Project Business Management[1, 2]

Contracting Models for Agile Projects

Antje Lehmann-Benz

> *"Effectively, change is almost impossible without industry-wide collaboration, cooperation, and consensus."*
> - Simon Mainwaring

Summary

When companies come together to do projects as customers, contractors, and in other roles, it is high-risk business for all parties involved. Often, this concerns agile projects as well.

Most agile methods and techniques pay little regard to cross-corporate team setups. Instead, they seem to have been described mostly for company-internal

[1] This is the 15th in a series of articles based on the book by Oliver Lehmann, *"Project Business Management"* (ISBN 978-1138197503), published by Auerbach / Taylor & Francis in 2018. See the author's profile at the end of this article.

[2] How to cite this article: Lehmann-Benz, A. (2019). Contracting Models for Agile Projects, *PM World Journal*, Vol. VIII, Issue II (February).

projects. However, the fourth value of the Agile Manifesto does urge us to consider "customer collaboration over contract negotiation". With more and more project-related work outsourced, this addresses an increasingly undeniable reality teams face. Does this not mean that we should take a closer look at this often ignored aspect of agile project work?

The following article intends to clarify some facts around cross-corporate setups for agile teams in general and give suggestions for possible contract models.

Case Study

Sweet Taffy Inc. was developing a new HR software interface for their customer Cold Wax Ltd., a large car manufacturer with rigid procurement processes. They had agreed on a fixed price, and at this time, many unexpected problems had come up that made them invest much more effort than anticipated at the beginning. Sweet Taffy was looking at a potential loss situation, and to make matters worse, the customer had recently made deadlines tighter. When Sweet Taffy raised concerns about being able to meet this deadline given the unresolved issues, the responsible manager at Cold Wax told them: "Well, you're probably not agile enough? Just have your development team work more agile and you will be faster."

Overview: Agile Methods

Unfortunately, the situation described above is extremely common. In reality, of course, things are much more complex than that. Agile methods were not developed simply to deliver results faster. Instead, contractual models have to reflect the necessities of a given project situation.

In 1997, Ken Schwaber introduced the framework that he and Jeff Sutherland had recently developed at a conference for object-oriented programming called OOPSLA (Schwaber, 1995). The framework took some ideas from Lean manufacturing (Jansson, 2017) and translated them from production to (software development) project environments. Also, new ideas were added to accommodate for this environment's realities and with hopes to mitigate the pain points many in the domain felt every day: Processes were too heavyweight for the complexity and uncertainties they faced, communication problems and disputes arose, features were developed that never got used later, reaction times to changing circumstances were slow, money and time was wasted, and teams felt they could not be creative or responsible for their own results. Schwaber and Sutherland hoped that by applying the principles and values of the Scrum framework and promoting its use also to stakeholders, while requiring a certain amount of discipline from the people in the project, such issues could be addressed more effectively.

The introduction of the Scrum framework marked the start of a series of events: In 2000, Kent Beck published his book "Extreme Programming Explained – Embrace Change" (Beck &

Andres, 2000), and one year later, Beck, Schwaber, and Sutherland met with several others who had also been active with developing more "lightweight" management methods for software projects. They all had in common that they felt traditional project management tools were often too heavyweight and hard to adapt to the software domain. After all, this domain had been growing more and more complex, and still is to this day. Although as a tendency, complexity is increasing across all industries, software development experts were among the first people to experience this first-hand.

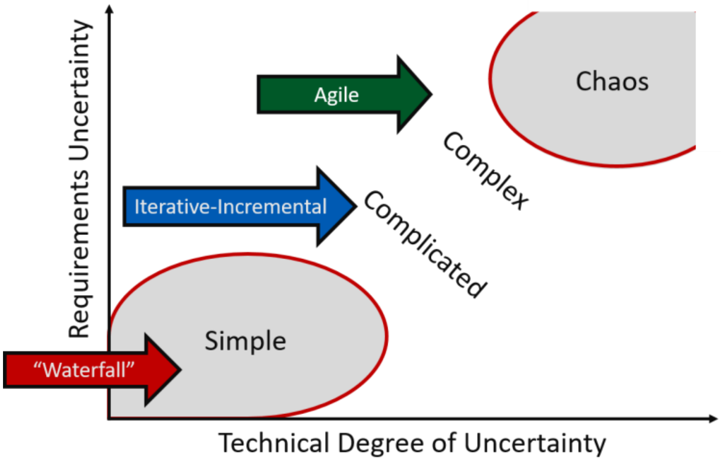

Figure 1: The Cynefin framework was used as a basis for this graphic. It attempts to help choose a suitable project management approach according to the uncertainty regarding technology and requirements in a given project situation (Wikipedia, 2019).

