# MAYONNAISE WITH

## AGILE & OUTSOURCING
# CHOCOLATE MAYONNAISE
### Better than you think!

> Articles

# CHOCOLATE

GIARTE

# WHAT MAKES OUTSOURCING WITH AGILE TASTE GOOD?

TEXTS MARCO GIANOTTEN

Outsourcing with agile? At first glance, this may seem like an illogical combination. After all, outsourcing is quickly associated with strong top-down management and, according to classical views, works better with tight constraints. That doesn't fit well with the almost philosophical principles of agile that focus on the ability to react quickly to change. And yet agile could well be the ingredient to spice up the usually bland-tasting outsourcing.

The idea of using agile working in an outsourcing relationship is a lot like a chocolate-mayonnaise cake recipe. The first time you read it, you think: 'that combination doesn't work.' Still, many people who have tried chocolate-mayonnaise cake, find the combination delicious. The same goes for agile and outsourcing. With the right recipe to combine these two ingredients, you can achieve the desired business objectives. This new recipe requires rethinking, and results in a different way of doing things in outsourcing.

Agile has taken off in recent years. But as with anything that gets popular, it runs the risk of overshooting its mark. So despite agile's rock star status, it's good to stay critical. Where outsourcing based on SLAs and KPIs has overshot and degenerated into 'metric obesity', agile can overshoot within the belief system.

## Fundamentalist traits

As long as the focus is on value, agile delivers what you expect. But in many organizations, discussions about agile take on fundamentalist

tendencies. With 'over-agile', bottom-up engineering can turn into a top-down management issue – think of the rapid rise of the *Scaled Agile Framework* (SAFe). The agile principle is based on small, cross-functional teams, sprints, and roles, SAFe is focused on scaling up: unleashing agile on large companies while embracing lean principles.

Agile has also made its appearance in outsourcing. This is not without reason because, with traditional outsourcing, there is often a great distance between the customer and the supplier, symbolized by governance structures and strict SLA reports. The first generation of outsourcers said: 'IT is not part of my core business, so I can outsource it to external parties.' Now organizations say: 'My business model is becoming increasingly dependent on IT, so I want to influence the speed of change and the production process.

## Multidisciplinary collaboration
The realization that agility and value creation are increasingly vital for IT has led to both outsourcers and service providers looking at agile principles to structurally restructure the required multidisciplinary collaboration between both

Companies are increasingly linking agile working to outsourcing. Companies that already apply this unique form of 'food pairing' are Heineken, Jumbo and ProRail, among others. Giarte is seeing the appearance of outsourcing in combination with agile in all domains and contracting lots: applications, infrastructure and workplaces.

In the agile world, the boundaries between applications and infrastructure are blurred. The consequence of this for IT outsourcing is that the strict separation between the various lots is being replaced by value streams as the central construct. This is why this article emphasizes the application of agile in the outsourcing of applications (development and management) and infrastructure because these lots – which are also  influenced by advancing technological developments – increasingly overlap.

The critical question here is: which specific ingredients of agile make outsourcing taste better?

## PARADOXICALLY, TO GAIN TRACTION, YOU MUST FIRST RELINQUISH CONTROL.

sides. Part of this change is that organizations have switched to backsourcing to do specific value-creating IT activities themselves again. But due to the scarcity in the IT labor market, the increased demands placed on software and IT environments, the increased size and impact of IT, and the frantic pace of innovation, 'doing everything yourself' is not an option for many organizations: outsourcing remains in many cases a necessary approach.

## Looking for a new recipe
In the search for the perfect combination of agile and outsourcing, we must first go back to the intentions of the founders of agile. In 2001, in the winter sport resort of Snowbird in Utah, seventeen software experts from various background – including the Dutchman Arie van Bennekum - laid the foundation for the *Manifesto for Agile Software Development*. Four core values were formulated, which subsequently evolved into the twelve famous agile principles. It is precisely those four core values developed at the beginning that now offer valuable starting points for bridging the gap between agile and outsourcing.

