

Inhaltsverzeichnis

Über dieses Dokument.....	1
Was ist eine Lino-Anwendung?.....	1
Systemkomponenten.....	2
Aufgaben des Auftraggebers.....	2
Aufgaben des Dienstleisters.....	3
Support.....	3
Lebenslauf eines Projektes.....	3
Neuentwicklungsprojekte.....	4
Produktionsbetrieb.....	4
Unsere Arbeit.....	5
Gemeinsame Entwicklungsphase.....	5
Aktualisierungen einer Instanz.....	6
Qualitätskontrolle.....	6
Einfaches Testverfahren.....	6
Erweitertes Testverfahren.....	7
Abrechnungsarten.....	7
Abrechnung per Stundenkredit.....	8
Pauschale Abrechnung.....	8
Engpässe und Deadlines.....	9
Datenschutz.....	9
Sonstiges.....	9

Über dieses Dokument

(1) Dieses Dokument beschreibt die allgemeinen Richtlinien für unsere Zusammenarbeit. Angebote und andere Dokumente, die sich auf dieses Dokument beziehen, können spezifische Optionen präzisieren.

(2) Kooperationspartner sind einerseits der **Auftraggeber** als Nutzer einer *Lino-Instanz* sowie andererseits der **Dienstleister** als technischer Verwalter dieser Instanz und Entwickler der darin verwendeten *Lino-Anwendung*.

Was ist eine Lino-Anwendung?

(3) Eine **Lino-Anwendung** ist eine den Bedürfnissen des *Auftraggebers* angepasste Computeranwendung, die auf einem *Server* installiert und von den *Clients* aus bedient werden kann.

(4) Eine **Lino-Instanz** ist, wenn eine bestimmte Lino-Anwendung auf einem bestimmten *Server* läuft. Jede Instanz hat ihren eigenen Domainnamen, ihre eigene Datenbank sowie eigene lokalen Einstellungen und Konfigurationsdateien zur Einbindung der *Lino-Anwendung* in die *Systemsoftware*. Auf einem *Server* können eine oder mehrere *Lino-Instanzen* laufen.

(5) Der Auftraggeber hat **Hoheit** über die *Lino-Anwendung*, d.h. er entscheidet über Funktionsumfang und Prioritäten bei der Entwicklung der Anwendung.

(6) Die *Lino-Anwendung* ist Freie Software und kann dadurch potentiell auch durch andere Organisationen und/oder Entwickler genutzt bzw. verändert werden. Der *Auftraggeber* ist nicht an den *Dienstleister* gebunden und kann sowohl Entwicklung als auch Wartung jederzeit einem anderen Anbieter übertragen.

(7) Lino-Anwendungen nutzen das **Lino-Framework**. Das *Lino-Framework* besteht aus dem *inneren Framework* und der *Extensions Library*.

(8) Das **innere Framework** bietet die Infrastruktur, mit der **Grundfunktionalitäten** wie Benutzern, Zugriffsrechte, Ansichtparameter, Arbeitsabläufe, Notifikationen, Berichte und Ausdrücke,

mehrsprachige Benutzerschnittstelle und mehrsprachige Kontakte usw. konfigurierbar sind.

(9) Die **Extensions Library** ist eine Sammlung von Modulen, die potentiell auch in anderen *Lino-Anwendungen* verwendet werden können.

(10) Jede Lino-Anwendung nutzt ihrerseits eine Vielzahl von weiteren freien Softwaremodulen (unter anderem Django, ExtJS, dutzende von kleineren Python-Modulen), die in einer **Python-Umgebung** auf dem Server installiert werden.

Systemkomponenten

(11) Das Gesamtsystem besteht aus *Hardware*, *Systemsoftware*, *Anwendungssoftware*, lokalen Konfigurationsdateien und *Dokumentation*.

(12) Die **Hardware** besteht aus (a) dem **Server**, d.h. einem eigenständigen (normalerweise virtuellen) Rechner innerhalb oder außerhalb der Gebäude des Auftraggebers, (b) den **Clients**, d.h. den Desktop-oder Notebook-Computern der Endbenutzer, sowie (c) dem **Netzwerk**, das die Verbindung der beiden ermöglicht.

