Kooperationsrichtlinien

(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 OÜ Rumma & Ko als **Dienstleister** und 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 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* sowie Backups aller lokalen Daten ist der *Auftraggeber* verantwortlich.

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

Arten der Zusammenarbeit

(13) Wir unterscheiden zwei grundlegende Arten der Zusammenarbeit : *Neuentwicklungsprojekte* und *Produktionsbetrieb*. *Im Produktionsbetrieb können gegebenenfalls Weiterentwicklungsprojekte gestartet werden.*

(14) Ein *Neuentwicklungsprojekt* wird durch einen **Entwicklungsauftrag** gestartet und endet, wenn die Anlage in den *Produktionsbetrieb* wechselt.

(15) Ziel eines *Neuentwicklungsprojektes* ist die Inbetriebnahme einer neuen *Anlage*, auf

der die *Software* läuft und benutzt werden kann.

(16) Ziel eines *Wartungsauftrags* ist die Aufrechterhaltung eines zuverlässigen *Produktionsbetriebs* einer *Anlage*.

Neuentwicklungsprojekte

(17) Bei einem *Neuentwicklungsprojekt* durchläuft unsere Zusammenarbeit üblicherweise folgende Phasen und Entwicklungsschritte:

(18) Der Dienstleister trifft sich mit dem Auftraggeber zu einem **Interview**. Ziel des Interviews ist es, eine Übersicht über die Funktionen.

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

(20) Der *Auftraggeber* unterschreibt den *Entwicklungsauftrag*.

(21) Zunächst eine **interne Entwicklungsphase** mit Analyse, gelegentlichen Fragen an die Kontaktperson, Programmierung der ersten Arbeitsversion und Einrichten eines *Prototypen*.

(22) Die *Vorbereitungsphase* endet, wenn der *Dienstleister* den *Prototypen* als zufriedenstellend erachtet. Er teilt dem dem Auftraggeber die Zugriffsdaten zum *Prototypen* mit und schreibt eine Rechnung über Einrichtung des Prototypen und Ankauf von Stundenkredit für die Weiterentwicklung.

(23) Nach Zahlung der Rechnung startet die **gemeinsame Entwicklungsphase**, in der sich die folgenden Etappen zyklisch wiederholen : *Baustellenbesichtigung* mit Kontaktperson, Bearbeitung der festgestellten Probleme, Entwicklung und Installation einer neue Version und Planung der nächsten *Baustellenbesichtigung*. Wir arbeiten während dieser Phase auf dem *Prototypen*, der sich ständig verändert und weiterentwickelt.

(24) Am Ende der gemeinsamen Entwicklungsphase ist die Anwendung so weit fertig, dass der **Vorproduktionsbetrieb** beginnen kann. Die fiktiven Testdaten werden gelöscht, die Konfigurationsdaten angepasst. Eventuell werden Daten importiert aus bestehenden Quellen. Die Datenbank wird ab jetzt bei Aktualisierungen nicht mehr gelöscht, sondern migriert. Der *Prototyp* wird zu einer produktiven Anlage. Die *Kontaktperson* organisiert Schulung der Endbenutzer und Erfassung der Stammdaten. Probleme, die sich bei der täglichen Arbeit ergeben, werden erfasst, analysiert und behoben.

(25) Das Entwicklungsprojekt gilt als beendet, wenn *Dienstleister* und *Auftraggeber* sich einig sind, dass das Ziel des Entwicklungsprojekts erreicht ist und man nun einen Wartungsauftrag starten kann.

(26) Die Dauer jeder Phase hängt von vereinbarten Terminen, Verfügbarkeit der *Kontaktperson* und den sich ergebenden Fragestellungen ab.

Produktionsbetrieb

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

(28) Ein *Wartungsauftrag* gilt für eine festgelegte Periode, üblicherweise ein Jahr.

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

(30) 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.

(31) Eventuelle Absicht zur Nicht-Verlängerung eines Wartungsauftrags wird dem Partner

unverzüglich mitgeteilt.

(32) Vereinbarte Preise werden jedes Jahr entsprechend des belgischen Gesundheitsindexes angepasst.

(33) Dienstleister und Auftraggeber können jederzeit einvernehmlich die Höhe des Jahresbeitrags anpassen.

(34) Rechnungen für Wartungsaufträge dürfen nach schriftlicher Absprache auch stückweise bezahlt werden, also z.B. ¼ der Gesamtsumme alle drei Monate. Wobei der *Auftraggeber* dann selbst dafür verantwortlich ist, dass jede einzelne Teilzahlung rechtzeitig ausgeführt wird.