When the rather diverse participants of the gathering met in the winter of 2001 in a skiing resort in Snowbird, Utah, many of them thought this would not lead to "anything substantive", as Alistair Cockburn put it (Beck et al., 2001). After all, they'd met before to loosely network and share and exchange ideas for lightweight software development processes. To call methods that would reflect these ideas "agile" did not go undiscussed—but in the end, not only that name was decided upon, but also something very substantive was indeed published and came to be known as the Agile Manifesto (Beck et al., 2001).

It consists of four main values and twelve underlying principles that have often been referenced since and hit a nerve in the software community. It could be referred to as nothing less than the start of a new era in software development, valuing

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

In essence, one could sum up the goal of reaching more "agility" as having mostly autonomous ("self-organized") teams with a focus on business needs that own their (preferably lightweight) processes and are responsible for them (Pieper & Roock, 2017, p. 2).

Of these four, the third value is not relevant in purely internal projects, where there are no contracts. But it's crucial in customer projects, and often forgotten or misinterpreted. This has even led to situations where project collaborators agreed not to have written contract because they thought those were not needed when you're just being "agile" enough.

Written Contracts

Many people are not aware of the fact that oral agreements can also form a type of contract, but one that will be hard to enforce in case of disputes. To have a good, balanced position, it makes sense to have a written contract signed by both parties, and to store this document in a way that makes it easy to refer to if necessary.

If the Agile Manifesto says customer collaboration is valued more than contract collaboration, this does not mean there should not be any contract (note also the remark below the four values stating "That is, while there is value in the items on the right, we value the items on the left more.")

The first of the 12 principles makes more references to customer projects, when it states: "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."

Unfortunately, the Manifesto stops there and has since left many people in the dark about how to

- set up project teams across organizations and
- design contracts in a way that pays respect to the way agile teams collaborate.

Many people have come to the conclusion that cross-organizational project collaboration makes it impossible to be "agile" and that you cannot apply agile project management methods in such situations, but only in purely internal projects[3]. If this was true, however, why would the fourth Agile Manifesto value and the first principle have been formulated? Moreover, the decision to outsource project work and project management tasks is getting more common, not less (Lehmann, 2018). This reality has to be faced in any case.

[3] For more on the difference between these projects types and how important it is to consider project typologies in general, see (Lehmann, 2018, p. 11) and (Lehmann, 2016).

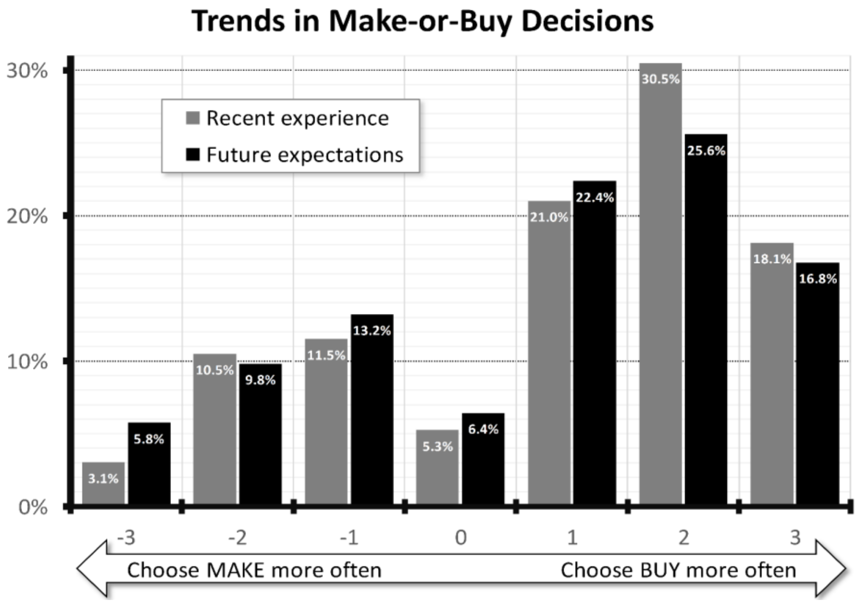

Figure 2: A survey made in 12-2016 with 590 respondents showed that there is a clear trend towards BUY decisions in project business (Lehmann, 2018).

It makes sense to look at possible contract models that can work in such setups—but before, we should look at possible roles in an agile project.

Roles and Contracts in Agile Projects

Many agile frameworks come with suggestions for how project roles should look like. Some more traditional approaches simply suggest a project manager with an agile mindset and keeping roles according to job titles. Others are a bit more radical and suggest that project management responsibilities be distributed across various new agile team roles. The most prominent example for this is the team setup in the Scrum Framework:

In a Scrum Team according to the Scrum Guide (Schwaber & Sutherland, 2017), there is a Product Owner, a cross-functional and self-organized Development Team consisting of 3-9 developers, and a Scrum Master.

