

Mapping #1: A Practical Guide to Avoiding Outsourcing Problems

Practice Paper

Chris Daniel



The Problem

Over the past couple of decades, the subject of outsourcing has been covered many times and in some depth by different researchers and practitioners, and a pretty consistent story line has emerged: outsourcing can do a lot of good, but only if applied appropriately.

Outsourcing can do a lot of good, but only if applied appropriately.

Even publications a decade or two old¹ report that outsourcing projects executed as a part of a bigger, strategic initiative have greater chances of success than those executed solely to cut costs or out of financial necessity. Unfortunately, despite that recognition, financial considerations are still a primary motivation to outsource². That fact is often exploited by vendors^{3,4} who promote outsourcing as a way of lowering the cost of doing business, without going into much detail and without mentioning how many outsourcing projects fail.

Among all this noise, it is quite difficult to spot more considered observations about IT outsourcing, such as:

It is also difficult to specify future needs.... Unless perfectly specified, the work ... will either: a) not meet the organization's ACTUAL requirements, or b) will (and should) cost more than originally priced⁵.

What you don't know can cost you.

Or, in other words: specification is crucial and what you don't know can cost you⁶.

While these comments acknowledge a problem, they do not nail it. They suggest that if you put in enough effort, you can create a perfect specification. This is one of the most misleading assumptions one can make about outsourcing. Uncertainty is a part of our life, and even if you are working in a stable environment, you cannot know what the future will bring. Therefore, it is impossible to write a perfect specification.

The perfect specification, or the perfect contract, does not exist. Some parts of your business are changing; some are new and experimental. In these areas, the only way forward is to experiment, learn and improve using what you have learned. This unfortunately limits planning to one step ahead, because you never know what you are about to discover.

This means that you have to accept that uncertainty is a part of business, and that it cannot be systematically removed. You need to learn how to deal with uncertainty.

Only recently, we started seeing public recognition of uncertainty. In 2016, Oliver Hart received the Nobel Prize in Economic Sciences for his lifetime study of the impact of ownership and incomplete contracts on the structure of companies⁷. Hart's research aligns with the work of LEF researcher Simon Wardley, who has invented a way to measure uncertainty and proposed how to deal with it efficiently.

Before we dive deep into that framework, let's look at whether you have an outsourcing problem, and how big it is.

You need to learn how to deal with uncertainty.

1. <https://www.sciencedirect.com/science/article/pii/S0969701297000014>

2. <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/process-and-operations/us-cons-sdt-gos-exec-summary-2016.pdf>

3. <https://www.orientsoftware.net/software-outsourcing/why-outsourcing/>

4. <http://ddi-dev.com/blog/case/seven-commandments-successful-outsourcing/>

5. <https://fedtech.quora.com/Outsourcing-Federal-Systems-does-it-work>

6. <http://ww2.cfo.com/it-value/2012/06/it-outsourcing-what-you-dont-know-can-cost-you/>

7. [https://en.wikipedia.org/wiki/Oliver_Hart_\(economist\)](https://en.wikipedia.org/wiki/Oliver_Hart_(economist))

How big is your outsourcing problem?

There is no easy answer to this question, because the amount of money at risk is not just the value of outsourced projects that fail to live up to expectations, but also the value of projects executed internally and less efficiently than if they had been outsourced. Only by reviewing every project and estimating how much money is being lost due to incorrectly executed outsourcing can you determine the scope of inefficiency in your company, and figure out whether this is something you accept, or (more likely) decide to tackle.

Of course, the problem of outsourcing is not as sexy as other modern challenges – such as becoming digital or reviving IT in the organization – but it establishes a rock-solid foundation on which the business can thrive. Getting outsourcing right is part of organizational and operational hygiene.

Unless you have a decision log that allows you to be sure you have made the best outsourcing decisions, you have to assume that you are wasting an enormous amount of money. Fortunately, this wastage can be easily stopped by applying the framework laid out in this paper.

Problems... or symptoms?

Press articles covering outsourcing failures^{8,9} provide numerous examples of what can go wrong, and seem to cite the same issues again and again:

- Unforeseen challenges
- Fluid scope
- Contracts difficult to renegotiate, or difficult change control
- Low quality of delivered solution, often in combination with imprecise SLA

We could look at these problems individually from different angles and propose various remedies to address them. But other things become clear when we map the problems onto the certainty axis, as shown in Figure 1.

