

SIX LESSONS FOR OUTSOURCING SOFTWARE DEVELOPMENT

This paper is for organizations who are considering an outsourcing strategy, as well as for those who have tried some form of outsourcing but have been disappointed in the results.

It offers six simple lessons that you can use as a starting point for your investigation into the world of outsourcing.

Macadamian
software@macadamian.com
1-877-779-6336

700 Industrial Ave.
Ottawa, ON Canada K1G 0Y9
www.macadamian.com

SIX LESSONS FOR OUTSOURCING SOFTWARE DEVELOPMENT

There are many reasons why you may have decided to look into outsourcing possibilities for your software project. You may have a hardware device that needs the right software to be sure it lives up to its full potential. Perhaps you need software ported from one platform to another. Or, you may be creating a web portal and feeling out of your comfort zone.

You've got some big decisions to make, and it can be hard to know where to start.

As a software development partner, Macadamian works with companies just like yours to complete complex, time-sensitive, or unusual software applications. So while we're not exactly unbiased, we can help you determine how to find the partner who is right for you and how to have a great outsourcing experience.

This paper is intended as an introduction to point you in the right direction. Use these six lessons as a starting point for your investigation into the world of outsourcing.

LESSON 1: "OUTSOURCING" DOESN'T ALWAYS MEAN "OFFSHORING"

Today, when most project managers hear the term "software development outsourcing," their thoughts immediately turn to India. For many companies, "outsourcing" is synonymous with "offshoring." In examining your options, though, it's important that you understand the difference.

Outsourcing refers to delegating a particular task, project, or area of responsibility to someone outside your organization. They could be a sub-contractor in the same city, a specialized firm on the same continent, or a development house overseas. By equating "outsourcing" with "offshoring," you may overlook the full range of your outsourcing options.

Many companies focus on offshore due to the tantalizingly low rates that can be found in countries like India. As a result, R&D organizations often send the wrong types of projects to offshore development firms because "you can't go wrong with \$14.00/hour labor!" But this can end up being more expensive than you planned because **offshoring the wrong kind of project is counterproductive and inefficient**. Hourly rates alone are not a sufficient gauge of the true costs – or risks – that offshoring represents.

The choice of onshore or offshore resources **should be driven by your particular development project**. If your project requires a high degree of coordination between teams, you'll probably find it easier to work with a partner with engineering and design resources on the same continent. This is often the case for projects where hardware and software are being developed at the same time. The software and hardware teams need to communicate and work on synchronized schedules.

With an onshore partner, linguistic, cultural and time zone considerations aren't an issue, and the partner will often be able to intuit your needs. In a strong onshore relationship, the partner

will be able to read between the lines of specifications – and may even be able to finish your sentences.

If your project doesn't need a lot of hands-on coordination and can be controlled by tools or configuration management systems, consider working with an offshore vendor. Projects that involve software updates to very stable, well documented, legacy hardware devices are often good candidates for offshore outsourcing.

It's even possible to use a **hybrid outsourcing model** that combines onshore project management and offshore development resources. Under a hybrid "dual-shore" model, you can engage an onshore partner who will assign project management and relevant engineering projects to in-house teams while sending project components that are better suited for offshore development to their offshore team.

But, be somewhat skeptical – some outsourcing partners who boast a dual-shore model may merely have a skeletal onshore team comprised of sales people only (see Lesson 5). If the dual-shore model appeals to you, search out partners with a strong onshore senior management and engineering presence.

LESSON 2: LOOK FOR A PARTNER WHO IS ON THE SAME PAGE

The requirements of your project should play a part in selecting your outsourcing partner, and so should critical knowledge and industry experience. No two software development firms have the same degree of engineering experience and expertise, so it's smart to shop around for a partner with a track record of success in developing the type of product you need.

Today's high tech products normally require a lot of innovation and experimentation, and the end product must be extremely reliable. Obviously, you need a partner with a skilled engineering team. But even a partner with strong software engineering experience can be tripped up by industry-specific standards, specifications or protocols that are second nature to you and your team.

In looking for a partner for your project, ask about their experience in your field and make sure that their client and project lists back up their claims. Your teams will find it easier to communicate with a partner who understands the context and scope of a your particular project.

If you're designing a next-generation device or launching a new product into a highly competitive market, you may value a partner who is familiar with **user research and human factors design**. Many organizations find that they can effectively differentiate their products through compelling product design. A partner with the expertise to

Partnering in Action

"Only two companies have ever tried what you're doing. The other one gave up." When Microsoft Technical Support tells you that, you know you're working on something special.

