

1.1 Motivation

Jährlich investieren Unternehmen viel Geld in die Verbesserung der kommunikativen Abläufe zwischen den Arbeitern. Denn die Kommunikation zwischen den Arbeitern steht in direkter Verbindung mit dem Thema Kooperation. Damit das Produkt eine bessere Qualität aufweist und bei der Entwicklung wenig Ressourcen verbraucht werden, ist Kooperation sehr wichtig. Um diesen wirtschaftlichen Vorteil gewährleisten zu können, bieten viele Unternehmen ihren Arbeitern spezielle Schulungen und Unternehmensstrukturen an. Für bessere Kooperation müssen sich die Mitarbeiter an einem Tisch zusammensetzen, um über Probleme und deren Lösungen zu diskutieren. Diese Diskussionen finden in täglichen Meetings statt. Ein Meeting muss nicht für eine Diskussion stattfinden, sondern auch um Ideen auszutauschen und den aktuellen Stand der Entwicklung zu berichten. Bei Projektleitern und Managern, aber auch Mitarbeitern aus den verschiedenen Bereichen, gehören Meetings sogar zur wichtigsten Beschäftigung. Diese Personen verbringen im Schnitt zwei Stunden und 45 Minuten pro Meeting und sitzen durchschnittlich sieben mal die Woche in Besprechungen.

1.2 Ziel der Arbeit

Ziel dieser Arbeit ist die Entwicklung einer Webapplikation, die entsprechend der Probleme aus dem letzten Abschnitt als zentrales Plattform für die Erstellung, Bearbeitung, Verwaltung und Ausführung von Meetings im Webbrowser dienen soll. Da die Webanwendung zur Verwaltung von Meetings und Sitzungen dienen soll und eine Agenda erstellt werden soll, trägt diese Applikation den Namen *SkyGate Agenda*.

Die Webanwendung SkyGate Agenda wird unter Anwendung der Skriptsprache PHP bzw. das in PHP implementierten Webframeworks Symfony implementiert. PHP ist eine funktionale, aber auch objektorientierte Skriptsprache, welche zur serverseitigen Entwicklung von Webanwendungen dient (eine ausführliche Erklärung für PHP erfolgt in Abschnitt 3.2.2). Das in PHP geschriebene und für die Entwicklung konzipierte quelloffene Symfony-Webframework (siehe Abschnitt 3.2.3) folgt dem Model-View-Controller-Schema (MVC).

Auch soll ein Teil des Ergebnisses auf einem Arduino visuell dargestellt werden. Ein an den Arduino angeschlossenes LCD-Display soll den aktuellen Agendapunkt des Meetings dem User und den Teilnehmern zeigen. Dieses Arduino soll somit immer bei einem Meeting benutzt werden.

Die Entwicklung der Webapplikation SkyGate Agenda soll daher zur effizienteren und kostengünstigen Meetings dienen.

1.3 Struktur der Arbeit

Die vorliegende Arbeit ist wie folgt strukturiert: In Kapitel 2 werden zunächst die Probleme und Fehler, welche während eines Meeting entstehen können, analysiert. Danach wird das theoretische Wissen zur Verbesserung dieser Fehler näher erläutert. Kapitel 3 befasst sich mit den notwendigen Grundlagen zur Erarbeitung dieser Arbeit. Diese umfassen die Grundlagen von Webapplikationen, wie diese funktionieren und ihren Eigenschaften. Die serverseitige Programmierung mit PHP, eine grundlegende Beschreibung des Frameworks Symfony detailliert erklärt. Auch die verwendete Entwicklungsumgebung *PHPStorm* und die benutzte Hardware Arduino werden in diesem Kapitel näher erläutert. In Kapitel 4 wird die Anforderung an das Gesamtsystem beschrieben. Dieses Kapitel beschäftigt sich mit der Funktion der Anwendung und der Anforderung an die Benutzeroberfläche. In Kapitel 5 wird die Implementierung der Webanwendung behandelt. Hier wird der Aufbau der Anwendung, die Benutzeroberfläche für den User und die Funktionalität der Anwendung näher beschrieben. Im sechsten und letztem Kapitel werden die Ergebnisse zusammengefasst und ein Fazit gezogen.

