



ILIAS open source e-Learning e.V.

Technical Board

//

Berrenrather Straße 177

D-50937 Köln

//

E-Mail: tb@ilias.de

Web: www.ilias.de

Fax: +49 221 56 07 84 58

Projektbericht

Removing of Legacy-UI

vorgelegt von Richard Klees

für das Technical Board des ILIAS open source e-Learning e.V.

Einleitung

Das Projekt "Removing of Legacy-UI" oder kurz "Removing LUI" ist das zentrale Projekt des Technical Boards der Wahlperiode '23-'25. Zentral sowohl in der Verausgabung der Mittel des TB als auch in der Bedeutung für die Software ILIAS und deren Community. Mit der Neuwahl des Technical Boards, aber auch mit der mittlerweile erheblichen Laufzeit ist nun ein Moment gekommen, um die bisherigen Ergebnisse des Projekts zu sichern.

Ob es sich bei diesem Bericht um einen Zwischen- oder Abschlussbericht handelt, kann an dieser Stelle nicht geklärt werden. Hier wird vom Technical Board zum einen eine unklare Haltung der Community gegenüber der gemeinsamen Arbeit an Querschnittsverbesserungen wahrgenommen. Zum einen sind viele laute und leise Stimmen der Kritik am Projekt "Removing LUI" aber auch zu Querschnittsprojekten generell zu hören. Teile davon werden im Bericht aufgegriffen. Auf der anderen Seite steht zum Beispiel die Umfrage im Beirat, die einen überwiegend positiven Blick der Umfrageteilnehmenden auf "Removing LUI" und Querschnittsprojekte im Allgemeinen aufzeichnet. Zu dieser Grundsatzdiskussion wollen wir unsere Sicht, aber auch einige Fakten beisteuern. Es ist am nächsten TB zu entscheiden, inwiefern das Projekt "Removing LUI" im Rahmen der TB-Arbeit fortgeführt wird.

Neben der Frage nach den Großprojekten ist "Removing LUI" wegen seiner exponierten Rolle und der breiten Befassung mit der ILIAS-Codebase und -Community auch Beispiel und Schauplatz für verschiedene andere Fragen und Diskussionen, die in der ILIAS-Community geführt werden. Am

Haben Sie Fragen?
verein@ilias.de

Geschäftsführer:
Matthias Kunkel
VR 16017 / Amtsgericht Köln
USt-ID: DE 815 319 517

Vorstand:
1. Vorsitzender: Oliver Samoila
2. Vorsitzende: Yvonne Fischer
Schatzmeister: Thomas Schroeder

Bankverbindung:
IBAN: DE26 3705 0198 1902 5687 14
SWIFT-BIC: COLSDE33
Sparkasse KölnBonn

Projekt lassen sich in diesem Sinne auch allgemeine Fragen und Probleme adressieren, deren Tragweite über die Frage hinausgeht, ob und wann nun die eine UI-Komponente durch eine andere ersetzt werden könnte. Auch zu diesen Diskussionen wollen wir unsere Beobachtungen und Meinungen festhalten.

Bevor dieser Bericht nun in die Details, die nüchternen Daten und Fakten absteigt, wollen wir einmal deutlich und sichtbar festhalten: Aus unserer Perspektive ist das Projekt "Removing of Legacy-UI" in vielen Bereichen ein Erfolg für die Community. Bezüglich seiner Ziele ist es zum aktuellen Zeitpunkt mindestens als Teilerfolg zu werten. Diese Erfolge haben sich die vielen Beteiligten zuzuschreiben: die Entwickler*innen, Programmierer*innen, Designer*innen, Konzepter*innen; die Kolleg*innen, die in ihren Institutionen versucht haben, Funding für die gemeinsamen Ziele aufzutreiben; die Tester*innen und Nutzer*innen, die die Ergebnisse kritisch geprüft haben; und viele andere, die wir vielleicht hier vergessen. Und nicht zum Schluss unser Projektmanager, Oliver Samoila, der die vielen Beteiligten und uns freundlich, aber bestimmt in Richtung Ziel geleitet hat. Wir wünschen jede* Auftraggeber*in eine*n Projektmanager*in wie ihn, d* nicht nur Stunden aufschreibt und einen Auftrag abarbeitet, sondern d* mit Motivation, Herzblut und Identifikation mit dem Thema bei der Sache ist. Vielen Dank.

Der Bericht ist im Folgenden so strukturiert: Zunächst folgt ein kurzer Abschnitt, der die nüchternen Realitäten des Projekts zusammenfasst. Ausführlichere Darstellungen der Zahlen, Daten und Fakten sollten sich normalerweise in den Vorträgen von Oliver und auf der Projektseite finden lassen. Die Zahlen, Daten und Fakten sind zum einen für sich selbst interessant, dienen zum anderen aber auch dazu, Aussagen aus den anderen Berichtsteilen zu stützen, wo möglich. Im Weiteren ist der Bericht dann in drei Teile eingeteilt. "Das Gute" erfasst Dinge, die so gelaufen sind, wie wir es uns gewünscht oder vorgestellt hatten. "Das Schlechte" thematisiert Probleme, Dinge, die anders gelaufen sind als gewünscht, die im Einflussbereich des Projekts liegen und durch eigene Aktivität hätten verändert oder anders angegangen werden können. "Das Hässliche", schließlich, thematisiert Probleme, die außerhalb der Reichweite des Projekthandelns liegen und die, wenn möglich, auf anderer Ebene angegangen werden sollten.

Zahlen, Daten, Fakten

Zum aktuellen Zeitpunkt hat das Projekt eine Laufzeit von bald 3 Jahren. Die Ausschreibung zum Projektmanagement wurde vom Technical Board Anfang August 2022 herausgegeben. Das Angebot von Oliver ging dann Ende August ein und wurde im September 2022 beauftragt. Projektauftritt mit dem Projektmanager Oliver und dem Technical Board im November durchgeführt. Der Gesamtpjektauftritt mit Beginn des Fundraisings, mit allen Beteiligten, war dann im Januar 2023.

Für die Arbeit des Projektmanagers wurden in 29 Monaten Laufzeit eine Gesamtsumme von 78.700€ ausgegeben (Stand März '23). Das entspricht, bei einem rabattierten Tagessatz von 945€, insgesamt 560 Stunden, von denen 120 Stunden für Fundraising und 440 Stunden für Projektmanagementtätigkeiten aufgewendet wurden.

Über das Fundraising wurden bisher 375.000€ als Budget eingebracht, 49.000€ sind aktuell noch über drei mögliche Budgets aufgebracht. Beteiligt an diesem Funding waren 23 Organisationen:

- ILIAS open source e-Learning e.V. (TB + Beirat)
- Katholische Hochschule Nordrhein-Westfalen
- Universität Bern
- Sächsische Informatik Dienste
- Fachhochschule Aachen
- Technische Hochschule Ostwestfalen-Lippe
- Fachhochschule Vorarlberg
- Katholische Hochschule Freiburg
- Pädagogische Hochschule Zürich
- Philipps-Universität Marburg
- Karlsruher Institut für Technologie (KIT)
- Fachhochschule Münster
- Deutsche Lebens-Rettungs-Gesellschaft e.V.
- Duale Hochschule Baden-Württemberg

- Universität Hohenheim
- Universität Basel
- Hochschule Bremen
- Hochschule Bonn-Rhein-Sieg
- Hochschule Bielefeld
- Universität Freiburg
- Universität zu Köln
- Hochschule Luzern
- Fachhochschule Dortmund

Vielen Dank!

Dadurch ergibt sich ein Funding von etwa 5€ pro durch das TB eingesetztem Euro für das Projektmanagement und eine Fundingergebniss von etwa 3000€ pro Stunde Fundraising.

Von Service Providern wurde Funding außerdem als Eigenleistung erbracht, zum Beispiel in Form von Teilnahmen an Abstimmungen, Meetings oder bei der Entwicklung von UI-Komponenten. Diese Eigenleistungen wurden in etwa folgenden Umfängen erbracht:

- CaT (45 PT)
- Databay (8-10 PT, außerdem Nachlass auf den PM-Tagessatz im Umfang von entsprechend 14,5 PT)
- Kergomard (5 PT)
- Leifos (16 PT)
- sr.solutions ag (8-9 PT)

Alle die vorgenannten Service Provider, wie auch die Kröpelin Projekt GmbH und die Internetlehrer GmbH, haben in unterschiedlichem Maß, aber insgesamt umfangreich mit Teilnahmen an und Inputs in den Projektmeetings beigesteuert. Zu den Eigenleistungen können ferner sowohl Konzept- als auch Implementierungs- und Dokumentationsleistungen gehören. Auch hierfür vielen Dank.