Our client wanted to be the first company to extend SharePoint with its own enterprise rights management functionality. With its own developers busy on the next version of their product, the company came to Macadamian.

SharePoint isn't really designed to facilitate plug-in development—but that doesn't really matter to us. Undocumented APIs just make us try harder.

While another engineering company might have given up, we didn't. And our client became the first company to have an enterprise rights management plug-in for the SharePoint server.

determine how users will actually behave with a product or service could be a major asset in helping you create and market an industry-changing device.

Of course, if your project doesn't require industry-specific knowledge or expertise (for example, building an internal web portal for staff use), you may not need a high degree of industry experience from your partner. As always, the context of your particular project should guide your decision.

LESSON 3: A SPECIFICATIONS DOCUMENT MIGHT NOT GET YOU WHAT YOU WANT

You could write an intensely detailed, 500-page specification for the application that you want for a particular device/application and still not get what you want. Some outsource partners, often offshore development shops, want you write a comprehensive requirements document to identify each function of your software, which they will then implement.

But a doorstep spec isn't always the best way to get a great application. You are without a doubt an expert in what you want to see in your product. At the same time, you're hiring someone else because you aren't an expert in software (or don't have the time to develop it in-house). So how do you ensure that your requirements document is complete and correct? How do you know if they should program it in Java or C++?

For a mature, stable product where the majority of development is bug fixing or adding non-risky new features that pose no risk of destabilizing the product, writing a spec for the outsource partner may be sufficient.

But when you're dealing with a next-generation product, look for an outsourcing partner who takes a different approach. Expect them to work with you to identify what features will function best in your environment or with your hardware. Rather than giving them a requirements document, you give them your ideas. Let the partner translate those ideas and needs into a specification and product features.

This works to your advantage in four ways:

1. You'll quickly learn how the partner works and how capable they are in the type of project at hand—for example, they may draw on experiences with similar projects to warn you about the potential risks of a certain approach.
2. You'll find out immediately if something isn't possible from a technical point of view.
3. While you might have pictured something working in a certain way, the partner might identify a more efficient option.
4. The partner may suggest ideas you hadn't considered or recommend features you haven't thought of.

Usability in Action

Imasight Vision combines X-rays with digital imaging technology, displaying X-rays instantly and eliminating the wait time of getting the film developed—and does so at 10% of the cost of a traditional system.

When Imasight was developing the Vision, they knew it would have a higher probability of being adopted by the target market if they designed the software around the users rather than the features of the product or the "cool" technology. Imasight worked with the Macadamian Usability team to study veterinary clinics and identify the tasks that vets wanted to accomplish. From there, they designed the features of the Vision to make those tasks easier and then let clinics test it under practical conditions to make sure they got it right.

The end result is that the Vision is a device tailored to help veterinarians get their jobs done instead of one designed by a software developer who wants to show off technology.

A partner with a usability team can add more value by talking to the people in your target market and turning their everyday tasks into features. Imagine creating an application designed around your clients instead of around technology – your clients would immediately realize that you created the product specifically to make their lives easier, and they will love the product for its short learning curve and ease of use.

LESSON 4: CHOOSE A PARTNER WHO HAS A PLAN FOR COLLABORATING WITH YOU

How will your development partner talk to you? How – and when – will they get your feedback? Will you be a part of the design and creation process?

Ask these questions when you're in negotiations – if the partner can't answer them quickly, it might be an indication that you're dealing with someone who hasn't been in the business very long, or one whose processes aren't mature.

You might think you don't need to be involved in the process – that the partner should do everything independently and come back with a final product, like ordering dinner in a restaurant.

But getting the software your users need is less like finding what you want on a menu and more like ordering a custom made suit. It's an iterative process that will take multiple fittings and adjustments. The tailor will ask for your feedback and input along the way.

Likewise, your feedback is vital to creating the product your clients need. It frees your development partner to try creative solutions, which you can then test to see if they work better than the initial design. Your feedback also identifies when the original design needs adjustment.

Experienced development partners will have various methods of gathering feedback in their development processes. Some of these may include:

Instant messaging

Some development partners connect with you through instant messaging, which is less interruptive than phone calls.

Email

Email is useful for questions, status reports, updates, and clarifications – in other words, non-urgent communication.

Access to everyone on the team

You should be able to talk to any member of the team, while having a single point of contact who manages day-to-day communication.

Status meetings

Status meetings are best used sparingly. They take the whole team away from development, bringing the project to a temporary standstill.

Regular progress reports/deliverables

You should receive deliverables on a regular basis, with a feedback loop that takes your comments into account. Deliverables should include:

- Feature specifications
- UI mockups
- QA plans
- QA test results

You should have plenty of opportunity to comment on deliverables. If you go more than 10% of the project without being asked for feedback, then there might be a problem.