Whoever crosses that bridge must remember that it will be different on the other side than people are used to in outsourcing. *'Becoming agile'* without organizational change is a pointless exercise: putting an agile sticker on outsourcing is not enough to achieve results.

## Bridges between core values and outsourcing
### 1. People and their cooperation make the difference, not processes and tools
The first core value of agile is that "individuals and interactions (between people) are more important than processes and tools". It is about prioritizing intrinsic motivation: professionals doing the right thing together without the influence of external incentives, such as a monetary reward or the threat of punishment, and without rigid processes.

Classical outsourcing often involves perverse incentives and noise, not least due to the adherence to far-reaching details in the contract, down to the requirements level, where the details are overtaken by reality shortly after signing the contract. Also, in outsourcing, following the pre-scribed process is often more important than doing the right thing. This behaviour is caused by fear of penalties.

Paradoxically, to gain traction, you must first relinquish control. This is only possible when people feel responsible for customer value and are proud of the result.
Outsourcing can only be successful when the people who design, manage and maintain IT, understand the true impact of IT on other people's work and lives. These are the business, their (end) users, and the end customers. By removing the noise of a traditional management organization and dealing with inappropriate objectives (and KPIs) that do not contribute to value creation, people and their cooperation become the driving force in outsourcing. In practice, therefore, you see a trend towards team-based working, in which the customer contributes the necessary business knowledge and the change mandate (in the role of product owner), and the service provider contributes the technical competencies.

In outsourcing with agile, control is kept to a minimum by decentralizing responsibilities and investing in the teams. This removes much of the noise because the value is central. Classic reports that mainly look back have been replaced by dashboards that provide the team – and the team's environment – with continuous insight into the current situation so that timely adjust-ments are possible. The focus on the co-creation of value is central, developed in a collaboration of experts who know, can, and may – and therefore have a mandate.

### 2. A product that works, is more important than redundant documentation
The second core value of agile is: "working soft-ware is more important than extensive documen-tation". Since almost everything will be software-defined, you can freely translate this into "focus on making a working (and evolving) product rather than creating documents".

In traditional outsourcing, documents are the starting point: all requirements must be recorded in detail at the start of the relationship. Then you are dealing with a kind of Big Design Up Front (BDUF), where all requirements are predefined, documented and contracted. We now know that these predefined 'end products' are never what is needed. The term 'detailed' is often difficult to

## THE FOCUS SHOULD BE ON VALUE CREATION, NOT ON BICKERING ABOUT REQUIREMENTS SET IN STONE.

apply in our current world: changes are a given, also in the short and medium-term. Details often serve to strive for a certain degree of perfection, namely that the paper design and the finished product are identical. But in practice, by the time you achieve your paper perfection, the world has changed a lot. Working based on a detailed design also takes much longer because obstacles in the development process must not lead to deviations. As a result, it takes longer before the customer receives his product, limiting the possibility of rapid adjustment or innovation. Long delivery times are obsolete. The cost of delay for new func-tionality can determine success or bankruptcy.

There is another type of perfection that is essen-tial these days. Software must meet non-func-tional requirements such as security, reliability, availability, scalability, user-friendliness, and compliance. This is independent of the functional requirements: the code is never finished and ideally moves with changing needs.

Back to the documentation fetish: in traditional outsourcing, the love for details leads to an enormous documentation flow. This is the direct result of the chosen method and has little to do with the end goal. Agile outsourcing is about-documenting differently and, above all, less. Instead, issues such as portability, integratability, and controllability of the code (or services) are handled through extensive automation. Instead of functional perfection on paper, we are moving towards automated code perfection.

### 3. During the production, cooperation is the basis of the relationship and not a contract debate
The third core value from the *Manifesto for Agile Software Development* is that "conversation with customers is more important than contract negotiations". In other words, the focus should be on value creation and not on bickering about requirements set in stone.

