

(1) Dieses Dokument präzisiert die allgemeinen Richtlinien für unsere Zusammenarbeit, deren spezifische Optionen im *Entwicklungs-* oder *Wartungsauftrag* vereinbart werden, der sich auf dieses Dokument bezieht.

(2) Kooperationspartner sind einerseits *Rumma & Ko OÜ* als **Dienstleister** und andererseits der **Auftraggeber** als Nutzer eines *Gesamtsystems*, das die Endbenutzer als „ihren Lino“ verstehen.

(3) Dieses Dokument ist öffentlich und gilt für alle Aufträge, die sich darauf beziehen. Die jeweils aktuelle Version steht unter <https://saffre-rumma.net/dl/Kooperationsrichtlinien.pdf>

(4) Der *Dienstleister* kann dieses Dokument jederzeit anpassen und informiert dann den *Auftraggeber* über die Änderung. Wenn der *Auftraggeber* innerhalb von zwei Wochen nach Mitteilung einer Änderung keinen Einspruch erhebt, gilt die Änderung als akzeptiert. Solange eine Änderung in diesem Dokument nicht mitgeteilt und akzeptiert wurde, gilt die am Datum der Auftragserteilung gültige Version.

Systemkomponenten

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

(6) Die **Hardware** besteht aus dem **Server**, d.h. einem eigenständigen (normalerweise virtuellen) Rechner innerhalb oder außerhalb der Gebäude des Auftraggebers, den **Clients**, d.h. den Rechnern der Endbenutzer, sowie dem **Netzwerk**, das die Verbindung der beiden ermöglicht.

(7) Unter **Anwendungssoftware** verstehen wir die im *Entwicklungs-* oder *Wartungsauftrag* benannte Lino-Anwendung, die der *Dienstleister* entwickelt und als freie Software öffentlich zur Verfügung stellt.

(8) Unter **Systemsoftware** verstehen wir alle Software oberhalb 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.

(9) Eine **Anlage** („Site“) ist, wenn eine konkrete Version der *Anwendungssoftware* auf einem konkreten *Server* läuft. Sie besteht aus einem Domainname, hinter dem ein Webserver läuft, der Datenbank sowie den lokalen Einstellungen und Konfigurationsdateien zur Einbindung der *Anwendungssoftware* in die *Systemsoftware*.

(10) Auf einem *Server* können eine oder mehrere *Anlagen* laufen.

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

(12) Für die Einrichtung und Wartung der *Anlagen* auf dem *Server* ist der *Dienstleister* verantwortlich.

Lebenslauf eines Projektes

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

(14) Wir unterscheiden zwischen *Neuentwicklungsprojekten*, Vorproduktionsbetrieb und

Produktionsbetrieb. Im *Produktionsbetrieb* können gegebenenfalls *Weiterentwicklungsprojekte* gestartet werden.

(15) Ziel eines **Neuentwicklungsprojektes** ist die Inbetriebnahme einer neuen *Anlage*, auf der die *Software* läuft und benutzt werden kann. Ein *Neuentwicklungsprojekt* wird durch einen **Entwicklungsauftrag** gestartet.

(16) Je nach Umständen kann als optionaler Zwischenschritt ein **Vorproduktionsbetrieb** eingefügt werden. Dabei wird die *Anlage* zu einem individuell vereinbarten pauschalen **Jahresbeitrag** gewartet und weiterentwickelt.

(17) Ziel des **Produktionsbetriebs** ist die Aufrechterhaltung eines zuverlässigen Funktionierens einer *Anlage*.

Schritt	Rechnung	Endet durch
Interview und Vorgespräche	Kostenlos	Entwicklungsauftrag
Interne Entwicklungsphase zur Vorbereitung des Prototyps.	Festpreis für <i>Einrichtung des Prototypen (zahlbar bei Abnahme)</i>	Abnahme des Prototypen
Gemeinsame Entwicklungsphase	<i>Stundenkontingent</i>	<i>Einigung über Jahresbeitrag</i>
Vorproduktionsbetrieb	Jahresbeitrag	Unterzeichnung <i>Wartungsauftrag</i>
Produktionsbetrieb	Stundenkontingent	Ende des Wartungsauftrags

Neuentwicklungsprojekte

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

(19) 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.

