Appendix A
Interaction Room Workshop Agendas

A.1 Interaction Room Workshop Agendas

The agendas suggested below can serve as guidelines for conducting Interaction Room workshops. The specified timeframe should only be interpreted as a rough orientation. Depending on the complexity of the project and which questions have to be resolved most urgently, it is conceivable to only apply certain elements of the methodology, or to spread the workshops over several days or even weeks. For example, it can be helpful to plan dedicated workshop days for the population of individual canvases, conducted at intervals of several days to give the stakeholders time for reflection, more in-depth understanding and preparation between the workshops.

A.1.1 IR:digital Workshop Agenda

Day 1

9:00 a.m.  Introduction of the stakeholders, overview of the IR:digital methodology
9:30 a.m.  Establishing the workshop objective
10:30 a.m. Population of the partner canvas
12:00 p.m. Lunch break
1:00 p.m.  Annotation of the partner canvas, discussion, and establishment of no more than the five most important partners
2:00 p.m.  Technology overview (presentation)
3:00 p.m.  Population of the physical object canvas (focus on identification of the OoIs)
4:30 p.m.  Annotation of the physical object canvas, discussion, and establishment of no more than the ten most important OoIs
5:00 p.m.  Summary of the day, establishing the focal points for day two
5:15 p.m.  Conclusion of the day.
Day 2
9:00 a.m. Population of the physical object canvas (focus on life cycles of the most important OoIs)
11:00 a.m. Technology overview (presentation)
12:00 p.m. Lunch break
1:00 p.m. Population of the touchpoint canvas for the five most important partners (including establishment of the touchpoint lanes)
2:30 p.m. Annotation of the touchpoint canvas, discussion
3:30 p.m. Establishing the top five digitalization proposals
4:00 p.m. Parallel preparation of “press releases” for the top five realization proposals
5:00 p.m. Presentation of the “press releases”
5:30 p.m. Summary of the results, feedback session
6:00 p.m. Conclusion of the day.

A.1.2 IR:scope Workshop Agenda

Day 1
9:00 a.m. Introduction of the stakeholders, overview of the IR:scope methodology
9:30 a.m. Establishing the workshop objective, formulating the “press release”
10:30 a.m. Population of the feature canvas, annotation and prioritization of the features
12:00 p.m. Lunch break
1:00 p.m. Population of the process canvas
3:00 p.m. Coffee break
3:30 p.m. Annotation and discussion of the process canvas
4:45 p.m. Brief summary of the insights, establishing the focal points for day 2
5:00 p.m. Conclusion of the day.

Day 2
9:00 a.m. Recap of key insights from day 1 and plan for day 2
9:15 a.m. Completion of the object canvas
10:30 a.m. Annotation and discussion of the object canvas
12:00 p.m. Lunch break
1:00 p.m. Completion of the integration canvas
2:00 p.m. Annotation and discussion of the integration canvas
3:00 p.m. Coffee break

1Depending on the project requirements, another canvas may also be chosen as the leading canvas.
A.1.3 IR:mobile Workshop Agenda

Day 1
9:00 a.m. Introduction of the stakeholders, overview of the IR:mobile methodology
9:30 a.m. Establishing the workshop objective, formulating the “press release”
10:30 a.m. Formulating, presenting, and weighting the personas
12:00 p.m. Lunch break
1:00 p.m. Population of the portfolio canvas
3:00 p.m. Coffee break
3:30 p.m. Annotation and discussion of the portfolio canvas
4:45 p.m. Brief summary of the insights, establishing the focal points for day 2
5:00 p.m. Conclusion of the day.

Day 2
9:00 a.m. Recap of the key insights from day 1 and the plan for day 2
9:15 a.m. Population of the touchpoint canvas
11:00 a.m. Annotation and discussion of the touchpoint canvas
12:00 p.m. Lunch break
1:00 p.m. Population of the interaction canvas
3:00 p.m. Coffee break
3:30 p.m. Annotation and discussion of the interaction canvas
4:30 p.m. Summary of the insights, establishing the next steps
5:00 p.m. Conclusion of the day.

A.1.4 IR:tech Workshop Agenda

Day 1
9:00 a.m. Introduction of the stakeholders, overview of the IR:tech methodology
9:30 a.m. Establishing the workshop objective
10:30 a.m. Population, annotation, and discussion of the feature canvas
12:00 p.m. Lunch break
1:00 p.m.  Population of the object, \(^2\) integration, and process canvas (current state)
3:00 p.m.  Coffee break
3:30 p.m.  Annotation and discussion of the canvases (current state)
4:45 p.m.  Brief summary of the insights, establishing the focal points for day 2
5:00 p.m.  Conclusion of the day.

**Day 2**

9:00 a.m.  Recap of the key insights from day 1 and the plan for day 2
9:15 a.m.  Population of the object canvas (target state)
10:30 a.m. Population of the integration canvas (target state)
12:00 p.m. Lunch break
1:00 p.m.  Population of the process canvas (target state)
2:30 p.m.  Annotation of the target canvases
3:00 p.m.  Coffee break
3:30 p.m.  Discussion of the canvases, deriving technology implementation potential and hurdles
4:30 p.m.  Summary of the insights, establishing the next steps
5:00 p.m.  Conclusion of the day.

\(^2\)Depending on the project requirements, another canvas may also be chosen as the leading canvas.

---

Appendix A: Interaction Room Workshop Agendas
Appendix B
Interaction Room Annotations

B.1 Interaction Room Annotations

The annotations that were briefly introduced in the chapters on the individual canvases are presented and explained in more detail in the following sections. This list is intended to help the IR coaches choose the most suitable annotations for a specific context. In addition to describing the meanings of the annotations, detailed questions are also listed to help coaches with precisely pinpointing and specifying the annotated issues in their project context.

B.1.1 Value Drivers

Value drivers are indicated by the symbols shown in Fig. B.1.

![Value Driver Symbols](image)

Fig. B.1 Value driver annotations

B.1.1.1 Business Value

From a service provider’s perspective, value creation by a system or process element can express itself in many ways. While financial contributions to the sales objective are the most obvious and easiest to measure (e.g., sales via a shop platform), contributions to other business objectives can be more difficult to comprehend, which means they can be easily lost in prioritization approaches. Contributions to objectives such as customer loyalty, the external image, or quality are difficult to measure may not be directly attributable to concrete features. The business value annotation encourages the explicit exploration of these points.
The following questions can be used to state the annotated business value more precisely:

- What system/process element has a particular influence on the business value?
- What company objectives are positively influenced by the annotated element, which ones negatively? (customer loyalty/sales/competition/market share/external image/marketing/sustainability/company development/costs/quality/productivity/other);
- What is required to achieve a positive influence or prevent a negative influence?

B.1.1.2  User Value
A question of added value also arises from the perspective of the system’s user. While the expectations of the user and service provider may complement each other in some cases (e.g., the user benefits from a certain function, the provider from the usage fee), they may also contradict each other in other cases (if the user expects a function that is unattractive for the provider, e.g., the link to a comparison portal). To uncover such areas of conflict, the perception of the user is highlighted by the distinct user value annotation.

Since value dimensions as clear and generally applicable as the various company objectives used for the business value annotation typically cannot be defined for users, the Kano (1984) classification is used here. The intensity of value creation is defined according to whether the annotated system or process element is a must-be, one-dimensional, or attractive quality, i.e., a characteristic that is expected as a matter of course, one that is perceived to improve performance, or one that positively surprises the user and may enhance acceptance of the system.

To state this assessment more precisely, the following questions should be answered for each user value annotation:

- What system/process element has a special influence on user value?
- What are the expectations for this element? (basic/performance/enthusiasm characteristic);
- What is required to achieve a positive influence or prevent a negative influence?

B.1.1.3  Innovation
The innovation annotation identifies process sections, system elements, features, or general ideas that are especially innovative, for instance novel from a business or technical perspective. It therefore constitutes an interface between value and effort drivers:

Innovation implies a special business value on the one hand—if there was no prospect of this, one would hardly be inclined to expend the effort and accept the risk of the innovation (see below). The innovation annotation usually also identifies a user value, often even in the sense of an enthusiasm characteristic. The feature offers a novel function or realizes a known function in a new way that positively surprises the user, thereby differentiating itself from the competition and improving user acceptance.
But on the other hand, the innovation annotation also implies effort: Implementing an innovative solution is usually more resource-consuming than using established solution templates, since new approaches to solutions first have to be developed and evaluated. Risks are also inherent in every innovation—regarding the feasibility of the technical implementation, the time and budget required for implementation, and the ultimate acceptance by the user.