You should also expect to see some deliverables early in the process: a week or two into the project. This will assure you that effort is being made to ramp up quickly, and will enable you to adjust any misunderstandings before they become problems.

Access to builds

It's your project; you should be able to download a build whenever you like. This allows you to see that the progress you're told about is actually happening.

Wiki/extranet

Some development partners will give you access to a wiki or extranet page dedicated to your project. There you can add things, edit, clarify requirements, and track progress—at any time of day or night.

Some development partners have additional tools to let you track progress as it happens. For example, a graph from a Macadamian project wiki is shown below. This particular graph tracks work remaining on an example project. The green line represents the forecasted remaining effort, while the red line shows actual remaining effort.

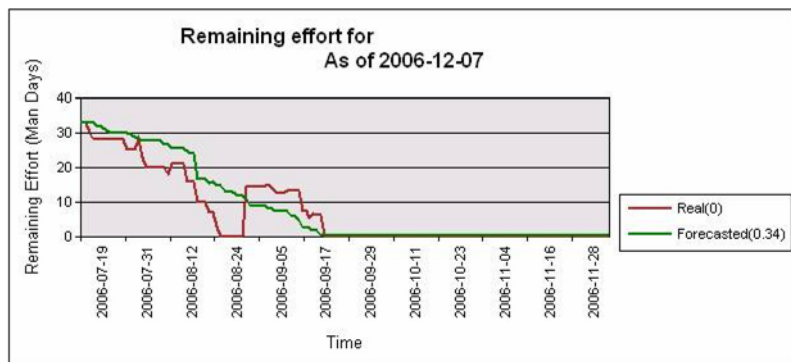


Figure 1: Sample Macadamian Project Wiki

Tools like this can give you a quick visual representation of how the project is going at any time, rather than relying on your gut or your partner's say-so. For example, in the graph above, at 2006-08-24, the remaining effort goes to zero because the work was proceeding ahead of schedule. The remaining effort rises at that point because the partners agreed to add a new feature.

This is just an example of the various ways an outsource partner can incorporate you into the process of collaboration.

LESSON 5: FIND A PARTNER YOU TRUST

For an outsourcing project to be successful, you must be able to trust your partner. You need to be confident that your partner can handle the software portion of your products professionally.

That trust must also work both ways. Your outsourcing partner will need you to provide clear product direction, a set of expectations and desired outcomes. They won't be able to develop the product you need if you are unable to provide them all with the details. If you feel you need to hide specifications or technologies even after signing a non-disclosure agreement, the partners won't have the information they need to deliver the best application possible.

Look for concrete "trust factors" when evaluating an outsourcing partner:

- 1) **Competence:** You must be able to trust the abilities of your partner. You'll never think of all the project and contract possibilities that could arise during development. When unexpected events occur, you need to be confident that your partner will "do the right thing" when required.
- 2) **Experience:** One way to assess the competence of a prospective outsourcing partner is to ask for references, and then check them out. Has the partner worked with reputable companies? Do they have experience relevant to your product or industry? Will they let you speak to past clients? Getting answers to these questions can build your confidence and help you assess what happened when others put their trust in the prospective partner.
- 3) **Openness:** It's important to be aware of the policies and processes of an outsourcing partner from the get-go. A good outsourcing partner will make their processes open to you and will give you a transparent view of how they work, how contracts are structured, how they handle problems, and how they plan to communicate with you. As we saw in Lesson #4, you should be able to collaborate easily and frequently with your partner. If you feel that a prospective partner isn't being up front with you about its processes, your "trust flags" should go up.

While trust can truly only be built and demonstrated through interactions over a period of time, if you ask some of these questions, you should be able to get a sense of the trustworthiness of the company in question. Look for partners and project managers who listen, ask good questions, and demonstrate an understanding of your needs right out of the gate.

Lack of trust can derail a project

"You don't need to do end-to-end testing. We'll do it and give you the results."

Despite our best efforts, we couldn't work out a few bugs for our client. The client wasn't happy we couldn't get the application to work reliably and we weren't either.

It didn't help that when we asked for the device to do end-to-end testing, we were told that they would take care of that.

After months of frustration, we figured out the technology they'd chosen for the device just wouldn't work for the planned functionality.

The product never passed the approval tests from the regulatory body. The project was eventually cancelled. Looks like the (very awesome) product won't be released.

If you have doubts about trusting your outsource partner, maybe you've got the wrong partner, or perhaps developing in house is a better option for you.