(20) Der Dienstleister setzt einen **Entwicklungsauftrag** auf, in dem die beim Interview besprochenen Funktionen der zu entwickelnden Anwendung beschrieben sind sowie eine Schätzung der Entwicklungskosten.

(21) Wenn der Auftraggeber den *Entwicklungsauftrag bestätigt hat*, beginnt der Dienstleister mit dem **Einrichten des Prototypen**. Ein **Prototyp** ist eine *Anlage* 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.

(22) Das Einrichten des Prototypen wird auch als **interne Entwicklungsphase** bezeichnet. Gegen Ende dieser Phase präsentiert der Auftraggeber den Prototypen.

(23) Mit der **Abnahme des Prototypen** bestätigt der Auftraggeber, dass die **gemeinsame Entwicklungsphase** beginnen kann. Der *Dienstleister* schreibt eine Rechnung über Einrichtung des Prototypen und Ankauf von Stundenkredit für die Weiterentwicklung.

(24) In der **gemeinsamen Entwicklungsphase** wiederholen sich die folgenden Etappen zyklisch: *Baustellenbesichtigung* mit dem Koordinator, Bearbeitung der festgestellten Probleme, Entwicklung und Installation einer neuen Version und Planung der nächsten *Baustellenbesichtigung*. Wir arbeiten während dieser Phase auf dem *Prototypen*, der sich

ständig verändert und weiterentwickelt.

(25) Gegen Ende der gemeinsamen Entwicklungsphase ist die Anwendung so weit fertig, dass die Datenbank bei Aktualisierungen nicht mehr gelöscht sondern migriert wird. In dieser Phase werden die fiktiven Testdaten gelöscht, die Konfigurationsdaten angepasst, eventuell werden Daten importiert aus bestehenden Quellen. Der *Koordinator* organisiert Schulung der Endbenutzer und Erfassung der Stammdaten. Probleme, die sich bei der täglichen Arbeit ergeben, werden erfasst, analysiert und behoben.

(26) Mit der **Abnahme der Anlage** bestätigt der *Auftraggeber*, dass die Anlage in den Vorproduktions- oder Produktionsbetrieb wechseln kann.

Produktionsbetrieb

(27) Als Anlage im **Produktionsbetrieb** bezeichnen wir eine Anlage, für die ein **Wartungsauftrag** besteht.

(28) Im *Wartungsauftrag* übernimmt der *Dienstleister* die technische Verantwortung für das zuverlässige Funktionieren der *Anlage*.

(29) Wartungsaufträge können formlos verlängert werden, indem der *Dienstleister* eine Rechnung für die nächste Wartungsperiode ausstellt. Mit der Zahlung der Rechnung erklärt der *Auftraggeber* sein Einverständnis zur Verlängerung.

(30) Rechnungen für Wartungsaufträge dürfen nach schriftlicher Absprache auch stückweise bezahlt werden, also z.B. $\frac{1}{4}$ der Gesamtsumme alle drei Monate. Wobei der *Auftraggeber* dann selbst dafür verantwortlich ist, dass jede einzelne Teilzahlung rechtzeitig ausgeführt wird.

Weiterentwicklungsprojekte

(31) Ein **Weiterentwicklungsprojekt** ist ein *Entwicklungsprojekt*, das parallel mit dem Wartungsauftrag auf einer Anlage im Produktionsbetrieb läuft.

(32) Die Kosten für *Analyse*, *Planung* und *Implementierung* eines Weiterentwicklungsprojekts trägt der Auftraggeber.

Aufgaben des Auftraggebers

(33) Der *Auftraggeber* muss Personal zur Verfügung stellen oder beauftragen, um folgende Aufgaben zu übernehmen:

(34) Ein **Koordinator** zentralisiert den Informationsfluss zwischen *Entwickler* und *Endbenutzern*, klärt Rückfragen des Dienstleisters zu Anfragen des Auftraggebers, verwaltet Zugriffsrechte der Endbenutzer und organisiert für diese je nach Bedarf Support, Dokumentation und Schulung.