Traditional outsourcing works with unshakable SLAs, boarded-up statements of work (SOWs), and project requirements cast in concrete. That leaves little room for a healthy conflict. Letting go of certainties is the first step to realizing that as a customer, you never know in advance what

exactly you want. Dealing with customers who don't know what they want is a challenge, but it has been precisely the proven strength of agile for many years.

In 2001, the founders of the *Manifesto for Agile Software Development* had considered the term *conversational software development* instead of the *catchy* word agile. Continuous feedback and dialogue are often lacking in outsourcing relationships, while 'conversations' form the basis of successful collaboration.

Conversations with the customer are essential for developers and engineers to discover what the customer needs. So they increasingly work within teams that have a clear relationship with the business. From a management perspective, contracts remain important: business and legal agreements for the parties involved. However, the question is: what are you negotiating, and what is essential? As a client, you may not know precisely what you want in terms of functional requirements, but you need to know exactly what you want (and can) from the point of view of budget, collaboration, and especially value creation. If you cannot articulate and record this, agile outsourcing also becomes a swamp in which you slowly sink. The result: the business that, after ten sprints, wonders when it will get something that works.

#### 4. Being able to handle change leads to more value than sticking to schedules

The fourth core value is: "Reacting to change is more important than following a plan". The counterpart is "following a plan is more important than moving with every gust of wind".

Outsourcers often accuse service providers of being insufficiently responsive to change. Service providers counter this by stating that customers already want something else before the first results are achieved. In classical outsourcing, you often encounter an unhealthy fixation on absolute

time, for example, during a transition and when reaching the future mode of operations. There are severe penalties for failing to meet deadlines. The result is finger-pointing when the kitchen timer goes off and the work is not finished. An additional complication: at the time of delivery, the delivered product is often outdated. Double bad luck: the result comes too late and is not to your liking.

An organization that wants to work agile no longer talks about deadlines. If you're going to have something ready in the future, you break the bigger picture into chunks until a schedule is created based on a series of short deliveries *(sprints or timeboxing)* that both the team and the customer agree to. Because the experts plan the work themselves and the teams are becoming increasingly accurate in estimating the amount of work they can handle, you can assume that the end goal of the sprint (not the deadline) will be achieved.

In agile, we no longer only talk about deadlines or about 'projects': a synonym for a temporary organization that is dissolved after delivery. Knowledge loss is a significant pain point that every organization has in outsourcing and other relationships between contract parties. When a product or value stream (a collection of activities required to deliver a complete product or service) changes, so does the backlog — the list of unfinished work that has been prioritized. This method is complex for traditional outsourcing organizations because developers are usually hired for a certain period and for a specific part, such as just building a system. That is why we see that customers are increasingly looking at outsourcing constructions in which teams work on both construction and management, a construction that results from the rise of DevOps teams. The service provider is then responsible for the continuity of the team — including the necessary knowledge assurance — that works on a system or value stream for a more extended period.

# LONG DELIVERY TIMES ARE OBSOLETE. THE COST OF DELAY FOR NEW FUNCTIONALITY CAN DETERMINE SUCCESS OR BANKRUPTCY.

# IN PRACTICE, BY THE TIME YOU HAVE ACHIEVED YOUR PAPER PERFECTION, THE WORLD HAS CHANGED A LOT.

## What can we do with these four core values?

Drawing up a new recipe for outsourcing comes down to letting go of the classic control paradigm: wanting to determine everything yourself in advance, making plans, and then micro-managing all the service provider's steps.

Agile in outsourcing is about managing multi- and interdisciplinary collaboration, in which the professionals of both customer and service provider(s) take joint responsibility for realizing customer value. The experts with a shared vision – or even better: a shared product goal – are in the lead.

By making the value creation measurable, visible and tangible in a transparent production process, the trust is created to saachieve the desired

results for the business, the users, and ultimately the customer's customer. This is the step from output to outcome and eventually to monetized impact. Focus on outcome (success for the customer) and monetized impact (the financial constraint for a win-win) argue for more emphasis on goals and less on a vision of an idealised end state. Artificially merging agile and outsourcing does not work.