Durch diese beiden Beiträge, als Funding oder Eigenleistung, ergibt sich (exklusive PM/Fundraising) eine Gesamtvolumen von etwa 500.000€ als Investments der ILIAS-Community.

Auf der Seite der Beteiligung ergibt sich bei sorgsamer Schätzung eine Zahl von etwa 30 Personen, die am Projekt mitgearbeitet haben. Diese Beteiligungen fand auf 39 Projektmeetings statt, wovon 8 als Workshops und alle anderen als Statusmeetings durchgeführt wurden. Bei den Statusmeetings waren jeweils zwischen 7 und 19 Teilnehmer*innen anwesend, im Durchschnitt waren es 12 Teilnehmer*innen pro Sitzung. Die etwa 30 Mitarbeitenden wurden dabei von 7 oben genannten Service Providern sowie den institutionellen Mitgliedern Uni Bern, Uni Hohenheim und Uni Stuttgart, entsandt.

Entfernt bzw. durch neue UI Komponenten ersetzt wurden durch die Aktivitäten im Projekt folgende Legacy-Komponenten:

- AdvancedSelectionListGUI
- Buttons
- ImageLinkButton
- JsLinkButton
- LinkButton
- SubmitButton
- Special Character Selector
- GlyphGUI
- GroupedListGUI
- LightboxGUI
- ModalGUI
- OverlayGUI
- PanelGUI
- SplitButtonGUI
- TooltipGUI



Mit der Ersetzung begonnen wurde bei folgenden Legacy-Komponenten:

- Table2GUI
- TableGUI
- ToolbarGUI
- Filter an Table2GUI und TableGUI
- ProgressBar

Im Projekt als Ziele identifiziert wurden folgende Legacy-Komponenten, bei denen mit der Ersetzung aber noch nicht begonnen wurde:

- CheckboxListOverlayGUI
- ConfirmationGUI & ConfirmationTableGUI
- Explorer
- NestedList

Von der Ersetzung zurückgestellt aufgrund der Komplexität des Vorhabens wurden diese Legacy-Komponente:

- TabsGUI
- SyntaxHighlighter
- TextHighlighterGUI

Für die Ersetzungen und Überarbeitung wurden folgenden neuen UI-Komponenten im UI-Framework im Rahmen des Projekts erstellt oder in den Einsatz gebracht:

- UI Data Table und Spaltentypen
- Ordering Table
- Input View Controls
- Ergänzung von KS Glyphen
- häufigere Nutzung: KS Launcher
- häufigere Nutzung: KS Modal

- häufigere Nutzung: KS Buttons in verschiedener Form
- häufigere Nutzung: KS Dropdowns
- häufigere Nutzung: KS ViewControls (Input und Erstversion)
- häufigere Nutzung: KS Filter
- Einsatz der HelpTopics
- häufigere Nutzung: Presentation Table und Listing Panels und assoziierte
- Modal » Lightbox Card Page
- Erweiterung Drilldown
- Erweiterung Footer
- KS Progress Bar
- Field Input Rating
- Field Input Markdown
- Prompt

Das Gute

Projektmanagement von Großprojekten

Wie schon in der Einleitung deutlich geworden sein sollte, ist das von Oliver praktizierte Projektmanagement als voller Erfolg im Projekt zu werten. Hier sind verschiedene Maßnahmen und Strategien herauszustellen, die aus Sicht des TBs zum Erfolg des Projekts wesentlich beigetragen haben.

Zunächst zu nennen sind die regelmäßigen Meetings zum Projekt. Diese fanden zu einem fixen Zeitpunkt am Montagnachmittag, im Wechsel mit dem Jour Fixe der ILIAS-Entwicklung, statt. Die Wahrscheinlichkeit, dass viele der Beteiligten an diesen Meetings teilnehmen konnten, war dadurch gut, da es sich um etwa die gleiche Zielgruppe handelt, die auch am Jour Fixe teilnehmen würde. Der Montagnachmittag sollte bei dieser im Normalfall regelmäßig ohnehin eher frei gehalten sein. Die Meetings waren also im Allgemeinen gut besucht und daher ein wichtiger regelmäßiger Kontaktpunkt

für die unmittelbar Projektbeteiligten. Gut war hierbei ebenfalls, dass die Meetings ausschließlich das Projekt zum Thema hatten und so alle Beteiligten sicher sein konnten, dass mit ihrer Zeit entsprechend wertschätzend umgegangen und keine anderen Themen diskutiert wurden. Durch die Vor- und Nachbereitung der Meetings im Etherpad war die Vorbereitung der Meetings effizient und auch Personen, die nicht teilnehmen konnten, war es möglich, sich im Nachgang zu informieren.

Neben der internen Kommunikation auf den regelmäßigen Meetings ist die Qualität und Konsistenz der Kommunikation nach innen und außen durch den Projektmanager ein wesentlicher Baustein beim Erfolg des Projekts gewesen. Dadurch, dass Oliver immer wieder selbst als Person mit dem Anliegen "Removing LUI" sichtbar geworden ist, war er nach kurzer Zeit das Gesicht des Projekts. Neben der reinen Sichtbarkeit und häufigen Wiederholung der Ziele des Projekts stellte seine Kommunikation auch einen wesentlichen Baustein zur Transparenz über den aktuellen Stand des Projektes dar. Die dafür erhobenen Zahlen und Übersichten waren sowohl auf der Projektseite zu finden, als auch in aufbereiteter Form Inhalt der diversen Vorträge und Inputs von Oliver. Die Projektkommunikation war dadurch aus einem Guss, es konnte wenig Unsicherheit über die Inhalte und Stände entstehen.

Das machte schließlich eine zentrale Sammlung und Verteilung der Fundings für das Projekt möglich. Zum einen waren hier die bisherigen Beiträge und Ergebnisse präsent, zum anderen waren die Zwecke und Ziele des Projekts in der Community weithin bekannt, so dass es bei der Suche nach Funding häufig um konkretere Fragen (siehe auch entsprechenden Abschnitt in "Das Schlechte") ging oder Funding sogar proaktiv angeboten wurde. Es ist fraglich, ob eine ähnliche Dynamik ohne Identifikationsfigur und konsistente Kommunikation möglich gewesen wäre. Gleichzeitig ist die zentrale Verteilung aber auch ein wesentlicher Punkt für die Effizienz im Projekt gewesen, da so viele Beteiligte entsprechend der Bedarfe möglichst fair mit Budgets bedient werden konnten und auf der anderen Seite Ansprechpartner*innen auf Auftraggeberseite nicht von unterschiedlichen Personen mit den gleichen Themen angesprochen wurden.

Die Investition der Mittel des Technical Boards in einen Projektmanager für das Projekt ist daher als gute Entscheidung zu werten. Dafür sprechen, neben dem Projektmanagement an sich, auch die begründete Vermutung, dass wahrscheinlich keine andere Institution fähig gewesen wäre, eine* Projektmanager*in mit einer ähnlich hohen Stundenzahl zu finanzieren oder zur Verfügung zu stellen. Auch die nackten Zahlen bezüglich Funding sprechen für diese Entscheidung. Nur im Hinblick auf die direkt zählbaren Beträge, die durch Oliver beim Fundraising eingeworben werden konnten, ergibt sich, dass jedem in das Projektmanagement investierte Euro 5 zusätzliche Euros aus der Community

geführt hat. Dabei sind Sachspenden, etwa in Form von Entwickler*innenzeit, noch nicht berücksichtigt. Das spricht für eine hohe Effizienz der eingesetzten Mittel.

Gemeinsame Arbeit an Großprojekt

In seinem Essay "Die Kathedrale und der Bazaar" vergleicht Eric S. Raymond proprietäre Software mit Open-Source-Software indem er eine Kathedrale und einen Marktplatz als Analogien heranzieht. Die Kathedrale, die für die proprietäre Software steht, verkörpert dabei strukturierte und zielgerichtete Gestaltung, Planung und Ressourceneinsatz. So entsteht ein hohes, schönes Gebäude, das kohärent strukturiert und gestaltet ist. Der Marktplatz, als Symbol für Open-Source-Software, ist auf der anderen Seite geprägt von den spontanen Interaktionen und individuellen Zielen der Verkäufer*innen und Käufer*innen, die sich dort begegnen. Struktur und Gestaltung sind nicht geplant, sondern emergent und der Marktplatz streckt sich eher weit aus als in die Höhe gebaut zu sein. In diesem Bild ist einiges an Wahrheit und Erkenntnis zu finden, aber auch diese Analogie bricht zusammen, wie alle Analogien. Auch der Marktplatz braucht Strukturen, auf denen er entstehen kann und auf dem sich die Akteure begegnen können. Diese müssen, trotz eventuell fehlender zentraler Steuerung, gemeinsam produziert werden.