(35) Als **Endbenutzer** bezeichnen wir alle Personen, denen der Auftraggeber Zugriff auf die Anlage gewährt. Der Auftraggeber ist verantwortlich für das Vergeben und Zurücknehmen von Zugriffsrechten.

(36) Der **Systemverwalter** ist eine vom *Auftraggeber* zur Verfügung gestellte oder beauftragte Person, die für Bereitstellung und Wartung der *Hardware* sowie Backups aller lokalen Daten verantwortlich ist.

(37) Ein einziger Mitarbeiter des Auftraggebers kann mehrere Rollen zugleich ausüben.

(38) Alle Rollen können bei Bedarf durch einen externen Partner ausgeübt werden, den der Auftraggeber getrennt beauftragt.

(39) Der Auftraggeber meldet eventuelle Probleme dem Dienstleister, beantwortet Rückfragen des Dienstleisters und liefert ihm alle nötigen Informationen, die er zur Lösungsfin-



dung braucht.

(40) Der Auftraggeber ist mindestens per E-Mail erreichbar, idealerweise auch telefonisch, über *instant messaging* oder Internet-Telefon

(41) Der *Systemverwalter* teilt dem *Dienstleister* alle Informationen über Änderungen an der Hardware mit.

Aufgaben des Dienstleisters

(42) Der Dienstleister ist prinzipiell werktags von 8 bis 18 Uhr telefonisch **erreichbar**. Garantierte Bereitschaft nur nach vorheriger Absprache.

(43) Der Dienstleister beantwortet jede Problemmeldung des Auftraggebers innerhalb einer entsprechend der Dringlichkeit des Problems vertretbaren Zeit.

(44) Der Dienstleister führt alle Einsätze auf dem Server gewissenhaft durch und ohne den Produktionsbetrieb zu gefährden. Sollt dabei durch mangelnde Vorsicht der Produktionsbetrieb lahmgelegt werden, arbeitet der Dienstleister auf eigene Kosten, bis der Produktionsbetrieb wieder hergestellt ist.

Dienstleistungen und Stundenkontingente

(45) Der *Dienstleister* schreibt monatliche **Dienstleistungsberichte** über die geleisteten Arbeiten. Je nach Aktivität des Projekts können Dienstleistungsberichte auch häufiger oder seltener verschickt werden.

(46) 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.

(47) 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.

(48) Wenn der Saldo des Stundenkontingents **auf unter 10 Stunden** fällt, schreibt der Dienstleister dem Auftraggeber ein Angebot zum Ankauf zusätzlichen Stundenkredits, um das Stundenkontingent zu füllen.

(49) Wenn der Stundenkredit **auf unter 1 Stunde** fällt, arbeitet der Dienstleister ausschließlich an kostenlosen Anfragen.

(50) Wenn der Auftraggeber nicht einverstanden ist, für eine in einem Dienstleistungsbericht erwähnte Arbeit zu zahlen, teilt er dies dem Dienstleister spätestens zwei Wochen nach Erhalt des Dienstleistungsberichts schriftlich mit.

Zahlbar oder nicht?

(51) Der Auftraggeber zahlt für alle **Änderungen, die er anfragt**. Dabei werden alle Arbeiten des Dienstleisters vom Stundenkontingent abgebucht. Dazu zählen zum Beispiel Support, Rückfragen des Dienstleisters zur Planung, Diagnose und *Analyse* von Problemen, Änderungen in Quellcode und Dokumentation, Planung und Durchführung von *Einsätzen* auf dem Server sowie das Aufsetzen von Berichten, Tests und Dokumentation über diese Arbeit.

(52) Der Auftraggeber zahlt prinzipiell für Beantwortung von Fragen zur Benutzung der Software.