By definition, projects with low certainty are expected to change, and Hart’s work tells us it is impossible to write a satisfactory specification to cover their requirements. On the other hand, SLAs and quality metrics are crucial for the success of projects that change rarely. But certainty only comes after a long and painful process of experimentation, so the majority of projects are somewhere between the extremes on the certainty scale – they require change management in some areas, and focus on SLA and quality metrics in others. What’s required is specific to a project, not consistent across all projects. One size does not fit all.

This leads us to the conclusion that what we often consider to be a problem with outsourcing is in fact a manifestation of a deeper issue: an incorrect, if not naïve, approach to the task.

One size does not fit all.



Figure 1 – Common causes of outsourcing failure, mapped against their certainty

8. <https://www.itproportal.com/2015/12/19/five-of-the-biggest-outsourcing-failures/>
9. <https://www.thebalance.com/lessons-from-indianas-failed-outsourcing-deal-2553038>

Responsibility

The previous section outlined the importance of choosing the right approach to the project; there is, however, a small caveat: hardly any project is homogenous enough to be suitable for just one approach. More often than not, we find projects consist of many smaller components, with various levels of uncertainty, and require multiple approaches in place for one contract – or even multiple contracts. One size not only does not fit all projects but is unlikely to fit the whole of one project.

This means that to ensure the contract is constructed properly before it is signed, you are responsible for breaking the project into small components. While this sounds like additional work for you to do, in doing this work you are discovering what is important for your business and making the important choice between retaining that knowledge yourself, or letting others learn about your business at your expense and use that knowledge for their own good. It is especially important when you are about to commence a project that may become your next competitive advantage – are you sure you want to let others learn about it at your expense?

Root cause(s)

We all know that outsourcing can significantly cut costs, but can also fail in a spectacularly expensive way. The work of Oliver Hart and Simon Wardley indicates clearly that outsourcing is not a universal tool. Not everything should be outsourced.

Projects fail if they are not specified precisely enough, and we have already learned that some components are impossible to specify. A project approached in the wrong way will show many of the common failure symptoms, such as:

- **Excessive costs** – change management is expensive for both sides, and the outsourcing supplier will want the client to pay
- **Cutting corners** – this is likely when the vendor is not in a position to renegotiate the contract and has to cope with ‘unanticipated’ changes
- **Communication breakdown** – the vendor may not be willing to renegotiate the contract if that would incur excessive costs (for example, the need to double the team or stretch the scope without changing the price)

The good news is that most outsourcing issues can be avoided with a relatively simple approach: using small components, dealing with uncertainty early in the project, and then establishing the right verification measures.

Current state of the art

At the subconscious level, we always knew that things are not as certain as we would like them to be. The current popularity of Agile frameworks shows that we have come to terms with the presence of unknowns, yet our reaction is far from perfect.

But our embrace of unpredictability has gone too far. The *plan-it-all* approach has been replaced with *don't-plan-it-at-all*, which seems to be at least as dangerous to outsourcing contracts, if not more so: the deep incompatibility between fixed budgets and agile contracts forces at least one party to gamble to get anything done.

The plan-it-all approach has been replaced with don't-plan-it-at-all.

We need to find a balance, an approach that can handle efficiently both things that are predictable and can be subjected to a plan, and things that cannot be planned, but have to be executed in order to discover new knowledge.

Our solution is presented below.

Not everything should be outsourced.

The Solution

The key to dealing efficiently with complexity and uncertainty is to identify and deal with them early in the project. You need to go through the entire project and methodically discover unknowns.

Identify goals

We have already shown that ownership and outsourcing are two different approaches to the same problem, and that they behave differently and support different types of components, depending on their uncertainty. But regardless of what approach is chosen, the question ‘why you are doing this?’ always matters. The ultimate purpose of every company is to earn money, and so every project should be accompanied by a study that shows how the value of the company will increase when the project is successfully implemented. It is a simple risk-and-reward calculation, especially important because the expected returns are likely to determine the investment budget.

While this may seem obvious, it is only half of the work required in this area, because your vendor has goals, too, and more often than not, its goals and yours are mutually exclusive. The conflict of interest is real. The more profitable the deal is for the vendor, the less profitable it is for you. Unless you consider the value to be delivered to both sides and the related uncertainties first,

you as an outsourcer can be easily exploited. A vendor may bid very low, and rely on contract uncertainty to increase its profits later, since the only thing that is certain is that there will be changes.

Note how this resonates with the responsibility discussed above – the vendor acts in its best interest. Who but yourself is to blame if you fail to take that into consideration?

An honest conversation covering the project’s goals for all interested parties is the essential first step. It not only determines the scope of a project, but also let you balance contradictory goals early in the process.