Die Eigenschaften und Konzepte einer Webapplikation

Eine Webapplikation basiert auf das Client-Server-Modell. Das Besondere an dem Modell ist, dass die Anwendung nicht lokal auf dem Rechner installiert und ausgeführt wird. Stattdessen ist die Anwendung auf einem Server installiert. Da durch findet die Verarbeitung und Auswertung der Daten

in dem Webserver statt. Auf dem lokalen Client-Rechner des Benutzers werden nur die Ausgabe der verarbeiteten Daten übertragen. Für die Ausgabe der Daten wird ein Webbrowser benötigt. Der Webserver dient einerseits zur Kommunikation zwischen dem Client und dem Server und andererseits auch zur Darstellung der Ergebnisse.

Zum Ausführen einer Webapplikation ist kein bestimmtes Betriebssystem nötig, denn eine Webapplikation ist plattformübergreifend. Dies ist auch ein großer Vorteil einer Webapplikation, denn so kann sie auf jedem Betriebssystem ausgeführt werden.

Jedoch kann es sein, dass ein bestimmter Webbrowser benötigt wird und manche Laufzeitumgebungen (z.B. Flash, JavaScript....) müssen auf dem Client-Rechner installiert sein. Laufzeitumgebungen nehmen dem Server Arbeit ab und beschleunigen damit den Prozess der Verarbeitung der Daten. JavaScript ist eine sehr verbreitete Laufzeitumgebung. Sie kann Aufgaben direkt auf dem Client-Rechner ausführen. JavaScript wird meist bei der Validierung von Benutzereingaben ausgeführt. Somit müssen die Daten nicht an den Webserver gesendet werden, sondern die Eingabe wird lokal auf Fehlern überprüft. Durch die Verbreitung internetfähiger, mobiler Geräte gewinnen Webapplikationen immer mehr an Bedeutung und Wichtigkeit.

Die allgemeine Funktionsweise einer Webapplikation

Eine Webanwendung wird durch das Eingeben der URLs des Webserver in einem Browser gestartet. Durch diese Eingabe wird eine Anfrage an den Server gesendet. Die Anfrage wird von dem Server angenommen und an die Anwendung weitergegeben, welcher dann eine HTML-Seite generiert und diese dann über den Webserver an den Clienten zurück schickt. Der Client zeigt schließlich diese HTML-Seite im Webbrowser an. Diese Art von Webseiten werden als statische Seiten bezeichnet.

Im Gegensatz zur statische Webseiten existieren auch dynamische Webseiten. Hierbei wird der Inhalt der Webdokumente in Abhängigkeit von Benutzeranfragen modifiziert[1]. Die Modifizierungen können durch das Klicken eines Hyperlinks oder das Ausfüllen und Absenden eines Formulars auf der Webseite stattfinden. Durch das Absenden an den Webserver werden die Eingaben durch die Webanwendung verarbeitet. Typischerweise gehört zu den Funktionen vieler Webanwendungen das Speichern der Daten die während der Benutzung in einer Datenbank. Diese Daten werden serverseitig durch Datenbankserver gespeichert. Diese Datenbankserver halten dann die Daten für die Webanwendung bereit. Ein typischer Einsatzbereich wäre das Speichern der Daten eines Benutzer bei der Registrierung auf einer Webseite, welcher durch den Webserver erstellt wurde.

Das 3-Schichten-Modell

Im generellen wird eine Webanwendung in 3 Schichten geteilt. Hierbei ist die erste Schicht die Präsentationsschicht. Hier findet das Senden der Anfrage des Clienten an den Webserver und das Darstellen der HTML-Seite als Antwort im Webbrowser. In der Anwendungsschicht werden die Anfragen des Clienten vom Webserver angenommen und verarbeitet. Falls als Antwort eine statische Seite geliefert werden soll, greift der Webserver auf das File-System, wo die statischen Seiten gespeichert sind und gibt die richtige Seite an den Webbrowser des Clienten zurück. Bei einer dynamischen Seite als Antwort greift der Webserver auf die Webanwendung. Die Webanwendung wertet die Eingaben des Clienten aus, bearbeitet sie und sendet eine dynamische HTML-Seite an den Webbrowser des Clienten. Die letzte Schicht ist die Datenbankschicht. In dieser Schicht sind Daten und Datensätze gespeichert auf welche die Webanwendung zugreifen kann, falls dies nötig ist. Typischer Einsatzzwecke einer Datenbank sind das Speichern von Userdaten oder Speichern von Shop-Elementen.

