# Richtlinie zum „Software-Sprint“

# VICollab – Zugängliches kollaboratives Arbeiten mit Emacspeak

## *Schlussbericht*

Zuwendungsempfänger:

Volz, Michael

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS23S20 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

## Kurze Darstellung der Aufgabenstellung und Motivation

Ziel des Projekts war es, web-basierte Kollaboration mit BigBlueButton für blinde Menschen zugänglich zu machen. Dabei wurde auf Emacspeak als Assistenz-Technologie gesetzt, eine freie Plattform für Sprachausgabe auf Basis des stark erweiterbaren und automatisierbaren Emacs-Editors. Das Ökosystem um Emacs und Emascpeak erschließt Menschen mit Sehbehinderung zahlreiche digitale Anwendungen und Systeme. Darauf aufbauend entwickelten wir einen spezialisierten Client für BigBlueButton, der den Nutzenden zielgerichtete Interaktionen für Online-Kollaboration bereitstellt, auf freier Software basiert, plattformübergreifend ist und beliebig konfiguriert und an die persönlichen Bedürfnisse angepasst werden kann.

Ich hatte während der Projektphase Unterstützung von mehreren Personen, die auf konzeptioneller Ebene, beim User-Testing und punktuell bei der Entwicklung mitgewirkt haben. Das Projekt war eine Teamleistung und wir hoffen, die gemeinsame Arbeit auch nach der Förderphase fortsetzen zu können. Auf technischer Ebene bestand das Projekt aus folgenden Meilensteinen:

1. Meilenstein: Wrapper um HTML5-Client sowie wiederverwendbare emacs-Bibliothek zur bidirektionalen Interaktion mit dem DOM oder dem accessibility tree.
2. Meilenstein: Emacs-Paket zur Fernsteuerung (join, mute/umute, raise hand, etc.) von BigBlueButton-Konferenzen.
3. Meilenstein: Abbilden der Chatfunktion von BBB in Emacspeak.
4. Meilenstein: Abbilden der geteilten Notizen aus BigBlueButton in Emacspeak

## Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Sogenannte Behinderungen bedeuten für betroffene Personen in unserer Gesellschaft eine erhebliche Mehrbelastung und häufig den Ausschluss aus bestimmten Kontexten. Unser Projekt hat sich mit den Schwierigkeiten, die eine Beeinträchtigung des Sehvermögens mit sich bringt, beschäftigt. Gerade digitale Systeme sind häufig nicht inklusiv entworfen und verlassen sich auf den visuellen Kanal für das User-Interface. Betroffene müssen auf Hilfstechnologien wie Screen-Reader zurückgreifen, die aber nicht dieselbe User-Experience bereitstellen können, wie integrierte Lösungen.

Kollaboratives Arbeiten kann eine sehr effektive und kreative Form der digitalen Zusammenarbeit sein, die immer mehr Organisationen und Gruppen erfolgreich nutzen. Die Software BigBlueButton findet beispielsweise in Bildungseinrichtungen und Verwaltungskontexten breite Verwendung. Sie unterstützt zwar Technologien für die Interaktion mit Screen-Readern, diese stellt aber nicht für alle Betroffene eine zufriedenstellende Lösung dar. Insbesondere textbasierte Kollaborationsformen wie die geteilten Notizen und Einzel- sowie Gruppenchats stellen eine Herausforderung dar. Sehenden Menschen hilft bei diesen der visuelle Kanal, um parallel und zeitgleich zu einer laufenden Konferenz stattfindende Ereignisse zu erfassen.

Unser Projekt ermöglicht blinden Menschen die Nutzung dieser Plattform unter Zuhilfenahme eines erweiterbaren und automatisierbaren Editors mit Ton- und Sprachausgabe. Über diesen direkten Nutzen hinaus wurde User-Testing durchgeführt um die Möglichkeiten von digitaler kollaborativer Arbeit jenseits eines visuellen User-Interfaces auszutesten und neue Ansätze zu entwickeln.

## Ausführliche Darstellung der Ergebnisse

Während der Projektphase sind zwei Softwarepakete entstanden: Zum einen das Programm, das durch das Kapseln eines Webbrowsers ohne sichtbares User-Interface die Webanwendungen ausführen kann und eine klar definierte Schnittstelle zu deren Fernsteuer bereitstellt. Dieses Programm wurde mit Blick auf BigBlueButton entworfen und implementiert, ist aber auch für andere Webanwendungen erweiterbar. Diese Software-Komponente habe ich quickbeam genannt.