(13) Unter **Systemsoftware** verstehen wir alle Software unterhalb von Lino. Dazu gehören das Betriebssystem sowie Systemdienste wie Web-Server, Datenbank-Server, Mail-Server, File-Server, cron, logrotate, supervisor. Wir verwenden üblicherweise *Debian* als Betriebssystem, *Apache* als Web-Server, *MySQL* oder *PostgreSQL* als Datenbank-Server, *Postfix* als Mail-Server.

(14) Unter **Anwendungssoftware** verstehen wir die im *Entwicklungs-* oder *Wartungsauftrag* benannte *Lino-Instanz*.

Aufgaben des Auftraggebers

(15) Der *Auftraggeber* ist verantwortlich für die Bereitstellung und Wartung der *Hardware*, einer *gesicherten Zugriffsmöglichkeit für den Dienstleister* auf den Server, sowie **Backups** aller Daten auf dem Server.

(16) Der *Auftraggeber* benennt eine **Kontaktperson**, die in allen Fragen zwischen *Dienstleister* und *Auftraggeber* vermittelt.

(17) Die *Kontaktperson* formuliert die *Anfragen* des *Auftraggebers* und deren Dringlichkeit, beantwortet *Rückfragen* des Dienstleisters und liefert ihm alle nötigen Informationen, die er zur Lösungsfindung braucht.

(18) Die *Kontaktperson* ist dem Dienstleister gegenüber zuständig für alle Fragen unserer Zusammenarbeit. Fragen, die ihre Kompetenz überschreiten, leitet sie intern weiter und rechtfertigt und koordiniert ihre Entscheidungen mit den Anforderungen des Auftraggebers.

(19) Die Kontaktperson ist mindestens per E-Mail erreichbar, idealerweise auch telefonisch, über *instant messaging* oder Internet-Telefon. Der *Auftraggeber* organisiert die Vertretung bzw. den Ersatz der *Kontaktperson* bei deren **Abwesenheit**.

(20) Der Auftraggeber organisiert **Support und Schulung** für die *Endbenutzer*

(21) Der Auftraggeber ist verantwortlich für das Vergeben und Zurücknehmen von **Zugriffsrechten** an die *Endbenutzer*.

(22) Der Auftraggeber teilt dem Dienstleister mit, wer außer dem Dienstleister noch Zugriff auf den Server hat.

Aufgaben des Dienstleisters

(23) Der *Dienstleister* ist zuständig für Einrichtung und Wartung der *Instanzen* auf dem Server. Er führt alle Einsätze auf dem Server gewissenhaft durch, nach Möglichkeit ohne einen eventuellen Produktionsbetrieb zu gefährden.

(24) Der Dienstleister gewährt *Support* an die *Kontaktperson* des *Auftraggebers*. (Siehe eigene Sektion „Support“).

(25) Der Dienstleister veröffentlicht den Quellcode der Anwendung als **Freie Software**.

(26) Der *Dienstleister* veröffentlicht und wartet **technische Dokumentation** über die *Anwendungssoftware* in englischer Sprache. Die *technische Dokumentation* hat als Zielgruppe andere Entwickler.



(27) Der *Dienstleister* veröffentlicht und wartet **allgemeine Dokumentation** für Endbenutzer von *Lino-Anwendungen* in englischer Sprache. Auf Wunsch veröffentlicht und wartet der *Dienstleister* zusätzlich anwendungsspezifische Dokumentation in der Sprache des Auftraggebers.

(28) Der *Dienstleister* wirkt bei der **allgemeinen Entwicklung des Frameworks** mit. *Dazu gehören* z.B. Entwicklung neuer Front-Ends, Verbesserungen der Entwicklerschnittstelle, die *Qualitätssicherung* und Dokumentation, sowie funktionale Änderungen von allgemeinem Interesse.

(29) Als **Qualitätssicherung** bezeichnen wir das Schreiben und Pflegen von automatisierten Testverfahren. Dazu gehören **doctests** auf der technischen Dokumentation und **unit tests** auf *Demo-Projekten*. Ein **Demo-Projekt** ist eine öffentlich einsehbare Instanz einer Lino-Anwendung mit fiktiven Daten, die zum Vorzeigen und Testen der Anwendung benutzt wird.