Product Owners manage the Product Backlog as the single source of requirements. *Scrum Masters* make sure Scrum has been understood and help the team and the organization overcome project-related and other impediments. *Developers* pull requirements from the Product Backlog each Sprint and create the results.

Lastly, *stakeholders*—in the Scrum understanding—are all other people with interest in and requirements to the product or result being developed who are outside the Scrum Team. Therefore, this can include anyone from sponsor to business representative to customer to end user.

Note: This is describing a one-team setup. When working on larger projects, try to scale up the development teams as needed and have them align regularly on possible dependency issues. These might grow when more than one company is involved. There are various scaling frameworks for guidance (Heusser, 2015).

When drafting a contract for projects where Scrum is applied partially or to full extent, two main challenges arise:

1. The stakeholders have to be known, involved, and considered sufficiently in the contract. This is to that someone who feels overlooked or ignored can affect the project negatively. Classical project stakeholder management (Wikipedia, 2019) offers many ideas and techniques to do this effectively.

2. It has to be clear which company provides which roles for the Scrum Team.

As for 2, some people advocate that Product Owner and Development Team always be from the same company: Their concern is that otherwise, loyalty conflicts can erupt. This can be the case if the Product Owner feels the need to protect the interests of his or her company only and uses the Development Team like a commodity.

Instead, however, good Product Owners should protect the team and their optimal throughput rate of creating high-quality and highly valuable results by prioritizing and rejecting new requirements from entering the Product Backlog[4].

When looking at projects in real-life, these conflict of interests do indeed happen on a daily basis in such team setups, that is with a company border between Product Owner and Development Team.

A recent example is a Scrum team in a large German automotive company that after every Sprint Planning meeting experienced a backlash: Managers from the customer company said they did not care about their recently created Sprint Backlog; other things had come up and were more important now. The Product Owner, being subordinate to these managers, simply passed this information on to the team. The Scrum Master felt like no one was listenting to him when he tried to intervene in this scenario. As a result, the team became unmotivated to do any Sprint Planning at all, and the entire project slowly went downhill.

However, other people say that they have achieved great successes with the customer company providing the Product Owner and the contractor providing the Development Team (Pieper & Roock, 2017, p. 36). Some topics will have to be clarified to avoid situations as described before:

- Mutual interests and a partnership in “Good Faith” have to be agreed upon from the start and in a written contract—it should be clear that the goal is to achieve a

[4] The list of everything deemed needed in the product under development and the single source of requirements for the team

common mission which with everyone can identify. A mission in the interest of all involved parties.

- It should be clear to everyone what working with agile methods means and what mindset has to be present—minimizing "bottom-down" decisions and micro-management, empowering the Scrum Team which should feel as a whole despite company borders between them.

- The Product Owner and stakeholders from the customer company should be part of the Sprint Planning meeting and have a clear understanding of the intended Sprint content. This can even be formalized in the contract (see Sprint-based contract model below).

- The Development Team should feel empowered to be self-organized, but also understand that this comes with a certain responsibility. They should be able to speak openly and share concerns, but in a constructive, positive manner with the clear intent to make the project a success. After all, they are the technical experts.

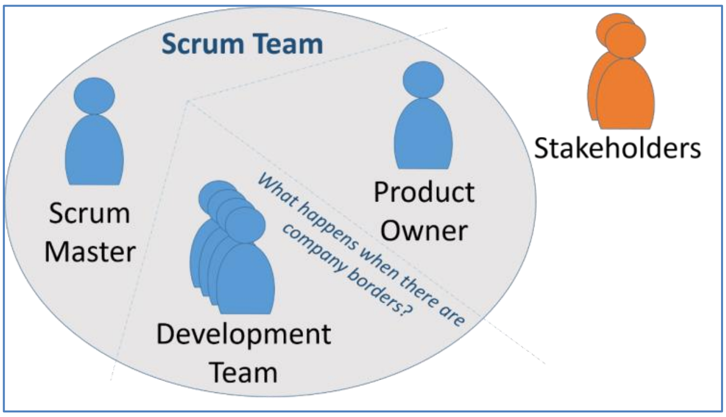

Figure 3: Company borders pose natural challenges to Scrum team setups—such borders are extremely common, however.

Legal Foundations for Project Contracting

Legal Systems

Whenever legal aspects are concerned, it is important to know in which law system a company operates. This can influence everything they do in contractual environments, even on a subconscious level. Sometimes, lawyers themselves are not aware of how significant the differences in thinking stemming from different legal systems can be.

The two most influential legal systems in the world are Common Law and Civil Law (see picture).