Begin by living the four core values: thinking about minimal viable agile based on the original idea. Anyone who works with this will discover that chocolate with mayonnaise can be quite tasty. We highly recommend chocolate-mayonnaise cake. ◄

☑ VALUE HORIZON FOR DIGITAL TRANSFORMATION © GIARTE

| **Output tells the story of what you make and deliver** | **Outcome tells the story of the added value of the output for the customer** | **Monetized Performance Goals tells the story of converting value into money** |
|---|---|---|
| Vital signs on the quality and efficiency of the digital product | Vital signs on performance improvement and new possibilities | Vital signs on the financial health and future of a company |
| **OUTPUT** | **OUTCOME** | **MONETIZED PERFORMANCE GOALS** |
| ↓ | ↓ | ↓ |
| Reliability<br>Availability<br>Scalability<br>Costs<br>Skills | Time to Market<br>Ability to Innovate<br>Productivity<br>Business Resilience<br>Customer Experience | Profit<br>Share of Wallet<br>Cashflow<br>Exponential growth<br>Valuation |

# KVIs AND KPIs AS THE NEW YIN AND YANG

TEXTS MARCO GIANOTTEN

Traditionally, IT works with *key performance indicators* (KPIs). Agile works with *key value indicators* (KVIs). What is the difference between KPIs and KPIs? And can KPIs and KVIs coexist? In outsourcing with agile, the topic 'KPI versus KVI' will come up sooner or later. Hence this short guide.

*Key performance indicators* (KPIs) are a standard control tool in companies. They help monitor various processes in all kinds of disciplines:
- Finance (turnover, profit margin, cash conversion cycle)
- Sales (sales qualified leads, customer lifetime value, net promoter score)
- Production processes (on-time delivery, unplanned downtime, non-conformance)

KPIs are also the most normal thing in the world of IT. Traditionally, KPIs in IT aim at managing output. They relate to the numerical properties of the digital product or service. Output is what you make or deliver. The problem is that KPIs are often not about the real impact of IT on the business and the people (employees, customers, chain partners). In the worst case, KPIs are the goal that justifies the means, and the watermelon effect is created: the KPIs are green in the reports (outside), but on the shop floor and with management in the business, the signals are red (inside). Especially with outsourcing, KPIs can become counterproductive.

## KVIs are about customer impact

Creating value is central to agile: everything revolves around *key value indicators* (KVIs). A KVI is an actual number that represents the impact of their actions on a team. In other words, KVIs focus on outcomes: what you realize or achieve in the customer's eyes. In agile, each team (or group of teams) has one KVI. For a game team, a KVI can be the number of daily active users. For a help-desk team, the KVI is, for example, the percentage of customer issues resolved. For a webshop team, the conversion rate is the KVI, and for a team that works on online learning, the KVI is, for example, the student achievement rate. The KVI is discussed periodically in the (usually biweekly) retrospective.

An essential difference with KPIs is that KVIs measure values that often cannot be directly translated into financial deals, such as costs or net profit. On the other hand, a value-oriented KVI is mainly future-oriented, while a KPI specifically looks back – or at the most at the performance of the moment. The most significant difference between KPIs and KVIs is that a KVI is always about customer impact.

### Side by side
In organizations where some teams are agile and some are not, two indicators can be used side by side. You can work perfectly well in an organization with KVIs and KPIs, as long as the whole remains clear and the two do not conflict. Here are two examples of potential conflicts and a possible solution.

### Complementaire KPI's en KVI's
When combining outsourcing with agile, stress testing your KPIs ('tickle your KPIs' is a map from Giarte's XLA method, see box) is an excellent method to arrive at KPIs and KVIs that are complementary. The XLA framework assumes that the interplay of KPIs and KVIs helps build a bridge between output-oriented and value-oriented operations. Or between 'doing what is asked' and 'making a difference'. Or between teams that work agile and 'the rest of the company'. Note that the KPI will still play a role in this, in monitoring operational processes. The extra step towards value-oriented operation is the real turning point. For those who need an extra hint: achieving a KPI ensures a green score, achieving a KVI ensures pride and confidence in the future.