Support

(30) Der Dienstleister ist werktags von 8 bis 18 Uhr **telefonisch erreichbar**.

(31) Der Dienstleister garantiert nach vorheriger Absprache **erhöhte Bereitschaft**, bei der ein Mitarbeiter Zeit und Ressourcen hat für operative Eingriffe über Internet.

(32) Der Dienstleister teilt dem Auftraggeber einen **Notrufplan** mit und aktualisiert diesen wenn nötig. Der Notrufplan enthält Anweisungen und Telefonnummern für den Fall, dass ein Mitarbeiter des Dienstleisters nicht per Telefon antwortet.

(33) Der *Dienstleister* **bestätigt** jede schriftliche *Anfrage* des *Auftraggebers* innerhalb eines Arbeitstages. Wenn nicht anders vereinbart, gelten die **Feiertage** in Estland. Siehe [Wikipedia](#).

Lebenslauf eines Projektes

(34) Im Laufe der Jahre durchläuft unsere Zusammenarbeit verschiedene Phasen.

(35) Wir unterscheiden zwischen *Neuentwicklungsprojekten* und *Produktionsbetrieb*. Im *Produktionsbetrieb* können gegebenenfalls **Weiterentwicklungsprojekte** gestartet werden, d.h. *Entwicklungsprojekte*, die parallel mit dem Wartungsauftrag auf einer *Instanz* im *Produktionsbetrieb* laufen.

Schritt	Endet durch
Interview und Vorgespräche	Entwicklungsangebot mit geplanten Tätigkeiten und Bedingungen
Interne Entwicklungsphase	Abnahme des Prototypen
Gemeinsame Entwicklungsphase	Wechsel zum Produktionsbetrieb
Produktionsbetrieb	Ende der Zusammenarbeit

Neuentwicklungsprojekte

(36) Ziel eines **Neuentwicklungsprojektes** ist die Inbetriebnahme einer neuen *Instanz*, auf der die *Software* läuft und benutzt werden kann.

(37) Bei einem *Neuentwicklungsprojekt* durchläuft unsere Zusammenarbeit üblicherweise folgende Phasen und Entwicklungsschritte. Die Dauer jeder Phase hängt von vereinbarten Terminen, Verfügbarkeit der *Kontaktperson* und den sich ergebenden Fragestellungen ab.

(38) Der Dienstleister trifft sich mit dem Auftraggeber zu einem **Interview**. Ziel des Interviews ist es, die Bedürfnisse zu analysieren und zu klären, ob eine Lino-Anwendung dabei helfen könnte und welche Funktionen sie idealerweise haben müsste.

(39) Der Dienstleister schreibt ein *Angebot*, in dem die beim Interview besprochenen Funktionen der zu entwickelnden Anwendung beschrieben sind sowie eine Schätzung der **Entwicklungskosten**. Diese unterteilen sich in (a) das **Einrichten des Prototypen** und (b) die **gemeinsame Entwicklungsphase**.

(40) Wenn der *Auftraggeber* den *Entwicklungsauftrag* bestätigt hat, beginnt der *Dienstleister* mit dem **Einrichten des Prototypen**. Diese Phase wird auch als **interne Entwicklungsphase** bezeichnet. Gegen Ende dieser Phase präsentiert der *Dienstleister* dem *Auftraggeber* den *Prototy-*



pen.

(41) Ein **Prototyp** ist eine *Lino-Instanz* mit fiktiven Demo-Daten, die als Arbeitsgrundlage für die *aktive Entwicklungsphase* dient. Der *Prototyp* kann entweder auf einem vom Dienstleister betriebenen Server installiert sein, oder direkt auf dem Server, auf dem die *Software* später im *Produktionsbetrieb* laufen wird.

(42) Mit der **Abnahme des Prototypen** beginnt die *gemeinsame Entwicklungsphase*. Der *Dienstleister* schreibt eine Rechnung über das *Einrichten des Prototypen* und ein Angebot für die *gemeinsame Entwicklungsphase*.