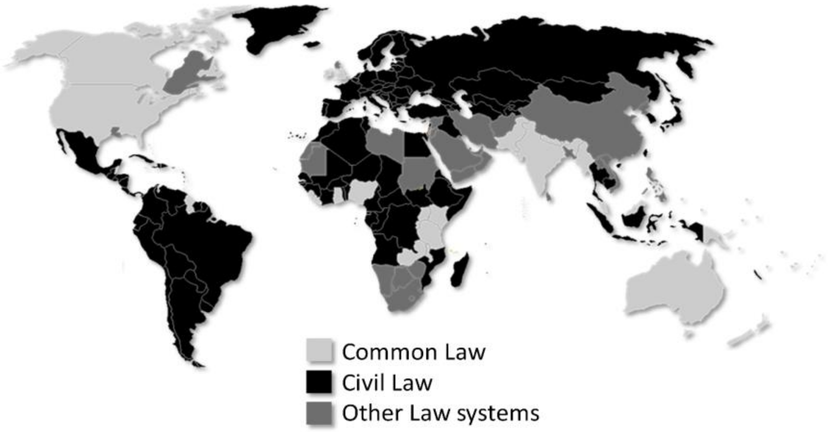

Figure 4: Distribution of legal systems across the globe (Lehmann, 2018, p. 163)

The doctrine of "*good faith*" is important in both legal systems, but understood a bit differently. It refers to the mutual understanding of trust between companies and the fact that no party will willingly take advantage of the other but instead form a partnership at least for the duration of the project.

This is less immanent in contracts based on common law than it is in those in civil law environments. Therefore, a project contract in common law has to contain all kinds of provisions and consider eventualities not needed that much in civil law, where judges will always assume good faith principles when evaluating a contract. People working in projects with such typical long common law contracts are confronted with the challenge to know what's in these contracts and how to act in compliance with it.

Figure 5: Contracts can turn into large piles of paper that become hard to handle. This is a problem especially in some common law countries

In civil law countries, contract parties have to consider that they will be protected in legal disputes based on the assumption that a partnership in Good Faith was at least the original intention behind their undertaking. This can make it harder to protect own interests if they are against possible interests of the other party.

Before drafting and especially before signing any project contract, therefore, it is important to decide upon the applicable legal system and local laws—if companies from various countries are involved. If there is no decision because both contract parties are from the same area, it is still important to be aware of the peculiarities and characteristics of the applicable law.

In Germany and some other European countries, for example, one should be aware of the general categorization of contracts in result-oriented, service-oriented and other forms. This categorization can have a big impact on how judges make decisions in case of disputes, and it is up to a judge's interpretation of the nature of the contract, even if the contract states to be a certain type (Pieper & Roock, 2017, p. 52).

Possible Contract Models for Agile Projects

1. Fixed Price Models

Fixed-price contracts minimize the customer's risk of having to pay for cost overruns. There are several problems with these models, including:

1. They cannot address risks of schedule delay or that a product gets developed that doesn't serve the customer much in terms of usage value.

2. While applying agile methods is actively addressing the maximization of usage value, one issue that stays is that it is extremely hard to find a price when the full functionality isn't defined yet—a totally typical situation in agile projects, if not the very reason why agile techniques are used in the first place.

3. A fixed-price contract often merely defines the development stage and doesn't consider what comes after (maintenance, ongoing development).

The following flow of documents is often used in contracting processes, especially in very classical fixed-price environments:

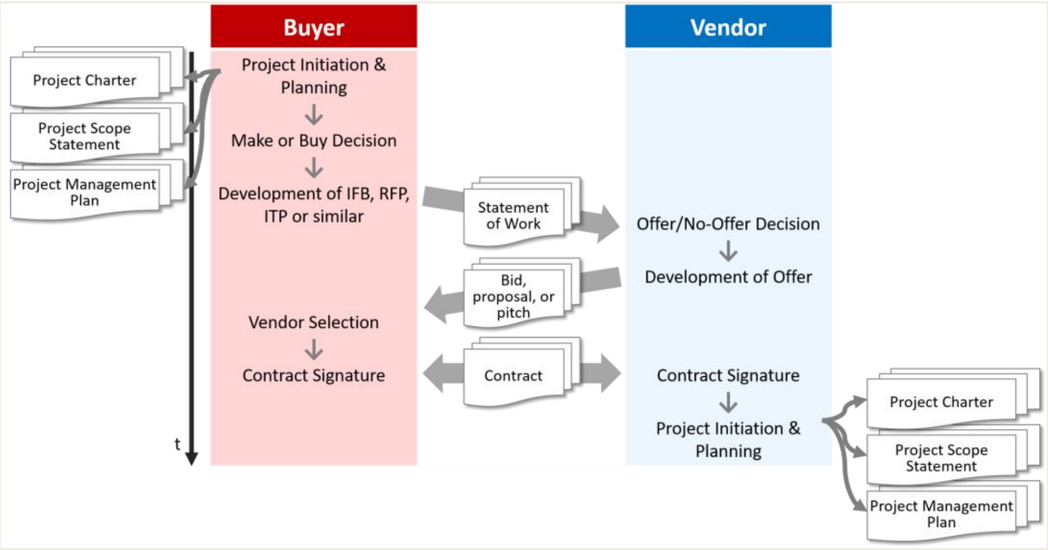

Figure 6: A classical flow of documents in project procurement processes. While statements of work detail important information from a customer perspective and project charters contain only high-level data, a project scope statement can limit flexibility with project scope and prove detrimental for agile teams who need to be innovative.