Das Removing-LUI-Projekt zeigt in diesem Sinne: Wir können das. In der ILIAS-Community können gemeinsame, langfristige und weitreichende Ziele auch gemeinsam erreicht werden, auch mit einer hohen Zahl an mehr oder weniger stark beteiligten Akteur*innen. Auch wenn es ein zentrales Projektmanagement gab: Die detaillierte Zielsetzung und die Erreichung der Teilziele sind keine Einzelleistung des Projektmanagers, sondern eine gemeinsame Leistung aller Beteiligten. Die genaue Zahl der Mitarbeitenden lässt sich mit vertretbarem Aufwand nur schätzen, aber die Zahl von etwa 18 wechselnden Beteiligten alleine am regelmäßigen Projektmeeting gibt einen Eindruck davon, wie viele Personen am Ende mitgearbeitet haben könnten. Die Schätzung von Technical Board und Projektmanager ist, dass um die 30 Personen zum engeren Kreis der Projektmitarbeitenden gehört haben dürften, die alle gemeinsam die verschiedenen Ziele gesetzt und erreicht haben. In der Tiefe und Breite der Ziele ist dies in der ILIAS-Community aktuell höchstens mit dem PHP 8-Projekt vergleichbar.

Eine wesentliche Taktik, um die gemeinsame Arbeit zu strukturieren, war es, die Gesamtaufgabe gemeinsam in kleinere, möglichst parallelisierbare Teil aufzuteilen. Deren Erfolg konnte dann sowohl vom Projektmanager gut gemonitored und kommuniziert werden. So konnte Beteiligung, sowohl

beim Funding als auch bei der Durchführung, auch in sehr kleinen Paketen geleistet werden und die Schwelle zur Teilnahme war gering. Das Aufteilen der Gesamtaufgabe in Kleinaufgaben ist dabei offensichtlich eine Standardtaktik in der Projektdurchführung. Wesentlich bei einer größtenteils auf Konsens angewiesenen Projektorganisation war jedoch die gemeinsame Befassung und Aufteilung der Aufgaben mit allen interessierten Beteiligten und die regelmäßige Versicherung darüber, dass die Aufteilung so weiter hilfreich und durchführbar ist. Die Methode musste hier also in einer Bottom-Up-Form angewandt werden.

Hilfreich bei dieser Aufteilung war weiterhin, dass es möglich war, sie mit automatisierten Tools zu unterstützen. So wurde regelmäßig und automatisch eine Liste erstellt, die die noch zu bearbeitenden Stellen aufgelistet hat. Dies war sowohl für die Gesamtsteuerung als auch für die Aufgabenfindung einzelner Beteiligter hilfreich. Hier empfiehlt sich auch in Zukunft weiterhin regelmäßig zu schauen, ob und wie verschiedene Aufgaben sinnvoll durch Automatisierungen unterstützt werden können.

Auch wenn ein guter Teil der gemeinsamen Arbeit durch diese Divide-and-Conquer-Taktik strukturiert war, lässt sich ebenfalls festhalten, dass im Rahmen des Projekts auch komplexe Vorgänge mit vielen Stationen, Abhängigkeiten und Beteiligten beherrscht wurden. Beispielhaft für einen solchen Vorgang ist die Erstellung der UI-Komponente Data Table, die in mehreren Iterationen mit vielen Beteiligten zunächst geplant, dann umgesetzt und schließlich als Ersatz für die Bestandstabelle eingebunden wurde. Insbesondere die Erstellung komplexer UI-Komponenten wurde mehrfach im Rahmen solcher langlaufenden, komplexen Vorgänge mit teilweise dutzenden Beteiligten bearbeitet. Auch wenn sich an Details sicher immer Kritikpunkte finden lassen, können wir feststellen, dass auch diese Vorgänge in der gemeinsamen Arbeit in der ILIAS-Community beherrscht wurden, sowohl aus Perspektive der Organisation, der Durchführung und der Finanzierung.

Menge und Qualität der ersetzten UI-Komponenten

Während der aktuellen Projektlaufzeit konnten keineswegs alle alten UI-Komponenten durch neue ersetzt werden und aktuell ist auch nicht realistisch abschätzbar, wann dieses Ziel erreicht sein könnte. Trotzdem lässt sich an der Menge, der Größe und auch an der Qualität der neuen Komponenten ein klarer Erfolg des Projekts festmachen.

Bei der Ersetzung fällt zunächst eine große Gruppe kleiner, nicht sonderlich komplexer UI-Komponenten auf: Glyphen, Buttons, Dropdowns,... Hier konnte, bis auf wenige Ausnahmen, jeweils der gesamte Bestand von Legacy-Komponenten durch die neuen Komponenten ersetzt werden.



Dementsprechend konnte auch der alte Code aus dem Repository entfernt werden und muss nun nicht mehr gepflegt werden.

Mit den Formularen und der Presentation Table waren zwei komplexere Komponenten bereits vor Beginn des Projekts verfügbar. Diese wurden im Rahmen des Projekts zu einer vermehrten Nutzung gebracht. Als wichtige und komplexe Komponente kam außerdem das Data Table als Ersatz für die iTable2GUI und das verwandte Ordering Table hinzu. Diese Komponenten konnten im Rahmen des Projekts sowohl entwickelt als auch teilweise zum Einsatz gebracht werden.

Als komplett neue Komponente zu nennen ist der UI Launcher zum Starten von bspw. Tests. Diese Komponente ist nicht lediglich eine Überarbeitung einer Bestandskomponente, sondern wurde im Hinblick auf Anforderungen aus verschiedenen ILIAS-Komponenten und Anforderungen aus der Usability und Barrierefreiheit komplett neu entworfen und in den Einsatz gebracht.

Bereits jetzt lässt sich in Bezug auf die Verwendung der neuen Komponenten feststellen, dass die Qualität der Benutzerschnittstelle von ILIAS in vielen Bereichen deutlich gesteigert werden konnte. Gestaltungsprobleme, z.B. uneinheitliche Gestaltung derselben Funktionalitäten in verschiedenen Komponenten, wurden aufgedeckt und angegangen; die Anzeige des Systems auf kleinen Bildschirmen ist bei Nutzung der neuen Komponenten deutlich verbessert; Usability und Accessibility-Issues können deutlich zentraler angegangen werden als bisher.

Neben den Verbesserungen für die Nutzer*innen und in der Wartbarkeit des Projekts, gibt es auch viele positive Rückmeldungen zur Nutzbarkeit der neuen UI-Komponenten durch die ILIAS-Entwickler*innen. So meldet z.B. Tim Schmitz von Leifos zu den "Prompts" für die "UI Data Table" zurück: "Just tried it out and works like a charm!"¹ Ähnlich positive Rückmeldungen der Entwickler*innen gab es zu vielen Gelegenheiten. Es ist anzunehmen, dass die neuen UI-Komponenten daher auch einen Beitrag dazu leisten, dass neue Funktionalitäten in ILIAS zukünftig bei Nutzung der neuen Komponenten mit viel Freude und wenig Reibung implementiert werden können.

1 <https://mantis.ilias.de/view.php?id=40688#c112175>

Das Schlechte

Schwierige Suche nach Funding

In reinen Zahlen ausgedrückt sieht ein Verhältnis von 5 eingeworbenen Euros pro ausgegebenem Euro aus Perspektive des TB-Budgets zunächst gut aus. Das soll jedoch nicht darüber hinwegtäuschen, dass die Suche nach Funding für dieses Großprojekt keineswegs eine leichte Aufgabe war, bei der vermutlich außerdem noch erhebliche Effizienzreserven zu heben wären. Ganz allgemein stellt sich die Frage, inwiefern sich, auf Basis der komplizierten Gesamtsituation die in Folge dargestellt wird, der Fundingserfolg in Zukunft oder mit einer anderen Person als Fundraiser*in, überhaupt noch einmal in dieser Form wiederholen lässt.