(43) Gegen Ende der *gemeinsamen Entwicklungsphase* ist die Anwendung so weit fertig, dass die Datenbank bei Aktualisierungen nicht mehr gelöscht sondern *migriert* wird. Es beginnt ein vorsichtiger **probeweiser Produktionsbetrieb**, bei dem der Dienstleister noch keine Verantwortung für reibungslosen Ablauf trägt. In dieser Phase werden die fiktiven Testdaten gelöscht, die Konfigurationsdaten angepasst, eventuell werden Daten importiert aus bestehenden Quellen. Die *Kontaktperson* organisiert Schulung der Endbenutzer und Erfassung erster Daten.

(44) Mit der **Abnahme des Neuentwicklungsprojekts** bestätigt der *Auftraggeber*, dass die Instanz in den *Produktionsbetrieb* wechseln kann.

Produktionsbetrieb

(45) Oberstes Ziel unserer Zusammenarbeit bei einer Instanz im **Produktionsbetrieb** ist die Aufrechterhaltung des zuverlässigen Funktionierens.

(46) Der *Produktionsbetrieb* einer *Instanz* beginnt, wenn der *Auftraggeber* das entsprechende Angebot des *Dienstleisters* bestätigt.

(47) Ebenso wichtig ist die Möglichkeit, bei Bedarf Anpassungen der Software aufzuspielen. Siehe hierzu auch das Kapitel **Aktualisierungen einer Instanz**.

(48) Der Dienstleister garantiert weder **Fehlerfreiheit** noch **Ausfallsicherheit** der Instanz im *Produktionsbetrieb*, sondern seine Hilfestellung bei Problemen. Sollte jemand aufgrund eines Fehlers oder Ausfalls zu **Schaden** kommen, kann der Dienstleister dafür **nicht haftbar** gemacht werden. Optional können erweiterte **Garantien** vereinbart werden.

Unsere Arbeit

(49) Der Auftraggeber ist verantwortlich für *Analyse und Planung* eines Weiterentwicklungsprojekts. Der Dienstleister ist verantwortlich für Implementierung und Ausführung der vereinbarten Schritte zu den vereinbarten Zeitpunkten.

(50) Eine **Anfrage** ist die Meldung des *Auftraggebers* (a) eines Problems bei der Benutzung der Software oder (b) eines Änderungswunsches.

(51) Eine **Rückfrage** des *Dienstleisters* ist die Mitteilung, dass der Dienstleister mehr Informationen benötigt, um eine Anfrage bearbeiten zu können.

(52) Als **Analyse** bezeichnen wir die Erfassung eines Problems, *Rückfragen* des Dienstleisters zur Präzisierung einer *Anfrage*, die Suche nach Ursachen und das Erarbeiten von Lösungsvorschlägen.

(53) **Optimierungen** sind Änderungen der Software, für die sich ein eigenes Angebot nicht lohnt.

(54) Als **Support** bezeichnen wir die Beantwortung von Fragen zur Benutzung der Software.

(55) Als **Implementierung** bezeichnen wir die Änderungen im Quellcode, der Dokumentation und der lokalen Konfiguration, den Support, die eventuelle weiterführende Analyse und die Durchführung aller Einsätze, die zur Lösung eines Problems nötig sind.

(56) Als **technologisch begründete Änderungen** bezeichnen wir Anpassungen bestehender Funktionen der Software, die durch Änderungen in anderen Softwaresystemen nötig sind. Dazu gehören (a) Verbesserungen im *Lino-Framework*, (b) Schnittstellen zu Diensten von Drittanbietern (BeID, ZDSS, ...), (c) Änderungen in Software, die Lino benutzt (Python, Django, ...) und (d) allgemeine Entwicklungen des Internet (Front-end, Sicherheit, Authentifizierung, ...)

(57) Wenn es verschiedene Alternativen zur technischen Implementierung einer angefragten Änderung gibt, erklärt der Dienstleister die Unterschiede und Auswirkungen dieser Alternativen, und



der Auftraggeber entscheidet sich für eine davon.

