

## B18: Do's and Don'ts in Vendor Outsourcing for Data Linkage Projects

- Who is this tool for? Health and other social protection practitioners who need to commission additional IT support and/or software for a population targeting data linkage project, and wish for advice and best practice in how to outsource this.

Achieving interoperability between data systems will often require additional software to facilitate the process. A key decision in this process is what option your organization chooses to develop or acquire for this. This will be highly dependent on the complexity of the data operations which you wish to support. Broadly, there are three potential approaches, for which the main benefits and risks are summarized in Box 1:

§ **Adapting an “Off-the-shelf” (OTS) product.** This involves customization of an already developed commercial software product to the specific business needs. There is an increasing range of such OTS software which can facilitate data sharing across agencies which are worth considering.

§ **Outsourcing software development.** Hiring a professional IT consulting company to develop the software or its components.

§ **In-house software development.** Mobilizing a team of internal (potentially supplemented by external) specialists to develop a system under the internal management.

**Box 1: Benefits and risks of different software development approaches:**

	Benefits	Risks
<b>Adapting an “off-the-shelf” (OTS) product</b>	<ul style="list-style-type: none"> <li>• May be quick and easy solution</li> <li>• Direct control over the business process</li> </ul>	<ul style="list-style-type: none"> <li>• Customization may take more time than expected due to local specifics</li> <li>• Maintenance may turn to be expensive</li> <li>• High dependency on provider</li> <li>• May be associated with high contingent and hidden costs (e.g., requirements of periodic upgrades)</li> </ul>

<b>Outsourcing development</b>	<ul style="list-style-type: none"> <li>• Advantage of external technical expertise and capacity</li> </ul>	<ul style="list-style-type: none"> <li>• It is impossible to specify all contingencies in the original contract</li> <li>• High risk of time and money cost to cover contingencies</li> <li>• High dependency on provider</li> <li>• Limited monitoring capacity during the development phase</li> </ul>
<b>In-house development</b>	<ul style="list-style-type: none"> <li>• Full ownership of the process</li> <li>• Full access to the source code</li> <li>• Know-how in context, needs, users</li> </ul>	<ul style="list-style-type: none"> <li>• Shortage of skilled IT and management specialists in the public sector</li> <li>• Private/public sector wage differentials may result in drain of the maintenance capacity during exploitation</li> </ul>

While there are benefits to outsourcing software development, and it may be necessary where in-house capacity is limited and suitable OTS software is not available, it involves risks which need to be anticipated and managed. These can be broadly grouped under contractual risks, project management risks, and external and post-implementation risks (see presentation by Oleskiy Sluchynskyy of World Bank ([here](#)) that accompanies this tool, and from which risks/mitigation below are drawn). More specifically, the main risks are:

#### **Contractual risks**

- Lack of specifications in initial technical requirements
- Lack of mechanisms of adjustment to changes
- Ambiguity in acceptance procedures
- Risk of cost and time overruns

#### **Managerial risks**

- Poor management plan and weak enforcement tools
- Lack of quality control skills and/or effort on the client side
- Drifting focus and resources of the IT provider to new projects and new clients
- Hands-off approach may only delay the crisis while micro-management by the client raises additional liabilities

#### **External and Post-implementation risks**

- Policy, legal, and regulatory changes
  - Management change
  - Fast technological changes
-

Sluchynskyy also provides **tips for addressing the first two categories of risks:**

### ***Contract Management***

- Recommend use of software/hardware tools that your agency would be able to maintain (in the software development TOR)
- Ensure flexibility in the contract terms in respect to specifications, payments, and timeline modifications
- Ensure that payment schedule is connected to main milestones/deliverables
- Require training to be provided to both IT personnel and operational staff as part of contracted delivery
- Ensure clear provisions in the contract on the ownership of the source code and any (even partial) delivery
- Negotiate gradual transfer of ownership of the system components as they get developed (needs to be a contract clause)

### ***Project Management***

- Establish clear governance provisions over the process
- Use third party expertise to help ensure quality of deliverables
- Involve operational staff in the process of the software design and acceptance testing
- Require in-house presence of the development team for the whole duration of the project
- Require and periodically review the implementation management plan
- Require periodic reporting from the project team and regular consultations with the in-house IT & management teams.

There are many other tips and checklists for reducing the risks of failure from outsourced software development, all of which share common elements. Some common themes include: (i) clearly define what you want developed and what you want out of the outsourcing relationship, and specify it clearly in the outsourcing contract; (ii) at the same time, build flexibility into the contract, as business needs and other factors will inevitably change, often during the development process but also over time; (iii) do not prioritize price above everything; and (iv) ensure that the software outsourcer is closely connected to your operational (and IT) teams to ensure that what is being developed matches the operational needs and is able to be understood and operated by your staff.