Als ein Grundproblem ist hier zunächst zu nennen, dass die wenigsten der beitragenden Organisationen problemlos einen generischen Auftrag für ein Projekt wie Removing-LUI erteilen können oder wollen. Auch Aufträge, die "Überarbeitungen" o.ä. versprechen, sind nicht ohne weiteres zu erhalten, da hier anscheinend oft unklar ist, warum überarbeitet werden muss oder was der konkrete Nutzen ist. Ebenso schwierig sind "Verbesserungen", so es sich nicht um spezifische Themen handelt, die aus welchem Grund auch immer auf der Agenda der entsprechenden Institution stehen. Diese wiederum werden im Wesentlichen abgebildet durch eine Menge von Codewörtern in denen sich die jeweiligen Zielsetzungen spiegeln: "Barrierefreiheit", "digitale Prüfungsszenarien", ... Bei dieser Problematik ist dabei nicht einmal anzunehmen, dass konkrete Ansprechpartner*innen unwillig sind, das Projekt zu finanzieren, lediglich steht Finanzierung nur für spezielle Zwecke zur Verfügung und nicht zur Pflege der Basissoftware.

In der Praxis führt dies zu oft kleinteiliger Fleißarbeit bei der Suche nach Funding. Nicht nur muss ein möglicher Aufhänger und Verpackung für Bedarfe im Projekt gefunden werden. Diese müssen dann auch auf die korrekten Auftragnehmer*innen und die geeigneten Änderungen übertragen werden. Hinzu kommt, dass es sich bei diesen Aufträgen häufig auch um eher kleinere Beträge handelt. So mussten vom Projektmanager eine große Menge von kleinteiligen Kombinationen aus Auftraggeber/Auftragnehmer/Änderungsziel verwaltet und in eine durchführbare und nachvollziehbare Reihenfolge gebracht werden. Dass es dabei zu Reibungsverlusten mindestens im Projektmanagement kommt, ist offensichtlich. Ebenfalls klar sollte sein, dass sich diese Aufgabe nur mit einem dichten Netz aus Beziehungen und Kenntnissen über die Umstände der vielen Beteiligten annähernd befriedigend lösen lässt.

Nicht nur bei der Verwaltung des Auftragsbestands sondern auch bei der konkreten Einwerbung von Funding wurde diese Aktivität teilweise als wenig motivierend wahrgenommen. Sie konkurriert auch mit allen anderen Aufgaben, organisatorisch oder inhaltlich, die im Projektmanagement zu erledigen gewesen wären. Lieber die Projektseite auf den neuesten Stand bringen oder das nächste "Nein" kassieren beim Versuch, ein Funding einzuwerben? Lieber bei eine* Entwickler*in nach dem Arbeitsstand fragen oder hinter einem möglicherweise freiwerdenden spezifischen Budget hinterherjagen? Hier zeigt sich, dass die Arbeit "Projektmanagement" und "Fundraising" zwar aus pragmatischen Gründen bei der gleichen Person lagen, aber doch inhaltlich miteinander konkurrieren und möglicherweise daher auch personell hätten getrennt werden können.

Gegen eine Trennung spricht allerdings, dass durch Finanzierungs- und Projektstruktur ein nicht unerheblicher Informationsfluss zwischen eigentlichem Projektmanagement und Fundraising notwendig ist. An den Ergebnissen des Fundraisings hängt mindestens die Planbarkeit der Aktivitäten im Projekt. Durch die vielen kleinen und teilweise sehr spezifischen Aufträge entstehen Abhängigkeiten bei der Durchführung und der Terminierung der Aufträge, die aufgelöst werden müssen. Unspezifischere Aufträge mussten genutzt werden, um spezifischere Aufträge durchführbar zu machen. Inwieweit sich dieser teilweise kleinteilige Abstimmungsbedarf in einem Team aus Projektmanager*in/Fundraiser*in hätte lösen lassen ist zumindest bedenkenswert. Inwieweit die Arbeit durch eine andere Person in ähnlicher Qualität hätte geleistet werden können ist ebenfalls fraglich.

Erhebliche Teile der Unsicherheiten in der Planung und Durchführung, die durch das schwierige Fundraising entstanden sind, wurden durch Vor- oder Eigenleistungen der Service Provider abgedeckt. Dadurch kommt als Dimension für die Projektorganisation noch die individuelle Leistungsfähigkeit und Risikobereitschaft, aber auch der verwaltete Bestand (siehe auch Abschnitt "Ungleiche Last") und die Geschäftsmodelle der einzelnen Beteiligten hinzu. Hier wurden von den Service Providern teilweise erhebliche Umfänge als Vorleistung abgedeckt, bei denen nicht zu jedem Zeitpunkt wirklich klar war, ob sich eine Finanzierung finden lassen würde. Positiv ist hier zu erwähnen, dass dieses Problem offen in den Projektmeetings thematisiert wurde und dementsprechend auch von der Gruppe aller Projektbeteiligten mindestens in Teilen mitgetragen werden konnte.

Die Gruppe der Organisationen schließlich, die Finanzierung in der Form von Funding zur Verfügung stellen konnte, ist mit 23 Organisationen zwar nicht klein, aber angesichts der Mitgliederzahlen des

Vereins und der Menge der ILIAS-nutzenden Organisationen dann doch überschaubar. Auch sind in der Liste der Organisationen für langjährige Beobachter*innen der ILIAS-Entwicklung kaum überraschende oder neue Namen zu finden. Es lässt sich feststellen: Die Finanzierung des Projekts ist von den üblichen Verdächtigen getragen worden. Ähnliches zeigt sich bei der Höhe der zur Verfügung gestellten Beträge. Diese sind nicht gleichverteilt. Einige Organisationen tragen erhebliche Summen bei, andere eher kleine Beträge. Auch hier findet sich keine Überraschung. Es lässt sich also mit einigem Recht die Frage stellen: Konnten durch das Projekt tatsächlich neue Beträge eingeworben werden? Oder handelt es sich bei dem Projektvolumen letztlich lediglich um Budget, das ohnehin in ILIAS geflossen wäre und das durch das Projekt eine gemeinsame Richtung bekommen hat? Ein durch das Projekt koordinierter Einsatz der Budgets wäre kein geringer Erfolg, lässt allerdings die eingangs aufgestellte Erfolgsrechnung für das eingesetzte Budget des Technical Boards in einem anderen Licht erscheinen.

Schließlich verweist das Bild beim Funding auf ein generelles Problem bei der Finanzierung von grundständigen Arbeiten an der gemeinsamen Software. Diese ist, mit den aktuellen Finanzierungsmodellen und Communitystrukturen, anscheinend immer prekär. Etwas überspitzt gesagt, scheinen viele der Institutionen und Beteiligten an der Entwicklung und Finanzierung von ILIAS ein ständiges Chicken Game miteinander zu spielen. Wenn diese oder jene Grundlagenverbesserung notwendig ist, wird sie irgendetwas am Ende finanzieren und ich man kann mein Budget für mein Partikularinteresse einsetzen. So oder so ähnlich scheint die bewusste oder unbewusste Logik vieler Akteure zu lauten. Der Verein, der dabei anscheinend oft als Träger der Gemeininteressen wahrgenommen wird, kann diese Rolle wegen der insgesamt unzureichenden Finanzierung nicht ausreichend ausfüllen. Unter dieser Perspektive ist Ressourceneinsatz des Technical Boards zur Bündelung der verfügbaren Budgets möglicherweise schon das effizienteste und effektivste, was im aktuellen Modell machbar ist. Eine grundlegende Änderung der prekären Finanzierung würde unter der Annahme, dass der Verein weiterhin Träger der Gemeininteressen sein soll, nur stattfinden können, wenn sich die Ressourcen des Vereins erheblich erhöhen, beispielsweise durch einen deutlichen Anstieg der Mitgliedsbeiträge.

Ungleiche Last

Als ebenfalls große Schwierigkeit zeigte sich im Projektverlauf die stark unterschiedlichen Änderungslast für die verschiedenen Komponenten und Entwickler*innen. Dass dies so sein würde, war zwar im Vorfeld durchaus klar, das Grundproblem zeigte sich aber in deutlich mehr Bereichen als

der reinen Arbeitslast, was im Technical Board in dieser Form im Vorfeld nicht antizipiert wurde. Keines der im Folgenden beschriebenen Probleme ist dabei wirklich speziell für das Removing LUI-Projekt, ähnliche Probleme ließen sich auch beim PHP 8-Projekt beobachten und werden sich bei weiteren Projekten der Größe und des Zuschnitts beobachten lassen.