Decompose complexity

The simplest way to tackle a large project is to break it into smaller components. Once the scope of the project is set, and you know what you want to achieve, you can break the expected solution down into small, manageable components, and identify the dependencies between them. Start with the user, and the user’s need, and list the components in the value chain required to satisfy that need. Include everything that is important – including activities that need to be performed, knowledge, data and practices. This analysis can be presented graphically, as in the example shown in Figure 2.

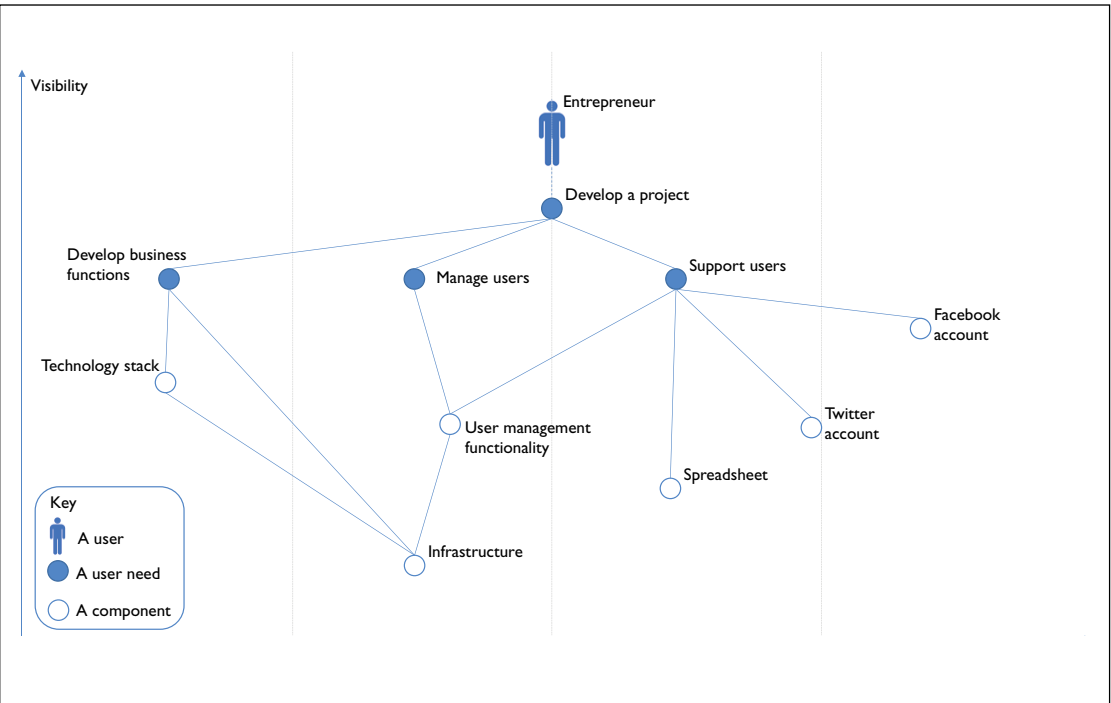


Figure 2 – A simple map of the components of a project and the dependencies between them

Uncertainty is heavily correlated with ubiquity.

Identify uncertainties

Simon Wardley, in his research into the pattern of evolution¹⁰, found that uncertainty is heavily correlated with ubiquity, as shown in Figure 3.

In other words, the more saturated the market is with a given component, the more certainty is associated with that component. The more people use something, the more is known about it.

Simon not only found this correlation, but also identified the major component types and their characteristics. Components can be categorized as belonging to one of four types or stages of evolution: genesis, custom-built, product and commodity/utility, as shown in Figure 4 and described below.