2. Probleme eines Meetings

2.1 Problemanalyse vieler Meetings

Die täglich stattfindenden Meetings in den Unternehmen dienen zur Besprechung des aktuellen Standes eines Projektes oder für Lösungsvorschläge von bestehenden Problemen in Projekten. Solche Meetings sind zeitintensiv und kosten das Unternehmen eine Menge Geld. Jeder Teilnehmer eines Meetings wird von der Arbeit zurück gehalten. Einige Studien beweisen, dass viele der Meetings die in Unternehmen gehalten werden, nicht effizient genug sind und es kein Ergebnis gefunden wird. Diese Aspekte von Problemen in Meetings werden in diesem Kapitel behandelt.

Eines dieser Probleme ist die fehlende Zeitstruktur in Meetings. Es wird zu viel Zeit in Meetings investiert. Ohne das Setzen eines Zeitlimits wird viel unwichtige Themen besprochen und dies kann zu endlosen Diskussionen führen. Ein anderes Problem in Meetings ist die fehlende Agenda. Bevor ein Meeting startet, wird es zwar ein Themengebiet für das Meeting bestimmt, jedoch fehlt trotzdem eine Struktur, die durch das ganze Meeting führt. Somit ist keine übersichtliche Agenda vorhanden. Es werden keine Prioritäten gesetzt und somit kann man sich nicht auf Wesentliches beschränken [3]. Auch bei einer übersichtlichen Agenda sollte man sich auf eine Zeitstruktur festlegen, damit unwichtige Themen bei unzureichender Zeit auf das nächste Meeting verschoben werden kann. Durch das Definieren einer Agenda hat man eine Orientierung im Meeting, weil viel kostbare Zeit durch orientierungslose Meetings verloren geht. In vielen Konferenzräumen stehen Whiteboards, Pinnwände oder Beamer, die kaum oder eher selten verwendet werden. Durch Visualisierung des Meetings anhand solcher Geräte könnte man ein Meeting strukturiert gestalten. Dies würde auch zur Orientierung zum Ablauf der Tagesordnungen [3] und einen Überblick über das

aktuelle Thema für die Teilnehmer verschaffen. In vielen Meetings werden auch oft Themen besprochen, welche nicht relevant für alle Teilnehmer sind [4]. So sitzen viele Mitarbeiter ohne einen wichtigen Grund in Meetings, welche irrelevant für sie sind. Viele dieser Teilnehmer nutzen einen Meeting zum Auffrischen von Kontakten oder zum Pflegen von wichtigen Beziehungen zur Entscheidungsträger [6]. Auch werden Meetings als Gelegenheit zur Selbstdarstellung um die Karrierechancen zu verbessern [6]. Ein anderer Grund für das Orientierungslosigkeit eines Meetings ist das Fehlen eines Moderators, welcher strikt die Agendapunkte durchgeht und wenig Raum für Diskussionen lässt [3]. Ein ganz anderer Aspekt ist, dass viele Sitzungen im „sitzen“ stattfinden und somit langweilig gestaltet sind und viele Teilnehmer müde durch das lange Sitzen werden. All diese Standpunkte führen dazu, dass ein Meeting nicht effizient genug ist und es viel sinnlose Zeit in Meetings verloren gehen.

2.2. Theoretisches Wissen zur Verbesserung von Meetings

Im letzten Kapitel wurden die Fehler aufgezählt, welche in Meetings gemacht werden können. Natürlich gibt es Maßnahmen und Techniken um solche Probleme beseitigen zu können oder dazu führen, dass solche Probleme gar nicht entstehen. In diesem Kapitel werden Lösungswege erläutert, wie solche Fehler und Probleme vermieden werden können.