Die unterschiedliche Lastverteilung entsteht dabei auf verschiedenen Dimensionen. Zum einen ist die reine Menge der Komponenten in der Pflege einzelner Kolleg*innen stark unterschiedlich. Während einige Kolleg*innen eine oder zwei Komponenten verantworten, geht diese Menge bei anderen in die Dutzende. Auch ist die Komplexität der Komponenten sehr unterschiedlich. Manche stellen recht gradlinig einzelne Funktionalitäten mit wenig Einstellungsmöglichkeiten zur Verfügung, andere Komponenten verfügen über große Mengen unterschiedlicher Einstellungen, Abläufe und Ansichten und es ist fraglich, ob diese überhaupt alle von einer einzelnen Person gekannt werden können. Schließlich ist die Ausgangssituation, in der die Komponenten vorzufinden sind, stark unterschiedlich. Manche Komponenten gibt es seit je her und die Ressourcen für die dauernde Pflege waren anscheinend überschaubar, so dass entsprechend viele Layer an Vorgehensweisen und Implementierungen im Code zu finden sind. Andere Komponenten sind neu oder vor kurzem überarbeitet worden, so dass hier eher moderne Ansätze in der Implementierung zu finden sind.

Neben den zeitlichen Aufwänden beim Einbau neuer UI-Komponenten in der Komponente selbst, entstanden auch andere Effekte, die die Arbeit im Projekt erschwerten oder verlangsamten. Zum einen war es schwer, auf einem allgemeinen Level Bedarfe zu planen, da die Ausgangszustände der Komponenten so unterschiedlich waren. Aufwände, um eine bestimmte UI-Komponente zu verbauen, waren, trotz scheinbar gleicher Aufgabenstellung und Mengengerüst, stark unterschiedlich. Bei Komponenten mit hoher interner Komplexität war teilweise bereits die reine Erhebung von Bedarfen oder Mengengerüsten eine Herausforderung. Selbst wenn alle Ansichten und UI-Komponenten bekannt waren, war es teilweise schwer, UI-Elemente auf speziellen Ansichten in allgemein nutzbare UI-Komponenten zu überführen. Die Arbeiten gingen oft weit über die reine Ersetzung der Komponenten hinaus, bis in die Neugestaltung kompletter Abläufe oder Ansichten. Der Bedarf für solche tiefgehenden Arbeiten ist dabei noch lange nicht abgeschlossen und umso größer, je stärker die einzelnen Komponenten im Grunde als fast autonome Teilsysteme im Gesamtsystem betrachtet werden können.

Auch kann an dieser Stelle hinterfragt werden, inwiefern die Geschäfts- und Finanzierungsmodelle der Service Provider in Bezug auf die von ihnen gewarteten Komponenten, deren Strategien zur Kundenpflege und zum Marketing, zu einer unterschiedlichen Wahrnehmung oder Artikulation der

Lastverteilung führen, insbesondere in der Verknüpfung mit den real sehr unterschiedlichen Aufwänden, die zu leisten waren. Auf der einen Seite steht ein Modell wie die pro-bono Pflege einer einzelnen, recht simplen Komponente. Dabei wird möglicherweise ohnehin ein bestimmter, aber gut abschätzbarer Aufwand pro Jahr erwartet und die konzeptuellen Ansprüche in der Komponente sind moderat oder kaum vorhanden. Die zu tragende Last wird dadurch ebenfalls als tragbar angenommen, gleichzeitig werden in den anderen Dimensionen keine weiteren Lasten erzeugt. Änderungen an der Komponente führen weder in komplexe Diskussionen um UI-Komponenten, noch treten Überraschungen bei der Befassung mit den Funktionen in der Komponente auf. Durch die geringe Gesamtlast beeinflusst das Projekt die gesamte Leistungsfähigkeit des Service Providers kaum.

Auf der anderen Seite steht die Pflege von Komponenten mit erheblicher Geschichte und Umfang, die möglicherweise auch wichtige Bestandteile des Portfolios eines Service Providers sind. Hier ist dann schon alleine das Mengengerüst der zu bearbeitenden UI-Elemente erheblich, aber auch die Ansprüche der Nutzenden, die Vorstellungen davon, wie bedient werden soll, wie Abläufe funktionieren. Dementsprechend sind auch UI-Komponenten dann nicht einfach aus dem Regal zu nehmen, sondern müssen mit erheblichem Aufwand gestaltet und mit den Ansprüchen anderer, vielleicht ebenso wertvoller und wichtiger Komponenten, überein gebracht werden. Die Zahl der Teilnehmenden an Diskussionen steigt dann genauso, wie die Zahl der Ansprüche, die an eine Überarbeitung gestellt werden. (Achtung, mir scheint hier ein Widerspruch zur Zahl oben - 25 - sk)

Schließlich ist die regelmäßige Erfüllung von Kundenwünschen oder regelmäßige Bearbeitung von dieser oder jener Komponente unter Umständen ein Teil des Selbstverständnisses oder der Strategie der Service Provider. Wenn nun viel Aufwand in Bedarfe aus dem Removing LUI Projekt gesteckt wird, ist vorstellbar, dass hierdurch die wahrgenommene oder tatsächliche Möglichkeit auch andere Kundenwünsche zu erfüllen geringer wird und damit auch bestehende Kundenbeziehungen gefährdet werden. Andere Szenarien, wie Geschäftsmodelle die Möglichkeit oder den Wunsch beeinflussen, größere Lasten in einem Projekt wie Removing LUI zu tragen, sind absolut vorstellbar.

Keines der Geschäftsmodelle oder keine der Unternehmensstrategie ist dabei per se besser oder schlechter als andere, sie fügen allerdings eine weitere Dimension von Komplexität bei Planung und Durchführung des Projektes hinzu, die im Projektmanagement bearbeitet werden muss. Dies ist umso schwerer, als dass diese Modelle oder Strategien kein typisches Diskussionsthema bei Meetings wie dem Projektmeeting sind, oder manchen Anwesenden vielleicht sogar gar nicht bewusst sind. Es muss

also, ähnlich wie in den Absätzen oben, eher mit Vermutungen gearbeitet werden, als dass entstehende Probleme wirklich offen angesprochen werden.

Positiv zu bemerken ist hier eine Entwicklung, die mindestens teilweise auch durch das Removing LUI-Projekt angestoßen wurde: Es scheint sich in der Gruppe der für die Komponenten verantwortlichen Personen ein verändertes Gefühl über die realen Risiken und daraus entstehenden Lasten in der Zukunft zu entwickeln. Dies führt aktuell zu einer verstärkten Abgabe von zu pflegenden Komponenten oder Teilkomponenten, sei es durch Aufgabe der Features oder durch Angebote, die Pflege der Komponenten an andere Personen zu übergeben. Für die Community ist dies zunächst traurig oder gar beängstigend, sind hier doch Funktionen von der Abschaffung bedroht. Dahinter stehen jedoch jeweils reale Probleme, die sich ohnehin früher oder später gezeigt hätten und dies nun in einer recht drucklosen Situation tun, die Übergaben und Zukunftspläne für die Komponente in einer geordneten und ruhigen Form ermöglichen. Es werden keine neuen Probleme geschaffen, sondern bestehende Probleme offenbart. Die Abgabe der Komponenten schafft Platz für neue Akteure, sich sinnvoll an der Entwicklung zu beteiligen und erlaubt es der Community, eine Bewertung der Wichtigkeit verschiedener Komponenten vorzunehmen.

Das Hässliche

Gefühlte Konkurrenz zu Features

Auch wenn die Kommunikation über Projektziele und -stände als durchaus erfolgreich bewertet werden kann, lässt sich ein kommunikatives Problem am Projekt beobachten, das nicht unerwähnt bleiben soll. Auch wenn in einer Umfrage im Beirat das Projekt als überwiegend positiv bewertet wurde, gab es doch zu verschiedenen Gelegenheiten von verschiedenen Mitgliedern der ILIAS-Community Rückmeldungen, die das Projekt in Konkurrenz zu anderen Vorhaben, vor allem bezüglich funktionaler Weiterentwicklungen der Software, gestellt haben.

