longer you wait to repay technical debt, the bigger it gets. With large-scale projects, it is of the utmost importance to continuously monitor technical debt, and to have a clear strategy to rein it in. A number of different methods can be used to control technical debt, such as having debt-specific Sprints, or devoting some project-development time to continuous improvement, for example.



Focusing on quality...

Everyone agrees that identifying a bug during the development process is far less costly than having it appear later, during user acceptance testing or, even worse, at software launch. Continuous integration techniques and proper code review will help identify problems that, if corrected promptly, will save you huge amounts of time, money and energy. You must therefore make sure that your technology partner has all of the standard quality assurance tools required to ensure your project's success: unit tests, integration tests, continuous integration tests, interface tests, function tests, etc.

Continuous integration

The continuous integration process is aimed at making sure that code updates do not cause setbacks and/or bugs, and that naming and programming standards are adhered to (these rules are generally validated during code review).

And security

Finally, always keep security considerations foremost in mind and never try to cheap out on security. Security onboarding at every step of the project is part of overall quality assurance.

Code review

This review is a systematic examination of the application's source code, which amounts to a more or less formal proof-reading by one or more peers. It is designed to detect programming errors, increase the overall quality of the software, and promote richer discussions among developers.

9

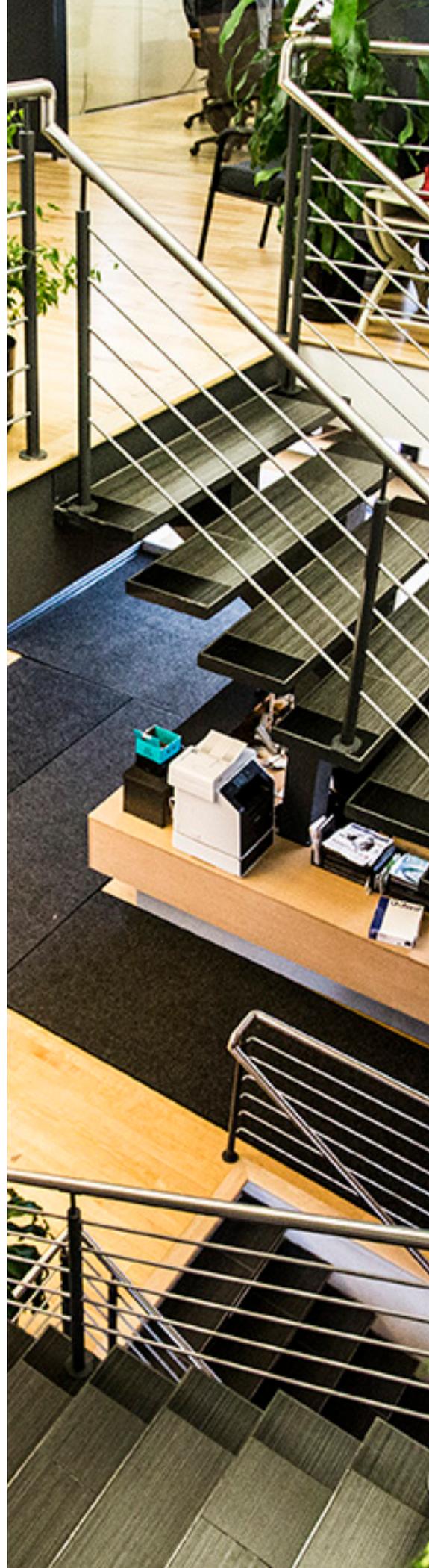
Handover

The successful implementation of a new software solution, and its uptake by users, usually requires training to ensure that the handover goes quickly and smoothly. It is important to make decisions early on about the necessary training, expected user documentation and responsibilities on the part of the development team and everyone else involved. Your technology partner must be able to provide this service. As a rule of thumb, the greater the complexity of the software solution, the more support its users will need, both from people who master the new tool, and in the form of clear documentation.

10

Maintenance

Software must be maintained throughout its life. It is important to have a partner that can offer post-launch maintenance and support services. Your technology partner must also be proactive about foreseeable problems, like those caused by a new version of your operating system. It makes sense to plan for maintenance costs from the outset of a project, add them to the software-development costs, and factor them into the ROI calculation.



Software Development Done Right

- 150 in-house digital experts committed to results.
- Over 1000 projects delivered across multiple industries.
- 15 years of digital services for over 400 customers.



CUSTOM SOFTWARE DEVELOPMENT



MOBILE APP DEVELOPMENT



WEB APP DEVELOPMENT



WEBSITE CREATION



EMBEDDED SYSTEMS, IOT AND M2M



DIGITAL TRANSFORMATION

SPIRIA

BOSTON

200 PORTLAND STREET
857 302-7071

OTTAWA - GATINEAU

60 PROMENADE DU PORTAGE
819 205-1800

MONTREAL

110 JEAN-TALON STREET WEST
514 272-0979

TORONTO

323 KERR STREET
SUITES 101-102, 202
647 477-3401