This combination of characteristics makes the innovation annotation an important anchor point for discussions. It requires especially diligent estimates of effort and prioritization as well as particularly competent development and quality assurance.

For a more precise definition, the following questions should be answered for every innovation annotation:

- What system/process element constitutes a special innovation?
- What is the innovation in this element?
- Is it a technology or business innovation? Is it disruptive in nature?

### B.1.2 Effort Drivers

Effort drivers are indicated by the symbols shown in Fig. B.2.

![Effort Driver Symbols](image.png)

- High use
- Time constraint
- Accuracy
- Reliability
- Security
- Usability
- Attractiveness
- Flexibility
- Mobility
- Automation
- Manual task
- Policy constraint
- Complexity
- Invariability
- Deprecation
- Need for improvement
- External resource

**Fig. B.2** Effort driver annotations
B.1.2.1 High Use

Parts of a system may be permanently or temporarily subject to high use—perhaps due to a high number of users, voluminous batch processing, spikes due to deadlines or unforeseen events. Architectural and infrastructure precautions have to be taken in order to be prepared for such cases and ensure reliable functioning of the system even under heavy load. For example, usage peaks require a flexible infrastructure that can adapt to different load levels, rather than permanently providing resources for a peak load that is only required rarely. Providing such a scalable architecture is not only a question of hardware, but can have far-reaching consequences, for example, on the conceptual design of replication mechanisms, ensuring data and code portability. The conceptual design of this infrastructure, its implementation and testing requires significant effort which can be identified by the high use annotation.

The type of requirement—and therefore also the subsequent solution—is essentially based on how the load is distributed over time and what system components it is concentrated on. Therefore, the stakeholders are asked to answer the following questions in the annotation discussion:

- What system/process component is expected to be subject to high use?
- What is the type of load? When, how often, and for how long is it expected?
- What would the effects of overloading be?

B.1.2.2 Time Constraint

Time constraints in information systems typically apply to prescribed processing or response times (e.g., the maximum time required to make a decision, or the deadline for filing an application). The time-constraint annotation can therefore indicate that developers have to meet real-time requirements or that certain business deadlines have to be observed by the system.

In view of later finding a solution, one has to differentiate whether the time limit is expressed as a *deadline* (point in time) or a *window*. Both are recorded as subtypes of the annotation. The following questions also have to be answered:

- What is the required time frame?
- What is supposed to happen within the time frame?
- What would the effects of exceeding the time frame be?

B.1.2.3 Accuracy

Accuracy appears a trivial requirement at first glance, which should be met by all components of a system. But although correctness in the implementation is an obvious goal, especially high requirements for the precision, timeliness, or consistency of the processed data apply to certain components. For example, components that calculate interest or life insurance premiums decades in advance have to round monetary amounts according to precise specifications, and different implementations (such as a premium calculator in the insurance’s back-end system and one in the mobile app at the point-of-sale) have to deliver exactly the same results. High requirements for the timeliness of data also apply, e.g., to the processing of securities prices. In designing
user interfaces in particular, there is often a need to ensure the correctness of data input through suitable validations as well. These requirements (and the risks associated with the failure to meet them) are indicated by the accuracy annotation which is differentiated into the subtypes precision, timeliness, and consistency.

The following questions should also be answered when discussing the annotation:

- What processes/data are supposed to be as precise/timely/consistent as possible?
- What degree of precision/timeliness/consistency is required?
- What is the expected benefit of precision/timeliness/consistency?

**B.1.2.4 Reliability**

Reliability also sounds like a basic requirement that should be satisfied by any software system. In fact, however, the same reliability standards do not usually apply to all parts of a system: While it is always annoying when an expected system function is not available, the actual consequences can be more or less dramatic. In the most harmless case, there is merely a delay in executing a function. In more critical cases, data is lost. How critical such a loss of time or even data is depends on its severity and the application domain: Delays or the loss of messages can be tolerated in a chat application, but may cause damages in the millions in a financial application. This annotation is therefore intended to identify system elements where reliability is especially critical to support the business domain.

The following questions should also be answered when discussing the annotation:

- What processes/data need to be as reliable as possible?
- What degree of availability is required?
- What benefits are expected from the high reliability?

**B.1.2.5 Security**

The nature of security requirements can vary widely. For example, they include the digital signing of datasets in order to guarantee authenticity and non-repudiation, the anonymization of datasets prior to evaluation or ensuring the confidentiality of certain datasets. Implementing these requirements often affects both business processes and data structures. The security annotation symbolizes process and system components where specific security precautions have to be taken that go beyond the company’s normal security standards. They are differentiated into the subcategories protection against unauthorized access and protection against data loss.

The following questions also have to be answered for a more precise definition:

- What is to be protected against unauthorized access/loss?
- What is the type of threat? How strong is the protection supposed to be?
- What would the effects of a lack of protection be?
B.1.2.6 Usability
Clearly, any software system with a user interface should meet fundamental usability requirements. It should be as intuitive as possible to understand, easy to use, suitable for the tasks of the user and so on. Yet there are often process steps or components that pose special usability challenges for business or technical reasons—perhaps because especially complex material has to be displayed (visualization subtype) or because special operating steps or user interface elements are required (interaction subtype), for example, gesture control on a touch screen. The challenges that require special attention in the interaction design (but possibly also in the process design) are identified by the usability annotation.

To define the usability requirement more precisely, the following questions have to be answered:

- What is supposed to be as understandable/easy to use as possible?
- What makes usability a special challenge in this element?
- What benefits are expected from usability in this element?

B.1.2.7 Attractiveness
The desire to make certain system or process elements as attractive as possible is related to the usability requirement at first glance. However, the attractiveness annotation can be used to identify a requirement that goes beyond an appealing user interface design. In some cases, a special incentive should be created to execute certain activities—whether this is through technical or business means. Examples are bonus systems or gamification techniques that reward certain activities beyond the fundamentally positive user experience.

The definition of attractiveness can be stated in more concrete terms by answering the following questions:

- What is supposed to be especially attractive?
- How is this incentive supposed to be created?
- What benefits are expected from the incentive?

B.1.2.8 Flexibility
For some process or system elements, it is already known at the time of conception that a single implementation cannot cover all requirements of the users, usage contexts, and basic conditions. An adequate measure of flexibility has to be planned in these cases, which can be achieved in various ways:

Minor adaptations of the system functionality to user-specific requirements or changed basic conditions can be realized through a suitable configuration of the system at runtime without requiring additional development effort (configurability subtype).

In other systems, certain (mainly technical) basic conditions may require realizing more than one version of a system from the outset to cover different applications. The versions are mutually independent instances of the system but
typically developed in close dependency on each other—for example, the realization of an Android and an iOS version of a mobile app (*variability* subtype).

Finally, it may already be known in the design phase of a system that the current implementation will have to be adapted or replaced in the foreseeable future (e.g., when changes in legal regulations are pending). The future change should be taken into account in the system from the outset in this situation—both in regard to the architecture and in allocating effort to future obsolete functions (*design for change* subtype).

The following specifying questions serve to improve understanding of the background that makes flexibility necessary:

- What system/process element needs to be designed for flexibility?
- What configurations/variants/evolution paths are required?
- What would the effects of inflexibility be at this point?

**B.1.2.9 Mobility**

Accessing an information system or executing a business process using a mobile device such as a smart phone poses a number of challenges for the mobile components, which are not as prominent in classic information systems. In addition to the diversity of platforms already discussed under the flexibility annotation, the primary mobile effort drivers are mainly the unreliability of the network connection and the inclusion of location data (including fallback mechanisms if these are not available).

The mobility annotation is therefore stated more precisely by the following subtypes: *mobile availability* (to identify functions or process segments where mobile availability is desired), *off-line availability* (to identify functions that are supposed to work even if a network connection is lacking) and *location dependence* (to identify functions that depend on location information).

The mobility requirements are defined in more concrete terms with the following specifying questions:

- What system/process element is to be available in mobile use, off-line, or location-dependently?
- Under what circumstances can functionality restrictions be expected?
- What are the expected benefits of mobility/off-line availability/location dependence?