Zum anderen wurde ein Paket für Emacs entwickelt, dass die von quickbeam bereitgestellte Schnittstelle benutzt. Mittels dieses Pakets kann die Benutzer:in sich in eine BigBlueButton-Konferenz einwählen, sich zu Wort melden, das Mikrofon steuern sowie Chatnachrichten lesen und schreiben. Auch dieses Paket, das quickbeam.el heißt, bringt zunächst spezifische Befehle und Automatisierungen für BigBlueButton mit, kann aber in Zukunft für andere Kollaborations-Plattformen und Webanwendungen im Allgemeinen erweitert werden.

Neben den direkten Ergebnissen in Form von Softwarecode hat das Projekt interessante und weitreichende Ergebnisse und Erkenntnisse hervorgebracht. Um nur die wichtigsten zu nennen, seien erwähnt:

- Umfassende Erkenntnisse im spezifischen Bereich von Projektmanagement für Softwareentwicklung und Accessibility
- Die massive Bedeutung von Möglichkeiten der Automatisierung im Umgang mit Webinhalten für Menschen mit und ohne Behinderung

- Das Konfliktfeld von Automatisierbarkeit gegenüber Mal-Usage und Missbrauch mit Fokus auf UX und Accessibility
- Konzepte für die Implementierung von Multi-Interface Zugängen
- Kritische und äußerst relevante Grenzen der WCAG für den Bereich Accessibility für Web-Usage und Web-Interaction
- Technische Unterschiede verschiedener Web-Browser, die Abhängigkeit der User davon und die drastischen Konsequenzen für eine barrierefreie Nutzung von Webinhalten

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Mit den Ergebnissen des Projekts ergeben sich verschiedene Nutzen für die Zielgruppe. Nutzer:innen können die implementierten Funktionen nutzen und so an einer BBB-Konferenz teilnehmen, ohne auf die Navigation im Browser mittels eines visuellen Interfaces angewiesen zu sein. Die Steuerung der interaktiven Objekte wird akustisch bestätigt, so dass eine Nutzung lediglich durch Tastatureingaben und akustischem Feedback umsetzbar ist.

Darüber hinaus gibt das Projekt durch die Nutzung Einblicke in Funktionsweisen abseits eines visio-normativen Nutzungsschemas. So werden Nutzer:inen für verschiedene Bedien- und Interaktionswege sensibilisiert. Die Darreichung im Rahmen einer Open-Source-Lizenz ist für die Vermittlung der Ergebnisse in der Nutzergruppe essentiell. Nicht nur kann so umfassende Zugänglichkeit zu allen Ergebnissen garantiert werden, sondern es ergibt sich ebenfalls die Möglichkeit die Ergebnisse zu prüfen und nach eigenen Vorstellungen ab- oder umzuwandeln.

Nur durch Open-Source als elementarem Bestandteil der Vermittlung erhalten Nutzer:innen die Möglichkeit an der Weiterentwicklung direkt teilzunehmen und ebenso von einer schnellen und engagierten Weiterentwicklung der bisherigen Ergebnisse zu profitieren.

Abseits des Projektergebnisses in Form von Software, hat unser Team durch die Projektphase, ihre Struktur und Verbindlichkeiten, extrem hilfreiche Lernergebnisse hervorgebracht. Als einflussreichste wären zu nennen: Reflexion von Selbst- und Zeitmanagement, Wissen über Community-Building, Projektorganisation, Ablaufplanung Teamarbeit und Ressourcenplanung.

Die Begleitung der Projektphase durch die OKF hat die Lern- und Leistungsmöglichkeiten unseres Teams massiv und spezifisch gefördert. Das vermittelte Coaching war ein wesentlicher Triebfaktor in unserem Vorankommen innerhalb der letzten sechs Monate. Durch die organisierten Events und die Angebote von Austausch und Vernetzung sind starke und äußerst hilfreiche Verbindungen zu anderen Teilnehmenden des Software Sprints entstanden. Diese Vernetzungsleistung hat für Informationen, kreativen Austausch und diverse andere Formen der Unterstützung gesorgt und wird auch über die Projektphase hinaus Bestand haben.