(53) Der Auftraggeber zahlt prinzipiell nicht für **Kundenbetreuung**. Dazu gehören das Erstellen von Angeboten, Auskunft zum Fortschreiten einer geplanten Arbeit und Antwort auf Rückfragen zu einem Dienstleistungsbericht.



(54) Die Behebung von **Fehlern in anwendungsübergreifenden** Funktionen ist kostenlos, wenn der Programmfehler keine Nebenwirkung einer angefragten Änderung sind. Auftretende Programmfehler muss der Auftraggeber dem Dienstleister schriftlich melden. Der Dienstleister entscheidet, ob und wie der Programmfehler kostenlos behoben wird. Änderungen in Funktionen, die theoretisch für andere nützlich sind, aber *de facto* bisher nur vom Auftraggeber benutzt werden, gelten prinzipiell als nicht anwendungsübergreifend.

(55) Schriftliche Vorschläge zur **Verbesserung der Dokumentation** werden kostenlos bearbeitet. Der Dienstleister entscheidet, wie und mit welcher Dringlichkeit der Vorschlag verwendet wird.

(56) Vorschläge zu allgemeinen **Verbesserungen im Framework** müssen schriftlich eingereicht werden. Sie werden nach Ermessen des Dienstleisters berücksichtigt und kostenlos programmiert.

Allgemeine Entwicklung des Frameworks

(57) Der Auftraggeber zahlt nicht für die **allgemeine Entwicklung des Frameworks**. Dazu gehören z.B. Verbesserungen der Entwicklerschnittstelle, Entwicklung neuer Benutzerschnittstellen, Anpassungen der Methoden für Qualitätssicherung, Deployment und Dokumentation, sowie funktionale Änderungen, für die von allgemeinem Interesse.

(58) Der *Dienstleister* veröffentlicht und wartet nach eigenem Ermessen **technische Dokumentation** über die *Anwendungssoftware* in englischer Sprache. Die *technische Dokumentation* hat als Zielgruppe andere Entwickler und ist ein integraler Bestandteil der *Testsuite*.

(59) Der *Dienstleister* veröffentlicht und wartet nach eigenem Ermessen ein Online-Handbuch für Endbenutzer.

Aktualisierungen einer Anlage

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

(61) 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.

(62) Bei einer **Aktualisierung** wird neuer Quellcode der *Anwendungssoftware* auf die *Anlage* geladen und aktiviert.

(63) Ein **Release** ist eine *Aktualisierung*, für die eine eigene Versionsnummer und eigene Release-Notizen erstellt und gewartet werden.

(64) Eine **einfache Aktualisierung** kann durchgeführt werden, um Probleme nach einem *Release* operativ zu beheben. Dabei wird keine neue Versionsnummer vergeben und keine neue Testphase durchgeführt, und die Release-Notizen der laufenden Version werden lediglich aktualisiert.

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

(66) Lino enthält eine umfangreiche **Testsuite** zum automatisierten Testen vieler Funktionen. Die Testsuite wird automatisiert vor jedem Release durchgeführt.

(67) Die *Testsuite* kann keine absolute Sicherheit vor Problemen nach einem *Einsatz* geben.

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



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

(70) Wir bieten zwei unterschiedliche Verfahren, um mit dieser Art von Problemen umzugehen. Im Testverfahren **Flex** sind **Flexibilität** und **Aufwand** prioritär, im Aktualisierungsverfahren **Safe** dagegen eher **Planbarkeit** und **Dokumentation**.

Einfaches Testverfahren

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

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

(73) Am nächsten Arbeitstag vereinbaren die *Endbenutzer* vor dem Einsatz eine **Testphase**, die eine oder mehrere Stunden dauern kann.

(74) In dieser Zeit testen ausgewählte Benutzer ihren neuen Lino im laufenden Produktionsbetrieb.

(75) Sie sind dabei auf *Regressionen* und Überraschungen gefasst, die sie dann melden müssen und die dann prioritär bearbeitet werden.