**B.1.2.10 Automation**

The objective of developing information systems is often the automation of process segments that were previously performed (semi-)manually. Such an automation project brings up a number of business and technical questions. Automation is usually no trivial mapping of activities to a technical solution, but requires the adaptation of processes and data, the definition of interfaces and—often the most difficult—establishing the extent of automation: To what degree will the process be...
automated? What special cases or errors are to be handled automatically, which ones are supposed to prompt for human intervention? What interfaces are used for such intervention? How can the process be simplified to avoid special cases as far as possible? Clarifying these questions and the resulting implementation requires significant effort and upheaval, which is highlighted by the automation annotation in the diagram.

For motivation and to state these requirements more precisely, the following questions should be discussed:

- What is to be automated?
- What degree of automation is desired?
- What does the user expect from the automation?

**B.1.2.11 Manual task**

The counterpart to the automation annotation is the manual task annotation. It indicates that a certain process segment will continue to be performed manually since it is not suitable for automated processing. This may be because specific expert knowledge is required, because it is based on an expert assessment that cannot be implemented algorithmically, or because human processing is preferable to automation for social reasons. Even a manually executed process step is a challenge for the realization of the information system, since one has to consider how manually and automatically processed data and process steps will be linked.

To define the requirements for manual processing steps more precisely, the following questions have to be asked:

- What requires human action/decision-making capability?
- How can the interface between the human and the system support/integrate manual tasks as well as possible?
- What effects would errors in manual processing have?
- Why is manual processing preferable to automation of this task?

**B.1.2.12 Policy Constraint**

Both the development and the operation of software systems take place in a project context that usually defines a number of different constraints. In general, most of these constraints are not formulated as dedicated requirements. This may be because they are assumed to be known to all stakeholders, or conversely because they are not known to any of the stakeholders, or they may be formulated somewhere but not be actually practiced or enforced.

The approach of not formulating all constraints as explicit requirements is initially due to a certain pragmatism. Since it is obvious that a patient file management system for a health insurance company has to comply with the applicable legal regulations, one is not going to include the entire text of the law in the requirements documentation. Yet it is important for the project stakeholders to be aware at what points in a process or system special attention must be paid to
certain constraints—especially those that express more of a quality than a functional requirement.

Beyond that, projects are often subject to a number of constraints that cannot be formulated as product requirements, but rather define specifications for the system’s design process—these may be technology decisions specified by existing system landscapes or quality assurance measures that apply throughout the organization.

The policy constraint annotation serves to highlight system or process elements where taking such constraints into account for the conceptual design or operation is especially critical or resource-consuming. According to the preceding discussion, one can differentiate legal, technical, and organizational constraints.

The following questions serve to state the constraint and its implications more precisely:

- What system/process element is subject to a constraint?
- What constraint has to be observed?
- Can this become a show stopper?

Since respecting any existing constraints is typically unavoidable, the question examining benefits is replaced by a risk-focused question for this annotation: A constraint may impose requirements that cannot be adequately implemented within the scope of the project and therefore endanger the success of the project. The question of the show-stopping character of a constraint helps identify such fundamental risks early on, which provides an opportunity to manage them.

B.1.2.13 Complexity

Some system or process segments that make a comparatively simple impression in the model can actually harbor a high degree of complexity which is not apparent to all stakeholders at first glance. This may be business complexity such as extensive calculation or decision rules, but also technical complexity such as major conversion or integration challenges. Such complexity may only be foreseen by a few domain experts or experienced developers, while stakeholders who are less familiar with the business or technology specifics typically underestimate the inherent effort and risks. The complexity annotation can be used to let all stakeholders know that the implementation of a certain system or process element requires expert knowledge and possibly extensive research or prototyping effort.

In stating the complexity annotation more precisely, one differentiates between the business and technical complexity subtypes since the solution typically requires the involvement of different stakeholder groups. The following specifying questions should be posed as well:

- What system/process element is especially complex?
- What does the complexity consist of?
- Can this become a show stopper?
B.1.2.14 Invariability
In extensive, organic system landscapes, there are typically numerous dependencies between components in different life cycle stages and with different development histories, technologies, and interfaces. While some legacy systems can be maintained and further developed, others may be facing imminent replacement.

A development stop may be imposed on especially critical legacy systems. Even though they continue to be used in productive operations, changes to the system are no longer permitted. This decision may, for example, be made because the system continues to work reliably, but certain decades-old knowledge about implementation specifics has eroded over time so that the effort and risk of re-engineering and adaptation of the code (which may, e.g., be written in COBOL) is considered unreasonably high. The invariability annotation can be used to notify all stakeholders that an existing system cannot be adapted to new conditions (e.g., in the course of process changes or the integration of additional components), but has to be encapsulated with suitable adapters.

To define the background and effects of the invariability decision more precisely, the following questions should be answered by stakeholders when this annotation is used:

- What system/process element is supposed to remain unchanged?
- How is this element supposed to be integrated into the changed system/process landscape?
- Why and on whose initiative is this element supposed to remain unchanged?

B.1.2.15 Deprecation
While some legacy components have to be considered invariable, other components may be easier to replace. Components slated for replacement or elimination can be identified with the deprecation annotation in the models. It indicates that the respective component will no longer be available in the future, so that the system under development cannot rely on it. Procedural or technical alternatives have to be developed for the deprecated component instead.

To state the background and effects of a component’s deprecation more precisely, the following questions have to be answered:

- What system/process component is deprecated?
- How will its tasks be implemented in the future (if at all)?
- Why and on whose initiative is the component designated as deprecated?

B.1.2.16 Need for Improvement
In the course of the maintenance, adaptation or new development of system components, changing existing functions or processes is sometimes expedient in order to optimize them or adapt them to new conditions. This may involve adaptation, expansion, or simplification of a business or technical nature. In all of these cases, the “need for improvement” annotation indicates that work is required
on a certain component or process step, thereby giving the stakeholders an overview in especially complex system landscapes of where “construction sites” are located and what areas remain stable.

The background of the planned improvement can be captured more precisely by answering the following questions:

- What system/process element is supposed to be adapted/expanded/simplified?
- What change is planned?
- Why and on whose initiative is this change being made?

**B.1.2.17 External Resource**

Interfaces to external resources often constitute effort drivers for two reasons: When the system being developed provides services for external components, the interface has to be designed with special care to optimize it for the current purpose but also be prepared for future expansions. When an existing interface is being expanded, one has to ensure that components already using this interface are not affected by the adaptations.

If the system being developed depends on external components, one should consider what happens if these components are temporarily or permanently unavailable, if their interfaces are altered or the technical or business conditions for their use change (e.g., regarding terms and conditions of use, prices etc.). In addition to the need to clarify these questions, the external resource annotation indicates that a fundamental make or buy decision regarding the externally linked functionality may have to be made.

External resources also prompt the question of the possibilities for influence on the resource provider. This is an important aspect since possible integration problems are much easier to solve through collaboration with the provider instead of assuming that the external interface cannot be changed.

- To what external resource does an interface exist?
- What information is exchanged with the external resource?
- What is the benefit/purpose of the connection for us and for the resource provider?
- What are the possibilities for influence on the resource provider?

**B.1.3 Uncertainty**

Uncertainty is indicated by the symbol shown in Fig. B.3.

---

*Fig. B.3 Uncertainty annotation*
The annotations introduced in the previous sections serve to highlight challenges of which at least some team members are already aware. However, it is just as important to be aware of points where there is still uncertainty in the team. These may be business aspects that are not fully understood yet, or open questions about the technical implementation. Such uncertainty is normal in any team, especially in early project phases. In classic system models, however, there is no possibility of expressing it. Quite to the contrary, the formality and precision of the modeling language suggest certainty about the modeled aspects which may not actually exist in the team. Once circumstances are modeled, they may no longer be sufficiently questioned even when they merely constitute initial ideas born out of uncertainty.

The uncertainty annotation addresses this problem by allowing all team members to clearly define the points of a system or process design where they still see a need for clarification. The respective uncertainties usually cannot be clarified immediately in the Interaction Room. Rather, they serve as an indication of where more extensive research is required or where hidden effort and risks may lurk.

Unlike the specifying questions for the preceding annotations, the focus with uncertainty is on strategies to record and clarify the points in question:

- What is the topic of uncertainty?
- What has to be done to eliminate the uncertainty?