An excellent example of a KVI for mission-critical IT is the available uptime of a service. For KLM, for example, this means 'no flight delays due to IT'. Relevant KPIs for the service providers are, for example, the reduction of incidents with the exact root cause, or 'zero repeat' and the reduction of 'major incidents resulting from change'. In this way, you have forces that are not contradictory but that complement each other. ◄

## XPERIENCE LEVEL AGREEMENT

The xperience level agreement® (XLA) has started the fight against context-free and meaningless KPIs. With the rise of agile and its KVIs, XLA helps in two ways: on the one hand, in bringing harmony between KVIs and KPIs, and on the other hand in the design of sentiment metrics for the user experience (UX), which consists of customer experience and employee experience.
An XLA is a commitment to pursue a predefined desired experience. This experience is made tangible and measurable using xperience indicators (XIs). These XIs represent an indication of the experience. To fully benefit from an XLA, not only sentiment data but also objective data is needed. XIs must therefore be combined with objective measures such as page load time, hang time, and webpage errors. It is precisely the combination of sentiment data and accurate data that provides a better understanding (correlation) and guidance (what to improve) for managing UX.

## DEFECT

KPIs and KVIs are used within the same organization, leading to a 'short circuit'. Take a large multinational where the IT organization directs their service provider to deploy IT as cheaply as possible for new employees. This includes KPIs such as delivery times of notebooks, the costs per digital workplace, and the speed with which the service desk processes tickets. At the same time, HR is working with an onboarding team in agile mode to successfully onboard the new talent needed for the company's digital transformation. The HR team uses the KVI 'happily onboarded': the percentage of newcomers who feel welcome in the first phase of their new job. IT plays a vital role in this.

However, it turns out that many newcomers within this multinational complain that they have been given the wrong notebook, that it takes a long time before they can do their work correctly, and that they constantly have to call the service desk, which is not at all welcoming. Some employees have experienced this so negatively that they have started looking for another job. This turnover ultimately costs the company more direct costs (replacement) and indirect costs (reputational damage) than the increase in IT costs per new employee would cost the company – a consequence of the lack of a value-oriented IT approach.

**Which value do you prioritize: the KVI 'happily onboarded employees' or the KPI 'the cheapest possible digital workplace'?**

## TRADE-OFF-SCENARIO

Sometimes an organization can end up in a trade-off scenario, as with a Dutch telecom provider. To sell a new mobile subscription online or in the store, the telecom provider needs information about the customer. For example, data about outstanding accounts with other providers, the maximum number of subscriptions allowed at a home address, and a proof of identity. The telecom provider uses a system for this, the customer acceptance module (CAM). Because not enough was spent on the maintenance of the CAM, it is regularly out of order, so that the verification process often fails.

The business is willing to invest in new systems but spends little on maintenance. A meaningful dialogue has arisen between IT and the business by reporting not on output (availability percentage) but outcome (lost turnover in euros). This gave the business insight into the value of maintenance: the loss of turnover due to lack of availability amounted to 1.7 million euros per month. Therefore, the telecom provider uses the KVI 'no lost sales' for the CAM team and, at the same time, as an outcome-oriented KPI for the business.

# THE NEW WAY OF CONTRACTING

TEXTS MARCO GIANOTTEN

IT outsourcing is based on an implicit essential conviction: 'customers lie, and suppliers are unreliable'. Distrust is the unconscious modus operandi. After each new trauma, the parties increasingly tighten the strings. The result is Verschlimmbesserung, a stream of intended improvements that ultimately lead to deterioration. It is a vicious circle that has significant consequences for contracting: constructing a prelude to warfare. How agile break this vicious circle?