„Ein altes Sprichwort sagt: „Die Dinge brauchen immer so lange, wie Zeit für sie vorhanden ist““ [3]. Also sollte man als erstes einen festen Zeitplan für das Meeting gestalten. Um dies präzise gestalten zu können, sollte man nicht nur ein Thema für das Meeting finden, sondern schon mehrere Agendapunkte definieren. Anhand dieser Punkte könnte man einen strukturierten Zeitplan erstellen. Dieser ist auch nötig, damit unnötige Diskussionen vermieden werden und alle Agendapunkte abgearbeitet werden. Eine Agenda kann auch dazu führen, dass unwichtige Themen nicht besprochen werden und es eine Orientierung im Meeting existiert. Somit kann auch gewährleistet werden, dass nur Teilnehmer zum Meeting eingeladen wurden, die die Themen auch betreffen. Dies führt dazu, dass die Kosten für das Meeting gesenkt werden und nicht betroffenen Mitarbeiter weiter an Ihren Projekten arbeiten können. Damit die Gliederungspunkte eingehalten werden können, sollte ein Moderator paar Tage vor dem Meeting bestimmt werden, welcher auch den Teilnehmern bekannt gemacht werden sollte. Der Moderator sollte Aufgaben wie das Abarbeiten der Tagesordnung, Moderieren des Meetings, Dokumentieren der Beschlüsse und Zeitmanagement übernehmen [4]. Zur Visualisierung eines Meeting sollten die verschiedensten Techniken eingesetzt werden, um Orientierung zum Ablauf der Tagesordnungsorgnen und einen Überblick über das aktuelle Thema für die Teilnehmer verschaffen [3]. Die Teilnehmer sollten vorab schriftlich informiert werden, welche Themen im Meeting besprochen werden. Eventuell könnte man eine Dokumentation per Mail an die Teilnehmer senden, mit den Themen des Meetings und den benötigten Unterlagen oder Informationen. Auch sollte eine Dokumentation während des Meetings erstellt werden, welcher die getroffenen Entscheidung und zu erledigenden Aufgaben beinhaltet. Am Ende eines Meetings könnte man diese Dokumentation an alle Teilnehmer schicken. Dementsprechend hätten alle Teilnehmer im Überblick welche Aufgaben sie erledigen zu haben [4]. Um die Meetingszeit zu verkürzen oder das „Aufweckt-Effekt“ auf die Teilnehmer wirken zu lassen und somit ihre Produktivität zu steigern, wäre auch eine Möglichkeit, die Sitzung im stehen zu gestalten [3].

All diese Punkte können dazu führen ein Meeting effizienter und kostengünstiger zu gestalten und viel schnellere und effektiver Ergebnisse zu liefern.

3.2 Für die Entwicklung eingesetzte Programmiersprachen

In diesem Abschnitt der Arbeit werden die benutzten Programmiersprachen für das Entwickeln der Webanwendung näher erklärt.

3.2.1 Entwicklung einer Webseite mit HTML und CSS

Jede Webseite im Netz wird in HTML geschrieben. HTML steht für die Abkürzung „HyperText Markup Language“. Sie stellt den Grundgerüst einer Webseite dar.

3.2.2 Entwicklung der serverseitigen Anwendung mit PHP

PHP ist eine Abkürzung für PHP: Hypertext Preprocessor und ist eine Skriptsprache mit einer an C und Perl angelehnten Syntax, die hauptsächlich zur Erstellung dynamische Webseiten oder Webanwendung verwendet wird. Die Sprache wurde 1995 von Rasmus Lerdor veröffentlicht und seit dem immer weiter entwickelt und erweitert. Mit PHP 3 wurde der Kern der Sprache komplett neu umgestellt. Ab PHP 5 wurde der Fokus der Sprach auf die objektorientierte Programmierung gelegt. Ein in PHP geschriebener Code wird serverseitig ausgeführt. Dazu wird der Code vom Clienten an einen Interpreter auf dem Webserver übermittelt. Der Webserver verarbeitet diese Daten und schickt das Ergebnis als Ausgabe an den Clienten zurück. Die Ausgabe kann eine PDF-Datei oder auch ein Bild sein, jedoch sind die meisten Ausgaben HTML-Dokumente, die im Webbrowser angezeigt werden. Für das Ausführen einer PHP-Datei durch einer Webanwendung, wird ein System benötigt, welches mit den Anweisungen in der Datei umgehen kann.

Um eine PHP-Datei im Rahmen einer [Webanwendung](#) ausführen zu können, benötigt man ein System, das mit den in der Datei enthaltenen Anweisungen umgehen kann. Aus diesem Grund wird durch eine Schnittstelle, wie [ISAPI](#) oder [CGI](#), der Interpreter von einem Server-[Daemon](#) oder Server-Dienst, wie [Apache](#) oder [IIS](#), ausgeführt.

Quellen:

- [1: https://www4.informatik.uni-erlangen.de/DE/Lehre/SS04/PS_KVBK/talks/Handout-Webanwendung-2.pdf]
- (2: http://praxistipps.chip.de/was-ist-html-verstaendlich-erklaert_40979)
- [3: <https://organisationsberatung.net/effiziente-meetings-besprechungen-sitzungen/>]
- [4: <http://www.faz.net/aktuell/beruf-chance/arbeitswelt/konferenzen-meetings-und-besprechungen-effektiv-gestalten-13694844.html>]
- [5: <http://www.business-netz.com/Zeitmanagement/Besprechungen-effizient-gestalten-auch-zeitlich>]
- [6: <http://www.business-wissen.de/artikel/besprechung-meetings-effizient-durchfuehren/>]
- [7: <http://karrierebibel.de/meeting-tipps/>]