### B.4 Documenting Annotations

In practice, pads with self-adhesive annotation symbols have proven useful for conducting the annotation rounds in practice. The annotation characteristics (what does the annotation mean, where is it localized, why is it important) can be documented with forms like the one shown in Fig. B.4.3.

For each annotation that is affixed to an element on a canvas, the IR domain coach records associated background information in one of these forms:

- In the “annotation ID” field in the top right corner, the annotation is numbered. This number is also noted on the annotation symbol affixed to the canvas.
- Next to the “stakeholders” heading, the name(s) of the stakeholder(s) who proposed this annotation are recorded in the “pro” field. If there are stakeholders opposing the annotation, their names are noted as well in the “con” field. This information is helpful for returning to these stakeholders later in the project for more information on dealing with the annotation.
- Under the “value drivers,” “effort drivers,” or “uncertainty driver” heading, the IR domain coach marks the type of annotation that is being documented. For most annotations, a subtype that provides more information on the nature of the challenge can be marked. (The preceding sections provide more information on categorizing these subtypes.)

---

3Annotation stickers and documentation forms are available at www.interaction-room.de.
In case we are dealing with a business value annotation, the IR domain coach can indicate whether the aspect that the stakeholder wanted to highlight has a positive (+) or negative (–) influence on a number of business goals.

For each annotations, the three boxes in the lower half of the form should be filled with notes regarding:

- **Potential benefit/risk:**
- **Frequency:**
- **Difficulty:**

---

**Fig. B.4** Template for the documentation of annotation characteristics

- In case we are dealing with a business value annotation, the IR domain coach can indicate whether the aspect that the stakeholder wanted to highlight has a positive (+) or negative (–) influence on a number of business goals.
- For each annotations, the three boxes in the lower half of the form should be filled with notes regarding
– in the “element” box, the precise model element (i.e., system component, process step or similar) that the annotation is referring to, e.g., “submission of previous period’s transaction data to regulatory authority”;
– in the “requirement/challenge” box, the precise challenge or requirement that the annotation conveys, e.g., “data must be submitted by 20th of the month, or the preceding weekday if the 20th falls on a weekend or holiday”;
– in the “benefit/risk” box, the positive or negative consequences of heeding or ignoring the annotation (whatever the dominant aspect is), e.g., “late filing of data will lead to significant fines.”

The IR domain coach should obtain this information from the stakeholders as they are discussing the annotation.

- In the three groups of fields on the bottom of the form, the IR domain coach should record
  - the team’s impression of the magnitude of the benefit or risk associated with this annotation,
  - the prevalence of the annotated challenge (e.g., its frequency of occurrence in a process—this might be every time the process is executed, or only in rare instances), and
  - the expected difficulty of addressing the annotated challenge.

Since these impressions can only be “gut feelings,” they are just indicated in the three qualitative categories “small,” “medium,” and “large” rather than attempting to quantify them. If there is significant disagreement over the qualification of some criteria, the IR domain coach can choose to mark all options that different stakeholder groups are arguing for (e.g., “S” and “L”), thereby indicating that the annotation is subject to particular contention and needs to be analyzed in more detail.

Together, these three criteria can be valuable indicators of an annotation’s impact and the priority with which it should be addressed, and thus help in project planning.

The documentations of all annotations should be made available to the team together with photographic records of the annotated canvas. This will help the stakeholders later in the project to refer back to the knowledge recorded with the annotations and consider it in their search for appropriate solutions.

References

Appendix C
adVANTAGE Contract Template

C.1 adVANTAGE Contract Template

This chapter presents a contract template for projects conducted according to the adVANTAGE model. We point out that the contract model described in detail in Chaps. 14 and 15 is a template that can be used as a foundation for concrete projects but that considerable adaptations may also make sense, depending on the project requirements or partner constellation. Therefore, the following contract template should not only be reviewed and negotiated by a legal practitioner, but also by the persons in charge of commercial and technical matters for the client and contractor. The contract was developed for the German legal system, which naturally means it needs to be adapted if it is to be applied in a different legal system.

Section 1: Object of the Contract

1. The object of this contract is the development of custom software and the granting of usage rights to this software by the contractor.
2. The individual software development steps—from determining the requirements to specification, design, and implementation to delivery to the client—are being performed in an agile development process.

Section 2: Agile Process Model

The parties have agreed on the application of an agile, iterative project model for software development. The adVANTAGE model applied for the performance of this contract is based on the following principles:

1. Proceeding in sprints
   The design and implementation of the software takes place in several cycles called sprints. Specific requirements jointly defined by the parties are implemented in a
sprint. The requirements are derived from features and refined in the conceptual design. Working software is delivered at the end of each sprint.

2. Service descriptions
Before design and implementation commences in the project, the parties roughly establish the features desired by the client. This description must be established in sufficient detail so the effort needed for the development of the features can be estimated. A detailed description and specification of the requirements to be developed in a sprint is prepared at the beginning of each sprint.

3. Prioritizing the functions for the sprint
The features to be developed in the respective sprint are prioritized by the client before the start of the sprint. This means the client decides which requirements corresponding to the features will be designed and implemented in which sprint. The client can change the prioritization of the features again before each sprint and therefore define the features to be developed in the next sprint after the end of a sprint.

If more features are added in the course of the project, the total effort budget is increased accordingly. The parties jointly establish whether these features are implemented in an additional sprint (the team size established in advance remains unchanged). Replacing features with others of the same or lower value is possible after each sprint without affecting the budget. The principles for the addition of new features apply when features are replaced with others of higher value.

4. Flexibility
After the end of a sprint, the client can terminate the project, define new features, or decide whether specific features will be developed or not. The client therefore has flexibility in responding to new insights and possibly changed requirements in the course of the project.

5. Duties to cooperate
Even more so than other project models, the adVANTAGE project model requires the active participation of the client. The approach and influence the client can exert on design and implementation in the course of the project require a high level of client availability and participation.

6. Settlement after every sprint
Settlement of the services provided takes place after every sprint. The compensation is made up of a base rate for analysis and project management, and effort-based compensation for feature development.

Section 3: High-Level Specification at the Start of the Project

1. At the start of the project, the parties jointly prepare a high-level specification for the software being developed. The high-level specification defines the purpose, field of application, function, future users, and similar parameters of the software being developed in general terms.
2. The business and (if applicable) also the technical requirements of the client are roughly described in the high-level specification in the form of features. A feature encompasses one or more functions of the software and can in principle be used on its own or together with one or more other features.

3. Features have to be described in sufficient detail so the contractor can estimate the design and implementation effort. Whether the initial high-level specification already meets these requirements or a more detailed specification is required has to be decided by the contractor for each feature. The contractor shall notify the client what information is required for the purpose of estimating. Should establishing a sufficient level of detail not be possible, the parties shall establish the compensation model for the respective feature by mutual agreement.

4. The contractor shall provide the client with an estimate of the design and implementation effort for each feature based on the high-level specification. This estimate shall be in person-days.

5. The contractor shall coordinate the high-level specification and estimated effort with the client. The high-level specification and estimated effort are part of this contract and form the basis for subsequent contractual performance. The high-level specification is included with this contract as Attachment 1, the estimated effort as Attachment 2.

6. If the high-level specification and/or estimated effort has already been prepared, coordinated, and approved by the parties before this contract is concluded (e.g., in the course of the quotation process, in the context of a workshop conducted in advance of the project or as the result of a proof of concept), the provisions of the two preceding subsections nevertheless apply correspondingly. The high-level specification and corresponding estimated effort agreed in this way, that is to say the documents containing them, are included with this contract as Attachment 1 and 2, respectively.

Section 4: Sprints: Prioritization, Target Budgets, and Detailed Specification

1. The features described in the high-level specification are prioritized by the client following the approval of the high-level specification, estimated effort, and base rate. In coordination with the contractor, the client decides which features are most important to the client and which ones are assigned a lower priority. While the client is largely free to assign the priorities, possible business and technical dependencies between different features must be taken into account. The contractor supports the client in establishing a reasonable prioritization. The result of this prioritization is recorded by the parties in a list. This list is included with the contract as Attachment 3 and is therefore part of the contract.