In the world of IT, both technology (due to advancing innovations) and requirements (due to changing needs from the business and end-users) are changing at an ever-increasing pace. When you freeze everything in the hunt for certainties in outsourcing and exclude every unpredictable change, you throw agility under the bus. That is why parties are looking for solutions to get out of this swamp.

Different names are circulating for 'the new contracting': vested outsourcing, conscious contracting, and agile contracting. These are various forms of the same basic belief system: providing guidance for solving problems by steering on value.

### Agile working and outsourcing
Agile stands for open and honest communication, a condition for being able to trust each other. That

# WHEN YOU FREEZE EVERYTHING IN THE HUNT FOR CERTAINTIES IN OUTSOURCING AND EXCLUDE EVERY UNPREDICTABLE CHANGE, YOU THROW AGILITY UNDER THE BUS.

same trust is also a condition for entering into the creative conflict: discussions based on mutual respect, interest, and enthusiasm in the interest of the best course and the best solution(s). It has therefore been a long-held belief that agile is incompatible with outsourcing. From an agile perspective, a hard split between client and contractor does not achieve the desired collaboration. That led to the argument that agile and outsourcing would be incompatible — just as 'gross' as chocolate and mayonnaise.

Since many organizations cannot recruit enough IT people and offer them an attractive career path, they need third parties to provide a workforce. This involves the assured delivery of individuals (role, training, certification, skills) or even complete multidisciplinary teams that then work for a product owner employed by the customer. This is a 'light form' of contracting, where the customer works with people who are not on their payroll.

## Adjust the contract to fit the collaboration
Many organizations claim to work agile, but their contracting method demonstrates otherwise. The contract must therefore be adapted to the desired collaboration method – and not the other way around. Then it is crucial to start by defining performance management. Not from opposing interests but common goals. Not from a short-term perspective but a long-term relationship. Something relatively simple like collaborative KPIs – which can only be achieved if parties work well together and do not clear their path with 'yes, but' for every obstacle – helps connect people, gain commitment and a chieve a productive collaboration.

Something similar can arise by focusing on subjective experiences, such as the customer effort score or team spirit. 'Esprit de corps' stands for

morality, in other words the enthusiasm, pride, and perseverance of the people within a team. Also, think of employee satisfaction as something to monitor; after all, both parties supply people for the joint teams. Furthermore, it is mainly the process of jointly defining indicators that reflects the collaboration, especially the discipline within it. 'Working differently' starts with strictly following the agreement. Varying is possible later, after a joint retrospective. It's like preparing a new recipe from someone else's cookbook: start by doing it exactly as described. You can always vary afterward based on your insight.

## Live together first, then get married
Make contracting easier by first establishing a proof of value (POV). A POV goes one step further than a proof of concept (POC) that demonstrates technical feasibility. A POV addresses the solution's value to the business so that investment can be justified and success measured. The contract is then a 'two-stage rocket': first the POV and then the rest. The initial agreement can already address the entire contract: common assumptions in terms of value, implementation risks, human resources, and costs. An approximate price is determined based on these assumptions. This price is not yet binding.

This requires that the purchasing fixation on 'assumption is reality' must be released. Often the customer has already formed an image for the transformation, with timelines and a budget. This means that you are immediately fighting against timelines that are not realistic, and budgets that are being eaten up by scarcity. The POV serves as a verification phase, after which the parties share their experiences and compare them with previous assumptions. Only then will it be decided how the conditions will be fixed for the entire project. Flexibility (agility) is essential for

value creation within a fixed budget framework. This 'living together before you get married' increases the chance that supplier and customer will become happy together.

## Thinking from a product that brings value

The most effective way to contract agile is fixed resourcing for an extended period. This can be scaled up or down based on priorities per product. Priorities are always about value for the business. It is unacceptable that someone shouts that 'it can be done with fewer people' and then randomly cuts into the resourcing without considering an exploding backlog. What happens then is that less and less value is added and that the work increasingly consists of putting out fires.