Gemeinsame Entwicklungsphase

(58) Eine **gemeinsame Entwicklungsphase** kann sowohl bei Neu- als auch bei Weiterentwicklungsprojekten stattfinden. Sie beginnt, wenn der Auftraggeber das entsprechende Angebot des Dienstleisters bestätigt.

(59) *Dienstleister* und *Kontaktperson* arbeiten während dieser Phase auf einer *dafür vorgesehenen Instanz*, die sich ständig verändert und weiterentwickelt. Dies kann ein *Prototyp* oder eine *Vorschau* sein.

(60) In einer *gemeinsamen Entwicklungsphase* wiederholen sich die folgenden Etappen zyklisch: **Baustellenbesichtigung** mit der Kontaktperson, Bearbeitung der festgestellten Mängel, Entwicklung einer neuen Version und Vorbereitung der nächsten *Baustellenbesichtigung*.

(61) Die gemeinsame Entwicklungsphase endet mit der *Abnahme* des Entwicklungsprojekts.

Aktualisierungen einer Instanz

(62) Jede *Instanz* ist individuell auf die Bedürfnisse des Auftraggebers zugeschnitten und kann bei Änderungen der Bedürfnisse angepasst werden.

(63) Der Dienstleister ändert die Anwendung nur (a) auf *Anfrage* des Auftraggebers oder (b) als vom Auftraggeber akzeptierte **Anpassung** an *technologisch bedingte Änderungen*.

(64) Auf Wunsch des Auftraggebers dokumentiert der Dienstleister vor der Anpassung alle Änderungen, die Einfluss auf die Arbeit der Endbenutzer haben.

(65) Ein **Einsatz** ist ein vom *Dienstleister* durchgeführter Eingriff auf den *Server* des *Auftraggebers*, bei dem die Konfigurierung des Servers geändert und/oder die Anwendungssoftware aktualisiert wird.

(66) Bei einer **Aktualisierung** wird neuer Quellcode der *Anwendungssoftware* auf die *Instanz* geladen und aktiviert. Wir unterscheiden zwei Arten von Aktualisierungen: *Releases* und *einfache Aktualisierungen*.

(67) Ein **Release** ist eine *Aktualisierung*, für die eine eigene Versionsnummer und eigene **Releasenotizen** erstellt und gewartet werden. Ein Release beginnt mit Analyse und Formulierung der Ziele, dem Einrichtung einer Instanz, durchläuft ggf. eine *gemeinsame Entwicklungsphase*, bei der die *Releasenotizen* aktualisiert werden, und endet mit der **Abnahme**, d.h. dem gemeinsamen Einverständnis, dass die geplanten Ziele erreicht sind.

(68) Eine **einfache Aktualisierung** kann durchgeführt werden, um den administrativen Aufwand eines Releases zu umgehen. Dabei wird keine neue Versionsnummer vergeben und keine neue Testphase durchgeführt, und die *Releasenotizen* der laufenden Version werden lediglich aktualisiert.

(69) Bei jeder *Aktualisierung* ist der *Dienstleister* verantwortlich für eine korrekte **Datenmigration**.

Qualitätskontrolle

(70) Lino enthält eine umfangreiche **Testsuite** zum automatisierten Testen vieler Funktionen. Die Testsuite wird vor jedem Release durchgeführt. Die *Testsuite* kann jedoch keine absolute Sicherheit vor Problemen nach einem *Einsatz* geben.

(71) Eine **Regression** ist, wenn etwas vor einer Aktualisierung funktionierte und nachher nicht mehr funktioniert.

(72) Eine **Nebenwirkung** ist etwas, das nach einer Aktualisierung anders funktioniert als zuvor und deshalb potentiell Verwirrung oder Supportbedarf schafft.

(73) Eine Aktualisierung kann **unausweichliche technologisch bedingte Anpassungen** mit sich bringen. In diesem Fall klärt der Dienstleister deren *Abnahme* mit der *Kontaktperson*, bevor die Aktualisierung in Produktion geht.

(74) Jede Aktualisierung enthält prinzipiell ein **Risiko**, den *Produktionsbetrieb* zu stören. Um dieses Risiko möglichst gering zu halten, planen *Kontaktperson* und *Dienstleister* jedes Release gemeinsam und gewissenhaft.