2. The features (currently) having the highest priority are developed in the course of each sprint.
3. After this (initial) establishment, each sprint is assigned a fixed duration as well as the budget derived from the base rate and estimated effort for developing the features of the respective sprint.

4. At the start of each sprint, the features assigned to the sprint are established in concrete terms and detail in a detailed specification. For the second and all subsequent sprints this is done by the product owner during the currently ongoing sprint as far as this is possible.

5. The contractor designs and implements the requirements derived from the features in the detailed specification during the respective sprint and tests whether the software meets the requirements. Then, the contractor provides the software to the client for testing. The client tests the software and confirms acceptance. Acceptance is also deemed to be declared if the client puts the software into operation without reservations.

6. At the end of the project, after the final sprint is concluded and following the handover to the client, the client is given the opportunity to perform a final review and final comprehensive test. Once this is concluded, the client declares

- that the project is complete,
- that the software conforms to the contract, or
- if deviations are noted, the desired rectification of defects.

7. The duration of the respective sprint and therefore the time budget for delivery for testing are fixed (known as time boxing). This means an individual sprint is not extended under any circumstances. Instead features for which requirements were not implemented in the software are shifted to another sprint, usually the following one (referred to as “carryover” in the following sections).

Section 5: Subcontractors

The contractor is authorized to employ one or more subcontractors in the performance of this contract.

Section 6: Principles of Cooperation, Project Organization, and Escalation

1. The success of the project depends on close cooperation between the parties and a constructive communication culture. Therefore, the parties within the scope of the applicable legal regulations declare their unrestricted readiness for mutual consideration, comprehensive information, precautionary warning of risks, joint and constructive resolution of differences in opinion and protection against disruptive third-party influences. This does not make the parties affiliated under company law.

2. The responsible product owner on the contractor side is named in Attachment 5 to this contract. The contractor may change the product owner and will inform the client accordingly if this is the case.
3. The responsible project manager on the client side and its deputy are also named in Attachment 5 to this contract. Both the project manager and deputy must have the required business and technical expertise for the tasks to be performed by them in the course of project implementation.

4. The contractor shall perform the design and implementation work pursuant to the contract at its own premises. However, performance may also be on site with the client or at the location of the hardware to prepare for delivery for the purpose of review and testing.

5. The project language is German.

6. Differences in opinion and disputes that arise or develop in the course of the project and that may endanger the successful realization of the project are initially discussed and clarified at the level of the contractor’s project owner and the client’s project manager. The resulting arrangements and agreements are jointly recorded immediately and exchanged in text form as a minimum. If timely resolution cannot be achieved at this level, the matter is immediately escalated to the steering committee for the project. The members of the steering committee are listed in Attachment 5 to this contract and are asked to resolve difficulties that arise promptly and in good faith.

Section 7: Duties to Cooperate

1. Active participation of the client is of particular importance under the chosen project model. The client therefore not only considers this a project of the contractor, but also its own project and is aware that adequate own resources of the client have to be scheduled on an ongoing basis for the successful realization of the project. The parties agree that the duties of the client to cooperate are actual obligations to perform.

2. The client obligates itself to promptly and at all times provide the business and technical information, deliver the documentation, and perform the acts required for the realization of the project, especially for the preparation of the detailed specifications for the individual sprints, for development and programming of the features and for testing.

3. The project manager and deputy are always available to the contractor for all questions related to the realization of the project. They can be reached by the contractor at any time by e-mail and telephone on working days between 8:00 a.m. and 8:00 p.m. and are authorized—to the extent obligated—to make all required decisions for the realization of the project.

   The client is authorized to replace the project manager or deputy with another person having equal qualifications and availability. However, the contractor must be notified of such measures in advance. Replacing the project manager and deputy at the same time is excluded.

4. In view of the fixed schedule for each sprint (time boxing), the parties agree to respond to inquiries and requests of the contractor within no more than 24
hours and to provide the respective information, take the required actions or make decisions. In cases where neither the client’s project manager nor deputy meets this obligation in a timely manner, or if neither the project manager nor deputy is deployed at the respective time contrary to the obligations, the contractor shall make all decisions and take actions that are due itself subject to the principle of good faith. Decisions made in this way are binding for the parties.

5. In addition, the client’s project manager or deputy is available to the contractor’s staff responsible for the project, in particular for preparation of the detailed specification for the next respective sprint, and shall develop this detailed specification jointly with the contractor and approve it following coordination.

6. Furthermore, the client shall prepare and deliver to the contractor all data required for development, programming, and conducting functional verifications and tests of the software. This applies in particular to test datasets and test content to verify the functions of the respective software. The client shall also generate corresponding test cases in coordination with the contractor.

7. By request of the contractor, the client shall also enable the contractor to test the implemented software in a production equivalent test environment of the client prior to delivery for testing. The client shall provide corresponding access for this purpose.

8. The client shall promptly report failures, disruptions, and impairments in the operation of the software, including an error description, whether before or after testing.

9. Failed, late, defective, or incomplete performance of the duties to cooperate shall be borne by the client. The contractor shall notify the client in case of default on duties to cooperate contrary to duty, if applicable with a grace period to make up or repeat performance by the client. Due to the strict schedule for the sprints, verbal notification is sufficient.

If the client fails to meet the respective duty to cooperate within the grace period, the contractor is authorized to provide the service based on the information already available or to proceed according to subsection 7.4, sentences two and three of this contract. If neither one nor the other is possible, the contractor can stop the corresponding work until the duty to cooperate is met. Damages incurred by the contractor due to failure to meet the contractual duties to cooperate, in particular by keeping resources available (especially in the form of wait times by employees scheduled for the project), are billed to the client at the regular daily rates pursuant to Attachment 7 to this contract. In this context, the parties agree that the contractor based on the chosen project model is not able to otherwise deploy the employees assigned to the project in the course of an ongoing sprint.

Features that could not be completed in the respective sprint due to the failed, late, defective, or incomplete performance of duties to cooperate are transferred to the following sprint. The effort-based budget for the features is also transferred on a pro-rata basis, but without impairing the rights of the contractor pursuant to subsection 7.9, second paragraph, sentences three and four of this contract. The corresponding delays in the project are fully borne by the client.
10. The client is solely responsible for the adequate, if applicable continuous backup of its data according to the importance of the respective data. In particular, the client in this context has to ensure that all possible affected data is backed up again on an external system or data carrier prior to all previously announced work of the contractor performed on the systems of the client as intended.

Section 8: Usage Rights

1. Custom development

(a) The client acquires all exclusive usage rights to the software being developed as well as the documentation that is prepared, in particular the rights to duplication, dissemination, making available to the public and editing including the unrestricted exploitation of editing using all and even unknown exploitation methods. The contractor shall provide the client with the source code on one or more conventional data carriers.

(b) The client can transfer these usage rights to third parties in whole or in part, and/or grant simple usage rights to them to third parties, without additional consent of the contractor.

2. Third-party software

The type, contents, and scope of usage rights granted to the client by the provider of third-party software are determined by the provisions agreed between the provider and the client.

Section 9: Compensation and Payment Terms

1. Compensation for performance and its settlement is based on the following principles according to the chosen project model:

(a) A total effort budget for the project is established before the start of the project. It is based among other things on the number and duration of the sprints, size of the project team and estimated effort for the realization of the features.

(b) Before the start of each sprint, the amount of compensation is established together with the joint estimate of effort in reference to the specific feature.

(c) Settlement takes place after the conclusion of each sprint.

(d) Compensation for the services provided in a sprint always consists of:

(i) The fixed base rate, which covers all services of project management (product owner), the scrum master, developing and preparing the detailed specification, the development and integration tests performed by the contractor and preparation of the release, and the warranty and
(ii) Effort-based compensation for the conceptual design and development work in the sprint, which was previously estimated by the contractor and approved by the client.

The base rate and approved estimated effort constitute the sprint budget.

Definition of roles:

The *product owner* is the business and organizational contact person for the client. S/he then provides the team with the requirements to be realized in the form of features and is available for business questions.

The *scrum master* ensures the optimization of the process, transparency, and improving the productivity of the team.

A *team member* develops the requirements of the corresponding features and actively implements value-oriented solutions in the course of the sprints. This results in usable software.

(e) In principle, compensation is paid for all design and implementation effort expended during the sprint for software that has been tested or made available for testing.