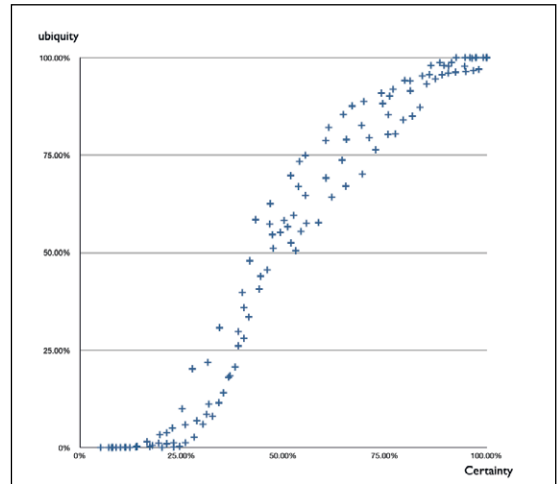


Figure 3 – The correlation between certainty and ubiquity is striking

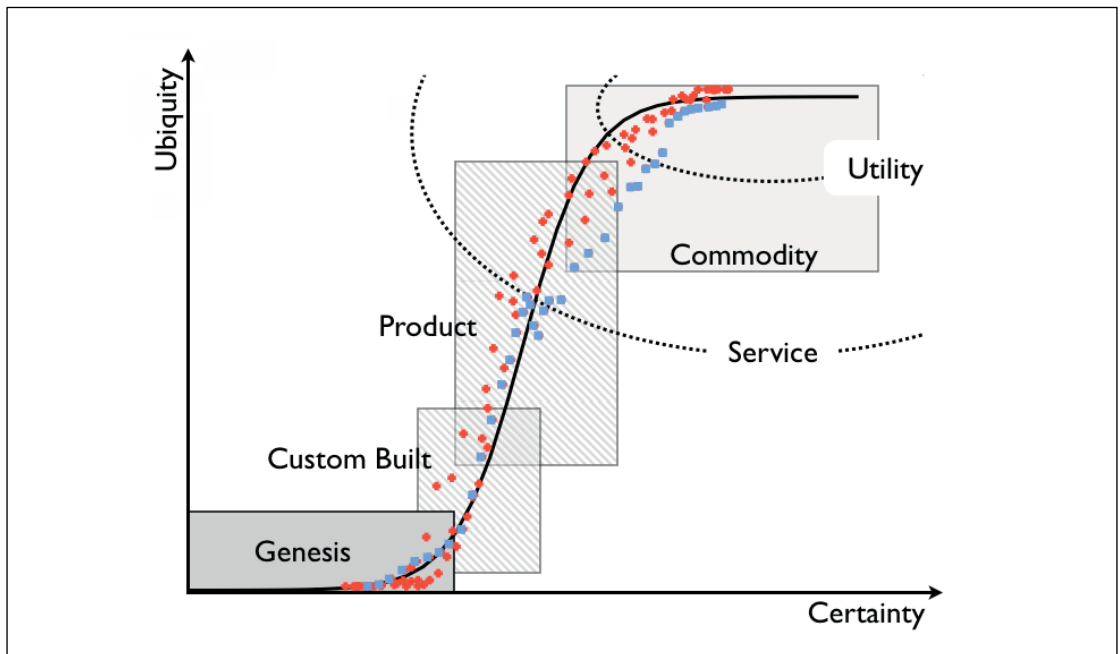


Figure 4 – The four component groups: Genesis, custom-built, product and commodity/utility

10. <http://blog.gardiance.org/2013/01/evolution.html>

- **Stage I (Genesis)** describes something very new and highly uncertain. These components are not usually used commercially, but they seem to have some future value that needs to be identified by experiments and exploited. Their future is highly uncertain.
- **Stage II (Custom-built)** describes components that are used commercially, but are not yet offered by vendors. Whether skills or solutions, you have to create them for yourself, or have them created specifically for you. The uncertainty is high, but so too is the potential reward from using the solution.
- **Stage III (Product)** describes components that are available as off-the-shelf solutions. You need them and you can buy them (though they may be expensive). The associated uncertainty is rather low, as you know what to expect from your investment.
- **Stage IV (Commodity/utility)** describes mature components with little to no uncertainty. They are so mature that they are often taken for granted, and attract our attention only when they fail.

Knowing one characteristic allows you to infer others. What's significant in the context of outsourcing decisions is that components with different degrees of uncertainty have different characteristics, and using the table in Figure 5 we can easily determine in which category any given component belongs.

Type	I - Genesis	II – Custom-built	III - Product	IV - Commodity
Characteristics				
Ubiquity	Rare	Slowly increasing adoption	Rapidly increasing consumption	Widespread and stabilizing
Certainty	Poorly understood	Rapid increases in learning	Rapid increases in use	Commonly understood in terms of use
Publication focus (Press, social media)	Wonder of the component	Building, constructing, awareness and learning	Maintenance, operations, installation, features	Use and what can be built with it
General properties				
Market	Not existing or undefined	Forming	Growing	Mature
Knowledge	Experimenting	Learning how to use	Learning how to operate efficiently	-
How people in the industry branch see it	Too early to say	Domain of experts	Expected to be used	Trivial and expected to be used
User perception	Exciting, surprising, confusing, different	Emerging	Expected	Standard
Perception by the business	Possibly a competitive advantage	A competitive advantage. Focus on ROI and example	Advantage through implementation or additional features	Cost of doing business
Focus on value	High future worth	Seeking profit	High profitability	Reducing margin compensated by volumes
Understanding	Unpredictable	Increasing understanding Development of measures	Increasing education Refinements of measures	Well defined, stable and measurable
When comparing with other companies using the same component	Difficult to understand what the other company is doing	Possible to learn from other companies	Possible to find feature differences and identify important ones	Possible to find out best practices worth copying
Failure	Tolerated if not assumed	Unsurprising	Not tolerated, focus on improvement	Not expected
Marketing approach	Gambling	Exploring a found value	Market analysis	Metric-driven
Efficiency	Reducing cost of change	Reducing cost of waste		Reducing cost of deviation
Decision drivers	Heritage/culture	Analyses and synthesis		Previous experience