(75) Wir bieten zwei unterschiedliche Verfahren zur Durchführung von Releases. Im einfachen Testverfahren sind **Flexibilität** und **Aufwand** prioritär, im erweiterten Testverfahren dagegen **Stabilität** und **Planbarkeit**.

Einfaches Testverfahren

(76) Der Vorteil des **einfachen Testverfahrens** besteht darin, dass es kostengünstig und flexibel ist.

(77) Bei diesem Verfahren werden neue Versionen nach internem Test durch den Dienstleister **über Nacht** aufgespielt.

(78) Zuvor haben Dienstleister und Auftraggeber für den nächsten Arbeitstag eine **kurze Testphase** vereinbart, die maximal mehrere Stunden dauern kann. In dieser Zeit testen ausgewählte Endbenutzer ihren neuen Lino im laufenden Produktionsbetrieb. Sie sind dabei auf *Regressionen* und Überraschungen gefasst, die sie dann melden müssen und die dann prioritär bearbeitet werden.

(79) Falls ein Problem gemeldet wird, das den Produktionsbetrieb gefährdet, wird die Instanz auf den Zustand vom Vortag zurück gesetzt.

(80) Der Auftraggeber ist dafür zuständig, dass alle Endbenutzer über die Testphase informiert und sich dessen bewusst sind, dass der Einsatz eventuell rückgängig gemacht werden kann.

Erweitertes Testverfahren

(81) Beim **erweiterten Testverfahren** arbeiten wir mit einer *Vorschau* auf die kommende Version, die von ausgewählten Benutzern über eine längere Zeit hinweg gründlich getestet wird, bevor sie in Produktion geht.

(82) Der Vorteil des *erweiterten Testverfahrens* besteht darin, dass es reibungslosen Produktionsbetrieb ohne Unterbrechung gewährleistet.

(83) Eine **Vorschau** ist eine zusätzliche Instanz neben der Produktionsinstanz, üblicherweise auf dem gleichen Server, der eine Kopie der Produktionsdaten zeigt, wie sie mit der nächsten Version der Software aussähen.

(84) Der *Dienstleister* schreibt bei jedem *Release* **Releasenotizen** in der vereinbarten Sprache (Deutsch, Französisch oder Englisch) mit möglichst detaillierten Testanweisungen für die Tester.

(85) Der Auftraggeber entscheidet, welche Endbenutzer als **Tester** fungieren.

(86) Eine **Testphase** startet, wenn der Dienstleister die Releasenotizen veröffentlicht hat. Während der Testphase prüfen die Endbenutzer, ob die kommende Version funktioniert. (87) Wenn eine *Testphase* begonnen hat, führt der Dienstleister keine weiteren ungefragten Änderungen an der Instanz durch.

(88) Die *Testphase* dauert so lange wie nötig. Sie endet, indem die *Vorschau* in den *Produktionsbetrieb* geschaltet wird. Der Auftraggeber entscheidet, wann das ist.

(89) Nach der *Testphase* wird die nächste *Testphase* in einer neuen Instanz vorbereitet. Wenn die Vorbereitungen beendet sind und die nächste Testphase beginnen kann, veröffentlicht der *Dienstleister* die neuen Releasenotizen.

Abrechnungsarten

(90) Es gibt zwei prinzipiell verschiedene **Abrechnungsarten** : *pauschal* oder *per Stundenkredit*. Unabhängig der Abrechnungsart gelten folgende Prinzipien.

(91) Der Auftraggeber zahlt nicht für **Kundenbetreuung**. Dazu gehören das Aufsetzen von Angeboten, Auskunft über Machbarkeit oder zum Fortschreiten einer geplanten Arbeit und Antwort auf Rückfragen zu einem Dienstleistungsbericht.

(92) Der Auftraggeber zahlt nicht für die allgemeine Entwicklung des **Frameworks** und Änderungen in *technischer Dokumentation*. Wenn eine *Anfrage* am sinnvollsten zu implementieren ist durch eine Änderung im *Framework*, gilt sie deshalb nicht als allgemeine Entwicklung. Eventueller Mehraufwand für die Integrierung einer Anfrage ins Framework geht zu Lasten des Dienstleisters.