This includes effort that goes beyond the estimated effort and therefore exceeds the target budget, and/or for possible error corrections and comparable activities. However, the parties agree on three different daily rates based on this background, namely:

(i) A regular daily rate for effort of the product owner expended within the target budget,

(ii) A regular daily rate for effort of the Team Members expended within the target budget,

and

(iii) A lower daily rate for all effort that goes beyond the estimated effort

[OPTIONAL: and for all design and implementation effort expended in the course of error correction and similar measures during an iteration]. The reference value for determining deviations is the respective feature.

All effort within the approved estimated effort is deemed to be within the target budget. Features completed within the target budget are settled at the regular daily rate according to the effort actually expended.

Effort that goes beyond the jointly estimated effort is settled according to the effort actually expended. The contractor informs the client if the target budget for a feature is exceeded.

This differentiation between the regular and lower daily rate also applies when a feature is not pursued further in the project (e.g., if it was not completed at the end of a sprint and not added to the next sprint).
The lower daily rate applies for additional effort expended after the conclusion of the last sprint (see section 4.6).

(f) After the conclusion of a sprint, settlement in addition to the base rate is only for the features that were tested or made available for testing. Carryovers are also transferred to the next sprint in regard to the time already expended.

2. The agreed daily rates are established in Attachment 7 to this contract. A person-day is defined as 8 working hours. Fewer or more hours worked on the respective day are settled on a pro-rata basis. Settlement is based on performance records. With the invoice, the client receives a printout of the activities of the corresponding employees recorded in the contractor’s IT system for review. Once two weeks have elapsed since submission with no objections, the activity report is deemed to be approved.

3. [OPTIONAL: Bonus provision] The contractor receives a bonus payment for on-time delivery. The total amount of the bonus is [AMOUNT] percent of the billed services. The contractor is entitled to payment of the bonus when the following requirements are met:
   - Meeting the deadline [DATE].
   - Readiness for acceptance of the services to be provided by the specified dates.

Upon meeting the deadline [DATE], the contractor is entitled to an installment payment of [AMOUNT] € plus VAT as required by law. The installment payment is deducted from the bonus payment. The entitlement to the bonus payment is not eliminated in case of failure to meet the deadlines because the client fails to meet its contractual duties to cooperate. In this case, the deadlines are postponed accordingly.

4. Travel to the registered office of the client is already included in the quotation. Travel costs and expenses incurred for travel to other deployment locations by request of the client are reimbursed upon the presentation of vouchers or at the respective maximum amount according to tax laws.
   [ALTERNATIVE: Travel and incidental costs required for the performance of the contract are reimbursed to the contractor by the client upon the presentation of vouchers. Travel time expended is billed to the client at 50 percent of the daily rate pursuant to this contract.]

5. The requirement for payment is the submission of a verifiable invoice in proper form. Invoices are due for payment 30 days after the invoice is received by the client. A three percent discount may be deducted in case of payment within eight days. The due date of incorrect invoices is delayed accordingly.
   [ALTERNATIVE: Invoices are generally due for payment within 30 days after receipt of a verifiable invoice, with no deductions.]
Section 10 Termination

1. Ordinary termination of this project contract by the client is permitted pursuant to Section 649, sentence one of the German Civil Code (BGB). The following applies in this case:

(a) Compensation paid for sprints that have already been completed remains with the contractor in any case.

(b) If compensation has not been paid yet for a sprint that was already completed, this compensation is due for payment immediately upon receipt of a corresponding invoice from the contractor.

(c) If requirements for features have already been implemented in software that has been made available for testing at the time termination takes effect, but testing by the client is still pending, the client has to perform these tests notwithstanding termination and declare acceptability if the conditions are met. If the client fails to do so even after the contractor grants a period of grace, it is assumed that the corresponding requirements were properly implemented. This also applies if the client puts the software to use without reservations. In this case, the contractor is authorized to bill for the completed features of the ongoing sprint; the client is obligated to pay the corresponding compensation.

(d) For ongoing sprints, the client compensates the contractor for effort already expended at the time termination takes effect, according to the agreed daily rates in Attachment 7 to this contract and the base rate for the current sprint. The contractor provides proof of employee deployment by submitting corresponding activity records and delivers the software at the current state of development to the client.

(e) For services not yet provided, the client pays additional compensation equal to the planned effort to be expended until the end of the current sprint according to the daily rates agreed for the sprint in Attachment 7. The actually expended effort for any carryovers from previous sprints has to be compensated in addition. However, the contractor has to permit the deduction of any savings realized by the contractor by the waiver of performance, and any proceeds gained or maliciously failed to be gained by otherwise deploying its employees.

2. Either party has the right to extraordinary cancellation if the respective legal conditions are met. In regard to compensation for services already provided in whole or in part, the preceding sections of this contract apply correspondingly. Subsection 10.1.d, however, applies subject to the limitation that compensation is waived in regard to services for which the client states within four weeks after the notice of cancellation that they are of no interest to the client.

3. A notice of cancellation must be in written form in order to be effective. Submitting the notice by fax does not meet this written form requirement.
Section 11: Warranty for Material Defects

1. The contractor warrants that the software meets the contractually agreed characteristics.
2. The warranty term is one year. This short warranty term does not apply to claims for compensation based on a material defect pursuant to Section 634, No. 4 BGB in case of intent or the malicious concealment of a defect by the contractor, in case of the loss of life, physical injury, or the impairment of health, or in case of claims pursuant to the Product Liability Act (ProdHaftG).
3. Defects not already listed in the declaration of acceptance have to be reported by the client to the contractor promptly and no later than within two weeks after they are discovered. If a notice is not submitted in a timely manner, the object of performance is deemed to be approved in regard to this defect. Insofar asserting warranty claims is excluded.
4. As far as possible and to the extent reasonable for the client in view of the effects of the defect, the contractor is authorized to provide an interim solution to work around the defect until it is rectified. Such an interim solution blocks possible rights of the client pursuant to Section 634, No. 2–4 BGB.
5. The warranty obligation is waived if the client alters the object of performance itself or has it altered by third parties, unless the defect is not due to the alterations that were made.
6. If the contractor is not able to rectify a material defect after two attempts, the client is authorized to assert the additional statutory warranty claims.

Section 12: Warranty for Defects of Title

1. The contractor warrants that the software is free of third-party proprietary rights and that, to the best knowledge of the contractor, no other rights exist that limit or exclude use by the client pursuant to the contract.
2. In warranty cases, the contractor to an extent reasonable for the client has the right to either modify the software so that it no longer falls within the protection of the asserted right but still meets the requirements pursuant to the contract, or to obtain authorization so it can be used without restrictions pursuant to the contract and without additional costs for the client.
3. The warranty period is one year and begins with acceptance. However, in Subsection 11.2, sentence two of this contract applies correspondingly.
4. The parties shall inform each other promptly in writing if claims for the violation of proprietary rights are asserted against them.

Section 13: Liability

1. For damages of the client caused by intent or gross negligence, the lack of a guaranteed characteristic, a culpable violation of essential contractual obligations (known as cardinal duties), a culpable impairment of health, physical injury or the loss of life, or in case of liability pursuant to the Product Liability
Act (ProdHaftG), the contractor is liable pursuant to the applicable legal regulations.

2. Cardinal duties are contractual duties, the performance of which makes the proper performance of the contract possible in the first place, for which the contractual partner is entitled to trust in regular performance, and the violation of which endangers achieving the purpose of the contract by the other side.

3. If a cardinal duty is violated, liability insofar as damages are based merely on simple negligence and not on death, physical injury, or the impairment of health is limited to damages that can be typically expected to occur within the scope of a contractual relationship such as this one.

4. In case of simple negligence, liability insofar as damages are not based on death, physical injury, or the impairment of health nor a promised guarantee is also fundamentally limited to an amount of 2 million €.

5. Any other liability is excluded regardless of the cause in law, both on the part of the contractor and its assistants and vicarious agents.

6. In case of damages incurred by the client due to the loss of data, the contractor is only liable insofar as the damages would not have been prevented by a backup of all relevant data by the client as described in subsection 7.10 of this contract.

Section 14: Confidentiality

1. In regard to all information about the respective other party that has become or becomes known to them in the context of this contract, identified as confidential or identifiable as business or trade secrets of the respective other party based on other circumstances, the parties are obligated to permanently maintain secrecy even after the end of this contract and to refrain from dissemination to third parties, recording or any other form of exploitation, unless the affected party has consented to disclosure or exploitation expressly in writing.