Figure 5 – Characteristics of the four component groups

Mapping #1: A Practical Guide to Avoiding Outsourcing Problems

Once the uncertainty of a component is identified and hence the stage it is in, you can add a second axis to the value chain diagram, and form a **Wardley Map**, as shown in Figure 6.

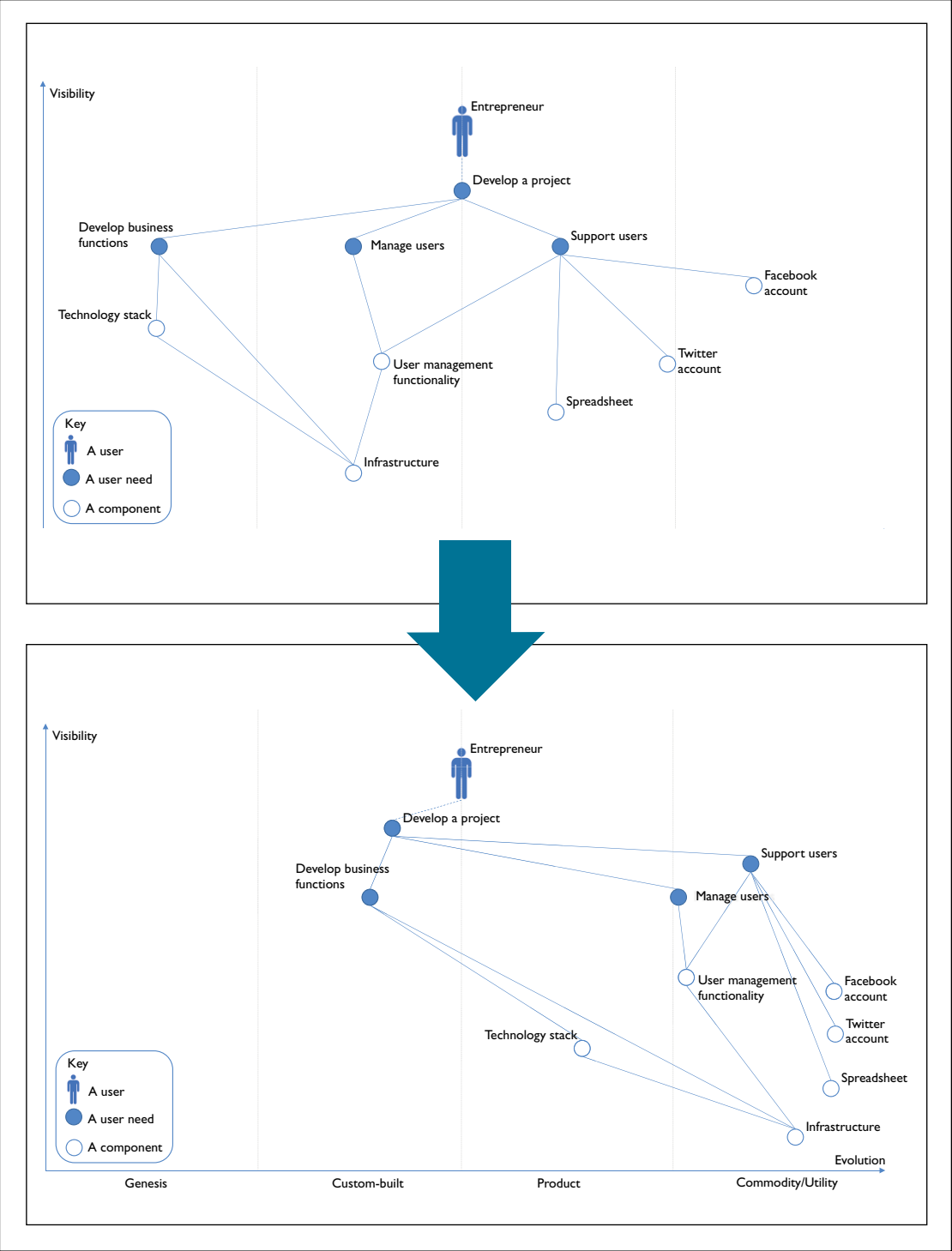


Figure 6 – Uncertainty gets added to the value chain

Close knowledge gaps

It may happen that the uncertainty of the components available in the market differs from that of the components used by your company, especially if you are using custom-built solutions while the market has moved on toward utilities. Where this is the case, there is likely to be inefficiency and in general such an ‘efficiency gap’ needs to be closed before the final solution is implemented. If knowledge already exists in the market, reuse it instead of making all the same mistakes that other companies did before you.

You need to be open to learning at this stage. The bigger the gap between your components and those available on the market, the less efficient you will be. You want to avoid building a system that is inefficient right from the start and becomes increasingly inefficient over time.

If knowledge already exists in the market, reuse it.