Fixed Price with Some Agile Elements

If we want to make fixed price models more suitable for agile projects (as opposed to simply using other models described below), we will have to apply some more agile principles in them.

The following factors could be considered in the contract then:

1. **Functionality is not fully defined** at this stage—parties agree they are all aware of this fact

2. There is a **shared understanding of common goals** as defined in the product vision and maybe roadmap—in any case, the parties agree that they will talk to each other continuously, instead of just exchanging documents.

3. **Problem-solving takes place in a cooperative manner** instead of going at a distance to each other, determined mainly by empowering people and trusting them ("Agile Mindset"), not so much by controlling and evaluating. If a specific framework such as Scrum and/or XP are applied, it helps to refer to their official definitions (e.g. the Scrum Guide) in the contract.

4. The **Definition of Done** could go into the contract as a minimal basis of understanding (however not excluding updates of this definition throughout the project (Arbogast, Larman & Vodde, 2012)). Acceptance processes at outlined in a general framework in the contract; acceptance criteria serve as important input for the Definition of Done. Contract parties need to understand the principles behind Definitions of Done and why they are team agreements (Agile Alliance, 2019).

5. **Time-boxing definitions** (iteration—in Scrum: Sprint—durations, overall roadmap, timing and regularity of meetings) help create a common basis for timing concepts applied in the project. For the concept of time-boxing, see (Wikipedia, 2018).

6. **Role clarifications** are important in any contract, and even more so in agile projects. Also: who participates in which meeting? Customer and contractor should both be present in Sprint Planning meetings to avoid surprises that have contractual impact.

7. We use no or only a very **lightweight** project scope statement. The statement of work will be turned into the initial **Product Backlog**.

8. **Estimation methods** can remain classical if that has worked fine in the past. Agile estimation techniques could be used if they haven't or if there are reasons to improve estimation predictability and learning speed when tracking progress (Radigan, 2019). Release Burndown and Budget Burnup charts can help with that as well.

If estimations prove to be false and problems arise, measures have to be taken to react. The worst case is a joint decision for a project cancellation. Otherwise, contractors will have to create reserves and shift project priorities and reserves to accommodate for the one in trouble in order to maintain liquidity.

A good way to prioritize incoming requirements is according to risk associated with them and the value they are expected to bring when fulfilled—as in the following order:

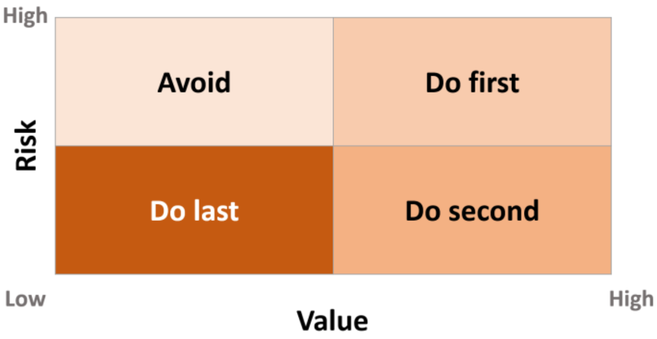

Figure 7: This scheme for prioritizing requirements is recommended in agile customer projects. It is an attempt to make sure work involving high risk is done early, when options are still manifold and expenditures relatively low (Zacarias, 2015).

Agile Fixed Price

While the above examples on how to introduce some elements of agile thinking into classical fixed-price contracts are merely ways of trying to adapt a rather inflexible contract type to projects requiring innovation and exploratory work, there is a model for making this contract types more agile in itself (Opelt et al., 2018). This model of an "Agile Fixed Price" proposes to first describe the contractual project scope based on the product visions and very low-granular features or epics, maybe some user stories as examples, if needed—but never too much specification. Based on this the total effort is estimated using Story Points (Radigan, 2019) and translated into person-days, for example by simulating the first Sprint Planning (Cohn, 2010). This initiates a sort of pre-project phase to learn more about the project and the collaboration, after which the undertaking can still be aborted if things have gone wrong. Risk sharing agreements make sure that risks are fairly distributed among contract parties.