2. Insofar as legally possible, the parties through suitable contractual agreements with their employees and all other persons working for them shall ensure that these persons also refrain from any disclosure, exploitation, dissemination, or recording of the confidential information.

Section 15: Data Privacy

1. The parties shall observe the applicable legal regulations for the collection, processing, and use of personal data within the scope of this contract.

2. [OPTIONAL: Test data used by the client may not include actual personal data.]

3. Should a situation corresponding to Section 11, Paragraph 5 of the Federal Data Protection Act (BDSG) arise in the course of performance pursuant to this contract, the parties shall conclude a job-order data processing agreement according to the requirements of Section 11 BDSG.
Section 16: Advertising and Investor Relations

1. With the consent of the client and no sooner than after the commencement of operation, the contractor is authorized to issue a press release regarding conclusion of the contract. The client shall not refuse consent without justifiable cause.

2. With the consent of the client and no sooner than after the commencement of operation, the contractor is authorized to name the client on the Web site and at the exhibition stands of the contractor as a client and to use the client’s company logo for these purposes. The client shall not refuse consent without justifiable cause.

3. Furthermore, the client with consent and no sooner than after the commencement of operation permits the publication of a project report. The client shall not refuse consent without justifiable cause. The client shall also be available to future prospects of the contractor as a reference contact.

Section 17: Choice of Law, Jurisdiction, and Place of Performance

1. For this contract and in regard to all legal relationships arising from the contract, the parties agree on the application of the laws of the Federal Republic of Germany. The application of the United Nations Convention on the International Sale of Goods as well as German and European international civil law is excluded.

2. The jurisdiction for all disputes arising from or in the context of this contract, and the place of performance, is [PLACE].

Section 18: Ranking

1. The ranking of the contractual agreements is as follows:

   (a) Individual amendments and/or endorsements to this contract after it is concluded
   (b) This contract without attachments
   (c) Attachment 4 to this contract with the appendices to Attachment 4 and all documents equivalent to these appendices
   (d) All other attachments to this contract

2. In case of contradictions, the provisions named first always take precedence over those named last. Gaps are filled by the respective subordinate provision. The same applies to amendments contained in the subordinate provisions. In case of documents with the same ranking, the more recent document takes precedence over the older document.
Section 19: Final Provisions

1. This contract including its attachments contains the entire agreement between the parties regarding the object of the contract. In particular, the general business terms and conditions of the parties do not apply.

2. Amendments or endorsements as well as the cancellation of this contract must be in written form. This also applies to the waiver of the written form requirement itself.

3. Should provisions of this contract become ineffective in whole or in part, the remaining provisions of this contact shall remain unaffected. The ineffective, incomplete, or infeasible provision shall be replaced by the applicable laws. If a suitable regulation or suitable legal principle to amend the contract is lacking, and eliminating the clause does not offer a solution that protects the interests of the parties, the gap shall be filled by the supplementary interpretation of the contract. In this case, a provision is deemed to be agreed that comes as close as possible to the original object and purpose of the ineffective, incomplete, or infeasible provision.

Signatures

Contractor

Client

Place, date

Place, date

(Name in block letters)  (Name in block letters)

Attachment 1: High-Level Specification

[...]

Attachment 2: Initial Estimate of Effort and Base Rate

[...]

Attachment 3: Prioritization of the Features

If the prioritization of the requirements formulated in the features changes by request of the client or if new features are added, the changes are recorded in an appendix to this Attachment 3. The parties agree that the prioritization details required for the sprints can also be recorded in other documents on a case-by-case basis and that these documents do not necessarily have to be physically connected to this Attachment 3 to the project contract insofar as they expressly or implicitly refer to this Attachment 3 to the project contract.

[...]
Attachment 4: Duration, Planned Budget, and Detailed Specification for the Individual Sprints

For each sprint, the appendices attached to this Attachment 4 to the project contract contain:

1. The duration of the sprint
2. The planned budget for the sprint
3. If applicable, a description of the features for the sprint

The parties agree that the information required for the sprint can also be recorded in other documents on a case-by-case basis and that these documents do not necessarily have to be physically connected to this Attachment 4 to the project contract, insofar as they expressly or implicitly refer to this Attachment 4 of the project contract or Section 4 of the project contract.

[...]

Attachment 5: Project Organization

1. The project manager on the contractor side is Mr./Ms. [NAME].
2. The project manager on the client side is Mr./Ms. [NAME].
   His/her deputy is Mr./Ms. [NAME].
3. The members of the steering committee are:
   Mr./Ms. [NAME]
   Mr./Ms. [NAME]
   [...]

Attachment 6: Software and Corresponding Licenses Provided by the Client

[...]

Attachment 7: Daily Rates

1. As the regular daily rate for all development and programming effort within the target budget, the parties agree on the net amount of [AMOUNT] €.
2. As the reduced daily rate for all development and programming effort expended outside the target budget and all error correction and similar measures during an iteration, the parties agree on the net amount of [AMOUNT] €.

[ALTERNATIVE: The daily rates according to the quotation apply.]
Index

A
Abstraction, 17, 18, 41
Accuracy (annotation), 302
  on interaction canvas, 136
  on object canvas, 108
  on partner canvas, 72
  on physical object canvas, 80
Actual effort, 160, 218, 268
adVANTAGE, 205
  applicability, 236, 278
  example, 229, 269
  principles, 206
  procedures, 213
AE. See actual effort
Agile development, 8, 49, 150, 178, 205
Agility, 5, 206, 287, 294
Annotation, 43, 56, 154, 299
  analysis, 45, 260
  documentation, 310
  in IR:digital, 84
  in IR:mobile, 137
  in IR:scope, 114
  in IR:tech, 145
  method, 46, 275
  on feature canvas in IR:scope, 94, 245
  on feature canvas in IR:tech, 143
  on integration canvas in IR:scope, 112, 259
  on interaction canvas, 135
  on object canvas in IR:scope, 106, 255
  on partner canvas, 72
  on persona canvas, 123
  on physical object canvas, 79
  on portfolio canvas, 125
  on process canvas in IR:scope, 99, 252
  on touchpoint canvas in IR:mobile, 83, 129
Application developer, 92
Architecture, 283
Attractiveness (annotation), 304
  on interaction canvas, 136
  on touchpoint canvas in IR:mobile, 130
AugIR. See Interaction Room, augmented
Automation, 60, 284
Automation (annotation), 305
  on object canvas, 108
  on process canvas, 102
B
Backlog, 56
Base rate, 214, 218, 234, 269
Big data, 288
Billing, 221, 269
BizDevOps, 290
BR. See base rate
Business data, 103
Business department, 25
Business developer, 121
Business object, 103
Business process, 69, 78, 95, 134, 154
Business value (annotation), 299
  on feature canvas in IR:scope, 94
  on feature canvas in IR:tech, 143
  on interaction canvas, 136
  on object canvas, 108
  on persona canvas, 124
  on physical object canvas, 80
  on portfolio canvas, 126
  on process canvas, 101
  on touchpoint canvas in IR:mobile, 130
C
Canvas, 41, 56, 155, 168, 275
  in IR:digital, 84
  in IR:mobile, 137
  in IR:scope, 114
  in IR:tech, 145
Change request, 185, 210, 222, 232
Channel, 81
Clarity, 25
Client, 154, 179, 181, 294
Client object, 75
Cloud computing, 3, 6, 288
Coach, 51
Domain expert. See domain coach
Method expert. See method coach
Communication, 19, 25, 167, 171, 292
Company boundary, 177, 181
Completeness, 23
Complexity (annotation), 307
on feature canvas in IR:scope, 94
on feature canvas in IR:tech, 144
on integration canvas, 113
on object canvas, 108
on physical object canvas, 80
on process canvas, 102
Computer scientist’s forecast, 161, 270, 276
Context boundary (of system), 22
Context (of touchpoint), 81
Continuous integration, 6, 287
Contract for work and labor, 183
Contract model
agile, 195, 202
multi-stage, 200
traditional, 181, 191
Contractor, 179, 181, 190, 294
Cooperative performance structure, 179
Cost forward progressing, 33, 155, 159, 270, 276
Customer journey map, 81
Customer safari, 81
Cyber-physical system, 61