Figure 7 shows an example comparison of components an organization is using (blue circles) and components available in the market (red circles). This organization has a lot of work to do to close its inefficiency gap in customer service spreadsheets and technology.

As a bonus, this exercise delivers basic competition analysis – how your solution will compare to those used in other corporations.

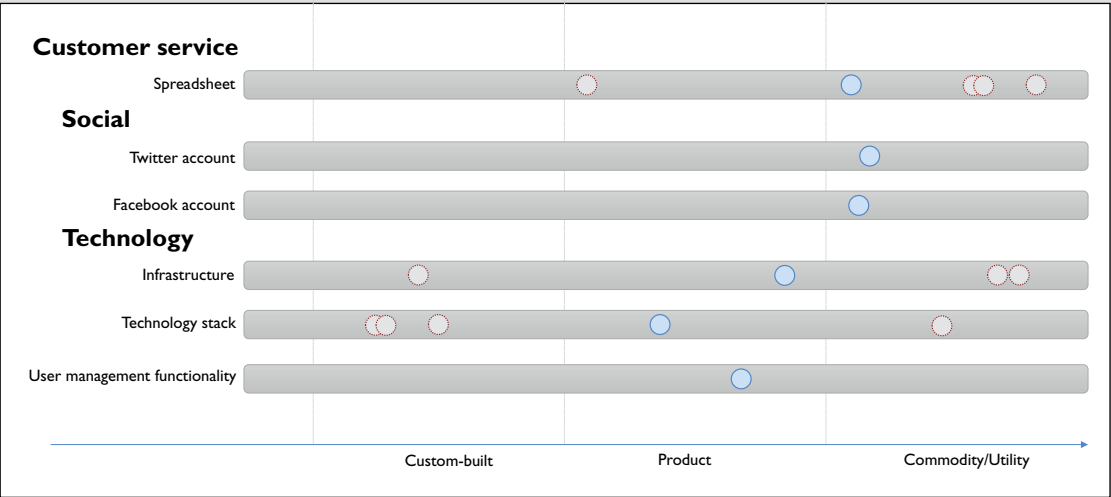


Figure 7 – Example comparison of components used by an organization and available in the market

Group components

At this point you have a huge number of small components, with various degrees of uncertainty. A big contract is always a big risk, but on the other hand, outsourcing large numbers of very small components will be difficult, too.

We recommend that you group components based on profiles of existing vendors and components’ uncertainty. Small components can be grouped together into a larger contract only if they:

- require similar expertise
- have a similar level of uncertainty
- have substantial internal connections

Potential groups can be readily seen on your Wardley Map, as the example in Figure 8 overleaf shows.

Group components based on profiles of existing vendors and components' uncertainty.