Guaranteed Maximum Price

A special form of fixed-price contract often used in construction can be tried out in an attempt to make the agreement more suitable for agile projects: *Guaranteed maximum price*, "where the contractor is compensated for actual costs incurred plus a fixed fee subject to a ceiling price. The contractor is responsible for cost overruns […]. Savings resulting from cost underruns are returned to the owner" (Wikipedia, 2018). This might lead to contractors offering higher prices that in classical fixed-price situations in order to still have a margin high enough to make the project assignment attractive. To offer this type of agreement will probably only work out if the customer's vendor selection process is requesting this from all bidders. Moreover, especially large customer organizations might not have the procurement processes in place to accept flexible price offers at all (Pieper & Roock, 2017, p. 107).

Guaranteed Minimal Scope

This type of offer requires a prioritization of requirements according to MoScoW criteria (Zacarias, 2015). Contractors guarantee to implement all must-have features at a certain price and assume the risk for cost overruns alone. If there is still money left when all must-haves are implemented, it will be used up for should-have and could-have requirements.

"Money for Nothing, Change for Free"

A contract model for agile projects based on fixed-price models was invented by Scrum co-founder Jeff Sutherland and—apparently inspired by the band Dire Straits—called "Money for Nothing and Your Change for Free" (Sutherland, 2008).

With this approach, the project initiation stage would look like in more predictive projects using classical fixed-price contracts: the customer describes the desired scope and the seller offers a price for its delivery. Starting with development, the customer will turn into the Product Owner and create, manage, and prioritize the Product Backlog. In every Sprint Review meeting where the contractor presents the results of the current iteration, the customer can provide feedback and revise the Product Backlog immediately. For each new requirement in the Product Backlog, another item with comparable effort has to be removed, and implementation of this change will incur no additional cost ("Change for free").

In case the project gets canceled because contract parties feel they have reached the point where new feature development would not create any significant value any more, unspent money from the initially agreed-upon fixed price will be shared among contract parties. Note that this might not work everywhere: larger organizations often do not have processes that allow flexible project financing. The "change for free" idea, however, has proven to make projects more successful because the customer could determine frequently what high-value features look like (Pieper & Roock, 2017, p. 112).

2. Time & Material

Time and materials (aka T&M) is a contract type "in which the employer agrees to pay the contractor based upon the time spent by the contractor's employees and subcontractors employees to perform the work, and for materials used" (Wikipedia, 2018).

In its nature, this contract type harmonizes with agile projects because budget, scope, and time constraints are all flexible. Note that this will however work well if all contract parties are familiar with the agile mindset and have incorporated it enough into their behavior. In other cases, contractors might try to take advantage of the situation by calculating higher daily rates for their experts, or they might fail to participate in process improvements—because after all, if they are more productive, it might simply lead to less money paid to them.

Another problem associated with this contract type could be that courts might interpret it as a contracted labor situation, rather than project collaboration.

Design to Cost, Pay per Productivity

These Time & Material variants can help address problems mentioned above. "Design to Cost" contracts make budgets fixed while keeping the project scope and schedule open. This can work well if the scope can be compared to similar projects, enabling contractors to suggest a price.

Another option is trying to secure a certain productivity on contractor side by tying payments to agreed-upon criteria. At the same time, the contractor provides the benefit of earlier delivery to the customer.

As quality problems will lead to less productivity over time, with such a model, teams have the incentive to keep quality high and technical debt low. Productivity can be measured using Function Points (Wikipedia, 2018). This can be combined with pay-per-Sprint models (see below) as well as risk-sharing and benefit-sharing concepts.

3. Payment per Sprint

Fixed Price per Sprint

A fixed-price-per-Sprint model is like many small fixed-price projects, where each iteration is a mini-project. Every iteration planning meeting leads to a new contract, and every iteration review serves as an acceptance meeting. This leaves overall project scope, time, and budget constraints open but fixates them on iteration level. Over time, experiences and empirical learning processes can lead to more realistic prices and more security for the customer as to what will be delivered in return for the money.

This framework contract model is often used by government offices. In the United States, they have taken the form of so-called Blanket Purchase Agreements (BPA) or Indefinite Delivery / Indefinite Quantity (IDIQ) (FedBid Client Services, 2014). In order to comply with federal vendor selection regulations (a specific challenge for government organizations, as outcomes are in the interest of the public), an option might be to start small pre-projects with several vendors to evaluate their performance (Lehmann-Benz, 2018).

Pay What You Get

A "Pay what you get" contract model is similar to the previous per-Sprint model, however, with a specific provision that the customer only pays what is accepted during iteration review and essentially "buys" the increment developed in the current Sprint. A negative decision on behalf of the customer will lead to a cancelled project, assuming a project phase is reached where new development efforts will not lead to much additional value any more. This model requires contractors to cover costs for every Sprint in advance. An advantage of this approach can be that customers will try to make sure more than in other models that this does not lead to a significant loss for the contractor—especially if they value the collaboration.