Weiterentwicklungsprojekte

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

(36) Für Weiterentwicklungsprojekte erstellt der Dienstleister eine Aufwandsprognose, die der Auftraggeber zum Ankauf weiterer Kontingentstunden berücksichtigt.

Übersicht Ablauf eines Lino-Projekts

Schritt	***Endet durch***
Interview und Vorgespräche.	Dienstleister schreibt Entwicklungsauftrag, den der Auftraggeber unterschreibt. Dienstleister schreibt Rechnung für Einrichten des *Prototypen*, die der Auftraggeber durch eine Anzahlung bestätigt.
Interne Entwicklungsphase zur Vorbereitung des Prototyps.	Dienstleister schreibt Rechnung für Ankauf von K*ontingentstunden*.
Gemeinsame Entwicklungsphase	Beschluss des Auftraggebers*, in den Vorproduktionsbetrieb zu wechseln*
Vorproduktionsbetrieb	Abnahme. Auftraggeber und Dienstleister sind sich einig, dass das Enwicjklungsziel erreicht ist. Einigung *Wartungskontingent* und *Wartungsauftrag*
Produktionsbetrieb	Ende des Wartungsauftrags

Rollen

(37) Der *Auftraggeber* designiert und stellt eine **Kontaktperson** zur Verfügung, die den Informationsfluss zwischen *Dienstleister* und *Endbenutzern* zentralisiert, Zugriffsrechte der Endbenutzer verwaltet und für diese je nach Bedarf Support, Dokumentation und Schulung organisiert.

(38) Der *Auftraggeber* designiert und stellt einen **Systemverwalter** zur Verfügung, der für die Bereitstellung und Wartung der *Hardware* sowie Backups aller lokalen Daten verantwortlich ist.

(39) 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.

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

(41) Alle Rollen können bei Bedarf durch einen externen Partner ausgeübt werden, den

der Auftraggeber getrennt beauftragt.

Kommunikation

(42) Der Auftraggeber meldet eventuelle Probleme dem Dienstleister, beantwortet Rückfragen des Dienstleisters und liefert ihm alle nötigen Informationen, die er zur Lösungsfindung braucht.

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

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

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

(46) Nach vorheriger Absprache garantiert der Dienstleister **erhöhte Bereitschaft**.

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

(48) Der *Dienstleister* teilt dem *Systemverwalter* alle Informationen mit, die dieser wünscht, um notfalls auch ohne Hilfe des Dienstleisters Einsätze auf der *Anlage* durchzuführen.

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

Dienstleistungen und Stundenkontingente

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

(51) 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.

(52) Beim Ankauf von Stundenkredit wird eine **vorgesehene Periode** festgelegt. Sollte der Kredit innerhalb dieser Periode nicht aufgebraucht werden, verfällt er prinzipiell, kann aber nach Einverständnis des *Dienstleisters* um maximal eine weitere Periode verlängert werden.

(53) 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.

(54) Bei **negativem Stundenkontingent** arbeitet der Dienstleister ausschließlich an Anfragen, für die eine schriftliche Einigung mit dem Auftraggeber besteht über einen **Tilgungsplan** für die Schuld.

(55) Wir unterscheiden zwischen *Entwicklungsarbeit* und *Kundenbetreuung*. Prinzipiell ist Entwicklungsarbeit zahlbar und Kundenbetreuung kostenlos.

(56) Als **Entwicklungsarbeit** zählen Antworten auf Fragen zur Benutzung der Software, Diagnose und *Analyse* von Problemen, Programmierung, das Durchführen von *Einsätzen* auf dem Server sowie das Aufsetzen von Berichten, Tests und Dokumentation über diese Arbeit.

(57) Als **Kundenbetreuung** zählen Gespräche mit dem Auftraggeber zu einem Angebot, einer Rechnung oder einem Dienstleistungsbericht.

(58) Wenn der Auftraggeber nicht einverstanden ist, für eine in einem Dienstleistungsbe-

richt erwähnte Arbeit zu zahlen, teilt er dies dem Dienstleister spätestens vor Ankauf des nächsten Stundenkredits schriftlich mit. Der *Dienstleister* gibt sein Einverständnis in Form eines Eintrags ins *Stundenkontingent*.