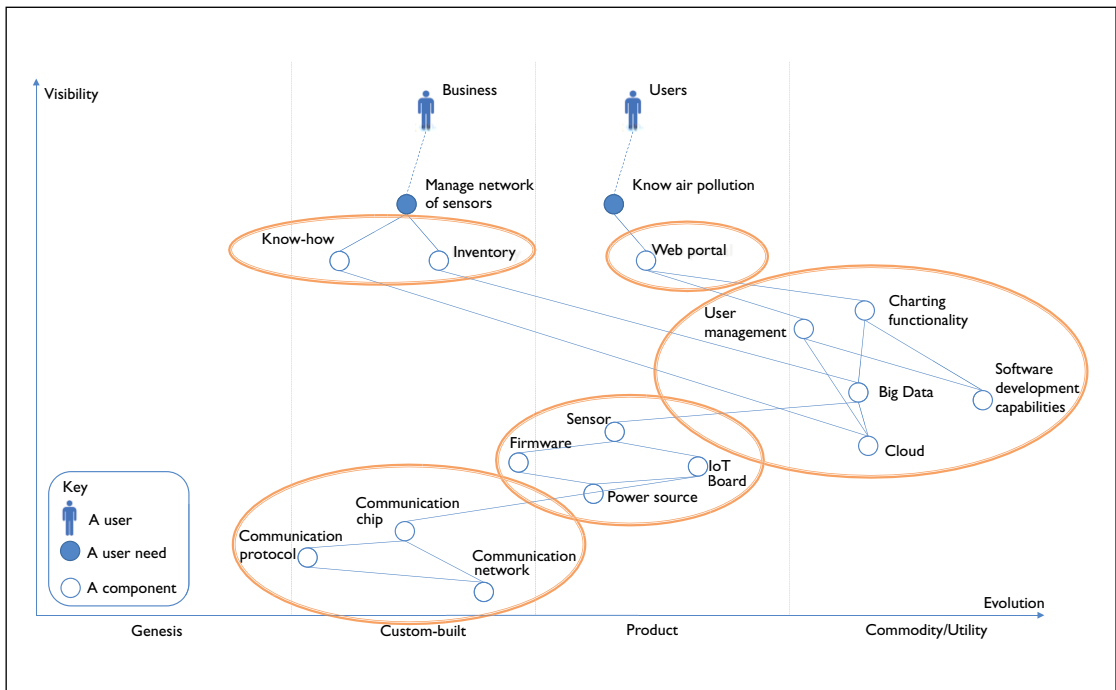


Figure 8 – The visual approach makes it far easier to split a large contract into smaller components

This approach consciously and purposefully reduces complexity; it delegates some responsibility to vendors, but in so doing, you acquire the knowledge necessary to verify the delivered product.

At this point, you can carry out traditional vendor-seeking activities – vetting, bidding and skill validation.

Execute

So far, we have discussed the difference between outsourcing and retaining ownership of projects, and reduced the decision to issues of cost and control. We have also seen how to discover what you still have to learn, how to split the project into smaller parts, and how to isolate uncertainty. These steps help avoid the most dangerous pitfalls of outsourcing. However, they do not mean you can stop paying attention to other aspects, such as conflict of interests and governance.

The isolation of uncertainty makes governance a little bit easier, as it reveals that more than one approach is needed to handle different components. For example, if a contract covering a single component or a group of components is

highly uncertain, no vendor can deliver the solution on fixed-scope, fixed-cost, fixed-time basis. Beware – this does not mean that vendors will reject this kind of contract; they may agree to it because they don't have sufficient negotiating power, or because they hope to make a profit through change control as the project progresses. In both cases, who will be responsible for the contract failure?

What is more, the vendor will be learning at your expense, and you will have no control over the IP transfer. As a general rule, uncertain components should be handled in-house or on input-based contracts with very frequent checkpoints. Checkpoints are essential to the whole process, as they offer you the opportunity to learn, and to readjust direction as you learn. This is supported by an Agile contract approach, and by direct ownership.

Uncertain components should be handled in-house or on input-based contracts with very frequent checkpoints.

The contract has to be constructed in such a way that it will balance the conflict between your own and your vendor’s interest and ensure that introducing changes (including early termination) will be as inexpensive as possible. This also means that you as a customer have to work with the vendor all the time, and that requires the right people at the right management levels (both operational and executive).

On the other hand, mature components beg for outsourcing – unless they are your ‘core’ product. Since there is no uncertainty, it is easy to select appropriate SLAs and verify delivered value. The knowledge is out there, easy to acquire. Established vendors already have good (or bad) reputations and are being constantly validated by other customers. If you keep the

activity in-house you will be competing with other providers that probably have much bigger scale and efficiency. This will not be good for your margins.

However, there is one significant danger area here: customizations. You may say that although an activity is mature in the market, you do it differently and it is a competitive advantage, so you must keep it in-house. In such a case, you have to be extremely careful about your assertion – it may be that the market is right, not you, and what you consider to be a custom-built solution (expensive, unstable) could be easily replaced with something more stable.

This general approach to outsourcing components is summarized in the table in Figure 9.