4. Benefit-Oriented Models

The idea to base contracts on the benefit or value of work results yielded is different to all other contract approaches discussed above because it turns away from just considering costs. This idea is fairly new and revolutionary, so that legal experts might not be too familiar with it. It might be the key to deal with the extraordinary amount of complexity and uncertainty that we find in so many software projects. Benefits and cost/benefit ratios will have to be measured continuously in order to make such a system work. While the value created should not get out of control, neither should the costs incurred to create that value—there needs to be a good balance.

Benefit-Oriented Award-Fee Contracts

There are contract models that require the customer to pay a low daily rate covering the contractor's costs, but not generating any profit for the supplier. Whenever a previously agreed-upon goal in terms of generated value is reached, an additional award fee is paid to the contractor. In order for this to work effectively, there has to be a transparent, mutual agreement on what is valuable for the customer on the receiving end. Impact Maps can help with finding such an agreement by defining goals according to SMART criteria and how to reach these goals (Adzic, 2012). The incentive for a contractor to create valuable results can be enhanced by adding profit-sharing provisions in the contract.

Pay per Use

In Pay-per-use contracts, payments by the customer are tied to the frequency of usage for certain features. This is the underlying principle of many SaaS offers and can generally harmonize quite well with agile projects (Beck & Andres, 2000).

Conclusion

Projects are unique, temporary endeavors. As such, no contract model will be perfect for all. Instead, project managers and all other people involved in negotiating project contracts on all sides will have to find a model that works in a specific project or situation. This is especially true for agile projects with all the insecurities that they bring.

This article has discussed various possible contract models. Legal experts can help choose one that fits the endeavor at hand and avoid pitfalls that can turn out to be very costly mistakes. However, they have to be familiar with project management in general and agile principles in specific enough to make sure contracts reflect the situation and protect contract parties' interests. The doctrine of Good Faith is especially important in agile projects, where a relationship of trust and respect is even more important than it is generally in project collaborations. "Mission Success First" (Lehmann, 2018) should always be the motto of companies working together. A well-designed contract fit for purpose can be a basis to achieve this mission success.

References

Adzic, G. (2012) *Make a big impact with software products and projects!* [Online]. Available from: https://www.impactmapping.org/ (Accessed: 25 January 2019).

Agile Alliance (2019) *Glossary - Definition of Done* [Online]. Available from: https://www.agilealliance.org/glossary/definition-of-done (Accessed: 24 January 2019).

Arbogast, T., Larman, C. & Vodde, B. (2012) *Agile Contracts Primer* [Online]. Available from: http://www.agilecontracts.org/agile_contracts_primer.pdf (Accessed: 24 January 2019).

Beck, K. & Andres, C. (2000) *Extreme Programming Explained: Embrace Change*, 1st edition, Michigan: Addison-Wesley Professional.

Beck, K. et al. (2001) *Manifesto for Agile Software Development* [Online]. Available from: agilemanifesto.org (Accessed: 23 January 2019).

Cohn, M. (2010) *Suceeding with Agile: Software Development Using Scrum*, Boston: Addison-Wesley.

FedBid Client Services (2014) *What are the BPA/IDIQ competition types?* [Online]. Available from: http://fedbid.custhelp.com/app/answers/detail/a_id/207/related/1/session/L2F2LzEvdGltZS8xNTQ4MjMyNjI3L3NpZC9GSU9KRXg1bw%3D%3D (Accessed: 25 January 2019).

Heusser, M. (2015) *Comparing scaling agile frameworks* [Online]. Available from: https://www.cio.com/article/2974436/agile-development/comparing-scaling-agile-frameworks.html (Accessed: 24 January 2019).

Jansson, K. (2017) *"Lean Thinking" and the 5 Principles of Lean Manufacturing*, 2 May [Online]. Available from: https://blog.kainexus.com/improvement-disciplines/lean/lean-thinking-and-the-5-principles-of-lean-manufacturing (Accessed: 25 January 2019).

Lehmann, O.F. (2017) 'Leading Project Teams Across Corporate Borders', *PM World Journal*, VI (11), November [Online]. Available from: https://pmworldjournal.net/article/leading-project-teams-across-corporate-borders/ (Accessed: 30 September 2018).

Lehmann, O.F. (2018) *Project Business Management*, Boca Raton: CRC Press.

Lehmann, O.F. (2018a) *Project Business Management*, New York City, USA: Taylor & Francis.

Lehmann, O.F. (2018b) 'Projects as Profit Centers - Must We Go Back to Square One Again?', *PM World Journal*, VII (XI), October, p. 14 [Online]. Available from: https://pmworldjournal.net/article/projects-as-profit-centers/ (Accessed: 28 October 2018).