(93) Die Behebung von **Fehlern im Framework** ist kostenlos, wenn der Fehler keine Nebenwirkung einer *Anfrage* des Auftraggebers ist.



(94) Schriftliche Vorschläge zu einer **Verbesserung im Framework** oder der technischen Dokumentation werden kostenlos bearbeitet.

(95) Sollte bei einem Einsatz durch **mangelnde Vorsicht** des Dienstleisters der Produktionsbetrieb ausfallen oder nennenswert gestört sein, arbeitet der Dienstleister auf eigene Kosten, bis der Produktionsbetrieb wieder hergestellt ist.

(96) Arbeiten, die durch **fehlerhaft funktionierende Hardware** hervorgerufen wurden, sind immer durch den Auftraggeber zu zahlen.

(97) Für nennenswerte *technologisch bedingte Änderungen* und **neue Features im Framework**, bei denen der Auftraggeber wählen kann, ob er sie aktiviert oder nicht, kann der Dienstleister einen Beitrag zu seinen Entwicklungskosten verlangen, bevor er das neue Feature in die Instanz des Auftraggebers aufnimmt.

Abrechnung per Stundenkredit

Bei **Abrechnung per Stundenkredit** gelten folgende besonderen Regeln.

(98) Der Auftraggeber zahlt für alle **Arbeiten, die er anfragt**. Dazu gehört jede Arbeit des Dienstleisters, die über *Kundenbetreuung* hinaus geht und zur Beantwortung einer *Anfrage* nötig ist, zum Beispiel Support, Rückfragen zur Planung, Diagnose, *Analyse*, Änderungen in Quellcode und Dokumentation, Planung und Durchführung von *Einsätzen* auf dem Server sowie das Aufsetzen von Berichten, Tests und Dokumentation über diese Arbeit.

(99) Der *Dienstleister* schreibt monatliche **Dienstleistungsberichte** über die geleisteten Arbeiten und den Stand des Projekts. Der Auftraggeber kann jederzeit einen Zwischendienstleistungsbericht anfragen.

(100) Der *Dienstleister* führt für den Auftraggeber ein **Stundenkontingent**, d.h. ein Konto, dessen Stand durch Ankauf von **Stundenkredit** erhöht und durch einen *Dienstleistungsbericht* verringert wird.

(101) Zahlbare Arbeiten des Dienstleisters werden durch einen Dienstleistungsbericht vom Stundenkontingent abgebucht.

(102) Sollte ein angekaufter Stundenkredit innerhalb eines Jahres nicht aufgebraucht werden, verfällt er prinzipiell, kann aber nach Einverständnis des *Dienstleisters* um maximal ein weiteres Jahr verlängert werden.

(103) Bei Projekten ohne *verbindliche Deadline* gilt: Wenn der Stundenkredit **auf unter 10 Stunden** fällt, schreibt der Dienstleister dem Auftraggeber ein Angebot zum Ankauf zusätzlichen Stundenkredits, um das Stundenkontingent zu füllen. Wenn der Stundenkredit **auf unter 1 Stunde** fällt, arbeitet der Dienstleister ausschließlich an kostenlosen *Anfragen*, bis eine Einigung über die weitere Finanzierung getroffen ist.

(104) Reklamationen an einem Dienstleistungsbericht teilt der Auftraggeber dem Dienstleister spätestens vor dem nächsten Monatsende schriftlich mit.

(105) Der Auftraggeber zahlt für **Support** auch dann, wenn dieser nicht zu einer Änderung oder Lösung der Anfrage führt.

Pauschale Abrechnung

(106) **Pauschale Abrechnung** ist eine andere Form der Zusammenarbeit, bei der eine Instanz zu einem individuell vereinbarten pauschalen **Jahresbeitrag** gewartet und/oder weiterentwickelt wird. Es gibt kein Stundenkontingent und keine Dienstleistungsberichte. Bei pauschaler Abrechnung gelten folgende besondere Regeln.

(107) Alle Arbeiten des Dienstleisters für **Support, Analyse, kleine Optimierungen und technologisch begründete Anpassungen** bestehender Funktionen sowie alle **Einsätze auf dem Server** sind im Jahresbeitrag enthalten. Pro Jahr wird mindestens ein *Release* gemacht, außer wenn der Auftraggeber keines wünscht.