Diese Behauptung, Removing LUI mache andere Vorhaben schwieriger, ist zunächst nicht von der Hand zu weisen. Die Ressourcen der Community, ob Zeit der Entwickler*innen, Budget oder Kapazitäten in den Entwicklungsprozessen, sind in jeder Hinsicht endlich. Werden sie von dem einen Projekt in Anspruch genommen, stehen sie für das andere Vorhaben nicht zur Verfügung. Daneben lässt sich auch beobachten, dass das Projekt während seiner Laufzeit auch einen nicht unerheblichen

Anteil im kommunikativen Raum der Community beansprucht hat, der ebenfalls zu dem Gefühl der Konkurrenz beigetragen haben mag. Sowohl die spezifisch auf das Projekt gerichteten Inputs von Oliver, als auch unspezifischere Inputs mit ähnlichem Thema, z.B. Matthias' Keynote in Bologna, nehmen Raum in Anspruch, der nicht für die Diskussion und Darstellung neuer funktionaler Ideen zur Verfügung steht. Neben den verbrauchten harten Entwicklungsressourcen lässt sich vermuten, dass auch diese kommunikative Inanspruchnahme von Platz zu einem Gefühl der Konkurrenz zwischen dem Projekt und anderen Vorhaben geführt haben könnte.

Es ist allerdings fraglich inwiefern das Projekt dabei tatsächlich das zentrale Problem darstellt oder ob es sich hierbei nicht eher um das sichtbarste Beispiel für eine Reihe von ähnlichen Vorhaben oder Debatten handelt, die parallel stattfinden. Als weitere ähnliche Grundlagenprojekte sind dabei das Update auf PHP 8 zu nennen, das als ähnlich strukturiertes Großprojekt vor Removing LUI durchgeführt wurde. Weiterhin laufen aktuell die Umstellungen der Dateispeicherung aller ILIAS-Komponenten auf den ILIAS Resource Storage Service (IRSS) und die Aufruflogik der GUI-Klassen, wie sie in ilCtrl hinterlegt ist, erfährt verschiedene Änderungen, die in alle Komponenten hineinreichen. Ebenfalls parallel findet eine Restrukturierung der ILIAS-Verzeichnisstruktur und des Integrationsmodells der ILIAS-Komponenten statt, die Aufmerksamkeit erfordert. Removing-LUI ist von diesen Projekten das größte und wohl sichtbarste. Eine Gemeinsamkeit besteht darin, dass es sich bei den Vorhaben um großflächige Grundlagenarbeiten handelt, deren Nutzen für die Software sich für Beobachter*innen, die vor allem an Funktionen interessiert sind, nicht oder nicht ohne weiteres erschließt. Es ist also anzunehmen, dass sich diesbezüglich Unzufriedenheit, die auch zu den anderen Projekten gehört, an Removing LUI sammelt.

Weiterhin kann vermutet werden, dass an dem beobachteten Kontrast "Removing LUI vs. Funktionen" auch zwei Diskussionen beispielhaft geführt werden, die ebenfalls seit geraumer Zeit in der ILIAS-Community mehr oder weniger latent sind. Eine Diskussion ist die Frage, wie Qualität der Umsetzungen und deren Geschwindigkeit zueinander gewichtet werden. Das Grundproblem gehört zur Softwareentwicklung, wie wahrscheinlich auch zu vielen anderen Umfeldern: Bei einer einigermaßen festen Menge von Ressourcen lässt sich eine beliebige Qualität nicht zu einer beliebigen Geschwindigkeit erreichen. Arbeiten in hoher Qualität kosten Zeit und gehen daher auf die Umsetzungsgeschwindigkeit. Viele der Arbeiten an Removing LUI, insbesondere bei der Erstellung von UI-Komponenten, sind von einem hohen Qualitätsanspruch gezeichnet und waren dementsprechend nicht schnell zu erledigen. Es kann also angenommen werden, dass sich in Unzufriedenheit von

diesem Aspekt von Removing LUI auch eine allgemeine Unzufriedenheit mit der Abwägung zwischen Qualität und Geschwindigkeit, wie sie aktuell teilweise in der Community gemacht wird, ausdrückt.

Eine ähnliche Diskussion entsteht an der Frage, inwieweit die Qualität einer Umsetzung mit einer niedrigen Schwelle für die Beteiligung unterschiedlicher Personen im Konflikt steht. Auch dies ist ein durchaus übliches Problem, in der Softwareentwicklung wie auch anderen Bereichen: Der Bestand ist komplex und kompliziert, ebenso das Umfeld, in dem das System entwickelt und betrieben wird. Scheinbar simple Anforderungen oder Änderungswünsche müssen in den Bestand eingepasst werden und unter verschiedenen, teilweise hochspezialisierten Aspekten, wie beispielsweise Datenschutz, Barrierefreiheit oder Softwarearchitektur, betrachtet und umgesetzt werden. Das benötigte Wissen, sowohl um das System als auch in den Aspekten, ist zwar möglicherweise frei- und umfangreich verfügbar, lässt sich aber nicht ohne regelmäßige Befassung ad-hoc in die Anwendung überführen. Es entsteht das Gefühl, dass ein einfacher Wunsch mit einem Berg von Problemen belastet wird, die eigentlich gar nicht dazu gehören. Expert*innen werden als Bremser statt als Wissensträger wahrgenommen und alles scheint unglaublich kompliziert zu werden. Auch bei diesem Problem ist anzunehmen, dass Removing LUI hier ein Kondensationspunkt für Unzufriedenheit darstellt. Der Wunsch, die Benutzeroberfläche strukturiert, logisch und einheitlich zu gestalten, führte zu tiefen Befassung mit vielen Bereichen des Systems und der Umgebung. Die Diskussionen wurden zu großen Teilen von Kolleg*innen mit tiefem Wissen über ihren jeweiligen Fachbereich getragen und geführt und mögen für entferntere Beobachter unnötig kompliziert oder akademisch gewirkt haben. Ein Eindruck, der wahrscheinlich auch bei anderen Diskussionen in der Community entsteht und bei dem Removing LUI als beispielhaft erscheint.

Die Frage also, inwieweit spezifisch bei Removing LUI eine Konkurrenz zu funktionalen Weiterentwicklungen bestanden hat, lässt sich also so beantworten: Removing LUI ist für dieses Gefühl und die dahinterstehenden Grundfragen und -konflikte wahrscheinlich eher Beispiel als Auslöser. Das Projekt erzeugt keine neuen Fragen oder Probleme, sondern setzt Tendenzen und Diskussionen fort, die auch an anderen Stellen in sehr ähnlicher Form geführt werden oder werden könnten. Wichtig ist hier die Erkenntnis: Eine Unzufriedenheit bezüglich dieser Fragen und Probleme ist nicht schlecht oder irrational, sondern einfach vorhanden und anzuerkennen. Die materiellen Grundlagen dafür liegen in den endlichen Ressourcen, die stets zur Verfügung stehen. Diese zu perfekt zu verteilen auf "Aufräumen oder Neues machen", "Qualität oder Geschwindigkeit", "Qualität oder einfache Beteiligung" ist ein prinzipiell nicht lösbares Problem, sondern eine Abwägung mit Vor- und Nachteilen, die stets neu getroffen wird und auch stets bei dem ein oder der anderen zu

Unzufriedenheit führen wird. Klar ist: Sowohl die dauerhafte Pflege als auch die regelmäßige Verbesserung der Funktionalität gehören zum Produkt ILIAS. Sie sollten nicht gegeneinander gestellt, sondern zusammen gedacht werden. In diesem Sinne ist vielleicht schon der Name "Removing LUI" irreführend, ist das Projektergebnis doch an vielen Stellen eben nicht nur eine einfache Ersetzung der vorhandenen Oberfläche ohne weiteren Mehrwert, sondern auch eine echte Verbesserung in der Bedienbarkeit des Systems.

Abschätzung der Änderungsbedarfen

Das berechtigte Gefühl, dass unklar sei, wie viel Funding oder Zeit noch bis zu einer kompletten Beendigung des Projekts nötig sei, war ein regelmäßiger Begleiter im Projekt und wurde auch in der Umfrage zum Projekt im Beirat als problematisch angemerkt. Neben den oben beschriebenen Problemen im Funding, zeigt sich hier auch ein ganz allgemeines IT-Problem, das sich wahrscheinlich nicht 100% befriedigend lösen lässt, zumal im Kontext eines Open Source Projekts.

Die Abschätzung von Änderungsaufwänden, insbesondere bei weitreichenden Änderungen in alter Bestandssoftware, ist generell eher eine Kunst als eine Wissenschaft. Hier zu einigermaßen zuverlässigen Aussagen zu kommen erfordert sowohl ein umfangreiches Verständnis über den Gesamtzustand der Software als auch belastbare und komplette Anforderungen und ein gutes Verständnis derer. Beide Komponenten sind in Removing LUI problematisch: Auf der einen Seite ist nicht anzunehmen, dass eine einzelne Person (oder eine kleine Gruppe von Personen) wirklich alle Komponenten in ausreichender Detailtiefe kennt. Auf der anderen Seite sind mindestens Teile der Projektziele von Removing LUI insofern offen, als dass sie explorative Anteile enthalten. Es ist also insofern nicht verwunderlich, dass keine abschließende Gesamtschätzung über die benötigten Zeiten und Aufwände zur Verfügung stand und steht.