LESSON 6: ESTABLISH A CONTRACT MODEL THAT MAKES SENSE

If you've been looking into outsourcing you probably already know the two most popular options when it comes to contracts: Fixed Price and Time and Materials. While these two approaches each have their advantages, there are other models you should also consider.

Fixed Price

The Fixed Price model is best when the project is stable and completely scoped out. The need for innovation is low, there are few dependencies, and you don't expect to change anything. Fixed Price gives you the advantage of knowing exactly what you're going to pay.

You can expect the partner to build some leeway into the price as a means of lowering their risk. In the realm of software development, a Fixed Price model will depend on:

- The number of technical unknowns
- Other dependencies that could slow the project (hardware, for example)
- The experience level of the partner

Use Fixed Price when:

- The project is stable/legacy
- The project is well scoped out
- The project is maintenance/bug fixing
- The project doesn't require innovation
- You don't expect to make any changes to the scope/features/technology

A Fixed Price contract doesn't allow you to make many changes. If you discover that an essential feature is missing, you'll have to renegotiate the contract. And if your partner already has another client lined up right after your contract ends, that might be a problem. At the very least, it will cost you more money; at worst, you might have to wait.

A Fixed Price model also means that you will pay full price for the project no matter how long it takes. If the project has been over-estimated and comes in 25 or 30% under budget, you will still pay the full amount.

Time and Materials

A Time and Materials (T&M) contract offers the most flexibility. Outsourcing partners like T&M projects because it gives them the chance to innovate, adding or changing features that you both agree will be most useful to your users.

You'll be able to make adjustments to the project as you go, and you won't pay a Fixed Price premium for these unknown elements. Furthermore, whenever the outsource partner makes use of third-party or open-source elements, you will share in the time and cost savings.

If you feel that a Time and Materials model doesn't provide enough incentive for the partner to deliver on time, you can build penalties/rewards into the contract to encourage it (more on this later). But good outsource partners aren't out to do just one project with you – they think long-term and they want to hit their targets. Their goal is to build a long-term relationship.

Use Time and Materials when you:

- Want to make changes as you go
- Think that certain factors might shift during the course of the project
- Want to use cutting-edge/beta/legacy technology
- Are working with changing hardware

With a Time and Materials contract, you need to make sure that you're working with a reputable partner – one you trust. If you don't have an established relationship with the

partner, one way to verify this is to ask for references from clients who've worked with them on T&M projects in the past. If they can't provide any references, you might want to consider another partner.

Two-Phase Model

Splitting the project into two phases often makes good sense. Design/research lends itself well to the Time and Materials model. Following that phase, you can then use Fixed Price for the Engineering work.

This lets the partner investigate technologies and design the product on a Time and Materials basis – free to innovate and be creative in their approach. Potential issues and solutions will be identified up front, before they become costly errors. Because the Engineering phase is Fixed Price, engineering will be the partner's risk, not yours.

Use the Two-Phase Model when:

- You want to ensure a complete and accurate estimate
- Risk and unknown factors are high
- You're not sure about the viability of the project

This approach also gives you time to test the interface with users so that you can see if they enjoy working with it – meaning they will adopt it more quickly once it's on the market.

Even if you aren't starting with user testing, you can still benefit from the Two-Phase Model because when it comes time to plan the Engineering phase, the partner will be able to estimate costs more accurately. Less risk for the partner means they can plan better, and you'll pay less of a premium.

Carrot & Stick Clauses

If your project is time-sensitive, you can encourage timely delivery by adding a bonus for on-time (or early) delivery and/or a penalty for late delivery.

This lets you share some of the risk with the partner. It's most useful for projects where there is a reasonable specifications document for the application, with stable elements: hardware, technologies, platforms, etc.

Include Carrot/Stick clauses when:

- The project is time-sensitive
- There are reasonable specifications
- Factors like hardware, technologies and platforms are stable

Penalties and bonuses encourage the partner to get creative about how to deliver on time or early. But, you will need to spend extra time up-front to agree on acceptance criteria, reasonable targets, and the conditions of receiving the bonus.

Choosing the right model

When deciding between the various contract options, think about the kind of behavior you want to encourage in the partner – and be aware of the tradeoffs.

THE BOTTOM LINE

Choosing a software outsourcing partner isn't easy – but these six lessons should make it easier. By following these guidelines, you will be in a better position to evaluate and select the best partner for your project.

At Macadamian, we help companies navigate the outsourcing process and would be happy to do the same for you. We encourage you to contact our software development experts at any time:

Macadamian

700 Industrial Ave, Suite 210

Ottawa, ON

Canada K1G 0Y9

Tel: (613) 739-5976

Toll-free: 877-779-6336

software@macadamian.com