	Genesis/ custom-built	Product	Commodity/utility
Uncertainty	Significant	Small	Virtually none
Focus	Acquiring knowledge	ROI	Efficiency
Approach	In-house or input-based contract, easy to renegotiate, with frequent checkpoints and close relationship on different levels (executive/operational)	If available, buy an off-the-shelf solution and ditch customizations	Outsource. Look for best practices in the market
SLA	The uncertainty renders most SLAs useless – they either measure invalid parameters, are incorrectly analyzed or are irrelevant to the project success	Good-to-have	Must-have for the key deliverables
Remarks	Prepare for failure and learn as much as you can. Mitigate risks. Since these projects may result in some IP, consider doing them in-house	Leverage market knowledge. Ditch customizations	Leverage market knowledge. Accept product limitations – <i>do not customize</i> . Use output-based contracts with clear acceptance criteria. Pay attention to standards – lock-in is dangerous

Figure 9 – Summary of the recommended approach to outsourcing contracts

Mature components beg for outsourcing – unless they are your ‘core’ product.

Summary

The key problem with outsourcing is uncertainty, and we have shown how to deal with it – by taming conflict of interest, structured project analysis, and careful experimentation. This approach will help you avoid most common outsourcing mistakes, including:

- Over-large outsourcing contracts – large contracts hide uncertainty and complexity. By splitting the contract into a set of smaller contracts each covering related components, you not only truly reduce complexity, but also get a chance to utilize available talents in the best possible way.
- Growing project scope – by defining first what you need and why, you put a context on the entire outsourcing endeavour. Potentially dangerous uncertain components receive special treatment that balances investment with potential returns. And, as a bonus, those components cannot cause the entire project to fail.
- Losing access to innovation – by doing things nobody has done before (uncertain components) in-house, you get a chance to discover something important that could become your future competitive advantage.

Disclaimer: while this approach aims to improve project success rate, it will not help you determine whether a given project is a reasonable, strategic initiative. For more information please visit leadingedgeforum.com

Regional Headquarters

The Americas

1775 Tysons Blvd
Tysons, VA 22102
USA

3000 Hanover St
Palo Alto, CA 94304-1112
USA

Asia, Middle East, Africa

Level 9, UE Biz Hub East
6 Changi Business Park Avenue 1
Singapore 468017
The Republic of Singapore

Australia & New Zealand

26 Talavera Road
Macquarie Park, NSW 2113
Australia

North and Central Europe

Schickardstrasse 32
71034 Boeblingen
Germany

South Europe

Tour Carpe Diem
31 place des Corolles
CS 40075
92098 Paris La Défense Cedex
France

UK and Ireland

Floor 4
One Pancras Square
London
N1C 4AG
United Kingdom

United Kingdom

Bracknell/Amen Corner B1-2, UK
Cain Rd. Amen Corner (Blgd BRA02)
Bracknell RG12 1HN
United Kingdom

About DXC Technology

DXC Technology (NYSE: DXC) is the world's leading independent, end-to-end IT services company, helping clients harness the power of innovation to thrive on change. Created by the merger of CSC and the Enterprise Services business of Hewlett Packard Enterprise, DXC Technology serves nearly 6,000 private- and public-sector clients across 70 countries. The company's technology independence, global talent and extensive partner alliance combine to deliver powerful next-generation IT services and solutions. DXC Technology is recognized among the best corporate citizens globally. For more information, visit www.dxc.technology.

Leading Edge Forum

Asia Pacific and Australia

135 King Street
Level 20, Sydney
NSW 2000
Australia

France

Tour Carpe Diem
31 place des Corolles
CS 40075
92098 Paris La Défense Cedex
France

Germany, Austria and Switzerland

Römerstrasse 11
D-82049 Pullach
Germany

United Kingdom, Ireland, Iberia, Italy, Benelux, The Nordic Region and South Africa

Floor 4
One Pancras Square
London
N1C 4AG
United Kingdom

United States and Canada

1775 Tysons Blvd
Tysons, VA 22102
USA

About Leading Edge Forum

Leading Edge Forum (LEF) is a global research and thought leadership programme dedicated to helping clients reimagine their organizations and leadership for a tech-driven future. We serve as a strategic touchpoint for CXO teams to provoke and challenge their thinking to help them win in the 21st century.

We believe that as business and IT become inseparable, virtually every aspect of work and the modern firm will need to be reimaged, and this creates exciting new digital opportunities.

Through an annual membership programme of research, events, onsite workshops and advisory services, we support senior leaders in areas such as strategy, organizational change, executive education, talent development and the future of the IT function. Members enjoy personalized access to our global network of thought leaders, clients and leading practitioners.

Leading Edge Forum is part of DXC Technology. For more information, visit leadingedgeforum.com.