Eine Taktik, um mit dieser Unsicherheit umzugehen, ist, vor der eigentlichen Durchführung des Projekts von entsprechenden Wissensträger*innen eine Untersuchung und Schätzung der Aufwände zu erbitten. Dies wurde, insofern möglich, im Removing LUI-Projekt durchaus erbeten. Die Taktik stößt insofern an ihre Grenzen, als dass bei vielen Entwicklungsaufgaben die Analyse zur Ermittlung der Aufwände nicht klar von der Entwicklung einer Lösung zu unterscheiden ist. Keine solide Aufwandsschätzung funktioniert ohne Lösungsskizze, je mehr die Skizze ein detaillierter Plan ist, desto deutlicher werden die Aufwände. Hier existiert also ein Grundkonflikt zwischen der Frage, wie viel Aufwand vor einer Umsetzung in Analyse gesteckt werden soll und ab wann denn nun eigentlich die

Umsetzung beginnt. Für das Removing LUI-Projekt, und wohl auch für ähnliche Projekte in der ILIAS-Community, stellt sich die zusätzliche Schwierigkeit, dass die entsprechenden Analysen wahrscheinlich zu großen Teilen von den Service Providern zu tragen wären. Es ist, angesichts der vorgefundenen Situation im Funding, nur schwer vorstellbar, dass sich viele Fundinggeber für Aufträge finden lassen würden, die lediglich eine Abschätzung von Aufwänden für spätere Beauftragungen finanzieren würden. Dies ist in einem gewissen Sinne auch inhaltlich absolut nachvollziehbar: Was bringt das Verständnis von Umfängen, wenn am Ende dann nicht angefangen wird?

Eine andere Taktik, um mit der Unsicherheit umzugehen, ist, das Projekt "agil" zu gestalten. Abgesehen von den vielen Methoden und Ideen, die sich hinter dem Buzzword verbergen, ist die Grundidee einfach: Ein Projekt, das in kleinen Iterationen vorgeht und möglichst nach jeder Iteration einen nutzbaren neuen Zustand erzeugt, muss nicht unbedingt vorher wissen, welche Gesamtaufwände entstehen. Auch, wenn ein "agiles" Vorgehen zu keinem Zeitpunkt im Projekt explizit thematisiert wurde, ist diese Art von Vorgehen schon alleine durch die organisatorische und finanzielle Struktur eigentlich vorgegeben. Es ist kaum möglich, Abschätzungen über Gesamtumfänge zu erstellen, selbst wenn diese vorhanden wären ist eine Gesamtfinanzierung der Aufwände kaum vorstellbar. Prioritäten und Fokus in einer offenen und vielfältigen Community über lange Zeiträume aufrechtzuerhalten ist anstrengend. Für jährliche Releases müssen nutzbare Iterationen erzeugt werden.

Eine Unsicherheit über letztliche Umfänge und Aufwände ist, verbunden mit der teilweise darüber vorhandenen Unzufriedenheit bei verschiedenen Projektbeteiligten, also etwas, das die ILIAS-Community aufgrund ihres Entwicklungs- und Finanzierungsmodells bei Projekten wie Removing LUI in Kauf nehmen muss. Sie ist letztlich Resultat des offenen Modells der Open Source Softwareentwicklung und gehört daher zu anderen, wünschenswerten, Qualitäten dieses Modells.

Geister der Vergangenheit

Ein ebenfalls immer wieder kehrendes Problem im Projekt waren die vielen Altlasten, die durch das breit angelegte Vorhaben ans Licht gezogen wurden. Die Suche in der Codebase nach einer bestimmten Klasse ist leicht. Aber teilweise schloss sich dann die Frage an: Wo ist diese Stelle denn nun genau über die Oberfläche zu finden? Und was soll das da eigentlich machen? Auch wenn diese Art von Fund nur wenige Stellen betraf und die Antworten auf diese Frage meistens vollkommen klar



waren: Teile dieser Stellen hatten es in sich und zeigten Stellen in ILIAS, die schon lange nicht mehr Ziel einer Be- oder Überarbeitung waren.

Um ein besonderes Beispiel zu nennen: Die ILIAS-SCORM-Komponente verfügt über ein Debugging-Tool, das vielen nicht bekannt sein dürfte. Das wiederum verwendet, ob aus Alters- oder sonstigen Gründen, eine ganz eigene Logik um die Oberfläche zu erstellen. Und soll trotzdem nun eigentlich zu den neuen UI-Komponenten portiert werden. Wie kann nun mit diesem Fall umgegangen werden, welche Mühen sind notwendig und angemessen, wer finanziert?

Auch auf konzeptueller Ebene förderte das Projekt Altlasten oder Leerstellen zu Tage. Hier ist zum Beispiel die "Toolbar" zu nennen, in der verschiedenste Buttons, Links oder sogar Eingabeelemente ihren Platz gefunden hatten. Anspruch aus Perspektive der neuen UI-Komponenten ist, dass die verschiedenen Komponenten starke semantische Beschreibungen bekommen, um Einsatzzwecke, Gestaltungen und weitere Eigenschaften davon ableiten zu können. Die war für die Toolbar in Teilen überraschend schwierig: Welche Gemeinsamkeiten haben die Komponenten, die in der Toolbar auftauchen? Worauf beziehen sie sich? Können sie gruppiert werden. Dies kann insofern als konzeptionelle Altlast bezeichnet werden, als dass diese Frage bisher anscheinend nicht geklärt wurde. Auch lassen sich konzeptionelle Leerstellen finden, die gefüllt werden wollte. Eine Komponente wie der Launcher war vorher ein besonders genutztes Formular und konnte in diesem Sinne überhaupt nicht identifiziert werden.

Es ist weder überraschend noch irgendwem anzulasten, dass bei einem Projekt wie Removing LUI Altlasten gefunden und bearbeitet werden wollten, stattdessen war dieser Teilaspekt im Projekt durchaus angelegt, überraschend war allerdings die Menge und teilweise auch Art dieser Altlasten, die eine sinnvolle Lösungsfindung oft schwierig machten.

Unklare Haltung der Community

Die zweifelnden oder ablehnenden Stimmen zu Zielsetzung und Durchführung des Projekts waren ein stetiger Begleiter der Tätigkeit über die gesamte Projektlaufzeit. Es liegt wohl in der Natur der Sache, dass solche Stimmen stets lauter sind oder lauter wahrgenommen werden als positive Stimmen zum Projekt. Es liegt ebenfalls in der Natur der Sache, dass entsprechende Stimmen in einer vielfältigen Community vorhanden sind. Irritierend war dieser Zustand trotzdem, da teilweise schwer zu verstehen ist, wie groß welche Unzufriedenheit wirklich ist oder ob sich in einer Stimme eine generelle

Unzufriedenheit ist, also, inwiefern die Unzufriedenheiten perspektivisch oder akut bearbeitet werden müssen.

Überraschend deutlich und damit auch hilfreich war die von Oliver im Beirat durchgeführte Umfrage zum Projekt und zu Großprojekten allgemein. Bei 25 teilnehmenden Organisationen ergab sich ein überwiegend positives Bild von Removing LUI und Großprojekten im Allgemeinen, auch wenn im Einzelnen natürlich verschiedene Punkte kritisch gesehen werden. Leider führte die Umfrage bisher nicht zu einer klaren Positionierung des Beirats als Gremium zum Projekt oder Großprojekten im Allgemeinen, so dass hier eine Chance für gemeinsame Klarheit leider aussteht.

Auch bei diesem Thema ist anzunehmen, dass sich an der unklaren Haltung der Community zum Projekt und auch weiteren Projekten wahrscheinlich leider nichts ändern wird. Es ist jedoch ebenfalls festzuhalten, dass diese unklare Haltung der Community bzw. die lauten ablehnenden Stimmen einzelner, zu weiteren Reibungen im Projekt führen. Auf der einen Seite entstehen so Motivationsprobleme und Zweifel am Gesamtvorhaben im Projektteam. Auf der anderen Seite werden materielle Ressourcen, z.B. in Form von Zeit des Projektmanagers, auf die vorseilende Bearbeitung möglicher Unzufriedenheiten aufgewendet. Auch wenn diese unklare Haltung oder unzufriedene Stimmen wohl in der ILIAS-Community dazu gehören und Kritik, Nachfragen und abweichende Meinungen gut sind und zum Prozess dazugehören, muss auch klar sein, dass sie Ressourcen von der eigentlichen Projektstätigkeit abziehen.