Lehmann-Benz, A. (2018) 'Culture Clashes and Speed of Change in Project Environments', *PM World Journal*, September, p. 8. Available from: https://pmworldjournal.net/wp-content/uploads/2018/09/pmwj74-Sep2018-Lehmann-Benz-Culture-Clashes-and-Speed-of-Change2.pdf (Accessed: 25 January 2019).

Opelt, A., Gloger, B., Pfarl, W. & Mittermayr, R. (2018) *Der agile Festpreis: Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge*, 3rd edition, München: Carl-Hanser-Verlag.

Pieper, F.-U. & Roock, S. (2017) *Agile Verträge ("Agile Contracts")*, Heidelberg: dpunkt.verlag GmbH.

PMI (2006) *Code of Ethics and Professional Conduct* [Online]. Available from: http://www.pmi.org/~/media/PDF/Ethics/ap_pmicodeofethics.ashx (Accessed: 27 August 2014).

PMI (2017) *A Guide to the Project Management Body of Knowledge - PMBOK Guide*, 6th edition, Newtown Square, PA, USA: PMI - The Project Management Institute, Inc.

Radigan, D. (2019) *The secrets behind story points and agile estimation* [Online]. Available from: https://www.atlassian.com/agile/project-management/estimation (Accessed: 24 January 2019).

Schwaber, K. (1995) 'SCRUM Development Process', SCRUM Development Process, Austin, Texas, 23 [Online]. Available from: http://www.jeffsutherland.org/oopsla/schwapub.pdf (Accessed: 22 January 2019).

Schwaber, K. & Sutherland, J. (2017) *Scrum Guide* [Online]. Available from: https://www.scrumguides.org/scrum-guide.html (Accessed: 24 Januar 2019).

Sutherland, J. (2008) *Agile Contracts: Money for Nothing and Your Change for Free*, 25 October [Online]. Available from: https://www.scruminc.com/agile-contracts-money-for-nothing-and/ (Accessed: 25 January 2019).

Wikipedia (2018) *Function Points*, 4 September [Online]. Available from: https://en.wikipedia.org/wiki/Function_point (Accessed: 25 January 2019).

Wikipedia (2018) *Guaranteed maximum price*, 10 December [Online]. Available from: https://en.wikipedia.org/wiki/Guaranteed_maximum_price (Accessed: 25 January 2019).

Wikipedia (2018) *Time and materials*, 30 October [Online]. Available from: https://en.wikipedia.org/wiki/Time_and_materials (Accessed: 25 January 2019).

Wikipedia (2018) *Timeboxing*, 7 November [Online]. Available from: https://en.wikipedia.org/wiki/Timeboxing (Accessed: 4 February 2019).

Wikipedia (2019) *Cynefin framework*, 8 January [Online]. Available from: https://en.wikipedia.org/wiki/Cynefin_framework (Accessed: 25 January 2019).

Wikipedia (2019) *Stakeholder management*, 23 January [Online]. Available from: https://en.wikipedia.org/wiki/Stakeholder_management (Accessed: 24 January 2019).

Zacarias, D. (2015) *20 Product Prioritization Techniques: A Map and Guided Tour*, November [Online]. Available from: https://foldingburritos.com/product-prioritization-techniques/ (Accessed: 25 January 2019).

Image References

Image references: Shutterstock/LightField Studios (title), Shutterstock / Billion Photos (paper stack), own images

About the Author

Antje Lehmann-Benz

Munich, Germany

🇩🇪

Antje Lehmann-Benz, MA, PMP, PMI-ACP, PSM is a project management and agile training professional for Oliver F. Lehmann Project Management Training, working with various training providers. Recent experience includes:

Since 2017: Lecturer at the Technical University Munich, teaching Scrum Fundamentals to PhD candidates in Informatics

2017: Agile training for a US military institution in Germany

2018: Online PMP preparation training sessions for a global telecommunications company

Since 2018: Scrum trainer for a German car manufacturer

Since app. 2009: Project management and Scrum practitioner, consultant in the semiconductor and IoT industries (Atlassian JIRA / Confluence implementations)

She is also active as a volunteer for the Project Management Institute Southern Germany Chapter e.V. (Editor-in-chief, chapter magazine "PMI SG Live"; Director at Large for English Speaking Meetings in Munich). Active in 2016-2017 for PMI International as Subject Matter Expert regarding specific industry experience.

Magister Artium in linguistics, (M.A., LMU Munich), Antje holds the following certifications: PMI-ACP, PMI Agile Certified Practitioner (http://www.pmi.org); PSM, Professional Scrum Master (http://www.scrum.org); PMP Project Management Professional (http://www.pmi.org)

She can be contacted at antje@oliverlehmann.com

To view other works by Antje Lehmann-Benz, visit her author showcase in the PM World Library at https://pmworldlibrary.net/authors/antje-lehmann-benz.