(108) Für **Weiterentwicklungsprojekte** findet Analyse und Entwicklungsarbeit gemeinsam mit der Kontaktperson in der *Vorschau* statt. Der Dienstleister schreibt ein **Angebot** spätestens bevor die Arbeiten in den Produktionsbetrieb gehen.



(109) Für das **Anlernen einer neuen Kontaktperson** ist ein Pauschalpreis laut Angebot zu zahlen. Eine einmal angelernte Kontaktperson kann für andere Auftraggeber ohne erneutes Anlernen arbeiten.

(110) In Ausnahmefällen wie z.B. Noteinsätzen durch *fehlerhaft funktionierende Hardware* kann eine zusätzliche Abrechnung im Nachhinein erfolgen.

Engpässe und Deadlines

(111) Alle *Anfragen* des Auftraggebers werden unverzüglich bearbeitet unter Berücksichtigung der Dringlichkeit des Problems und der verfügbaren Ressourcen.

(112) Wir sind uns dessen bewusst, dass der *Dienstleister* begrenzte Ressourcen hat. Bei Engpässen geben alle Beteiligten ihr Bestes, um eine Lösung zu finden.

(113) Auf Wunsch des Auftraggebers kann der Dienstleister **verbindliche Deadlines** für konkrete Projekte nennen und verpflichtet sich dadurch zur Einhaltung des zugesagten Termins. Die genauen Bedingungen dazu sind im betreffenden *Angebot* vereinbart.

(114) Kostenschätzungen sind nur verbindlich, wenn sie in einem *Angebot* ausdrücklich als solche bezeichnet werden. Verbindliche Kostenschätzungen dürfen bei triftiger Rechtfertigung um maximal 15% erhöht werden.

Datenschutz

(115) **Lokale Konfigurationsdateien** und **Datenbankinhalt** der Lino-Instanz sind Eigentum des Auftraggebers und werden vom Dienstleister vertraulich behandelt. Dies gilt auch nach Beendigung der Zusammenarbeit. Dies gilt nicht für Daten, die unabhängig von unserer Zusammenarbeit an anderen Stellen bekannt sind.

(116) Die allgemeinen Datenschutzbestimmungen des Dienstleisters laut DSGVO sind unter <https://saffre-rumma.net/dl/Datenschutz.pdf> abrufbar.

(117) Der Dienstleister unterliegt bei allen Arbeiten **auch** den internen Datenschutzbestimmungen des Auftraggebers. Der Auftraggeber ist dafür verantwortlich, den Dienstleister über diese zu informieren.

(118) Sowohl Auftraggeber als auch Dienstleister können interne oder externe **Mitarbeiter** oder **Partner** mit einer oder mehrerer ihrer Aufgaben beauftragen und sind dafür verantwortlich, diese vertraglich an ihre Datenschutzbestimmungen zu **binden**.

(119) **Vornamen von Einzelpersonen** gelten **nicht** als vertraulich und dürfen in Quellcode und Dokumentation veröffentlicht werden.

Sonstiges

(120) Die Zusammenarbeit ist automatisch **beendet**, wenn das *Stundenkontingent* des Auftraggebers auf Null fällt oder der *Jahresbeitrag* abläuft und keine Einigung über die Fortsetzung der Zusammenarbeit gefunden wird.

(121) Dieses Dokument ist öffentlich und gilt für alle Aufträge, die sich darauf beziehen. Es wurde zuletzt bearbeitet am **Mittwoch, 17. April 2019, a** um **16:38** Uhr. Eine eventuell aktuellere Version steht unter <https://saffre-rumma.net/dl/Kooperationsrichtlinien.pdf>

(122) Der *Dienstleister* kann dieses Dokument jederzeit anpassen und informiert den *Auftraggeber* über wichtige Änderungen. Es gilt die zum Zeitpunkt des Angebotes veröffentlichte Version.

(123) Eine E-Mail gilt als **schriftliche Mitteilung**, wenn ihr Eingang vom Empfänger bestätigt wurde.