The sourcing of talent and teams are forms of managing input. As a customer, you determine precisely what, when, and by whom (if several suppliers are involved) is required. The supplier determines how delivery is made. Managing input is the simplest but cannot be determined in isolation from purchasing. You can't limit the actual amount of work by going for the lowest price and a 'promise' of OTOB (on time, on budget). Everything is often promised to win the contract ('comply or die'), only to point out tiny discrepancies after the signature. Then the process turns out to be over time and over budget.

## More skin in the game

By only steering on input, the service provider has no skin in the game, for example, to deliver an increasingly stable velocity or to bear co-responsibility for implementing a solution that is important to the business based on on-time on-budget. When managing input, conflicting interests can arise. The supplier wants to charge hours over a more extended period at the maximum achievable market price; the customer wants quality, speed, and reliability at the lowest possible market price. But who bears the risks? And who benefits from the realized value? If you want to go a step further than managing input, you can't do this without pain-gain sharing. What principles are there for 'win-win by design'? This quickly leads to securing output and outcome.

## Output and outcome

Output is about the quality and price of the product itself. Output is, for example, the increasingly efficient and predictable delivery of story points, the reduction of waste (disruptions, bugs, non-compliance), and a higher degree of automation. When managing output, a supplier also benefits from an increasingly better product for the customer.

Managing outcome goes a step further than managing output. Management based on outcome is about the actual economic impact of the product on the customer's business. For example, outcome is about increasing customer loyalty or market share by bringing new functionality to the market faster than the competition.

There is a strong relationship between key-value indicators (KVIs) with output and outcome: which value do you add that is measurable in every retrospective perspective? If the supplier wants to contract on output or outcome, it is required that the supplier has sufficient experience in translating his input into output or outcome. Please note: this requires stable teams and a genuine collaborative relationship. When the business' participation is not sufficient, when there is a product owner who cannot make value decisions, or when the customer organization is insufficiently involved in terms of participation and feedback, a supplier who is paid based on outcome will not be able to create a viable business case.

## MOVING IN TOGETHER BEFORE GETTING MARRIED INCREASES THE CHANCE THAT SUPPLIER AND CUSTOMER WILL BE HAPPY TOGETHER.

# IF YOU WANT TO GO A STEP FURTHER THAN STEERING ON INPUT, YOU CAN'T DO THIS WITHOUT *PAIN-GAIN SHARING*.

## T2M and A2I as a guide for value

When there is a business case for time to market (T2M) in the collaboration, this can be included in the contractual agreements. T2M is about the speed at which value is delivered. This is also known as time to value (T2V). When delivering on time, for example, a payment of 100% of the agreed amount applies. If delivery is made sooner, a bonus is paid (e.g. 20% of the agreed amount), and in case of late delivery, a discount is calculated (for example, only 80% of the agreed amount has to be paid). The more economic value can be attached to T2M, the more pain-gain can be agreed to guarantee the outcome.

In addition to T2M, agreements can also be made about the ability to innovate(A2I), which looks at maximizing the organizational capacity to absorb and benefit from new technological innovations. Many opportunities cannot be exploited because the customer gets stuck in old technology. A2I can reduce technical debt when this leads to improvements in the operating model (cheaper, faster, more agile) or the business model (new and more stable revenues). In economic terms, A2I is about creative destruction and balancing sunk costs with opportunity costs.

## Ability to Innovate

A concrete model for increasing A2I is dividing the work (e.g., into hours or production story points) into a two-by-two quadrant along the axes "now – future" and "conservation – innovation". Now and conservation together are about the percentage of work spent on reactive problem-solving. This is necessary but does not create value. Maintenance and future together are all about keeping the systems healthy through preventive care. These two quadrants are 'under the hood' and are not visible as value to the business. Above the hood, there are also two quadrants.
The first is now and innovation with incremental improvements for the business with changes. The second is future and innovation with strategic changes such as moving to new platforms that are a significant leap forward for the industry.
The division of work (hours or story points) provides guidance on what is needed (such as making up for maintenance backlogs) and what the parties are striving for (such as a switch to target architecture). In this way, A2I can be framed for the agreements.