Durch die zeitliche Struktur, die von der Projektbegleitung etabliert wurde, sind wesentliche Verbindlichkeiten und wichtige Strukturen für die Arbeitszeitgestaltung entstanden. Dies hat wichtige Impulse für unsere Arbeit gesetzt.

## Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Am Anfang der Projektphase stand die Evaluation verschiedener technischer Ansätze, um das Ziel zu erreichen. Dabei wurde klar, dass ein Nachbau des BigBlueButton-Clients sowohl auf Grund des damit verbundenen Aufwands als mehr noch, weil die verwendeten Schnittstellen weder öffentlich dokumentiert noch stabil sind, nicht durchführbar war. Ich habe mich in Folge entschieden, den offiziellen Webclient zu nutzen und fern zu steuern.

Diese Architektur umzusetzen war im ersten Drittel des Projektes von erheblichem Lernaufwand begleitet. Es kamen Software-Bibliotheken zum Einsatz, die wir nicht kannten und die auch mehrmals Unzulänglichkeiten gezeigt haben. Wir haben zweimal zentrale Bibliotheken gewechselt und mussten von Chromium auf Firefox umsteigen, weil sich im Projektverlauf herausstellte, dass die Sicherheitsarchitektur von Chromium unser Vorhaben unmöglich machte.

Die Umsetzung der gemeinsamen Notizen, einer Funktion von BigBlueButton, bei der alle Konferenzteilnehmenden gleichzeitig ein gemeinsames Textdokument bearbeiten können, hat uns mehrfach vor Herausforderungen gestellt. Beispielsweise hat sich herausgestellt, dass durch die Art, wie das User-Interface des Webclients von BigBlueButton implementiert ist, es nicht zuverlässig möglich ist, kontinuierlich Veränderung von anderen Benutzer:innen im User-Interface selbst zu verfolgen. Dieser Ansatz musste aufgegeben werden. Danach haben wir versucht, die Kommunikation zwischen Client und Server per Websocket mitzuverfolgen und darauf zu reagieren. Aber auch dieser Weg wurde durch die Architektur des Clients kompliziert und scheiterte am Ende auf Grund des Aufwands.

## Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wir haben ein Projekt auf der Open-Source-Platform Sourcehut: https://sr.ht/~michl/VICOllab. Dort ist der Quelltext beider Softwarepakete zu finden sowie eine Mailingliste zur Kontaktaufnahme und die Möglichkeit, Fehler zu melden. Wir planen, die Dokumentation für neue Benutzer:innen noch zu verbessern und eine Webseite für den schnellen Einstieg zu erstellen. Diese existieren aber noch nicht.

## Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Konzeption und Implementierung des Wrappers um den HTML5-Client haben sich als aufwändiger als geplant herausgestellt. Wie beschrieben hing das auch damit zusammen, dass wir mit einigen Softwarebibliotheken nicht vertraut waren und unsere Einschätzung des Umfangs und der Qualität dieser Bibliotheken nicht adäquat waren.

Die Architektur des HTML5-Clients stellte uns auch vor nicht-geplante Herausforderungen. Als Beispiele seien hier fehlende feste Identifikatoren für Elemente des Interfaces genannt, sowie die kontinuierliche Umgestaltung des Dokumentenbaumes bei Interaktions-Ereignissen. Letztere manifestiert sich sogar bis in immer wieder neu hergestellte und geschlossene Websocket-Verbindungen. Dies kostete uns vor allem bei der Umsetzung von hoch-interaktiven Funktionen wie dem Chat und der geteilten Notizen mehr Aufwand als eingeplant.

Das Umsetzen der geteilten Notizen scheiterte am Ende, wie bereits oben erwähnt, wegen der sehr hohen Komplexität der Aufgabe selbst und der Verzögerung auf Grund der genannten Punkte früher im Projekt.

## Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während der Projektphase wurde eine neue Version von BigBlueButton veröffentlicht. Bei dieser haben sich einige Elemente im Interface verändert, sowohl in der Anordnung als auch bei der technischen Referenz darauf. Wir konnten diese Änderungen aber in das Produkt integrieren.