D Daily rate, 211, 232, 269
reduced, 221, 223, 234
regular, 215, 234
Daily scrum, 214
DE. See detailed estimate
\( \Delta DE \). See deviation (average), between \( AE \) and \( DE \)
Definition of Done, 207, 222, 236, 268
Deprecation (annotation), 308
on integration canvas, 113
on object canvas, 109
Detailed estimate, 160, 218, 267, 269
Development risk, 192, 210, 222
Deviation (average), 276
between \( AE \) and \( DE \), 162, 270
between \( DE \) and \( IE \), 161, 271
Device, 286
DevOps, 290, 294
Dialog, 132
Dialog flow, 134
Digital business expert, 65
Digital company, 60, 74
Digitalization, 3, 48, 59, 284
Digital technology expert, 67
Disaster, 33, 153
Disaster point, 157, 265, 277
Display, 168
DoD. See definition of done
Domain coach, 53
Domain knowledge, 20, 273, 289, 294
\( DR_1 \). See daily rate, regular
\( DR_2 \). See daily rate, reduced

E Earned value analysis, 219
Efficiency incentive, 211, 223, 234
Effort, 154
actual. See actual effort
detailed estimate. See detailed estimate
initial estimate. See initial estimate
overall estimate. See overall estimate
remaining. See remaining effort
Effort driver, 26, 45, 301
Effort tracking, 218
EI. See efficiency incentive
Elasticity, 3, 5, 49
Enterprise architect, 142
Enterprise IT, vii, 3, 25, 283, 289, 291, 293
Estimate, 218, 276
detailed (effort). See detailed estimate
initial (budget). See initial budget
initial (effort). See initial estimate
overall (effort). See overall estimate
External resource (annotation), 309
on integration canvas, 114
on object canvas, 109
on process canvas, 102
External service provider, 179

F Feature, 159
Feature canvas, 274
example, 245
in IR:agile, 150
in IR:scope, 93
in IR:tech, 142
Feature point, 197
Fixed price, 183, 190
per iteration, 195, 201
per point, 196, 201
per project, 184, 192
Flexibility, 288
Flexibility (annotation), 304
on feature canvas in IR:tech, 144
on interaction canvas, 136
on object canvas, 108
on physical object canvas, 80
on process canvas, 101

Follow-up activities
example, 261
IR:digital, 86
IR:mobile, 138
IR:scope, 116
IR:Tech, 146

Function point, 197, 199

H
Health insurance benefit system, 241
HIB system. See health insurance benefit system

High use (annotation), 302
on feature canvas in IR:tech, 143
on integration canvas, 113
on partner canvas, 72
on physical object canvas, 80
on process canvas, 101

I
IB. See initial budget
IE. See initial estimate
ΔIE. See deviation (average), between DE and IE

Improvement, need for (annotation). See need for improvement (annotation)
Industry 4.0, 62
Information technology, vii
In-house development, 179
Initial budget, 215, 269
Initial estimate, 159, 213, 218, 263

Innovation (annotation), 300
on feature canvas in IR:tech, 143
on interaction canvas, 136
on portfolio canvas, 126
on touchpoint canvas in IR:mobile, 130

Insurance benefit, 241
Integration, 287
Integration canvas, 132, 274
example, 258
in IR:scope, 109
in IR:tech, 145
Interaction channel, 128
Interaction engineer, 68
Interaction process, 134
Interaction Room, 17, 40
applicability, 172
augmented, 168
coach, 51
distributed, 167, 169

follow-up activities, 56
for agile project monitoring, 49, 149, 263
for digitalization strategy development, 48, 59
for mobile application development, 48, 119
for software project scoping, 49, 91, 243
for technology evaluation, 49, 141
layout, 18, 165
method, 40
principles, 18
results, 56, 276
temporary, 165
variant, 48, 171
workshop, 55, 295

Interaction trigger, 128
Interface, 69

Invariability (annotation), 308
on integration canvas, 113
on object canvas, 109

IR. See Interaction Room
IR:agile. See Interaction Room, for agile project monitoring
IR:digital. See Interaction Room, for digitalization strategy development
IR:mobile. See Interaction Room, for mobile application development
IR:scope. See Interaction Room, for software project scoping
IR:tech. See Interaction Room, for technology evaluation
IT. See information technology
Iteration, 196

K
Kano model, 94

L
Lean software, v, 8, 13, 18, 27, 29, 153, 277, 290, 291

M
Manual task (annotation), 306
on object canvas, 108
on process canvas, 102
Maturity level, 190
Method coach, 46, 52
Mobility, 3, 4, 48, 60, 119, 284, 293
Mobility (annotation), 305
on integration canvas, 113
on object canvas, 108
on portfolio canvas, 126
in fixed price per iteration contracts, 195
in fixed price per point contracts, 196
in Money for Nothing, Change for Free contracts, 197
in multi-stage contracts, 200
in pay-per-use contracts, 189
in service contracts, 184
in shared pain/shared gain contracts, 198
in time and materials contracts, 187
in traditional contract models, 191
Risk driver, 216
Risk management, 157
Risk map, 34, 153, 263, 265, 277
Role, 52, 54

S
SaaS. See software-as-a-service
Scope, 21
Scrum master, 214, 269
Security, 286
Security (annotation), 303
    on feature canvas in IR:tech, 144
    on integration canvas, 113
    on object canvas, 108
    on partner canvas, 72
    on process canvas, 101
Semantics, 25
Service blueprint, 81
Service contract, 183
Settlement, 221, 269
    fully completed sprint, 222
    partially completed sprint, 224
Shared pain/shared gain, 198, 201
Skill, 283
Socio-technical system, vi, 177, 186, 292
Software, v
Software-as-a-service, 188
Software development. See software engineering
Software engineering, v, 283
Software metrics, 27
Software process, vi
Sprint, 160, 196
Sprint backlog, 160, 216, 267
Sprint completion
    full, 221
    partial, 224
Sprint planning, 151, 155, 160, 166, 226, 267
Sprint scope, 216, 233, 267
Stakeholder, 51, 54, 167, 283, 291, 294
    example, 244
    in IR:digital, 64
    in IR:mobile, 120
in IR:scope, 92
in IR:tech, 142
Statistician’s extrapolation, 161, 270, 276
Storyboard, 132
Story point, 197
Supplier, 179, 181
System
    boundary, 22, 154
    context, 21
    of engagement, 22
    of records, 21

T
T&M. See time and materials
Tailoring, 56
Tamed agility, 9, 11, 13, 291
Team, 51
distributed, 167
Team room, 166
Technology expert, 142
Telecommunication, 285
Termination, 198, 226
Testing, 287
Time and materials, 183, 187, 190, 192
Time constraint (annotation), 302
    on feature canvas in IR:tech, 143
    on physical object canvas, 80
    on process canvas, 101
Timeline, 81
Touchpoint, 81, 127
Touchpoint canvas, 275
    in IR:digital, 81
    in IR:mobile, 127
Touchpoint event, 81
Touchpoint lane, 82
Transparency, 19
Trust, 34, 200, 207, 222, 228, 230
Trust point, 81, 127

U
Uncertainty, vi, 30, 177, 273
Uncertainty (annotation), 46, 47, 309
    on feature canvas in IR:scope, 94
    on feature canvas in IR:tech, 144
    on integration canvas, 114
    on interaction canvas, 136
    on object canvas, 109
    on process canvas, 102
    on touchpoint canvas in IR:mobile, 130
Underspend, 218
US. See underspend
Usability (annotation), 304
    on interaction canvas, 136
User, 93
User experience, 132, 284, 285, 293
User journey, 127
User value (annotation), 300
  on feature canvas in IR:scope, 94
  on feature canvas in IR:tech, 143
  on interaction canvas, 136
  on object canvas, 108
  on physical object canvas, 80
  on portfolio canvas, 126
  on process canvas, 101
  on touchpoint canvas in IR:mobile, 130
UX. See user experience

V
Value, 19, 26
Value driver, 26, 45, 154, 197, 216, 294, 299
Value risk, 192, 210, 222

W
Whiteboard, 39, 168
Workshop, 165, 168, 295
  IR:digital, 86, 295
  IR:mobile, 138, 297
  IR:scope, 116, 296
  IR:tech, 146, 297
  preparation, 55