Geringes Funding des Vereins

Kehrseite der beschriebenen Schwierigkeiten beim Funding ist die generelle Finanzierungssituation des Vereins. Über das Vereinsbudget bekommt das Technical Board ein Budget zur Verbesserung der Nachhaltigkeit der Software zur Verfügung gestellt. Dieses Budget wurde, wie beschrieben, in den Jahren '23 und '24 fast komplett in das Removing LUI-Projekt investiert und über die Fundingsuche verfünffacht. Trotzdem stellt das Budget nur einen Bruchteil dessen dar, was am Ende alleine für das Projekt Removing LUI nötig gewesen sein wird. Auch wird das Projekt nicht das letzte Nachhaltigkeitsprojekt in der Software geblieben sein.

Die aktuelle Strategie des Vereins zur Verbreiterung der Fundingbasis ist vor allem die Gewinnung neuer Mitglieder und nicht die Erhöhung von Beiträgen. Gleichzeitig existiert eine Strategie der Offenheit aller Angebote des Vereins, so dass die allermeisten Möglichkeiten auch von nicht-zahlenden Community-Mitgliedern oder auch Organisationen und Personen, die überhaupt nicht

sichtbar werden, genutzt werden können. Die Vergangenheit zeigt, dass diese Strategie höchstens ein langsames Wachstum der Mitgliedszahlen und damit auch des Vereinsbudgets bewirkt.

Mit der aktuellen Vereinsstruktur und den aktuellen Mitgliedern scheint es schwer vorstellbar, hier eine andere Strategie und damit auch ein deutlich erhöhtes Vereinsbudget zu erreichen. Gleichzeitig zeigt der Finanzierungsbedarf von Removing LUI, dass der Verein eigentlich deutlich erhöhte Budgets benötigen würde, wenn er der Aufgabe einer nachhaltigen Pflege der Software gerecht werden soll. Wie oben argumentiert sind die fehlenden eigenen Budgets außerdem ein wesentliches Hindernis, Projekte mit zuverlässiger Zeit- und Finanzplanung durchzuführen.

Wenn also gefordert wird, der Verein möge sich um die nachhaltige Entwicklung der Software kümmern und dies mit belastbaren Zeit- und Finanzplänen durchführen, muss auch die Frage beantwortet werden, wodurch der Verein ausreichend eigene Budgets bekommen kann, um dies auch zu leisten. Wird diese Frage nicht beantwortet kann entweder die Software nicht nachhaltig gepflegt werden oder entsprechende Projekte werden wahrscheinlich wieder ähnliche Formen haben, wie Removing LUI. Angesichts des Gesamtzustands der Software (nicht dramatisch, aber renovierungsbedürftig) und der wachsenden Anforderungen an die Software aus verschiedensten Bereichen, rät das TB allen Vereinsmitgliedern das Thema "Wie kann der Verein die Software nachhaltig finanzieren?" auf die Agenda der Community zu setzen.

Fazit

Weiterführung "Removing LUI"

Das Technical Board der Wahlperiode '23-'25 spricht keine eigene Empfehlung zur Weiterführung des Projektes aus. Es sind verschiedene Möglichkeiten zu Form und Fokus einer Fortführung des Projektes denkbar und wir halten es für sinnvoll, wenn diese Entscheidungen von einem neuen Technical Board getroffen werden. Die Spannweite der Möglichkeiten scheint dabei ziemlich weit:

1. Beendigung des Projekts mit diesem Bericht, Aufgabe der Ziele
2. Beendigung des Projekts mit diesem Bericht, Bitte die Ziele ohne Projekt weiter zu verfolgen
3. Weiterführung des Projekts in einer reduzierten Form, z.B. als Fundraiser



4. Weiterführung des Projekts in seiner bisherigen Form

Die Diskussionen im aktuellen TB bewegten sich meistens ab etwa Variante 3., auch 4. wurde durchaus erwogen. Eine abschließende Meinung haben wir uns nicht gebildet und bieten dem neuen TB an, unsere Gedanken und Erwägungsgründe zu den verschiedenen Varianten in einem Meeting zu teilen.

Womit wir uns jedoch ziemlich sicher sind: Selbst wenn das Projekt aufgeben wird, hat der Vorgang, die Legacy UI durch neue UI-Komponenten im UI-Framework abzulösen, jetzt schon einen Stand erreicht, bei dem eine gewisse Sogwirkung zu erwarten ist. Die Qualität, nicht zuletzt für die Entwickler*innen, spricht für einen Einsatz von neuen UI-Komponenten für neue Features. Im Bereich der Pflege und Weiterentwicklung haben die neuen UI-Komponenten deutliche Vorteile, nicht zuletzt, weil Personen zur Verfügung stehen, die die Komponenten warten wollen. Wir gehen daher davon aus, dass das Removing LUI-Projekt, selbst wenn es nicht formal weitergeführt wird, einen wertvollen Beitrag zur Umsetzung der gesteckten Ziele geleistet hat und weiter leisten wird.

Empfehlungen für weitere Großprojekte

Angesichts des Gesamtzustands der Software, aber auch angesichts der positiven Stimmung gegenüber Großprojekten, die in der Beiratssitzung geäußert wurden, gehen wir davon aus, dass Removing LUI nicht das letzte Großprojekt zur Pflege des Softwarequerschnitts gewesen sein wird. Vorstellbar sind hier zum Beispiel Projekte von ähnlichem Zuschnitt im Bereich Security, Privacy oder Webservices.

Was auch immer die genauen Ziele eines nächsten Projektes sein werden, oder sei es nur die Fortführung dieses: Wir raten jedem Projektträger, die Durchführung und Ergebnisse von Removing LUI aufmerksam zu betrachten und zu bewerten und für das neue Projekt entsprechende Schlüsse zu ziehen. Hierbei sind die Rolle des Projektmanagers und die verwendeten Strategien und Taktiken in der Projektdurchführung genauso relevant, wie die Begrenzungen, die bei diesem Projekt deutlich geworden sind.

Unabhängig von neuen Großprojekten empfiehlt es sich für die Gesamtcommunity, an der generellen Projektfähigkeit der ILIAS-Community für ähnliche Projekte zu arbeiten. Dazu gehört eine offene Diskussion und gegebenenfalls Änderung der Finanzierungsbasis des ILIAS-Vereins in Verbindung mit den Aufgaben in der Nachhaltigkeit, die übernommen werden sollen. Genauso gehört eine Verbesserung des Verständnisses von gemeinsamer Arbeit in der Community und Arten und



Anlässen, wie Kritik und Zweifel geäußert werden, dazu. Schließlich empfiehlt sich, die Rolle der Projektmanager*innen und Fundraiser*innen in der Community zu schärfen und hier neue Personen auf die Übernahme solcher Aufgaben in kleineren und größeren Projekten vorzubereiten.

Removing LUI verspricht, ILIAS mit einer modernisierten UI, verbesserter Usability und User Experience auszustatten. Verbessert werden auch die Bereiche Barrierefreiheit und Performance und das Projekt stelle eine Investition auch in Security-Eigenschaften der Software dar. Pflege einer alten und großen Software in dieser Form ist unumgänglich und muss laufend geleistet werden, als Daueraufgabe im Prozess, aber auch in Form gezielterer Projekte. Diese Pflege stellt keinen Gegensatz zur funktionalen Weiterentwicklung der Software dar, sondern ist Basis und Bedingung dafür, dass diese dauerhaft gelingen kann. Lasst uns, auch im Sinne des ILIAS-Manifestes, weiterhin wagen unsere Software und Community in dieser Form auf die Zukunft vorzubereiten. ILIAS wird gebraucht.

Haben Sie Fragen?
verein@ilias.de

Geschäftsführer:
Matthias Kunkel
VR 16017 / Amtsgericht Köln
USt-ID: DE 815 319 517

Vorstand:
1. Vorsitzender: Oliver Samoila
2. Vorsitzende: Yvonne Fischer
Schatzmeister: Thomas Schroeder

Bankverbindung:
IBAN: DE26 3705 0198 1902 5687 14
SWIFT-BIC: COLSDE33
Sparkasse KölnBonn