Jahresbeitrag

(59) Bei einer Anlage im *Produktionsbetrieb* zahlt der *Auftraggeber* zusätzlich zum Ankauf der vereinbarten Anzahl von Kreditstunden einen **Jahresbeitrag** zur *allgemeinen Entwicklung des Frameworks.*

(60) Als **allgemeine Entwicklung des Frameworks** bezeichnen wir Arbeiten des Dienstleisters, die nicht einem einzelnen Auftraggeber zugeordnet werden. Dazu gehören z.B. Verbesserungen der Entwicklerschnittstelle, Entwicklung neuer Benutzerschnittstellen, Anpassungen der Methoden für Qualitätssicherung, Deployment und Dokumentation, sowie funktionale Änderungen von allgemeinem Interesse.

Aktualisierungen einer Anlage

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

(62) 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.

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

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

(65) 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.

(66) Bei jeder *Aktualisierung* ist der *Dienstleister* verantwortlich für eine korrekte **Datenmigration** und sorgt dafür, dass der Einsatz falls nötig rückgängig gemacht werden kann.

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

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

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

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

(71) 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 (Flex)

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

(73) Bei diesem Verfahren werden neue Versionen nach internem Test durch den Dienst-

leister über Nacht aufgespielt.

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

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

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

(77) 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 (Safe)

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

(79) 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.

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

(81) Die Testphase dauert so lange wie nötig.

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

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

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

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

Dokumentation

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

(87) Für jede *Anlage* im *Produktionsbetrieb* veröffentlicht und wartet der *Dienstleister* **technische Dokumentation** über die *Anwendungssoftware* in englischer Sprache.

(88) Die *technische Dokumentation* umfasst standardmäßig eine **Spezifikation** aller Funktionen, deren Zielgruppe andere Entwickler sind und die ein integraler Bestandteil der *Testsuite* ist.

(89) Auf Wunsch des Auftraggebers schreibt der Dienstleister ein **Benutzerhandbuch** und passt es bei Änderungen der Software an.

(90) Umfang und Qualität aller Dokumentation richten sich nach den Bedürfnissen des Auftraggebers. Der Dienstleister arbeitet Verbesserungsvorschläge des Auftraggebers in das Handbuch ein.

(91) Das *Benutzerhandbuch* ist online verfügbar und kann optional als pdf-Datei heruntergeladen werden.

(92) Der Dienstleister veröffentlicht das *Benutzerhandbuch* sowie den zur Erstellung nötigen Quellcode unter einer freien Lizenz.

Autorenrecht und Datenschutz

(93) 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.

(94) Der Auftraggeber ist dafür verantwortlich, den Dienstleister über seine Datenschutzbestimmungen zu informieren.

(95) Der Dienstleister ist dafür verantwortlich, seine Mitarbeiter an die Datenschutzbestimmungen des Auftraggebers zu binden.

(96) Der Dienstleister ist verpflichtet, den im Rahmen der Zusammenarbeit erstellen Quellcode und Dokumentation der Software unter einer freien Lizenz zu veröffentlichen entsprechend der Anforderungen der Initiative *Public Money? Public Code!* (https://publiccode.eu/de/openletter/)

(97) **Betriebsabläufe** und **Vornamen** von *Kontaktpersonen* gelten nicht als vertraulich und dürfen in Quellcode und Dokumentation veröffentlicht werden.

(98) **Lokale** Konfigurationsdateien auf dem Server gelten als vertraulich und dürfen nicht veröffentlicht werden.

Haftungsausschluss

(99) Der Dienstleister garantiert nicht die Fehlerfreiheit der Software, sondern seine Hilfestellung bei Problemen.

(100) Sollte jemand aufgrund eines Fehlers oder Ausfalls der *Anlage* zu Schaden kommen, kann der Dienstleister dafür nicht rechtlich haftbar gemacht werden.

Glossar

(101) Als **Prototyp** bezeichnen wir eine *Anlage* mit fiktiven Demo-Daten, die als Arbeitsgrundlage für die *aktive Entwicklungsphase* dient. Ein *Prototyp* kann entweder auf einem vom Dienstleister betriebenem *Server* installiert sein, oder direkt auf dem *Server*, auf dem die *Software* später im *Produktionsbetrieb* laufen wird.

(102) Als **Analyse** bezeichnen wir die Diagnose und Erfassung eines Problems mit dem Auftraggeber, Suche nach Ursachen, Erarbeiten einer Lösung.

(103) Als **kleine Optimierungen** bezeichnen wir Änderungen der Software, für die sich ein *Weiterentwicklungsauftrag* nicht lohnt.

(104) Als **Qualitätssicherung** bezeichnen wir das Schreiben und Pflegen von automatisierten Testverfahren.

(105) Als **Deployment** bezeichnen wir die Veröffentlichung der Software mit dem Ziel, sie auch für andere Entwickler verfügbar zu machen.