Adapted excerpts from one useful set of outsourcing tips from Relevant Software are provided below (full text available at: <https://relevant.software/blog/10-tips-on-how-to-avoid-it-outsourcing-failures/>)

1. ***Determine outsourcing goals and scope.*** First and foremost, measurably define what you want to achieve (e.g., new software product development, cost savings, optimization of business processes, service quality improvement) and what resources you have to start implementing your idea. Then, find out what kind of things you lack (e.g., man hours, skills or expertise, budget to employ in-house developers).

For successful outsourcing, it is critical to identify both long-term and short-term requirements. Make sure you have a clear vision of the software development project that you are outsourcing and prepare written responsibilities and expectations for your outsourced team. Be prepared to answer multiple questions about the future software product from your potential outsourcing company. A suitable contractor should ask many questions about your project goals, will try to clarify as many requirements and expectations

---



as possible before the project is underway, and will do their best to understand the unique nature of the project. The more information you give to a vendor, the better they can estimate the size and cost of your project, align their processes with yours, and choose the most appropriate development methodology.

**2. *Choosing the right vendor.*** Available checklists and tips universally emphasize the importance of trust and business culture “fit” between the hiring agency and the outsourcer, and this is confirmed by systematic academic research which has analyzed the experience of organizations using outsourcing. In addition, you should examine: (i) vendors’ services models and cooperation approaches; (ii) expertise domains; (iii) portfolio of previous projects; and (iv) client feedback from previous work.

**3. *Focus on value over cost and don’t go for a Fixed Price contract where possible.*** First, your fixed price projects always cost more. All the project risks will be included in your fixed bid, because your vendor will seek ways to cover uncertainties that inevitably occur in every project. Some will add up to 50% on the price for high-risk projects. Second, changes to the scope are inevitable as it’s almost impossible to determine every single detail of the project beforehand. Every change to the scope will go through time-consuming negotiations on whether this change is included in the fixed price or not. Third, you will end up compromising on quality, creativity, and improvements, because the value of the work becomes less important than the price. The quality of work may suffer as your vendor will have to focus on hitting the agreed price, rather than putting efforts on making better software.

**4. *Frame a detailed written contract.*** Write a contract with clearly defined outputs and service levels based on an approach of shared risk. The contract should be linked to a detailed Statement of Work (SOW) in which everyone agrees to take responsibility for fulfillment of the appointed tasks. The SOW outlines the guidelines and expectations of the project. This includes the timeline and milestones (and what happens if they are not met), features, pay structure, acceptance criteria etc. It is also crucial to negotiate ownership of the intellectual property (IP), so that the code is your property after it is written, not just licensed to you. If there are accessories, plug-ins, or hardware that are compatible with your software, specify who owns them. You may also wish to include a Non-disclosure agreement (NDA) which protects your agency information to which the vendor is exposed during the assignment from release to third parties. Finally, make sure to spell out the risk management or contingency/exit plan (including refund provisions) in the event of a failure to deliver and spell out the dispute resolution mechanisms.

**5. *Prepare your in-house team and have a clear communication plan for interactions between your in-house team and the vendor team.*** Hold a meeting involving all relevant players on your side and from the outsourcer and explain to your team the ways how they should work with the remote team: discuss possible communication models, educate about collaborative software and project management tools you are about to install, and encourage your people to cooperate with the new team in a productive and friendly way. According to the Project Management Institute (PMI) study, ineffective communication is the primary contributor to software project failure.

**6. *Set up a clear management structure for the project on both sides with clear metrics.*** You should agree: (i) what type of management and reporting structure will be in place to manage the project? (ii) what measurement criteria will the vendor use to measure and track metrics for the project? and (iii) how will measurement criteria be reported and presented?

---



Some additional blogs and resources with guidance and industry experience on outsourcing software development:

Agarwal, N. (2021) "Software Development Outsourcing Tips From 15 Industry Experts" appinventiv.  
<https://appinventiv.com/blog/software-development-outsourcing-tips-by-industry-experts/>

Cardonet. (2020) "IT Outsourcing due diligence checklist 2021" Cardonet: Guidance, IT Support.  
<https://www.cardonet.com/news/it-outsourcing-due-diligence-checklist/>

Hojman, M. (2021) "Outsourcing Software Development: Tips for success in 2021" Hexacta.  
<https://www.linkedin.com/pulse/outourcing-software-development-tips-success-2021-martin-hojman/>

Sluchynskyy, O. (n.d.) "Procurement of Information Management Systems" Modernization of SP Programs. World Bank.  
<https://docs.google.com/presentation/d/1bkAwPJIPXH7kOJxQflgUzF0OYTWc5y-u/edit#slide=id.p3>

Taplin, S. (2020) "Five Tips to Manage Outsourced Software Providers" Forbes Technology Council. Forbes.  
<https://www.forbes.com/sites/forbestechcouncil/2021/08/09/five-tips-to-manage-outsourced-software-providers/?sh=6465b83a150a>

---