# 'WORKING DIFFERENTLY' STARTS WITH STRICTLY FOLLOWING THE AGREEMENT. VARYING IS POSSIBLE LATER AFTER A JOINT RETROSPECTIVE.

The business often wants to move to the 'future – innovation' quadrant as quickly as possible, without sufficiently realizing that it takes time and money to transform the application crown jewels. If there is a relatively large amount of overdue maintenance, it is crucial to first invest in the 'future – conservation' quadrant. This is the big clean-up that is often overlooked in budgetary terms: when allocating money, the focus is on the new world, and the budgets for the old world are cut sharply. Then you see that at some point, the work in the quadrant 'now - conservation' explodes with corrective maintenance. From an agile perspective, it is simple: the wrong priorities are set for the backlog.

## Conclusion

More is possible than just hiring staff when a customer organization embraces agile and has to contract third parties. In addition to managing input, output and outcome are also possible as a basis for contracting. Actual agile working in outsourcing starts with realizing that you no longer have to think in terms of projects with start dates, end dates, and fixed prices. You have to remain agile and keep evaluating your priorities to deliver valuable software and services. ◁

ABILITY TO INNOVATE (A2I)
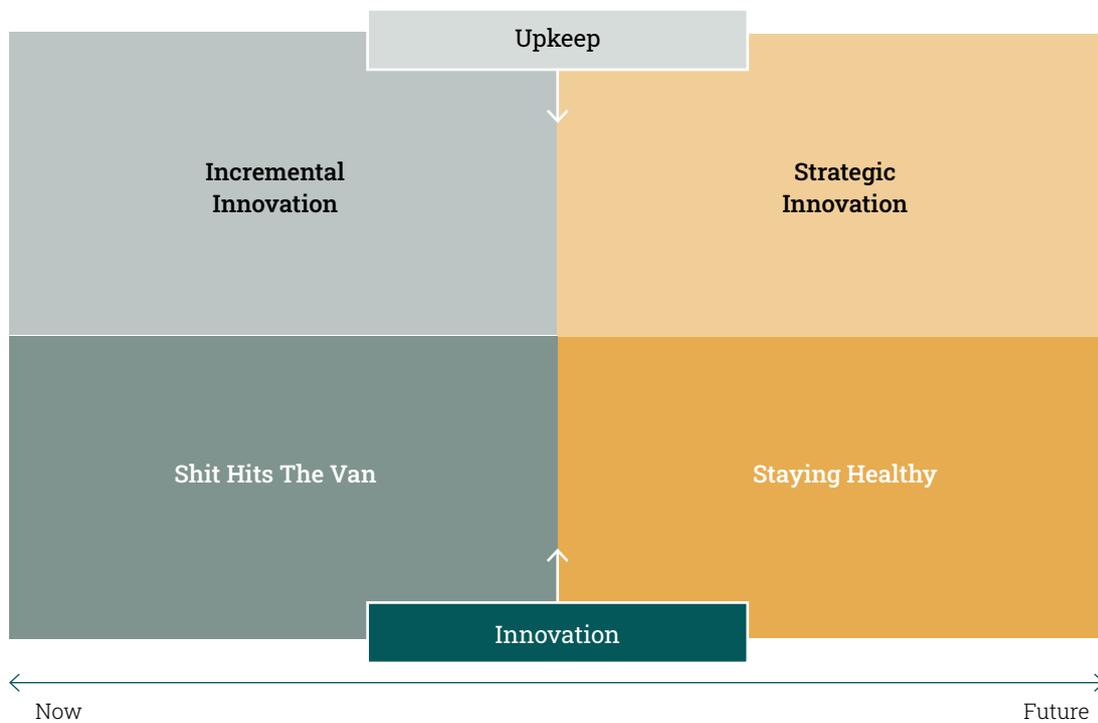
GIARTE  Master the art of a
         perfect IT experience