(76) Alle sind sich dessen bewusst, dass der Einsatz rückgängig gemacht und die Anlage auf den Zustand vom Vortag zurück gesetzt werden könnte, falls ein Release-Blocker gemeldet wird.

Erweitertes Testverfahren

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

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

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

(80) Die Testphase dauert so lange wie nötig.

(81) Auf Wunsch des Auftraggebers schreibt und wartet der Dienstleister detaillierte Testanweisungen für die Tester.

(82) Wenn das Testen einer neuen Version begonnen hat, kommen keine neuen Features mehr hinzu.

(83) Ohne Benutzerdokumentation gibt es keine formalen Testanweisungen, sondern die Benutzer testen intuitiv bzw. nach selbst erstellten Checklisten.

(84) Das Umschalten der *Vorschau* in den *Produktionsbetrieb* findet statt, wenn alle Probleme in der *Vorschau* behoben wurden.

(85) Der *Dienstleister* schreibt bei jedem *Release* **Release-Notizen** in der vereinbarten Sprache (Deutsch, Französisch oder Englisch).

Autorenrecht und Datenschutz

(86) Alle Mitarbeiter des Dienstleisters, die Zugriff auf den Server oder Kontakt mit Endbenutzern haben, unterliegen den Datenschutzbestimmungen, die der Auftraggeber an seine eigenen Mitarbeiter stellt.



- (87) Der Auftraggeber ist dafür verantwortlich, den Dienstleister über seine Datenschutzbestimmungen zu informieren.
- (88) Der Dienstleister ist dafür verantwortlich, seine Mitarbeiter an die Datenschutzbestimmungen des Auftraggebers zu binden.
- (89) Der Dienstleister veröffentlicht den im Rahmen der Zusammenarbeit erstellen Quellcode und Dokumentation der Software unter einer freien Lizenz.
- (90) **Betriebsabläufe** und **Vornamen von Mitarbeitern** gelten nicht als vertraulich und dürfen in Quellcode und Dokumentation veröffentlicht werden.
- (91) **Lokale** Konfigurationsdateien auf dem Server gelten als vertraulich und dürfen nicht veröffentlicht werden.

Haftungsausschluss

- (92) Der Dienstleister garantiert nicht die Fehlerfreiheit der Software, sondern seine Hilfestellung bei Problemen.
- (93) Sollte jemand aufgrund eines Fehlers oder Ausfalls der *Anlage* zu Schaden kommen, kann der Dienstleister dafür nicht rechtlich haftbar gemacht werden.

Sonstiges

- (94) Vereinbarte Preise werden jedes Jahr entsprechend des belgischen Gesundheitsindex angepasst.
- (95) Eine E-Mail gilt als **schriftliche Mitteilung**, wenn ihr Eingang vom Empfänger bestätigt wurde.

Glossar

- (96) Als **Analyse** bezeichnen wir die Diagnose und Erfassung eines Problems, Rückfragen des Dienstleisters zur Präzisierung einer Änderungsanfrage, die Suche nach Ursachen und das Erarbeiten von Lösungsvorschlägen.
- (97) Ein **Angebot** ist ein Dokument, in dem der Dienstleister einen Vorschlag zur Lösung und Kostenplanung eines zuvor gemeldeten und analysierten Problems macht.
- (98) Eine **Bestellung** ist ein Dokument, in dem der Auftraggeber ein vom Dienstleister aufgesetztes *Angebot* bestätigt.
- (99) Als **kleine Optimierungen** bezeichnen wir Änderungen der Software, für die sich ein eigenes *Angebot* nicht lohnt.
- (100) Als **Qualitätssicherung** bezeichnen wir das Schreiben und Pflegen von automatisierten Testverfahren.
- (101) Als **Deployment** bezeichnen wir die Veröffentlichung der Software mit dem Ziel, sie auch für andere Entwickler verfügbar zu machen.
- (102) Als **Implementierung** bezeichnen wir die Änderungen im Quellcode, der Dokumentation und der lokalen Konfiguration, die Nachanalyse, und die Durchführung der Einsätze, die zur Lösung eines Problems nötig sind.

